Fast Order Basis and Kernel Basis Computation and Related Problems

by

Wei Zhou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2012

Abstract

In this thesis, we present efficient deterministic algorithms for polynomial matrix computation problems, including the computation of order basis, minimal kernel basis, matrix inverse, column basis, unimodular completion, determinant, Hermite normal form, rank and rank profile for matrices of univariate polynomials over a field. The algorithm for kernel basis computation also immediately provides an efficient deterministic algorithm for solving linear systems. The algorithm for column basis also gives efficient deterministic algorithms for computing matrix GCDs, column reduced forms, and Popov normal forms for matrices of any dimension and any rank.

We reduce all these problems to polynomial matrix multiplications. The computational costs of our algorithms are then similar to the costs of multiplying matrices, whose dimensions match the input matrix dimensions in the original problems, and whose degrees equal the average column degrees of the original input matrices in most cases. The use of the average column degrees instead of the commonly used matrix degrees, or equivalently the maximum column degrees, makes our computational costs more precise and tighter. In addition, the shifted minimal bases computed by our algorithms are more general than the standard minimal bases.

Acknowledgments

Thanks to my supervisor, George Labahn, for supporting me and providing me with the opportunity and the freedom to explore in this fascinating world of polynomial matrix computation. Also thanks to my other committee members, Mark Giesbrecht, Cameron Stewart, Arne Storjohann, and Gilles Villard for reading this thesis. Thanks especially to Arne, whose accomplishments and magical abilities to come up with new ideas have always amazed me and influenced my work. Thanks to Gilles for flying all the way to Canada from France to attend my defense and for all the encouraging words. Thanks to Mark for the helpful comments on this thesis.

Thanks to all my friends in Waterloo, especially my longtime roommate and good friend Zhirong Li, my resourceful academic brother Reinhold Burger, and my good friends Wei Li and Jun Chen, for all the fun conversations and memorable activities.

I am grateful for having a wonderful family that made everything in my life possible.

Dedication

To my parents and my sister, for always believing in me
To my daughter, for bringing new energy and joy
To my wife, for patience and support

Contents

List of Algorithms

Chapter 1

Introduction

In this thesis, we present efficient deterministic algorithms for a number of problems involving matrices of univariate power series or polynomials over a field. The first problem we consider is the computation of order bases, which can be viewed as the most fundamental among all the problems considered in this thesis, since order basis computation is used by the algorithms for all other problems. The second problem, minimal kernel basis computation, provides another essential tool used by the algorithms for the remaining problems, including the computation of matrix inverse, determinant, column basis, unimodular completion, Hermite normal form, rank and rank profile. The algorithm for kernel basis computation also immediately allows us to efficiently solve linear systems. The algorithm for column basis also allows us to efficiently compute matrix GCDs, column reduced forms and Popov normal forms for matrices of any dimension and any rank.

Let us first look at order bases and kernel bases in more detail.

Let $\mathbf{F} \in \mathbb{K}\left[[x]\right]^{m\times n}$ be a matrix of power series over a field $\mathbb{K}$. Given a non-negative integer σ, we say a vector $\mathbf{p} \in \mathbb{K}\left[x\right]^{n\times 1}$ of polynomials has order $(\mathbf{F}, \sigma)$, if

$$\mathbf{F} \cdot \mathbf{p} \equiv \mathbf{0} \mod x^{\sigma},$$

that is, the first σ terms of $\mathbf{F} \cdot \mathbf{p}$ are zero. Historically the problem of finding such vectors dates back to their use in Hermite's proof of the transcendence of e in 1873. In 1893 Padé, a student of Hermite, formalized the concepts introduced by Hermite and defined what is now known as Hermite-Padé approximants (where $m = 1$), Padé approximants (where $m = 1, n = 2$) and simultaneous Padé approximants (where $\mathbf{F}$ has a special structure). Such rational approximations also specified degree constraints on the polynomials $\mathbf{p}$ and had their order conditions related to these degree constraints. Additional cases of such order problems include vector and matrix versions of rational approximation, partial realizations of matrix sequences and vector rational reconstruction just to name a few (cf. the references in Beckermann and Labahn [1997]). As an example, the factorization of differential operators algorithm of Van Hoeij [1997] makes use of vector Hermite-Padé approximation to reconstruct differential factorizations over rational functions from factorizations of differential operators over power series domains.

The set of all such order $(\mathbf{F}, \sigma)$ approximations forms a module over $\mathbb{K}[x]$. An *order basis* – or minimal approximant basis or σ-basis – is a basis of this module having a type of minimal degree property (called a reduced order basis in [Beckermann and Labahn, 1997]). More details on this minimality property is given in the next chapter. In the case of rational approximation, order bases can be viewed as a natural generalization of the Padé table of a power series [Baker and Graves-Morris, 1996] since they are able to describe *all* solutions to such problems given particular degree bounds. They can even be used to show the well known block structure of the Padé and related Rational Interpolation tables [Beckermann and Labahn, 1997]. Order bases are used in such diverse applications as the inversion of structured matrices [Labahn, 1992], normal forms of matrix polynomials [Beckermann et al., 1999, 2006b], and other important problems in matrix polynomial arithmetic including matrix inversion, column reduction, determinant and nullspace basis computation

[Giorgi et al., 2003, Storjohann and Villard, 2005, Jeannerod and Villard, 2006, 2005].

Kernel bases are closely related to order bases.

For a matrix of polynomials $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$ with rank r. The set

$$\{\mathbf{p} \in \mathbb{K}[x]^{n} \mid \mathbf{F}\mathbf{p} = 0\},$$

is the (right) kernel of $\mathbf{F}$, which is also a $\mathbb{K}[x]$-module. It can be generated by a basis – a kernel basis of $\mathbf{F}$, that can be represented as a matrix in $\mathbb{K}[x]^{n \times (n-r)}$, with the columns being the basis elements.

Kernel bases of polynomial matrices appear in a large number of applications, being first used as an algebraic formalism in the area of control theory [Kucera, 1979]. For example, in linear system theory if a system is represented by a transfer function given in terms of a left coprime matrix fraction decomposition $\mathbf{T} = \mathbf{D}_\ell^{-1}\mathbf{N}_\ell$, with $\mathbf{D}_\ell$ and $\mathbf{N}_\ell$ polynomial matrices, then one often wants to find a right coprime matrix fraction representation $\mathbf{T} = \mathbf{N}_r\mathbf{D}_r^{-1}$ with $\mathbf{D}_r$ and $\mathbf{N}_r$ polynomial matrices of appropriate dimensions [Kailath, 1980]. This is equivalent to the kernel basis computation

$$[\mathbf{D}_\ell \quad -\mathbf{N}_\ell] \begin{bmatrix} \mathbf{N}_r \\ \mathbf{D}_r \end{bmatrix} = 0. \tag{1.1}$$

Solving and determining fundamental properties of the basic matrix equation $\mathbf{A}\mathbf{Z} = \mathbf{B}$ where $\mathbf{A}$ and $\mathbf{B}$ have polynomial elements can be determined by finding a complete description (that is, a basis) of the kernel of $[\mathbf{A}, -\mathbf{B}]$. Other examples of the use of kernels and their bases include fault diagnostics [Frisk, 2001] and column reduction of matrix polynomials [Beelen et al., 1988, Neven and Praagman, 1993].

In most applications one is interested in finding a *minimal kernel basis* of $\mathbf{F}$ in $\mathbb{K}[x]^n$ [Forney, 1975]. A kernel basis $\mathbf{N}$ of $\mathbf{F}$ is said to be minimal if it has the minimal possible column degrees among all right kernel bases. More details on this minimality is given in Section 2.7. A minimal kernel basis is also often referred to as a *minimal polynomial basis.* Examples where minimality are needed include the right coprime matrix factorization problem and the problem of column reducing a polynomial matrix. As an example, finding a basis for the kernel corresponding to the right matrix fraction problem (1.1) finds a matrix fraction while a minimal kernel basis finds such a fraction in reduced form having a minimal column degree denominator (needed for example in minimal partial realization problems).

1.1 Shifted Degrees

The standard way to measure the size of a matrix is to use its dimension and its degree. A major complication in many polynomial matrix computation problems is that the degrees of the intermediate results or the output can be much larger than the input. This seems to prevent these problems to be computed efficiently, since the size of the intermediate results and the size of the output provides lower bounds on the computational cost of any algorithm. But it is possible that the matrix degree just may not be the best choice to be used in these computations. In this thesis, instead of the standard matrix degrees, we use the more general shifted column degrees to guide the computations, and use the sum of the shifted column degrees to measure the size of polynomial matrices. We will see that the shifted column degree is in fact a more natural choice, as it guides the computation so that the sizes of the output and the intermediate results are indeed bounded by the size as the input for all these problems. Closely examination of the shifted degrees reveals new structures of the problems in this thesis, leading to better understanding of the problems, and allowing the development of simple and efficient algorithms.

For a column vector $\mathbf{p} = [p_1, \ldots, p_n]^T$ of univariate polynomials over a field $\mathbb{K}$, its column degree, denoted by $\operatorname{cdeg} \mathbf{p}$, is just the maximum of the degrees of the entries of $\mathbf{p}$, that is,

$$\operatorname{cdeg} \mathbf{p} = \max_{1 \le i \le n} \deg p_i.$$

The *shifted column degree* generalizes this standard column degree by taking the maximum after shifting the degrees by a given integer vector that is known as a *shift*. More specifically, the shifted column degree of $\mathbf{p}$ with respect to a shift $\vec{s} = [s_1, \ldots, s_n] \in \mathbb{Z}^n$, or the $\vec{s}$*-column degree* of $\mathbf{p}$ is

$$\operatorname*{cdeg}_{\vec{s}} \mathbf{p} = \max_{1 \le i \le n} [\deg p_i + s_i] = \deg(x^{\vec{s}} \cdot \mathbf{p}),$$

where

$$x^{\vec{s}} = \operatorname{diag}\left([x^{s_1}, x^{s_2}, \ldots, x^{s_n}]\right) = \begin{bmatrix} x^{s_1} & & & \\ & x^{s_2} & & \\ & & \ddots & \\ & & & x^{s_1} \end{bmatrix}.$$

For a matrix $\mathbf{P}$, we use $\operatorname{cdeg} \mathbf{P}$ and $\operatorname{cdeg}_{\vec{s}} \mathbf{P}$ to denote respectively the list of its column degrees and the list of its shifted $\vec{s}$-column degrees. When $\vec{s} = [0, \ldots, 0]$, the shifted column degree specializes to the standard column degree. The shifted row degree of a row vector $\mathbf{q}$ is defined in the same way.

$$\operatorname*{rdeg}_{\vec{s}} \mathbf{q} = \max_{1 \le i \le n} [\deg q_i + s_i] = \deg(\mathbf{q} \cdot x^{\vec{s}}).$$

The shifted degrees have been used previously in polynomial matrix computations and to generalize matrix normal forms [Beckermann et al., 2006b]. The shifted column degree is equivalent to the notion of *defect* commonly used in the literature. Our definition of $\vec{s}$-column degree is a special case of the $\mathbf{H}$-degree from [Beckermann and Labahn, 1997], where in this case $\mathbf{H} = x^{\vec{s}}$.

1.2 Order Basis Computation

The first problem considered in this thesis is the efficient computation of order basis. Algorithms for fast computation of order basis include that of Beckermann and Labahn [1994] which converts the matrix problem into a vector problem of higher order (which they called the Power Hermite-Padé problem). Their divide and conquer algorithm has complexity of $O^{\sim}(n^2m\sigma + nm^2\sigma)$ field operations. As usual, the soft-O notation $O^{\sim}$ is simply Big-O with polylogarithmic multiplicative factors $(\log(nm\sigma))^{O(1)}$ omitted. By working more directly on the input $m \times n$ input matrix, Giorgi et al. [2003] give a divide and conquer method with cost $O^{\sim}(n^{\omega}\sigma)$ arithmetic operations when $m \leq n$. Their method is very efficient if the row dimension m is close to the size of the column dimension n but can be improved if m is much smaller than n.

In a novel construction, Storjohann [2006] effectively reverses the approach of Beckermann and Labahn. Namely, rather than convert a high dimension matrix order problem into a lower dimension vector problem of higher order, Storjohann converts a low dimension problem to a high dimension problem with lower order. For example, computing an order basis for a $1 \times n$ vector input $\mathbf{f}$ and a large enough order σ can be converted to a problem of order basis computation with an $O(n) \times O(n)$ input matrix and an order about $2\sigma/n$. Using this conversion, the method of Giorgi et al. can then be used effectively for problems with small row dimensions to achieve a cost of $O^{\sim}(n^{\omega}a)$, where $a = m\sigma/n$.

However, while order bases of the original problem can have degree up to σ, the nature of Storjohann's conversion limits the degree of an order basis of the converted problem to $O(a)$ in order to be computationally efficient. In other words, this approach does not, in general, compute a complete order basis. Rather, in order to achieve efficiency, it only computes a partial order basis containing basis elements with degrees within $O(a)$, referred to by Storjohann as a *minbasis*. Fast methods for computing a minbasis are particularly useful for certain problems, for example, in the case of inversion of structured block matrices where one needs only precisely the minbasis [Labahn, 1992]. However, in other applications, such as those arising in polynomial matrix arithmetic, one needs a complete basis which specifies all solutions of a given order, not just those within a particular degree bound (cf. Beckermann and Labahn [1997]).

In Chapter 3 we present algorithms which compute an entire order basis with a cost of $O^{\sim}(n^{\omega}a)$ field operations. The algorithms differ depending on the nature of the degree shift required for the reduced order basis. In the first case we use a transformation that can be considered an extension of Storjohann's transformation. This new transformation provides a way to extend the results from one transformed problem to another transformed problem of a higher degree. This enables us to use an idea from the null space basis algorithm found in [Storjohann and Villard, 2005] in order to achieve efficient computation. At each iteration, basis elements within a specified degree bound are computed via a Storjohann transformed problem. Then the partial result is used to simplify the next Storjohann transformed problem of a higher degree, allowing basis elements within a higher degree bound to be computed efficiently. This is repeated until all basis elements are computed.

In order to compute an order basis efficiently, the first algorithm requires that the degree shifts are balanced. A balanced shift means $\max \vec{s} - \min \vec{s} \in O\left(m\sigma/n\right)$. In the case where the shift is not balanced, the row degrees of the basis can also become unbalanced in addition to the unbalanced column degrees. We give a second algorithm that balances the high degree rows and uses $O^{\sim}(n^{\omega}a)$ field operations when the shift $\vec{s}$ is unbalanced but satisfies the condition $\sum_{i=1}^{n}(\max(\vec{s}) - \vec{s}_i) \le m\sigma$. This condition essentially allows us to locate the high degree unbalanced rows that need to be balanced. The algorithm converts a problem of unbalanced shift to one with balanced shift, based on a second idea from [Storjohann, 2006]. Then the first algorithm is used to efficiently compute the elements of an order basis whose shifted degrees exceed a specified parameter. The problem is then reduced to one where we remove the computed elements. This results in a new problem with smaller dimension and higher degree. The same process is repeated again on this new problem in order to compute the elements with the next highest shifted degrees.

At the end of Chapter 3, we discuss how a more refined cost of $O^{\sim}(n^{\omega-1}m\sigma)$ instead of $O^{\sim}(n^{\omega}a)$ field operations can be achieved when the shifts are balanced. Note that the cost $O^{\sim}(n^{\omega}a)$ is less refined as it assumes that $a = m\sigma/n$ tends to infinity in the big O notation. However, $m\sigma/n$ can be arbitrary in the cost $O^{\sim}(n^{\omega-1}m\sigma)$.

Some results on order basis computation have appeared in [Zhou and Labahn, 2009, 2012].

1.3 Kernel Basis Computation

We are interested in fast computation of minimal kernel bases and shifted minimal kernel bases in exact environments. Historically computation of a minimal kernel basis has made use of either matrix pencil or resultant methods (often called a linearized approaches) or use of elimination methods for matrix polynomials. Matrix pencil methods convert a kernel basis computation problem to one of larger matrix size but having polynomial degree one. In this case a minimal kernel basis is determined from the computation of the Kronecker canonical form, with efficient algorithms given by Beelen and Dooren [1988], Misra et al. [1994], Oara and Dooren [1997]. The cost of these algorithms is $O(m^2nd^3)$, where d is the degree of the input matrix. Resultant methods convert the kernel basis computation of the matrix polynomial $\mathbf{F}$ into a block Toeplitz kernel problem with much higher dimension with the resulting complexity again being high. In [Storjohann and Villard, 2005] the authors give a randomized Las Vegas algorithm for computing a set of $n-r$ linearly independent elements in the kernel of $\mathbf{F}$ with a cost of $O^{\sim}\left(nmr^{\omega-2}d\right)$ where $O^{\sim}$ is the same as Big-O but without log factors and where ω is the power of fast matrix multiplication. These linearly independent elements do not in general form a basis for the kernel, as they may not generate all the elements in the kernel. A set of any such $n-r$ linearly independent elements only form a basis for the $\mathbb{F}\left(x\right)$-vector space $\{\mathbf{p}\in\mathbb{K}\left(x\right)^{n} \mid \mathbf{F}\mathbf{p}=0\}$ when the ring $\mathbb{K}\left[x\right]$ is extended to the field $\mathbb{K}\left(x\right)$.

In Chapter 5 we present a deterministic algorithm for computing a minimal kernel basis with a cost of $O^{\sim}\left(n^{\omega-1}md\right)$ field operations in $\mathbb{K}$. The same algorithm can also compute a $\vec{s}$-minimal kernel basis of $\mathbf{F}$ with a cost of $O^{\sim}(n^{\omega}s)$ if the entries of $\vec{s}$ bound the corresponding column degrees of $\mathbf{F}$, where s is the average of the m largest entries of $\vec{s}$.

A key component of the algorithm is the computation of order basis. We use order basis computation to compute a partial kernel basis, which also reduces the column dimension of the problem. The problem can then be separated to two subproblems of smaller row dimensions, which can then be handled in the same way as the original problem.

Some results on kernel basis computation have appeared in [Zhou et al., 2012].

1.4 Overview

The remainder of this thesis is structured as follows. Basic definitions and properties are given in the next chapter. The details of our order basis computation can be found in Chapter 3. Kernel basis computation is described in Chapter 5. Chapter 6 discusses the algorithm for computing matrix inverse. Chapter 7 discusses the computation of column bases. Unimodular completion is then discussed in Chapter 8. Then we look at determinant computation in Chapter 9, Hermite normal form computation in Chapter 10, and rank profile and rank-sensitive computations in Chapter 11.

Chapter 2

Preliminaries

In this chapter, we provide some of the background needed in order to understand the basic concepts needed for order basis computation and nullspace basis computation.

2.1 Notation

Since we are interested in computing bases with minimal degrees, it is useful to have convenient notations for comparing two lists of degrees. In addition, our matrices often represent sets of column vectors, so the arrangement of these columns are not important. To compare two lists of column degrees from two matrices, we first sort each list in increasing order, and then do the comparison.

Comparing Unordered Lists For two lists $\vec{a} \in \mathbb{Z}^n$ and $\vec{b} \in \mathbb{Z}^n$, let $\bar{a} = [\bar{a}_1, \ldots, \bar{a}_n]$ and $\bar{b} = \left[\bar{b}_1, \ldots, \bar{b}_n\right]$ be the lists consists of the entries of $\vec{a}$ and $\vec{b}$ but sorted in increasing order.

$$\begin{cases} \vec{a} \geq \vec{b} & \text{if } \bar{a}_i \geq \bar{b}_i \text{ for all } i \in [1, \dots n] \\ \vec{a} \leq \vec{b} & \text{if } \bar{a}_i \leq \bar{b}_i \text{ for all } i \in [1, \dots n] \\ \vec{a} > \vec{b} & \text{if } \vec{a} \geq \vec{b} \text{ and } \bar{a}_j > \bar{b}_j \text{ for at least one } j \in [1, \dots n] \\ \vec{a} < \vec{b} & \text{if } \vec{a} \leq \vec{b} \text{ and } \bar{a}_j < \bar{b}_j \text{ for at least one } j \in [1, \dots n]\,. \end{cases}$$

Summation Notation For a list $\vec{a} = [a_1, \dots, a_n] \in \mathbb{Z}^n$, we write $\sum \vec{a}$ without index to denote the summation of all entries in $\vec{a}$.

Uniformly Shift a List For a list $\vec{a} = [a_1, \dots, a_n] \in \mathbb{Z}^n$ and $c \in \mathbb{Z}$, we write $\vec{a} + c$ to denote $\vec{a} + [c, \dots, c] = [a_1 + c, \dots, a_n + c]$, and similarly for $-$.

Compare a List with a Integer For a list $\vec{a} = [a_1, \dots, a_n] \in \mathbb{Z}^n$ and $c \in \mathbb{Z}$, we write $\vec{a} < c$ to denote $\vec{a} < [c, \dots, c]$, and similarly for $>, \leq, \geq, =$.

Example 2.1. Let $\mathbf{A} = [1, x^2]$, and $\mathbf{B} = [x, 1]$, we can write $\operatorname{cdeg} \mathbf{A} > \operatorname{cdeg} \mathbf{B}$, since the sorted lists of column degrees of $\mathbf{A}$ and $\mathbf{B}$ are $[0, 2]$ and $[0, 1]$ respectively.

2.2 Model of Computation

The computational cost in this thesis is analyzed by bounding the number of arithmetic operations (additions, subtractions, multiplications, and divisions) in the coefficient field $\mathbb{K}$ on an algebraic random access machine. We reduce all the problems to polynomial matrix multiplications. We use $\mathrm{MM}(n, d)$ to denote the cost of multiplying two polynomial matrices with dimension n and degree d, and $\mathrm{M}(n)$ to denote the cost of multiplying two polynomials with degree d. We assume that $\mathrm{M}(st) \in O\left(\mathrm{M}(s)\,\mathrm{M}(t)\right)$ and $\mathrm{M}(n) \in O(n^{\omega - 1})$, where the multiplication exponent ω

is assumed to satisfy $2 < \omega \leq 3$. We take $\mathrm{MM}(n, d) \in O\left(n^{\omega}\, \mathrm{M}(d)\right) \subset O^{\sim}\left(n^{\omega} d\right)$. We refer to the book by Gathen and Gerhard [2003] for more details and reference about the cost of polynomial multiplication and matrix multiplication.

2.3 Computational Cost in Terms of Average Degrees

In this thesis, we often state the computational costs in terms of the average column degrees, in contrast to the matrix degrees that are also the maximum column degrees typically used in the literature. For example, the cost of a problem involving an input matrix with dimension $n \times n$ and column degrees $\vec{s} = [s_1, \ldots, s_n]$ is usually $O^{\sim}\left(n^{\omega} s\right)$, where $s = \sum \vec{s}/n$ is the average column degree of the input matrix. This is similar to the cost of multiplying two matrices with dimension n and degree s.

It is tempting to also write the cost as $O^{\sim}\left(n^{\omega-1}\xi\right)$ with $\xi = \sum \vec{s}$ being the sum of the column degrees. However, note that if ξ is asymptotically much smaller than n, such as $\xi \in O\left(\log\left(n\right)\right)$, this cost becomes $O^{\sim}\left(n^{\omega-1}\right)$, which is incorrect since this is smaller than the size of the input matrix. This issue is caused by the subtleties involving the use of degree as the size while the degree of a polynomial is not quite the same as its size. There are two differences to be noted. First, a degree 0 polynomial does not have size 0. It still has one coefficient, which is one field element to store. Another exception is the zero polynomial, whose degree is sometimes defined as $-\infty$ or -1 depending on the context. But a negative size does not make much sense.

There are two ways to address this issue. First, instead of using the degree in the cost, we can use the actual number of coefficients, which is

$$\begin{cases} 0 & \text{if } p = 0 \\ 1 + \deg p & \text{if } \deg p \geq 0, \end{cases}$$

where p is a polynomial, a polynomial vector, or a polynomial matrix. For example, a column vector with column degree 2 then has 3 column coefficients. Then ξ in the cost represents the sum of the number of column coefficients for all the columns.

Perhaps the easiest way is to just use the average column degrees $s = \sum \vec{s}/n$ in the cost, and state the cost as $O^{\sim}(n^{\omega}s)$. Then the above issue no longer exists because the cost $O^{\sim}(n^{\omega}s)$ is similar to the cost of multiplying two matrices of dimension n and degree s, with s assumed to tend to infinity in the asymptotic Big-O notation, hence degree zero or zero polynomials are no longer an issue. For simplicity, this is the approach we take in this thesis. We describe the cost in terms of the average column degrees instead of the sum of the column degrees. But keep in mind that the first approach can be used to give a more refined cost.

For order basis computation problems considered in this thesis, a problem with a dimension $m \times n$ input matrix and order σ is computed with a cost of $O^{\sim}(n^{\omega}a)$, where $a = m\sigma/n$ can be viewed as the average size of the rows if we treat the original input matrix as a square matrix by appending $n - m$ zero rows. If the cost is stated as $O^{\sim}(n^{\omega-1}m\sigma)$ in the case of $m\sigma \in o(n)$, we have a similar issue as before, as we may obtain a cost of $O^{\sim}(n^{\omega-1})$. This cost may seem strange, as it is less than the cost of multiplying two matrices of dimension n and degree 0. However, it does not contradict with the input size, which is $O(nm\sigma) = o(n^2)$. In fact, when we consider the case of order basis computation with balanced shift, we indeed provide a way to handle the case of $m\sigma \in o(n)$ with a cost of $O^{\sim}(n^{\omega-1}m\sigma)$ in Section 3.6 of Chapter 3.

2.4 Shifted Degrees

In this section, we look at some properties of shifted degrees, which may help in understanding their usefulness in efficient computations of polynomial matrix problems.

Lemma 2.2. *A matrix $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ has $\vec{u}$-column degrees bounded by $\vec{v}$ if and only if its $-\vec{v}$-row degrees are bounded by $-\vec{u}$.*

Proof. The lemma follows from the fact $x^{\vec{u}}\mathbf{A}x^{-\vec{v}}$ has degree no more than 0 (Note that the negative degrees are defined here by settingdeg $x^{-d} = -d$ for $d \in \mathbb{Z}_{>0}$. If one wishes to avoid negative degrees, one can simply shift the degrees up by multiplying the matrix by x^a for some large a). Note the symmetry between the shifted row degrees and the shifted column degrees. □

Example 2.3. Let $\mathbf{A} = \left[1, x^2\right]$, $\vec{u} = [1]$ and $\vec{v} = [2,4]$. Then $\operatorname{cdeg}_{\vec{u}}\mathbf{A} = [1,3] \le \vec{v}$, and $\operatorname{rdeg}_{-\vec{v}}\mathbf{A} = [-2] \le -\vec{u}$. If $\vec{u} = [1]$ and $\vec{v} = [1,1]$, then $\operatorname{cdeg}_{\vec{u}}\mathbf{A} = [1,3] \not\le \vec{v}$, and $\operatorname{rdeg}_{-\vec{v}}\mathbf{A} = [1] \not\le -\vec{u}$.

Lemma 2.4. *If the $\vec{u}$-column degrees of $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ are bounded by the corresponding entries of an integer list $\vec{v} \in \mathbb{Z}^n$, (or equivalently, the $-\vec{v}$-row degrees of $\mathbf{A}$ are bounded by $-\vec{u}$) and the $\vec{v}$-column degrees of $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$ are bounded by $\vec{w} \in \mathbb{Z}^k$, then the $\vec{u}$-column degrees of $\mathbf{AB}$ are bounded by $\vec{w}$.*

Proof. Note that $x^{\vec{u}}\mathbf{A}x^{-\vec{v}}$ and $x^{\vec{v}}\mathbf{B}^{-\vec{w}}$ have degrees bounded by 0. Therefore

$$x^{\vec{u}}\mathbf{A}x^{-\vec{v}}x^{\vec{v}}\mathbf{B}^{-\vec{w}} = x^{\vec{u}}\mathbf{AB}^{-\vec{w}}$$

also has degree bounded by 0, or equivalently, $\operatorname{cdeg}_{\vec{u}}\mathbf{AB} \le \vec{w}$. □

Corollary 2.5. *Let $\vec{v}$ be a shift whose entries bound the corresponding column degrees of $\mathbf{A}$. Then for any polynomial matrix $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$, the column degrees of $\mathbf{AB}$ are bounded by the corresponding $\vec{v}$-column degrees of $\mathbf{B}$.*

Proof. Just set the shift $\vec{u}$ to 0 in Lemma 2.4 $\square$

2.5 Unimodular Matrices and Unimodular Transformations

A square matrix in $\mathbb{K}[x]^{n\times n}$ is said to be *unimodular* if its determinant is in $\mathbb{K}\setminus\{0\}$. Equivalently, a matrix in $\mathbb{K}[x]^{n\times n}$ is unimodular if and only if it has an inverse in $\mathbb{K}[x]^{n\times n}$. Therefore, a unimodular transformation can be undone by the multiplication with the inverse transformation matrix in $\mathbb{K}[x]^{n\times n}$. This allows us to talk about unimodular equivalence of the polynomial matrices. Two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{K}[x]^{m\times n}$ are said to be right unimodularly equivalent if $\mathbf{A} = \mathbf{BU}$ for some unimodular matrix $\mathbf{U}$. This also means that the columns of $\mathbf{A}$ and $\mathbf{B}$ generates the same set of vectors. That is, if a vector $\mathbf{q} = \mathbf{Ap}$ for some $\mathbf{p} \in \mathbb{K}[x]^{n\times 1}$, then $\mathbf{q} = \mathbf{BUp}$.

2.6 Column Basis

The column module of a nonzero matrix $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ is the $\mathbb{K}[x]$-module generated by the columns of $\mathbf{A}$, that is, this module contains all the column vectors that are $\mathbb{K}[x]$-linear combinations of the columns of $\mathbf{A}$. A column basis of $\mathbf{A}$ is just a basis for this module. Any column basis of $\mathbf{A}$ can be represented as a matrix $\mathbf{T}$, whose columns are the basis elements. The matrix $\mathbf{T}$ has full-rank since basis elements must be linearly independent. In addition, any two bases for the same module are unimodularly equivalent.

Example 2.6. If $\mathbf{A} = \begin{bmatrix} 1 & 1+x \\ x & x+x^2 \end{bmatrix}$, then $\mathbf{T} = \begin{bmatrix} 1 \\ x \end{bmatrix}$ is a column basis of $\mathbf{A}$, as $\mathbf{A} = \mathbf{T}[1, 1+x]$.

Lemma 2.7. *If the matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ are both column bases of $\mathbf{A}$, then $\mathbf{T}_1$ and $\mathbf{T}_2$ are right unimodularly equivalent.*

Proof. Any column of $\mathbf{T}_1$ or $\mathbf{T}_2$ is generated by $\mathbf{T}_1$ and also by $\mathbf{T}_2$. In other words, $\mathbf{T}_1 = \mathbf{T}_2\mathbf{U}$ and $\mathbf{T}_2 = \mathbf{T}_1\mathbf{V}$ for polynomial matrices $\mathbf{U}$ and $\mathbf{V}$. Hence $\mathbf{T}_1 = \mathbf{T}_1\mathbf{V}\mathbf{U}$ and $\mathbf{T}_2 = \mathbf{T}_2\mathbf{U}\mathbf{V}$, implying $\mathbf{U}\mathbf{V} = \mathbf{V}\mathbf{U} = I$, which requires both $\mathbf{U}$ and $\mathbf{V}$ to be unimodular. □

A column basis is not unique and indeed any column basis right multiplied by a unimodular polynomial matrix gives another column basis. As a result, a column basis can have arbitrarily high degree

2.7 Minimality and Column Reducedness

For many polynomial matrix computation problems, we would like the output matrix to be not only easy to describe, but also convenient to use. This usually means the column degrees or the more general shifted column degrees are small comparing to other matrices that are right unimodularly equivalent. For example a matrix $\mathbf{A} = [x, x^2, x^2]$ with column degrees $[1, 2, 2]$ can be unimodularly transformed to a matrix $\mathbf{B} = [x, x, x^2]$ with column degrees $[1, 1, 2]$, which is more desirable with lower degrees.

To unimodularly transform a matrix to one with lower column degrees, we can look at its *leading column coefficient matrix*, which is defined as follows.

Definition 2.8. Given a matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \mathbb{K}[x]^{m \times n}$, the leading column coefficient matrix A of $\mathbf{A}$ is

$$\begin{aligned} A &= \operatorname{lcoeff}(\mathbf{A}) \\ &= [\operatorname{lcoeff}(\mathbf{a}_1), \ldots, \operatorname{lcoeff}(\mathbf{a}_k)] \\ &= [\operatorname{coeff}(\mathbf{a}_1, \operatorname{cdeg}(\mathbf{a}_1)), \ldots, \operatorname{coeff}(\mathbf{a}_k, \operatorname{cdeg}(\mathbf{a}_k))]. \end{aligned}$$

Then, the matrix $\mathbf{A}$ can be unimodularly reduced to another matrix with lower column degrees if $\operatorname{lcoeff}(\mathbf{A})$ is not full-rank.

Lemma 2.9. *Given a matrix* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *with no zero columns. If* $\operatorname{lcoeff}(\mathbf{A})$ *is not full-rank, then there is a unimodular matrix* $\mathbf{U}$ *such that* $\operatorname{cdeg}(\mathbf{AU}) < \operatorname{cdeg}\mathbf{A}$.

Proof. We may assume the columns of $\mathbf{A}$ are arranged in increasing column degrees. Let the column degrees of $\mathbf{A}$ be $[d_1, \ldots, d_n]$. Let $A = \operatorname{lcoeff}(\mathbf{A})$. Suppose the ith column A_i of A is a linear combination of the first $i-1$ columns. That is, $A_i = A'a$, where A' is the submatrix of A consists of the first $i-1$ columns of A, and $a = [a_1, \ldots, a_{i-1}]^T \in \mathbb{K}^{(i-1)\times 1}$. Let

$$\mathbf{U} = \begin{bmatrix} 1 & & & & -a_1 x^{d_i - d_1} & & \\ & 1 & & & -a_2 x^{d_i - d_2} & & \\ & & \ddots & & \vdots & & \\ & & & 1 & -a_{i-1} x^{d_i - d_{i-1}} & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}.$$

Then $\mathbf{B} = \mathbf{AU}$ has column degrees $\left[d_1, \ldots, d_{i-1}, \bar{d}_i, d_{i+1}, \ldots, d_n\right]$, with $\bar{d}_i < d_i$. □

The leading column coefficient matrix can also help to determine the degree of the matrix.

Lemma 2.10. *The degree of the determinant of a matrix* $\mathbf{A} \in \mathbb{K}[x]^{n\times n}$ *is bounded by* $\sum \operatorname{cdeg} \mathbf{A}$. *If* $\operatorname{lcoeff}(\mathbf{A})$ *is nonsingular, then* $\deg\det\mathbf{A} = \sum \operatorname{cdeg}\mathbf{A}$. *More generally, for any shift* $\vec{s} \in \mathbb{Z}^n$, $\deg\det\mathbf{A} \le \sum \operatorname{cdeg}_{\vec{s}}\mathbf{A} - \sum \vec{s}$. *If* $\operatorname{lcoeff}\left(x^{\vec{s}}\mathbf{A}\right)$ *is nonsingular, then* $\deg\det\mathbf{A} = \sum \operatorname{cdeg}_{\vec{s}}\mathbf{A} - \sum \vec{s}$.

Proof. The determinant is the sum of products, with each product involving exactly one entry from each column. So the largest possible degree of each product is $\sum \operatorname{cdeg}\mathbf{A}$. For the second statement, note that the coefficient of $\det\mathbf{A}$ corresponds to the largest possible degree $\sum \operatorname{cdeg}\mathbf{A}$ is $\det\operatorname{lcoeff}(\mathbf{A})$. The more general results with shift can be shown by considering the the determinant of $x^{\vec{s}}\mathbf{A}$. $\square$

It is still not always possible to order two lists of integer degrees, as in the case of matrices $[x, x^2, x^2]$ and $[x, x, x^3]$ with column degrees $[1, 2, 2]$ and $[1, 1, 3]$. Although the lists of column degrees of the set of unimodular equivalent matrices are not well-ordered, there always exists some matrices with the *minimal* column degrees, as shown below below by Lemma 2.11 and Corollary 2.12.

Lemma 2.11. *Given any two right unimodularly equivalent matrices* $\mathbf{A}$ *and* $\mathbf{B}$, *the matrix* $[\mathbf{A}, \mathbf{B}]$ *can be unimodularly reduced to* $[0, \mathbf{C}]$ *with a matrix* $\mathbf{C}$ *that is unimodularly equivalent to both* $\mathbf{A}$ *and* $\mathbf{B}$, *and satisfies* $\operatorname{cdeg}\mathbf{C} \le \operatorname{cdeg}\mathbf{A}$ *and* $\operatorname{cdeg}\mathbf{C} \le \operatorname{cdeg}\mathbf{B}$. *More generally, with a shift* $\vec{s}$, *the matrix* $[\mathbf{A}, \mathbf{B}]$ *can be unimodularly reduced to* $[0, \mathbf{D}]$ *with a matrix* $\mathbf{D}$ *that is unimodularly equivalent to both* $\mathbf{A}$ *and* $\mathbf{B}$, *and satisfies* $\operatorname{cdeg}_{\vec{s}}\mathbf{D} \le \operatorname{cdeg}_{\vec{s}}\mathbf{A}$ *and* $\operatorname{cdeg}_{\vec{s}}\mathbf{D} \le \operatorname{cdeg}_{\vec{s}}\mathbf{B}$.

Proof. If r is the rank of $\mathbf{A}$ and $\mathbf{B}$, we can compute a matrix $\mathbf{C}$ the column degrees of $\mathbf{C}$ are bounded by the column degrees of the r linear independent columns of $[\mathbf{A}, \mathbf{B}]$ with the smallest column degrees, as higher degree columns can be reduced using Lemma 2.9. For the more general result with shift, we can again multiply $x^{\vec{s}}$ to the matrices. $\square$

Corollary 2.12. *Given a matrix* $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ *and a shift* $\vec{s} \in \mathbb{Z}^n$*. There exists a matrix* $\mathbf{G}$ *that is right unimodularly equivalent to* $\mathbf{F}$ *and* $\operatorname{cdeg}\mathbf{G} \le \operatorname{cdeg}(\mathbf{FU})$ *for any unimodular* $\mathbf{U}$*. More generally, with a shift* $\vec{s}$*, there exists a matrix* $\mathbf{H}$ *that is right unimodularly equivalent to* $\mathbf{F}$ *and* $\operatorname{cdeg}_{\vec{s}}\mathbf{G} \le \operatorname{cdeg}_{\vec{s}}(\mathbf{FU})$ *for any unimodular* $\mathbf{U}$

Proof. Just repeatedly apply Lemma 2.11 to matrices that are right unimodularly equivalent to $\mathbf{F}$. □

The existence of matrices with minimal column degrees allows us to define *column reduced.*

Definition 2.13. A matrix $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ is said to be *column reduced* if $\operatorname{cdeg}\mathbf{A} \le \operatorname{cdeg}\mathbf{AU}$ for any unimodular matrix $\mathbf{U}$. More generally, for a shift $\vec{s} \in \mathbb{Z}^n$, a matrix $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ is said to be $\vec{s}$*-column reduced* if $\operatorname{cdeg}_{\vec{s}}\mathbf{A} \le \operatorname{cdeg}_{\vec{s}}\mathbf{AU}$ for any unimodular matrix $\mathbf{U}$.

The leading column coefficient matrix provides a useful test for being column reduced.

Lemma 2.14. *A matrix* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *with no zero columns is column reduced if and only if* $\operatorname{lcoeff}(\mathbf{A})$ has full column rank.

Proof. We just need to show that the full-rank $\operatorname{lcoeff}(\mathbf{A})$ implies a column reduced $\mathbf{A}$, since the other direction is covered in Lemma 2.9. Suppose $\operatorname{lcoeff}(\mathbf{A})$ is full-rank but not column reduced. Let $\mathbf{B}$ be a unimodularly equivalent column reduced matrix, which exists from Corollary 2.12, then $\operatorname{cdeg}\mathbf{A} > \operatorname{cdeg}\mathbf{B}$. Let $\mathbf{U}$ be the unimodular matrix satisfying $\mathbf{AU} = \mathbf{B}$. Let $\bar{\mathbf{A}}$ be a square matrix with n rows chosen from $\mathbf{A}$ such that $\operatorname{cdeg}\bar{\mathbf{A}} = \operatorname{cdeg}\mathbf{A}$. Then $\bar{\mathbf{B}} = \bar{\mathbf{A}}\mathbf{U}$ is a matrix consisting of rows from $\mathbf{B}$ with the same indices. It follows that $\deg\det\bar{\mathbf{A}} = \deg\det\bar{\mathbf{B}}$. But $\operatorname{cdeg}\bar{\mathbf{B}} \le \operatorname{cdeg}\mathbf{B} < \operatorname{cdeg}\mathbf{A} = \operatorname{cdeg}\bar{\mathbf{A}}$, while from Lemma 2.10 $\deg\det\bar{\mathbf{B}} \le \sum\operatorname{cdeg}\bar{\mathbf{B}}$ and $\deg\det\bar{\mathbf{A}} = \sum\operatorname{cdeg}\bar{\mathbf{A}}$, which gives $\deg\det\bar{\mathbf{B}} < \deg\det\bar{\mathbf{A}}$, a contradiction. □

We can show that the nonzero columns of a column reduced matrix are linearly independent, which means they form a column basis.

Lemma 2.15. *The nonzero columns of a column reduced matrix are linearly independent.*

Proof. If the nonzero columns of a matrix $\mathbf{A}$ are not linearly independent, then the matrix $\mathbf{A}'$ consists of the nonzero columns of $\mathbf{A}$ satisfies $\mathbf{A}'\mathbf{p} = 0$ for some polynomial vector $\mathbf{p}$. Let $\vec{a} = \operatorname{cdeg}\mathbf{A}'$. Then $\mathbf{A}'\mathbf{p} = \left(\mathbf{A}'x^{-\vec{a}}\right)\cdot\left(x^{\vec{a}}\mathbf{p}\right) = 0$, which requires $\operatorname{lcoeff}\left(\mathbf{A}'x^{-\vec{a}}\right)\operatorname{lcoeff}\left(x^{\vec{a}}\mathbf{p}\right) = 0$, implying $\operatorname{lcoeff}\left(\mathbf{A}'x^{-\vec{a}}\right) = \operatorname{lcoeff}\left(\mathbf{A}'\right)$ is not full rank, hence $\mathbf{A}$ is not column reduced by Lemma 2.14. $\square$

Corollary 2.16. *Any matrix $\mathbf{F} \in \mathbb{K}\left[x\right]^{m\times n}$ can be unimodularly transformed to $[0,\mathbf{T}] \in \mathbb{K}\left[x\right]^{m\times n}$ with a full rank matrix $\mathbf{T} \in \mathbb{K}\left[x\right]^{m\times r}$, that is, $\mathbf{FU} = [0,\mathbf{T}]$ for some unimodular matrix $\mathbf{U}$. Any such matrix $\mathbf{T}$ is a column basis of $\mathbf{F}$.*

Proof. We can just repeatedly apply the unimodular transformation from Lemma 2.9. By Lemma 2.14, this eventually gives a column reduced form $[0,\mathbf{T}']$, which has linearly independent nonzero columns $\mathbf{T}'$ by Lemma 2.15. Then $\mathbf{T}'$ is a column basis of $\mathbf{F}$ since its columns are linearly independent and the $m\times n$ matrix $[0,\mathbf{T}']$ is unimodularly equivalent with $\mathbf{F}$, implying $\mathbf{T}'$ and $\mathbf{F}$ each has columns that generate the same $\mathbb{F}\left[x\right]$-module. $\square$

In Lemma 2.4, when the matrix $\mathbf{A}$ is $\vec{u}$-column reduced, the bound becomes an equality, which then gives the following lemma. This can be viewed as a stronger version of the predictable-degree property [Kailath, 1980].

Lemma 2.17. *Let $\mathbf{A} \in \mathbb{K}\left[x\right]^{m\times n}$ be a $\vec{u}$-column reduced matrix with no zero columns and with $\operatorname{cdeg}_{\vec{u}}\mathbf{A} = \vec{v}$. Then a matrix $\mathbf{B} \in \mathbb{K}\left[x\right]^{n\times k}$ has $\vec{v}$-column degrees $\operatorname{cdeg}_{\vec{v}}\mathbf{B} = \vec{w}$ if and only if $\operatorname{cdeg}_{\vec{u}}\left(\mathbf{AB}\right) = \vec{w}$.*

Proof. $\vec{u}$-column reduced means the leading column coefficient matrix A of $x^{\vec{u}}\mathbf{A}x^{-\vec{v}}$ has linearly independent columns. Now the leading coefficient matrix B of $x^{\vec{v}}\mathbf{B}^{-\vec{w}}$ has no zero column if and only if the leading column coefficient matrix AB of $x^{\vec{u}}\mathbf{A}x^{-\vec{v}}x^{\vec{v}}\mathbf{B}^{-\vec{w}} = x^{\vec{u}}\mathbf{A}\mathbf{B}^{-\vec{w}}$ has no zero column, in other words, $x^{\vec{v}}\mathbf{B}^{-\vec{w}}$ has column degrees $[0,\ldots,0]$ if and only if $x^{\vec{u}}\mathbf{A}\mathbf{B}^{-\vec{w}}$ has column degrees $[0,\ldots,0]$. □

We also have the following similar result on column reducedness.

Lemma 2.18. *If* $\mathbf{A}$ *is a full-rank* $\vec{u}$*-column reduced matrix with* $\operatorname{cdeg}_{\vec{u}}\mathbf{A} = \vec{v}$*, then* $\mathbf{B}$ *is* $\vec{v}$*-column reduced if and only if* $\mathbf{AB}$ *is column reduced.*

Proof. This again follows by looking at the full rank leading column coefficient matrices. □

2.8 Order Basis

We now look at order basis in more detail.

Let $\mathbb{K}$ be a field, $\mathbf{F} \in \mathbb{K}[[x]]^{m\times n}$ a matrix of power series and $\vec{\sigma} = [\sigma_1,\ldots,\sigma_m]$ a vector of non-negative integers.

Definition 2.19. We say a column vector of polynomials $\mathbf{p} \in \mathbb{K}[x]^{n\times 1}$ has *order* $(\mathbf{F},\vec{\sigma})$ (or *order* $\vec{\sigma}$ with respect to $\mathbf{F}$) if $\mathbf{F}\cdot\mathbf{p} \equiv \mathbf{0} \mod x^{\vec{\sigma}}$, that is,

$$\mathbf{F}\cdot\mathbf{p} = x^{\vec{\sigma}}\mathbf{r} = \begin{bmatrix} x^{\sigma_1} & & \\ & \ddots & \\ & & x^{\sigma_m} \end{bmatrix}\mathbf{r}$$

for some $\mathbf{r} \in \mathbb{K}[[x]]^{m\times 1}$. If $\vec{\sigma} = [\sigma,\ldots,\sigma]$ has entries uniformly equal to σ, then we say that $\mathbf{p}$ has order $(\mathbf{F},\sigma)$. The set of all order $(\mathbf{F},\vec{\sigma})$ vectors is a free $\mathbb{K}[x]$-module denoted by $\langle(\mathbf{F},\vec{\sigma})\rangle$.

An order basis for $\mathbf{F}$ and $\vec{\sigma}$ is simply a basis for the $\mathbb{K}[x]$-module $\langle(\mathbf{F},\vec{\sigma})\rangle$. In this thesis we compute those order bases having minimal or shifted minimal degrees (also referred to as a reduced order basis in [Beckermann and Labahn, 1997]).

An order basis [Beckermann and Labahn, 1994, 1997] $\mathbf{P}$ of $\mathbf{F}$ with order $\vec{\sigma}$ and shift $\vec{s}$, or simply an $(\mathbf{F},\vec{\sigma},\vec{s})$-basis, is a basis for the module $\langle(\mathbf{F},\vec{\sigma})\rangle$ having minimal $\vec{s}$-column degrees. If $\vec{\sigma}=[\sigma,\ldots,\sigma]$ is uniform then we simply write $(\mathbf{F},\sigma,\vec{s})$-basis. The precise definition of an $(\mathbf{F},\vec{\sigma},\vec{s})$-basis is as follows.

Definition 2.20. A polynomial matrix $\mathbf{P}$ is an order basis of $\mathbf{F}$ of order $\vec{\sigma}$ and shift $\vec{s}$, denoted by $(\mathbf{F},\vec{\sigma},\vec{s})$-basis, if the following properties hold:

1. $\mathbf{P}$ is a nonsingular matrix of dimension n and is $\vec{s}$-column reduced.
2. $\mathbf{P}$ has order $(\mathbf{F},\vec{\sigma})$ (or equivalently, each column of $\mathbf{P}$ is in $\langle(\mathbf{F},\vec{\sigma})\rangle$).
3. Any $\mathbf{q}\in\langle(\mathbf{F},\vec{\sigma})\rangle$ can be expressed as a linear combination of the columns of $\mathbf{P}$, given by $\mathbf{P}^{-1}\mathbf{q}$.

It follows from Definition 2.20 and Lemma 2.7 that any pair of $(\mathbf{F},\vec{\sigma},\vec{s})$-bases $\mathbf{P}$ and $\mathbf{Q}$ are column bases of each other and are unimodularly equivalent.

From [Beckermann and Labahn, 1997] we have the following lemma.

Lemma 2.21. *The following are equivalent for a polynomial matrix* $\mathbf{P}$*:*

1. $\mathbf{P}$ *is a* $(\mathbf{F},\vec{\sigma},\vec{s})$*-basis.*
2. $\mathbf{P}$ *is comprised of a set of* n *minimal* $\vec{s}$*-column degree polynomial vectors that are linearly independent and each having order* $(\mathbf{F},\vec{\sigma})$*.*
3. $\mathbf{P}$ *does not contain a zero column, has order* $(\mathbf{F},\vec{\sigma})$*, is* $\vec{s}$*-column reduced, and any* $\mathbf{q}\in\langle(\mathbf{F},\vec{\sigma})\rangle$ *can be expressed as a linear combination of the columns of* $\mathbf{P}$*.*

In some cases an entire order basis is unnecessary and instead one looks for a minimal basis that generates only the elements of $\langle(\mathbf{F},\vec{\sigma})\rangle$ with $\vec{s}$-column degrees bounded by a given δ. Such a minimal basis is a partial $(\mathbf{F},\vec{\sigma},\vec{s})$-basis comprised of elements of a $(\mathbf{F},\vec{\sigma},\vec{s})$-basis with $\vec{s}$-column degrees bounded by δ. This is called a *minbasis* in Storjohann [2006].

Definition 2.22. Let $\langle(\mathbf{F},\vec{\sigma},\vec{s})\rangle_{\delta}\subset\langle(\mathbf{F},\vec{\sigma})\rangle$ denote the set of order $(\mathbf{F},\vec{\sigma})$ polynomial vectors with $\vec{s}$-column degree bounded by δ. A $(\mathbf{F},\vec{\sigma},\vec{s})_{\delta}$-basis is a polynomial matrix $\mathbf{P}$ not containing a zero column and satisfying:

1. $\mathbf{P}$ has order $(\mathbf{F},\vec{\sigma})$.

   (a) Any element of $\langle(\mathbf{F},\vec{\sigma},\vec{s})\rangle_{\delta}$ can be expressed as a linear combination of the columns of $\mathbf{P}$.

   (b) $\mathbf{P}$ is $\vec{s}$-column reduced.

A $(\mathbf{F},\vec{\sigma},\vec{s})_{\delta}$-basis is, in general, not square unless δ is large enough to contain all n basis elements in which case it is a complete $(\mathbf{F},\vec{\sigma},\vec{s})$-basis.

2.9 Kernel Basis

Recall that the kernel of $\mathbf{F}\in\mathbb{K}[x]^{m\times n}$ is the $\mathbb{F}[x]$-module

$$\{\mathbf{p}\in\mathbb{K}[x]^{n}\ \mid \mathbf{F}\mathbf{p}=0\}.$$

A kernel basis of $\mathbf{F}$ is just a basis of this module. Kernel bases are closely related to order bases, as can be seen from the following definitions.

Definition 2.23. Given $\mathbf{F}\in\mathbb{K}[x]^{m\times n}$, a polynomial matrix $\mathbf{N}\in\mathbb{K}[x]^{n\times *}$ is a (right) kernel basis of $\mathbf{F}$ if the following properties hold:

1. $\mathbf{N}$ is full-rank.

2. $\mathbf{N}$ satisfies $\mathbf{F} \cdot \mathbf{N} = 0$.

3. Any $\mathbf{q} \in \mathbb{K}[x]^n$ satisfying $\mathbf{Fq} = 0$ can be expressed as a linear combination of the columns of $\mathbf{N}$, that is, there exists some polynomial vector $\mathbf{p}$ such that $\mathbf{q} = \mathbf{Np}$.

Again, it follows from Definition 2.23 and Lemma 2.7 that any pair of kernel bases $\mathbf{N}$ and $\mathbf{M}$ of $\mathbf{F}$ are column bases of each other and are unimodularly equivalent.

An $\vec{s}$-minimal kernel basis of $\mathbf{F}$ is just a kernel basis that is $\vec{s}$-column reduced.

Definition 2.24. Given $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$, a polynomial matrix $\mathbf{N} \in \mathbb{K}[x]^{n \times *}$ is a $\vec{s}$-minimal (right) kernel basis of $\mathbf{F}$ if $\mathbf{N}$ is a kernel basis of $\mathbf{F}$ and $\mathbf{N}$ is $\vec{s}$-column reduced. We also call a $\vec{s}$-minimal (right) kernel basis of $\mathbf{F}$ a $(\mathbf{F}, \vec{s})$-kernel basis in this thesis.

Chapter 3

Order Basis with Balanced Shifts

In this chapter and the next chapter we give algorithms for computing a shifted order basis of an $m \times n$ matrix of power series over a field $\mathbb{K}$ with $m \leq n$. For a given order σ and balanced shift $\vec{s}$ the algorithm in this chapter determines an order basis with a cost of $O^{\sim}(n^{\omega}a)$ field operations in $\mathbb{K}$, where ω is the exponent of matrix multiplication and $a = m\sigma/n$. Here, an input shift is balanced when $\max(\vec{s}) - \min(\vec{s}) \in O(a)$. This extends earlier work of Storjohann which only determines a subset of an order basis that is within a specified degree bound δ using $O^{\sim}(n^{\omega}\delta)$ field operations for $\delta \geq \lceil a \rceil$. In the end of this chapter, we show how a more refined cost of $O^{\sim}(n^{\omega-1}m\sigma)$ instead of $O^{\sim}(n^{\omega}a)$ field operations can be achieved when the shifts are balanced.

In this chapter, we assume, without any loss of generality, that n/m and σ are powers of two. This can be achieved by padding zero rows to the input matrix and multiplying it by some power of x.

We first give a brief description of Storjohann's transformation for computing a partial order basis.

3.1 Balancing Input with Storjohann's Transformation

For computing a $(\mathbf{F}, \sigma, \vec{s})$-basis with input matrix $\mathbf{F} \in \mathbb{K}[[x]]^{m \times n}$, shift $\vec{s}$ and order σ one can view $\mathbf{F}$ as a polynomial matrix with degree $\sigma - 1$, as higher order terms are not needed in the computation. As such the total input size of an order basis problem is $mn\sigma$ coefficients. One can apply the method of Giorgi et al. [2003] directly, which gives a cost of

$$\begin{aligned}
\sum_{i=0}^{\log \sigma} 2^i \operatorname{MM}(n, 2^{-i}\sigma) =& \sum_{i=0}^{\log \sigma} 2^{-i}\sigma \operatorname{MM}(n, 2^i) \\
\subset& O\left(\sum_{i=0}^{\log \sigma} \sigma n^\omega \operatorname{M}\left(2^i\right) 2^{-i}\right) \\
\subset& O\left(n^\omega \sum_{i=0}^{\log \sigma} \operatorname{M}(\sigma)\right) \qquad (3.1) \\
=& O(n^\omega \operatorname{M}(\sigma) \log \sigma).
\end{aligned}$$

Equation (3.1) follows from the fact $2^i \leq \sigma$ implies $\operatorname{M}(2^i)\, 2^{-i} \leq \operatorname{M}(\sigma)/\sigma$. This cost is close to the cost of multiplying two matrices with dimension n and degree σ. Note that this cost is independent of the degree shift. This is very efficient if $m \in \Theta(n)$. However, for small m, say $m = 1$ as in Hermite Padé approximation, the total input size is only $n\sigma$ coefficients. Matrix multiplication cannot be used effectively on a such vector input.

Storjohann [2006] provides a novel way to transform an order basis problem with small row dimension to a problem with higher row dimension and possibly lower degree to take advantage of Giorgi et al. [2003]'s algorithm. We provide a quick overview of a slightly modified version of Storjohann's method. Our small modification allows a nonuniform degree shift for the input and provides a slightly simpler degree shift, degree, and order for the transformed problem. The proof of its correctness is provided in section §3.3. In order to compute a $(\mathbf{F}, \sigma, \vec{s})$-basis, assuming without loss of generality that $\min(\vec{s}) = 0$, we first write

$$\mathbf{F} = \mathbf{F}_0 + \mathbf{F}_1 x^{\delta} + \mathbf{F}_2 x^{2\delta} + \cdots + \mathbf{F}_l x^{l\delta},$$

with $\deg \mathbf{F}_i < \delta$ for a positive integer δ, and where we assume (again without loss of generality) that $\sigma = (l+1)\,\delta$. Set

$$\bar{\mathbf{F}} = \left[\begin{array}{c|cccc} \mathbf{F}_0 + \mathbf{F}_1 x^{\delta} & \mathbf{0}_m & \mathbf{0}_m & \cdots & \mathbf{0}_m \\ \hline \mathbf{F}_1 + \mathbf{F}_2 x^{\delta} & \mathbf{I}_m & \mathbf{0}_m & & \\ \mathbf{F}_2 + \mathbf{F}_3 x^{\delta} & \mathbf{0}_m & \mathbf{I}_m & & \\ \vdots & & & \ddots & \\ \mathbf{F}_{l-1} + \mathbf{F}_l x^{\delta} & & & & \mathbf{I}_m \end{array}\right]_{ml \times (n+m(l-1))}.$$

On the left side of $\bar{\mathbf{F}}$, each block $\mathbf{F}_i + \mathbf{F}_{i+1} x^{\delta}$ has dimension $m \times n$. On the right side, there are $l \times (l-1)$ blocks of $\mathbf{0}_m$'s or $\mathbf{I}_m$'s each having dimension $m \times m$. The overall dimension of $\bar{\mathbf{F}}$ is $ml \times (n + m(l-1))$. Set $\vec{s'} = [\vec{s}, 0, \ldots, 0]$ ($\vec{s}$ followed by $m\,(l-1)$ 0's). A $(\bar{\mathbf{F}}, 2\delta, \vec{s'})$-basis can then be computed by the method of Giorgi et al. with a cost of $O^{\sim}(n^{\omega}\delta)$ for $\delta \geq \lceil a \rceil$, where $a = m\sigma/n$. This transformation of Storjohann can be viewed as a partial linearization of the original problem, where $\bar{\mathbf{F}}$ is analogous to the coefficient matrix of $\mathbf{F}$. Note that $\bar{\mathbf{F}}$ has l block rows each containing m rows. We continue to use each block row to represent m rows for the remainder of this chapter.

Clearly an $(\bar{\mathbf{F}}, 2\delta, \vec{s'})$-basis $\bar{\mathbf{P}}$ of the transformed problem is not a $(\mathbf{F}, \sigma, \vec{s})$-basis of the original problem, as $\bar{\mathbf{P}}$ has a higher dimension and lower degree. However, the first n rows of the $(\bar{\mathbf{F}}, 2\delta, \vec{s'})_{\delta-1}$-basis contained in $\bar{\mathbf{P}}$ is a $(\mathbf{F}, \sigma, \vec{s})_{\delta-1}$-basis.

Note that there is no need to set the degree parameter δ to less than $\lceil a \rceil$, as this produces fewer basis elements without a better cost. The lowest cost is achieved when $\bar{\mathbf{F}}$ is close to square so matrix multiplication can be used most effectively. This requires the number of block rows l of $\bar{\mathbf{F}}$ to be close to n/m, which requires $\delta = \Theta(a)$. Recall that $mn\sigma$ is the total size of the original $m \times n$ input matrix $\mathbf{F}$, hence $a = mn\sigma/n^2 = m\sigma/n$ is the average size of each entry of $\mathbf{F}$ if the m rows of $\mathbf{F}$ are spread out over n rows. Choosing $\delta = \Theta(a)$, the cost of computing a $(\bar{\mathbf{F}}, 2\delta, \vec{s'})$-basis is then $O^{\sim}(n^{\omega}a)$. In the first part of this chapter, we use the average size $a = m\sigma/n$ in the asymptotic cost notation. Therefore, a is assumed to be tending to infinity, which means $m\sigma > n$. Together with the assumption that σ and n/m are both powers of two, $m\sigma/n$ is then always a positive integer in this paper.

Example 3.1. Let $\mathbb{K} = \mathbb{Z}_2$, $\sigma = 8$, $\delta = 2$ and

$$\mathbf{F} = [x+x^2+x^3+x^4+x^5+x^6,\ 1+x+x^5+x^6+x^7,\ 1+x^2+x^4+x^5+x^6+x^7,\ 1+x+x^3+x^7]$$

a vector of size 1×4. Then

$$\bar{\mathbf{F}} = \left[\begin{array}{cccc|cc} x+x^2+x^3 & 1+x & 1+x^2 & 1+x+x^2 & 0 & 0 \\ \hline 1+x+x^2+x^3 & x^3 & 1+x^2+x^3 & x & 1 & 0 \\ 1+x+x^2 & x+x^2+x^3 & 1+x+x^2+x^3 & x^3 & 0 & 1 \end{array}\right]_{3\times 6}$$

and a $\left(\bar{\mathbf{F}}, 4, \vec{0}\right)$-basis is given by

$$\bar{\mathbf{P}} = \left[\begin{array}{cc|cccc} 1 & x & 1 & x^2+x^3 & 0 & x+x^2+x^3 \\ 0 & 1 & 0 & x^2 & x^2+x^3 & 0 \\ 1 & 1+x & x+x^2 & x^2 & x^2 & x^2 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 0 & x^2 & x+x^2+x^3 \\ 0 & 1 & 1+x^2 & 0 & x^2 & x+x^2 \end{array}\right].$$

The first two columns of $\bar{\mathbf{P}}$ have degree less than 2, hence its top left 4×2 submatrix is a $\left(\mathbf{F}, 8, \vec{0}\right)_1$-basis. This is a low degree part of the $(\mathbf{F}, 8, \vec{0})$-basis

$$\mathbf{P} = \begin{bmatrix} 1 & x & 1 & x^2 \\ 0 & 1 & x^2+x^3 & 0 \\ 1 & 1+x & x & x^3+x^4 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Note that if δ is set to $\sigma/2 = 4$, then the transformed problem is the same as the original problem.

3.2 Unbalanced Output

Storjohann's transformation can be used to efficiently compute a $(\mathbf{F}, \sigma, \vec{s})_{\delta-1}$-basis if the degree parameter δ is close to the average degree $d = m\sigma/n$. However, if δ is large, say $\delta = \Theta(\sigma)$, or if we want to compute a complete $(\mathbf{F}, \sigma, \vec{s})$-basis, then the current analysis for the computation still gives the cost estimate of $O^{\sim}(n^{\omega}\sigma)$.

The underlying difficulty with computing a complete order basis is that the basis can have degree up to σ. As the output of this problem has dimension $n \times n$

and degree up to $\Theta(\sigma)$, this may seem to suggest $O^{\sim}(n^{\omega}\sigma)$ is about the best that can be done. However, the total size of the output, that is, the total number of coefficients of all n^2 polynomial entries can still be bounded by $O(mn\sigma)$, the same as the size of the input. This gives some hope for a more efficient method.

Lemma 3.2. *Let $\vec{t}$ be the $\vec{s}$-column degrees of a $(\mathbf{F}, \sigma, \vec{s})$-basis. Then $\sum \vec{t} \leq m\sigma + \sum \vec{s}$. In addition, the total size of any $(\mathbf{F}, \sigma, \vec{s})$-basis in $\vec{s}$-Popov form is bounded by $nm\sigma$.*

Proof. The sum of the $\vec{s}$-column degrees is $\sum \vec{s}$ at order 0, since the identity matrix is a $(\mathbf{F}, 0, \vec{s})$-basis. This sum increases by at most r for each order increase, as can be seen from the iterative computation of order bases in [Beckermann and Labahn, 1994, Giorgi et al., 2003]. The second statement follows from the fact that the row degrees and the $\vec{s}$-column degrees of any $\vec{s}$-Popov form are represented by the pivot entries.. $\square$

Let us now look at the average column degree of the output. In the first part of our discussion on order basis computation, we assume, without loss of generality, that $\min(\vec{s}) = 0$ so $\deg \mathbf{q} \leq \deg_{\vec{s}} \mathbf{q}$ for any $\mathbf{q} \in \mathbb{K}[x]^n$. The situation is simpler if the shift $\vec{s}$ is uniform since then $\sum \vec{t} \leq m\sigma$ by lemma 3.2 and the average column degree is therefore bounded by $a = m\sigma/n$. In the first part of this thesis, we consider a slightly more general case, when the shift $\vec{s}$ is *balanced*, which is defined as follows.

Definition 3.3. A shift $\vec{s}$ is balanced if $\max \vec{s} - \min \vec{s} \in O(a)$ or if $\max \vec{s} - \min \vec{s} \in O(m\sigma/n)$.

Note that we only need to use the second definition using $\max \vec{s} - \min \vec{s} \in O(m\sigma/n)$ when we discuss the more refined cost in section §3.6.

By assuming $\min \vec{s} = 0$, $\vec{s}$ is balanced if $\max \vec{s} \in O(a)$. In this case, lemma 3.2 implies $\sum(\vec{t}) \leq m\sigma + \sum(\vec{s}) \in O(m\sigma + na) = O(m\sigma)$. Hence the average column degree of the output basis remains $O(a)$.

The fact that a $(\mathbf{F}, \sigma, \vec{s})$-basis can have degree up to σ while its average column degree is $O(a)$ implies that an order basis can have quite unbalanced column degrees, especially if m is small. A similar problem with unbalanced output is encountered in null space basis computation. Storjohann and Villard [2005] deal with this in the following way.

Let d be the average column degree of the output. Set the degree parameter δ to twice that of d. This allows one to compute at least half the columns of a basis (since the number of columns with degree at least δ must be at most a half of the total number of columns). One can then simplify the problem, so that the computed basis elements are completely removed from the problem. This reduces the dimension of the problem by at least a factor of 2. One then doubles the degree bound δ in order to have at least 3/4 of the basis elements computed. Repeating this, at iteration i, at most $1/2^i$ of the basis elements are remaining. Therefore, no more than $\log n$ iterations are needed to compute all basis elements.

3.3 Extending Storjohann's Transformation

In this section, we introduce a transformation that can be viewed as an extension of Storjohann's transformation which allows for computation of a full, rather than partial, order basis. More generally (as discussed in the next section) this transformation provides a link between two Storjohann transformed problems constructed using different degree parameters. For easier understanding, we first focus on a particular case of this transformation in Subsection 3.3.1 and then generalize this in Subsection 3.3.2.

3.3.1 A Particular Case

Consider the problem of computing a $(\mathbf{F}, \sigma, \vec{s})$-basis. We assume $\sigma = 4\delta$ for a positive integer δ and write the input matrix polynomial as $\mathbf{F} = \mathbf{F}_0 + \mathbf{F}_1 x^{\delta} + \mathbf{F}_2 x^{2\delta} + \mathbf{F}_3 x^{3\delta}$ with $\deg \mathbf{F}_i < \delta$. In the following, we show that computing a $(\mathbf{F}, \sigma, \vec{s})$-basis can be done by computing a $(\mathbf{F}', \vec{\omega}, \vec{s'})$-basis where

$$\mathbf{F}' = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{F}'_{21} & \mathbf{F}'_{22} \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{F}_0 + \mathbf{F}_1 x^{\delta} + \mathbf{F}_2 x^{2\delta} + \mathbf{F}_3 x^{3\delta} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{F}_1 + \mathbf{F}_2 x^{\delta} & \mathbf{I}_m & \mathbf{0} \\ \mathbf{F}_2 + \mathbf{F}_3 x^{\delta} & \mathbf{0} & \mathbf{I}_m \end{array}\right] \tag{3.2}$$

with order $\vec{\omega} = [4\delta, \dots, 4\delta, 2\delta, \dots, 2\delta]$ (with m 4δ's and $2m$ 2δ's) and degree shift $\vec{s'} = [\vec{s}, e, \dots, e]$ (with $2m$ e's), where e is an integer less than or equal to 1. We set e to 0 in this paper for simplicity[1].

We first look at the correspondence between the elements of $\langle(\mathbf{F}, \sigma, \vec{s})\rangle_{\tau}$ and the elements of $\langle(\mathbf{F}', \vec{\omega}, \vec{s'})\rangle_{\tau}$ in Lemma 3.4 to Lemma 3.8. The correspondence between $(\mathbf{F}, \sigma, \vec{s})$-bases and $(\mathbf{F}', \vec{\omega}, \vec{s'})$-bases is then considered in Corollary 3.10 to Theorem 3.13.

Let

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_n \\ x^{-\delta}\mathbf{F}_0 \\ x^{-2\delta}\left(\mathbf{F}_0 + \mathbf{F}_1 x^{\delta}\right) \end{bmatrix}.$$

Lemma 3.4. *If* $\mathbf{q} \in \langle(\mathbf{F}, \sigma)\rangle$, *then* $\mathbf{B}\mathbf{q} \in \langle(\mathbf{F}', \vec{\omega})\rangle$.

[1] Storjohann used $e = 1$ in [Storjohann, 2006]. All results in this section still hold for any other $e \leq 1$.

Proof. The lemma follows from

$$\mathbf{F}'\mathbf{B}\mathbf{q} = \begin{bmatrix} \mathbf{F}_0 + \mathbf{F}_1 x^{\delta} + \mathbf{F}_2 x^{2\delta} + \mathbf{F}_3 x^{3\delta} \\ \mathbf{F}_0 x^{-\delta} + \mathbf{F}_1 + \mathbf{F}_2 x^{\delta} \\ \mathbf{F}_0 x^{-2\delta} + \mathbf{F}_1 x^{-\delta} + \mathbf{F}_2 + \mathbf{F}_3 x^{\delta} \end{bmatrix} \mathbf{q} \equiv \mathbf{0} \mod x^{\vec{\omega}}.$$

Note that the bottom rows of $\mathbf{B}$ may not be polynomials. However, $\mathbf{B}\mathbf{q}$ is a polynomial vector since $\mathbf{q} \in \langle (\mathbf{F}, \sigma) \rangle$ implies $\mathbf{q} \in \langle (\mathbf{F}_0, \delta) \rangle$ and $\mathbf{q} \in \left\langle \left(\mathbf{F}_0 + \mathbf{F}_1 x^{\delta}, 2\delta\right) \right\rangle$. □

The following lemma shows that the condition $e \leq 1$ forces $\deg_{\vec{s}'} \mathbf{B}\mathbf{q}$ to be determined by $\mathbf{q}$.

Lemma 3.5. *If* $\mathbf{q} \in \langle (\mathbf{F}, \sigma, \vec{s}) \rangle_{\tau}$ *for any degree bound* $\tau \in \mathbb{Z}$*, then* $\deg_{\vec{s}'} \mathbf{B}\mathbf{q} = \deg_{\vec{s}} \mathbf{q}$.

Proof. By assumption $s_i \geq 0$, so $\deg \mathbf{q} \leq \deg_{\vec{s}} \mathbf{q}$. Now consider the degree of the bottom $2m$ entries, $\mathbf{q}_2, \mathbf{q}_3$, of

$$\begin{bmatrix} \mathbf{q} \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} = \mathbf{B}\mathbf{q} = \begin{bmatrix} \mathbf{q} \\ x^{-\delta}\mathbf{F}_0 \cdot \mathbf{q} \\ x^{-2\delta}\left(\mathbf{F}_0 + \mathbf{F}_1 x^{\delta}\right) \cdot \mathbf{q} \end{bmatrix}.$$

Our goal is to show $\deg_{\vec{e}} \left[\mathbf{q}_2^T, \mathbf{q}_3^T\right]^T \leq \deg_{\vec{s}} \mathbf{q}$. Note that

$$\deg \mathbf{q}_2 = \deg\left(\mathbf{F}_0 \mathbf{q} / x^{\delta}\right) \leq \deg \mathbf{q} + \delta - 1 - \delta \leq \deg_{\vec{s}} \mathbf{q} - 1,$$

and similarly $\deg \mathbf{q}_3 \leq \deg_{\vec{s}} \mathbf{q} - 1$. Therefore

$$\deg_{\vec{e}} \begin{bmatrix} \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} = \deg \begin{bmatrix} \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} + e \leq \deg_{\vec{s}} \mathbf{q} - 1 + e \leq \deg_{\vec{s}} \mathbf{q}.$$

□

Corollary 3.6. *If* $\mathbf{q} \in \langle(\mathbf{F},\sigma,\vec{s})\rangle_\tau$ *for any degree bound* $\tau \in \mathbb{Z}$ *, then* $\mathbf{Bq} \in \langle(\mathbf{F}',\vec{\omega},\vec{s'})\rangle_\tau$.

Corollary 3.7. *Let* $\bar{\mathbf{S}}_\tau$ *be a* $(\mathbf{F}',\vec{\omega},\vec{s'})_\tau$*-basis and* $\mathbf{S}_\tau$ *be the top* n *rows of* $\bar{\mathbf{S}}_\tau$ *for any bound* $\tau \in \mathbb{Z}$*. Then any* $\mathbf{q} \in \langle(\mathbf{F},\sigma,\vec{s})\rangle_\tau$ *is a linear combination of the columns of* $\mathbf{S}_\tau$.

Proof. By Corollary 3.6, $\mathbf{Bq} \in \langle(\mathbf{F}',\vec{\omega},\vec{s'})\rangle_\tau$, and so is a linear combination of columns of $\bar{\mathbf{S}}_\tau$. That is, there exists a polynomial vector $\mathbf{u}$ such that $\mathbf{Bq} = \bar{\mathbf{S}}_\tau\mathbf{u}$. This remains true if we restrict the equation to the top n rows, that is, $\mathbf{q} = [\mathbf{I}_n, \mathbf{0}]\,\mathbf{Bq} = [\mathbf{I}_n, \mathbf{0}]\,\bar{\mathbf{S}}_\tau\mathbf{u} = \mathbf{S}_\tau\mathbf{u}$. □

Lemma 3.8. *Let* $\bar{\mathbf{q}} \in \langle(\mathbf{F}',\vec{\omega},\vec{s'})\rangle_\tau$ *for any degree bound* $\tau \in \mathbb{Z}$*, and* $\mathbf{q}_1$ *the first* n *entries of* $\bar{\mathbf{q}}$*. Then* $\mathbf{q}_1 \in \langle(\mathbf{F},\sigma,\vec{s})\rangle_\tau$.

Proof. The top rows of

$$\mathbf{F}'\mathbf{q} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{F}'_{21} & \mathbf{F}'_{22} \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{F}\mathbf{q}_1 \\ \mathbf{F}'_{21}\mathbf{q}_1 + \mathbf{F}'_{22}\mathbf{q}_2 \end{bmatrix} \equiv \mathbf{0} \mod x^{\vec{\omega}}$$

give $\mathbf{F}\mathbf{q}_1 \equiv \mathbf{0} \mod x^\sigma$. □

The next lemma shows a $(\mathbf{F}',\vec{\omega},\vec{s'})$-basis can be constructed from a $(\mathbf{F},\sigma,\vec{s})$-basis. This $(\mathbf{F}',\vec{\omega},\vec{s'})$-basis has a structure that restricts the elements of $\langle(\mathbf{F}',\vec{\omega},\vec{s'})\rangle$ to a simple form shown in Corollary 3.10. This in turn helps to establish a close correspondence between a $(\mathbf{F}',\vec{\omega},\vec{s'})$-basis and a $(\mathbf{F},\sigma,\vec{s})$-basis in Lemma 3.11, Lemma 3.12, and Theorem 3.13.

Lemma 3.9. *If* $\mathbf{P}$ *is a* $(\mathbf{F},\sigma,\vec{s})$*-basis, then*

$$\bar{\mathbf{T}} = \left[\mathbf{BP} \left| \begin{matrix} \mathbf{0}_{n\times 2m} \\ x^{2\delta}\mathbf{I}_{2m} \end{matrix} \right.\right] = \left[\begin{array}{r|cc} \mathbf{P} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times m} \\ \hline x^{-\delta}\mathbf{F}_0 \cdot \mathbf{P} & x^{2\delta}\mathbf{I}_m & \mathbf{0}_m \\ x^{-2\delta}\left(\mathbf{F}_0 + \mathbf{F}_1 x^\delta\right) \cdot \mathbf{P} & \mathbf{0}_m & x^{2\delta}\mathbf{I}_m \end{array}\right]$$

is a $(\mathbf{F}', \vec{\omega}, \vec{s'})$*-basis.*

Proof. By Lemma 3.4, $\bar{\mathbf{T}}$ has order $(\mathbf{F}', \vec{\omega})$ and is $\vec{s'}$-column reduced since $\mathbf{P}$ dominates the $\vec{s'}$-degrees of $\bar{\mathbf{T}}$ on the left side by Lemma 3.5. It remains to show that any $\bar{\mathbf{q}} \in \langle(\mathbf{F}', \vec{\omega}, \vec{s'})\rangle$ is a linear combination of the columns of $\bar{\mathbf{T}}$.

Let $\mathbf{q}$ be the top n entries of $\bar{\mathbf{q}}$. Then by Lemma 3.8, $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle$, hence is a linear combination of the columns of $\mathbf{P}$, that is $\mathbf{q} = \mathbf{P}\mathbf{u}$ with $\mathbf{u} = \mathbf{P}^{-1}\mathbf{q} \in \mathbb{K}[x]^{n\times 1}$. Subtracting the contribution of $\mathbf{P}$ from $\bar{\mathbf{q}}$, we get

$$\mathbf{q}' = \bar{\mathbf{q}} - \mathbf{B}\mathbf{P}\mathbf{u} = \bar{\mathbf{q}} - \mathbf{B}\mathbf{q} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v} \end{bmatrix},$$

which is still in $\langle(\mathbf{F}', \vec{\omega}, \vec{s'})\rangle$, that is,

$$\mathbf{F}'\mathbf{q}' = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{2m}\mathbf{v} \end{bmatrix} \equiv \mathbf{0} \mod x^{\vec{\omega}}.$$

This forces $\mathbf{v}$ to be a linear combination of the columns of $x^{2\delta}\mathbf{I}_{2m}$, the bottom right submatrix of $\bar{\mathbf{T}}$. Now $\bar{\mathbf{q}} = \bar{\mathbf{T}}\left[\mathbf{u}^T, \mathbf{v}^T\right]^T$ as required. □

Corollary 3.10. *Let $\tau \in \mathbb{Z}$ be any degree bound and $\mathbf{P}_\tau \in \mathbb{K}[x]^{n\times t}$ be a $(\mathbf{F}, \sigma, \vec{s})_\tau$-basis. If $\bar{\mathbf{q}} \in \langle(\mathbf{F}', \vec{\omega}, \vec{s'})\rangle_\tau$ and $\mathbf{q}$ is the top n entries of $\bar{\mathbf{q}}$, then $\bar{\mathbf{q}}$ must have the form*

$$\bar{\mathbf{q}} = \mathbf{B}\mathbf{P}_\tau\mathbf{u} + x^{2\delta}\begin{bmatrix} \mathbf{0} \\ \mathbf{v} \end{bmatrix} = \mathbf{B}\mathbf{q} + x^{2\delta}\begin{bmatrix} \mathbf{0} \\ \mathbf{v} \end{bmatrix}$$

for some polynomial vector $\mathbf{u} \in \mathbb{K}[x]^{t\times 1}$ and $\mathbf{v} \in \mathbb{K}[x]^{2m\times 1}$. In particular, if $\deg_{\vec{s'}} \bar{\mathbf{q}} < 2\delta$, then $\bar{\mathbf{q}} = \mathbf{B}\mathbf{P}_\tau\mathbf{u} = \mathbf{B}\mathbf{q}$.

Proof. This follows directly from Lemma 3.9 with $\vec{s'}$-degrees restricted to τ. □

Lemma 3.11. *If* $\bar{\mathbf{S}}^{(1)}$ *is a* $(\check{\mathbf{F}}, \vec{\omega}, \vec{s'})_{2\delta-1}$*-basis, then the matrix* $\mathbf{S}^{(1)}$ *consisting of its first* n *rows is a* $(\mathbf{F}, \sigma, \vec{s})_{2\delta-1}$*-basis.*

Proof. By Lemma 3.8, $\mathbf{S}^{(1)}$ has order $(\mathbf{F}, \sigma)$. By Corollary 3.7, any $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle_{2\delta-1}$ is a linear combination of $\mathbf{S}^{(1)}$. It remains to show that $\mathbf{S}^{(1)}$ is $\vec{s}$-column reduced.

By Corollary 3.10, $\bar{\mathbf{S}}^{(1)} = \mathbf{B}\mathbf{S}^{(1)}$, and by Lemma 3.8, the columns of $\mathbf{S}^{(1)}$ are in $\langle(\mathbf{F}, \sigma, \vec{s})\rangle_{2\delta-1}$. Thus, by Lemma 3.5, $\mathbf{S}^{(1)}$ determines the $\vec{s'}$-column degrees of $\mathbf{S}^{(1)}$. Therefore, $\bar{\mathbf{S}}^{(1)}$ being $\vec{s'}$-column reduced implies that $\mathbf{S}^{(1)}$ is $\vec{s}$-column reduced. □

Lemma 3.12. *Let* $\bar{\mathbf{S}}^{(12)} = [\bar{\mathbf{S}}^{(1)}, \bar{\mathbf{S}}^{(2)}]$ *be a* $(\mathbf{F}', \vec{\omega}, \vec{s'})_{2\delta}$*-basis, with* $\deg_{\vec{s'}} \bar{\mathbf{S}}^{(1)} \leq 2\delta - 1$ *and* $\deg_{\vec{s'}} \bar{\mathbf{S}}^{(2)} = 2\delta$*, and* $\mathbf{S}^{(12)}, \mathbf{S}^{(1)}, \mathbf{S}^{(2)}$ *the first* n *rows of* $\bar{\mathbf{S}}^{(12)}, \bar{\mathbf{S}}^{(1)}, \bar{\mathbf{S}}^{(2)}$*, respectively. Let* I *be the column rank profile (the lexicographically smallest sequence of column indices that indicates a full column rank submatrix) of* $\mathbf{S}^{(12)}$*. Then the submatrix* $\mathbf{S}_I^{(12)}$ *comprised of the columns of* $\mathbf{S}^{(12)}$ *indexed by* I *is a* $(\mathbf{F}, \sigma, \vec{s})_{2\delta}$*-basis.*

Proof. Consider doing $\vec{s}$-column reduction on $\mathbf{S}^{(12)}$. From Lemma 3.11, we know that $\mathbf{S}^{(1)}$ is a $(\mathbf{F}, \sigma, \vec{s})_{2\delta-1}$-basis. Therefore, only $\mathbf{S}^{(2)}$ may be $\vec{s}$-reduced. If a column $\mathbf{c}$ of $\mathbf{S}^{(2)}$ can be further $\vec{s}$-reduced, then it becomes an element of $\langle(\mathbf{F}, \sigma, \vec{s})\rangle_{2\delta-1}$, which is generated by $\mathbf{S}^{(1)}$. Thus $\mathbf{c}$ must be reduced to zero by $\mathbf{S}^{(1)}$. The only nonzero columns of $\mathbf{S}^{(12)}$ remaining after $\vec{s}$-column reduction are therefore the columns that cannot be $\vec{s}$-reduced. Hence $\mathbf{S}^{(12)}$ $\vec{s}$-reduces to $\mathbf{S}_I^{(12)}$. In addition, $\mathbf{S}_I^{(12)}$ has order $(\mathbf{F}, \sigma)$ as $\mathbf{S}^{(12)}$ has order $(\mathbf{F}, \sigma)$ by Lemma 3.8. From Corollary 3.7 any $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle_{2\delta}$ is a linear combination of $\mathbf{S}^{(12)}$ and hence is also a linear combination of $\mathbf{S}_I^{(12)}$. □

To extract $\mathbf{S}_I^{(12)}$ from $\mathbf{S}^{(12)}$, note that doing $\vec{s}$-column reduction on $\mathbf{S}^{(12)}$ is equivalent to the more familiar problem of doing column reduction on $x^{\vec{s}}\mathbf{S}^{(12)}$. As $\mathbf{S}^{(12)}$ $\vec{s}$-column reduces to $\mathbf{S}_I^{(12)}$, this corresponds to determining the column rank profile of the *leading column coefficient matrix* of $x^{\vec{s}}\mathbf{S}^{(12)}$. Recall that the leading column coefficient matrix of a matrix $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_k]$ used for column reduction is

$$\begin{aligned}\operatorname{lcoeff}(\mathbf{A}) &= [\operatorname{lcoeff}(\mathbf{a}_1), \dots, \operatorname{lcoeff}(\mathbf{a}_k)] \\ &= [\operatorname{coeff}(\mathbf{a}_1, \deg(\mathbf{a}_1)), \dots, \operatorname{coeff}(\mathbf{a}_k, \deg(\mathbf{a}_k))].\end{aligned}$$

The column rank profile of $\operatorname{lcoeff}(x^{\vec{s}}\mathbf{S}^{(12)})$ can be determined by (the transposed version of) LSP factorization [Ibarra et al., 1982], which factorizes $\operatorname{lcoeff}(x^{\vec{s}}\mathbf{S}^{(12)}) = PSU$ as the product of a permutation matrix P, a matrix S with its nonzero columns forming a lower triangular submatrix, and an upper triangular matrix U with 1's on the diagonal. The indices, I, of the nonzero columns of S then give $\mathbf{S}_I^{(12)}$ in $\mathbf{S}^{(12)}$.

Theorem 3.13. *Let $\bar{\mathbf{S}} = [\bar{\mathbf{S}}^{(12)}, \bar{\mathbf{S}}^{(3)}]$ be a $(\mathbf{F}', \vec{\omega}, \vec{s'})$-basis, with $\deg_{\vec{s'}} \bar{\mathbf{S}}^{(12)} \leq 2\delta$ and $\deg_{\vec{s'}} \bar{\mathbf{S}}^{(3)} \geq 2\delta + 1$, and $\mathbf{S}, \mathbf{S}^{(12)}, \mathbf{S}^{(3)}$ the first n rows of $\bar{\mathbf{S}}, \bar{\mathbf{S}}^{(12)}, \bar{\mathbf{S}}^{(3)}$, respectively. If I is the column rank profile of $\mathbf{S}^{(12)}$, then the submatrix $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$ of $\mathbf{S}$ is a $(\mathbf{F}, \sigma, \vec{s})$-basis.*

Proof. By Lemma 3.8, $\mathbf{S}$ has order $(\mathbf{F}, \sigma)$, and so $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$ also has order $(\mathbf{F}, \sigma)$. By Corollary 3.7, any $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle$ is a linear combination of the columns of $\mathbf{S}$, and so $\mathbf{q}$ is also a linear combination of the columns of $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$. It only remains to show that $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$ is $\vec{s}$-column reduced.

Let $\mathbf{P}$ be a $(\mathbf{F}, \sigma, \vec{s})$-basis and $\bar{\mathbf{T}}$ be the $(\mathbf{F}', \vec{\omega}, \vec{s'})$-basis constructed from $\mathbf{P}$ as in Lemma 3.9. Let $\bar{\mathbf{T}}^{(3)}$ be the columns of $\bar{\mathbf{T}}$ with $\vec{s'}$-degrees greater than 2δ, and $\mathbf{P}^{(3)}$ be the columns of $\mathbf{P}$ with $\vec{s}$-degrees greater than 2δ. Assume without loss of generality that $\mathbf{S}$, $\mathbf{P}$, and $\bar{\mathbf{T}}$ have their columns sorted according to their $\vec{s}$-degrees and $\vec{s'}$-degrees, respectively. Then $\deg_{\vec{s}} \mathbf{S}^{(3)} \leq \deg_{\vec{s'}} \bar{\mathbf{S}}^{(3)} = \deg_{\vec{s'}} \bar{\mathbf{T}}^{(3)} = \deg_{\vec{s}} \mathbf{P}^{(3)}$. Combining this with the $\vec{s}$-minimality of $\mathbf{S}_I^{(12)}$ from Lemma 3.12, it follows that $\deg_{\vec{s}}[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}] \leq \deg_{\vec{s}} \mathbf{P}$. This combined with the fact that $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$ still generates $\langle(\mathbf{F}, \sigma, \vec{s})\rangle$ implies that $\deg_{\vec{s}}[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}] = \deg_{\vec{s}} \mathbf{P}$. Therefore, $[\mathbf{S}_I^{(12)}, \mathbf{S}^{(3)}]$

is a $(\mathbf{F}, \sigma, \vec{s})$-basis. $\square$

Corollary 3.14. *Let* $\bar{\mathbf{S}}$ *be a* $(\mathbf{F}', \vec{\omega}, \vec{s'})$*-basis with its columns sorted in an increasing order of their* $\vec{s'}$ *degrees, and* $\mathbf{S}$ *the first* n *rows of* $\bar{\mathbf{S}}$*. If* J *is the column rank profile of* $\operatorname{lcoeff}(x^{\vec{s}}\mathbf{S})$*, then the submatrix* $\mathbf{S}_J$ *of* $\mathbf{S}$ *indexed by* J *is a* $(\mathbf{F}, \sigma, \vec{s})$*-basis.*

Proof. This follows directly from Theorem 3.13. $\square$

This rank profile J can be determined by LSP factorization on $\operatorname{lcoeff}(x^{\vec{s}} \cdot \mathbf{S}^{(12)})$.

Example 3.15. For the problem in Example 3.1, $\check{\mathbf{F}}$ is given by

$$\begin{bmatrix} x+x^2+x^3+x^4+x^5+x^6 & 1+x+x^5+x^6+x^7 & 1+x^2+x^6+x^7 & 1+x+x^3+x^7 & 0 & 0 \\ 1+x+x^2+x^3 & x^3 & 1+x^2+x^3 & x & 1 & 0 \\ 1+x+x^2 & x+x^2+x^3 & 1+x+x^2+x^3 & x^3 & 0 & 1 \end{bmatrix}$$

and a $\left(\mathbf{F}', [8,4,4], \vec{0}\right)$-basis is given as

$$\left[\begin{array}{cccc|cc} 1 & x & 1 & x^2 & x^2+x^4 & 1+x^2+x^3+x^4 \\ 0 & 1 & x^2+x^3 & 0 & x^3 & 0 \\ 1 & 1+x & x & x^3+x^4 & 0 & x+x^2+x^3 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1+x^2 & x^2 & x^2+x^3 & 1+x^2+x^3+x^4 \\ 0 & 1 & 1 & x^2+x^4 & x^2+x^3 & 1+x^3 \end{array}\right].$$

Column reduction on the top 4 rows gives the top left 4×4 submatrix, which is a $(\mathbf{F}, 8, \vec{0})$-basis.

The following two lemmas verify Storjohann's result in the case of degree parameter $\delta = \sigma/4$. More specifically, we show that the matrix of the top n rows of a $(\bar{\mathbf{F}}, 2\delta, \vec{s'})_{\delta-1}$-basis is a $(\mathbf{F}, \sigma, \vec{s})_{\delta-1}$-basis, with the transformed input matrix

$$\bar{\mathbf{F}} = \left[\begin{array}{c|cc} \mathbf{F}_0 + \mathbf{F}_1 x^\delta & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{F}_1 + \mathbf{F}_2 x^\delta & \mathbf{I}_m & \mathbf{0} \\ \mathbf{F}_2 + \mathbf{F}_3 x^\delta & \mathbf{0} & \mathbf{I}_m \end{array}\right] \equiv \mathbf{F}' \mod x^{2\delta}. \tag{3.3}$$

Lemma 3.16. *If $\bar{\mathbf{q}} \in \langle(\bar{\mathbf{F}}, 2\delta, \vec{s'})\rangle_{\delta-1}$ and $\mathbf{q}$ denotes the first n entries of $\bar{\mathbf{q}}$, then $\bar{\mathbf{q}}$ must have the form*

$$\bar{\mathbf{q}} = \mathbf{B}\mathbf{q} = \begin{bmatrix} \mathbf{q} \\ x^{-\delta}\mathbf{F}_0 \cdot \mathbf{q} \\ x^{-2\delta}\left(\mathbf{F}_0 + \mathbf{F}_1 x^\delta\right) \cdot \mathbf{q} \end{bmatrix}$$

and $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle_{\delta-1}$.

Proof. Let $\mathbf{q}, \mathbf{q}_2, \mathbf{q}_3$ consist of the top n entries, middle m entries, and bottom m entries, respectively, of $\bar{\mathbf{q}}$ so that

$$\bar{\mathbf{F}}\bar{\mathbf{q}} \equiv \begin{bmatrix} \mathbf{F}_0\mathbf{q} + x^\delta \mathbf{F}_1 \mathbf{q} \\ \mathbf{q}_2 + \mathbf{F}_1\mathbf{q} + x^\delta \mathbf{F}_2\mathbf{q} \\ \mathbf{q}_3 + \mathbf{F}_2\mathbf{q} + x^\delta \mathbf{F}_3 \mathbf{q} \end{bmatrix} \equiv \mathbf{0} \mod x^{2\delta}. \tag{3.4}$$

From the first and second block rows, we get $\mathbf{F}_0\mathbf{q} + x^\delta\mathbf{F}_1\mathbf{q} \equiv \mathbf{0} \mod x^{2\delta}$ and $\mathbf{q}_2 + \mathbf{F}_1\mathbf{q} \equiv \mathbf{0} \mod x^\delta$, which implies

$$\mathbf{F}_0\mathbf{q} \equiv x^\delta \mathbf{q}_2 \mod x^{2\delta}. \tag{3.5}$$

Similarly, from the second and third rows, we get $\mathbf{q}_2 + \mathbf{F}_1\mathbf{q} + x^\delta\mathbf{F}_2\mathbf{q} \equiv \mathbf{0} \mod x^{2\delta}$ and $\mathbf{q}_3 + \mathbf{F}_2\mathbf{q} \equiv \mathbf{0} \mod x^\delta$, which implies $\mathbf{q}_2 + \mathbf{F}_1\mathbf{q} \equiv x^\delta\mathbf{q}_3 \mod x^{2\delta}$.

Since $\deg \mathbf{q} \le \deg_{\vec{s}} \mathbf{q} = \delta - 1$, we have $\deg \mathbf{F}_0\mathbf{q} \le 2\delta - 2$, hence from (3.5)

$\deg \mathbf{q}_2 \le \delta - 2$ and $\mathbf{q}_2 x^{\delta} = \mathbf{F}_0\mathbf{q}$. Similarly, $\deg \mathbf{q}_3 \le \delta - 2$ and $\mathbf{q}_3 x^{2\delta} = \mathbf{q}_2 x^{\delta} + \mathbf{F}_1\mathbf{q}x^{\delta} = \mathbf{F}_0\mathbf{q}+\mathbf{F}_1\mathbf{q}x^{\delta}$. Substituting this to $\mathbf{F}\mathbf{q} = (\mathbf{F}_0\mathbf{q}+\mathbf{F}_1\mathbf{q}x^{\delta})+(\mathbf{F}_2\mathbf{q}x^{2\delta}+\mathbf{F}_3\mathbf{q}x^{3\delta})$, we get $\mathbf{F}\mathbf{q} = \mathbf{q}_3 x^{2\delta} + (\mathbf{F}_2\mathbf{q}x^{2\delta} + \mathbf{F}_3\mathbf{q}x^{3\delta}) \equiv \mathbf{0} \mod x^{4\delta}$ using the bottom block row of (3.4). □

Lemma 3.17. *If $\bar{\mathbf{S}}_{\delta-1}$ is a $(\bar{\mathbf{F}}, 2\delta, \vec{s'})_{\delta-1}$-basis, then the matrix of its first n rows, $\mathbf{S}_{\delta-1}$, is a $(\mathbf{F}, \sigma, \vec{s})_{\delta-1}$-basis.*

Proof. By Lemma 3.16, $\mathbf{S}_{\delta-1}$ has order $(\mathbf{F}, \sigma)$. Following Lemmas 3.4 and 3.5 and Corollaries 3.6 and 3.7 (replacing $\vec{\omega}$ by 2δ), we conclude that any $\mathbf{q} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle_{\delta-1}$ is a linear combination of the columns of $\mathbf{S}_{\delta-1}$. In addition, since $\bar{\mathbf{S}}_{\delta-1} = \mathbf{B}\mathbf{S}_{\delta-1}$ by Lemma 3.16, and the columns of $\mathbf{S}_{\delta-1}$ are in $\langle(\mathbf{F}, \sigma, \vec{s})\rangle_{\delta-1}$, it follows from Lemma 3.5 that $\mathbf{S}_{\delta-1}$ determines the $\vec{s'}$-column degrees of $\bar{\mathbf{S}}_{\delta-1}$. Hence $\bar{\mathbf{S}}_{\delta-1}$ $\vec{s'}$-column reduced implies that $\mathbf{S}_{\delta-1}$ is $\vec{s}$-column reduced. □

3.3.2 More General Results

Let us now consider an immediate extension of the results in the previous subsection. Suppose that instead of a $(\mathbf{F}, \sigma, \vec{s})$-basis we now want to compute a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis with a Storjohann transformed input matrix

$$\bar{\mathbf{F}}^{(i)} = \left[\begin{array}{c|cccc} \mathbf{F}_0 + \mathbf{F}_1 x^{\delta^{(i)}} & \mathbf{0}_m & \cdots & \cdots & \mathbf{0}_m \\ \hline \mathbf{F}_1 + \mathbf{F}_2 x^{\delta^{(i)}} & \mathbf{I}_m & & & \\ \mathbf{F}_2 + \mathbf{F}_3 x^{\delta^{(i)}} & & \mathbf{I}_m & & \\ \vdots & & & \ddots & \\ \mathbf{F}_{l^{(i)}-1} + \mathbf{F}_{l^{(i)}} x^{\delta^{(i)}} & & & & \mathbf{I}_m \end{array}\right]_{ml^{(i)} \times (n+m(l^{(i)}-1))}$$

made with degree parameter $\delta^{(i)} = 2^i d$ for some integer i between 2 and $\log(\sigma/d)-1$, and a shift $\vec{s}^{(i)} = [\vec{s}, 0, \ldots, 0]$ (with $m(l^{(i)}-1)$ 0's), where $l^{(i)} = \sigma/\delta^{(i)} - 1$ is the number of block rows[2]. To apply a transformation analogous to (3.2), we write each $\mathbf{F}_j = \mathbf{F}_{j0} + \mathbf{F}_{j1}\delta^{(i-1)}$ and set

$$\mathbf{F}'^{(i)} = \left[\begin{array}{l|c} \mathbf{F}_{00} + \mathbf{F}_{01}x^{\delta^{(i-1)}} + \mathbf{F}_{10}x^{2\delta^{(i-1)}} + \mathbf{F}_{11}x^{3\delta^{(i-1)}} & \mathbf{0} \\ \hline \mathbf{F}_{01} + \mathbf{F}_{10}x^{\delta^{(i-1)}} & \\ \mathbf{F}_{10} + \mathbf{F}_{11}x^{\delta^{(i-1)}} + \mathbf{F}_{20}x^{2\delta^{(i-1)}} + \mathbf{F}_{21}x^{3\delta^{(i-1)}} & \\ \mathbf{F}_{11} + \mathbf{F}_{20}x^{\delta^{(i-1)}} & \\ \vdots & \mathbf{I} \\ \mathbf{F}_{\left(l^{(i)}-1\right)0} + \mathbf{F}_{\left(l^{(i)}-1\right)1}x^{\delta^{(i-1)}} + \mathbf{F}_{l^{(i)}0}x^{2\delta^{(i-1)}} + \mathbf{F}_{l^{(i)}1}x^{3\delta^{(i-1)}} & \\ \mathbf{F}_{\left(l^{(i)}-1\right)1} + \mathbf{F}_{l^{(i)}0}x^{\delta^{(i-1)}} & \\ \mathbf{F}_{l^{(i)}0} + \mathbf{F}_{l^{(i)}1}x^{\delta^{(i-1)}} & \end{array}\right], \quad (3.6)$$

and $\vec{\omega}^{(i)} = \left[\left[[2\delta^{(i)}]^m, [\delta^{(i)}]^m\right]^{l^{(i)}}, [\delta^{(i)}]^m\right]$, where $[\circ]^k$ represents $\circ$ repeated k times. The order entries $2\delta^{(i)}$, $\delta^{(i)}$ in $\vec{\omega}^{(i)}$ correspond to the degree $2\delta^{(i)} - 1$, degree $\delta^{(i)} - 1$ rows in $\mathbf{F}'^{(i)}$ respectively. Let

$$\mathbf{E}^{(i)} = \left[\begin{array}{c||c|c|c|c||c} \mathbf{I}_n & & & & & \mathbf{0}_{n\times m} \quad \mathbf{0}_{n\times m} \\ \hline\hline & \mathbf{0}_m \quad \mathbf{I}_m & & & & \\ \hline & & \mathbf{0}_m \quad \mathbf{I}_m & & & \\ \hline & & & \ddots \quad \ddots & & \\ \hline & & & & \mathbf{0}_m \quad \mathbf{I}_m & \end{array}\right]$$

with $l^{(i)}-1$ blocks of $[\mathbf{0}_m, \mathbf{I}_m]$ and hence an overall dimension of $(n+m(l^{(i)}-1)) \times (n+m(l^{(i-1)}-1))$. Thus $\mathbf{E}^{(i)}\mathbf{M}$ picks out from $\mathbf{M}$ the first n rows and the even block rows from the remaining rows except the last block row for a matrix $\mathbf{M}$ with $n+m(l^{(i-1)}-$

[2]Recall that $d = m\sigma/n$ is the average degree of the input matrix $\mathbf{F}$ if we treat $\mathbf{F}$ as a square $n \times n$ matrix. Also, i starts at 2 because $i = 1$ is our base case in the computation of an order basis, which may become more clear in the next section. The base case can be computed efficiently using the method of Giorgi et al. [2003] directly and does not require the transformation discussed in this section.

1) rows. In particular, if $i = \log(n/m) - 1$, then $(\mathbf{F}'^{(i)}, \vec{\omega}^{(i)}, \vec{s}^{(i-1)}) = (\mathbf{F}', \vec{\omega}, \vec{s'})$, which for $d = m\sigma/n$ gives the problem considered earlier in Subsection 3.3.1, and $\mathbf{E}^{(i)} = [\mathbf{I}_n, \mathbf{0}_{n\times m}, \mathbf{0}_{n\times m}]$ is used to select the top n rows of a $(\mathbf{F}', \vec{\omega}, \vec{s'})$-basis for a $(\mathbf{F}, \sigma, \vec{s})$-basis to be extracted.

We can now state the analog of Corollary 3.14:

Theorem 3.18. *Let $\mathbf{S}'^{(i)}$ be a $(\mathbf{F}'^{(i)}, \vec{\omega}^{(i)}, \vec{s}^{(i-1)})$-basis with its columns sorted in an increasing order of their $\vec{s}^{(i-1)}$ degrees. Let $\hat{\mathbf{S}}^{(i)} = \mathbf{E}^{(i)}\mathbf{S}'^{(i)}$. Let J be the column rank profile of* $\operatorname{lcoeff}(x^{\vec{s}^{(i)}}\hat{\mathbf{S}}^{(i)})$. *Then $\hat{\mathbf{S}}^{(i)}_J$ is a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis.*

Proof. One can follow the same arguments used before from Lemma 3.4 to Corollary 3.14. Alternatively, this can be derived from Corollary 3.14 by noticing the redundant block rows that can be disregarded after applying transformation (3.2) directly to the input matrix $\bar{\mathbf{F}}^{(i)}$. □

Lemma 3.17 can also be extended in the same way to capture Storjohann's transformation with more general degree parameters:

Lemma 3.19. *If $\bar{\mathbf{P}}_1^{(i-1)}$ is a $(\bar{\mathbf{F}}^{(i-1)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})_{\delta^{(i-1)}-1}$-basis, then $\mathbf{E}^{(i)}\bar{\mathbf{P}}_1^{(i-1)}$ is a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})_{\delta^{(i-1)}-1}$-basis and the matrix consists of the top n rows of $\bar{\mathbf{P}}_1^{(i-1)}$ is a $(\mathbf{F}, \sigma, \vec{s})_{\delta^{(i-1)}-1}$-basis.*

Proof. Again, this can be justified as done in Lemma 3.17. Alternatively, one can apply Storjohann's transformation with degree parameter $\delta^{(i-1)}$ to $\bar{\mathbf{F}}^{(i)}$ as in (3.3). The lemma then follows from Lemma 3.17 after noticing the redundant block rows that can be disregarded. □

Notice that if $i = \log(n/m) - 1$, then Theorem 3.18 and Lemma 3.19 specialize to Corollary 3.14 and Lemma 3.17.

3.4 Computation of Order Bases

In this section, we establish a link between two different Storjohann transformed problems by dividing the transformed problem from the previous section into two subproblems and then simplifying the second subproblem. This leads to a recursive method for computing order bases. We also present an equivalent, iterative method for computing order bases. The iterative approach is usually more efficient in practice, as it uses just $O(1)$ iterations in the generic case.

3.4.1 Dividing into Subproblems

In Section 3.3 we have shown that the problem of computing a $(\mathbf{F}, \sigma, \vec{s})$-basis can be converted to the problem of computing a $(\mathbf{F}', \vec{\omega}, \vec{s'})$-basis and, more generally, that the computation of a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis, a Storjohann transformed problem with degree parameter $\delta^{(i)}$, can be converted to the problem of computing a $(\mathbf{F}'^{(i)}, \vec{\omega}^{(i)}, \vec{s}^{(i-1)})$-basis. We now consider dividing the new converted problem into two subproblems.

The first subproblem is to compute a $(\mathbf{F}'^{(i)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})$-basis or equivalently a $(\bar{\mathbf{F}}^{(i-1)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})$-basis $\bar{\mathbf{P}}^{(i-1)}$, a Storjohann transformed problem with degree parameter $\delta^{(i-1)}$. The second subproblem is computing a $(\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}, \vec{\omega}^{(i)}, \vec{t}^{(i-1)})$-basis $\bar{\mathbf{Q}}^{(i)}$ using the residual $\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}$ from the first subproblem along with a degree shift $\vec{t}^{(i-1)} = \deg_{\vec{s}^{(i-1)}} \bar{\mathbf{P}}^{(i-1)}$. From Theorem 5.1 in [Beckermann and Labahn, 1997] we then know that the product $\bar{\mathbf{P}}^{(i-1)}\bar{\mathbf{Q}}^{(i)}$ is a $(\mathbf{F}'^{(i)}, \vec{\omega}^{(i)}, \vec{s}^{(i-1)})$-basis and $\deg_{\vec{s}^{(i-1)}} \bar{\mathbf{P}}^{(i-1)}\bar{\mathbf{Q}}^{(i)} = \deg_{\vec{t}^{(i-1)}} \bar{\mathbf{Q}}^{(i)}$. For completeness, we state a version of this theorem specialized for our needs below and provide a simpler proof.

Theorem 3.20. *For an input matrix $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$, an order vector $\vec{\sigma}$, and a shift vector $\vec{s}$, if $\mathbf{P}$ is a $(\mathbf{F}, \vec{\sigma}, \vec{s})$-basis with $\vec{s}$-column degrees $\vec{t}$, and $\mathbf{Q}$ is a $(\mathbf{FP}, \vec{\tau}, \vec{t})$ -basis with $\vec{t}$-column degrees $\vec{u}$, where $\vec{\tau} \geq \vec{\sigma}$ component-wise, then $\mathbf{PQ}$ is a $(\mathbf{F}, \vec{\tau}, \vec{s})$-basis* *with $\vec{s}$-column degrees $\vec{u}$.*

Proof. It is clear that $\mathbf{PQ}$ has order $(\mathbf{F}, \vec{\tau})$. We now show that $\mathbf{PQ}$ is $\vec{s}$-column reduced and has $\vec{s}$-column degrees $\vec{u}$, or equivalently, $x^{\vec{s}}\mathbf{PQ}$ is column reduced and has column degrees $\vec{u}$. Notice that $x^{\vec{s}}\mathbf{P}$ has column degrees $\vec{t}$ and a full rank leading column coefficient matrix P. Hence $x^{\vec{s}}\mathbf{P}x^{-\vec{t}}$ has column degrees $[0, \ldots 0]$. Similarly, $x^{\vec{t}}\mathbf{Q}x^{-\vec{u}}$ has column degrees $[0, \ldots, 0]$ and a full rank leading column coefficient matrix Q. Therefore, $x^{\vec{s}}\mathbf{P}x^{-\vec{t}}x^{\vec{t}}\mathbf{Q}x^{-\vec{u}} = x^{\vec{s}}\mathbf{PQ}x^{-\vec{u}}$ has column degrees $[0, \ldots, 0]$ and a full rank leading column coefficient matrix PQ, implying that $x^{\vec{s}}\mathbf{PQ}$ is column reduced and that $x^{\vec{s}}\mathbf{PQ}$ has column degrees $\vec{u}$, or equivalently, the $\vec{s}$-column degrees of $\mathbf{PQ}$ is $\vec{u}$.

It remains to show that any $\mathbf{t} \in \langle(\mathbf{F}, \vec{\tau})\rangle$ is generated by the columns of $\mathbf{PQ}$. Since $\mathbf{t} \in \langle(\mathbf{F}, \vec{\sigma})\rangle$, it is generated by the $(\mathbf{F}, \vec{\sigma})$-basis $\mathbf{P}$, that is, $\mathbf{t} = \mathbf{Pa}$ for $\mathbf{a} = \mathbf{P}^{-1}\mathbf{t} \in \mathbb{K}[x]^n$. Also, $\mathbf{t} \in \langle(\mathbf{F}, \vec{\tau})\rangle$ implies that $\mathbf{a} \in \langle(\mathbf{FP}, \vec{\tau})\rangle$ since $\mathbf{FPa} = \mathbf{Ft} \equiv 0 \mod x^{\vec{\tau}}$. It follows that $\mathbf{a} = \mathbf{Qb}$ for $\mathbf{b} = \mathbf{Q}^{-1}\mathbf{a} \in \mathbb{K}[x]^n$. Therefore, $\mathbf{a} = \mathbf{P}^{-1}\mathbf{t} = \mathbf{Qb}$, which gives $\mathbf{t} = \mathbf{PQb}$. □

Example 3.21. Let us continue with Example 3.1 and Example 3.15 in order to compute a $\left(\mathbf{F}, 8, \vec{0}\right)$-basis (or equivalently a $(\bar{\mathbf{F}}^{(2)}, 8, \vec{0})$-basis). This can be determined by computing a $(\mathbf{F}'^{(2)}, [8, 4, 4], \vec{0})$-basis as shown in Example 3.15 where we have $\mathbf{F}'^{(2)} = \mathbf{F}'$. Computing a $(\mathbf{F}'^{(2)}, [8, 4, 4], \vec{0})$-basis can be divided into two subproblems. The first subproblem is computing a $(\bar{\mathbf{F}}^{(1)}, 4, \vec{0})$-basis $\bar{\mathbf{P}}^{(1)}$, the Storjohann partial linearized problem in Example 3.1. The residual $\mathbf{F}'^{(2)}\bar{\mathbf{P}}^{(1)} =$

$$\begin{bmatrix} 0 & x^8 & x^6 + x^9 & x^4 + x^6 + x^9 & x^6 + x^8 + x^9 + x^{10} & x^5 + x^8 \\ 0 & 0 & x^5 & x^4 + x^6 & x^4 + x^6 & x^5 + x^6 \\ 0 & x^4 & x^5 & x^5 & x^4 + x^5 + x^6 & x^4 \end{bmatrix}$$

is then used as the input matrix for the second subproblem. The shift for the second subproblem $\vec{t}^{(1)} = [0, 1, 2, 3, 3, 3]$ is the list of column degrees of $\bar{\mathbf{P}}^{(1)}$ and so the second subproblem is to compute a $(\mathbf{F}'^{(2)}\bar{\mathbf{P}}^{(1)}, [8,4,4], [0,1,2,3,3,3])$-basis, which is

$$\bar{\mathbf{Q}}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x^2 & x & 1 \\ 0 & 0 & 0 & 0 & x & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & x \end{bmatrix}. \tag{3.7}$$

Then $\bar{\mathbf{P}}^{(1)}\bar{\mathbf{Q}}^{(2)}$ gives the $(\mathbf{F}'^{(2)}, [8,4,4], \vec{0})$-basis shown in Example 3.15.

We now show that the dimension of the second subproblem can be significantly reduced. First, the row dimension can be reduced by over a half. Let $\hat{\mathbf{P}}^{(i-1)} = \mathbf{E}^{(i)}\bar{\mathbf{P}}^{(i-1)}$.

Lemma 3.22. *A $(\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}^{(i-1)}, 2\delta^{(i)}, \vec{t}^{(i-1)})$-basis is a $(\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}, \vec{\omega}^{(i)}, \vec{t}^{(i-1)})$-basis.*

Proof. This follows because $\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}^{(i-1)}$ is a submatrix of $\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}$ after removing rows which already have the correct order $2\delta^{(i-1)}$. □

The column dimension of the second subproblem can be reduced by disregarding the $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})_{\delta^{(i-1)}-1}$-basis which has already been computed. More specifically, after sorting the columns of $\bar{\mathbf{P}}^{(i-1)}$ in an increasing order of their $\vec{s}^{(i-1)}$-degrees, let $[\bar{\mathbf{P}}_1^{(i-1)}, \bar{\mathbf{P}}_2^{(i-1)}] = \bar{\mathbf{P}}^{(i-1)}$ be such that $\deg_{\vec{s}^{(i-1)}} \bar{\mathbf{P}}_1^{(i-1)} \leq \delta^{(i-1)} - 1$ and $\deg_{\vec{s}^{(i-1)}} \bar{\mathbf{P}}_2^{(i-1)} \geq \delta^{(i-1)}$. Then $\hat{\mathbf{P}}_1^{(i-1)} = \mathbf{E}^{(i)}\bar{\mathbf{P}}_1^{(i-1)}$ is a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})_{\delta^{(i-1)}-1}$-basis by Lemma 3.19. In the second subproblem, the remaining basis elements of a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis can then be computed without $\bar{\mathbf{P}}_1^{(i-1)}$.

Let $\hat{\mathbf{P}}_2^{(i-1)} = \mathbf{E}^{(i)}\bar{\mathbf{P}}_2^{(i-1)}$, $\vec{b}^{(i-1)} = \deg_{\vec{s}^{(i-1)}} \bar{\mathbf{P}}_2^{(i-1)}$, $\bar{\mathbf{Q}}_2^{(i)}$ be a $(\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}_2^{(i-1)}, 2\delta^{(i)}, \vec{b}^{(i-1)})$-basis (or equivalently a $(\mathbf{F}'^{(i)}\bar{\mathbf{P}}_2^{(i-1)}, \vec{\omega}^{(i)}, \vec{b}^{(i-1)})$-basis), and $k^{(i-1)}$ be the column dimension of $\bar{\mathbf{P}}_1^{(i-1)}$. We then have the following result.

Lemma 3.23. *The matrix*

$$\bar{\mathbf{Q}}^{(i)} = \begin{bmatrix} \mathbf{I}_{k^{(i-1)}} & \\ & \bar{\mathbf{Q}}_2^{(i)} \end{bmatrix}$$

is a $(\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}^{(i-1)}, 2\delta^{(i)}, \vec{t}^{(i-1)})$*-basis (equivalently a* $(\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}, \vec{\omega}^{(i)}, \vec{t}^{(i-1)})$*-basis).*

Proof. First note that $\bar{\mathbf{Q}}^{(i)}$ has order $(\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}^{(i-1)}, 2\delta^{(i)})$ as

$$\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}^{(i-1)}\bar{\mathbf{Q}}^{(i)} = [\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}_1^{(i-1)}, \bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}_2^{(i-1)}\bar{\mathbf{Q}}_2^{(i)}] \equiv 0 \mod x^{2\delta^{(i)}}.$$

In addition, $\bar{\mathbf{Q}}^{(i)}$ has minimal $\vec{t}^{(i-1)}$ degrees as $\bar{\mathbf{Q}}_2^{(i)}$ is $\vec{b}$-minimal. Hence, by Lemma 2.21, $\bar{\mathbf{Q}}^{(i)}$ is a $(\bar{\mathbf{F}}^{(i)} \cdot \hat{\mathbf{P}}^{(i-1)}, 2\delta^{(i)}, \vec{t}^{(i-1)})$-basis. □

Lemma 3.23 immediately leads to the following.

Lemma 3.24. *Let* $\hat{\mathbf{S}} = [\hat{\mathbf{P}}_1^{(i-1)}, \hat{\mathbf{P}}_2^{(i-1)}\bar{\mathbf{Q}}_2^{(i)}]$*, and let* I *be the column rank profile of* $\mathrm{lcoeff}(x^{\vec{s}^{(i)}}\hat{\mathbf{S}})$*. Then* $\hat{\mathbf{S}}_I$ *is a* $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$*-basis.*

Proof. From Lemma 3.23, $\bar{\mathbf{Q}}^{(i)}$ is a $(\mathbf{F}'^{(i)}\bar{\mathbf{P}}^{(i-1)}, \vec{\omega}^{(i)}, \vec{t}^{(i-1)})$-basis and hence $\bar{\mathbf{P}}^{(i-1)}\bar{\mathbf{Q}}^{(i)}$ is a $(\mathbf{F}'^{(i)}, \vec{\omega}^{(i)}, \vec{s}^{(i-1)})$-basis. Since $[\hat{\mathbf{P}}_1^{(i-1)}, \hat{\mathbf{P}}_2^{(i-1)}\bar{\mathbf{Q}}_2^{(i)}] = \mathbf{E}^{(i)}\bar{\mathbf{P}}^{(i-1)}\bar{\mathbf{Q}}^{(i)}$, the result follows from Theorem 3.18. □

Example 3.25. Continuing with Example 3.1, Example 3.15, and Example 3.21, notice that in the computation of the second subproblem, instead of using $\mathbf{F}'^{(2)}$, $\bar{\mathbf{P}}^{(1)}$, $\bar{\mathbf{Q}}^{(2)}$, and $\bar{\mathbf{P}}^{(1)}\bar{\mathbf{Q}}^{(2)}$, the previous lemmas show that we can just use their submatrices, $\bar{\mathbf{F}}^{(2)}$ the top left 1×4 submatrix of $\mathbf{F}'^{(2)}$, $\hat{\mathbf{P}}_2^{(1)}$ the top right 4×4 submatrix of $\bar{\mathbf{P}}^{(1)}$, $\bar{\mathbf{Q}}_2^{(2)}$ the bottom right 4×4 submatrix of $\bar{\mathbf{Q}}^{(2)}$, and $\hat{\mathbf{P}}_2^{(1)}\bar{\mathbf{Q}}_2^{(2)}$ the top right 4×4 submatrix of $\bar{\mathbf{P}}^{(1)}\bar{\mathbf{Q}}^{(2)}$of lower dimensions.

Lemma 3.24 gives us a way of computing a $(\mathbf{F}, \sigma, \vec{s})$-basis. We can set i to $\log(n/m) - 1$ so that $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)}) = (\mathbf{F}, \sigma, \vec{s})$, and compute a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis.

By Lemma 3.24, this can be divided into two subproblems. The first produces $[\hat{\mathbf{P}}_1^{(i-1)}, \hat{\mathbf{P}}_2^{(i-1)}] = \hat{\mathbf{P}}^{(i-1)} = \mathbf{E}^{(i)}\bar{\mathbf{P}}^{(i-1)}$ from computing a $(\bar{\mathbf{F}}^{(i-1)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})$-basis $\bar{\mathbf{P}}^{(i-1)}$. The second subproblem then computes a $(\bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}_2^{(i-1)}, 2\delta^{(i)}, \vec{b}^{(i-1)})$-basis $\bar{\mathbf{Q}}_2^{(i)}$. Note the first subproblem of computing a $(\bar{\mathbf{F}}^{(i-1)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})$-basis can again be divided into two subproblems just as before. This can be repeated recursively until we reach the base case with degree parameter $\delta^{(1)} = 2d$. The total number of recursion levels is therefore $\log(n/m) - 1$.

Notice that the transformed matrix $\mathbf{F}'^{(i)}$ is not used explicitly in the computation, even though it is crucial for deriving our results.

3.4.2 The Iterative View

In this subsection we present our algorithm, which uses an iterative version of the computation discussed above. The iterative version is usually more efficient in practice, considering that the generic case has balanced output that can be computed with just one iteration, whereas the recursive method has to go through $\log(n/m) - 1$ levels of recursion.

Algorithm 3.1 uses a subroutine orderBasis, the algorithm from Giorgi et al. [2003], for computing order bases with balanced input. Specifically, $[\mathbf{Q}, \vec{a}] =$ orderBasis$(\mathbf{G}, \sigma, \vec{b})$ computes a $(\mathbf{G}, \sigma, \vec{b})$-basis and also returns its $\vec{b}$-column degrees $\vec{a}$. The other subroutine StorjohannTransform is the transformation described in Subsection 3.1.

Algorithm 3.1 proceeds as follows. In the first iteration, which is the base case of the recursive approach, we set the degree parameter $\delta^{(1)}$ to be twice the average degree d and apply Storjohann's transformation to produce a new input matrix $\bar{\mathbf{F}}^{(1)}$, which has $l^{(1)}$ block rows. Then a $(\bar{\mathbf{F}}^{(1)}, 2\delta^{(1)}, \vec{s}^{(1)})$-basis $\bar{\mathbf{P}}^{(1)}$ is computed. Note this is in fact the first subproblem of computing a $(\bar{\mathbf{F}}^{(2)}, 2\delta^{(2)}, \vec{s}^{(2)})$-basis, which is another Storjohann transformed problem and also the problem of the second iteration. At the second iteration, we work on a new Storjohann transformed problem with the degree doubled and the number of block rows $l^{(2)} = (l^{(1)} - 1)/2$ reduced by over a half. The column dimension is reduced by using the result from the previous iteration. More specifically, we know that the basis $\bar{\mathbf{P}}^{(1)}$ already provides a $(\bar{\mathbf{F}}^{(2)}, 2\delta^{(2)}, \vec{s}^{(2)})_{\delta^{(1)}-1}$-basis $\hat{\mathbf{P}}_1^{(1)}$, which can be disregarded in the remaining computation. The remaining work in the second iteration is to compute a $(\bar{\mathbf{F}}^{(2)}\hat{\mathbf{P}}_2^{(1)}, 2\delta^{(2)}, \vec{b}^{(1)})$-basis $\bar{\mathbf{Q}}^{(2)}$, where $\vec{b}^{(1)} = \deg_{\vec{s}^{(1)}} \bar{\mathbf{P}}_2^{(1)}$, and then to combine it with the result from the previous iteration to form a matrix $[\hat{\mathbf{P}}_1^{(1)}, \hat{\mathbf{P}}_2^{(1)}\bar{\mathbf{Q}}^{(2)}]$ in order to extract a $(\bar{\mathbf{F}}^{(2)}, 2\delta^{(2)}, \vec{s}^{(2)})$-basis $\bar{\mathbf{P}}^{(2)}$.

With a $(\bar{\mathbf{F}}^{(2)}, 2\delta^{(2)}, \vec{s}^{(2)})$-basis computed, we can repeat the same process to use it for computing a $(\bar{\mathbf{F}}^{(3)}, 2\delta^{(3)}, \vec{s}^{(3)})$-basis. Continue this, using the computed $(\bar{\mathbf{F}}^{(i-1)}, 2\delta^{(i-1)}, \vec{s}^{(i-1)})$-basis to compute a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis, until all n elements of a $(\mathbf{F}, \sigma, \vec{s})$-basis have been determined.

3.5 Computational Complexity

In this section, we analyze the computational complexity of Algorithm 3.1.

Lemma 3.26. *Algorithm 3.1 computes a* $(\mathbf{F}, \sigma, \vec{s})$*-basis in no more than* $\log(n/m) - 1$ *iterations.*

Proof. Each iteration i computes a $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})$-basis. At iteration $i^* = \log(n/m) - 1$, the degree parameter is $\sigma/2$ and $(\bar{\mathbf{F}}^{(i^*)}, 2\delta^{(i^*)}, \vec{s}^{(i^*)}) = (\mathbf{F}, \sigma, \vec{s})$. □

Lemma 3.27. *If the shift* $\vec{s} = [0, \ldots, 0]$*, then a* $(\mathbf{F}, \sigma, \vec{s})_{\delta^{(i)}-1}$*-basis (or equivalently a* $(\bar{\mathbf{F}}^{(i)}, 2\delta^{(i)}, \vec{s}^{(i)})_{\delta^{(i)}-1}$*-basis) computed at iteration* i *has at least* $n - n/2^i$ *elements, and hence at most* $n/2^i$ *elements remain to be computed. If the shift* $\vec{s}$ *is balanced, that is,* $\max \vec{s} \in O(a)$ *assuming* $\min \vec{s} = 0$*, then the number* $n^{(i)}$ *of remaining basis elements at iteration* i *is* $O(n/2^i)$*.*

Proof. The uniform case follows from the idea of Storjohann and Villard [2005] on null space basis computation discussed in Subsection 3.2. For the balanced case, the average column degree is bounded by $ca = cm\sigma/n$ for some constant c. The first iteration λ such that $\delta^{(\lambda)}$ reaches ca is therefore a constant. That is, $\delta^{(\lambda)} = 2^\lambda a \geq ca > \delta^{(\lambda-1)}$ and hence $\lambda = \lceil \log c \rceil$. By the same argument as in the uniform case, the number of remaining basis elements $n^{(i)} \leq n/2^{i-\lambda} = 2^\lambda (n/2^i) \in O(n/2^i)$ at iteration $i \geq \lambda$. For iterations $i < \lambda$, certainly $n^{(i)} \leq n < 2^\lambda(n/2^i) \in O(n/2^i)$. □

Theorem 3.28. *If the shift* $\vec{s}$ *is balanced with* $\min(\vec{s}) = 0$*, then Algorithm 3.1 computes a* $(\mathbf{F}, \sigma, \vec{s})$*-basis with a cost of* $O(n^\omega \operatorname{M}(a) \log \sigma) \subset O^\sim(n^\omega a)$ *field operations.*

Proof. The computational cost depends on the degree, the row dimension, and the column dimension of the problem at each iteration. The degree parameter $\delta^{(i)}$ is $2^i a$ at iteration i. The number of block rows $l^{(i)}$ is $\sigma/\delta^{(i)} - 1$, which is less than $\sigma/(2^i a) = n/(2^i m)$ at iteration i. The row dimension is therefore less than $n/2^i$ at iteration i.

The column dimension of interest at iteration i is the column dimension of $\hat{\mathbf{P}}_2^{(i-1)}$ (equivalently the column dimension of $\bar{\mathbf{P}}_2^{(i-1)}$), which is the sum of two components, $n^{(i-1)} + (l^{(i-1)} - 1)m$. The first component $n^{(i-1)} \in O(n/2^i)$ by Lemma 3.27. The second component $(l^{(i-1)} - 1)m < n/2^{i-1} - m < n/2^{i-1}$ comes from the size of the identity matrix added in Storjohann's transformation. Therefore, the overall column dimension of the problem at iteration i is $O(n/2^i)$.

At each iteration, the four most expensive operations are the multiplications at line 15 and line 19, the order basis computation at line 17, and extracting the basis at line 20.

The matrices $\bar{\mathbf{F}}^{(i)}$ and $\hat{\mathbf{P}}_2^{(i-1)}$ have degree $O(2^i a)$ and dimensions $O(n/2^i) \times O(n)$ and $O(n) \times O(n/2^i)$. The multiplication cost is therefore $2^i \operatorname{MM}(n/2^i, 2^i a)$ field operations, which is bounded by

$$\begin{aligned}
2^i \operatorname{MM}(n/2^i, 2^i a) &\in O\left(2^i \left(n/2^i\right)^\omega \operatorname{M}(2^i a)\right) \\
&\subset O\left(n^\omega \left(2^i\right)^{1-\omega} \operatorname{M}\left(2^i\right) \operatorname{M}(a)\right) \qquad (3.8)\\
&\subset O\left(n^\omega \left(2^i\right)^{1-\omega} \left(2^i\right)^{\omega-1} \operatorname{M}(a)\right) \qquad (3.9)\\
&\subset O\left(n^\omega \operatorname{M}(a)\right).
\end{aligned}$$

Equation (3.8) follows from $\operatorname{M}(st) \in O\left(\operatorname{M}(s)\operatorname{M}(t)\right)$. Equation (3.9) follows from $\operatorname{M}(t) \in O(t^{\omega-1})$.

The matrices $\hat{\mathbf{P}}_2^{(i-1)}$ and $\bar{\mathbf{Q}}^{(i)}$ of the second multiplication have the same degree $O(2^i a)$ and dimensions $O(n) \times O(n/2^i)$ and $O(n/2^i) \times O(n/2^i)$ and can also be multiplied with a cost of $O\left(n^\omega \operatorname{M}(a)\right)$ field operations. The total cost of the multiplications over $O(\log(n/m))$ iterations is therefore $O\left(n^\omega \operatorname{M}(a)\log(n/m)\right)$.

The input matrix $\mathbf{G}^{(i)} = \bar{\mathbf{F}}^{(i)}\hat{\mathbf{P}}_2^{(i-1)}$ of the order basis computation problem at iteration i has dimension $O(n/2^i) \times O(n/2^i)$ and the order of the problem is $2\delta^{(i)} \in O(2^i a)$. Thus, the cost of the order basis computation at iteration i is $O\left(\left(n/2^i\right)^\omega \operatorname{M}\left(2^i a\right)\log\left(2^i a\right)\right)$. The total cost over $O(\log(n/m))$ iterations is bounded by

$$\begin{aligned}
&O\left(\sum_{i=1}^{\infty}\left(\left(n/2^i\right)^\omega \operatorname{M}\left(2^i a\right)\log\left(2^i a\right)\right)\right)\\
\subset &O\left(\sum_{i=1}^{\infty}\left(\left(n/2^i\right)^\omega \operatorname{M}\left(2^i\right)\log\left(2^i\right)\operatorname{M}(a)\log(a)\right)\right)\\
\subset &O\left(\sum_{i=1}^{\infty}\left(n^\omega \left(2^i\right)^{-\omega}\left(2^i\right)^{\omega-1}\operatorname{M}(a)\log(a)\right)\right)\\
\subset &O\left(n^\omega \operatorname{M}(a)\log(a)\sum_{i=1}^{\infty}\left(2^{-i}\right)\right)\\
\subset &O\left(n^\omega \operatorname{M}(a)\log(a)\right).
\end{aligned}$$

Finally, extracting an order basis by LSP factorization costs $O(n^{\omega})$, which is dominated by the other costs. Combining the above gives

$$O\left(n^{\omega}\,\mathrm{M}(a)\log(n/m) + n^{\omega}\,\mathrm{M}(a)\log a\right) = O\left(n^{\omega}\,\mathrm{M}(a)\log\sigma\right)$$

as the total cost of the algorithm. $\square$

3.6 More Refined Cost and the Case $m\sigma \in o(n)$

Theorem 3.28 states the cost of computing a $(\mathbf{F},\sigma,\vec{s})$-basis as $O^{\sim}(n^{\omega}a)$, where $a = m\sigma/n$. In this cost, a is assumed to tend to infinity, which means $m\sigma > n$. This allows us to transform the original problem with dimension $m \times n$ and degree σ to one with dimension $\Theta(n)\times\Theta(n)$ and degree $\Theta(a) = \Theta(m\sigma/n)$, allowing order basis computation to be efficient with a final cost of $O^{\sim}(n^{\omega}a)$. However, if we attempt to state the cost as $O^{\sim}(n^{\omega-1}m\sigma)$, the case of $m\sigma \in o(n)$ becomes problematic and requires special attention. In this case, the average degree $a = m\sigma/n \in o(1)$ but 1 is the lowest possible degree and $m\sigma$ is the maximum possible row dimension of our transformed problems. In other words, we cannot obtain a nearly square transformed problem for our algorithms to behave efficiently, which means our algorithms still require $O^{\sim}(n^{\omega})$ field operations. We now look how this cost can be improved to $O^{\sim}(n^{\omega-1}m\sigma)$ in the case of $m\sigma \in o(n)$.

3.6.1 Balanced Case

First note that in this case, using Definition 3.3, a balanced shift $\vec{s}$ is also uniform, since $\max(\vec{s}) - \min(\vec{s}) \in O(m\sigma/n) \subseteq o(1)$, which makes $\max(\vec{s}) - \min(\vec{s}) = 0$. So let us just consider the uniform shift case.

We first compute all degree 0 basis elements, which then helps to eliminate the columns of the input that are never going to be needed as pivots. The remaining columns can then be used as the input to compute the remaining basis elements efficiently. The degree 0 elements of a $(\mathbf{F}, \sigma)$-basis correspond to a nullspace basis of a linearized matrix

$$\bar{F} = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{\sigma-1} \end{bmatrix} \in \mathbb{K}^{(m\sigma)\times n}$$

of $\mathbf{F} = F_0 + F_1 x + F_2 x^2 + \cdots + F_{\sigma-1} x^{\sigma-1}$.

Lemma 3.29. *The elements of a nullspace basis of $\bar{F}$ over $\mathbb{K}$ are also the degree 0 elements of a $(\mathbf{F}, \sigma)$-basis.*

Proof. The columns of $\bar{F}$ and the columns of $\mathbf{F}$ are equivalent representations of the same elements of the same $\mathbb{K}$-module, which is also a vector space over $\mathbb{K}$. $\square$

To compute these basis elements, we can use the Gauss Jordan transform algorithm from Storjohann [2000] on $\bar{F}$ with a cost of $O\left(nm\sigma\bar{r}^{\omega-2}\right)$, where $\bar{r} \le m\sigma$ is the rank of $\bar{F}$. The algorithm finds a permutation matrix P and a unimodular matrix U in $\mathbb{K}^{n\times n}$ such that $\bar{F}PU$ is in the reduced column echelon form of $\bar{F}$. Note that P permutes the columns of $\bar{F}$ so that the first $\bar{r}$ columns of $\bar{F}$ are linearly independent. Let $[U_1, U_0] := U$ with U_0 correspond to the zero columns of $\bar{F}PU$. Then the matrix consists of the bottom $n - \bar{r}$ rows of U_0 is the identity matrix, and only the first $\bar{r}$ rows of U_1 are nonzero. Because of this simpler structure after permutation, let us compute a $(\mathbf{F}P, \sigma)$-basis $\mathbf{P}$ instead, which also gives us a $(\mathbf{F}, \sigma)$-basis $P\mathbf{P}$. Notice that U_0 consists of all the degree 0 elements of a $(\mathbf{F}P, \sigma)$-basis. We can then use $\mathbf{F}PU_1$ as the input matrix to compute the remaining basis elements. But to further simplify our future computation, let us replace U_1 with $V = [I, 0]^T$ of the same dimension, where the identity matrix I replaces the first nonzero $\bar{r}$ rows in U_1. In essence, PV picks $\bar{r}$ columns from $\mathbf{F}$ for computing the remaining basis elements. Since U_0 has at least $n - m\sigma$ columns, there are at most $m\sigma$ columns in U_1, and hence at most $m\sigma$ columns in V and in $\mathbf{F}PV$.

Lemma 3.30. *If we compute a* $(\mathbf{F}PV, \sigma)$*-basis* $\mathbf{Q}$*, then* $[V\mathbf{Q}, U_0]$ *is a* $(\mathbf{F}P, \sigma)$*-basis.*

Proof. Note that the matrix $[V, U_0]$, which has the structure

$$\begin{bmatrix} I & * \\ 0 & I \end{bmatrix}$$

with $*$ representing the first r rows of U_0, is a $(\mathbf{F}P, 0)$-basis since it is unimodular and column reduced. From 3.20, we can use the residual $\mathbf{F}P[V, U_0] = [\mathbf{F}PV, 0]$ to compute a $([\mathbf{F}PV, 0], \sigma)$-basis $\bar{\mathbf{Q}}$, then $[V, U_0]\bar{\mathbf{Q}}$ is a $(\mathbf{F}P, \sigma)$-basis. Also note that if $\mathbf{Q}$ is a $(\mathbf{F}PV, \sigma)$-basis, then

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \\ & I \end{bmatrix}$$

is a $([\mathbf{F}PV, 0], \sigma)$-basis, and $[V, U_0]\bar{\mathbf{Q}} = [V\mathbf{Q}, U_0]$ is a $(\mathbf{F}P, \sigma)$-basis. □

Our new problem of computing a $(\mathbf{F}PV, \sigma)$-basis now satisfies the condition of having column dimension bounded by $m\sigma$. We can therefore compute a $(\mathbf{F}PV, \sigma)$-basis using Algorithm 3.1 with a cost of $O^{\sim}((m\sigma)^{\omega}) \subset O^{\sim}(n^{\omega-1}m\sigma)$.

The last thing to check is making sure that the multiplications for computing the residual $\mathbf{F}PV$, and for combining the results $V\mathbf{Q}$, and for obtaining the final result $P[V\mathbf{Q}, U_0]$ can all be done efficiently, which is not difficult since P is a permutation matrix, and V consists of an identity matrix and zeros. Therefore, the $(\mathbf{F}, \sigma)$-basis $P[V\mathbf{Q}, U_0]$ can be computed with a cost of $O^{\sim}(n^{\omega-1}m\sigma)$. This allows us to refine the cost $O^{\sim}(n^{\omega}d)$ to $O^{\sim}(n^{\omega-1}m\sigma)$.

Theorem 3.31. *A $(\mathbf{F}, \sigma, \vec{s})$-basis can be computed with a cost of*

$$O\left(n^{\omega}\, \mathrm{M}(m\sigma/n) \log \sigma\right) \subset O^{\sim}\left(n^{\omega-1} m\sigma\right)$$

field operations.

Algorithm 3.1 fastOrderBasis $(\mathbf{F}, \sigma, \vec{s})$

Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$, $\sigma \in \mathbb{Z}_{\geq 0}$, $\vec{s} \in \mathbb{Z}^n$ satisfying $n \geq m$, n/m and σ are powers of 2, $m\sigma \in \Omega(n)$ and $\min(\vec{s}) = 0$
Output: a $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P} \in \mathbb{K}[x]^{n\times n}$ and $\deg_{\vec{s}} \mathbf{P}$

```
1: if $2m \geq n$ then return orderBasis$(\mathbf{F}, \sigma, \vec{s})$;
2: $i := 1$; $d := m\sigma/n$; $\delta^{(1)} := 2d$;
3: $\bar{\mathbf{F}}^{(1)} :=$ StorjohannTransform$(\mathbf{F}, \delta^{(1)})$;
4: $l^{(1)} :=$ rowDimension$(\bar{\mathbf{F}}^{(1)})/m$;
5: $\vec{b}^{(0)} := [\vec{s}, 0, \ldots, 0]$;        // $m(l_1 - 1)$ 0's
6: $[\bar{\mathbf{P}}^{(1)}, \vec{a}^{(1)}] :=$ orderBasis$(\bar{\mathbf{F}}^{(1)}, 2\delta^{(1)}, \vec{b}^{(0)})$;
7: Sort the columns of $\bar{\mathbf{P}}^{(i)}$ and $\vec{a}^{(i)}$ by the shifted column degrees $\vec{a}^{(i)} = \deg_{\vec{b}} \bar{\mathbf{P}}^{(i)}$
   in increasing order;
8: $\vec{t}^{(i)} := \vec{a}^{(i)}$;
9: $k^{(i)} :=$ number of entries of $\vec{a}^{(i)}$ less than $\delta^{(i)}$;
10: $[\bar{\mathbf{P}}_1^{(i)}, \bar{\mathbf{P}}_2^{(i)}] := \bar{\mathbf{P}}^{(i)}$ with $\bar{\mathbf{P}}_1^{(i)} \in \mathbb{K}[x]^{n\times k^{(i)}}$;
11: while columnDimension$(\bar{\mathbf{P}}_1^{(i)}) < n$ do
12:    $i := i + 1$; $\delta^{(i)} := 2\delta^{(i-1)}$; $l^{(i)} := (l^{(i-1)} - 1)/2$;
13:    $\bar{\mathbf{F}}^{(i)} :=$ StorjohannTransform$(\mathbf{F}, \delta^{(i)})$;
14:    $\hat{\mathbf{P}}_2^{(i-1)} := \mathbf{E}^{(i)} \bar{\mathbf{P}}_2^{(i-1)}$;
15:    $\mathbf{G}^{(i)} := \bar{\mathbf{F}}^{(i)} \hat{\mathbf{P}}_2^{(i-1)}$;
16:    $\vec{b}^{(i-1)} := \vec{t}^{(i-1)}[k^{(i-1)} + 1 \ldots n + m(l^{(i-1)} - 1)]$;
        // $w := v[k..l]$means that w receives a slice of v
        // whose indices range from $k$to $l$
17:    $[\mathbf{Q}^{(i)}, \vec{a}^{(i)}] :=$ orderBasis$(\mathbf{G}^{(i)}, 2\delta^{(i)}, \vec{b}^{(i-1)})$;
18:    Sort the columns of $\mathbf{Q}^{(i)}$ and $\vec{a}^{(i)}$ by $\vec{a}^{(i)} = \deg_{\vec{b}^{(i-1)}} \mathbf{Q}^{(i)}$ in increasing order;

19:    $\check{\mathbf{P}}^{(i)} := \hat{\mathbf{P}}_2^{(i-1)} \mathbf{Q}^{(i)}$;
20:    $J :=$ the column rank profile of lcoeff$(x^{[\vec{s},0,\ldots,0]}[\mathbf{E}^{(i)} \bar{\mathbf{P}}_1^{(i-1)}, \check{\mathbf{P}}^{(i)}])$;
21:    $\bar{\mathbf{P}}^{(i)} := [\mathbf{E}^{(i)} \bar{\mathbf{P}}_1^{(i-1)}, \check{\mathbf{P}}^{(i)}]_J$,
22:    $\vec{t}^{(i)} := \deg_{[\vec{s},0,\ldots,0]} \bar{\mathbf{P}}^{(i)}$;
23:    $k^{(i)} :=$ number of entries of $\vec{t}^{(i)}$ less than $\delta^{(i)}$;
24:    $[\bar{\mathbf{P}}_1^{(i)}, \bar{\mathbf{P}}_2^{(i)}] := \bar{\mathbf{P}}^{(i)}$ with $\bar{\mathbf{P}}_1^{(i)} \in K[x]^{n\times k^{(i)}}$ ;
25: end while
26: return the top $n$ rows of $\bar{\mathbf{P}}_1^{(i)}$, $\vec{t}^{(i)}[1..n]$;
```

Chapter 4

Order Basis with Unbalanced Shifts

Theorem 3.31 shows that Algorithm 3.1 can efficiently compute a $(\mathbf{F}, \sigma, \vec{s})$-basis when the shift $\vec{s}$ is balanced. When $\vec{s}$ is unbalanced (something important for example in normal form computation [Beckermann et al., 1999, 2006b]), then Algorithm 3.1 still returns a correct answer but may be less efficient. The possible inefficiency results because there may not be enough partial results from the intermediate subproblems to sufficiently reduce the column dimension of the subsequent subproblem. This is clear from the fact that the column degrees of the output can be much larger and no longer sum up to $O\left(m\sigma\right)$ as in the balanced shift case. The shifted $\vec{s}$-column degrees, however, still behave well. In particular, the total $\vec{s}$-degree increase is still bounded by $m\sigma$ as stated in Lemma 3.2, while the shifted degree of any column can also increase by up to σ. Recall that Lemma 3.2 states that for any shift $\vec{s}$, there exists a $(\mathbf{F}, \sigma, \vec{s})$-basis still having a total size bounded by $nm\sigma$ which gives hope for efficient computation. In the following, we look at two special cases of unbalanced shifts. In the first case where the input shift $\vec{s}$ satisfies $\sum_{i=1}^{n}(\vec{s}_i - \min(\vec{s})) \in O(m\sigma)$, the sum of the column degrees of a $(\mathbf{F}, \sigma, \vec{s})$-basis is still in $O(m\sigma)$, which allows us to use Algorithm 3.1 to compute a $(\mathbf{F}, \sigma, \vec{s})$-basis efficiently as in the balanced case. The second case where $\vec{s}$ satisfies

$\sum_{i=1}^{n}(\max(\vec{s}) - \vec{s}_i) \in O(m\sigma)$ is more complicated and is the main focus of this section.

4.1 First unbalanced case

We first consider the case where the input shift $\vec{s}$ satisfies

$$\sum_{i=1}^{n}(\vec{s}_i - \min(\vec{s})) \in O(m\sigma).$$

As before, we may use the equivalent condition

$$\vec{s} \geq 0 \text{ and } \sum \vec{s} \in O(m\sigma), \tag{4.1}$$

which can always be obtained from the previous condition by using $\vec{s} - \min \vec{s}$ as the new shift. Note that translating every entry of the shift by the same constant does not change the problem. In this case, Algorithm 3.1 works efficiently as before.

Lemma 4.1. *If the shift $\vec{s}$ satisfies condition (4.1), then a $(\mathbf{F}, \sigma, \vec{s})$-basis can be computed with $O\left(n^{\omega}\,\mathrm{M}(a) \log \sigma\right) \subset O^{\sim}\left(n^{\omega} a\right)$ field operations.*

From Lemma 3.2, we know that the sum of the $\vec{s}$-column degrees of any $(\mathbf{F}, \sigma, \vec{s})$-basis is $\vec{t} = \sum \vec{s} + m\sigma \in O(m\sigma)$, and since the entries of $\vec{s}$ are non-negative, the sum of the column degrees is less than $\sum \vec{t}$. So the sum of the column degrees of any $(\mathbf{F}, \sigma, \vec{s})$-basis is also in $O(m\sigma)$. Now the same analysis from Section 3.5 applies.

4.2 Second unbalanced case

We now look at another important case of unbalanced shift – when the input shift $\vec{s}$ satisfies the condition:

$$\sum_{i=1}^{n}(\max(\vec{s}) - \vec{s}_i) \leq m\sigma.$$

For simplicity, we use the equivalent condition

$$\vec{s} \leq 0 \text{ and } -\sum \vec{s} \leq m\sigma, \tag{4.2}$$

which can always be obtained from the previous condition by using $\vec{s} - \max \vec{s}$ as the new shift.

In the balanced shift case, a central problem is to find a way to handle unbalanced column degrees of the output order basis. In this section, the unbalanced shift makes row degrees of the output also unbalanced, which is a major problem that needs to be resolved. Here we note a second transformation by Storjohann [2006] which converts the input in such a way that each high degree row of the output becomes multiple rows of lower degrees. We refer to this as Storjohann's second transformation to distinguish it from that described in Subsection 3.1. The transformed problem can then be computed efficiently using Algorithm 3.1. After the computation, rows can then be combined appropriately to form a basis of the original problem. The method is computationally efficient.

Unfortunately, the bases computed this way are not minimal and hence do not in general produce our reduced order bases. In the following, we describe a transformation that incorporates Storjohann's second transformation and guarantees the minimality of some columns of the output, hence providing a partial order basis. We can then work on the remaining columns iteratively as done in the balanced shift case to compute a full order basis.

Condition (4.2) essentially allows us to locate the potential high degree rows that need to be balanced. In more general cases, we may not know in advance which are the high degree rows that need to be balanced, so our approach given in this section does not work directly. This suggests that one possible future direction to pursue is to find an effective way to estimate the row degree of the result pivot entries. Such an estimate may allow us to apply the method given in this section efficiently for general unbalanced shifts.

4.2.1 Transform to Balanced Shifts

We now describe the transformation for balancing the high degree rows of the resulting basis. Consider the problem of computing a $(\mathbf{F}, \sigma, \vec{s})$-basis, where the input shift $\vec{s}$ satisfies the conditions (4.2). Let $\alpha, \beta \in \mathbb{Z}_{>0}$ be two parameters. For each shift entry s_i in $\vec{s}$ with $-s_i > \alpha + \beta$, let

$$r_i = \operatorname{rem}\left(-s_i - \alpha - 1, \beta\right) + 1$$

be the remainder when $-s_i - \alpha$ is divided by β, and where $r_i = \beta$ in the case where the remainder is 0, and set

$$q_i = \begin{cases} 1 & \text{if } -s_i \leq \alpha + \beta \\ 1 + \left(-s_i - \alpha - r_i\right)/\beta & \text{otherwise} \end{cases}$$

Then, for each $q_i > 1$, we expand the corresponding ith column $\mathbf{f}_i$ of $\mathbf{F}$ and shift s_i to

$$\tilde{\mathbf{F}}^{(i)} \quad = \quad \left[\, \mathbf{f}_i,\ x^{r_i}\mathbf{f}_i,\ x^{r_i+\beta}\mathbf{f}_i,\ \ldots\ ,\ x^{r_i+(q_i-2)\beta}\mathbf{f}_i \right], \quad \tilde{s}_i = [-\alpha-\beta,\ \ldots\ , -\alpha-\beta]$$

with q_i entries in each case. When $q_i = 1$, the corresponding shift entry and input column remain the same, that is, $\tilde{s}_i = s_i$, and $\tilde{\mathbf{F}}^{(i)} = \mathbf{f}_i$. Then for the transformed problem, the new shift becomes $\bar{s} = [\tilde{s}_1, \ldots, \tilde{s}_n] \in \mathbb{Z}_{\leq 0}^{\bar{n}}$, and the new input matrix becomes $\bar{\mathbf{F}} = [\tilde{\mathbf{F}}^{(1)}, \ldots, \tilde{\mathbf{F}}^{(n)}] \in \mathbb{K}\left[x\right]^{m \times \bar{n}}$, with the new column dimension $\bar{n}$ satisfies $\bar{n} = \sum_{i=1}^{n} q_i$. Note that every entry of the new shift $\bar{s}$ is an integer from $-\alpha - \beta$ to

0. Let

$$\mathbf{E} = \left[\begin{array}{c|c|c} 1 \quad x^{r_1} \quad x^{r_1+\beta} \quad \cdots \quad x^{r_1+(q_1-2)\beta} & & \\ \hline & \ddots & \\ \hline & & 1 \quad x^{r_n} \quad x^{r_n+\beta} \quad \cdots \quad x^{r_n+(q_n-2)\beta} \end{array}\right]_{n\times\bar{n}} .$$

Then $\bar{\mathbf{F}} = \mathbf{F}\mathbf{E}$. Storjohann's second transformation is determined by setting $\alpha = -1$, a value not allowed in our transformation (we show later in Theorem 4.11 that this value is not useful in our case). One can verify that the new dimension

$$\bar{n} = \sum_{i=1}^{n} q_i \leq n + \sum_{i=1}^{n} -s_i/\beta \leq m\sigma/\beta + n.$$

Thus by setting $\beta \in \Theta\left(a\right)$, where $a = m\sigma/n$, we can make $\bar{n} \in \Theta\left(n\right)$. Furthermore, by also setting $\alpha \in \Theta\left(a\right)$, we have a balanced shift problem since

$$\max \bar{s} - \min \bar{s} \leq -\min \bar{s} \leq \alpha + \beta \in \Theta(a).$$

Hence Algorithm 3.1 can compute a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis with cost $O^{\sim}\left(n^{\omega} a\right)$ in this case.

With a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis $\bar{\mathbf{P}} \in \mathbb{K}\left[x\right]^{\bar{n}\times\bar{n}}$ computed, let us now consider $\mathbf{E}\bar{\mathbf{P}} \in \mathbb{K}\left[x\right]^{n\times\bar{n}}$. While it is easy to see that $\mathbf{E}\bar{\mathbf{P}}$ has order $(\mathbf{F}, \sigma)$ since $\mathbf{F}\mathbf{E}\bar{\mathbf{P}} = \bar{\mathbf{F}}\bar{\mathbf{P}} \equiv 0 \mod x^{\sigma}$, in general it is not a minimal basis (in fact, $\mathbf{E}\bar{\mathbf{P}}$ is not even square). However, our transformation does guarantee that the highest degree columns of $\mathbf{E}\bar{\mathbf{P}}$ having $\vec{s}$-degrees exceed $-\alpha$ are minimal. That is, the columns of $\mathbf{E}\bar{\mathbf{P}}$ whose $\vec{s}$-degrees exceed $-\alpha$ are exactly the columns of a $(\mathbf{F}, \sigma, \vec{s})$-basis whose $\vec{s}$-degrees exceed $-\alpha$. We have therefore correctly computed a partial $(\mathbf{F}, \sigma, \vec{s})$-basis.

Example 4.2. Let us use the same input as in Example 3.1, but with shift $\vec{s} = [0, -3, -5, -6]$, and parameters $\alpha = \beta = 1$. Then we get the transformed input

$\bar{\mathbf{F}} =$

$$[x + x^2 + x^3 + x^4 + x^5 + x^6,\ 1 + x + x^5 + x^6 + x^7,\ x + x^2 + x^6 + x^7 + x^8,$$
$$1 + x^2 + x^4 + x^5 + x^6 + x^7,\ x + x^3 + x^5 + x^6 + x^7 + x^8,\ x^2 + x^4 + x^6 + x^7 + x^8 + x^9,$$
$$x^3 + x^5 + x^7 + x^8 + x^9 + x^{10},\ 1 + x + x^3 + x^7,\ x + x^2 + x^4 + x^8,$$
$$x^2 + x^3 + x^5 + x^9,\ x^3 + x^4 + x^6 + x^{10},\ x^4 + x^5 + x^7 + x^{11}]$$

having 12 components, and $\bar{s} = [0, -2, -2, -2, -2, -2, -2, -2, -2, -2, -2, -2]$. In this case $r_1 = r_2 = r_3 = r_4 = 1$, $q_1 = 1$, $q_2 = 2$, $q_3 = 4$, $q_4 = 5$ and the transformation matrix is

$$\mathbf{E} = \left[\begin{array}{c|cc|cccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & x & x^2 & x^3 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x^2 & x^3 & x^4 \end{array}\right].$$

Using the earlier algorithm for balanced shift, we compute a $(\bar{\mathbf{F}}, 8, \bar{s})$-basis

$$\bar{\mathbf{P}} = \left[\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
x & 1 & 0 & 0 & 1 & 0 & x & 0 & 0 & 0 & x & 0 \\
0 & 0 & 1 & 0 & 0 & x & 1+\ x & x & x & x & 1 & 0 \\
\hline
x & 1 & 0 & 1 & 1+x & 1 & x & 0 & 0 & 0 & 0 & 1 \\
x & 0 & 1 & 1 & 1+x & 1+x & 1 & x & x & 0 & 0 & 0 \\
x & 0 & 0 & 1 & 1+x & 1+x & 1 & x & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 1 & x & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & x & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

with $\bar{s}$-degrees $[-1, -2, -2, -2, -1, -1, -1, -1, -1, -1, -1, 0]$. Only the last column has $\bar{s}$-degree exceeding $-\alpha = -1$ and so is the only column guaranteed to give a correct $(\mathbf{F}, 8, \vec{s})$-basis element. Comparing $\mathbf{E}\bar{\mathbf{P}} =$

$$\left[\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
x & 1 & x & 0 & 1 & x^2 & x^2 & x^2 & x^2 & x^2 & 0 & 0 \\
x+x^2+x^3+x^4 & 1 & x & 1+x+x^2+x^3 & 1 & 1+x+x^3 & x^2 & x^2 & x^2 & x^2 & 0 & 1 \\
0 & x & x^2 & 1+x^3+x^4 & x & 1+x^4 & x^3 & x^3 & x^3 & x^3 & 0 & 1
\end{array}\right]$$

to a $(\mathbf{F}, 8, \vec{s})$-basis

$$\mathbf{P} = \begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
1 & x^2+x^3+x^4 & 1+x+x^2+x^3 & 1 \\
x & x^2 & 1+x^3+x^4 & 1
\end{bmatrix}$$

with $\vec{s}$-degrees $[-3,-1,-2,\ 0\ ]$, we see that the last column of $\mathbf{E}\bar{\mathbf{P}}$ is a element of a $(\mathbf{F},8,\vec{s})$-basis.

If we set $\alpha=2,\beta=1$, then the new transformed problem gives

$$\bar{\mathbf{P}}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & x & 1+x & x & x & x & 0 \\ 1 & x^2 & 1 & x & 1 & x & x & 0 & 1 \\ 0 & x^2 & 1 & x & 1 & x & 0 & 1 & 0 \\ 0 & x^2 & 1+x & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & x^2 & 1 & 0 & x & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x & 1+x & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & x & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with $\bar{s}$-degrees $[-3,-1,-2,-2,-2,-2,-2,-2,\ 0\ ]$. In this case the second column also has $\bar{s}$-degree exceeding $-\alpha=-2$, and so it is guaranteed to produce another element of a $(\mathbf{F},8,\vec{s})$-basis. Computing

$$\mathbf{E}\bar{\mathbf{P}}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & x & 1+x & x & x & x & 0 \\ 1 & x^2+x^3+x^4 & 1+x+x^2+x^3 & x & 1+x & x & x & x & 1 \\ x & x^2 & 1+x^3+x^4 & x^2 & x+x^2 & x^2 & x^2 & x^2 & 1 \end{bmatrix},$$

we notice the second column is indeed an element of a $(\mathbf{F},8,\vec{s})$-basis.

4.2.2 Correspondence Between the Original Problem and the Transformed Problem

We now work towards establishing the correspondence between the high degree columns of a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis whose $\bar{s}$-degrees exceed $-\alpha$ and those of a $(\mathbf{F}, \sigma, \vec{s})$-basis whose $\vec{s}$-degrees exceed $-\alpha$. A useful link is provided by the following matrix.

Set

$$\mathbf{A}_i = \begin{bmatrix} x^{r_i} & & & \\ -1 & x^{\beta} & & \\ & -1 & \ddots & \\ & & \ddots & x^{\beta} \\ & & & -1 \end{bmatrix}_{q_i \times (q_i - 1)} \qquad \text{and} \qquad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n \end{bmatrix}_{\bar{n} \times (\bar{n} - n)} .$$

If $q_i = 1$, $\mathbf{A}_i$ has dimension 1×0, which just adds a zero row and no column in $\mathbf{A}$.

We now show that for any $\bar{\mathbf{w}} \in \left\langle \left(\bar{\mathbf{F}}, \sigma, \bar{s}\right) \right\rangle$, $\bar{\mathbf{w}}$ can be transformed by $\mathbf{A}$ to one of the two forms that correspond to the original problem and transformed problem. This is made more precise in the following lemma. We then use unimodular equivalence of these two forms to show the equivalence between the high degree part of the result from the transformed problem and that of the original problem.

Lemma 4.3. *Let*

$$\bar{\mathbf{w}} = \begin{bmatrix} \bar{\mathbf{w}}_1 \\ \vdots \\ \bar{\mathbf{w}}_n \end{bmatrix} \in \langle (\bar{\mathbf{F}}, \sigma, \bar{s}) \rangle \text{ with } \bar{\mathbf{w}}_i = \begin{bmatrix} \bar{w}_{i,0} \\ \vdots \\ \bar{w}_{i,q_i - 1} \end{bmatrix}_{q_i \times 1} .$$

Then there exists a vector $\mathbf{u} \in \mathbb{K}[x]^{(\bar{n}-n) \times 1}$ *such that* $\bar{\mathbf{w}} + \mathbf{A}\mathbf{u}$ *has one of the following two forms.*

(a) The first form is

$$\mathbf{w}^{[1]} = \begin{bmatrix} \mathbf{w}_1^{[1]} \\ \vdots \\ \mathbf{w}_n^{[1]} \end{bmatrix} \text{ with } \mathbf{w}_i^{[1]} = \begin{bmatrix} w_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{q_i \times 1},$$

where $w_i = \bar{w}_{i,0} + \bar{w}_{i,1}x^{r_i} + \bar{w}_{i,2}x^{r_i+\beta} + \cdots + \bar{w}_{i,q_i-1}x^{r_i+(q_i-2)\beta}$.

(b) The second form is

$$\mathbf{w}^{[2]} = \begin{bmatrix} \mathbf{w}_1^{[2]} \\ \vdots \\ \mathbf{w}_n^{[2]} \end{bmatrix} \text{ with } \mathbf{w}_i^{[2]} = \begin{bmatrix} w_{i,0} \\ \vdots \\ w_{i,q_i-1} \end{bmatrix},$$

where $\deg w_{i,j} < r_i \leq \beta$ when $j = 0$ and $\deg w_{i,j} < \beta$ when $j \in \{1, \ldots, q_i - 2\}$. There is no degree restriction on w_{i,q_i-1}.

Proof. The first form is obtained by setting

$$\mathbf{u}^{[1]} = \begin{bmatrix} \mathbf{u}_1^{[1]} \\ \vdots \\ \mathbf{u}_n^{[1]} \end{bmatrix} \text{ with } \mathbf{u}_i^{[1]} = \begin{bmatrix} \bar{w}_{i,1} + \bar{w}_{i,2}x^{\beta} + \bar{w}_{i,3}x^{2\beta} + \cdots + \bar{w}_{i,q_i-1}x^{(q_i-2)\beta} \\ \bar{w}_{i,2} + \bar{w}_{i,3}x^{\beta} + \cdots + \bar{w}_{i,q_i-1}x^{(q_i-3)\beta} \\ \vdots \\ \bar{w}_{i,q_i-1} \end{bmatrix}.$$

Then $\bar{\mathbf{w}} + \mathbf{A}\mathbf{u}^{[1]}$ gives the first form. Note that $\mathbf{u}_i^{[1]}$ is empty if $q_i = 1$ and $\bar{\mathbf{w}}_i = \mathbf{w}_i^{[1]} = [\bar{w}_{i,0}]$ is not changed by the transformation.

The second form can be obtained based on the first form. Let

$$t_{i,j} = \begin{cases} r_i & \text{if } j = 0 \\ \beta & \text{if } j \in \{1, \ldots, q_i - 2\} \end{cases}$$

and write w_i from the first form as

$$w_i = w_{i,0} + w_{i,1}x^{r_i} + w_{i,2}x^{r_i+\beta} + \cdots + w_{i,q_i-1}x^{r_i+(q_i-2)\beta} \tag{4.3}$$

with $\deg w_{i,j} < t_{i,j}$ for $j < q_i - 1$. Note that in general $w_{i,j} \neq \bar{w}_{i,j}$, as $\deg \bar{w}_{i,j}$ may not be less than $t_{i,j}$. Now set

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} \text{ with } \mathbf{v}_i = \begin{bmatrix} w_{i,1} + w_{i,2}x^{\beta} + w_{i,3}x^{2\beta} + \cdots + w_{i,q_i-1}x^{(q_i-2)\beta} \\ w_{i,2} + w_{i,3}x^{\beta} + \cdots + w_{i,q_i-1}x^{(q_i-3)\beta} \\ \vdots \\ w_{i,q_i-1} \end{bmatrix}$$

and $\mathbf{u}^{[2]} = \mathbf{u}^{[1]} - \mathbf{v}$, which comes from the unimodular transformation

$$[\bar{\mathbf{w}}, \mathbf{A}] \left[\begin{array}{c|c} 1 & \\ \hline \mathbf{u}^{[1]} & \mathbf{I} \end{array}\right] \left[\begin{array}{c|c} 1 & \\ \hline -\mathbf{v} & \mathbf{I} \end{array}\right] = [\bar{\mathbf{w}}, \mathbf{A}] \left[\begin{array}{c|c} 1 & \\ \hline \mathbf{u}^{[1]} - \mathbf{v} & \mathbf{I} \end{array}\right].$$

Then $\mathbf{w}^{[2]} = \bar{\mathbf{w}} + \mathbf{A}\mathbf{u}^{[2]}$ is in the second form. Again note that $\mathbf{v}_i$ and $\mathbf{u}_i^{[2]}$ are empty if $q_i = 1$ and $\mathbf{w}_i^{[2]} = \bar{\mathbf{w}}_i = [\bar{w}_{i,0}]$. $\square$

Lemma 4.4. *Let* $\bar{\mathbf{w}} \in \left\langle \left(\bar{\mathbf{F}}, \sigma, \bar{s}\right) \right\rangle$ *and* $\mathbf{w}^{[2]}$ *be in the second form. If* $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} > -\alpha$ *or* $\deg_{\bar{s}} \mathbf{w}^{[2]} > -\alpha$*, then* $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} = \deg_{\bar{s}} \mathbf{w}^{[2]}$.

Proof. Consider the ith entry w_i of $\mathbf{E}\bar{\mathbf{w}}$ and the entries $\mathbf{w}_i^{[2]} = [w_{i,0}, \ldots, w_{i,q_i-1}]^T$ in $\mathbf{w}^{[2]}$. If $q_i = 1$, then $w_i = w_{i,0}$ and the corresponding shifts satisfies $s_i = \bar{s}_{\ell(i)}$, where $\ell(i) = \sum_{k=1}^{i} q_k$. Hence $\deg w_i + s_i = \deg w_{i,0} + \bar{s}_{\ell(i)}$. Thus we only need to consider the case where $q_i > 1$. Write w_i as in Equation (4.3). Note that $\deg w_{i,q_i-1} = \deg w_i - r_i - \beta\,(q_i - 2)$ and hence $\deg w_{i,q_i-1} - \alpha - \beta = \deg w_i - r_i - \alpha - \beta\,(q_i - 1)$, that is, $\deg w_{i,q_i-1} + \bar{s}_{\ell(i)} = \deg w_i + s_i$. It follows that

$$\begin{aligned}\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} &= \max_i(\deg w_i + s_i) = \max_i(\deg w_{i,q_i-1} + \bar{s}_{\ell(i)}) \\ &\le \max_{i,j}\left(\deg w_{i,j} + \bar{s}_{\ell(i-1)+j+1}\right) = \deg_{\bar{s}} \mathbf{w}^{[2]}.\end{aligned}$$

The only possible indices j where the inequality can be strict occur when $j < q_i - 1$. But $\deg w_{i,j} < \beta$ for all $j < q_i - 1$, which implies $\deg w_{i,j} + \bar{s}_{\ell(i-1)+j+1} = \deg w_{i,j} - \alpha - \beta < -\alpha$, and so it follows that the entries at these indices j do not contribute to $\deg_{\bar{s}} \mathbf{w}^{[2]}$ when $\deg_{\bar{s}} \mathbf{w}^{[2]} > -\alpha$ or $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} = \max_i(\deg w_{i,q_i-1} + \bar{s}_{\ell(i)}) > -\alpha$. In other words, if one of them exceeds $-\alpha$, then $\deg_{\bar{s}} \mathbf{w}^{[2]}$ and $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}}$ are determined only by entries at indices $j = q_i - 1$, but the equality always holds for these entries. $\square$

Remark 4.5. Notice that the first form $\mathbf{w}^{[1]}$ of $\bar{\mathbf{w}}$ has nonzero entries only at indices $I = [1, q_1 + 1, \ldots, \sum_{k=1}^{n-1} q_k + 1]$. Let $\mathbf{B}$ be a $\bar{n} \times n$ matrix with 1's at position $(\sum_{k=1}^{n-1} q_k + 1, i)$ and 0's everywhere else. Then the first form satisfies $\mathbf{w}^{[1]} = \mathbf{B}\mathbf{E}\bar{\mathbf{w}}$. Hence Lemma 4.4 provides the degree correspondence between the degrees of the first form $\mathbf{B}\mathbf{E}\bar{\mathbf{w}}$, which is just $\mathbf{E}\bar{\mathbf{w}}$ with zero rows added, and the second form $\bar{\mathbf{w}}^{[2]}$ of $\bar{\mathbf{w}}$.

Corollary 4.6. *Let* $\bar{\mathbf{w}} \in \left\langle\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)\right\rangle$ *and* $\mathbf{w}^{[2]}$ *be its second form. Then* $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} > -\alpha$ *if and only if* $\deg_{\bar{s}} \mathbf{w}^{[2]} > -\alpha$.

Proof. The proof follows directly from Lemma 4.4. $\square$

Lemma 4.7. *Let* $\bar{\mathbf{w}} \in \left\langle\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)\right\rangle$. *Then* $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} \le \deg_{\bar{s}} \bar{\mathbf{w}}$.

Proof. As in Lemma 4.4, consider the ith entry w_i of $\mathbf{E}\bar{\mathbf{w}}$ and the corresponding entries $\bar{\mathbf{w}}_i = \left[\bar{w}_{i,0}, \ldots, \bar{w}_{i,q_i-1}\right]^T$ in $\bar{\mathbf{w}}$. If $q_i = 1$, then $\deg w_i + s_i = \deg w_{i,0} + \bar{s}_{\ell(i)}$ as before. Thus we just need to consider the case $q_i > 1$, where the shifts for $\bar{\mathbf{w}}_i$ are

$-\alpha-\beta$. Since $w_i = \bar{w}_{i,0} + \bar{w}_{i,1}x^{r_i} + \bar{w}_{i,2}x^{r_i+\beta} + \cdots + \bar{w}_{i,q_i-1}x^{r_i+(q_i-2)\beta}$, we get

$$\begin{aligned}&\deg w_i\\ &= \max\left\{\deg \bar{w}_{i,0}, \deg \bar{w}_{i,1} + r_i, \deg \bar{w}_{i,2} + r_i + \beta, \ldots, \deg \bar{w}_{i,q_i-2} + r_i + (q_i-2)\beta\right\}.\end{aligned}$$

Then

$$\begin{aligned}\deg w_i + s_i &= \deg w_i - r_i - \alpha - \beta(q_i-1)\\ &= \max\{\deg \bar{w}_{i,0} - r_i - \alpha - \beta(q_i-1),\ \deg \bar{w}_{i,1} - \alpha - \beta(q_i-1),\ \ldots,\\ &\qquad \ldots, \deg \bar{w}_{i,q_i-2} - \alpha - \beta\}\\ &\le \max\left\{\deg \bar{w}_{i,0} - \alpha - \beta, \deg \bar{w}_{i,1} - \alpha - \beta, \ldots, \deg \bar{w}_{i,q_i-2} - \alpha - \beta\right\},\end{aligned}$$

and so $\deg_{\vec{s}} \mathbf{E}\bar{\mathbf{w}} \le \deg_{\bar{s}} \bar{\mathbf{w}}$. □

Corollary 4.8. *Let* $\bar{\mathbf{P}} = [\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2]$ *be a* $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$*-basis, where* $\deg_{\bar{s}} \bar{\mathbf{P}}_1 \le -\alpha$ *and* $\deg_{\bar{s}} \bar{\mathbf{P}}_2 > -\alpha$. *Let* $\bar{\mathbf{P}}_2^{[2]}$ *be the second form of* $\bar{\mathbf{P}}_2$. *Then* $\deg_{\bar{s}} \bar{\mathbf{P}}_2 = \deg_{\bar{s}} \bar{\mathbf{P}}_2^{[2]} = \deg_{\vec{s}} \mathbf{E}\bar{\mathbf{P}}_2$. *Hence* $[\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2^{[2]}]$ *is also a* $(\bar{\mathbf{F}}, \sigma, \bar{s})$*-basis.*

Proof. Since any column $\bar{\mathbf{p}}$ of $\bar{\mathbf{P}}_2$ satisfies $\deg_{\bar{s}} \bar{\mathbf{p}} > -\alpha$, from Lemma 4.4 and Lemma 4.7, we get

$$\deg_{\bar{s}} \bar{\mathbf{p}}^{[2]} = \deg_{\vec{s}} \mathbf{E}\bar{\mathbf{p}} \le \deg_{\bar{s}} \bar{\mathbf{p}}.$$

The inequality is in fact an equality, since otherwise, $\bar{\mathbf{p}}$ in $\bar{\mathbf{P}}$ can be replaced by $\bar{\mathbf{p}}^{[2]}$ to get a basis of lower degree, contradicting the minimality of $\bar{\mathbf{P}}$. Note that $\bar{\mathbf{P}}$ with its column $\bar{\mathbf{p}}$ replaced by $\bar{\mathbf{p}}^{[2]}$ remains to be a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis, since $\bar{\mathbf{p}}^{[2]} = \bar{\mathbf{p}} + \mathbf{A}\mathbf{u}$ involves column operations with only columns in $\bar{\mathbf{P}}_1$ as $\mathbf{A}$ has $\bar{s}$-degrees bounded by $-\alpha$ and hence is generated by $\bar{\mathbf{P}}_1$. □

Lemma 4.9. *If* $\mathbf{P}$ *is a* $(\mathbf{F}, \sigma, \vec{s})$*-basis, then* $[\mathbf{B}\mathbf{P}, \mathbf{A}]$ *is a basis for* $\left\langle\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)\right\rangle$.

Proof. Any $\bar{\mathbf{w}} \in \left\langle \left(\bar{\mathbf{F}}, \sigma, \bar{s}\right) \right\rangle$ can be transformed by $\mathbf{A}$ to the first form

$$\mathbf{w}^{[1]} = \bar{\mathbf{w}} + \mathbf{A}\mathbf{u}^{[1]} = \mathbf{B}\mathbf{E}\bar{\mathbf{w}},$$

where $\mathbf{E}\bar{\mathbf{w}} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle$ is generated by $\mathbf{P}$. That is,

$$\bar{\mathbf{w}} = \mathbf{w}^{[1]} - \mathbf{A}\mathbf{u}^{[1]} = \mathbf{B}\mathbf{E}\bar{\mathbf{w}} - \mathbf{A}\mathbf{u}^{[1]} = \mathbf{B}\mathbf{P}\mathbf{v} - \mathbf{A}\mathbf{u}^{[1]} = [\mathbf{B}\mathbf{P}, \mathbf{A}]\,[\mathbf{v}, -\mathbf{u}^{[1]}]^T.$$

One can also see that the columns of $\mathbf{A}$ and the columns of $\mathbf{BP}$ are linearly independent, as each zero row of $\mathbf{BP}$ has a -1 from a column of $\mathbf{A}$. □

Lemma 4.10. *If $\bar{\mathbf{P}}$ is a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis, then $\mathbf{E}\bar{\mathbf{P}}$ generates $\langle(\mathbf{F}, \sigma, \vec{s})\rangle$. That is, for any $\mathbf{w} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle$, there is an $\mathbf{u} \in \mathbb{K}[x]^{\bar{n}\times 1}$ such that $\mathbf{w} = \mathbf{E}\bar{\mathbf{P}}\mathbf{u}$.*

Proof. For any $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P}$, the columns of $\mathbf{BP}$ are in $\langle(\bar{\mathbf{F}}, \sigma, \bar{s})\rangle$ generated by $\bar{\mathbf{P}}$, that is, $\mathbf{BP} = \bar{\mathbf{P}}\mathbf{U}$ for some $\mathbf{U} \in \mathbb{K}[x]^{\bar{n}\times n}$. Hence $\mathbf{EBP} = \mathbf{P}$ is generated by $\mathbf{E}\bar{\mathbf{P}}$. That is, $\mathbf{P} = \mathbf{E}\bar{\mathbf{P}}\mathbf{U}$. Then any $\mathbf{w} \in \langle(\mathbf{F}, \sigma, \vec{s})\rangle$, which satisfies $\mathbf{w} = \mathbf{P}\mathbf{v}$ for some $\mathbf{v} \in \mathbb{K}[x]^{n\times 1}$, satisfies $\mathbf{w} = \mathbf{E}\bar{\mathbf{P}}\mathbf{U}\mathbf{v}$. □

We are now ready to prove the main result on the correspondence between a high degree part of a basis of the transformed problem and that of the original problem.

Theorem 4.11. *Let $\bar{\mathbf{P}} = [\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2]$ be a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis, where $\deg_{\bar{s}} \bar{\mathbf{P}}_1 \leq -\alpha$ and $\deg_{\bar{s}} \bar{\mathbf{P}}_2 > -\alpha$. Then $\mathbf{E}\bar{\mathbf{P}}_2$ is the matrix of the columns of a $(\mathbf{F}, \sigma, \vec{s})$-basis whose $\vec{s}$-degrees exceed $-\alpha$.*

Proof. We want to show that $[\mathbf{P}_1, \mathbf{E}\bar{\mathbf{P}}_2]$ is a $(\mathbf{F}, \sigma, \vec{s})$-basis for any $(\mathbf{F}, \sigma, \vec{s})_{-\alpha}$-basis $\mathbf{P}_1$. First, $\mathbf{E}\bar{\mathbf{P}}$ has order $(\mathbf{F}, \sigma)$ since $\bar{\mathbf{F}}\bar{\mathbf{P}} = \mathbf{F}\mathbf{E}\bar{\mathbf{P}}$ and $\bar{\mathbf{P}}$ has order $\left(\bar{\mathbf{F}}, \sigma\right)$. Also, since $\mathbf{E}\bar{\mathbf{P}}$ generates $\langle(\mathbf{F}, \sigma, \vec{s})\rangle$ by Lemma 4.10, and from Corollary 4.6 $\mathbf{E}\bar{\mathbf{P}}_1$ has $\vec{s}$-degree bounded by $-\alpha$ hence is generated by $\mathbf{P}_1$, it follows that $\left[\mathbf{P}_1, \mathbf{E}\bar{\mathbf{P}}_2\right]$ generates $\langle(\mathbf{F}, \sigma, \vec{s})\rangle$.

It only remains to show that the $\vec{s}$-degrees of $\mathbf{E}\bar{\mathbf{P}}_2$ are minimal. Suppose not, then $[\mathbf{P}_1, \mathbf{E}\bar{\mathbf{P}}_2]$ can be reduced to $[\mathbf{P}_1, \tilde{\mathbf{P}}_2]$ where $\tilde{\mathbf{P}}_2$ has a column having lower $\vec{s}$-degree than that of the corresponding column in $\mathbf{E}\bar{\mathbf{P}}_2$. That is, assuming the columns of $\tilde{\mathbf{P}}_2$ and $\mathbf{E}\bar{\mathbf{P}}_2$ are in non-decreasing $\vec{s}$-degrees order, then we can find the first index i where the $\vec{s}$-degree of ith column of $\tilde{\mathbf{P}}_2$ is lower than the $\vec{s}$-degree of the ith column of $\mathbf{E}\bar{\mathbf{P}}_2$. It follows that $[\mathbf{B}\mathbf{P}_1, \mathbf{B}\mathbf{E}\bar{\mathbf{P}}_2]$ can be reduced to $[\mathbf{B}\mathbf{P}_1, \mathbf{B}\tilde{\mathbf{P}}_2]$ and $[\mathbf{B}\mathbf{P}_1, \mathbf{B}\mathbf{E}\bar{\mathbf{P}}_2, \mathbf{A}]$ can be reduced to $[\mathbf{B}\mathbf{P}_1, \mathbf{B}\tilde{\mathbf{P}}_2, \mathbf{A}]$. Since $[\mathbf{B}\mathbf{P}_1, \mathbf{B}\tilde{\mathbf{P}}_2, \mathbf{A}]$ generates $\langle(\bar{\mathbf{F}}, \sigma, \bar{s})\rangle$ by Lemma 4.9, it can be reduced to $\bar{\mathbf{P}} = [\bar{\mathbf{P}}_1, \bar{\mathbf{P}}_2]$. But it can also be reduced to $[\bar{\mathbf{P}}_1, \tilde{\mathbf{P}}_2^{[2]}, \mathbf{A}]$ with $\tilde{\mathbf{P}}_2^{[2]}$ the second form of $\mathbf{B}\tilde{\mathbf{P}}_2$, and to $[\bar{\mathbf{P}}_1, \tilde{\mathbf{P}}_2^{[2]}]$ as the columns of $\mathbf{A}$ are generated by the $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)_{-\alpha}$-basis $\bar{\mathbf{P}}_1$.

In order to reach a contradiction we just need to show that $\tilde{\mathbf{P}}_2^{[2]}$ has a column with $\bar{s}$-degree less than that of the corresponding column in $\bar{\mathbf{P}}_2$. Let $\tilde{\mathbf{w}}$ be the first column of $\tilde{\mathbf{P}}_2$ with $\vec{s}$-degree less than that of the corresponding column $\mathbf{w}$ in $\mathbf{E}\bar{\mathbf{P}}_2$ and let $\bar{\mathbf{w}}$ be the corresponding column in $\bar{\mathbf{P}}_2$. By Corollary 4.8 $\deg_{\vec{s}} \mathbf{w} = \deg_{\bar{s}} \bar{\mathbf{w}}$. Let $\tilde{\mathbf{w}}^{[2]}$ be the second form of $\mathbf{B}\tilde{\mathbf{w}}$, which is a column in $\tilde{\mathbf{P}}_2^{[2]}$ corresponding to the column $\bar{\mathbf{w}}$ in $\bar{\mathbf{P}}_2$. We know that either $\deg_{\bar{s}} \tilde{\mathbf{w}}^{[2]} \leq -\alpha$ or $\deg_{\bar{s}} \tilde{\mathbf{w}}^{[2]} = \deg_{\vec{s}} \tilde{\mathbf{w}}$ by Lemma 4.4, as $\mathbf{E}\tilde{\mathbf{w}}^{[2]} = \mathbf{E}(\mathbf{B}\tilde{\mathbf{w}} + \mathbf{A}\mathbf{u}) = \tilde{\mathbf{w}}$. In either case, $\deg_{\bar{s}} \tilde{\mathbf{w}}^{[2]} < \deg_{\bar{s}} \bar{\mathbf{w}}$, as $\deg_{\bar{s}} \bar{\mathbf{w}}$ is greater than both $-\alpha$ and $\deg_{\vec{s}} \tilde{\mathbf{w}}$. Hence we have $[\bar{\mathbf{P}}_1, \tilde{\mathbf{P}}_2^{[2]}]$ is another $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis with lower $\bar{s}$-degrees than $\bar{\mathbf{P}}$, contradicting with the minimality of $\bar{\mathbf{P}}$. $\square$

4.2.3 Achieving Efficient Computation

Theorem 4.11 essentially tells us that a high degree part of a $(\mathbf{F}, \sigma, \vec{s})$-basis can be determined by computing a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis, something we know can be done efficiently. Notice the parallel between the situation here and in the earlier balanced shift case, where the transformed problem also allows us to compute a partial $(\mathbf{F}, \sigma, \vec{s})$-basis, albeit a low degree part, in each iteration.

After a $\left(\bar{\mathbf{F}}, \sigma, \bar{s}\right)$-basis, or equivalently a high degree part of a $(\mathbf{F}, \sigma, \vec{s})$-basis, is computed, for the remaining problem of computing the remaining basis elements, we can in fact reduce the dimension of the input $\mathbf{F}$ by removing some of its columns corresponding to the high shift entries.

Theorem 4.12. *Suppose without loss of generality that the entries of $\vec{s}$ are in non-decreasing order. Let I be the index set containing the indices of entries s_i in $\vec{s}$ such that $s_i \leq -\alpha$. Let $\mathbf{F}_I$ be the columns of $\mathbf{F}$ indexed by I. Then a $(\mathbf{F}_I, \sigma, \vec{s})_{-\alpha}$-basis $\mathbf{P}_1$ gives a $(\mathbf{F}, \sigma, \vec{s})_{-\alpha}$-basis $\left[\mathbf{P}_1^T, \mathbf{0}\right]^T$.*

Proof. For any $\mathbf{p} \in \mathbb{K}[x]^{n \times 1}$ and $\deg_{\vec{s}} \mathbf{p} \leq -\alpha$, note that if the ith entry of the shift satisfies $s_i \leq -\alpha$, then the corresponding entry p_i of $\mathbf{p}$ is zero. Otherwise, if $p_i \neq 0$ then the $\vec{s}$-degree of $\mathbf{p}$ is at least $s_i > -\alpha$, contradicting the assumption that the $\vec{s}$-degree of $\mathbf{p}$ is lower than or equal to $-\alpha$. □

Thus, these zero entries do not need to be considered in the remaining problem of computing a $(\mathbf{F}, \sigma, \vec{s})_{-\alpha}$-basis. As such the corresponding columns from the input matrix $\mathbf{F}$ can be removed.

Example 4.13. Let us return to Example 4.2. When the parameters $\alpha = \beta = 1$, after computing an element of a $(\mathbf{F}, 8, \vec{s})$-basis with $\vec{s}$-degree 0 that exceeds $-\alpha = -1$, the first row of any $(\mathbf{F}, \sigma, \vec{s})_{-1}$-basis must be zero by Theorem 4.12 (since the first entry of $\vec{s} = [0, -3, -5, -6]$ is $0 > -\alpha$). This is illustrated by the $(\mathbf{F}, 8, \vec{s})$-basis $\mathbf{P}$ given in Example 4.2. This implies that the first column of $\mathbf{F}$ is not needed in the subsequent computation of the remaining basis elements.

Corollary 4.14. *If the shift $\vec{s}$ satisfies condition (4.2) and c is a constant greater than or equal to 1, then a $(\mathbf{F}, \sigma, \vec{s})_{-ca}$-basis has at most n/c basis elements.*

Proof. Since $a = m\sigma/n \geq -\sum_{i=1}^{n} s_i/n$ under condition (4.2), there cannot be more than n/c entries of $\vec{s}$ less than or equal to $-ca$. By Theorem 4.12, the only possible nonzero rows of a $(\mathbf{F}, \sigma, \vec{s})_{-cd}$-basis are the ones corresponding to (with the same indices as) the shift entries that are less than or equal to $-ca$. Hence there cannot be more than n/c nonzero rows and at most n/c columns, as the columns are linearly independent. $\square$

We now have a situation similar to that found in the balanced shift case. Namely, for each iteration we transform the problem using appropriate parameters α and β to efficiently compute the basis elements with degrees greater than $-\alpha$. Then we can remove columns from the input matrix $\mathbf{F}$ corresponding to the shift entries that are greater than $-\alpha$. We can then repeat the same process again, with a larger α and β, in order to compute more basis elements.

Theorem 4.15. *If the shift $\vec{s}$ satisfies condition (4.2), then a $(\mathbf{F}, \sigma, \vec{s})$-basis can be computed with cost $O\left(n^{\omega} \operatorname{M}(a) \log \sigma\right) \subset O^{\sim}(n^{\omega} a)$.*

Proof. We give the following constructive proof. Initially, we set transformation parameters $\alpha_1 = \beta_1 = 2a$ with $a = m\sigma/n \geq -\sum_{i=1}^{n} s_i/n$. Algorithm 3.1 works efficiently on the transformed problem as the shift $\bar{s}^{(1)}$ is balanced and the dimension of $\bar{\mathbf{F}}_1$ remains $O(n)$. By Theorem 4.11 this gives the basis elements of $(\mathbf{F}, \sigma, \vec{s})$-basis with $\vec{s}$-degree exceeding $-\alpha_1 = -2a$. By Corollary 4.14, the number of basis elements remaining to be computed is at most $n/2$, hence the number of elements correctly computed is at least $n/2$. By Theorem 4.12, this also allows us to remove at least half of the columns from the input $\mathbf{F}$ and correspondingly at least half of the rows from the output for the remaining problem. Thus the new input matrix $\mathbf{F}_2$ has a new column dimension $n_2 \leq n/2$ and the corresponding shift $\vec{s}^{(2)}$ has n_2 entries. The average degree of the new problem is $a_2 = m\sigma/n_2$.

For the second iteration, we set α_2 and β_2 to $2a_2$. Since

$$\alpha_2 = 2m\sigma/n_2 \geq -2\sum_{i=1}^{n} s_i/n_2 \geq -2\sum_{i=1}^{n_2} s_i^{(2)}/n_2,$$

this allows us to reduce the dimension n_3 of $\mathbf{F}_3$ to at most $n_2/2$ after finishing computing a $\left(\bar{\mathbf{F}}_2, \sigma, \bar{s}^{(2)}\right)_{-\alpha_1}$-basis. Again, this can be done using Algorithm 3.1 with a cost of $O\left(n_2^\omega \operatorname{M}(a_2)\log\sigma\right)$ as the shift $\bar{a}_2$ is balanced and the dimension of $\bar{\mathbf{F}}_2$ is $O\left(n_2\right)$. Repeating this process, at iteration i, we set $\alpha_i = \beta_i = 2a_i = 2m\sigma/n_i$. The transformed problem has a balanced shift $\bar{a}_i$ and column dimension $O\left(n_i\right)$. So a $\left(\bar{\mathbf{F}}_i, \sigma, \bar{s}^{(i)}\right)_{-\alpha_{i-1}}$-basis can be computed with a cost of

$$\begin{aligned} O\left(n_i^\omega \operatorname{M}\left(a_i\right)\log\sigma\right) &\subset O\left(\left(2^{-i}n\right)^\omega \operatorname{M}\left(2^i a\right)\log\sigma\right) \\ &\subset O\left(\left(2^{-i}n\right)^\omega \operatorname{M}\left(2^i\right)\operatorname{M}\left(a\right)\log\sigma\right) \\ &\subset O\left(\left(2^{-i}n\right)^\omega \left(2^i\right)^{\omega-1}\operatorname{M}\left(a\right)\log\sigma\right) \\ &\subset O\left(2^{-i}n^\omega \operatorname{M}(a)\log\sigma\right). \end{aligned}$$

Since

$$\alpha_i = 2m\sigma/n_i \geq -2\sum_{i=1}^{n} s_i/n_i \geq -2\sum_{i=1}^{n_i} s_i^{(i)}/n_i,$$

the column dimension n_{i+1} of the next problem can again be reduced by a half. After iteration i, at most $n/2^i$ $(\mathbf{F}, \sigma, \vec{s})$-basis elements remain to be computed. We can stop this process when the column dimension n_i of the input matrix $\mathbf{F}_i$ reaches the row dimension m, as an order basis can be efficiently computed in such case. Therefore, a complete $(\mathbf{F}, \sigma, \vec{s})$-basis can be computed in at most $\log(n/m)$ iterations, so the overall cost is

$$\begin{aligned} O\left(\sum_{i=1}^{\log(n/m)}\left(2^{-i}n^\omega \operatorname{M}(a)\log\sigma\right)\right) &= O\left(n^\omega \operatorname{M}(a)\log\sigma \sum_{i=1}^{\log(n/m)} 2^{-i}\right) \\ &\subset O\left(n^\omega \operatorname{M}(a)\log\sigma\right) \end{aligned}$$

field operations. □

Finally, we remark that when the condition (4.2) is relaxed to $\sum_{i=1}^{n} -s_i \in$

Algorithm 4.1 unbalancedFastOrderBasis $(\mathbf{F}, \sigma, \vec{s})$

Input: $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$, $\sigma \in \mathbb{Z}_{\geq 0}$, $\vec{s}$ satisfies condition (4.2).
Output: $\mathbf{P} \in \mathbb{K}[x]^{n \times n}$, an $(\mathbf{F}, \sigma, \vec{s})$-basis.
Uses:
(a) TransformUnbalanced : converts an unbalanced shift problem to a balanced one using the transformation described in Chapter 4. Returns transformed input matrix, transformed shift, and transformation matrix.
(b) fastOrderBasis : computes order basis with balanced shift.

```
1:  $i := 1$; $\mathbf{P} = [\,]$;
2:  $\mathbf{F}^{(i)} := \mathbf{F}$, $\vec{s}^{(i)} := \vec{s}$;
3:  while columnDimension($\mathbf{P}$) $\neq n$ do
4:      $d_i = \lceil m\sigma / \text{columnDimension}(\mathbf{F}^{(i)}) \rceil$;
5:      $\alpha_i := \beta_i := 2d_i$;
6:      $[\bar{\mathbf{F}}^{(i)}, \bar{s}^{(i)}, \mathbf{E}]$ := TransformUnbalanced $(\mathbf{F}^{(i)}, \vec{s}^{(i)}, \alpha_i, \beta_i)$;
7:      $\bar{\mathbf{P}}^{(i)}$ := fastOrderBasis $(\bar{\mathbf{F}}^{(i)}, \sigma, \bar{s}^{(i)})$;
8:      Set $\mathbf{P}^{(i)}$ to be the columns of $\mathbf{E}\bar{\mathbf{P}}^{(i)}$ with $\bar{s}_i$-column degrees in $(-\alpha_i, -\alpha_{i-1}]$;
9:      $\mathbf{P} := [\mathbf{P}^{(i)}, \mathbf{P}]$;
10:     Set $I$ as the set of indices $i$ satisfying $s_i \leq -\alpha_i$;
11:     $\mathbf{F}^{(i+1)} := \mathbf{F}^{(i)}_I$, $\vec{s}^{(i+1)} := \vec{s}^{(i)}_I$;
12:     $i := i + 1$;
13: end while
14: return $\mathbf{P}$ ;
```

$O(m\sigma)$, so that $\sum_{i=1}^{n} -s_i \leq cm\sigma$ for a constant c, we can still compute a $(\mathbf{F}, \sigma, \vec{s})$-basis with the same complexity, by setting $\alpha_i = \beta_i = 2cm\sigma/n_i$ at each iteration i and following the same procedure as above. The cost at each iteration i remains $O^{\sim}(n^{\omega}a)$, and the entire computation still uses at most $\log(n/m)$ iterations.

Chapter 5

Kernel Basis

In this chapter we discuss the computation of minimal kernel bases.

Minimal kernel bases can be directly computed via order basis computation. Indeed if the order σ of a $(\mathbf{F}, \sigma, \vec{s})$-basis $\mathbf{P}$ is high enough, then $\mathbf{P}$ contains a $\vec{s}$-minimal kernel basis $\mathbf{N}$. However, this approach may require the order σ to be quite high. For example, if $\mathbf{F}$ has degree d and $\vec{s}$ is uniform, then its minimal kernel bases can have degree up to md. In that case the order σ would need to be set to $d + md$ in the order basis computation in order to fully compute a minimal kernel basis. The fastest method of computing such a $(\mathbf{F}, d + md)$-basis would cost $O^{\sim}(n^{\omega-1}m^2d)$ using the algorithm from chapter 3. We can see from this cost that there is room for improvement when m is large. For example, in the worst case when $m \in \Theta(n)$, this cost would be $O^{\sim}(n^{\omega+1}d)$. This points to a root cause for the inefficiency in this approach. Namely, when m is large, the computed kernel basis, which can have a column dimension of $n - m$, is a much smaller subset of the order basis computed. Hence considerable effort is put in the computation of order basis elements that are not part of a kernel basis. A key to reducing the cost is therefore to reduce such computation of unneeded order basis elements, which is achieved in our algorithm by only using order basis computation to compute partial kernel bases of low degrees.

5.1 Minimal Kernel Basis Computation

In this section, we describe a new, efficient algorithm for computing a shifted minimal kernel basis. The algorithm uses two computation processes recursively. The first process, described in Subsection 5.1.2, uses an order basis computation to compute a subset of kernel basis elements of lower degree, and results in a new problem of lower column dimension. The second process, described in Subsection 5.1.4, reduces the row dimension of the problem by computing a kernel basis of a submatrix formed by a subset of the rows of the input matrix.

We assume that the row dimension m is bounded by the column dimension n in this chapter. But this assumption is later removed in Subsection 11.3.1 with results from Chapter 11.

We require that the shift $\vec{s}$ bounds the column degrees of $\mathbf{F}$, that is, $\vec{s} \geq \operatorname{cdeg} \mathbf{F}$. For example, we can set each entry of $\vec{s}$ to be the corresponding column degree of $\mathbf{F}$, or we can simply set each entry of $\vec{s}$ to be the maximum column degree of $\mathbf{F}$. This is a very useful condition as it helps us to keep track of and bound the shifted degrees throughout the kernel basis computation, as we will see in Subsection 5.1.1.

For simplicity, we will also assume without loss of generality that the columns of $\mathbf{F}$ and the corresponding entries of $\vec{s} = [s_1, \ldots, s_n]$ are arranged so that the entries of $\vec{s}$ are in increasing order.

Let $\rho = \sum_{n-m+1}^{n} s_i$ be the sum of m largest entries of $\vec{s}$, and $s = \rho/m$ be their average. The algorithm we present in this section computes a $\vec{s}$-minimal kernel space basis $\mathbf{N}$ with a cost of $O^{\sim}(n^{\omega}s)$ field operations. For uniform shift $\vec{s} = [s, \ldots, s]$, we improve this later to $O^{\sim}\left(n^{\omega-1}ms\right)$.

5.1.1 Bounds based on the shift

A key requirement for efficient computation is making sure that the intermediate computations do not blow up in size. We will see that this requirement is satisfied by the existence of a bound, $\xi = \sum \vec{s} = \sum_{i=1}^{n} s_i,$ on the sum of all entries of the input shift of all subproblems throughout the computation.

The following lemma shows that any $(\mathbf{F}, \sigma, \vec{s})$-basis contains a partial $\vec{s}$-minimal kernel basis of $\mathbf{F}$, and as a result, any $(\mathbf{F}, \sigma, \vec{s})$-basis with high enough σ contains a $\vec{s}$-minimal kernel basis of $\mathbf{F}$.

Lemma 5.1. *Let $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ be any $(\mathbf{F}, \sigma, \vec{s})$-basis and $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2]$ be any $\vec{s}$-minimal kernel basis of $\mathbf{F}$, where $\mathbf{P}_1$ and $\mathbf{N}_1$ contain all columns from $\mathbf{P}$ and $\mathbf{N}$, respectively, whose $\vec{s}$-column degrees are less than σ. Then $[\mathbf{P}_1, \mathbf{N}_2]$ is a $\vec{s}$-minimal kernel basis of $\mathbf{F}$, and $[\mathbf{N}_1, \mathbf{P}_2]$ is a $(\mathbf{F}, \sigma, \vec{s})$-basis.*

Proof. From Lemma 2.5, any column $\mathbf{p}$ of $\mathbf{P}_1$ satisfies $\deg \mathbf{F}\mathbf{p} \leq \deg_{\vec{s}} \mathbf{p} < \sigma$. Combining this with the fact that $\mathbf{F}\mathbf{p} \equiv 0 \mod x^{\sigma}$ we get $\mathbf{F}\mathbf{p} = 0$. Thus $\mathbf{P}_1$ is generated by $\mathbf{N}_1$, that is, $\mathbf{P}_1 = \mathbf{N}_1\mathbf{U}$ for some polynomial matrix $\mathbf{U}$. On the other hand, $\mathbf{N}_1$ has order $(\mathbf{F}, \sigma)$ and therefore satisfies $\mathbf{N}_1 = \mathbf{P}_1\mathbf{V}$ for some polynomial matrix $\mathbf{V}$. We now have $\mathbf{P}_1 = \mathbf{P}_1\mathbf{V}\mathbf{U}$ and $\mathbf{N}_1 = \mathbf{N}_1\mathbf{U}\mathbf{V}$, implying both $\mathbf{U}$ and $\mathbf{V}$ are unimodular. The result then follows from the unimodular equivalence of $\mathbf{P}_1$ and $\mathbf{N}_1$ and the fact that they are $\vec{s}$-column reduced. $\square$

We can now provide a simple bound on the $\vec{s}$-minimal kernel basis of $\mathbf{F}$.

Theorem 5.2. *Suppose $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$ and $\vec{s} \in \mathbb{Z}_{\geq 0}^{n}$ is a shift with entries bounding the corresponding column degrees of $\mathbf{F}$. Then the sum of the $\vec{s}$-column degrees of any $\vec{s}$-minimal kernel basis of $\mathbf{F}$ is bounded by $\xi = \sum \vec{s}$.*

Proof. Let $\mathbf{P}$ be a $(\mathbf{F}, \sigma, \vec{s})$-basis with high enough order σ so that $\mathbf{P} = \left[\mathbf{N}, \bar{\mathbf{N}}\right]$ contains a complete kernel basis, $\mathbf{N}$, of $\mathbf{F}$. By Lemma 5.1 we just need σ to be greater than the $\vec{s}$-column degree of a $\vec{s}$-minimal kernel basis of $\mathbf{F}$. Let r be the column dimension of $\bar{\mathbf{N}}$. Note that this is the same as the rank of $\mathbf{F}$. By Lemma 3.2 the sum of the $\vec{s}$-column degrees of $\mathbf{P}$ is at most $\xi + r\sigma$. By Lemma 2.5 the sum of the $\vec{s}$-column degrees of $\bar{\mathbf{N}}$ is greater than or equal to the sum of the column degrees of $\mathbf{F} \cdot \bar{\mathbf{N}}$, which is at least $r\sigma$, since every column of $\mathbf{F}\bar{\mathbf{N}}$ is nonzero and has order σ. So the sum of the $\vec{s}$-column degrees of $\mathbf{N}$ is bounded by $\xi + r\sigma - r\sigma = \xi$. □

Theorem 5.2 specializes to the following well-known results in the case of uniform shift:

Corollary 5.3. *Given $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ with degree d . The sum of the column degree of its minimal kernel basis is bounded by md.*

Proof. From Theorem 5.2 and the definition of shifted column degrees we have

$$nd \geq \sum \operatorname{cdeg}_{[d,\dots d]} \mathbf{N} \geq (n-m)d + \sum \operatorname{cdeg} \mathbf{N},$$

which gives

$$\sum \operatorname{cdeg} \mathbf{N} \leq nd - (n-m)d = md.$$

□

5.1.2 Reducing the column dimension via order basis computation

In this subsection we look at how an order basis computation can be used to reduce the column dimension of our problem. While order basis computations were also used in [Storjohann and Villard, 2005] to reduce the column dimensions of their problems, here order basis computations are used in a more comprehensive way. In particular, Theorem 5.8 given later in this section, allows us to maintain the minimality of the bases with the use of the shifted degrees and the residuals.

We begin by computing a $(\mathbf{F}, 3s, \vec{s})$-basis $\mathbf{P}$, which can be done with a cost of $O^{\sim}(n^{\omega}s)$ using the algorithm from Giorgi et al. [2003]. Note that if $\vec{s}$ is balanced, then we can compute this with a cost of $O^{\sim}(n^{\omega-1}ms)$ using the algorithm from Chapter 3. We will show that at most $\frac{3m}{2}$ columns of $\mathbf{P}$ are not elements of a kernel basis of $\mathbf{F}$.

Remark 5.4. Note that it is not essential to choose $3s$ for the order. The order can be set to ℓs for any constant $\ell > 1$. A smaller ℓ means less work to compute a $(\mathbf{F}, \ell s, \vec{s})$-basis, but also results in fewer kernel basis elements computed and leaves more work for computing the remaining basis elements. On the other hand, a larger ℓ means more work is needed for order basis computation, but leaves less remaining work. It may be possible to better balance these computations with a better choice of ℓ. However, as we will see later, the resulting complexity given in this paper would remain the same for any $\ell > 1$ as long as we use the big O notation and do not care about the constant factors in the cost.

Theorem 5.5. *Let $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ be a $(\mathbf{F}, \sigma, \vec{s})$-basis with $\sigma > s$ and $\mathbf{P}_1$ containing all columns $\mathbf{n}$ of $\mathbf{P}$ satisfying $\mathbf{F}\mathbf{n} = 0$. Then for $\ell = \sigma/s$ the column dimension κ of $\mathbf{P}_2$ is bounded by $\frac{\ell m}{(\ell-1)}$.*

Proof. Any column $\mathbf{p}$ of $\mathbf{P}_2$ has order σ but also satisfies $\mathbf{F}\mathbf{p} \neq 0$. Thus the degree of $\mathbf{F}\mathbf{p}$ must be at least σ and, by Lemma 2.5, $\mathbf{p}$ must have $\vec{s}$-column degree at least σ. It follows that the sum of the $\vec{s}$-column degrees of the columns of $\mathbf{P}_2$ must satisfy $\sum \deg_{\vec{s}} \mathbf{P}_2 \geq \kappa\sigma$. From Lemma 3.2 we know that the sum of the $\vec{s}$-column degrees of the columns of $\mathbf{P}$ satisfies $\sum \deg_{\vec{s}} \mathbf{P} \leq \sum \vec{s} + m\sigma$, and hence the sum of $\vec{s}$-column degrees of the columns of $\mathbf{P}_1$ must satisfy

$$\sum \deg_{\vec{s}} \mathbf{P}_1 = \sum \deg_{\vec{s}} \mathbf{P} - \sum \deg_{\vec{s}} \mathbf{P}_2 \leq \sum \vec{s} + m\sigma - \kappa\sigma.$$

On the other hand, the lowest possible value of $\sum \deg_{\vec{s}} \mathbf{P}_1$ is $\sum_{i=1}^{n-\kappa} s_i$, the sum of the $n-\kappa$ smallest entries of $\vec{s}$ (which occurs when $\mathbf{P}_1 = [\mathbf{I}, 0]^T$). It follows that

$$\sum \vec{s} + m\sigma - \kappa\sigma \geq \sum_{i=1}^{n-\kappa} s_i,$$

or, after rearrangement,

$$m\sigma \geq \kappa\sigma - \left(\sum \vec{s} - \sum_{i=1}^{n-\kappa} s_i \right).$$

Combining this with the fact that for $\kappa \geq m$ the average of the κ largest entries of $\vec{s}$ is no more than the average of the m largest entries of $\vec{s}$, that is,

$$\left(\sum \vec{s} - \sum_{i=1}^{n-\kappa} s_i \right) / \kappa \leq s, \text{ or } \sum \vec{s} - \sum_{i=1}^{n-\kappa} s_i \leq \kappa s,$$

we get $m\sigma \geq \kappa\sigma - \kappa s$, which gives $\kappa \leq m\sigma/(\sigma - s)$ for $\sigma > s$. Substituting in $\sigma = \ell s$, we get $\kappa \leq \frac{\ell m}{(\ell - 1)}$ as required. □

Let $[\mathbf{P}_1, \mathbf{P}_2] = \mathbf{P}$ with $\mathbf{P}_1$ consisting of the kernel basis elements computed. Then the residual $\mathbf{FP} = [\mathbf{0}, \mathbf{FP}_2]$ can be used to compute the remaining kernel basis elements. Before showing this can be correctly done, let us first make sure that the matrix multiplication $\mathbf{FP}_2$ can be done efficiently, which may not be obvious since $\mathbf{F}$, $\mathbf{P}_2$, and their product $\mathbf{FP}_2$ can all have degrees up to $\Theta(\xi)$. But we do have the sum of the column degrees of $\mathbf{F}$, that of $\mathbf{FP}_2$, and the sum of the $\vec{s}$-column degrees of $\mathbf{P}_2$ all bounded by $O(\xi)$, which means their total size are not too big but their column degrees can be quite unbalanced. We will encounter this type of multiplication again multiple times, for computing residuals and combining results. In fact, almost all of the matrices in our kernel basis computation can have such unbalanced degrees. To efficiently multiply these matrices, we provide the following theorem, whose proof we defer until the end of this section. In the following, let $t = \xi/m$ and s be the average of the largest m entries of $\vec{s}$ as before.

Theorem 5.6. *Let* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *with* $m \leq n$, $\vec{s} \in \mathbb{Z}^n$ *a shift with entries bounding the column degrees of* $\mathbf{A}$ *and* ξ, *a bound on the sum of the entries of* $\vec{s}$. *Let* $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$ *with* $k \in O(m)$ *and the sum* θ *of its* $\vec{s}$*-column degrees satisfying* $\theta \in O(\xi)$. *Then we can multiply* $\mathbf{A}$ *and* $\mathbf{B}$ *with a cost of* $O^{\sim}(nm^{\omega-1}t)$.

With Theorem 5.6, we can now do the multiplication $\mathbf{FP}_2$ efficiently.

Corollary 5.7. *The multiplication of* $\mathbf{F}$ *and* $\mathbf{P}_2$ *can be done with a cost of* $O^{\sim}(nm^{\omega-1}t)$.

Proof. Since $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ is a $(\mathbf{F}, 3s, \vec{s})$-basis, we have from Lemma 3.2 that the sum of the $\vec{s}$-column degrees of $\mathbf{P}_2$ satisfies $\sum \deg_{\vec{s}} \mathbf{P}_2 \leq 3sm + \xi \leq 4\xi$. Hence Theorem 5.6 applies. $\square$

It remains to show that the residual $\mathbf{FP}_2$ can be used to compute the remaining kernel basis elements.

Theorem 5.8. *Let* $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ *be a* $(\mathbf{F}, \sigma, \vec{s})$*-basis such that* $\mathbf{P}_1$ *consists of all the kernel basis elements of* $\mathbf{F}$ *in* $\mathbf{P}$. *Let* $\vec{b} = [\vec{b}_1, \vec{b}_2]$ *be the* $\vec{s}$*-column degrees of* $\mathbf{P}$, *where* $\vec{b}_1, \vec{b}_2$ *are the* $\vec{s}$*-column degrees of* $\mathbf{P}_1$, $\mathbf{P}_2$ *respectively. Let* $\mathbf{Q}$ *be a* $\vec{b}_2$*-minimal kernel basis of* $\mathbf{FP}_2$ *with* $\vec{b}_2$*-column degrees* $\vec{b}_2'$. *Then* $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$ *is a* $\vec{s}$*-minimal kernel basis of* $\mathbf{F}$ *with* $\vec{s}$*-column degrees* $[\vec{b}_1, \vec{b}_2']$.

Proof. Let $\mathbf{Q}' = \operatorname{diag}([I, \mathbf{Q}])$, where the dimension of the identity matrix I matches that of $\mathbf{P}_1$. Then $\mathbf{Q}'$ is a $\vec{b}$-minimal kernel basis of $\mathbf{FP}$ since $\mathbf{FPQ}' = [\mathbf{FP}_1, \mathbf{FP}_2\mathbf{Q}] = 0$. It follows that $\mathbf{PQ}' = [\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$ is a kernel basis of $\mathbf{F}$. We now show that $\mathbf{PQ}'$ is $\vec{s}$-column reduced and has $\vec{s}$-column degrees $[\vec{b}_1, \vec{b}_2']$, or equivalently, $x^{\vec{s}}\mathbf{PQ}'$ is column reduced and has column degrees $[\vec{b}_1, \vec{b}_2']$. Notice that $x^{\vec{s}}\mathbf{P}$ has column degrees $[\vec{b}_1, \vec{b}_2]$ and a full rank leading column coefficient matrix P. Hence $x^{\vec{s}}\mathbf{P}x^{-[\vec{b}_1, \vec{b}_2]}$ has column degrees $[0, \ldots 0]$. (If one is concerned about the entries not being polynomials, one can simply multiply the matrix by x^{ξ} to shift the degrees up.) Similarly, $x^{\vec{b}_2}\mathbf{Q}x^{-\vec{b}_2'}$ has column degrees $[0, \ldots, 0]$, and so $x^{[\vec{b}_1, \vec{b}_2]}\mathbf{Q}'x^{-[\vec{b}_1, \vec{b}_2']}$ also has column degrees $[0,\ldots,0]$ and a full rank leading column coefficient matrix Q'. Putting these together, we see that $x^{\vec{s}}\mathbf{P}x^{-[\vec{b}_1,\vec{b}_2]}x^{[\vec{b}_1,\vec{b}_2]}\mathbf{Q}'x^{-[\vec{b}_1,\vec{b}_2']} = x^{\vec{s}}\mathbf{P}\mathbf{Q}'x^{-[\vec{b}_1,\vec{b}_2']}$ has column degrees $[0,\ldots,0]$ and a full rank leading column coefficient matrix PQ'. It follows that $x^{\vec{s}}\mathbf{P}\mathbf{Q}'$ has column degrees $[\vec{b}_1,\vec{b}_2']$, or equivalently, the $\vec{s}$-column degrees of $\mathbf{P}\mathbf{Q}'$ is $[\vec{b}_1,\vec{b}_2']$.

It remains to show that any $\mathbf{n}$ satisfying $\mathbf{F}\mathbf{n}=0$ must be a linear combination of the columns of $\mathbf{P}\mathbf{Q}'$. Since $\mathbf{n}\in\langle(\mathbf{F},\sigma)\rangle$, it is generated by the $(\mathbf{F},\sigma)$-basis $\mathbf{P}$, that is, $\mathbf{n}=\mathbf{P}\mathbf{a}$ with $\mathbf{a}=\mathbf{P}^{-1}\mathbf{n}\in\mathbb{K}[x]^n$. Also, $\mathbf{F}\mathbf{n}=0$ implies $\mathbf{F}\mathbf{P}\mathbf{a}=0$, hence $\mathbf{a}=\mathbf{Q}'\mathbf{b}$ for some vector $\mathbf{b}$ as $\mathbf{Q}'$ is a kernel basis of $\mathbf{F}\mathbf{P}$. We now have $\mathbf{n}=\mathbf{P}\mathbf{Q}'\mathbf{b}$ as required. $\square$

Example 5.9. Let us look at an example of computing kernel basis using Theorem 5.8. Let $\mathbf{F}$ be given by

$$\begin{bmatrix} x+x^2+x^3 & 1+x & 0 & 1+x \\ 1+x^2+x^3 & x+x^2+x^3 & x+x^2 & x^3 \end{bmatrix} \in \mathbb{Z}_2[x]^{2\times 4}.$$

Let $\sigma=3$, $\vec{s}=[3,3,3,3]$. We first compute a $(\mathbf{F},\sigma,\vec{s})$-basis

$$\mathbf{P}=\begin{bmatrix} 0 & 0 & x^2 & x \\ 1 & 0 & 0 & x^2 \\ 1 & x^2 & x+x^2 & 1+x \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

with the $\vec{s}$-column degrees $\vec{b}=[3,5,5,5]$ and the residual

$$\mathbf{F}\mathbf{P}=\begin{bmatrix} 0 & 0 & x^3+x^4+x^5 & x^4 \\ 0 & x^3+x^4 & x^5 & x^3+x^5 \end{bmatrix}.$$

Thus $\mathbf{P}_1=[0,1,1,1]^T$, with $\vec{s}$-column degree 3, is the only kernel basis element computed. Let $\mathbf{P}_2$ contain the remaining columns of $\mathbf{P}$ and $\vec{b}_2 = [5,5,5]$ be its $\vec{s}$-column degrees. Next we compute a $\vec{b}_2$-minimal kernel basis of $\mathbf{F}\mathbf{P}_2$

$$\mathbf{Q} = [1+x+x^4,\ x+x^2,\ 1+x^3]^T$$

which has $\vec{b}_2$-column degree 9. Then

$$[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}] = \begin{bmatrix} 0 & x+x^3 \\ 1 & x^2+x^5 \\ 1 & 1+x+x^6 \\ 1 & 0 \end{bmatrix}$$

is a complete $\vec{s}$-minimal kernel basis of $\mathbf{F}$ with $\vec{s}$-column degrees $[3,9]$.

Theorem 5.8 shows that the remaining $\vec{s}$-minimal kernel basis elements $\mathbf{P}_2\mathbf{Q}$ can be correctly computed from the residual $\mathbf{F}\mathbf{P}_2$. Before discussing the computation of a $\vec{b}_2$-minimal kernel basis $\mathbf{Q}$ of $\mathbf{F}\mathbf{P}_2$, let us first note that the multiplication $\mathbf{P}_2\mathbf{Q}$ can be done efficiently, which again follows from Theorem 5.6.

Lemma 5.10. *The matrices $\mathbf{P}_2$ and $\mathbf{Q}$ can be multiplied with a cost of $O^{\sim}(nm^{\omega-1}t)$.*

Proof. Note that the dimension of $\mathbf{P}_2$ is $n \times O(m)$ from Theorem 5.5 and the dimension of $\mathbf{Q}$ is $O(m) \times O(m)$. The column degrees of $\mathbf{P}_2$ are bounded by the $\vec{s}$-column degrees $\vec{b}_2$ of $\mathbf{P}_2$ since $\vec{s}$ is non-negative. Also recall that $\sum \vec{b}_2 \le 4\xi$ from the proof of Corollary 5.7. By Lemma 2.5 the column degrees of $\mathbf{F}\mathbf{P}_2$ are bounded by the $\vec{s}$-column degrees $\vec{b}_2$ of $\mathbf{P}_2$. By Theorem 5.2, the sum of the $\vec{b}_2$-column degrees of $\mathbf{Q}$ is also bounded by $\sum \vec{b}_2 \le 4\xi$. Now if we separate $\mathbf{P}_2$ to n/m blocks rows each with no more than m rows, Theorem 5.6 can be used to multiply each block row with $\mathbf{Q}$. Each multiplication involves matrices of dimension $O(m) \times O(m)$. In addition, both the sum of the column degrees of $\mathbf{P}_2$ and the sum of the $\vec{b}_2$-column degrees of $\mathbf{Q}$ are bounded by 4ξ. So each multiplication costs $O^{\sim}(m^{\omega}t)$, where $t = \xi/m$. Hence doing this for all n/m block rows costs $O^{\sim}\left(nm^{\omega-1}t\right)$. □

5.1.3 Reducing the degrees

Our next task is computing a $\vec{b}_2$-minimal kernel basis of the residual $\mathbf{FP}_2$. It is useful to note that the lower degree terms of $\mathbf{FP}_2$ are zero since it has order σ. Hence we can use $\mathbf{G} = \mathbf{FP}_2/x^{\sigma}$ instead to compute the remaining basis elements. In the following, we show that just like the original input matrix $\mathbf{F}$, this new input matrix $\mathbf{G}$ has column degrees bounded by the corresponding entries of $\vec{s}$.

Lemma 5.11. *If an $(\mathbf{F}, \sigma, \vec{s})$-basis has columns arranged in increasing $\vec{s}$-column degrees with $\vec{s}$-column degrees $\vec{b}$, then the entries of $\vec{b} - [\sigma, \ldots, \sigma] = [b_1 - \sigma, \ldots, b_n - \sigma]$ are bounded component-wise by $\vec{s}$.*

Proof. A $(\mathbf{F}, 0, \vec{s})$-basis of order 0 has $\vec{s}$-column degrees given by $\vec{s}$. For each order increase, any column of the basis has its $\vec{s}$-column degree increases by at most one, which occurs when its order is increased by multiplying the column by x. Hence at order σ, the $\vec{s}$-column degree increase for each column is at most σ. □

Corollary 5.12. *The column degrees of $\mathbf{FP}/x^{\sigma}$ are bounded component-wise by $\vec{s}$.*

Proof. From Lemma 2.5, the column degrees of $\mathbf{FP}$ are bounded component-wise by $\vec{b}$, the $\vec{s}$-column degrees of $\mathbf{P}$. Hence the column degrees of $\mathbf{FP}/x^{\sigma}$ are bounded component-wise by $\vec{b} - [\sigma, \ldots, \sigma]$. The result then follows from Lemma 5.11. □

From Corollary 5.12, the column degrees of $\mathbf{FP}_2/x^{\sigma}$ are bounded by the entries of the corresponding subset $\vec{t}$ of $\vec{b} - [\sigma, \ldots, \sigma]$, which is in turn bounded by the entries of the corresponding subset of $\vec{s}$.

Example 5.13. From Example 5.9, note that instead of using the residual

$$\mathbf{FP}_2 = \begin{bmatrix} 0 & x^3+x^4+x^5 & x^4 \\ x^3+x^4 & x^5 & x^3+x^5 \end{bmatrix}$$

to compute a $[5,5,5]$-minimal kernel basis of $\mathbf{F}$, we can instead use

$$\mathbf{G} = \mathbf{FP}_2/x^3 = \begin{bmatrix} 0 & 1+x+x^2 & x \\ 1+x & x^2 & 1+x^2 \end{bmatrix}$$

to compute a $[2,2,2]$-minimal kernel basis of $\mathbf{G}$. The column degrees of $\mathbf{G}$ are bounded by the new shift $[2,2,2]$, which is in turn bounded by the corresponding entries $[3,3,3]$ of $\vec{s}$.

At this point, using Theorem 5.8 and Corollary 5.12, the problem is reduced to computing a $\vec{t}$-minimal kernel basis of $\mathbf{G} = \mathbf{FP}_2/x^{3s}$, which still has row dimension m. But its column dimension is now bounded by $3m/2$. Also notice that as in the original problem, the column degrees of the new input matrix $\mathbf{G}$ are bounded by the corresponding entries of the new shift $\vec{t}$. In addition, as the new shift $\vec{t}$ is bounded component-wise by a subset of the old shift $\vec{s}$, the new problem is no more difficult than the original problem.

5.1.4 Reducing the row dimension

We now turn to the new problem of computing a $\vec{t}$-minimal kernel basis of $\mathbf{G}$. Let

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix}$$

with $\mathbf{G}_1$ having $\lfloor m/2 \rfloor$ rows and $\mathbf{G}_2$ having $\lceil m/2 \rceil$ rows. If we compute a $\vec{t}$-minimal kernel basis $\mathbf{N}_1$ of $\mathbf{G}_1$, where $\mathbf{N}_1$ has $\vec{t}$-column degrees $\vec{u}$, then compute a $\vec{u}$-minimal kernel basis $\mathbf{N}_2$ of $\mathbf{G}_2\mathbf{N}_1$, then the next theorem shows that $\mathbf{N}_1\mathbf{N}_2$ is a $\vec{t}$-minimal kernel basis of $\mathbf{G}$.

Theorem 5.14. *Let $\mathbf{G} = \left[\mathbf{G}_1^T, \mathbf{G}_2^T\right]^T \in \mathbb{K}\left[x\right]^{m\times n}$ and $\vec{t} \in \mathbb{Z}^n$ a shift vector. If $\mathbf{N}_1$ is a $\vec{t}$-minimal kernel basis of $\mathbf{G}_1$ with $\vec{t}$-column degrees $\vec{u}$, and $\mathbf{N}_2$ is a $\vec{u}$-minimal kernel basis of $\mathbf{G}_2\mathbf{N}_1$ with $\vec{u}$-column degrees $\vec{v}$, then $\mathbf{N}_1\mathbf{N}_2$ is a $\vec{t}$-minimal kernel basis of $\mathbf{G}$ with $\vec{t}$-column degrees $\vec{v}$.*

Proof. The proof is very similar to the proof of Theorem 5.8. It is clear that $\mathbf{G}\mathbf{N}_1\mathbf{N}_2 = 0$ hence $\mathbf{N}_1\mathbf{N}_2$ is a kernel basis of $\mathbf{G}$. We now show that $\mathbf{N}_1\mathbf{N}_2$ is $\vec{t}$-column reduced and has $\vec{t}$-column degrees $\vec{v}$, or equivalently, $x^{\vec{t}}\mathbf{N}_1\mathbf{N}_2$ is column reduced. Notice that $x^{\vec{t}}\mathbf{N}_1$ has column degrees $\vec{u}$ and a full rank leading column coefficient matrix N_1. Hence $x^{\vec{t}}\mathbf{N}_1x^{-\vec{u}}$ has column degrees $[0,\ldots,0]$. Again, if one is concerned about the entries not being polynomials, one can simply multiply the matrix by x^{ξ} to shift the degrees up. Similarly, $x^{\vec{u}}\mathbf{N}_2x^{\vec{v}}$ has column degrees $[0,\ldots,0]$ and a full rank leading column coefficient matrix N_2. Putting them together, $x^{\vec{t}}\mathbf{N}_1x^{-\vec{u}}x^{\vec{u}}\mathbf{N}_2x^{-\vec{v}} = x^{\vec{t}}\mathbf{N}_1\mathbf{N}_2x^{-\vec{v}}$ has column degrees $[0,\ldots,0]$ and a full rank leading column coefficient matrix N_1N_2. It follows that $x^{\vec{t}}\mathbf{N}_1\mathbf{N}_2$ has column degrees $\vec{v}$, or equivalently, the $\vec{t}$-column degrees of $\mathbf{N}_1\mathbf{N}_2$ is $\vec{v}$.

It remains to show that any $\mathbf{n}$ satisfying $\mathbf{G}\mathbf{n} = 0$ must be a linear combination of the columns of $\mathbf{N}_1\mathbf{N}_2$. First notice that $\mathbf{n} = \mathbf{N}_1\mathbf{a}$ for some polynomial vector $\mathbf{a}$ since $\mathbf{N}_1$ is a kernel basis of $\mathbf{G}_1$. Also, $\mathbf{G}\mathbf{n} = 0$ implies that $\mathbf{G}_2\mathbf{N}_1\mathbf{a} = 0$, hence $\mathbf{a} = \mathbf{N}_2\mathbf{b}$ for some vector $\mathbf{b}$ as $\mathbf{N}_2$ is a kernel basis of $\mathbf{G}_2\mathbf{N}_1$. We now have $\mathbf{n} = \mathbf{N}_1\mathbf{N}_2\mathbf{b}$ as required. $\square$

Example 5.15. Let us compute a $\vec{t}$-minimal kernel basis of

$$\mathbf{G} = \begin{bmatrix} 0 & 1+x+x^2 & x \\ 1+x & x^2 & 1+x^2 \end{bmatrix}$$

from Example 5.13, where $\vec{t} = [2, 2, 2]$. Then

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 1+x+x^2 & x \end{bmatrix} \text{and } \mathbf{G}_2 = \begin{bmatrix} 1+x & x^2 & 1+x^2 \end{bmatrix}.$$

We first compute a $\vec{t}$-minimal kernel basis $\mathbf{N}_1$ of $\mathbf{G}_1$:

$$\mathbf{N}_1 = \begin{bmatrix} 1 & 0 \\ 0 & x \\ 0 & 1+x+x^2 \end{bmatrix}$$

with its $\vec{t}$-column degrees $\vec{u} = [2, 4]$. Next, we compute a $\vec{u}$-minimal kernel basis $\mathbf{N}_2$ of $\mathbf{G}_2\mathbf{N}_1 = \begin{bmatrix} 1+x & 1+x+x^4 \end{bmatrix}$:

$$\mathbf{N}_2 = [1+x+x^4,\ 1+x]^T.$$

Then

$$\mathbf{N}_1\mathbf{N}_2 = [1+x+x^4,\ x+x^2,\ 1+x^3]^T$$

is a $\vec{t}$-minimal kernel basis of $\mathbf{G}$.

While Theorem 5.8 allows us to compute kernel bases by columns, which then reduces the column dimensions, Theorem 5.14 shows that that the kernel bases can also be computed by rows, which then reduces the row dimensions. Again, we need to check that these computations can be done efficiently. In the following, Lemma 5.16 and Lemma 5.17 show that the multiplication $\mathbf{G}_2\mathbf{N}_1$ and the multiplication $\mathbf{N}_1\mathbf{N}_2$ can be done efficiently, which are again consequences of Theorem 5.6. Note that $t = \xi/m$ is a bound on the average of the entries of $\vec{t}$.

Lemma 5.16. *The multiplication of $\mathbf{G}_2$ and $\mathbf{N}_1$ can be done with a cost of $O^\sim(m^\omega t)$.*

Proof. Theorem 5.6 applies directly here. □

Lemma 5.17. *The multiplication of* $\mathbf{N}_1$ *and* $\mathbf{N}_2$ *can be done with a cost of* $O^{\sim}(m^{\omega}t)$.

Proof. Theorem 5.6 applies because the sum of the column degrees of $\mathbf{N}_1$ is bounded by the sum of the $\vec{t}$-column degrees of $\mathbf{N}_1$, which is $\sum \vec{u} \leq \xi$, and by Theorem 5.2 the sum of $\vec{u}$-column degrees of $\mathbf{N}_2$ is also bounded by ξ. □

5.1.5 Recursive computation

The computation of $\mathbf{N}_1$ and $\mathbf{N}_2$ is identical to the original problem, only the dimension has decreased. For computing $\mathbf{N}_1$, the dimension of the input matrix $\mathbf{G}_1$ is bounded by $\lfloor m/2 \rfloor \times (3m/2)$. For computing $\mathbf{N}_2$, the dimension of input matrix $\mathbf{G}_2\mathbf{N}_1$ is bounded by $\lceil m/2 \rceil \times (3m/2)$. The column degrees of $\mathbf{G}_1$ are bounded by the entries of $\vec{t}$, with $\sum \vec{t} \leq \xi$. Similarly, the column degrees of $\mathbf{G}_2\mathbf{N}_1$ are bounded by the entries of $\vec{u}$, with $\sum \vec{u} \leq \xi$. Hence, the same computation process can be repeated on these two smaller problems. This gives a recursive algorithm, shown in Algorithm 11.1.

Before analyzing the computational complexity of Algorithm 11.1 in the following section, we provide a proof of Theorem 5.6, which is needed to efficiently multiply matrices with unbalanced degrees in the algorithm.

5.1.6 Proof of Theorem 5.6

In this subsection we give a proof of Theorem 5.6.

Proof. Recall that $\vec{s}$ is a shift with entries ordered in terms of increasing values and ξ is a bound on the sum of the entries of $\vec{s}$. We wish to determine the cost of multiplying the two polynomials matrices $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ and $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$ where $\mathbf{A}$ has column degrees bounded by $\vec{s}$ and where $\mathbf{B}$'s column dimension $k \in O(m)$ and the sum θ of its $\vec{s}$-column degrees satisfies $\theta \in O(\xi)$. The goal is to show that these polynomial matrices can be multiplied with a cost of $O^{\sim}(nm^{\omega-1}t)$, where $t = \xi/m$.

Algorithm 5.1 minimalKernelBasis $(\mathbf{F}, \vec{s})$

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$, $\vec{s} = [s_1, \dots, s_n] \in \mathbb{Z}^n$ with entries arranged in non-decreasing
    order and bounding the corresponding column degrees of $\mathbf{F}$.
Output: A $\vec{s}$-minimal kernel basis of $\mathbf{F}$.
 1: $\xi := \sum_{i=1}^{n} s_i$; $\rho := \sum_{i=n-m+1}^{n} s_i$; $s := \rho/m$;
 2: $\left[\mathbf{P}, \vec{b}\right]$ := orderBasis$(\mathbf{F}, 3s, \vec{s})$, a $(\mathbf{F}, 3s, \vec{s})$-basis with the columns of $\mathbf{P}$ and
    the entries of is $\vec{s}$-column degrees $\vec{b}$ arranged so that the entries of $\vec{b}$ are in
    non-decreasing order;
 3: $[\mathbf{P}_1, \mathbf{P}_2] := \mathbf{P}$ where $\mathbf{P}_1$ consists of all columns $\mathbf{p}$ of $\mathbf{P}$ satisfying $\mathbf{F}\mathbf{p} = 0$;
 4: if $m = 1$ then
 5:     return $\mathbf{P}_1$
 6: else
 7:     $\vec{t} := \deg_{\vec{s}} \mathbf{P}_2 - [3s, 3s, \dots, 3s]$;
 8:     $\mathbf{G} := \mathbf{F}\mathbf{P}_2 / x^{3s}$;
 9:     $\left[\mathbf{G}_1^T, \mathbf{G}_2^T\right]^T := \mathbf{G}$, with $\mathbf{G}_1$ having $\lfloor m/2 \rfloor$ rows and $\mathbf{G}_2$ having $\lceil m/2 \rceil$ rows;
10:     $\mathbf{N}_1$ := minimalKernelBasis $\left(\mathbf{G}_1, \vec{t}\right)$;
11:     $\mathbf{N}_2$ := minimalKernelBasis $\left(\mathbf{G}_2\mathbf{N}_1, \mathrm{cdeg}_{\vec{t}}\mathbf{N}_1\right)$;
12:     $\mathbf{Q} := \mathbf{N}_1\mathbf{N}_2$;
13:     return $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}]$
14: end if
```

For simplicity we assume m is a power of 2, something which can be achieved by appending zero rows to $\mathbf{F}$. We divide the matrix $\mathbf{B}$ into $\log m$ column blocks according to the $\vec{s}$-column degrees of its columns. Let $t = \xi/m = ns/m$ and

$$\mathbf{B} = \left[\begin{array}{ccccc} \mathbf{B}^{(\log m)} & \mathbf{B}^{(\log m - 1)} & \cdots & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} \end{array} \right],$$

with $\mathbf{B}^{(\log m)}$, $\mathbf{B}^{(\log m-1)}$, $\mathbf{B}^{(\log m-2)}$, ... , $\mathbf{B}^{(2)}$, $\mathbf{B}^{(1)}$ having $\vec{s}$-column degrees in the range $[0, 2t]$, $(2t, 4t]$, $(4t, 8t]$, ...,$(tm/4, tm/2]$, $(tm/2, \theta]$, respectively. We will multiply $\mathbf{A}$ with each $\mathbf{B}^{(i)}$ separately.

We also divide the matrix $\mathbf{A}$ into $\log m$ column blocks and each matrix $\mathbf{B}^{(i)}$ into

$\log m$ row blocks according to the size of the corresponding entries in $\vec{s}$. Set

$$\begin{aligned}
\vec{s} &= \begin{bmatrix} \vec{s}_{\log m} & \vec{s}_{\log m-1} & \cdots & \vec{s}_1 \end{bmatrix} \\
\mathbf{A} &= \begin{bmatrix} \mathbf{A}_{\log m} & \mathbf{A}_{\log m-1} & \cdots & \mathbf{A}_1 \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} \mathbf{B}^{(\log m)} & \mathbf{B}^{(\log m-1)} & \cdots & \mathbf{B}^{(1)} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{B}^{(\log m)}_{\log m} & \mathbf{B}^{(\log m-1)}_{\log m} & \cdots & \mathbf{B}^{(1)}_{\log m} \\ \vdots & & & \vdots \\ \mathbf{B}^{(\log m)}_{1} & \mathbf{B}^{(\log m-1)}_{1} & \cdots & \mathbf{B}^{(1)}_{1} \end{bmatrix}
\end{aligned}$$

with $\vec{s}_{\log m}, \vec{s}_{\log m-1}, \ldots, \vec{s}_1$ having entries in the range $[0, 2t]$, $(2t, 4t]$, $(4t, 8t]$, ..., $(tm/2, tm]$ respectively. Also the column dimension of $\mathbf{A}_j$ and the row dimension of $\mathbf{B}^{(i)}_j$ match that of $\vec{s}_j$ for j from 1 to $\log m$.

Notice that $\mathbf{B}^{(i)}_{(j)}$ for $i > j$ must be zero. Otherwise, as $\vec{s}_j > tm/2^j \ge tm/2^{i-1}$, the $\vec{s}$-column degree of $\mathbf{B}^{(i)}$ would exceed $tm/2^{i-1}$, a contradiction since by definition the $\vec{s}$-column degree of $\mathbf{B}^{(i)}$ is bounded by $tm/2^{i-1}$ when $i > 1$. So $\mathbf{B}$ in fact has a block triangular shape

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{(\log m)}_{\log m} & \mathbf{B}^{(\log m-1)}_{\log m} & \cdots & \mathbf{B}^{(1)}_{\log m} \\ & \mathbf{B}^{(\log m-1)}_{\log m-1} & & \vdots \\ & & \ddots & \\ & & & \mathbf{B}^{(1)}_{1} \end{bmatrix}$$

(while remembering that the blocks have varying sizes).

First consider the multiplication

$$\mathbf{A}\mathbf{B}^{(1)} = \begin{bmatrix} \mathbf{A}_{\log m} & \cdots & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{B}^{(1)}_{\log m} \\ \vdots \\ \mathbf{B}^{(1)}_{1} \end{bmatrix}.$$

Note that there are $O(1)$ columns in $\mathbf{B}^{(1)}$ since $\theta \in O(\xi)$. We do this in $\log m$ steps. At step j for j from 1 to $\log m$ we multiply $\mathbf{A}_j$ and $\mathbf{B}_j^{(1)}$. The column dimension of $\mathbf{A}_j$, which is the same as the row dimension of $\mathbf{B}_j^{(1)}$, is less than 2^j. The degree of $\mathbf{B}_j^{(1)}$ is $O(\xi)$. To use fast multiplication, we expand $\mathbf{B}_j^{(1)}$ to a matrix $\bar{\mathbf{B}}_j^{(1)}$ with degree less than $\delta \in \Theta(tm/2^j)$ and column dimension $q \in O(2^j)$ as follows. Write

$$\mathbf{B}_j^{(1)} = \mathbf{B}_{j,0}^{(1)} + \mathbf{B}_{j,1}^{(1)} x^{\delta} + \cdots + \mathbf{B}_{j,q-1}^{(1)} x^{\delta(q-1)} = \sum_{k=0}^{q-1} \mathbf{B}_{j,k}^{(1)} x^{\delta k}$$

with each $\mathbf{B}_{j,k}^{(1)}$ having degree less than δ. Set

$$\bar{\mathbf{B}}_j^{(1)} = \left[\mathbf{B}_{j,0}^{(1)}, \mathbf{B}_{j,1}^{(1)}, \ldots, \mathbf{B}_{j,q-1}^{(1)}\right].$$

We can then multiply $\mathbf{A}_j$, which has dimension $m \times O(2^j)$ for $j < \log m$, and $\bar{\mathbf{B}}_j^{(1)}$, which has dimension $O(2^j) \times O(2^j)$ for $j < \log m$, with a cost of

$$\begin{aligned} O^{\sim}\left((m/2^j)\left(2^j\right)^{\omega} tm/2^j\right) &= O^{\sim}\left(\left(2^j\right)^{\omega-2} m^2 t\right) \\ &\subset O^{\sim}\left(m^{\omega} t\right). \end{aligned}$$

For $j = \log m$, $\mathbf{A}_j$ has dimension $m \times O(n)$, $\bar{\mathbf{B}}_j^{(1)}$ has dimension $O(n) \times O(m)$, and their degrees are $O(t)$. Hence they can be multiplied with a cost of $O^{\sim}((n/m)m^{\omega}t) = O^{\sim}(nm^{\omega-1}t)$. Adding up the columns of $\mathbf{A}_j\bar{\mathbf{B}}_j^{(1)}$ gives $\mathbf{A}_j\mathbf{B}_j^{(1)}$ and costs $O(m^2t)$. Therefore, multiplying $\mathbf{A}$ and $\mathbf{B}^{(1)}$ over $\log(m)$ steps costs

$$O^{\sim}\left(m^{\omega} t\right) + O^{\sim}\left(nm^{\omega-1} t\right) + O(m^2 t) = O^{\sim}\left(nm^{\omega-1} t\right).$$

Next we multiply $\mathbf{A}$ with $\mathbf{B}^{(2)}$. We proceed in the same way as before, but notice that $\mathbf{A}_1\mathbf{B}_1^{(2)}$ is no longer needed since $\mathbf{B}_1^{(2)} = 0$. Multiplying $\mathbf{A}$ and $\mathbf{B}^{(2)}$ also costs $O^{\sim}\left(nm^{\omega-1}t\right)$.

Continuing to doing this, gives a costs of $O^{\sim}\left(nm^{\omega-1}t\right)$ to multiply $\mathbf{A}$ with the columns $\mathbf{B}^{(i)}$ for i from 1 to $\log m$. As before, we recall that $\mathbf{B}^{(i)}_{(j)} = 0$ for $j > i$. The overall cost for i from 1 to $\log m$ is therefore $O^{\sim}\left(nm^{\omega-1}t\right)$ to multiply $\mathbf{A}$ and $\mathbf{B}$. $\square$

5.2 Computational Complexity

For the cost analysis we first consider the case where the column dimension n is not much bigger than the row dimension m.

Theorem 5.18. *If $n \in O(m)$, then the cost of Algorithm 11.1 is $O^{\sim}(m^{\omega}s)$ field operations.*

Proof. We may assume m is a power of 2, which can be achieved by appending zero rows to $\mathbf{F}$. The order basis computation at line 2 costs $O^{\sim}(n^{\omega}s) = O^{\sim}(m^{\omega}s)$. The multiplications at line 8 and line 13 cost $O^{\sim}\left(nm^{\omega-1}t\right) = O^{\sim}(m^{\omega}s)$. The remaining operations including multiplications at line 11 and line 12 cost $O^{\sim}(m^{\omega}t) = O^{\sim}(m^{\omega}s)$. Let $g(m)$ be the computational cost of the original problem. Then we have the recurrence relation

$$g(m) \in O^{\sim}(m^{\omega}s) + g(m/2) + g(m/2),$$

with the base case $g(1) \in O^{\sim}(s)$, the cost of just an order basis computation at $m = 1$. This gives $g(m) \in O^{\sim}(m^{\omega}s)$ field operations as the cost of the algorithm. $\square$

We now consider the general case where the column dimension n can be much bigger than the row dimension m.

Theorem 5.19. *Algorithm 11.1 costs $O^{\sim}(n^{\omega}s)$ field operations in general.*

Proof. The order basis computation at line 2 costs $O^{\sim}(n^{\omega}s)$ in general, which dominates the cost of other operations. The problem is then reduced to one where we have column dimension $O(m)$, which is handled by Theorem 5.18 with a cost of $O^{\sim}(m^{\omega}t) \subset O^{\sim}(n^{\omega}s)$, where $t = \xi/m \leq ns/m$. □

When we have the important special case where the shift $\vec{s} = [s, \ldots, s]$ is uniform then Algorithm 11.1 has a lower cost. Indeed we notice that the order basis computation at line 2 costs $O^{\sim}(n^{\omega-1}ms)$ using the algorithm from Chapter 3. In addition, the multiplication of $\mathbf{F}$ and $\mathbf{P}_2$ at line 8 and the multiplication of $\mathbf{P}_2$ and $\mathbf{Q}$ at line 13 both cost $O^{\sim}(nm^{\omega-1}s)$ as shown below in Lemma 5.20 and Lemma 5.21.

Lemma 5.20. *If the degree of* $\mathbf{F}$ *is bounded by* s*, then the multiplication of* $\mathbf{F}$ *and* $\mathbf{P}_2$ *at line 8 costs* $O^{\sim}(nm^{\omega-1}s)$*.*

Proof. Since $\mathbf{P}_2$ is a part of a $(\mathbf{F}, 3s, \vec{s})$-basis, its degree is bounded by $3s$. It has dimension $n \times O(m)$ from Theorem 5.5. Multiplying $\mathbf{F}$ and $\mathbf{P}_2$ therefore costs $(n/m)O^{\sim}(m^{\omega}s) = O^{\sim}(nm^{\omega-1}s)$. □

Lemma 5.21. *If* $\mathbf{F}$ *has degree* s*, then the multiplication of* $\mathbf{P}_2$ *and* $\mathbf{Q}$ *at line 13 costs* $O^{\sim}(nm^{\omega-1}s)$*.*

Proof. First note that the dimension of $\mathbf{Q}$ is $O(m) \times O(m)$ since it is a $\vec{t}$-minimal kernel basis of $\mathbf{G} = \mathbf{F}\mathbf{P}_2/x^{3s}$, which has dimension $m \times O(m)$. In addition, by Theorem 5.2, the sum of the $\vec{t}$-column degrees of $\mathbf{Q}$ is bounded by $\sum \vec{t}$, which is bounded by $O(ms)$ since $\vec{t}$ has $O(m)$ entries all bounded by s.

Now Theorem 5.6 and its proof still work. The current situation is even simpler as we do not need to subdivide the columns of $\mathbf{P}_2$, which has degree bounded by $3s$ and dimension $n \times O(m)$. We just need to separate the columns of $\mathbf{Q}$ to $O(\log m)$ groups with degree ranges $[0, 2s]$, $(2s, 4s]$, $(4s, 8s]$, $\ldots$, and multiply $\mathbf{P}_2$ with each group in the same way as in Theorem 5.6, with each of these $O(\log m)$ multiplications costs $(n/m)O^{\sim}(m^{\omega}s) = O^{\sim}(nm^{\omega-1}s)$. □

Theorem 5.22. *If* $\vec{s} = [s, \ldots, s]$ *is uniform, then Algorithm 11.1 costs* $O^{\sim}(n^{\omega-1}ms)$*.*

Proof. After the initial order basis computation, which costs $O^{\sim}\left(n^{\omega-1}ms\right)$, and the multiplication of $\mathbf{F}$ and $\mathbf{P}_2$, which costs $O^{\sim}\left(nm^{\omega-1}s\right)$ from Lemma 5.20, the column dimension is reduced to $O\left(m\right)$, allowing Theorem 5.18 to apply for computing a $\vec{t}$-minimal kernel basis of $\mathbf{F}\mathbf{P}_2/x^{3s}$. Hence the remaining work costs $O^{\sim}\left(m^{\omega}s\right)$. The overall cost is therefore dominated by the cost $O^{\sim}\left(n^{\omega-1}ms\right)$ of the initial order basis computation. □

Corollary 5.23. *If the input matrix* $\mathbf{F}$ *has degree* d*, then a minimal kernel basis of* $\mathbf{F}$ *can be computed with a cost of* $O^{\sim}\left(n^{\omega-1}md\right)$*.*

Proof. We can just set the shift $\vec{s}$ to $[d, \ldots, d]$ and apply Theorem 5.22. □

Chapter 6

Matrix inverse

In this chapter, we consider the problem of computing the inverse of a $n \times n$ polynomial matrix with degree d. Jeannerod and Villard [2005] gave a deterministic algorithm for this problem that costs $O^{\sim}(n^3 d)$ field operations. But their algorithm only works well on input matrices that are generic with dimension a power of 2. Storjohann [2010] gave another algorithm that works for general input matrices and with a similar cost, but the algorithm is randomized Las Vegas. In the following, we show that Jeannerod and Villard's algorithm can be improved to handle any matrix with a cost of $O^{\sim}(n^3 d)$ using new results from this thesis. The algorithm given here is still deterministic. If ξ is the minimum of the sum of the column degrees and the sum of the row degrees of the input matrix and $s = \xi / n$ is the average, then the inverse can be computed with $O^{\sim}(n^3 s)$. Note that Gupta et al. [2012] has also provided a method that can be used by some existing algorithms for polynomial matrix computation problems including matrix inverse, to obtain a cost stated in terms of the average column degrees. In the following, we assume without loss of generality that the sum of the column degrees is the minimum sum.

Algorithm 6.1 is a recursive version of the algorithm from Jeannerod and Villard [2005], except that we replace the kernel basis computation at line 4 and the matrix

Algorithm 6.1 inverse$(\mathbf{F}, \vec{s})$

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$; $\vec{s}$ is initially set to the column degrees of $\mathbf{F}$. It keeps track
    of the degrees.
Output: $\mathcal{A} = [\mathbf{A}_1, \dots, \mathbf{A}_{\lceil \log n \rceil}], \mathbf{B}$ with $\mathbf{A}_1, \dots, \mathbf{A}_{\lceil \log n \rceil}, \mathbf{B} \in \mathbb{K}[x]^{n\times n}$ such that
    $\mathbf{A}_1 \dots \mathbf{A}_{\lceil \log n \rceil}\mathbf{B}^{-1} = \mathbf{F}^{-1}$ if $\mathbf{F}$ is nonsingular, or fail if $\mathbf{F}$ is singular.
1: $[\mathbf{F}_1^T, \mathbf{F}_2^T] := \mathbf{F}^T$ with $\mathbf{F}_1$ consists of the top $\lceil n/2 \rceil$ rows of $\mathbf{F}$;
2: if $\mathbf{F} = 0$ then fail endif;
3: if $n = 1$ then return $\{1, \mathbf{F}\}$; endif;
4: $\mathbf{N}_1 :=$ minimalKernelBasis $(\mathbf{F}_1, \vec{s})$;$\mathbf{N}_2 :=$ minimalKernelBasis $(\mathbf{F}_2, \vec{s})$;
5: if columnDimension($\mathbf{N}_1$) $\neq \lfloor n/2 \rfloor$ or columnDimension($\mathbf{N}_2$) $\neq \lceil n/2 \rceil$ then
    fail; endif;
6: $\mathbf{R}_1 := \mathbf{F}_1\mathbf{N}_2$;$\mathbf{R}_2 := \mathbf{F}_2\mathbf{N}_1$;
7: $\{\mathcal{A}^{(1)}, \mathbf{H}_1\} :=$ inverse($\mathbf{R}_1$, cdeg$_{\vec{s}}\mathbf{N}_2$); $\{\mathcal{A}^{(2)}, \mathbf{H}_2\} :=$ inverse($\mathbf{R}_2$, cdeg$_{\vec{s}}\mathbf{N}_1$);
8: $\mathcal{A} := \left[[\mathbf{N}_2, \mathbf{N}_1], \mathrm{diag}(\mathcal{A}_1^{(1)}, \mathcal{A}_1^{(2)}), \dots, \mathrm{diag}(\mathcal{A}_{\lceil \log n \rceil - 1}^{(1)}, \mathcal{A}_{\lceil \log n \rceil - 1}^{(2)})\right]$
9: return $\{\mathcal{A}, \mathrm{diag}([\mathbf{H}_1, \mathbf{H}_2])\}$;
```

multiplications at line 6 with the new algorithms from this thesis. The algorithm also returns a list of matrices $\mathbf{A}_1, \dots, \mathbf{A}_{\lceil \log n \rceil}, \mathbf{B}$ satisfying $\mathbf{A}_1 \dots \mathbf{A}_{\lceil \log n \rceil}\mathbf{B}^{-1} = \mathbf{F}^{-1}$, instead of just two matrices $\mathbf{A}, \mathbf{B}$ satisfying $\mathbf{A}\mathbf{B}^{-1} = \mathbf{F}^{-1}$. We can then compute the product $\mathbf{A} = \mathbf{A}_1 \dots \mathbf{A}_{\lceil \log n \rceil}$ with a cost of $O^{\sim}(n^3 s)$. It is interesting to note that the output $\mathbf{A}_1, \dots, \mathbf{A}_{\lceil \log n \rceil}, \mathbf{B}$ takes only $O(n^2 s \log n)$ space, but the product $\mathbf{A} = \mathbf{A}_1 \dots \mathbf{A}_{\lceil \log n \rceil}$ takes $O(n^3 s)$ space.

Let us first look at the cost of the kernel basis computation and matrix multiplications, since they dominate the cost of Algorithm 6.1.

Lemma 6.1. *The kernel basis computation at line 4 costs $O^{\sim}(n^{\omega} s)$.*

Proof. Just use the earlier kernel basis algorithm with the shift set to the column degrees of the input matrix. □

Lemma 6.2. *The multiplications $\mathbf{R}_1 := \mathbf{F}_1\mathbf{N}_2$ and $\mathbf{R}_2 := \mathbf{F}_2\mathbf{N}_1$ at line 6 cost $O^{\sim}(n^{\omega} s)$.*

Proof. From Theorem 5.2 we know that the sum of the $\vec{s}$-column degrees of $\mathbf{N}_1$ and that of $\mathbf{N}_2$ are both bounded by ξ. Now Theorem 5.6 can be applied. □

Theorem 6.3. *Algorithm 6.1 costs $O^{\sim}(n^{\omega}s)$ field operations to compute an inverse of a nonsingular matrix $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$.*

Proof. If the sum of the row degrees is smaller, we can just transpose the matrix. Let the cost be $g(n)$. Then we have the following recurrence relation:

$$\begin{aligned} g(n) &\in O^{\sim}(n^{\omega}s) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor) \\ &\in O^{\sim}(n^{\omega}s) + 2g(\lceil n/2 \rceil) \\ &\in O^{\sim}(n^{\omega}s). \end{aligned}$$

Note that always rounding up $n/2$ to $\lceil n/2 \rceil$ is no worse than assuming n is a power of 2. In other words, the entries in the sequence $[\lceil n/2 \rceil, \lceil n/4 \rceil, \ldots, 1]$ is no larger than the corresponding entries in the sequence $[n'/2, n'/4, \ldots, 1]$, where n' is the smallest power of 2 that is no less than n, that is, $n' = 2^{\lceil \log_2 n \rceil}$. □

Lemma 6.4. *The multiplications $\mathbf{A} = \mathbf{A}_1 \ldots \mathbf{A}_{\lceil \log n \rceil}$ can be done with a cost of $O^{\sim}(n^3 s)$.*

Proof. Note that $\mathbf{A}_i$ for $i \le \log n$ has 2^i blocks on the diagonal. Each block of $\mathbf{A}_i$ is used to compute two corresponding blocks of $\mathbf{A}_{i+1}$. Let us first look at $\mathbf{A}_1 = [\mathbf{N}_2, \mathbf{N}_1]$ and

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{N}_2' & \mathbf{N}_1' & & \\ & & \mathbf{M}_2' & \mathbf{M}_1' \end{bmatrix},$$

where $\mathbf{N}_1'$, $\mathbf{N}_2'$ are the kernel bases of the submatrices $\mathbf{F}_1'$, $\mathbf{F}_2'$ contained in

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{F}_1' \\ \mathbf{F}_2' \end{bmatrix} = \mathbf{F}_1 \mathbf{N}_2.$$

When multiplying $\mathbf{A}_1$ and $\mathbf{A}_2$, the submatrix $\mathbf{N}_2$ of $\mathbf{A}_1$ is multiplied with the block $[\mathbf{N}_2', \mathbf{N}_1']$ in $\mathbf{A}_2$. Let $\vec{s}'$ be the list of the $\vec{s}$-column degrees of $\mathbf{N}_2$, where $\vec{s}$ is list of the column degrees of the input matrix $\mathbf{F}$. Then $\sum \vec{s}' \leq \sum \vec{s} = \xi$ by Theorem 5.2. From Lemma 2.5, we know the column degrees of $\mathbf{R}_1 = \mathbf{F}_1\mathbf{N}_2$ are bounded componentwise by the $\vec{s}$-column degrees $\vec{s}'$ of $\mathbf{N}_2$, hence the sum of the column degrees of $\mathbf{R}_1$ is also bounded by ξ. It follows that the sum of $\vec{s}'$-column degrees of $\mathbf{N}_1'$ and that of $\mathbf{N}_2'$ are each bounded by ξ. We can therefore use Theorem 5.6 to multiply $\mathbf{N}_2$ and $[\mathbf{N}_2', \mathbf{N}_1']$ with a cost of $O^{\sim}(n^{\omega}s)$. From Lemma 2.5, the $\vec{s}$-column degrees of the product $\mathbf{N}_2[\mathbf{N}_2', \mathbf{N}_1']$ are bounded by the $\vec{s}'$-column degrees of $[\mathbf{N}_2', \mathbf{N}_1']$, hence the sum of the $\vec{s}$-column degrees of each column block in $\mathbf{N}_2[\mathbf{N}_2', \mathbf{N}_1'] = [\mathbf{N}_2\mathbf{N}_2', \mathbf{N}_2\mathbf{N}_1']$ is still bounded by ξ. The multiplication involving $\mathbf{N}_1$ and the second block of $\mathbf{A}_2$ is done in the same way as the multiplication $\mathbf{N}_2[\mathbf{N}_2', \mathbf{N}_1']$, hence the multiplication $\mathbf{A}_1\mathbf{A}_2$ cost $O^{\sim}(n^{\omega}s)$, with the sum of $\vec{s}$-column degrees of each of the four column blocks in $\mathbf{A}_1\mathbf{A}_2 = [\mathbf{N}_2\mathbf{N}_2', \mathbf{N}_2\mathbf{N}_1', \mathbf{N}_1\mathbf{M}_2', \mathbf{N}_1\mathbf{M}_1']$ bounded by ξ.

Next, we multiply $\mathbf{A}_1\mathbf{A}_2$ with $\mathbf{A}_3$. The matrix $\mathbf{A}_3$ now has four blocks on the diagonal. Consider $\mathbf{N}_2\mathbf{N}_2'$, the first column block of $\mathbf{A}_1\mathbf{A}_2$, multiplied with the first block $[\mathbf{N}"_2, \mathbf{N}"_1]$ on the diagonal of $\mathbf{A}_3$. Let $\vec{s}"$ be the $\vec{s}'$-column degrees of $\mathbf{N}_2'$, which bound the $\vec{s}$-column degrees of $\mathbf{N}_2\mathbf{N}_2'$. Then $\sum \vec{s}" \leq \sum \vec{s}' \leq \sum \vec{s} = \xi$. Following the same reasoning as before, the sum of the $\vec{s}"$-column degrees of $\mathbf{N}"_2$ is still bounded by ξ. We can therefore again use Theorem 5.6 to multiply $\mathbf{N}_2\mathbf{N}_2'$ and $\mathbf{N}"_2$. The multiplication of the remaining blocks are done in the same way. The product $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$ now has 8 column blocks, with the sum of the $\vec{s}$-column degrees of each column block bounded by ξ.

Repeating this process, we multiply $\mathbf{A}_1 \cdots \mathbf{A}_i$ with $\mathbf{A}_{i+1}$ at step i for i from 1 to $\lfloor \log n \rfloor$. Each of the 2^i column blocks of $\mathbf{A}_1 \cdots \mathbf{A}_i$ has dimension $n \times O(n/2^i)$. Each of the $O(2^i)$ column blocks on the diagonal of $\mathbf{A}_{i+1}$ has dimension $O(n/2^i) \times O(n/2^i)$. (Big O notation is used here because $n/2^i$ may not be an integer.) Let $\vec{u}_j$ be the shift used to compute the jth block in $\mathbf{A}_{i+1}$. Then as before, the $\vec{s}$-column degrees of the jth column block in $\mathbf{A}_1 \cdots \mathbf{A}_i$ are bounded by $\vec{u}_j$, with $\sum \vec{u}_j \leq \xi$. The sum of the $\vec{u}$-

column degrees of the jth block in $\mathbf{A}_{i+1}$ is bounded by 2ξ. (Each of the left half and the right half has the sum bounded by ξ.) Therefore, multiplying $\mathbf{A}_1 \cdots \mathbf{A}_i$ with $\mathbf{A}_{i+1}$ cost $O^{\sim}\left(2^i 2^i \left(n/2^i\right)^{\omega-1} 2^i u\right) = O^{\sim}\left(\left(2^i\right)^{3-\omega} n^{\omega} s\right)$, where $u = \xi/2^i = ns/2^i$. Take $\omega = 3$, we get $O^{\sim}\left(n^3 s\right)$ as desired. □

Again, it is interesting to note that Algorithm 6.1 costs only $O^{\sim}\left(n^{\omega} s\right)$ and represents the inverse with $O\left(n^2 s \log n\right)$ space. It is possible that this representation is useful in some applications. For example, if we wish to multiply another low degree matrix or a row vector $\mathbf{H}$ by $\mathbf{F}^{-1}$, representing $\mathbf{F}^{-1} = \mathbf{A}\mathbf{B}^{-1}$ requires us to multiply $\mathbf{H}$ with a high degree matrix $\mathbf{A}$. This can be more expensive than the multiplication using the representation $\mathbf{F}^{-1} = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_{\lceil \log n \rceil} \mathbf{B}^{-1}$, then $\mathbf{H}\mathbf{F}^{-1} = \mathbf{H}\mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_{\lceil \log n \rceil} \mathbf{B}^{-1}$, which is less expensive. It may be interesting to look for other applications where this smaller representation is useful.

Chapter 7

Column Basis

Column bases are fundamental constructions in polynomial matrix algebra. As an example, when the row dimension is one (i.e. $m = 1$), then finding a column basis coincides with finding a greatest common divisor (GCD) of all the polynomials in the matrix. Similarly, the nonzero columns of column reduced forms, Popov normal forms, and Hermite normal forms are all column bases satisfying additional degree constraints. A column reduced form gives a special column basis whose column degrees are the smallest possible, while Popov and Hermite forms are special column reduced or shifted column reduced forms satisfying additional conditions that make them unique. Efficient column basis computation immediately leads to fast computation for such core procedures as determining matrix GCDs Beckermann and Labahn [2000], column reduced forms Beelen et al. [1988] and Popov forms Villard [1996] of $\mathbf{F}$ with any dimension and rank. Column basis computation also provides a deterministic alternative to randomized lattice compression Li [2006], Storjohann and Villard [2005].

While column bases are produced by column reduced, Popov and Hermite forms and considerable work has been done on computing such forms, for example Beckermann et al. [2006a], Beelen and Dooren [1988], Giorgi et al. [2003], Gupta et al.

[2012], Sarkar [2011], Sarkar and Storjohann [2011]. However most of these existing algorithms require that the input matrices be square nonsingular and so start with existing column bases. It is however pointed out in Sarkar [2011], Sarkar and Storjohann [2011] that randomization can be used to relax the square nonsingular requirement.

In this chapter, we consider the problem of computing a column basis of an input matrix $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ with $n \geq m$ and column degrees $\vec{s}$. we give a fast, deterministic algorithm for the computation of a column basis for $\mathbf{F}$ having complexity $O^{\sim}(nm^{\omega-1}s)$ field operations in $\mathbb{K}$ with s being the average average column degree of $\mathbf{F}$. To compute a column basis, we know from Corollary 2.16 that any matrix polynomial $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ can be unimodularly transformed to a column basis by repeatedly working with the leading column coefficient matrices. However this method of computing a column basis can be expensive. Indeed one needs to work with up to $\sum \vec{s}$ such coefficient matrices, which could involve up to $\sum \vec{s}$ polynomial matrix multiplications. Before discussing the efficient computation of column basis, it is useful to look at following relationship between column basis, kernel basis, and unimodular matrix.

Lemma 7.1. *Given $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$. If $\mathbf{U}$ is a unimodular matrix such that $\mathbf{FU} = [0, \mathbf{T}]$ gives a full column rank $\mathbf{T}$, then $\mathbf{U}$ can be separated into two submatrices $\mathbf{U} = [\mathbf{U}_L, \mathbf{U}_R]$, where $\mathbf{U}_L$ is a kernel basis of $\mathbf{F}$ and $\mathbf{FU}_R = \mathbf{T}$ is a column basis of $\mathbf{F}$. In addition, the kernel basis $\mathbf{U}_L$ can be replaced by any other kernel basis $\mathbf{N}$ of $\mathbf{F}$ and still gives a unimodular matrix $[\mathbf{N}, \mathbf{U}_R]$, which can also be used to unimodularly transform $\mathbf{F}$ to $[0, \mathbf{T}]$.*

Proof. Note that $\mathbf{T}$ is a column basis of $\mathbf{F}$ by Corollary 2.16. It remains to show that $\mathbf{U}_L$ is a kernel basis of $\mathbf{F}$. Since $\mathbf{FU}_L = 0$, $\mathbf{U}_L$ is generated by any kernel basis $\mathbf{N}$, that is, $\mathbf{U}_L = \mathbf{NC}$ for some polynomial matrix $\mathbf{C}$. Let r be the rank of $\mathbf{F}$, which is also the column dimension of $\mathbf{T}$ and $\mathbf{U}_R$. Then both $\mathbf{N}$ and $\mathbf{U}_L$ have column dimension $n-r$. Hence $\mathbf{C}$ is a square $(n-r)\times(n-r)$ matrix. Now the unimodular matrix $\mathbf{U}$ can be factored as

$$\mathbf{U} = [\mathbf{N}\mathbf{C}, \mathbf{U}_R] = [\mathbf{N}, \mathbf{U}_R]\begin{bmatrix}\mathbf{C} & \\ & I\end{bmatrix},$$

implying that both factors $[\mathbf{N}, \mathbf{U}_R]$ and $\begin{bmatrix}\mathbf{C} & \\ & I\end{bmatrix}$ are unimodular. Therefore, $\mathbf{C}$ is unimodular and $\mathbf{U}_L = \mathbf{N}\mathbf{C}$ is also a kernel basis. Notice that the unimodular matrix $[\mathbf{N}, \mathbf{U}_R]$ also transforms $\mathbf{F}$ to $[0, \mathbf{T}]$. $\square$

Lemma 7.1 gives the following result for a unimodular matrix and its inverse.

Corollary 7.2. *Let $\mathbf{U} = [\mathbf{U}_L, \mathbf{U}_R]$ be any unimodular matrix with columns separated arbitrarily to $\mathbf{U}_L$ and $\mathbf{U}_R$. Let its inverse $\mathbf{V} = \begin{bmatrix}\mathbf{V}_U \\ \mathbf{V}_D\end{bmatrix}$, where the row dimension of $\mathbf{V}_U$ matches the column dimension of $\mathbf{U}_L$. So we have*

$$\mathbf{V}\mathbf{U} = \begin{bmatrix}\mathbf{V}_U \\ \mathbf{V}_D\end{bmatrix}\begin{bmatrix}\mathbf{U}_L, \mathbf{U}_R\end{bmatrix} = \begin{bmatrix}\mathbf{V}_U\mathbf{U}_L & \mathbf{V}_U\mathbf{U}_R \\ \mathbf{V}_D\mathbf{U}_L & \mathbf{V}_D\mathbf{U}_R\end{bmatrix} = \begin{bmatrix}I & 0 \\ 0 & I\end{bmatrix}.$$

Then $\mathbf{V}_U\mathbf{U}_L = I$ is a column basis of $\mathbf{V}_U$ and a row basis of $\mathbf{U}_L$, while $\mathbf{V}_D\mathbf{U}_R = I$ is a column basis of $\mathbf{V}_D$ and a row basis of of $\mathbf{U}_R$. In addition, $\mathbf{V}_D$ and *$\mathbf{U}_L$* are kernel bases of each other, while $\mathbf{V}_U$ and $\mathbf{U}_R$ are kernel bases of each other.

Proof. This follows directly from Lemma 7.1, by taking $\mathbf{F}$ from Lemma 7.1 to be $\mathbf{V}_U$, $\mathbf{V}_D$, $\mathbf{U}_L^T$, and $\mathbf{U}_R^T$ here. $\square$

To compute a column basis of $\mathbf{F}$, we use the following procedure. We first compute a right kernel basis $\mathbf{N}$ of $\mathbf{F}$. Then we compute a left kernel basis $\mathbf{G}$ of $\mathbf{N}$. This matrix $\mathbf{G}$ is a right factor of $\mathbf{F}$, that is, $\mathbf{F} = \mathbf{T}\mathbf{G}$ for some $\mathbf{T} \in \mathbb{K}[x]^{m\times r}$. Then we can compute the left factor $\mathbf{T}$, which is in fact a column basis of $\mathbf{F}$.

Lemma 7.3. *Given* $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$*. Let* $\mathbf{N} \in \mathbb{K}[x]^{n\times(n-r)}$ *be any right kernel basis of* $\mathbf{F}$*, and* $\mathbf{G} \in \mathbb{K}[x]^{r\times n}$ *be any left kernel basis of* $\mathbf{N}$*, where* r *is the rank of* $\mathbf{F}$*. Then* $\mathbf{F} = \mathbf{T}\mathbf{G}$ *for some* $\mathbf{T} \in \mathbb{K}[x]^{m\times r}$ *and* $\mathbf{T}$ *is a column basis of* $\mathbf{F}$*.*

Proof. Let the matrix $\mathbf{U} = \begin{bmatrix}\mathbf{U}_L, \mathbf{U}_R\end{bmatrix}$ from Corollary 7.2 be a unimodular matrix that transforms $\mathbf{F}$ to a column basis $\mathbf{B} \in \mathbb{K}[x]^{m\times r}$ of $\mathbf{F}$, where $\mathbf{U}_L$ is any right kernel basis of $\mathbf{F}$. From $\mathbf{F}\mathbf{U} = [0, \mathbf{B}]$, we get $\mathbf{F} = [0, \mathbf{B}]\,\mathbf{U}^{-1} = \mathbf{B}\,[0, I]\,\mathbf{V} = \mathbf{B}\mathbf{V}_D$. Since $\mathbf{V}_D$ is a left kernel basis of $\mathbf{U}_L$ by Corollary 7.2, any other left kernel basis $\mathbf{G}$ of $\mathbf{U}_L$ is unimodularly equivalent to $\mathbf{V}_D$, that is, $\mathbf{V}_D = \mathbf{W}\mathbf{G}$ for some unimodular matrix $\mathbf{W}$. Now $\mathbf{F} = \mathbf{B}\mathbf{W}\mathbf{G}$, where $\mathbf{B}\mathbf{W} = \mathbf{T}$ a column basis of $\mathbf{F}$ since it is unimodularly equivalent to the column basis $\mathbf{B}$. □

Lemma 7.3 outlines a procedure for computing a column basis of $\mathbf{F}$ with three main steps. The first step is to compute a $(\mathbf{F}, \vec{s})$-kernel basis $\mathbf{N}$, which can be efficiently done using Algorithm 5.1. However, we still need to work on the second step of computing a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis $\mathbf{G}^T$ and the third step of computing the column basis $\mathbf{T}$ from $\mathbf{F}$ and $\mathbf{G}$. Note that while Lemma 7.3 does not require the bases computed to be minimal, working with minimal bases keeps the degrees well-managed and helps to make the computation efficient.

Example 7.4. Let

$$\mathbf{F} = \begin{bmatrix} x^2 & x^2 & x+x^2 & 1+x^2 \\ 1+x+x^2 & x^2 & 1+x^2 & 1+x^2 \end{bmatrix}.$$

Then the matrix

$$\mathbf{N} = \begin{bmatrix} x & 1 \\ 1 & x \\ x & 1 \\ 0 & x \end{bmatrix}$$

is a right kernel basis of $\mathbf{F}$ and the matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ x & x^2 & 0 & 1+x^2 \end{bmatrix}$$

is a left nulllspace basis of $\mathbf{N}$. Finally the matrix

$$\mathbf{T} = \begin{bmatrix} x+x^2 & 1 \\ 1+x^2 & 1 \end{bmatrix}$$

satisfies $\mathbf{F} = \mathbf{T}\mathbf{G}$, and is a column basis of $\mathbf{F}$.

7.1 Computing a Right Factor

Let us now look at the computation of a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis $\mathbf{G}^T$. For this problem, Algorithm 11.1 does not work well directly, since the input matrix $\mathbf{N}^T$ has nonuniform row degrees and negative shift. Comparing to the earlier problem of computing a $(\mathbf{F}, \vec{s})$-kernel basis $\mathbf{N}$, it is interesting to note that the old output $\mathbf{N}$ now becomes the new input matrix $\mathbf{N}^T$, while the new output matrix $\mathbf{G}$ has size bounded by $\mathbf{F}$. In other words, the new input has degrees that matches the old output, while the new output has degrees bounded by the old input. It is therefore reasonable to expect that the new problem can be computed efficiently. However, we need to find some way to work with the more complicated input degree structure. On the other hand, the simpler output degree structure makes it easier to apply order basis computation to compute a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis.

To see how order basis computations can be applied here, let us first extend Lemma 5.1, which provides a relationship between order bases and kernel bases, to accommodate our situation here.

Lemma 7.5. *Given a matrix* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *and a degree shift* $\vec{u}$ *with* $\operatorname{rdeg}_{\vec{u}} \mathbf{A} \le \vec{v}$*, or equivalently,* $\operatorname{cdeg}_{-\vec{v}} \mathbf{A} \le -\vec{u}$*. Let* $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ *be any* $(\mathbf{A}, \vec{v}+1, -\vec{u})$*-basis and* $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2]$ *be any* $(\mathbf{A}, -\vec{u})$*-kernel basis, where* $\mathbf{P}_1$ *and* $\mathbf{Q}_1$ *contain all columns from* $\mathbf{P}$ *and* $\mathbf{Q}$*, respectively, whose* $-\vec{u}$*-column degrees are no more than* 0*. Then* $[\mathbf{P}_1, \mathbf{Q}_2]$ *is an* $(\mathbf{A}, -\vec{u})$*-kernel basis, and* $[\mathbf{Q}_1, \mathbf{P}_2]$ *is an* $(\mathbf{A}, \vec{v}+1, -\vec{u})$*-basis.*

Proof. We know $\operatorname{cdeg}_{-\vec{v}} \mathbf{A}\mathbf{P}_1 \le \operatorname{cdeg}_{-\vec{u}} \mathbf{P}_1 \le 0$, or equivalently, $\operatorname{rdeg} \mathbf{A}\mathbf{P}_1 \le \vec{v}$, but it has order greater than $\vec{v}$, hence $\mathbf{A}\mathbf{P}_1 = 0$. The result then follows the same reasoning as in the proof of Lemma 5.1.

□

Now with the help of Lemma 7.5, let us get back to our problem of computing a $(\mathbf{F}, \vec{s})$-kernel basis. In fact, we just need to use a special case of Lemma 7.5, where all the elements of the kernel basis have shifted degrees bounded by 0, making the partial kernel basis a complete kernel basis.

Lemma 7.6. *Let* $\mathbf{N}$ *be a* $(\mathbf{F}, \vec{s})$*-kernel basis with* $\operatorname{cdeg}_{\vec{s}} \mathbf{N} = \vec{b}$*. Let* $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ *be a* $\left(\mathbf{N}^T, \vec{b}+1, -\vec{s}\right)$*-basis, where* $\mathbf{P}_1$ *consists of all columns* $\mathbf{p}$ *with* $\operatorname{cdeg}_{-\vec{s}} \mathbf{p} \le 0$*. Then* $\mathbf{P}_1$ *is a* $(\mathbf{N}^T, -\vec{s})$*-kernel basis.*

Proof. Let the rank of $\mathbf{F}$ be r, which is also the column dimension of any $(\mathbf{N}^T, -\vec{s})$-kernel basis $\mathbf{G}^T$. Since both $\mathbf{F}$ and $\mathbf{G}$ are in the left kernel of $\mathbf{N}$, we know $\mathbf{F}$ is generated by $\mathbf{G}$, and the $-\vec{s}$-row degrees of $\mathbf{G}$ are bounded by the corresponding r largest $-\vec{s}$-row degrees of $\mathbf{F}$, which are in turn bounded by 0 since $\operatorname{cdeg} \mathbf{F} \le \vec{s}$. Therefore, any $(\mathbf{N}^T, -\vec{s})$-kernel basis $\mathbf{G}^T$ satisfies $\operatorname{cdeg}_{-\vec{s}} \mathbf{G}^T \le 0$. The result now follows from Lemma 7.5. □

We can use Theorem 5.14 to compute a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis by rows. If we separate $\mathbf{N}$ to $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2]$ with $\vec{s}$-column degrees $\vec{b}_1$, $\vec{b}_2$ respectively, and first compute a $\left(\mathbf{N}_1^T, -\vec{s}\right)$-kernel basis $\mathbf{Q}_1$ with $-\vec{s}$-column degrees $-\vec{s}_2$, and then compute a $\left(\mathbf{N}_2^T\mathbf{Q}_1, -\vec{s}_2\right)$-kernel basis $\mathbf{Q}_2$, then $\mathbf{Q}_1\mathbf{Q}_2$ is a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis. To compute kernel bases $\mathbf{Q}_1$ and $\mathbf{Q}_2$, we can use order basis computation. However, we need to make sure that the order bases we compute do contain these kernel bases.

Lemma 7.7. *Let* $\mathbf{N}$ *be partitioned as* $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2]$*, with* $\vec{s}$*-column degrees* $\vec{b}_1$*,* $\vec{b}_2$ *respectively. Then a* $\left(\mathbf{N}_1^T, \vec{b}_1 + 1, -\vec{s}\right)$*-basis contains a* $\left(\mathbf{N}_1^T, -\vec{s}\right)$*-kernel basis whose* $-\vec{s}$*-column degrees are bounded by 0. Let* $\mathbf{Q}_1$ *be this kernel basis, and* $-\vec{s}_2 = \operatorname{cdeg}_{-\vec{s}} \mathbf{Q}_1$*. Then a* $\left(\mathbf{N}_2^T\mathbf{Q}_1, \vec{b}_2 + 1, -\vec{s}_2\right)$*-basis contains a* $\left(\mathbf{N}_2^T\mathbf{Q}_1, -\vec{s}_2\right)$*-kernel basis* $\mathbf{Q}_2$ *whose* $-\vec{s}$*-column degrees are bounded by 0. The product* $\mathbf{Q}_1\mathbf{Q}_2$ *is then a* $\left(\mathbf{N}^T, -\vec{s}\right)$*-kernel basis.*

Proof. To see that a $\left(\mathbf{N}_1^T, \vec{b}_1 + 1, -\vec{s}\right)$-basis contains a $\left(\mathbf{N}_1^T, -\vec{s}\right)$-kernel basis whose $-\vec{s}$-column degrees are bounded by 0, we just need to show that $\operatorname{cdeg}_{-\vec{s}} \bar{\mathbf{Q}}_1 \le 0$ for any $\left(\mathbf{N}_1^T, -\vec{s}\right)$-kernel basis $\bar{\mathbf{Q}}_1$ and then apply Lemma 7.5. Note that there exists a polynomial matrix $\bar{\mathbf{Q}}_2$ such that $\bar{\mathbf{Q}}_1\bar{\mathbf{Q}}_2 = \bar{\mathbf{G}}$ for any $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis $\bar{\mathbf{G}}$, as $\bar{\mathbf{G}}$ satisfies $\mathbf{N}_1^T\bar{\mathbf{G}} = 0$ and is therefore generated by the $\left(\mathbf{N}_1^T, -\vec{s}\right)$-kernel basis $\bar{\mathbf{Q}}_1$. If $\operatorname{cdeg}_{-\vec{s}} \bar{\mathbf{Q}}_1 \not\le 0$, then Lemma 2.17 forces $\operatorname{cdeg}_{-\vec{s}} \left(\bar{\mathbf{Q}}_1\bar{\mathbf{Q}}_2\right) = \operatorname{cdeg}_{-\vec{s}} \bar{\mathbf{G}} \not\le 0$, a contradiction since we know from the proof of Lemma 7.6 that $\operatorname{cdeg}_{-\vec{s}} \mathbf{G}^T \le 0$.

As before, to see that a $\left(\mathbf{N}_2^T\mathbf{Q}_1, \vec{b}_2 + 1, -\vec{s}_2\right)$-basis contains a $\left(\mathbf{N}_2^T\mathbf{Q}_1, -\vec{s}_2\right)$-kernel basis whose $-\vec{s}$-column degrees are no more than 0, we can just show $\operatorname{cdeg}_{-\vec{s}_2} \hat{\mathbf{Q}}_2 \le 0$ for any $\left(\mathbf{N}_2^T\mathbf{Q}_1, -\vec{s}_2\right)$-kernel basis $\hat{\mathbf{Q}}_2$ and then apply Lemma 7.5. Since $\operatorname{cdeg}_{\vec{s}} \mathbf{N}_2 = \vec{b}_2$, we have $\operatorname{rdeg}_{-\vec{b}_2} \mathbf{N}_2 \le -\vec{s}$ or equivalently, $\operatorname{cdeg}_{-\vec{b}_2} \mathbf{N}_2^T \le -\vec{s}$. Then combining this with $\operatorname{cdeg}_{-\vec{s}} \mathbf{Q}_1 = -\vec{s}_2$ we get $\operatorname{cdeg}_{-\vec{b}_2} \mathbf{N}_2^T\mathbf{Q}_1 \le -\vec{s}_2$ using Lemma 2.17. Let $\hat{\mathbf{G}} = \mathbf{Q}_1\hat{\mathbf{Q}}_2$, which is now a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis by Theorem 5.14. Note that $\operatorname{cdeg}_{-\vec{s}_2} \hat{\mathbf{Q}}_2 = \operatorname{cdeg}_{-\vec{s}} \mathbf{Q}_1\hat{\mathbf{Q}}_2 = \operatorname{cdeg}_{-\vec{s}} \hat{\mathbf{G}} \le 0$. □

Now that we can correctly compute a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis by rows with the help of order basis computation using Lemma 7.7, we need to look at how to do it efficiently. One major difficulty is that the order $\vec{b} + 1$, or equivalently, the $\vec{s}$-row degrees of $\mathbf{N}_1^T$ are nonuniform and can have degree as large as $\sum \vec{s}$. To overcome

Algorithm 7.1 minimaKernelBasisReversed$(\mathbf{M}, \vec{s}, \xi)$

Input: $\mathbf{M} \in \mathbb{K}[x]^{k\times n}$ and $\vec{s} \in \mathbb{Z}^n_{\geq 0}$ such that $\sum \mathrm{rdeg}_{\vec{s}}\mathbf{M} \leq \xi$, $\sum \vec{s} \leq \xi$, and any $(\mathbf{M}, -\vec{s})$-kernel basis having row degrees bounded by $\vec{s}$ (equivalently, having $-\vec{s}$-column degrees bounded by 0).

Output: $\mathbf{G} \in \mathbb{K}[x]^{n\times *}$, a $(\mathbf{M}, -\vec{s})$-kernel basis.

```
 1: $[\mathbf{M}_1^T, \mathbf{M}_2^T, \cdots, \mathbf{M}_{\log k-1}^T, \mathbf{M}_{\log k}^T] := \mathbf{M}^T$, with $\mathbf{M}_{\log k}, \mathbf{M}_{\log k-1}, \cdots, \mathbf{M}_2, \mathbf{M}_1$ having $\vec{s}$-row degrees in the range $[0, \frac{2\xi}{k}], (\frac{2\xi}{k}, \frac{4\xi}{k}], ..., (\frac{\xi}{4}, \frac{\xi}{2}], (\frac{\xi}{2}, \xi]$.
 2: for $i$ from 1 to $\log k$ do
 3:     $\vec{\sigma}_i := [\frac{\xi}{2^{i-1}} + 1, \ldots, \frac{\xi}{2^{i-1}} + 1]$, number of entries matching the row dimension of $\mathbf{M}_i$;
 4: end for
 5: $\vec{\sigma} := [\vec{\sigma}_1, \vec{\sigma}_2, \ldots, \vec{\sigma}_{\log k}]$;
 6: $\hat{\mathbf{N}} := x^{\vec{\sigma}-\vec{b}-1}\mathbf{M}$;
 7: $\mathbf{G}_0 := I_n$; $\tilde{\mathbf{G}}_0 := I_n$;
 8: for $i$ from 1 to $\log k$ do
 9:     $\vec{s}_i := -\,\mathrm{cdeg}_{-\vec{s}}\mathbf{G}_{i-1}$; (note $\vec{s}_1 = \vec{s}$)
10:     $\mathbf{P}_i :=$ unbalancedFastOrderBasis $\left(\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}, \vec{\sigma}_i, -\vec{s}_i\right)$;
11:     $[\mathbf{G}_i, \mathbf{Q}_i] := \mathbf{P}_i$, where $\mathbf{N}_i$ is a $\left(\hat{\mathbf{M}}_i, -\vec{s}_i\right)$-kernel basis;
12:     $\tilde{\mathbf{G}}_i := \tilde{\mathbf{G}}_{i-1} \cdot \mathbf{G}_i$;
13: end for
14: return $\tilde{\mathbf{G}}_i$
```

this, we separate the rows of $\mathbf{N}^T$ into blocks according to their $\vec{s}$-row degrees, and then work with these blocks one by one successively using Theorem 5.14.

Let k be the column dimension of $\mathbf{N}$ and ξ be an upper bound of $\sum \vec{s}$. Since $\sum \mathrm{cdeg}_{\vec{s}}\mathbf{N} = \sum \vec{b} \leq \sum \vec{s} \leq \xi$, at most k/c columns of $\mathbf{N}$ have $\vec{s}$-column degrees greater than or equal to $c\xi/k$ for any $c \geq 1$. We assume without loss of generality that the rows of $\mathbf{N}^T$ are arranged in decreasing $\vec{s}$-row degrees. We divide $\mathbf{N}^T$ into $\log k$ row blocks according to the $\vec{s}$-row degrees of its rows, or equivalently, divide $\mathbf{N}$ to blocks of columns according to the $\vec{s}$-column degrees. Let

$$\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2, \cdots, \mathbf{N}_{\log k-1}, \mathbf{N}_{\log k}]$$

with $\mathbf{N}_{\log k}, \mathbf{N}_{\log k-1}, \ldots, \mathbf{N}_2, \mathbf{N}_1$ having $\vec{s}$-column degrees in the range $[0, 2\xi/k]$, $(2\xi/k, 4\xi/k]$, $(4\xi/k, 8\xi/k]$, ..., $(\xi/4, \xi/2]$, $(\xi/2, \xi]$. Let

$$\vec{\sigma}_i = \left[\xi/2^{i-1}+1, \ldots, \xi/2^{i-1}+1\right]$$

with the same dimension as the row dimension of $\mathbf{N}_i$. Let

$$\vec{\sigma} = [\vec{\sigma}_{\log k}, \vec{\sigma}_{\log k-1}, \ldots, \vec{\sigma}_1]$$

be the order in the order basis computation.

To further simply our task, we also make the order of our problem in each block uniform. Rather than of using $\mathbf{N}^T$ as the input matrix, we use

$$\hat{\mathbf{N}} = \begin{bmatrix} \hat{\mathbf{N}}_1 \\ \vdots \\ \hat{\mathbf{N}}_{\log k} \end{bmatrix} = x^{\vec{\sigma}-\vec{b}-1} \begin{bmatrix} \mathbf{N}_1^T \\ \vdots \\ \mathbf{N}_{\log k}^T \end{bmatrix} = x^{\vec{\sigma}-\vec{b}-1}\mathbf{N}^T$$

instead, so that a $\left(\hat{\mathbf{N}}, \vec{\sigma}, -\vec{s}\right)$-basis is a $\left(\mathbf{N}^T, \vec{b}+1, -\vec{s}\right)$-basis.

We are now ready to compute a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis, which is done by a series of order basis computations that computes a series of kernel bases as follows.

Let $\vec{s}_1 = \vec{s}$. First we compute an $\left(\hat{\mathbf{N}}_1, \vec{\sigma}_1, -\vec{s}_1\right)$-basis $\mathbf{P}_1 = [\mathbf{G}_1, \mathbf{Q}_1]$, where $\mathbf{G}_1$ is a $\left(\hat{\mathbf{N}}_1, -\vec{s}_1\right)$-kernel basis.

Let $\tilde{\mathbf{G}}_1 = \mathbf{G}_1$. Let $\vec{s}_2 = -\operatorname{cdeg}_{-\vec{s}} \mathbf{G}_1$. We then compute an $\left(\hat{\mathbf{N}}_2\tilde{\mathbf{G}}_1, \vec{\sigma}_2, -\vec{s}_2\right)$-basis $\mathbf{P}_2 = [\mathbf{G}_2, \mathbf{Q}_2]$ with $\vec{s}_3 = -\operatorname{cdeg}_{-\vec{s}_2} \mathbf{G}_2$. Let $\tilde{\mathbf{G}}_2 = \tilde{\mathbf{G}}_1\mathbf{G}_2$.

Continuing this process, at step i we compute an $\left(\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}, \vec{\sigma}_i, -\vec{s}_i\right)$-basis $\mathbf{P}_i = [\mathbf{G}_i, \mathbf{Q}_i]$. Let $\tilde{\mathbf{G}}_i = \prod_{j=1}^{i} \mathbf{G}_i = \tilde{\mathbf{G}}_{i-1}\mathbf{G}_i$. Note that $\tilde{\mathbf{G}}_{\log k}$ is a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis.

This process of computing a $\left(\mathbf{N}^T, -\vec{s}\right)$-kernel basis gives Algorithm 7.1.

Now let us check the cost of this algorithm. The cost is dominated by the order basis computation and the multiplications $\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}$ and $\tilde{\mathbf{G}}_{i-1}\mathbf{G}_i$. Let $s = \xi/n$.

Lemma 7.8. *An $\left(\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}, \vec{\sigma}_i, -\vec{s}_i\right)$-basis can be computed with a cost of $O^\sim(n^\omega s)$.*

Proof. Note that $\mathbf{N}_i$ has less than 2^i columns. Otherwise,

$$\sum \operatorname{cdeg}_{\vec{s}} \mathbf{N}_i > 2^i\xi/2^i = \xi,$$

contradicting with

$$\sum \operatorname{cdeg}_{\vec{s}} \mathbf{N} = \sum \vec{b} \le \sum \vec{s} \le \xi.$$

It follows that $\hat{\mathbf{N}}_i$, and therefore $\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}$, also have less than 2^i rows. We also have $\vec{\sigma}_i = [\xi/2^{i-1}+1, \ldots, \xi/2^{i-1}+1]$ with entries in $\Theta(\xi/2^i)$. Therefore, Algorithm 4.1 can be used with a cost of $O^\sim(n^\omega s)$ by Theorem 4.15. $\square$

Lemma 7.9. *The multiplications $\hat{\mathbf{N}}_i\tilde{\mathbf{G}}_{i-1}$ can be done with a cost of $O^\sim(n^\omega s)$.*

Proof. The dimension of $\hat{\mathbf{N}}_i$ is bounded by $2^{i-1}\times n$ and $\sum \operatorname{rdeg}_{\vec{s}} \hat{\mathbf{N}}_i \le 2^{i-1}\cdot\xi/2^{i-1} = \xi$. We also have $\operatorname{cdeg}_{-\vec{s}} \tilde{\mathbf{G}}_{i-1} \le 0$, or equivalently, $\operatorname{rdeg} \tilde{\mathbf{G}}_{i-1} \le \vec{s}$. We can now use Theorem 5.6 to multiply $\tilde{\mathbf{G}}_{i-1}^T$ and $\hat{\mathbf{N}}_i^T$ with a cost of $O^\sim(n^\omega s)$. $\square$

Lemma 7.10. *The multiplication $\tilde{\mathbf{G}}_{i-1}\mathbf{G}_i$ can be done with a cost of $O^\sim(n^\omega s)$.*

Proof. We know $\operatorname{cdeg}_{-\vec{s}} \tilde{\mathbf{G}}_{i-1} = -\vec{s}_i$, and $\operatorname{cdeg}_{-\vec{s}_i} \mathbf{G}_i = -\vec{s}_{i+1} \le 0$. In other words, $\operatorname{rdeg} \mathbf{G}_i \le \vec{s}_i$, and $\operatorname{rdeg}_{\vec{s}_i} \tilde{\mathbf{G}}_{i-1} \le \vec{s}$, hence we can again use Theorem 5.6 to multiply $\mathbf{G}_i^T$ and $\tilde{\mathbf{G}}_{i-1}^T$ with a cost of $O^\sim(n^\omega s)$. $\square$

Lemma 7.11. *Given an input matrix $\mathbf{M} \in \mathbb{K}[x]^{k\times n}$, a shift $\vec{s} \in \mathbb{Z}^n$, and an upper bound $\xi \in \mathbb{Z}$ such that*

- $\sum \operatorname{rdeg}_{\vec{s}} \mathbf{M} \le \xi$,
- $\sum \vec{s} \le \xi$,

- *and any* $(\mathbf{M}, -\vec{s})$*-kernel basis having row degrees bounded by* $\vec{s}$*, or equivalently, having* $-\vec{s}$*-column degrees bounded by 0.*

Then Algorithm 7.1 costs $O^{\sim}(n^{\omega}s)$ *field operations to compute a* $(\mathbf{M}, -\vec{s})$*-kernel basis.*

Note that ξ *can be simply set to* $\sum \vec{s}$*.*

Theorem 7.12. *A right factor* $\mathbf{G}$ *satisfying* $\mathbf{F} = \mathbf{T}\mathbf{G}$ *for a column basis* $\mathbf{T}$ *can be computed with a cost of* $O^{\sim}(n^{\omega}s)$*.*

7.2 Computing a Column Basis

With a right factor $\mathbf{G}$ of $\mathbf{F}$ computed, we are now ready to compute a column basis $\mathbf{T}$ using the equation $\mathbf{F} = \mathbf{T}\mathbf{G}$. To do so efficiently, the degree of $\mathbf{T}$ cannot be too big, which is indeed the case as shown by the following lemmas.

Lemma 7.13. *The column degrees of* $\mathbf{T}$ *are bounded by the corresponding entries of* $\vec{t} = -\operatorname{rdeg}_{-\vec{s}}\mathbf{G}$*.*

Proof. Since $\mathbf{G}$ is $-\vec{s}$-row reduced, and $\operatorname{rdeg}_{-\vec{s}}\mathbf{F} \le 0$, by Lemma 2.17 $\operatorname{rdeg}_{-\vec{t}}\mathbf{T} \le 0$, or equivalently, $\mathbf{T}$ has column degrees bounded by $\vec{t}$. □

Lemma 7.14. *Let* $\vec{t} = -\operatorname{rdeg}_{-\vec{s}}\mathbf{G}$*, a vector with* r *entries and bounds* $\operatorname{cdeg}\mathbf{T}$ *from Lemma 7.13. Let* $\vec{s}'$ *be the list of the* r *largest entries of* $\vec{s}$*. Then* $\vec{t} \le \vec{s}'$*.*

Proof. Let $\mathbf{G}'$ be the $-\vec{s}$-row Popov form of $\mathbf{G}$, and the square matrix $\mathbf{G}$" consists of only the columns of $\mathbf{G}'$ that contains pivot entries, and has the rows permuted so the pivots are in the diagonal. Let $\vec{s}$" be the list of the entries in $\vec{s}$ that correspond to the columns of $\mathbf{G}$" in $\mathbf{G}'$. Note that $\operatorname{rdeg}_{-\vec{s}"}\mathbf{G}" = -\vec{t}"$ is just a permutation of $-\vec{t}$ with the same entries. By the definition of shifted row degree, $-\vec{t}$" is the sum of $-\vec{s}$" and the list of the diagonal pivot degrees, which are nonnegative. Therefore,

$-\vec{t}'' \geq -\vec{s}''$. The result then follows as $\vec{t}$ is a permutation of $\vec{t}''$ and $\vec{s}'$ has the largest entries of $\vec{s}$. $\square$

With the bound on the column degrees of $\mathbf{T}$ determined, we are now ready to compute $\mathbf{T}$. This is done again using an order basis computation.

Lemma 7.15. *Let $\vec{t}' = \left[0, \ldots, 0, \vec{t}\right] \in \mathbb{Z}^{m+r}$. Any $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}'\right)$-kernel basis has the form $\begin{bmatrix} V \\ \bar{\mathbf{T}} \end{bmatrix}$, where $V \in \mathbb{K}^{m \times m}$ is a unimodular matrix and $\left(\bar{\mathbf{T}} V^{-1}\right)^T$ is a column basis of $\mathbf{F}$.*

Proof. Note first that the matrix $\begin{bmatrix} -I \\ \mathbf{T}^T \end{bmatrix}$ is a kernel basis of $\left[\mathbf{F}^T, \mathbf{G}^T\right]$ and is therefore unimodularly equivalent to any other kernel basis. Hence any other kernel basis has the form $\begin{bmatrix} -I \\ \mathbf{T}^T \end{bmatrix} U = \begin{bmatrix} V \\ \bar{\mathbf{T}} \end{bmatrix}$, with U and $V = -U$ unimodular. Thus $\mathbf{T} = \left(\bar{\mathbf{T}} V^{-1}\right)^T$. Also note that the $-\vec{t}'$-minimality forces the unimodular matrix V in any $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}'\right)$-kernel basis to be degree 0, the same degree as I. $\square$

To compute a $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}'\right)$-kernel basis, we can again use order basis computation.

Lemma 7.16. *Any $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], \vec{s}+1, -\vec{t}'\right)$-basis contain a $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}'\right)$-kernel basis whose $-\vec{t}'$-row degrees are bounded by 0.*

Proof. As before, Lemma 7.5 can be used here. We just need to show that a $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}'\right)$-kernel basis has $-\vec{t}'$-row degrees no more than 0, which is true since $\operatorname{rdeg}_{-\vec{t}'} \begin{bmatrix} I \\ \mathbf{T}^T \end{bmatrix} \leq 0$. $\square$

Example 7.17. Let

$$\mathbf{F} = \begin{bmatrix} x^2 & x^2 & x + x^2 & 1 + x^2 \\ 1 + x + x^2 & x^2 & 1 + x^2 & 1 + x^2 \end{bmatrix}$$

with

$$\mathbf{G}=\begin{bmatrix}1 & 0 & 1 & 0\\ x & x^2 & 0 & 1+x^2\end{bmatrix}$$

being a minimal left kernel basis of a right kernel basis of $\mathbf{F}$. In order to compute the column basis $\mathbf{T}$ satisfying $\mathbf{F}=\mathbf{TG}$, first we can determine $\operatorname{cdeg}\mathbf{T}\le\vec{t}=[2,0]$ from Lemma 7.13. Then we can compute a $\left[0,0,-\vec{t}\right]$-minimal left kernel basis of $\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}$. The matrix

$$\left[V,\bar{\mathbf{T}}\right]=\begin{bmatrix}1 & 0 & x+x^2 & 1\\ 1 & 1 & 1+x & 0\end{bmatrix}$$

is such a left kernel basis. A column basis can then be computed as by

$$\mathbf{T}=V^{-1}\bar{\mathbf{T}}=\begin{bmatrix}x+x^2 & 1\\ 1+x^2 & 1\end{bmatrix}.$$

In order to compute a $\left(\left[\mathbf{F}^T,\mathbf{G}^T\right],-\vec{t^*}\right)$ kernel basis efficiently, we notice that we have the same type of problem as in Section 7.1 and hence we can again use Algorithm 7.1.

Lemma 7.18. *A $\left(\left[\mathbf{F}^T,\mathbf{G}^T\right],-\vec{t^*}\right)$-kernel basis can be computed using Algorithm 7.1 with a cost of $O^\sim(n^\omega s)$, where $s=\xi/n$ is the average column degree of $\mathbf{F}$ as before.*

Proof. Just use the algorithm with input $\left(\left[\mathbf{F}^T,\mathbf{G}^T\right],\vec{t^*},\xi\right)$. We can verify the conditions on the input are satisfied.

- To see that $\sum\operatorname{rdeg}_{\vec{t^*}}\left[\mathbf{F}^T,\mathbf{G}^T\right]\le\xi$, note that from $\vec{t}=-\operatorname{rdeg}_{-\vec{s}}\mathbf{G}$ and Lemma 2.2 $\operatorname{cdeg}_{\vec{t}}\mathbf{G}\le\vec{s}$, or equivalently, $\operatorname{rdeg}_{\vec{t}}\mathbf{G}^T\le\vec{s}$. Since we also have $\operatorname{rdeg}\mathbf{F}^T\le\vec{s}$, it follows that $\operatorname{rdeg}_{\vec{t^*}}\left[\mathbf{F}^T,\mathbf{G}^T\right]\le\vec{s}$.

- The second condition $\sum\vec{t^*}\le\xi$ follows from Lemma 7.13.

Algorithm 7.2 colBasis($\mathbf{F}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$.
Output: a column basis of $\mathbf{F}$.
1: $\vec{s} :=$ cdeg $\mathbf{F}$;
2: $\mathbf{N} :=$ minimalKernelBasis $(\mathbf{F}, \vec{s})$;
3: $\mathbf{G} := \left(\text{minimaKernelBasisReversed}(\mathbf{N}^T, \vec{s})\right)^T$;
4: $\vec{t} := \left[0, \ldots, 0, -\text{rdeg}_{-\vec{s}}\mathbf{G}\right]$, with rowDimension($\mathbf{G}$) number of 0's ;
5: $\left[V^T, \bar{\mathbf{T}}^T\right]^T :=$ minimaKernelBasisReversed($\left[\mathbf{F}^T, \mathbf{G}^T\right], \vec{t}$) with a square $V$;
6: $\mathbf{T} = \left(\bar{\mathbf{T}}V^{-1}\right)^T$;
7: return $\mathbf{T}$;
```

- The third condition holds since $\begin{bmatrix} -I \\ \mathbf{T}^T \end{bmatrix}$ is a kernel basis with row degrees bounded by $\vec{t}^*$.

□

With a $\left(\left[\mathbf{F}^T, \mathbf{G}^T\right], -\vec{t}^*\right)$-kernel basis $\begin{bmatrix} V \\ \bar{\mathbf{T}} \end{bmatrix}$ computed, a column basis is then given by $\mathbf{T} = \left(\bar{\mathbf{T}}V^{-1}\right)^T$.

The complete algorithm for computing a column basis is then given in Algorithm 7.2.

Theorem 7.19. *A column basis $\mathbf{T}$ of $\mathbf{F}$ can be computed with a cost of $O^\sim(n^\omega s)$, where $s = \xi/n$ is the average column degree of $\mathbf{F}$ as before.*

Proof. The cost is dominated by the cost of the three kernel basis computations in the algorithm. The first one is handled by the algorithm from Zhou et al. [2012] and Theorem 5.19, while the remaining two are handled by Algorithm 7.1, Lemma 7.11 and Lemma 7.18. □

7.3 A Simple Improvement

When the input matrix $\mathbf{F}$ has column dimension much larger n than the row dimension m, we can separate $\mathbf{F} = \left[\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{n/m}\right]$ to n/m blocks, each with dimension $m \times m$, assuming without loss of generality n is a multiple of m, and the columns are arranged in increasing degrees. We then do a series of column basis computations. First we compute a column basis $\mathbf{T}_1$ of $[\mathbf{F}_1, \mathbf{F}_2]$. Then compute a column basis $\mathbf{T}_2$ of $[\mathbf{T}_1, \mathbf{F}_3]$. Repeating this process, at step i, we compute a column basis $\mathbf{T}_i$ of $[\mathbf{T}_{i-1}, \mathbf{F}_{i+1}]$, until $i = n/m - 1$, when a column basis of $\mathbf{F}$ is computed.

Lemma 7.20. *At step i, computing a column basis $\mathbf{T}_i$ of $[\mathbf{T}_{i-1}, \mathbf{F}_{i+1}]$ can be done with a cost of $O^{\sim}\left(m^{\omega}(s_i + s_{i+1})/2\right)$ field operations, where $s_i = \left(\sum \operatorname{cdeg} \mathbf{F}_i\right)/m$.*

Proof. From Lemma 7.13, the column basis $\mathbf{T}_{i-1}$ of $[\mathbf{F}_1, \ldots, \mathbf{F}_i]$ has column degrees bounded by the largest column degrees of $\mathbf{F}_i$, hence $\sum \operatorname{cdeg} \mathbf{T}_{i-1} \le \sum \operatorname{cdeg} \mathbf{F}_i$. The lemma then follows by combining this with the result from Theorem 7.19 that a column basis $\mathbf{T}_i$ of $[\mathbf{T}_{i-1}, \mathbf{F}_{i+1}]$ can be computed with a cost of $O^{\sim}\left(m^{\omega}\bar{s}\right)$, where

$$\bar{s} = \left(\sum \operatorname{cdeg} \mathbf{T}_{i-1} + \sum \operatorname{cdeg} \mathbf{F}_{i+1}\right)/2m \le \left(s_i + s_{i+1}\right)/2.$$

□

Theorem 7.21. *A column basis of $\mathbf{F}$ can be computed with a cost of $O^{\sim}\left(m^{\omega}s\right)$, where $s = \left(\sum \operatorname{cdeg} \mathbf{F}\right)/n$.*

Proof. Summing up the cost of all the column basis computations,

$$\begin{aligned} &\sum_{i=1}^{n/m-1} O^{\sim}\left(m^{\omega}\left(s_i + s_{i+1}\right)/2\right) \\ &\qquad \subset O^{\sim}\left(m^{\omega}\left(\sum_{i=1}^{n/m} s_i\right)/(n/m)\right) \\ &\qquad = O^{\sim}\left(m^{\omega}s\right). \end{aligned}$$

□

Remark 7.22. In this section, the computational efficiency is improved by reducing the original problem to about n/m subproblems whose column dimensions are close to the row dimension m. This is done by successive column basis computations. Note that we can also reduce the column dimension by using successive order basis computations, and only do a column basis computation at the very last step. The computational complexity of using order basis computation to reduce the column dimension would remain the same, but in practice it maybe more efficient since order basis computations are simpler.

7.4 Column Reduced Form and Popov Form

Let us now look how column basis computation leads to efficient deterministic algorithms for computing column reduced form and Popov form for matrices of any dimension. Since Sarkar and Storjohann [2011] already provided algorithms to transform column reduced forms to Popov forms, we just need to consider the problem of computing column reduced form. In addition, since Gupta et al. [2012] provided a deterministic algorithm for the column reduction of a square nonsingular input matrix, we just need to reduce the problem with general input matrix to the square nonsingular case. For this problem, we only give the cost in terms of the less refined matrix degree d instead of the sum of the column degrees and aim for a cost of $O^{\sim}(nm^{\omega-1}d)$. So there is more room for improvement here.

Theorem 7.23. *The column reduced form and Popov form of any matrix* $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ *can be computed deterministically with a cost of* $O^{\sim}(nm^{\omega-1}d)$.

Proof. We may now assume that the input matrix $\mathbf{F}$ has full column rank, which can be done by a direct application of the column basis computation. It only remains to consider the case that the row dimension m of $\mathbf{F}$ is higher than its column dimension n. Using the transposed version of Lemma 7.3, we can factor $\mathbf{F}$ as $\mathbf{F} = \mathbf{G}\mathbf{T}$, where $\mathbf{G}$ is column reduced and $\mathbf{T}$ is a square nonsingular row basis of $\mathbf{F}$. Let $\vec{t} = -\operatorname{cdeg}_{[-d,\ldots,-d]} \mathbf{G}$, or equivalently, $\vec{t} = d - \operatorname{cdeg} \mathbf{G}$, then from Lemma 7.13 we have $\operatorname{cdeg}_{-\vec{t}} \mathbf{T} \le 0$, and from Lemma 7.14 we know that $\vec{t} \le d$. Now using Lemma 2.18, a $-\vec{t}$-column reduced form $\mathbf{T}'$ of $\mathbf{T}$ makes $\mathbf{G}\mathbf{T}'$ a column reduced form of $\mathbf{F}$. To compute a $-\vec{t}$-column reduced form $\mathbf{T}'$ of $\mathbf{T}$, we can just compute a column reduced form of $x^{d-\vec{t}}\mathbf{T}$, which is a square nonsingular matrix of degree d. □

Example 7.24. To column reduce

$$\mathbf{F} = \begin{bmatrix} x^2 & 1+x+x^2 \\ x^2 & x^2 \\ x+x^2 & 1+x^2 \\ 1+x^2 & 1+x^2 \end{bmatrix},$$

we factor $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{G}\mathbf{T} = \begin{bmatrix} 1 & x \\ 0 & x^2 \\ 1 & 0 \\ 0 & 1+x^2 \end{bmatrix} \begin{bmatrix} x+x^2 & 1+x^2 \\ 1 & 1 \end{bmatrix}$$

as before with a column reduced $\mathbf{G}$. The column degrees of $\mathbf{G}$ $\operatorname{cdeg} \mathbf{G} = [0, 2]$. So we compute $[0,2]$-column reduced form $\mathbf{T}'$ of $\mathbf{T}$

$$\mathbf{T}' = \begin{bmatrix} 1+x & x+x^2 \\ 0 & 1 \end{bmatrix}.$$

Now

$$\mathbf{G}\mathbf{T}' = \begin{bmatrix} 1+x & x^2 \\ 0 & x^2 \\ 1+x & x+x^2 \\ 0 & 1+x^2 \end{bmatrix}$$

is a column reduced form of $\mathbf{F}$.

Chapter 8

Unimodular Completion

Given a matrix $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ with $n > m$ and column degrees $\vec{s}$, we consider the problem of efficiently computing a matrix $\mathbf{G} \in \mathbb{K}[x]^{(n-m)\times n}$ such that $\begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix}$ is unimodular. Unimodular completion is a useful basic operation in matrix computations [Newman, 1972]. Our goal is to do this with a cost of $O^{\sim}(n^{\omega}s)$ field operations, where s is the average of the m largest column degrees of $\mathbf{F}$.

Before discussing the computation of a unimodular completion, we need to check the existence of unimodular completion for a given matrix. In fact, a unimodular completion does not exist for some input matrices, as in the case of $\mathbf{F} = [0, x]$. So we need to know what type of input matrices admit a unimodular completion.

Lemma 8.1. *A unimodular completion of* $\mathbf{F}$ *exists if and only if* $\mathbf{F}$ *has unimodular column bases.*

Proof. If $\mathbf{F}$ has a non-unimodular column basis $\mathbf{A}$, then $\operatorname{diag}([\mathbf{A}, I])$ is always a factor of $\begin{bmatrix} \mathbf{F} \\ \mathbf{B} \end{bmatrix}$ for any polynomial matrix $\mathbf{B}$, implying that the matrix $\begin{bmatrix} \mathbf{F} \\ \mathbf{B} \end{bmatrix}$ is non-unimodular. On the other hand, if $\mathbf{F}$ has a unimodular column basis, then there exists a unimodular matrix $\mathbf{U}$ such that $\mathbf{FU} = [I_m, 0]$, or $\mathbf{F} = [I_m, 0]\,\mathbf{U}^{-1}$ after rearranging, that is, $\mathbf{F}$ must be consists of the top m rows of $\mathbf{U}^{-1}$. The matrix $\mathbf{U}^{-1}$

is therefore a unimodular completion of the matrix $\mathbf{F}$. $\square$

Since a unimodular completion is only possible for input matrices with unimodular column bases, we assume for simplicity this is the case with our input matrix $\mathbf{F}$. This also requires $\mathbf{F}$ to be full rank. For other matrices without unimodular column bases, our method still works directly to compute a matrix completion for a right factor of $\mathbf{F}$ that has its column basis factor removed. In other words, let $\mathbf{F}$ be factored as $\mathbf{F} = \mathbf{T}\mathbf{R}$ as in Lemma 7.3, where $\mathbf{T}$ is a column basis of $\mathbf{F}$ and $\mathbf{R}$ is the remaining right factor, the our method always works to compute a unimodular completion of $\mathbf{R}$. In the special case where $\mathbf{T}$ is unimodular, the unimodular completion computed is also a unimodular completion of $\mathbf{F}$.

The proof of Lemma 8.1 shows that a unimodular completion of $\mathbf{F}$ can be obtained from the unimodular matrix $\mathbf{U}$ that transforms $\mathbf{F}$ to its column bases. However, we may not be able to compute this $\mathbf{U}$ efficiently since its degree might be too large. More specifically, $\mathbf{U}$ contains a kernel basis of $\mathbf{F}$ that may have degree ξ, while each of the remaining columns of $\mathbf{U}$ may also have degree ξ.

Before discussing the actual matrix completion, let us look at the operations that reverses the coefficients of a polynomial, the coefficients of the polynomial entries of a vector, and the coefficients of the polynomial entries of a polynomial matrix. These operations are needed in the computation of our matrix completion.

8.1 Reversing polynomial coefficients

First let us look at the operation that reverses the coefficients of a polynomial.

Definition 8.2. For a polynomial $p = p_0 + p_1 x + \cdots + p_u x^u \in \mathbb{K}[x]$ with degree bounded by u, we define the operation

$$\operatorname{rev}(p, u) = \left(p(x^{-1})\right) x^u = p_u + p_{u-1} x + \cdots + p_1 x^{u-1} + p_0 x^u.$$

We now extend this definition to column vectors and row vectors with shifted degrees.

Definition 8.3. Let $\vec{u} = [u_1, \dots u_n] \in \mathbb{Z}^n$ be a degree shift, and a column vector $\mathbf{a} \in \mathbb{K}[x]^{n \times 1}$ with $\vec{u}$-column degree bounded by v. We define

$$\operatorname{colRev}(\mathbf{a}, \vec{u}, v) = x^{-\vec{u}} \left(\mathbf{a}(x^{-1})\right) x^{v} = \begin{bmatrix} \operatorname{rev}(p, v - u_1) \\ \vdots \\ \operatorname{rev}(p, v - u_n) \end{bmatrix}.$$

Similarly for a row vector $\mathbf{b} \in \mathbb{K}[x]^{1 \times n}$ with $\vec{u}$-row degree bounded by v, where $\vec{u} = [u_1, \dots u_n] \in \mathbb{Z}^n$ is a degree shift, we define

$$\operatorname{rowRev}(\mathbf{b}, \vec{u}, v) = \operatorname{colRev}(\mathbf{b}^T, \vec{u}, v)^T = x^{v} \left(\mathbf{b}(x^{-1})\right) x^{-\vec{u}}.$$

Example 8.4. If $\mathbf{f} = [10 + x, 5 + x + 2x^2]$, $\vec{u} = [-1, -2]$, and $v = 0$, then

$$\operatorname{rowRev}(\mathbf{f}, \vec{u}, v) = x^0 \left[10 + x^{-1}, 5 + x^{-1} + 2x^{-2}\right] \begin{bmatrix} x & \\ & x^2 \end{bmatrix} = \left[10x + 1, 5x^2 + x + 2\right].$$

We can extend the reverse operation further to polynomial matrices.

Definition 8.5. Let $\vec{u} = [u_1, \dots u_n] \in \mathbb{Z}^n$ be a degree shift. Let $\mathbf{A} \in \mathbb{K}[x]^{n \times k}$ with $\vec{u}$-column degrees bounded component-wise by $\vec{v} = [v_1, \dots, v_k]$, we define

$$\operatorname{colRev}(\mathbf{A}, \vec{u}, \vec{v}) = x^{-\vec{u}} \left(\mathbf{A}(1/x)\right) x^{\vec{v}}$$

Similarly, for $\vec{u} = [u_1, \dots, u_n]$ and $\mathbf{B} \in \mathbb{K}[x]^{k \times n}$ with $\vec{u}$-row degrees bounded component-wise by $\vec{v} = [v_1, \dots, v_k]$,

$$\operatorname{rowRev}(\mathbf{B}, \vec{u}, \vec{v}) = \operatorname{colRev}(\mathbf{B}^T, \vec{u}, \vec{v})^T = x^{\vec{v}} \left(\mathbf{B}(1/x)\right) x^{-\vec{u}}$$

Note that we also have $\text{rowRev}(\mathbf{B}, \vec{u}, \vec{v}) = \text{colRev}(\mathbf{B}, -\vec{v}, -\vec{u})$.

It is useful to note that any degree bound remains the same after the reverse operations.

Lemma 8.6. *If* $\mathbf{A} \in \mathbb{K}[x]^{n\times k}$ *has* $\vec{u}$*-column degrees bounded by the corresponding entries of* $\vec{v}$*, then* $\text{colRev}(\mathbf{A}, \vec{u}, \vec{v})$ *also has* $\vec{u}$*-column degrees bounded by the corresponding entries of* $\vec{v}$*.*

As one would expect, applying two reverse operations gives back the original input.

Lemma 8.7. *The following equalities holds:*

$$\begin{aligned} \text{colRev}\left(\text{colRev}(\mathbf{A}, \vec{u}, \vec{v}), \vec{u}, \vec{v}\right) &= \mathbf{A} \\ \text{rowRev}\left(\text{rowRev}(\mathbf{B}, \vec{u}, \vec{v}), \vec{u}, \vec{v}\right) &= \mathbf{B} \end{aligned}$$

Let us look at a degree bound on the product of a row vector and a column vector, based on their shifted degrees, when opposite shifts are used.

Lemma 8.8. *If* $\mathbf{a} \in \mathbb{K}[x]^{1\times n}$ *and* $\mathbf{a}^T$ *has* $(-\vec{u})$*-column degree bounded by* α *(or equivalently,* $\mathbf{a}$ *has* $(-\vec{u})$*-row degree bounded by* α*) and* $\mathbf{b} \in \mathbb{K}[x]^{n\times 1}$ *has* $\vec{u}$*-column degree bounded by* β*, then* $\mathbf{ab}$ *has degree bounded by* $\alpha + \beta$*.*

Proof. Since $\mathbf{a}x^{-\vec{u}}$ has degree bounded by α and $x^{\vec{u}}\mathbf{b}$ has degree bounded by β, $\mathbf{a}x^{-\vec{u}}x^{\vec{u}}\mathbf{b} = \mathbf{ab}$ has degree bounded by $\alpha + \beta$. □

The following lemma shows that the reverse operation and the multiplication are commutative when we use the opposite shifts.

Lemma 8.9. *If* $\mathbf{a} \in \mathbb{K}[x]^{1\times n}$ *has* $(-\vec{u})$*-row degree bounded by* α *and* $\mathbf{b} \in \mathbb{K}[x]^{n\times 1}$ *has* $\vec{u}$*-column degree bounded by* β*, then*

$$\text{rowRev}(\mathbf{a}, -\vec{u}, \alpha) \cdot \text{colRev}(\mathbf{b}, \vec{u}, \beta) = \text{rev}(\mathbf{ab}, \alpha + \beta).$$

Proof.

$$\begin{aligned}
& \operatorname{rowRev}(\mathbf{a}, -\vec{u}, \alpha) \cdot \operatorname{colRev}(\mathbf{b}, \vec{u}, \beta) \\
= \; & x^{\alpha} \left(\mathbf{a}(1/x)\right) x^{\vec{u}} x^{-\vec{u}} \left(\mathbf{b}(1/x)\right) x^{\beta} \\
= \; & \left(\mathbf{a}(1/x)\right) x^{\vec{u}} x^{-\vec{u}} \left(\mathbf{b}(1/x)\right) x^{\alpha+\beta} \\
= \; & \left(\mathbf{a}(1/x)\right) \left(\mathbf{b}(1/x)\right) x^{\alpha+\beta} \\
= \; & \left(\left(\mathbf{ab}\right)(1/x)\right) x^{\alpha+\beta} \\
= \; & \operatorname{rev}(\mathbf{ab}, \alpha+\beta)
\end{aligned}$$

We also have the following similar result on the reverse operation and matrix multiplication □

Lemma 8.10. *If* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *has* $\vec{u}$*-column degrees bounded by* $\vec{v}$*, and* $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$ *has* $\vec{v}$*-column degrees bounded by* $\vec{w}$*, then*

$$\operatorname{colRev}(\mathbf{A}, \vec{u}, \vec{v}) \operatorname{colRev}(\mathbf{B}, \vec{v}, \vec{w}) = \operatorname{colRev}(\mathbf{AB}, \vec{u}, \vec{w})$$

has $\vec{u}$*-column degrees bounded by* $\vec{w}$*.*

Proof.

$$\begin{aligned}
& \operatorname{colRev}(\mathbf{A}, \vec{u}, \vec{v}) \operatorname{colRev}(\mathbf{B}, \vec{v}, \vec{w}) \\
= \; & x^{-\vec{u}} \left(\mathbf{A}(1/x)\right) x^{\vec{v}} x^{-\vec{v}} \left(\mathbf{B}(1/x)\right) x^{\vec{w}} \\
= \; & x^{-\vec{u}} \left(\mathbf{AB}\right)(1/x) x^{\vec{w}}.
\end{aligned}$$

□

Lemma 8.11. *If* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *has* $\vec{u}$*-row degrees bounded by* $\vec{v}$*, and* $\mathbf{B} \in \mathbb{K}[x]^{n\times k}$

has $-\vec{u}$-column degrees bounded by $\vec{w}$, then

$$\mathrm{rowRev}(\mathbf{A},\vec{u},\vec{v})\,\mathrm{colRev}(\mathbf{B},-\vec{u},\vec{w}) = \mathrm{colRev}(\mathbf{AB},-\vec{v},\vec{w}).$$

.

Proof.

$$\begin{aligned}
& \mathrm{rowRev}(\mathbf{A},\vec{u},\vec{v})\,\mathrm{colRev}(\mathbf{B},-\vec{u},\operatorname*{cdeg}_{-\vec{u}}\mathbf{B}) \\
= \;& x^{\vec{v}}\left(\mathbf{A}(1/x)\right)x^{-\vec{u}}x^{\vec{u}}\left(\mathbf{B}(1/x)\right)x^{\mathrm{cdeg}_{-\vec{u}}\mathbf{B}} \\
= \;& x^{\vec{v}}\left(\mathbf{A}(1/x)\right)\left(\mathbf{B}(1/x)\right)x^{\mathrm{cdeg}_{-\vec{u}}\mathbf{B}} \\
= \;& x^{\vec{v}}\left(\mathbf{AB}(1/x)\right)x^{\mathrm{cdeg}_{-\vec{u}}\mathbf{B}}.
\end{aligned}$$

□

8.2 Unimodular completion

In this section, we look at how a unimodular completion can be done using a combination of kernel basis computations, order basis computations, and reverse operations. First, we have the following natural relationship between a kernel basis and the reverse operation.

Lemma 8.12. *Let $\vec{u} \in \mathbb{Z}^n$, $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ with $(-\vec{u})$-row degrees bounded componentwise by $\vec{a}$, and $\mathbf{A}^r = \mathrm{rowRev}\left(\mathbf{A},-\vec{u},\vec{a}\right)$. Then a matrix $\mathbf{N} \in \mathbb{K}[x]^{n\times k}$ with $\vec{u}$-column degrees $\vec{b}$ is a $(\mathbf{A},\vec{u})$-kernel basis if and only if $\mathbf{N}^r = \mathrm{colRev}\left(\mathbf{N},\vec{u},\vec{b}\right)$ is a $(\mathbf{A}^r,\vec{u})$-kernel basis.*

Proof. If $\mathbf{N}$ is a kernel basis of $\mathbf{A}$, then we know from Lemma 8.9 that

$$\mathrm{rowRev}\left(\mathbf{A},-\vec{u},\vec{a}\right)\cdot\mathrm{colRev}\left(\mathbf{N},\vec{u},\vec{b}\right) = 0,$$

so $\operatorname{colRev}\left(\mathbf{N}, \vec{u}, \vec{b}\right)$ is a kernel basis of $\operatorname{rowRev}\left(\mathbf{A}, -\vec{u}, \vec{a}\right)$. Suppose $\operatorname{colRev}\left(\mathbf{N}, \vec{u}, \vec{b}\right)$ is not $\vec{u}$-minimal and we have another kernel basis $\mathbf{M}$ of $\operatorname{rowRev}\left(\mathbf{A}, -\vec{u}, \vec{a}\right)$ with $\vec{u}$-column degrees $\vec{c}$ that has some entry lower than the corresponding entry in $\vec{b}$. Then $\operatorname{colRev}\left(\mathbf{M}, \vec{u}, \vec{c}\right)$ is also a kernel of $\mathbf{A}$ with lower $\vec{u}$-column degrees than $\vec{b}$, contradicting the $\vec{u}$-minimality of $\mathbf{N}$. □

The following lemma shows the unimodular equivalence between any matrix $\mathbf{A}$ that has a unimodular column basis, and a left kernel basis of any right kernel basis of $\mathbf{A}$.

Lemma 8.13. *Given a matrix* $\mathbf{A} \in \mathbb{K}[x]^{m\times n}$ *with unimodular column basis. Let* $\mathbf{N} \in \mathbb{K}[x]^{n\times(n-m)}$ *be a right kernel basis of* $\mathbf{A}$. *Let* $\mathbf{B}$ *be a left kernel basis of* $\mathbf{N}$. *Then* $\mathbf{A} = \mathbf{U}\mathbf{B}$ *for a unimodular matrix* $\mathbf{U}$.

Proof. This follows from Lemma 7.3, which tells us that $\mathbf{U}$ is just a column basis of $\mathbf{A}$. □

Now let us look at how an order basis can lead to a unimodular matrix.

Lemma 8.14. *Let* $\vec{u} = [u_1, \ldots u_n] \in \mathbb{Z}^n$ *be a degree shift. Any* $(\mathbf{A}, \sigma, \vec{u})$*-basis* $\mathbf{P}$ *with* $\operatorname{cdeg}_{\vec{u}} \mathbf{P} = \vec{v} = [v_1, \ldots, v_k]$ *has* $\det(\mathbf{P}) = cx^{\sum \vec{v} - \sum \vec{u}}$ *and* $\det(\operatorname{colRev}(\mathbf{P}, \vec{u}, \vec{v})) = c$ *for some constant* $c \in \mathbb{K}$. *In other words,* $\operatorname{colRev}(\mathbf{P}, \vec{u}, \vec{v})$ *is unimodular.*

Proof. To see that $\det(\mathbf{P}) = cx^{\sum \vec{v} - \sum \vec{u}}$, note that an identity matrix is an $(\mathbf{A}, 0, \vec{u})$-basis, which has $\vec{u}$-column degrees $\vec{u}$ and determinant 1. Then the $\vec{u}$-column degrees only increases by multiplying some column of $\mathbf{P}$ by x. The second property $\det(\operatorname{colRev}(\mathbf{P}, \vec{u}, \vec{v})) = c$ follows from the definition

$$\operatorname{colRev}(\mathbf{P}, \vec{u}, \vec{v}) = x^{-\vec{u}} \left(\mathbf{P}(1/x)\right) x^{\vec{v}}.$$

□

Lemma 8.14 suggests that a unimodular completion of $\mathbf{F}$ can be computed by embedding $\mathbf{F}$ in a reversed order basis, or equivalently, embedding a reversed $\mathbf{F}$ in an order basis. The next question is therefore how to embed a matrix in an order basis. Recall that Lemma 7.6 shows how kernel bases can be embedded in order bases. Therefore, if we can make the reversed $\mathbf{F}$ a kernel basis of some matrix $\mathbf{M}$, then there is an order basis of $\mathbf{M}$ that contains the reversed $\mathbf{F}$. A natural choice for $\mathbf{M}$ is a kernel basis of the reversed $\mathbf{F}$. We actually have two choices here. We can either reverse the coefficients of $\mathbf{F}$, as we do in Lemma 8.15 below, or we can reverse the coefficients of a kernel basis of $\mathbf{F}$.

Lemma 8.15. *Let* $\mathbf{F}^r = \operatorname{rowRev}(\mathbf{F}, -\vec{s}, 0)$ *and* $\mathbf{M}$ *be a* $(\mathbf{F}^r, \vec{s})$*-kernel basis with* $\operatorname{cdeg}_{\vec{s}}\mathbf{M} = \vec{b}$*. Let* $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ *be a* $\left(\mathbf{M}^T, \vec{b}+1, -\vec{s}\right)$*-basis, where* $\mathbf{P}_1$ *consists of all columns* $\mathbf{p}$ *with* $\operatorname{cdeg}_{-\vec{s}}\mathbf{p} \le 0$*. If* $\mathbf{P}_2^r = \operatorname{colRev}\left(\mathbf{P}_2, -\vec{s}, \operatorname{cdeg}_{-\vec{s}}\mathbf{P}_2\right)$*, then* $\left[\mathbf{F}^T, \mathbf{P}_2^r\right]$ *is a unimodular matrix.*

Proof. Let $\mathbf{P}_1^r = \operatorname{colRev}\left(\mathbf{P}_1, -\vec{s}, \operatorname{cdeg}_{-\vec{s}}\mathbf{P}_1\right)$. We know from Lemma 8.14 that $[\mathbf{P}_1^r, \mathbf{P}_2^r]$ is unimodular. Let $\mathbf{M}^r = \operatorname{colRev}\left(\mathbf{M}, \vec{s}, \vec{b}\right)$. Then from Lemma 8.12 we know $\mathbf{M}^r$ is a $(\mathbf{F}, \vec{s})$-kernel basis and $\mathbf{P}_1^r$ is a $\left((\mathbf{M}^r)^T, -\vec{s}\right)$-kernel basis, hence by Lemma 8.13 $\mathbf{F} = \mathbf{U}\left(\mathbf{P}_1^r\right)^T$ for some unimodular matrix $\mathbf{U}$. Now $\left[\mathbf{F}^T, \mathbf{P}_2^r\right]^T = \operatorname{diag}([\mathbf{U}, I])\left[\mathbf{P}_1^r, \mathbf{P}_2^r\right]^T$. □

Lemma 8.15 provides a way to correctly compute a unimodular completion of $\mathbf{F}$. To improve the computational efficiency, we can in fact separate the rows of $\mathbf{M}^T$ and just work with one subset of rows at a time.

Lemma 8.16. *Let the matrix* $\mathbf{F}^r = \operatorname{rowRev}(\mathbf{F}, -\vec{s}, 0)$*. Let the matrix* $\mathbf{M}$ *be a* $(\mathbf{F}^r, \vec{s})$*-kernel basis with* $\operatorname{cdeg}_{\vec{s}}\mathbf{M} = \vec{b}$ *and be partitioned as* $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2]$*. Let* $\mathbf{P}_1$ *be a* $\left(\mathbf{M}_1^T, \operatorname{cdeg}_{\vec{s}}\mathbf{M}_1 + 1, -\vec{s}\right)$*-basis and be partitioned as* $\mathbf{P}_1 = [\mathbf{N}_1, \mathbf{Q}_1]$*, where* $\mathbf{N}_1$ *consists of all columns* $\mathbf{p}$ *of* $\mathbf{P}_1$ *with* $\operatorname{cdeg}_{-\vec{s}}\mathbf{p} \le 0$*. Let* $\vec{t} = \operatorname{cdeg}_{-\vec{s}}\mathbf{N}_1$ *and* $\mathbf{P}_2$ *be a* $\left(\mathbf{M}_2^T\mathbf{N}_1, \operatorname{cdeg}_{\vec{s}}\mathbf{M}_2 + 1, \vec{t}\right)$*-basis and be partitioned as* $\mathbf{P}_2 = [\mathbf{N}_2, \mathbf{Q}_2]$*, where*

$\mathbf{N}_2$ *consists of all columns* $\mathbf{p}$ *of* $\mathbf{P}_2$ *with* $\mathrm{cdeg}_{-\vec{t}}\mathbf{p} \le 0$. *Let* $\mathbf{R} = [\mathbf{N}_1\mathbf{Q}_2, \mathbf{Q}_1]$ *and* $\mathbf{R}^r = \mathrm{colRev}\left(\mathbf{R}, -\vec{s}, \mathrm{cdeg}_{-\vec{s}}\mathbf{R}\right)$. *Then* $\left[\mathbf{F}^T, \mathbf{R}^r\right]$ *is a unimodular matrix.*

Proof. We know from Lemma 8.14 that $\mathbf{P}_1^r = \mathrm{colRev}\left(\mathbf{P}_1, -\vec{s}, \mathrm{cdeg}_{-\vec{s}}\mathbf{P}_1\right)$ and $\mathbf{P}_2^r = \mathrm{colRev}\left(\mathbf{P}_1, \vec{t}, \mathrm{cdeg}_{\vec{t}}\mathbf{P}_2\right)$ are both unimodular. Hence $\mathbf{P}_1^r \cdot \mathrm{diag}\left([\mathbf{P}_2^r, I]\right) = [\mathbf{N}_1^r\mathbf{N}_2^r, \mathbf{N}_1^r\mathbf{Q}_2^r, \mathbf{Q}_1] = [\mathbf{N}_1^r\mathbf{N}_2^r, \mathbf{R}^r]$ is unimodular, where $\mathbf{N}_1\mathbf{N}_2$ is a kernel basis of $\mathbf{M}$. The result follows by the same reasoning as in Lemma 8.15. □

8.3 Efficient Computation

Lemma 8.16 provides a way to correctly compute a unimodular completion of $\mathbf{F}$. Our next task is to make sure it can be computed efficiently and analyze its computational cost. We already know that a $(\mathbf{F}^r, \vec{s})$-kernel basis can be computed with a cost of $O^\sim(n^\omega s)$. Therefore, it only remains to check the cost of the order basis computations. Note that the non-uniform order makes our problem here a little more difficult. But on the other hand, the output basis has its $-\vec{s}$-column degrees bounded by 1, which is a consequence of the fact $\mathbf{M}$ is a $\vec{s}$-minimal kernel basis, as shown in Lemma 8.20 below. But we first need a few general lemmas on the degree bounds of order bases and kernel bases.

First, the following lemma is a simple extension of Lemma 3.2 for dealing with nonuniform orders.

Lemma 8.17. *Given an input matrix* $\mathbf{A} \in \mathbb{K}^{m\times n}[x]$, *a shift* $\vec{u} \in \mathbb{Z}^n$, *and an order list* $\vec{\sigma} \in \mathbb{Z}^m$. *Let* $\vec{v}$ *be the* $\vec{u}$*-column degrees of a* $(\mathbf{A}, \vec{\sigma}, \vec{u})$*-basis. Then* $\sum \vec{t} \le \sum \vec{s} + \sum \vec{\sigma}$.

Proof. The sum of the $\vec{s}$-column degrees is $\sum \vec{s}$ at order $[0, \ldots, 0]$, since the identity matrix is a $(\mathbf{A}, [0, \ldots, 0], \vec{s})$-basis. This sum increases by 1 for each order increase of each row. The total number of order increases required for all rows is at most

$\sum \vec{\sigma}$. Note that from Theorem 3.20, we can work with just one row at a time to increase its order in the order basis computation. $\square$

The following lemma extends Theorem 5.2 to give a bound based on the shifted column degrees or shifted row degrees, instead of just the column degrees of the input matrix.

Lemma 8.18. *If* $\mathbf{A} \in \mathbb{K}^{m\times n}[x]$ *has* $\mathrm{rdeg}_{\vec{u}} \mathbf{A} \le \vec{v}$ *or equivalently* $\mathrm{cdeg}_{-\vec{v}} \mathbf{A} \le -\vec{u}$, *then any* $(\mathbf{A}, -\vec{u})$*-kernel basis has* $-\vec{u}$*-column degrees bounded by* $\sum \vec{v} - \sum \vec{u}$.

Proof. Let $\mathbf{P} = \left[\mathbf{B}, \bar{\mathbf{B}}\right]$ be a $(\mathbf{A}, \vec{v} + [\sigma, \dots, \sigma], -\vec{u})$-basis containing a kernel basis, $\mathbf{B}$, of $\mathbf{A}$. Then $\sum \mathrm{cdeg}_{-\vec{u}} \mathbf{P}$ is at least $m\sigma + \sum \vec{v} - \sum \vec{u}$. We also know that $\sum \mathrm{cdeg}_{-\vec{u}} \bar{\mathbf{B}} \ge \sum \mathrm{cdeg}_{-\vec{v}} \mathbf{A}\bar{\mathbf{B}}$, but $\mathrm{cdeg}\, \mathbf{A}\bar{\mathbf{B}} \ge \vec{v} + [\sigma, \dots, \sigma]$ or $\sum \mathrm{cdeg}_{-\vec{v}} \mathbf{A}\bar{\mathbf{B}} \ge m\sigma$, therefore $\sum \mathrm{cdeg}_{-\vec{u}} \bar{\mathbf{B}} \ge m\sigma$. It follows that $\sum \mathrm{cdeg}_{-\vec{u}} \mathbf{B} \le m\sigma + \sum \vec{v} - \sum \vec{u} - m\sigma = \sum \vec{v} - \sum \vec{u}$. $\square$

When the matrix $\mathbf{A}$ is also a $\left(\mathbf{B}^T, \vec{u}\right)$-kernel basis, as in our case, the bound in fact becomes tight.

Lemma 8.19. *Let* $\mathbf{A} \in \mathbb{K}^{m\times n}[x]$ *and* $\mathbf{B} \in \mathbb{K}^{n\times(n-m)}[x]$. *If* $\mathbf{B}$ *is a* $(\mathbf{A}, -\vec{u})$*-kernel basis with* $\mathrm{cdeg}_{-\vec{u}} \mathbf{B} = \vec{w}$ *and* $\mathbf{A}^T$ *is a* $\left(\mathbf{B}^T, \vec{u}\right)$*-kernel basis with* $\mathrm{rdeg}_{\vec{u}} \mathbf{A} = \vec{v}$, *then* $\sum \vec{w} = \sum \vec{v} - \sum \vec{u}$.

Proof. This follows from Lemma 8.18, which gives $\sum \vec{w} \le \sum \vec{v} - \sum \vec{u}$ and also $\sum \vec{v} \le \sum \vec{w} + \sum \vec{u}$ in the reverse direction. $\square$

From Lemma 7.6, we know that any $\left(\mathbf{M}^T, \vec{b}+1, -\vec{s}\right)$-basis contains a $\left(\mathbf{M}^T, -\vec{s}\right)$-kernel basis whose $-\vec{s}$-column degrees bounded by 0. The following lemma shows that the remaining part of the $\left(\mathbf{M}^T, \vec{b}+1, -\vec{s}\right)$-basis has degrees bounded by 1.

Lemma 8.20. *Let* $\mathbf{F}^r = \mathrm{rowRev}\left(\mathbf{F}, -\vec{s}, 0\right)$ *and* $\mathbf{M}$ *be a* $(\mathbf{F}^r, \vec{s})$*-kernel basis with* $\mathrm{cdeg}_{\vec{s}} \mathbf{M} = \vec{b}$ *as before. Let* $\mathbf{P}$ *be a* $\left(\mathbf{M}^T, \vec{b}+1, -\vec{s}\right)$*-basis. Then* $\mathrm{cdeg}_{-\vec{b}-1} \mathbf{M}^T\mathbf{P}_2 = [0, \dots, 0]$ *and* $\mathrm{cdeg}_{-\vec{s}} \mathbf{P}_2 = [1, \dots, 1]$.

Proof. We already know that $\mathbf{P}$ contains a $\left(\mathbf{M}^T, -\vec{s}\right)$-kernel basis. Let this kernel basis be $\mathbf{P}_1$ in $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$. We know that $\sum \mathrm{cdeg}_{-\vec{s}}\mathbf{P} = -\sum \vec{s} + \sum \vec{b} + n - m$ and for the kernel basis $\mathbf{P}_1$ in $\mathbf{P}$, we know $\sum \mathrm{cdeg}_{-\vec{s}}\mathbf{P}_1 = \sum \vec{b} - \sum \vec{s}$ from Lemma 8.19. Therefore, $\sum \mathrm{cdeg}_{-\vec{s}}\mathbf{P}_2 = n - m$. It follows that $\sum \mathrm{cdeg}_{-\vec{b}}\mathbf{M}^T\mathbf{P}_2 \leq \sum \mathrm{cdeg}_{-\vec{s}}\mathbf{P}_2 = n - m$, or $\sum \mathrm{cdeg}_{-\vec{b}-1}\mathbf{M}^T\mathbf{P}_2 = 0$. But since $\mathbf{P}_2$ is nonzero and has order $\left(\mathbf{F}, \vec{b} + 1\right)$, we have $\mathrm{cdeg}_{-\vec{b}-1}\mathbf{M}^T\mathbf{P}_2 \geq [0, \ldots, 0]$, implying $\sum \mathrm{cdeg}_{-\vec{b}-1}\mathbf{M}^T\mathbf{P}_2 \geq 0$. It follows that $\sum \mathrm{cdeg}_{-\vec{b}-1}\mathbf{M}^T\mathbf{P}_2 = 0$, hence $\mathrm{cdeg}_{-\vec{b}-1}\mathbf{M}^T\mathbf{P}_2 = [0, \ldots, 0]$ or $\mathrm{cdeg}_{-\vec{b}}\mathbf{M}^T\mathbf{P}_2 = [1, \ldots, 1]$. Combining this with $\sum \mathrm{cdeg}_{-\vec{b}}\mathbf{M}^T\mathbf{P}_2 \leq \sum \mathrm{cdeg}_{-\vec{s}}\mathbf{P}_2 = n - m$ we then get $\mathrm{cdeg}_{-\vec{s}}\mathbf{P}_2 = [1, \ldots, 1]$. □

We are now ready to look at the algorithm for computing a $\left(\mathbf{M}^T, \vec{b} + 1, -\vec{s}\right)$-basis, given in Algorithm 8.1. We follow the same process as in Section 7.1. We assume without loss of generality that the rows of $\mathbf{M}^T$ are arranged in decreasing $\vec{s}$-row degrees. We divide $\mathbf{M}^T$ into $\log k$ row blocks according to the $\vec{s}$-row degrees of its rows. Let

$$\mathbf{M}^T = \left[\mathbf{M}_1^T, \mathbf{M}_2^T, \cdots, \mathbf{M}_{\log k - 1}^T, \mathbf{M}_{\log k}^T\right]$$

with $\mathbf{M}_{\log k}, \mathbf{M}_{\log k - 1}, \cdots, \mathbf{M}_2, \mathbf{M}_1$ having $\vec{s}$-row degrees in the range

$$[0, 2\xi/k]\,,\ (2\xi/k, 4\xi/k],\ (4\xi/k, 8\xi/k],\ ...,\ (\xi/4, \xi/2],\ (\xi/2, \xi].$$

Let $\vec{\sigma}_i = [\xi/2^{i-1} + 1, \ldots, \xi/2^{i-1} + 1]$ with the same dimension as the row dimension of $\mathbf{M}_i$. Let $\vec{\sigma} = [\vec{\sigma}_{\log k}, \vec{\sigma}_{\log k - 1}, \ldots, \vec{\sigma}_1]$ be the order in the order basis computation. For simplicity, instead of using $\mathbf{M}^T$ as the input matrix, we use

$$\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}_1 \\ \vdots \\ \hat{\mathbf{M}}_{\log k} \end{bmatrix} = x^{\vec{\sigma} - \vec{b} - 1} \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_{\log k} \end{bmatrix} = x^{\vec{\sigma} - \vec{b} - 1}\mathbf{M}$$

instead, so that a $\left(\hat{\mathbf{M}}, \vec{\sigma}, -\vec{s}\right)$-basis is a $\left(\mathbf{M}, \vec{b}+1, -\vec{s}\right)$-basis.

We now do a series of order basis computations in order to compute a unimodular completion of $\mathbf{F}$ based on Lemma 8.16.

Let $\vec{s}_1 = \vec{s}$. First we compute an $\left(\hat{\mathbf{M}}_1, \vec{\sigma}_1, -\vec{s}_1\right)$-basis $\mathbf{P}_1 = [\mathbf{N}_1, \mathbf{Q}_1]$, where $\mathbf{N}_1$ is a $\left(\hat{\mathbf{M}}_1, -\vec{s}_1\right)$-kernel basis. This computation can be done using Algorithm 4.1 with a cost of $O^{\sim}(n^\omega s)$, where $s = \xi/n$.

Let $\tilde{\mathbf{N}}_1 = \mathbf{N}_1$. Let $\vec{s}_2 = -\operatorname{cdeg}_{-\vec{s}} \mathbf{N}_1$ and $\vec{t}_2 = -\operatorname{cdeg}_{-\vec{s}} \mathbf{Q}_1$. We then compute an $\left(\hat{\mathbf{M}}_2\tilde{\mathbf{N}}_1, \vec{\sigma}_2, -\vec{s}_2\right)$-basis $\mathbf{P}_2 = [\mathbf{N}_2, \mathbf{Q}_2]$ with $\vec{s}_3 = -\operatorname{cdeg}_{-\vec{s}_2} \mathbf{N}_2$ and $\vec{t}_3 = -\operatorname{cdeg}_{-\vec{s}_2} \mathbf{Q}_2$. Let $\tilde{\mathbf{N}}_2 = \tilde{\mathbf{N}}_1\mathbf{N}_2$

Continue this process, at step i, we compute an $\left(\hat{\mathbf{M}}_i\tilde{\mathbf{N}}_{i-1}, \vec{\sigma}_i, -\vec{s}_i\right)$-basis $\mathbf{P}_i = [\mathbf{N}_i, \mathbf{Q}_i]$. Let $\tilde{\mathbf{N}}_i = \prod_{j=1}^{i} \mathbf{N}_i = \tilde{\mathbf{N}}_{i-1}\mathbf{N}_i$. Note that $\tilde{\mathbf{N}}_{\log k}$ is a $(\mathbf{M}, -\vec{s})$-kernel basis. Let

$$\mathbf{R} = \left[\mathbf{Q}_1, \tilde{\mathbf{N}}_1\mathbf{Q}_2, \ldots, \tilde{\mathbf{N}}_{\log k-2}\mathbf{Q}_{\log k-1}, \tilde{\mathbf{N}}_{\log k-1}\mathbf{Q}_{\log k}\right]$$

, and $\mathbf{R}^r = \operatorname{colRev}\left(\mathbf{R}, -\vec{s}, \operatorname{cdeg}_{-\vec{s}} \mathbf{R}\right)$, then from Lemma 8.16 we can conclude that $\left[\mathbf{F}^T, \mathbf{R}^r\right]$ is a unimodular matrix.

We still need to check the cost of the multiplications $\hat{\mathbf{M}}_i\tilde{\mathbf{N}}_{i-1}$, $\tilde{\mathbf{N}}_{i-1}\mathbf{N}_i$, and $\tilde{\mathbf{N}}_{i-1}\mathbf{Q}_i$.

Lemma 8.21. *The multiplications $\hat{\mathbf{M}}_i\tilde{\mathbf{N}}_{i-1}$ can be done with a cost of $O^{\sim}(n^\omega s)$.*

Proof. The dimension of $\hat{\mathbf{M}}_i$ is bounded by $2^{i-1}\times n$ and $\sum \operatorname{rdeg}_{\vec{s}} \hat{\mathbf{M}}_i \le 2^{i-1}\cdot\xi/2^{i-1} = \xi$. We also have $\operatorname{cdeg}_{-\vec{s}} \tilde{\mathbf{N}}_{i-1} \le 0$, or equivalently, $\operatorname{rdeg} \tilde{\mathbf{N}}_{i-1} \le \vec{s}$. We can now use Theorem 5.6 to multiply $\tilde{\mathbf{N}}_{i-1}^T$ and $\hat{\mathbf{M}}_i^T$ with a cost of $O^{\sim}(n^\omega s)$. □

Lemma 8.22. *The multiplication $\tilde{\mathbf{N}}_{i-1}\mathbf{N}_i$ can be done with a cost of $O^{\sim}(n^\omega s)$.*

Proof. We know $\operatorname{cdeg}_{-\vec{s}} \tilde{\mathbf{N}}_{i-1} = -\vec{s}_i$, and $\operatorname{cdeg}_{-\vec{s}_i} \mathbf{N}_i = -\vec{s}_{i+1} \le 0$. In other words, $\operatorname{rdeg} \mathbf{N}_i \le \vec{s}_i$, and $\operatorname{rdeg}_{\vec{s}_i} \tilde{\mathbf{N}}_{i-1} \le \vec{s}$, hence we can again use Theorem 5.6 to multiply $\mathbf{N}_i^T$ and $\tilde{\mathbf{N}}_{i-1}^T$ with a cost of $O^{\sim}(n^\omega s)$. □

Algorithm 8.1 unimodularCompletion($\mathbf{F}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ with full row rank; $\vec{s}$ is initially set to the column degrees of
    $\mathbf{F}$. It keeps track of the degrees.
Output: $\mathbf{G} \in \mathbb{K}[x]^{(n-m)\times n}$ such that $\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}$ is unimodular.
 1: $\vec{s} := \operatorname{cdeg}\mathbf{F}$;
 2: $\mathbf{F}^r := \operatorname{rowRev}(\mathbf{F}, -\vec{s}, 0)$;
 3: $\mathbf{M} :=$ minimalKernelBasis $(\mathbf{F}^r, \vec{s})$; $\vec{b} := \operatorname{cdeg}_{\vec{s}}\mathbf{M}$; $k := n - m$;
 4: $\left[\mathbf{M}_1^T, \mathbf{M}_2^T, \cdots, \mathbf{M}_{\log k-1}^T, \mathbf{M}_{\log k}^T\right] := \mathbf{M}$, with $\mathbf{M}_{\log k}, \mathbf{M}_{\log k-1}, \cdots, \mathbf{M}_2, \mathbf{M}_1$ hav-
    ing $\vec{s}$-row degrees in the range $[0, 2\xi/k], (2\xi/k, 4\xi/k], ..., (\xi/4, \xi/2], (\xi/2, \xi]$.
 5: for $i$ from $1$ to $\log k$ do
 6:     $\vec{\sigma}_i := [\xi/2^{i-1}+1, \ldots, \xi/2^{i-1}+1]$, with the number of entries matches the row
        dimension of $\mathbf{M}_i$;
 7: end for
 8: $\vec{\sigma} := [\vec{\sigma}_{\log k}, \vec{\sigma}_{\log k-1}, \ldots, \vec{\sigma}_1]$;
 9: $\hat{\mathbf{M}} := x^{\vec{\sigma}-\vec{b}-1}\mathbf{M}$;
10: $\mathbf{N}_0 := I_n$; $\tilde{\mathbf{N}}_0 := I_n$;
11: for $i$ from $1$ to $\log k$ do
12:     $\vec{s}_i := -\operatorname{cdeg}_{-\vec{s}}\mathbf{N}_{i-1}$; (note $\vec{s}_1 = \vec{s}$)
13:     $\mathbf{P}_i :=$ unbalancedFastOrderBasis $\left(\hat{\mathbf{M}}_i\tilde{\mathbf{N}}_{i-1}, \vec{\sigma}_i, -\vec{s}_i\right)$;
14:     $[\mathbf{N}_i, \mathbf{Q}_i] := \mathbf{P}_i$, where $\mathbf{N}_i$ is a $\left(\hat{\mathbf{M}}_i, -\vec{s}_i\right)$-nullspace basis;
15:     $\tilde{\mathbf{N}}_i := \tilde{\mathbf{N}}_{i-1}\cdot\mathbf{N}_i$;
16:     $\mathbf{R} := \left[\mathbf{R}, \tilde{\mathbf{N}}_{i-1}\mathbf{Q}_i\right]$;
17: end for
18: $\mathbf{R}^r := \operatorname{colRev}\left(\mathbf{R}, -\vec{s}, \operatorname{cdeg}_{-\vec{s}}\mathbf{R}\right)$;
19: return $(\mathbf{R}^r)^T$
```

Lemma 8.23. *The multiplication $\tilde{\mathbf{N}}_{i-1}\mathbf{Q}_i$ can be done with a cost of $O^{\sim}(n^{\omega}s)$.*

Proof. We know $\operatorname{cdeg}_{-\vec{s}_i}\mathbf{Q}_i \le \max\operatorname{cdeg}_{\vec{s}}\mathbf{P} = 1$, or equivalently, $\operatorname{rdeg}\mathbf{Q}_i \le \vec{s}_i + 1$. But we also know that this $\mathbf{Q}_i$ from the order basis computation has a factor xI. Therefore, $\operatorname{rdeg}(\mathbf{Q}_i/x) \le \vec{s}_i$. In addition, $\operatorname{rdeg}_{\vec{s}_i}\tilde{\mathbf{N}}_{i-1} \le \vec{s}$ as before. So we can again use Theorem 5.6 to multiply $\mathbf{Q}_i^T$ and $\tilde{\mathbf{N}}_{i-1}^T$ with a cost of $O^{\sim}(n^{\omega}s)$. $\square$

Theorem 8.24. *A unimodular completion of $\mathbf{F}$ can be computed with a cost of $O^{\sim}(n^{\omega}s)$ field operations.*

Chapter 9

Diagonal Entries of Hermite Normal Form and Determinant

In this chapter, we consider the problem of computing the diagonal entries of the Hermite normal form and the determinant of a nonsingular input matrix $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$ with column degrees $\vec{s}$. Storjohann [2002, 2003] gave an efficient Las Vegas for computing the determinant. Here we give a efficient deterministic algorithm that costs $O^{\sim}(n^{\omega}s)$ field operations, where $s = \sum \vec{s}/n$ The computation is done by using the column basis and kernel basis computation to compute the diagonal entries of the Hermite form of $\mathbf{F}$, and then multiply these diagonal entries.

Consider unimodularly transforming $\mathbf{F}$ to

$$\mathbf{FU} = \mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & 0 \\ * & \mathbf{G}_2 \end{bmatrix}, \tag{9.1}$$

After this unimodular transformation, which eliminated the top right block of $\mathbf{G}$, the matrix is now closer to the Hermite normal form of $\mathbf{F}$. This procedure can then be applied recursively to $\mathbf{G}_1$ and $\mathbf{G}_2$, until the matrices reaching dimension 1, which then gives the diagonal entries of the Hermite normal form of $\mathbf{F}$.

Although this procedure correctly computes the diagonal entries of the Hermite normal form of $\mathbf{F}$, a major problem is that the degree of the unimodular $\mathbf{U}$ can be too large for $\mathbf{U}$ to be efficiently computed. However, with the tools we have developed in the earlier chapters, we can efficiently compute $\mathbf{G}_1$ and $\mathbf{G}_2$ without computing $\mathbf{U}$.

If we separate $\mathbf{F}$ to $\mathbf{F} = \begin{bmatrix} \mathbf{F}_U \\ \mathbf{F}_D \end{bmatrix}$, each has full-rank as $\mathbf{F}$ is assumed to be nonsingular, and also separate $\mathbf{U}$ to $\mathbf{U} = \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix}$, where the column dimension of $\mathbf{U}_L$ matches the row dimension of $\mathbf{F}_U$, then

$$\mathbf{F}\mathbf{U} = \begin{bmatrix} \mathbf{F}_U \\ \mathbf{F}_D \end{bmatrix} \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix} = \mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & 0 \\ * & \mathbf{G}_2 \end{bmatrix}.$$

Notice that the matrix $\mathbf{G}_1$ is nonsingular and is therefore just a column basis of $\mathbf{F}_U$, and can be efficiently computed using Algorithm 7.2. To compute $\mathbf{G}_2 = \mathbf{F}_D\mathbf{U}_R$, notice that the matrix $\mathbf{U}_R$ is a right kernel basis of $\mathbf{F}$, which makes the top right block of $\mathbf{G}$ zero. As we have seen from Lemma 7.1, the kernel basis $\mathbf{U}_R$ can be replaced by any other kernel basis of $\mathbf{F}$ to give another unimodular matrix that also works.

Lemma 9.1. *Given a polynomial matrix* $\mathbf{F} = \begin{bmatrix} \mathbf{F}_U \\ \mathbf{F}_D \end{bmatrix}$. *If* $\mathbf{G}_1$ *is a column basis of* $\mathbf{F}_U$ *and* $\mathbf{N}$ *is a kernel basis of* $\mathbf{F}_U$, *then there is a unimodular matrix* $\mathbf{U} = [*, \mathbf{N}]$ *such that*

$$\mathbf{F}\mathbf{U} = \begin{bmatrix} \mathbf{G}_1 & \\ * & \mathbf{G}_2 \end{bmatrix},$$

where $\mathbf{G}_2 = \mathbf{F}_D\mathbf{N}$. *If* $\mathbf{F}$ *is square nonsingular, then* $\mathbf{G}_1$ *and* $\mathbf{G}_2$ *are also square nonsingular.*

Note that we do not compute the blocks represented by the symbol $*$, which

Algorithm 9.1 hermiteDiagonal($\mathbf{F}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$ is nonsingular.
Output: $\mathbf{d} \in \mathbb{K}[x]^{n}$ a list of diagonal entries of the Hermite normal form of $\mathbf{F}$.
1: $\begin{bmatrix}\mathbf{F}_U\\ \mathbf{F}_D\end{bmatrix} := \mathbf{F}$, with $\mathbf{F}_U$ consists of the top $\lceil n/2\rceil$ rows of $\mathbf{F}$;
2: if $n = 1$ then return $\mathbf{F}$; endif;
3: $\mathbf{G}_1 :=$ colBasis($\mathbf{F}_U$);
4: $\mathbf{N} :=$ minimalKernelBasis ($\mathbf{F}_U, \operatorname{cdeg}\mathbf{F}$);
5: $\mathbf{G}_2 := \mathbf{F}_D\mathbf{N}$;
6: $\mathbf{d}_1 :=$ hermiteDiagonal($\mathbf{G}_1$);$\mathbf{d}_2 :=$ hermiteDiagonal($\mathbf{G}_2$);
7: return $[\mathbf{d}_1, \mathbf{d}_2]$;
```

may have very large degrees and cannot be computed efficiently.

Lemma 9.1 allows us to compute $\mathbf{G}_1$ and $\mathbf{G}_2$ independently without computing the unimodular matrix. $\mathbf{G}_1$ can be computed using the method from Chapter 7, while the kernel basis computation from Chapter 5 can be used to compute a kernel basis $\mathbf{N}$ of $\mathbf{F}_U$, which can then be used to compute $\mathbf{G}_2 = \mathbf{F}_D\mathbf{N}$.

After $\mathbf{G}_1$ and $\mathbf{G}_2$ are computed, we can repeat the same process on each of these two matrices, which now have lower dimensions, until the dimension becomes one. This procedure of computing the diagonal entries gives Algorithm 9.1

9.1 Computing the Determinant

The product of the diagonal entries computed from Algorithm 9.1 is an associate of the determinant, that is, the product equals $a \det \mathbf{F}$ for some $c \in \mathbb{K}$, since the unimodular matrices that eliminate the top right blocks may not have its determinant equal to 1. Therefore, to get the determinant of $\mathbf{F}$, we need to scale the product of the diagonal entries by c^{-1}, where c is the determinant of the unimodular matrix that transforms $\mathbf{F}$ to the diagonal entries from the algorithm.

Lemma 9.2. *Let* $\mathbf{U} = [\mathbf{U}_L, \mathbf{U}_R]$ *be a unimodular that eliminates the top right block* *of* $\mathbf{F}$ *as before, that is,*

$$\mathbf{F}\mathbf{U} = \begin{bmatrix} \mathbf{F}_U \\ \mathbf{F}_D \end{bmatrix} \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_R \end{bmatrix} = \begin{bmatrix} \mathbf{G}_1 & 0 \\ * & \mathbf{G}_2 \end{bmatrix} = \mathbf{G}.$$

Let $\mathbf{V} = \begin{bmatrix} \mathbf{V}_U \\ \mathbf{V}_D \end{bmatrix} = \mathbf{U}^{-1}$. *Let* $U_R = \mathbf{U}_R \mod x$, $V_U = \mathbf{V}_U \mod x$, *and* $U_L^* \in \mathbb{K}^{n \times *}$ *is a matrix that gives a unimodular completion* $U^* = [U_L^*, U_R]$ *of* U_R *with* $\det U^* = a \in \mathbb{K}$. *Then* $\det \mathbf{F} = \det \mathbf{G} \det (V_U U_L^*)/a$.

Proof. Since $\det \mathbf{F} = \det \mathbf{G} \det \mathbf{V}$, we just need to show that $\det \mathbf{V} = \det (V_U U_L^*)/a$. In fact, we just need to check the degree 0 coefficient matrix $V = \mathbf{V} \mod x$ to show that $\det V = \det (V_U U_L^*)/a$, since $\mathbf{V}$ is unimodular, which makes $\det V = \det \mathbf{V}$. Consider now

$$\begin{aligned} \det V \det U^* &= \det (VU^*) \\ &= \det \left(\begin{bmatrix} V_U \\ V_D \end{bmatrix} \begin{bmatrix} U_L^* & U_R \end{bmatrix} \right) \\ &= \det \left(\begin{bmatrix} V_U U_L^* & 0 \\ * & I \end{bmatrix} \right) \\ &= \det (V_U U_L^*), \end{aligned}$$

hence $\det V = \det (V_U U_L^*)/a$. $\square$

Lemma 9.2 requires us to compute a unimodular completion of a matrix U_R, which is a very simple special case of the unimodular completion from Chapter 8 and can be obtained easily from the unimodular matrix that transforms V_U to its reduced column echelon form. Such unimodular matrix can be computed using the Gauss Jordan transform algorithm from Storjohann [2000] on $\bar{F}$ with a cost of

Algorithm 9.2 hermiteDiagonalWithScale($\mathbf{F}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$ is nonsingular.
Output: $\mathbf{d} \in \mathbb{K}[x]^n$ a list of diagonal entries of the Hermite normal form of $\mathbf{F}$,
    and $c$ a scaling factor to be multiplied to obtain the determinant.
1: $\begin{bmatrix}\mathbf{F}_U\\ \mathbf{F}_D\end{bmatrix} := \mathbf{F}$, with $\mathbf{F}_U$ consists of the top $\lceil n/2 \rceil$ rows of $\mathbf{F}$;
2: if $n = 1$ then return $\mathbf{F}, 1$; endif;
3: $\mathbf{G}_1, \mathbf{U}_R, \mathbf{V}_L :=$ colBasis($\mathbf{F}_U$); (Here we make colBasis() also return the kernel
    bases it computed.)
4: $\mathbf{G}_2 := \mathbf{F}_D \mathbf{U}_R$;
5: $U_R := \mathbf{U}_R \mod x$; $V_U := \mathbf{V}_U \mod x$;
6: compute $U_L^* \in \mathbb{K}^{n\times *}$ , a matrix that gives a unimodular completion $U^* =$
    $[U_L^*, U_R]$;
7: $\mathbf{d}_1, c_1 :=$ hermiteDiagonal($\mathbf{G}_1$);$\mathbf{d}_2, c_2 =$ hermiteDiagonal($\mathbf{G}_2$);
8: return $[\mathbf{d}_1, \mathbf{d}_2], c_1 c_2 \det(V_U U_L^*) / \det(U)^*$;
```

$O(nm^{\omega-1})$.

We can now compute the actual determinant of $\mathbf{F}$ by simply computing the scaling factor at each step. The updated algorithm that also computes the scaling factor is given in Algorithm 9.2.

9.2 Computational Cost

Let us look at the computational cost of Algorithm 9.1 and Algorithm 9.2.

Theorem 9.3. *Algorithm 9.1 and Algorithm 9.2 cost $O^{\sim}(n^{\omega}s)$ field operations to compute the diagonal entries for the Hermite normal form of a nonsingular matrix $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$, where s is the average column degree of $\mathbf{F}$.*

Proof. The three main operations are computing a column basis of $\mathbf{F}_U$, computing a kernel basis $\mathbf{N}$ of $\mathbf{F}_U$, and the matrix multiplication $\mathbf{F}_D\mathbf{N}$.

For the column basis computation, by Theorem 7.19 we know that a column basis $\mathbf{G}_1$ of $\mathbf{F}_U$ can be computed with a cost of $O^{\sim}(n^{\omega}s)$. By Lemma 7.14 the column degrees of the computed column basis $\mathbf{G}_1$ are also bounded by the original column degrees $\vec{s}$.

For the kernel basis computation, it also costs $O^{\sim}(n^{\omega}s)$ to compute a $\vec{s}$-minimal kernel basis $\mathbf{N}$ of $\mathbf{F}_U$ from Theorem 5.18. The sum of the $\vec{s}$-column degrees of the output kernel basis $\mathbf{N}$ is bounded by $\sum \vec{s}$ by Theorem 5.2.

Finally for the matrix multiplication $\mathbf{F}_D\mathbf{N}$, since the sum of the column degrees of $\mathbf{F}_D$ and the sum of the $\vec{s}$-column degrees of $\mathbf{N}$ are both bounded by $\sum \vec{s}$, Theorem 5.6 applies and the multiplication can be done with a cost of $O^{\sim}(n^{\omega}s)$.

Now if we let the cost of Algorithm 9.1 be $g(n)$ for a input matrix of dimension n, then we have the recurrence relation

$$g(n) \in O^{\sim}(n^{\omega-1}) + g(\lceil n/2 \rceil) + g(\lfloor n/2 \rfloor),$$

which is the same as in Theorem 6.3 for computing the matrix inverse. Therefore, we also get $g(n) = O^{\sim}(n^{\omega}s)$ as in the inverse computation.

The only extra cost of Algorithm 9.2 is the cost unimodular completion of U_R, which is just $O(n^{\omega})$. So it has the same cost as Algorithm 9.1. □

Corollary 9.4. *The determinant of a nonsingular matrix $\mathbf{F} \in \mathbb{K}[x]^{n\times n}$ can be computed with a cost of $O^{\sim}(n^{\omega}s)$ field operations, where s is the minimum of the average column degree and the average row degree of the input matrix.*

Proof. We can just use Algorithm 9.2 to compute the diagonal entries of the Hermite normal form of either $\mathbf{F}$ or $\mathbf{F}^T$ with the scaling factor, and then multiply the diagonal entries with the scaling factor. □

Chapter 10

Hermite Normal Form

In Chapter 9, we have shown how the diagonal entries of the Hermite normal form of a nonsingular input matrix $\mathbf{F} \in \mathbb{K}[x]^{n \times n}$ can be computed efficiently. In this Chapter, we consider the problem of computing the complete Hermite normal form $\mathbf{H}$ of $\mathbf{F}$. Gupta and Storjohann [2011], Gupta [2011] gave a randomized Las Vegas algorithm that costs $O^{\sim}(n^{\omega} d)$ to compute the Hermite normal form. We make use of some of their ideas and follow a similar path. But we do not use Smith normal form and our algorithm is deterministic.

For simplicity, we assume $\mathbf{F}$ is already column reduced and has column degrees $\vec{d} = [d_1, \ldots, d_n]$ and let $d = \max \vec{d}$. Let $\vec{s} = [s_1, \ldots, s_n]$ be the degrees of the diagonal entries of the Hermite form $\mathbf{H}$. Then if $\vec{u} = [\max \vec{s}, \ldots, \max \vec{s}]$ with n entries, we can obtain the Hermite normal form from a $[-\vec{u}, -\vec{s}]$-minimal kernel basis of $[\mathbf{F}, I]$.

Lemma 10.1. *If* $\begin{bmatrix} \mathbf{V} \\ \mathbf{G} \end{bmatrix}$ *is a* $[-\vec{u}, -\vec{s}]$*-minimal kernel basis of* $[\mathbf{F}, -I]$*, where each block is* $n \times n$ *square, then* $\mathbf{G}$ *is unimodularly equivalent with the Hermite normal form* $\mathbf{H}$ *of* $\mathbf{F}$ *and has row degrees* $\vec{s}$*, the same as the row degrees of* $\mathbf{H}$.

Proof. Notice that the unimodular matrix $\mathbf{U}$ satisfying $\mathbf{F}\mathbf{U} = \mathbf{H}$ has $\vec{d}$-column degrees bounded by $\max \vec{s}$ from the predictable-degree property Lemma 2.17, which means $\mathbf{U}$ has degree bounded by $\max \vec{s}$, or equivalently, $\operatorname{cdeg}_{-\vec{u}} \mathbf{U} \leq 0$, hence $\operatorname{cdeg}_{[-\vec{u},-\vec{s}]} \begin{bmatrix} \mathbf{U} \\ \mathbf{H} \end{bmatrix} = \operatorname{cdeg}_{-\vec{s}} \mathbf{H} = 0$, making $\begin{bmatrix} \mathbf{U} \\ \mathbf{H} \end{bmatrix}$ $[-\vec{u},-\vec{s}]$-column reduced and a $[-\vec{u},-\vec{s}]$-minimal kernel basis of $[\mathbf{F},-I]$. If we compute a $[-\vec{u},-\vec{s}]$-minimal kernel basis $\begin{bmatrix} \mathbf{V} \\ \mathbf{G} \end{bmatrix}$ of $[\mathbf{F},-I]$, we know that $\begin{bmatrix} \mathbf{V} \\ \mathbf{G} \end{bmatrix}$ is unimodularly equivalent to $\begin{bmatrix} \mathbf{U} \\ \mathbf{H} \end{bmatrix}$, implying that $\mathbf{V}$ is also unimodular. The minimality also ensures that $\operatorname{cdeg}_{[-\vec{u},-\vec{s}]} \begin{bmatrix} \mathbf{V} \\ \mathbf{G} \end{bmatrix} = \operatorname{cdeg}_{[-\vec{u},-\vec{s}]} \begin{bmatrix} \mathbf{U} \\ \mathbf{H} \end{bmatrix} = 0$, implying $\operatorname{cdeg}_{-\vec{s}} \mathbf{G} \leq 0 = \operatorname{cdeg}_{-\vec{s}} \mathbf{H}$. But the minimality of $\mathbf{H}$ ensures that $\operatorname{cdeg}_{-\vec{s}} \mathbf{G} = \operatorname{cdeg}_{-\vec{s}} \mathbf{H} = 0$, or equivalently, $\operatorname{rdeg} \mathbf{G} = \operatorname{rdeg} \mathbf{H} = \vec{s}$. □

Knowing that $\mathbf{G}$ has the same row degrees as $\mathbf{H}$ and is unimodularly equivalent with $\mathbf{H}$, the Hermite form $\mathbf{H}$ can then be obtained from $\mathbf{G}$ using Lemma 8 from [Gupta and Storjohann, 2011], restated as follows:

Lemma 10.2. *If the Hermite normal form* $\mathbf{H}$ *of* $\mathbf{F}$ *is a column basis of a matrix* $\mathbf{A} \in \mathbb{K}[x]^{n \times k}$, *and has the same row degrees as* $\mathbf{A}$, *then the matrix* $U \in \mathbb{K}^{n \times n}$ *putting* $\operatorname{lcoeff}\left(x^{-\vec{s}} \mathbf{A}\right) U$ *in reduced column echelon form also gives the Hermite normal form* $\mathbf{H}$ *as the principal* $n \times n$ *submatrix of* $\mathbf{A}U$.

Proof. This follows from the fact that $\mathbf{H}$ and $\mathbf{A}$ all have uniform $-\vec{s}$ column degrees 0, which allows their relationship to be completely determined by $\operatorname{lcoeff}\left(x^{-\vec{s}} \mathbf{A}\right)$ and the diagonal matrix $\operatorname{lcoeff}\left(x^{-\vec{s}} \mathbf{H}\right)$ alone. □

Although the Hermite normal form $\mathbf{H}$ of $\mathbf{F}$ can be computed from a $[-\vec{u},-\vec{s}]$-minimal kernel basis of $[\mathbf{F}, I]$, a major problem here is that $\max \vec{s}$ can be very large. So the existing algorithms would be inefficient if applied directly.

However, since we know the row degrees of $\mathbf{H}$, we can expand each of the high degree rows of $\mathbf{H}$ to multiple rows with lower degrees, as done in [Gupta and Storjohann, 2011, Gupta, 2011] and also in the computation of order basis with unbalanced shift from Chapter 4, which then allows to compute an alternative matrix $\mathbf{H}'$ with lower degrees but a higher row dimension that is still in $O(n)$, such that $\mathbf{H}'$ can be easily transformed to $\mathbf{H}$. Our task here is in fact easier than in Chapter 4 as we already know the exact row degrees of $\mathbf{H}$.

For each entry s_i of the shift $\vec{s}$, let q_i and r_i be the quotient and remainder of s_i divided by d. Then, we expand the ith column e_i of the identity matrix I in $[\mathbf{F}, I]$ and shift s_i to

$$\tilde{\mathbf{E}}^{(i)} = \left[e_i, x^{s_i - q_i d} e_i, \dots, x^{s_i - d} e_i\right] \text{ and } \tilde{s}_i = [r_i, d, \dots, d,],$$

where $\tilde{s}_i$ has with $q_i + 1$ entries in each case. For the transformed problem, the shift $\vec{s}$ becomes $\bar{s} = [\tilde{s}_1, \dots, \tilde{s}_n] \in \mathbb{Z}^{\bar{n}}_{\leq 0}$, and the identity matrix becomes

$$\begin{aligned}\mathbf{E} &= [\tilde{\mathbf{E}}, \dots, \tilde{\mathbf{E}}^{(n)}] \\ &= \left[\begin{array}{ccccc|c|ccccc} 1 & x^{s_1 - q_1 d} & \cdots & x^{s_1 - 2d} & x^{s_1 - d} & & & & & & \\ \hline & & & & & \ddots & & & & & \\ \hline & & & & & & 1 & x^{s_n - q_n d} & \cdots & x^{s_n - 2d} & x^{s_n - d} \end{array}\right]_{n \times \bar{n}},\end{aligned}$$

with the new column dimension $\bar{n}$ satisfying

$$\bar{n} = n + \sum_{i=1}^{n} q_i \leq n + \sum_{i=1}^{n} s_i / d \leq n + nd/d = 2n.$$

We can now recover $\mathbf{H}$ from a $[-\vec{u}, -\bar{s}]$-minimal kernel basis of $[\mathbf{F}, -\mathbf{E}]$.

Lemma 10.3. *Let* $\mathbf{B} = \begin{bmatrix} \mathbf{V}' \\ \mathbf{G}' \end{bmatrix}$ *be a* $([\mathbf{F}, -\mathbf{E}], [-\vec{u}, -\bar{s}])$*-kernel basis, where* $\mathbf{G}'$ *has* *dimension $\bar{n} \times \bar{n}$. Let $\bar{\mathbf{B}}_0$ be the matrix consisting of the columns of $\mathbf{G}'$ whose $-\bar{s}$-column degrees are bounded by 0. Then $\mathbf{H}$ is a column basis of $\mathbf{E}\bar{\mathbf{B}}_0$ and the nonzero columns of $\mathbf{E}\bar{\mathbf{B}}_0$ have $-\vec{s}$-column degrees 0, allowing us to recover $\mathbf{H}$ from $\mathbf{E}\bar{\mathbf{B}}_0$ using Lemma 10.2.*

Proof. Note that we can construct a $([\mathbf{F}, -\mathbf{E}], [-\vec{u}, -\bar{s}])$-kernel basis

$$\mathbf{A} = \begin{bmatrix} \mathbf{U} & 0 \\ \mathbf{H_E} & \mathbf{N_E} \end{bmatrix},$$

where $\mathbf{U}$ is the unimodular matrix satisfying $\mathbf{FU} = \mathbf{H}$, $\mathbf{H_E}$ is the matrix $\mathbf{H}$ expanded according to $\mathbf{E}$, so that $\mathbf{H_E}$ has row degrees $\bar{s}$ and $\mathbf{H_E E} = \mathbf{H}$, and $\mathbf{N_E}$ is a $(\mathbf{E}, -\bar{s})$-kernel basis. Let $\mathbf{B} = \begin{bmatrix} \mathbf{V}' \\ \mathbf{G}' \end{bmatrix}$ be another $([\mathbf{F}, -\mathbf{E}], [-\vec{u}, -\bar{s}])$-kernel basis. Then the matrix $\mathbf{A}_0$ and $\mathbf{B}_0$ consists of the columns from $\mathbf{A}$ and $\mathbf{B}$, respectively, whose $[-\vec{u}, -\bar{s}]$-column degrees are bounded by 0, are unimodularly equivalent, that is, $\mathbf{A}_0\mathbf{U}_0 = \mathbf{B}_0$ for some unimodular matrix $\mathbf{U}_0$. As a result, the matrices $\bar{\mathbf{A}}_0$ and $\bar{\mathbf{B}}_0$ consists of the bottom $\bar{n}$ rows of $\mathbf{A}_0$ and $\mathbf{B}_0$ respectively, also satisfies $\bar{\mathbf{A}}_0\mathbf{U}_0 = \bar{\mathbf{B}}_0$. Therefore, we also get $\mathbf{E}\bar{\mathbf{A}}_0\mathbf{U}_0 = \mathbf{E}\bar{\mathbf{B}}_0$, with $\operatorname{cdeg}_{-\vec{s}} \mathbf{E}\bar{\mathbf{A}}_0 \leq 0$ and $\operatorname{cdeg}_{-\vec{s}} \mathbf{E}\bar{\mathbf{B}}_0 \leq 0$, since $\operatorname{cdeg}_{-\bar{s}} \bar{\mathbf{A}}_0 \leq 0$ and $\operatorname{cdeg}_{-\bar{s}} \bar{\mathbf{B}}_0 \leq 0$. Let $\bar{\mathbf{A}}_0 = [\mathbf{H_E}, \mathbf{N}']$, where $\mathbf{N}'$ consists of the columns of $\mathbf{N}_E$ with $-\bar{s}$-column degrees bounded by 0. Then $\mathbf{E}\bar{\mathbf{A}}_0\mathbf{U}_0 = \mathbf{E}[\mathbf{H_E}, \mathbf{N}']\mathbf{U}_0 = [\mathbf{H}, 0]\mathbf{U}_0 = \mathbf{E}\bar{\mathbf{B}}_0$. Now since $\mathbf{H}$ is $-\vec{s}$-column reduced and has $-\vec{s}$-column degrees 0, the nonzero columns of $\mathbf{E}\bar{\mathbf{B}}_0$ must have $-\vec{s}$-column degrees no less than 0, hence their $-\vec{s}$-column degrees are equal to 0. □

The problem of computing a $([\mathbf{F}, I], [-\vec{u}, -\vec{s}])$-kernel basis is now reduced to computing a $([\mathbf{F}, \mathbf{E}], [-\vec{u}, -\bar{s}])$-kernel basis. However, the degree of $\mathbf{E}$ and the shift $\vec{u}$ are still too big to make the computation efficient. To lower these, we can reduce $\mathbf{E}$ against $\mathbf{F}$ in a way similar to [Gupta and Storjohann, 2011, Gupta, 2011], where the authors used the Smith normal form to reduce the degrees.

Lemma 10.4. *Let* $\mathbf{R} = \mathbf{E} - \mathbf{F}\mathbf{Q}$ *for some polynomial matrix* $\mathbf{Q}$ *such that* $\mathbf{R}$ *has degree less than d and* $\bar{u} = [2d, \ldots, 2d] \in \mathbb{Z}^n$. *Let* $\mathbf{D} = \begin{bmatrix} \bar{\mathbf{V}} \\ \mathbf{G}' \end{bmatrix}$ *be a* $([\mathbf{F}, -\mathbf{R}], [-\bar{u}, -\bar{s}])$*-kernel basis, where the block* $\mathbf{G}'$ *has dimension* $n \times n$*, and* $\bar{\mathbf{D}}_0$ *be the matrix consisting of the columns of* $\mathbf{G}'$ *whose* $-\bar{s}$*-column degrees are bounded by 0. Then* $\mathbf{H}$ *is a column basis of* $\mathbf{E}\bar{\mathbf{D}}_0$ *and the nonzero columns of* $\mathbf{E}\bar{\mathbf{D}}_0$ *have* $-\vec{s}$*-column degrees 0, allowing us to recover* $\mathbf{H}$ *from* $\mathbf{E}\bar{\mathbf{D}}_0$ *using Lemma 10.2.*

Proof. First note that from the kernel basis $\mathbf{A} = \begin{bmatrix} \mathbf{U} & 0 \\ \mathbf{H}_\mathbf{E} & \mathbf{N}_\mathbf{E} \end{bmatrix}$ of $[\mathbf{F}, -\mathbf{E}]$ constructed in Lemma 10.3, we can construct a kernel basis

$$\mathbf{C} = \begin{bmatrix} I & -\mathbf{Q} \\ 0 & I \end{bmatrix} \mathbf{A} = \begin{bmatrix} \mathbf{U} - \mathbf{Q}\mathbf{H}_\mathbf{E} & -\mathbf{Q}\mathbf{N}_E \\ \mathbf{H}_\mathbf{E} & \mathbf{N}_\mathbf{E} \end{bmatrix}$$

of $[\mathbf{F}, -\mathbf{R}] = [\mathbf{F}, -\mathbf{E}] \begin{bmatrix} I & \mathbf{Q} \\ 0 & I \end{bmatrix}$. Now if $\mathbf{D}$ is a $([\mathbf{F}, -\mathbf{R}], [-\bar{u}, -\bar{s}])$-kernel basis, it satisfies $\mathbf{C}\mathbf{V} = \mathbf{D}$ for a unimodular $\mathbf{V}$. Also, the matrix $\mathbf{C}_0$ and $\mathbf{D}_0$ consist of the columns from $\mathbf{C}$ and $\mathbf{D}$, respectively, whose $[-\bar{u}, -\bar{s}]$-column degrees are bounded by 0, satisfy $\mathbf{C}_0 = \mathbf{D}_0\mathbf{V}_0$ for some polynomial matrix $\mathbf{V}_0$. Then the matrices $\bar{\mathbf{C}}$, $\bar{\mathbf{D}}$, $\bar{\mathbf{C}}_0$, $\bar{\mathbf{D}}_0$ consist of the bottom $\bar{n}$ rows of $\mathbf{C}$, $\mathbf{D}$, $\mathbf{C}_0$, $\mathbf{D}_0$ respectively, satisfy $\bar{\mathbf{C}}\mathbf{V} = \bar{\mathbf{D}}$ and $\bar{\mathbf{C}}_0 = \bar{\mathbf{D}}_0\mathbf{V}_0$. It then follows that $\mathbf{E}\bar{\mathbf{C}}\mathbf{V} = \mathbf{E}[\mathbf{H}_\mathbf{E}, \mathbf{N}_\mathbf{E}]\mathbf{V} = [\mathbf{H}, 0]\mathbf{V} = \mathbf{E}\bar{\mathbf{D}}$ and $\mathbf{E}\bar{\mathbf{C}}_0 = \mathbf{E}[\mathbf{H}_\mathbf{E}, \mathbf{N}'] = [\mathbf{H}, 0] = \mathbf{E}\bar{\mathbf{D}}_0\mathbf{V}_0$, where $\mathbf{N}'$ consists of the columns of $\mathbf{N}_E$ with $-\bar{s}$-column degrees bounded by 0. From $[\mathbf{H}, 0]\mathbf{V} = \mathbf{E}\bar{\mathbf{D}}$ we know that the nonzero columns of $\mathbf{E}\bar{\mathbf{D}}$ has $-\vec{s}$-column degrees no less than $\operatorname{cdeg}_{-\vec{s}}\mathbf{H} = 0$. On the other hand, we know that $\operatorname{cdeg}_{-\vec{s}}\mathbf{E}\bar{\mathbf{D}}_0 \leq 0$ since $\operatorname{cdeg}_{-\bar{s}}\bar{\mathbf{D}}_0 \leq 0$, therefore the nonzero columns of $\mathbf{E}\bar{\mathbf{D}}_0$ has $-\vec{s}$-column degrees equal 0. Also from $[\mathbf{H}, 0]\mathbf{V} = \mathbf{E}\bar{\mathbf{D}}$ and $[\mathbf{H}, 0] = \mathbf{E}\bar{\mathbf{D}}_0\mathbf{V}_0$ we know that $\mathbf{H}$ is a column basis of $\mathbf{E}\bar{\mathbf{D}}_0$. □

A $([\mathbf{F}, -\mathbf{R}], [-\bar{u}, -\bar{s}])$-kernel basis from Lemma 10.4 can now be efficiently computed, and can then be used to recover the Hermite normal form. A big question remaining, however, is how to efficiently compute the remainder $\mathbf{R}$ from $\mathbf{E}$ and $\mathbf{F}$. For this, we can use the series expansion of the inverse of

$$\mathbf{F}^r = \operatorname{colRev}(\mathbf{F}, 0, \vec{d}) = \mathbf{F}(1/x) \begin{bmatrix} x^{d_1} & & \\ & \ddots & \\ & & x^{d_n} \end{bmatrix},$$

as noted in the proof of Lemma 3.4 from [Giorgi et al., 2003]. The series expansion can be done using the series solution algorithm from Storjohann [2003]. Note that since $\mathbf{F}$ is assumed to be column reduced, $\deg\det\mathbf{F} = \sum\vec{d}$ exactly, and therefore x is not a factor of $\deg\det\mathbf{F}^r$, which means the series expansion of $(\mathbf{F}^r)^{-1}$ always exists. It also means that using x^d-adic lifting always works, and the series solution algorithm from Storjohann [2003] becomes deterministic.

Let us now look at how the series expansion of $(\mathbf{F}^r)^{-1}$ gives a remainder of x^kI divide by $\mathbf{F}$.

Lemma 10.5. *Let* the series expansion of $(\mathbf{F}^r)^{-1}$ be $\bar{\mathbf{F}} = F_0 + F_1x + F_2x^2 + \ldots$. *Then for any integer $k \geq d$, we have*

$$I = \mathbf{F}^r\left(\bar{\mathbf{F}} \mod x^{k-\vec{d}}\right) + x^{k-d}\mathbf{C}, \tag{10.1}$$

where $\bar{\mathbf{F}} \mod x^{k-\vec{d}}$ denotes the ith row of $\bar{\mathbf{F}} \mod x^{k-d_i}$ for each row i. Then $\mathbf{C}^r = \operatorname{colRev}\left(x^{k-d}\mathbf{C}, 0, k\right)$ has degree less than d and satisfies $x^kI = \mathbf{F}\cdot(*) + \mathbf{C}^r$.

Proof. Since the first term of Equation (10.1) has degree less than k, the the degree of $x^{k-d}\mathbf{C}$ must be also less than k, or equivalently, the degree of $\mathbf{C}$ must be less than d. If we now reverse the coefficients, we get

$$\begin{aligned}
& \operatorname{colRev}(I, 0, k) \\
= \ & \operatorname{colRev}(\mathbf{F}^r, 0, \vec{d}) \cdot \operatorname{colRev}\left(\left(\bar{\mathbf{F}} \mod x^{k-\vec{d}}\right), \vec{d}, k\right) + \operatorname{colRev}\left(x^{k-d}\mathbf{C}, 0, k\right),
\end{aligned}$$

that is,

$$x^k I = \mathbf{F} \cdot (*) + \mathbf{C}^r,$$

which gives us $\mathbf{C}^r$ as a remainder of $x^k I$ divided by $\mathbf{F}$, where $\mathbf{C}^r$ has degree less than d. □

Lemma 10.5 shows how the series expansion $\bar{\mathbf{F}}$ can be used to compute a remainder of $x^k I$ divided by $\mathbf{F}$ for any $k \geq d$. Similarly, the ith column $\bar{\mathbf{F}}_i = \bar{\mathbf{F}} e_i$ of $\bar{\mathbf{F}}$ allows us to compute a remainder $\mathbf{r}$ of $x^k e_i$ divided by $\mathbf{F}$, with $\deg \mathbf{r} < d$. Note that the degrees of columns correspond to e_i are bounded by s_i, so we need to compute the series expansion $\bar{\mathbf{F}}_i$ to at least order s_i. Now let us look how these series expansions can be computed efficiently.

Lemma 10.6. *Computing the series expansions $\bar{\mathbf{F}}_i$ to order s_i for all i's where $s_i \geq d$ can be done with a cost of $O^\sim(n^\omega d)$* field operations.

Proof. As before, we assume without loss of generality that the columns of $\mathbf{F}$ and the corresponding entries of $\vec{s} = [s_1, \ldots, s_n]$ are arranged so that the entries of $\vec{s}$ are in increasing order. We separate $\vec{s}$ to $\lceil \log n \rceil + 1$ disjoint lists $\vec{s}_{\bar{j}^{(0)}}, \vec{s}_{\bar{j}^{(1)}}, \vec{s}_{\bar{j}^{(2)}}, \ldots, \vec{s}_{\bar{j}^{(\lceil \log n \rceil)}}$ with entries in the ranges $[0, d)$, $[d, 2d)$, $[2d, 4d)$, $[4d, 8d)$, ...,$[2^{\lceil \log n \rceil - 2}d, 2^{\lceil \log n \rceil - 1}d)$, $[2^{\lceil \log n \rceil - 1}d, nd]$ respectively, where each $\bar{j}^{(i)}$ consists a list of indices of the entries of $\vec{s}$ that belong to $\vec{s}_{\bar{j}^{(i)}}$. Note that $\bar{j}^{(i)}$ has at most $n/2^{i-1}$ entries, otherwise, the sum of the entries of $\vec{s}_{\bar{j}^{(i)}}$ would exceed $\sum \vec{s} = nd$. Then we compute series expansions $\bar{\mathbf{F}}_{\bar{j}^{(1)}}, \bar{\mathbf{F}}_{\bar{j}^{(2)}}, \ldots, \bar{\mathbf{F}}_{\bar{j}^{(\lceil \log n \rceil)}}$ separately, to order $2d, 4d, \ldots, 2^{\lceil \log n \rceil - 1}d/2, nd$ respectively, where again $\bar{\mathbf{F}}_{\bar{j}^{(i)}}$ consists of the columns of

$\bar{\mathbf{F}}$ that are indexed by the entries in $\bar{j}^{(i)}$. We can use the series solution algorithm from Storjohann [2003] to do these computations. For $\bar{\mathbf{F}}_{\bar{j}^{(i)}}$, there are at most $n/2^{i-1}$ columns, so computing the series expansion to order $2^i d$ cost $O^\sim(n^\omega d)$. Then doing this for i from 1 to $\lceil \log n \rceil$ costs $O^\sim(n^\omega d)$ field operations. □

With the series expansions computed, we can now compute a remainder $\mathbf{R}$ of $\mathbf{E}$ divide by $\mathbf{F}$.

Lemma 10.7. *A remainder* $\mathbf{R}$ *of* $\mathbf{E}$ *divide by* $\mathbf{F}$*, where* $\deg \mathbf{R} < d$*, can be computed with a cost of* $O^\sim(n^\omega d)$ *field operations.*

Proof. The remainder $\mathbf{r}$ of $x^k e_i$ divide by $\mathbf{F}$ can be obtained by

$$\left(e_i - \mathbf{F}^r \left(\bar{\mathbf{F}}_i \mod x^{k-\vec{d}}\right)\right) / x^{k-d}.$$

Note that only the terms from $\bar{\mathbf{F}}_i$ with degrees in the range $[k-2d, k)$ are needed for this computation, which means we are just multiplying $\mathbf{F}^r$ with a polynomial vector with degree bounded by $2d$. To make the multiplication more efficient, we can compute all the remainder vectors at once. Since there at most n columns with degrees no less than d, the cost is just the multiplication of matrices of dimension n and degrees bounded by $2d$, which costs $O^\sim(n^\omega d)$ field operations. □

With the remainder $\mathbf{R}$ computed, we can now compute a $([\mathbf{F}, -\mathbf{R}], [-\bar{u}, -\bar{s}])$-kernel basis that can be used to recover the Hermite normal form using Lemma 10.4.

Theorem 10.8. *A Hermite normal form of* $\mathbf{F}$ *can be computed deterministically with a cost of* $O^\sim(n^\omega d)$ field operations.

Chapter 11

Rank Profile and Rank Sensitive Computation of Kernel Basis

In this chapter, we consider the problems of computing the row rank profile of an input matrix $\mathbf{F} \in \mathbb{K}[x]^{m \times n}$, which also immediately gives us the rank. If $n \geq m$, the rank can already be computed by either kernel basis computation or column basis computation from the earlier chapters. The column basis computation (Algorithm 7.2) can compute the rank with a cost of $O^{\sim}(nm^{\omega-1}s)$ field operations, where s is the average column degree of $\mathbf{F}$. However, we would like to refine this cost to $O^{\sim}(nmr^{\omega-2}s)$, where r is the rank of $\mathbf{F}$. We also would like to compute a row rank profile with the same cost.

We use the following approach to achieve the desired cost. We first modify our kernel basis algorithm, Algorithm 11.1, slightly to allow the rank profile to be computed along with a kernel basis. Then we do a series of computations with increasing number of rows from $\mathbf{F}$. For each set of rows we do successive column basis computation (or order basis computation) as in Section 7.3 to reduce the column dimension of the problem, so the modified Algorithm 11.1 can work efficiently to compute the rank profile of this set of rows.

11.1 Rank Profile from Kernel Basis Computation

Recall that the row rank profile of $\mathbf{F}$ is the lexicographically smallest list of row indices $[i_1, i_2, \dots i_r]$ such that these rows of $\mathbf{F}$ are linearly independent, where r is the rank of $\mathbf{F}$. Let us see how the row rank profile can be computed by our kernel basis algorithm. The following lemma provides a key to the rank profile computation using Algorithm 5.1.

Lemma 11.1. *At any base case of running Algorithm 5.1 on the input matrix* $\mathbf{F}$*, we work with an input matrix* $\mathbf{g}$ *consisting of a single row. Let* $\mathbf{f}$ *be the original row in* $\mathbf{F}$ *corresponding to* $\mathbf{g}$ *and* $\mathbf{F}'$ *be the submatrix of* $\mathbf{F}$ *consists of the rows above* $\mathbf{f}$*. Then* $\mathbf{g} = 0$ *if and only if* $\mathbf{f}$ *is linearly dependent with the rows of* $\mathbf{F}'$*.*

Proof. When the algorithm has reached the base case involving the single row matrix $\mathbf{g}$, it has finished processing $\mathbf{F}'$ and has produced a number of order bases and kernel bases from the earlier subproblems, where the kernel bases computed only involved all the rows of $\mathbf{F}'$. The matrix $\mathbf{g}$ is the residual from multiplying $\mathbf{f}$ with these order bases and kernel bases. Note that such multiplications do not change the linear dependency of $\mathbf{g}$ with the rows of $\mathbf{F}'$. But if $\mathbf{f}$ is linearly dependent with the rows of $\mathbf{F}'$, the residual $\mathbf{g}$ becomes 0 after multiplying with kernel bases of the rows of $\mathbf{F}'$. □

Lemma 11.1 now allows us to provide a small modification to Algorithm 5.1 to produce the rank profile of $\mathbf{F}$. The modified algorithm is given in Algorithm 11.1. Note that the rank profile in our algorithm is represented using a list of n indicators that indicate the first r linearly independent rows of $\mathbf{F}$. At this point, the rank profile of $\mathbf{F}$ still costs the same to compute as a kernel basis of $\mathbf{F}$. In the following, we see how column basis computation can be used to improve this.

Algorithm 11.1 minimalKernelBasisWithRankProfile($\mathbf{F}, \vec{s}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$, $\vec{s} = [s_1, \ldots, s_n] \in \mathbb{Z}^n$ with entries arranged in non-decreasing
    order and bounding the corresponding column degrees of $\mathbf{F}$.
Output: A $\vec{s}$-minimal kernel basis $\mathbf{N}$ of $\mathbf{F}$ and the row rank profile of $\mathbf{F}$ given by
    a list binary indicators $\bar{e} = [e_1, \ldots, e_m]$, with 1's indicating the columns in the
    rank profile.
1: $\xi := \sum_{i=1}^{n} s_i$; $\rho := \sum_{i=n-m+1}^{n} s_i$; $s := \rho/m$;
2: $\left[\mathbf{P}, \vec{b}\right]$ := orderBasis$(\mathbf{F}, 3s, \vec{s})$, a $(\mathbf{F}, 3s, \vec{s})$-basis with the columns of $\mathbf{P}$ and
    the entries of is $\vec{s}$-column degrees $\vec{b}$ arranged so that the entries of $\vec{b}$ are in
    non-decreasing order;
3: $[\mathbf{P}_1, \mathbf{P}_2] := \mathbf{P}$ where $\mathbf{P}_1$ consists of all columns $\mathbf{p}$ of $\mathbf{P}$ satisfying $\mathbf{Fp} = 0$;
4: if $m = 1$ then
5:   if $\mathbf{F} = 0$ then
6:     return $\mathbf{P}_1, [0]$
7:   else
8:     return $\mathbf{P}_1, [1]$
9:   end if
10: else
11:   $\vec{t} := \deg_{\vec{s}} \mathbf{P}_2 - [3s, 3s, \ldots, 3s]$;
12:   $\mathbf{G} := \mathbf{F}\mathbf{P}_2 / x^{3s}$;
13:   $\left[\mathbf{G}_1^T, \mathbf{G}_2^T\right]^T := \mathbf{G}$, with $\mathbf{G}_1$ having $\lfloor m/2 \rfloor$ rows and $\mathbf{G}_2$ having $\lceil m/2 \rceil$ rows;
14:   $\mathbf{N}_1, \bar{e}_1 :=$ minimalKernelBasis $\left(\mathbf{G}_1, \vec{t}\right)$;
15:   $\mathbf{N}_2, \bar{e}_2 :=$ minimalKernelBasis $\left(\mathbf{G}_2\mathbf{N}_1, \mathrm{cdeg}_{\vec{t}}\mathbf{N}_1\right)$;
16:   $\mathbf{Q} := \mathbf{N}_1\mathbf{N}_2$;
17:   return $[\mathbf{P}_1, \mathbf{P}_2\mathbf{Q}], [\bar{e}_1, \bar{e}_2]$
18: end if
```

11.2 Successive Rank Profile Computation

To compute the rank and rank profile of $\mathbf{F}$ in a rank-sensitive way, we do a series of computations with sets of increasing number of rows from $\mathbf{F}$.

We start with the first nonzero row of $\mathbf{F}$, which is the first row in the rank-profile. Suppose we have found the indices $\bar{j} = [j_1, \ldots, j_k]$ in the rank profile. To find the next linearly independent rows, we work with the matrix $\mathbf{G} = \mathbf{F}_{[\bar{j}, j_k+1 \ldots, j_k+k]}$, the matrix consists of the k linearly independent rows indexed by $\bar{j}$ and the next k rows. We compute a column basis $\mathbf{T}$ of this matrix, which has the same rank profile as $\mathbf{G}$. Now we can use Algorithm 11.1 to compute the rank profile of $\mathbf{T}$, which gives

Algorithm 11.2 rankProfile($\mathbf{F}$)

```
Input: $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$.
Output: A row rank profile $\bar{j} = [j_1, \ldots, j_r] \in \mathbb{Z}^r$ of $\mathbf{F}$.
1: $k := 1$; ($k$ keeps track of the current row)
2: while $k \le m$ and $\mathbf{F}_k \ne 0$ do $k := k+1$ end while; (find the first nonzero
   row $\mathbf{F}_k$)
3: $\bar{j} = [k]$; $r := 1$; ($r$ is the current rank)
4: while $k < m$ do
5:     $k' := \min(m, k+r-1)$; (last row in the block)
6:     $\mathbf{G} := \mathbf{F}_{[\bar{j}, k\ldots, k']}$; (the $r$ linearly independent rows and the next block of rows)
7:     $\mathbf{T} :=$ colBasis($\mathbf{G}$);
8:     $\mathbf{N}, \bar{e} :=$ minimalKernelBasisWithRankProfile($\mathbf{G}$);
9:     for $i$ from $r+1$ to $2r$ (convert indicator to indices and append to $\bar{j}$) do
10:        if $e_i = 1$ then $\bar{j} := [\bar{j}, e_i - r + k - 1]$ end if;
11:    end for
12:    $k := k + r$;
13: end while
14: return $\bar{j}$;
```

more indices for the rank profile of $\mathbf{F}$. We repeat this procedure until all rows of $\mathbf{F}$ are processed. This gives us Algorithm 11.2 for computing the rank profile of $\mathbf{F}$.

The main remaining task is to analyze the computational cost of Algorithm 11.2.

Theorem 11.2. *The cost of Algorithm 11.2 is $O^{\sim}(nmr^{\omega-2}s)$ field operations for computing a rank profile of $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$.*

Proof. Let r_i be the number of rows in the new block considered at step i. Then the cost at step i is $O^{\sim}\left(r_i^{\omega-1}ns\right)$ field operations for the column basis and the kernel basis with rank profile computation. The total cost is then

$$\sum O^{\sim}\left(r_i^{\omega-1}ns\right) = O^{\sim}\left(ns\sum r_i^{\omega-1}\right).$$

We also know that $\sum r_i = m$, and the maximum of r_i is the rank r of $\mathbf{F}$. Note that

$$\sum r_i^{\omega-1} \le \frac{m}{r} r^{\omega-1} = mr^{\omega-2}.$$

Hence

$$\sum O^{\sim}\left(r_i^{\omega-1}ns\right) = O^{\sim}\left(ns\sum r_i^{\omega-1}\right) = O^{\sim}\left(nmr^{\omega-2}s\right).$$

□

11.3 Applications of Rank Profile Computation

11.3.1 Remove the assumption $n \geq m$

In order basis, kernel basis, and column basis computations from previous chapters, we have assumed that the column dimension n is no less than the row dimension m. We can now use the rank profile computation to ensure that this is always the case. For order basis and kernel basis computation, we can just determine the rank profile $\bar{j}$ of the input matrix $\mathbf{F}$, and then work with just $\mathbf{F}_{\bar{j}}$, which consists of only r linearly independent rows, as we know the rank r is always bounded by the column dimension n. For column basis computation, the assumption is only required by the kernel basis computation used. Therefore removing this assumption from the kernel basis computation also removes this assumption from the column basis computation.

11.3.2 Rank-sensitive computation of minimal kernel bases

With the ability to compute a rank profile efficiently, we can now slightly improve Corollary 5.23 on the cost of kernel basis computation with a matrix of degree d, by using only the linearly independent rows from $\mathbf{F}$, hence reducing the row dimension of the input matrix from m to r, after a cost of $O^{\sim}\left(nmr^{\omega-2}d\right)$ to compute the rank profile.

Theorem 11.3. *Given a matrix $\mathbf{F} \in \mathbb{K}[x]^{m\times n}$ with degree d, a minimal kernel basis of $\mathbf{F}$ can be computed with a cost of $O^{\sim}(nmr^{\omega-2}d + n^{\omega-1}rd)$.*

It looks difficult to further improve this cost by removing the exponent ω from the column dimension n if a minimal kernel basis is required. Since the minimality requires us to work with some matrix involving all n columns at the same time.

Chapter 12

Conclusion

In this thesis, we have presented efficient deterministic algorithms for a number of polynomial matrix computation problems, including the computation of order basis, minimal nullspace basis, matrix inverse, column basis, unimodular completion, determinant, Hermite normal form, rank, and rank profile. The algorithm for kernel basis computation also immediately gives us a new way to solve linear systems. An existing efficient deterministic method for solving linear systems was given by Gupta et al. [2012]. The algorithm for column basis also immediately allows us to compute matrix GCD, column reduced forms and Popov normal forms for matrices of any dimension.

We first gave algorithms for computing a shifted order basis of an $m \times n$ matrix of power series over a field $\mathbb{K}$ with $m \le n$. For a given order σ and balanced shift $\vec{s}$ the first algorithm determines an order basis with a cost of $O^{\sim}(n^{\omega} a)$ field operations in $\mathbb{K}$, where $a = m\sigma/n$. We then provided a method to refine the cost to $O^{\sim}(n^{\omega-1} m\sigma)$. While the first algorithm addresses the case when the column degrees of a complete order basis are unbalanced given a balanced input shift, it is not efficient in the case when an unbalanced shift results in the row degrees also becoming unbalanced. We have presented a second algorithm which balances the high degree rows and computes an order basis also using $O^{\sim}(n^{\omega}a)$ field operations in the case that the shift is unbalanced but satisfies the condition $\sum_{i=1}^{n}(\max(\vec{s}) - \vec{s}_i) \leq m\sigma$. This condition essentially allows us to locate those high degree rows that need to be balanced.

We then presented an algorithm for the computation of a minimal nullspace basis of an $m \times n$ input matrix of univariate polynomials over a field $\mathbb{K}$ with $m \leq n$. This algorithm computes a minimal nullspace basis of a degree d input matrix with a cost of $O^{\sim}\left(n^{\omega-1}md\right)$ field operations in $\mathbb{K}$. The same algorithm also works in the more general situation on computing a shifted minimal nullspace basis, with a given degree shift $\vec{s} \in \mathbb{Z}^n$ whose entries bound the corresponding column degrees of the input matrix. In this case a $\vec{s}$-minimal right nullspace basis can be computed with a cost of $O^{\sim}(n^{\omega}s)$ field operations, where s is the average of the largest m entries of $\vec{s}$.

Order basis computation and nullspace basis computation were then applied to the remaining problems. An algorithm for computing the inverse of an matrix in $\mathbb{K}[x]^{n\times n}$ was then given with a cost of $O^{\sim}\left(n^{3}s\right)$ field operations, where s is the average of the column or row degrees of the input matrix. The inverse represented alternatively by a product of $\lceil \log n \rceil$ matrices costs only $O^{\sim}\left(n^{\omega}s\right)$ to compute. We then discussed the computation of a column basis of an input matrix in $\mathbb{K}[x]^{m\times n}$ with a cost of $O^{\sim}\left(m^{\omega}ns\right)$, where s is again the average column degree of the input matrix. Next, an algorithm was presented for computing an unimodular completion of an input matrix in $\mathbb{K}[x]^{m\times n}$, $m < n$ with a cost of $O^{\sim}\left(n^{\omega}s\right)$, where s is the average of the m largest column degrees of the input matrix. Then an algorithm for computing the determinant of an input matrix in $\mathbb{K}[x]^{n\times n}$ with a cost of $O^{\sim}\left(n^{\omega}s\right)$ was given, where s is the average column or row degree of the input matrix. Then we looked at an algorithm for computing the Hermite normal form of a degree d input matrix in $\mathbb{K}[x]^{n\times n}$ with a cost of $O^{\sim}\left(n^{\omega}d\right)$. Finally, we provided algorithms for rank-sensitive computations of the rank and rank profile of an input matrix in $\mathbb{K}[x]^{m\times n}$ with a cost of $O^{\sim}(mr^{\omega-2}ns)$, where s is the average column degree of the input matrix, and then applied the rank profile algorithm to rank-sensitive computation of minimal kernel basis to obtain a cost of $O^{\sim}(nmr^{\omega-2}d+n^{\omega-1}rd)$.

We reduce all these problems to polynomial matrix multiplications. The computational costs of our algorithms are then similar to the costs of multiplying matrices, whose dimensions match the input matrix dimensions in the original problems, and whose degrees equal the average column degrees of the original input matrices in most cases. The use of the average column degrees instead of the commonly used matrix degrees, or equivalently the maximum column degrees, makes our computational costs more precise and tighter. In addition, the shifted minimal bases computed by our algorithms are more general than the standard minimal base.

Bibliography

G. Baker and P. Graves-Morris. *Padé Approximants, 2nd edition*. Cambridge, 1996. 2

B. Beckermann and G. Labahn. A uniform approach for the fast computation of matrix-type Padé approximants. *SIAM Journal on Matrix Analysis and Applications*, 15(3):804–823, 1994. 6, 23, 31

B. Beckermann and G. Labahn. Recursiveness in matrix rational interpolation problems. *Journal of Computational and Applied Math*, 5-34, 1997. 2, 5, 7, 23, 44

B. Beckermann and G. Labahn. Fraction-free computation of matrix rational interpolants and matrix GCDs. *SIAM Journal on Matrix Analysis and Applications*, 22(1):114–144, 2000. 102

B. Beckermann, G. Labahn, and G. Villard. Shifted normal forms of polynomial matrices. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC'99, pages 189–196, 1999. 2, 57

B. Beckermann, H. Cheng, and G. Labahn. Fraction-free row reduction of matrices of ore polynomials. *Journal of Symbolic Computation*, 41(1):513–543, 2006a. 102

B. Beckermann, G. Labahn, and G. Villard. Normal forms for general polynomial matrices. *Journal of Symbolic Computation*, 41(6):708–737, 2006b. 2, 5, 57

Th. G. J. Beelen and P.M. Van Dooren. An improved algorithm for the computation of Kronecker's canonical form of a singular pencil. *Linear Algebra and its Applications*, 105:9–65, 1988. 8, 102

Th. G. J. Beelen, G. J. van den Hurk, and C. Praagman. A new method for computing a column reduced polynomial matrix. *Syst. Control Lett.*, 10(4):217–224, 1988. 3, 102

G.D. Forney. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM Journal of Control*, 13:493–520, 1975. 3

E Frisk. *Residual Generation for Fault Diagnostics.* PhD thesis, Linköping, University, Sweden, 2001. 3

J. von zur Gathen and J. Gerhard. *Modern Computer Algebra.* Cambridge University Press, 2nd edition edition, 2003. 13

P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, Philadelphia, Pennsylvania, USA*, pages 135–142. ACM Press, 2003. 3, 6, 27, 31, 42, 48, 81, 102, 144

S. Gupta and A. Storjohann. Computing hermite forms of polynomial matrices. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 155–162, 2011. 139, 140, 141, 142

S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $k[x]$. *Journal of Symbolic Computation: Special issue in honour of the research and influence of Joachim von zur Gathen at 60*, 47(4):422–453, 2012. 97, 102, 117, 153

Somit Gupta. Hermite forms of polynomial matrices. Master's thesis, University of Waterloo, 2011. 139, 141, 142

O.H. Ibarra, S. Moran, and R. Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *J. Algorithms*, 3(1):45–56, 1982. 38

C. P. Jeannerod and G. Villard. Essentially optimal computation of the inverse of generic polynomial matrices. *Journal of Symbolic Complexity*, 21(1):72–86, 2005. 3, 97

C. P. Jeannerod and G. Villard. Asymptotically fast polynomial matrix algorithms for multivariable systems. *Int. J. Control*, 79(11):1359– 1367, 2006. 3

T. Kailath. *Linear Systems.* Prentice-Hall, 1980. 3, 21

V Kucera. *Discrete Linear Control : The Polynomial Equation Approach.* John Wiley and Sons, 1979. 3

George Labahn. Inversion components for block Hankel-like matrices. *Linear Algebra and Its Applications*, 177:7–48, 1992. 2, 7

Chao Li. Lattice compression of polynomial matrices. Master's thesis, School of Computer Science, University of Waterloo, 2006. 102

P. Misra, P. Van Dooren, and A. Varga. Computation of structural invariants of generalized state-space systems. *Automatica*, 30:1921–1936, 1994. 8

W.H.L. Neven and C. Praagman. Column reduction of polynomial matrices. *Linear Algebra and its Applications*, 188:569–589, 1993. 3

Morris Newman. *Integral matrices.* Academic Press, 2nd edition edition, 1972. 120

C. Oara and P. Van Dooren. An improved algorithm for the computation of structural invariants of a system pencil and related geometric aspects. *Systems and Control Letters*, 30:38–48, 1997. 8

S. Sarkar. Computing popov forms of polynomial matrices. Master's thesis, University of Waterloo, 2011. 103

S. Sarkar and A. Storjohann. Normalization of row reduced matrices. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 297–304, 2011. 103, 117

A. Storjohann. *Algorithms for Matrix Canonical Forms.* PhD thesis, Department of Computer Science, Swiss Federal Institute of Technology—ETH, 2000. 53, 136

A. Storjohann. High-order lifting. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC'02, pages 246–254, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-484-3. 133

A. Storjohann. High-order lifting and integrality certification. *Journal of Symbolic Computation,*, 36:613–648, 2003. 133, 144, 146

A. Storjohann. Notes on computing minimal approximant bases. In *Challenges in Symbolic Computation Software.* Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006. 6, 7, 24, 27, 33, 59

A. Storjohann. On the complexity of inverting integer and polynomial matrices. *Accepted for publication in Computational Complexity*, 2010. 97

A. Storjohann and G. Villard. Computing the rank and a small nullspace basis of a polynomial matrix. In *Proceedings of the International Symposium on Symbolic* *and Algebraic Computation*, ISSAC'05, pages 309–316, 2005. 3, 7, 9, 32, 50, 80, 102

Mark Van Hoeij. Factorization of differential operators with rational functions coefficients. *Journal of Symbolic Computation*, 24:537–561, November 1997. 2

G. Villard. Computing popov and hermite forms of polynomial matrices. In *International Symposium on Symbolic and Algebraic Computation*, 1996. 102

W. Zhou and G. Labahn. Efficient computation of order bases. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC'09, pages 375–382. ACM, 2009. 8

W. Zhou and G. Labahn. Efficient algorithms for order basis computation. *Journal of Symbolic Computation*, 47:793–819, 2012. 8

W. Zhou, G. Labahn, and A. Storjohann. Computing minimal nullspace bases. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC'12, pages 375–382. ACM, 2012. 9, 115