Algorithms in computational algebraic analysis

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

ANTON LEYKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GENNADY LYUBEZNIK, Adviser

September 2003

Anton Leykin

194 words

ABSTRACT

This thesis studies algorithms for symbolic computation of systems of linear partial differential equations using the corresponding ring of linear differential operators with polynomial coefficients, which is called the Weyl algebra A_n.

Bernstein-Sato polynomials, one of the central notions in the algebraic analysis of D-modules, is the topic of the first part of this work. We consider the question of constructibility of the stratum of polynomials of bounded number of variables and degree that produce a fixed Bernstein-Sato polynomial. Not only do we give a positive answer, but we construct an algorithm for computing these strata.

Another theme of this thesis is two theorems of Stafford that say that every (left) ideal of A_n can be generated by two elements, and every holonomic A_n-module is cyclic, i.e. generated by one element. We reprove these results in an effective way that leads to algorithms for computation of these generators.

The main engine of all our algorithms is Gröbner bases computations in the Weyl algebra. In order to speed these up we developed a parallel version of a Buchberger algorithm, which has been implemented and tested out using supercomputers and has delivered impressive speedups on several important examples.

Gennady Lyubeznik

(Faculty Adviser)

Acknowledgments

First and foremost, I would like to thank my adviser Gennady Lyubeznik, who introduced me to the area of my research and provided so much needed encouragement and every kind of support over my years in the graduate school. I am grateful to Mike Stillman for organizing a couple of summer visits to Cornell and introducing me to Macaulay 2, Harry Tsai for writing the toughest bits of the D-modules package, Bernd Sturmfels for fruitful ideas and help during my stay at Berkeley, as well as the other people that made the mathematical environment lively.

Last, but not least, I thank my friends and family for love and understanding and not insisting on a detailed answer to the question: "So what is it that you do exactly?"

Contents

1 Introduction **1**

1.1 History . . . 2

1.2 Thesis preview . . . 2

1.3 Conventions and Preliminaries . . . 3

2 Stratification by Bernstein-Sato polynomials **5**

2.1 Preliminaries . . . 6

2.1.1 Constructible Sets . . . 7

2.1.2 Parametric Gröbner Bases . . . 8

2.1.3 Oaku's Algorithm . . . 10

2.1.4 Properties of the Bernstein-Sato polynomial . . . 11

2.2 The Main Results . . . 12

2.3 Examples . . . 16

3 On minimal generation of holonomic D-modules and Weyl algebra ideals **25**

3.1 Preliminaries . . . 26

3.1.1 A_n is simple . . . 26

3.1.2 A_n is an Ore domain . . . 27

3.1.3 More rings . . . 27

3.2 Holonomic modules are cyclic . . . 28

3.3 Ideals are 2-generated . . . 30

3.3.1 Lemmas for S . . . 31

3.3.2 Lemmas for $\mathcal{R}_r$ . . . 33

3.3.3 Final chords . . . 36

3.4 Conclusion . . . 37

4 Parallel computation of Gröbner bases **39**

4.1 Introduction . . . 39

4.2 Preliminaries . . . 40

4.3 Parallel Buchberger algorithm . . . 42

4.4 Experimental results . . . 44

Bibliography **49**

Chapter 1

Introduction

There is an algebraic way to look at a system of partial differential equations with polynomial coefficients. Namely, one may view it as a module over a ring of differential operators and explore its properties via *algebraic analysis*, in other words, by understanding the algebraic structure of this associated module. The founding fathers of this area – Sato, Kashiwara, Malgrange, Kawai, Bernstein, Beilinson – called it D*-module theory.* The theory of D-modules has proved to be useful to many areas of modern mathematics such as differential equations, mathematical physics, singularity theory, etc.

In the recent years the computational side of D-modules was an area of active development and led to interesting and diverse applications. For computational purposes, the main focus of attention has been the Weyl algebra

$$A_n = A_n(k) = k\left\langle x_1, ..., x_n, \partial_1, ..., \partial_n \right\rangle,$$

which is an associative k-algebra generated by x's and ∂'s with the relations

$$x_i x_j = x_j x_i,\ \partial_i \partial_j = \partial_j \partial_i,\ \partial_j x_i - x_i \partial_j = \delta_i j.$$

The Weyl algebra may be thought of as the algebra of linear differential operators with polynomial coefficients. In this thesis, we say D-modules – we mean A_n-modules.

1.1 History

The algorithmic aspects of the Weyl algebra were first explored by Galligo [12], Takayama [35] and others in mid-80's. They laid out the theory of Gröbner bases in this slightly noncommutative setting. In general, Gröbner bases play the central role in modern computational algebra providing not only a theoretical but a practical tool for constructing such objects of commutative algebra as Hilbert functions, free resolutions, homological functors, etc. In the field of computational D-modules a similar set of effective constructions for modules over the Weyl algebra has been made available for by Oaku and Takayama [29]. One of many important D-algorithms is described in Chapter 2 of this thesis: the algorithm for computing the Bernstein-Sato polynomial by Oaku [28]. All of the results described in this work are either on Gröbner bases or use Gröbner bases as a computational engine.

The algorithmic development of D-modules has also been accompanied by implementation of the Weyl algebra in computer algebra systems. Let us list the software that we know: Takayama pioneered this area implementing many D-algorithms in his specialized system **kan/sm1** [34]. Chyzak wrote a package for **Maple** called **Mgfun** [10]. Together with Harry Tsai we produced the ***D*-modules for Macaulay 2** package [24] using the general purpose computer algebra system **Macaulay 2** [15]; for an introduction to the package see [23]. At the moment of writing, the system **Plural** (a part of **Singular** [16] in the future) developed by Levandovskyy contains an implementation of Weyl algebra.

1.2 Thesis preview

Let us announce the topics covered by this thesis. In Chapter 2 after a brief introduction to the basic concepts of the theory of D-modules we describe a solution to the problem of constructibility of the stratum of polynomials corresponding to a given Bernstein-Sato polynomial. Indeed, such strata turn out to be constructible, moreover, we provide an algorithm to compute them. The main results of this chapter were published in [22].

The subject of Chapter 3 is rather ring-theoretic: we solve the problem of minimal generation of an ideal in Weyl algebra, as well as the problem of finding a cyclic generator for a holonomic D-module. Again, our answer is given in an effective form: the algorithm to construct a generating set of two elements for any ideal of A_n is laid out, and a simple algorithm to determine a cyclic generator for any holonomic D-module is developed.

The topic of Chapter 4 is parallelization of reduction algorithms for noncommutative associative algebras. We describe our parallel software implemented for the Weyl algebra and the so-called PBW-algebra, as well as (commutative) ring of polynomials. Since this part of the thesis belongs to computer science at least as much as to mathematics, it is written in a form of a technical report, i.e. contains more tables and figures than theoretical results.

1.3 Conventions and Preliminaries

Throughout the thesis k shall denote a field of characteristic zero; the most popular algebras shall be $R_n(k) = k[\bar{x}] = k[x_1, \ldots, x_n]$, the ring of polynomials over k in n variables, and $A_n(k) = k\langle \bar{x}, \bar{\partial} \rangle = k\langle x_1, \partial_1, \ldots, x_n, \partial_n \rangle$, the Weyl algebra over k in n variables. When k is specified, we abbreviate $R_n = R_n(k)$ and $A_n = A_n(k)$. All ideals of the Weyl algebra and all A_n modules are assumed to be left ideals and modules respectively.

We shall use multi-index notation as follows. For example, we will write a monomial in R_n as $a_\alpha x^\alpha$ and mean $a_\alpha x_1^{\alpha_1} \ldots x_n^{\alpha_n}$.

Where necessary we will include output from Macaulay 2 sessions in the following typical format:

```
i1 : statement 1;
i2 : statement 2
o2 = output
```

Here `i1`, `i2`, `i3`, `...` are input lines with corresponding output lines `o1`, `o2`, `o3`, `....` The output of a statement followed by a semicolon, for instance, `statement 1`, is suppressed.

A good introduction to D-modules is the book by Björk [5]. Another good source is a book by Sato, Sturmfels and Takayama [31] that describes applications of D-modules to hypergeometric differential equations. Algebraic geometry basics one may look up in Hartshorne [17].

Weyl algebra. It is easy to see that monomials $x^\alpha \partial^\beta \in A_n(k)$, $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$ form a k-basis of A_n. Thus, every element of $Q \in A_n$ may be presented in the *right normal form*

$$Q = \sum a_{\alpha\beta} x^\alpha \partial^\beta,$$

where all but finitely many of $a_{\alpha\beta}$ are zeros.

The Weyl algebra is simple, i.e. the only two-sided ideals are 0 and the whole of A_n. The are more left/right ideals; in what follows we will encounter only left ideals. Obviously, the Weyl algebra is noetherian, moreover, in Chapter 3 we show that every ideal of it can be generated by two elements.

D-modules. For our purposes a D-module is a left A_n-module. It is shown that the Gelfand-Kirillov dimension of the algebra A_n equals $2n$, moreover, if M is a nontrivial D-module, then $n \leq \dim M \leq 2n$. The modules of dimension n (minimal possible dimension) constitute the *Bernstein class*.

One of the distinctive properties of the modules in the Bernstein class, which are also called *holonomic* D-modules, is their finite length. It also may be shown that every holonomic module is cyclic. Moreover, in Chapter 3 we develop an algorithm that constructs a cyclic generator of a holonomic D-module. From the homological point of view a holonomic D-module M is a D-module for which $\mathrm{Ext}^j_{A_n}(M, A_n)$ vanishes unless $j = n$.

The simplest example of a holonomic D-module is R_n, which is the quotient of A_n by the left ideal $A_n \cdot (\partial_1, ..., \partial_n)$. Also if $f \in R_n$, then the R_n-module $R_n[f^{-1}]$ has a structure of a A_n-module:

$$x_i(\frac{g}{f^k}) = \frac{x_i g}{f^k},\ \partial_i(\frac{g}{f^k}) = \frac{\partial_i(g)f - k\partial_i(f)g}{f^{k+1}}.$$

Gröbner bases. The notion of Gröbner basis of a (left) ideal can be defined for Weyl algebras in the same way as it is defined in the case of polynomials. The Buchberger algorithm for computing Gröbner bases works, leading to algorithms for computing intersections of ideals, kernels of maps, syzygy modules, resolutions, etc.

A good reference on Gröbner bases for the Weyl algebra is [31], also see [19] for Gröbner techniques for algebras of solvable type, a.k.a. Gröbner-ready. We make some remarks on parametric Gröbner bases in Chapter 2 and study a possible parallelization of the Buchberger algorithm in Chapter 4.

Chapter 2

Stratification by Bernstein-Sato polynomials

For every polynomial $f \in R_n(k)$ there are $b(s) \in k[s]$ and $Q(x, \partial, s) \in A_n(k)[s]$ such that

$$b(s)f^s = Q(x, \partial, s) \cdot f^{s+1}. \tag{2.0}$$

For the proof of existence of such $b(s) \neq 0$ see [5], for example. The polynomials $b(s)$ for which equation (2) exists form an ideal in $k[s]$. The monic generator of this ideal is denoted by $b_f(s)$ and called the *Bernstein-Sato polynomial* of f, which was first introduced by Bernstein in [4] and is also called the *global b-function* of f (e.g. in [31]).

The simplest characteristics of a polynomial f are its degree d and its number of variables n. This paper is motivated by the following natural question: what can one say about $b_f(s)$ in terms of n and d? We give what may be regarded as a complete answer to this question. Namely, we describe an algorithm that for fixed n and d gives a complete list of all possible Bernstein-Sato polynomials and, for each polynomial $b(s)$ in this list, a complete description of the polynomials f such that $b_f(s) = b(s)$.

Let $\mathcal{P}(n, d; k)$ be the set of all the non-zero polynomials of degree at most d in n variables with coefficients in k and let $P(n, d; k)$ be $\mathcal{P}(n, d; k)$ modulo the equivalence relation $f \sim g \Leftrightarrow f = c \cdot g$ for some $0 \neq c \in k$. Note that $b_f(s) = b_g(s)$ if $f \sim g$. We view $P(n, d; k)$ as the set of the k-rational points of the projective space $\mathbb{P}(n, d; k) \cong \mathbb{P}_k^{N-1}$ where N is the number of monomials in n variables of degree at most d. Lyubeznik in [25] defined $B(n, d)$ as the set of all the Bernstein-

Sato polynomials of all the polynomials from $\mathcal{P}(n, d; k)$ as k varies over all fields of characteristic 0 and proved that $B(n, d)$ is a finite set. He also asked if the subset of $\mathcal{P}(n, d; k)$ corresponding to a given element of $B(n, d)$ is constructible. We will show that the corresponding subset of $\mathbb{P}(n, d; k)$ is indeed constructible thus giving an affirmative answer to Lyubeznik's question. It turns out that these constructible sets can be defined over $\mathbb{Q}$, i.e. their defining equations and inequalities are the same for all fields k.

A crucial ingredient in our proof is an algorithm discovered by Oaku [28] that given a polynomial f returns its Bernstein-Sato polynomial $b_f(s)$. Using Oaku's algorithm we have developed an algorithm that computes the complete set of the Bernstein-Sato polynomials $B(n, d)$ for each pair (n, d) and for each $b(s) \in B(n, d)$ constructs a finite number of locally closed sets $V_i = V_i' \setminus V_i''$, where V_i' and V_i'' are Zariski closed subsets of $\mathbb{P}(n, d; \mathbb{Q})$ defined by explicit polynomial equations with rational coefficients, such that for every field k of characteristic 0, the subset of $P(n, d; k)$ having $b(s)$ as the Bernstein-Sato polynomial is the set of k-rational points of

$$S(b(s), k) = (\cup_i V_i) \otimes_{\mathrm{Spec}\mathbb{Q}} \mathrm{Spec} k \subset \mathbb{P}(n, d; \mathbb{Q}) \otimes_{\mathrm{Spec}\mathbb{Q}} \mathrm{Spec} k = \mathbb{P}(n, d; k).$$

A similar approach applies also to computing the Bernstein-Sato polynomial of a polynomial with parameters. Namely, one can prove that the number of different possible Bernstein-Sato polynomials in this case is finite and the corresponding stratum for each of them is a constructible set in the space of parameters.

Moreover, using a similar technique we develop an algorithm for computing the annihilator of $\frac{1}{f^s}$ in $A_n(k)$, which, provided such s is known that $\frac{1}{f^s}$ generates $R_n(k)_f$, gives a presentation of $R_n(k)_f$ as an $A_n(k)$-module (see Example 28). This algorithm is particularly important for Walther's algorithmic computation of local cohomology modules [37].

These applications are discussed in Section 2.3.

A different approach to the question of constructibility was laid out recently in [6].

2.1 Preliminaries

In this section we have collected the ingredients for the main algorithm of this chapter. We define the canonical form of a constructible set, consider parametric Gröbner bases for Weyl algebras, give a description of Oaku's algorithm for computing the Bernstein-Sato polynomial, and finally discuss some properties of the Bernstein-Sato polynomial, in particular, the rationality of the roots of the Bernstein-Sato polynomial.

2.1.1 Constructible Sets

We recall that a set is constructible iff it is a finite union of locally closed sets and a set is locally closed iff it is the difference of two closed sets.

Lemma 1 *Let C be a constructible subset of a k-variety X. Then C may be presented uniquely as a disjoint union $\bigcup_{i=1}^{m}(V_i' \setminus V_i'')$, where for all i the sets V_i' and V_i'' are closed, $V_1' \supset V_1'' \supset V_2' \supset V_2'' \supset ... \supset V_m' \supset V_m''$ and no two sets in this chain have an irreducible component in common. We call it a canonical presentation of C as a union of locally closed subsets.*

Proof. Let $d(C)$ be the maximal dimension of an irreducible component in $\bar{C}$. Let $V_1' = \bar{C}$ and $V_1'' = \overline{V_1' \setminus C}$ and let $C' = C \cap V_1''$. Note that $d(C') < d(C)$ and we may assume by induction on d that the chain $V_2' \supset V_2'' \supset ... \supset V_m' \supset V_m''$ such that $C' = \bigcup_{i=2}^{m}(V_i' \setminus V_i'')$ exists and is unique. Then $V_1' \supset V_1'' \supset V_2' \supset V_2'' \supset ... \supset V_m' \supset V_m''$ is the unique chain for C, which satisfies the condition in the statement. ■

Remark 2 *There is an algorithmic way for constructing such a presentation, starting with C presented as a union of nonempty sets $W_\alpha \setminus (W_\alpha^{(1)} \cup ... \cup W_\alpha^{(h_\alpha)})$, where W_α and $W_\alpha^{(i)}$ are closed irreducible subsets and $W_\alpha \supset W_\alpha^{(i)}$ for all i. Let $d(C) = \max_\alpha \dim W_\alpha$ (which agrees with the definition in the proof of the theorem).*

Let V_1' be the union of all maximal elements in the set $\{W_\alpha\}$ and V_1'' be the union of all $W_\alpha^{(i)}$ that are minimal with the following property: there is a set of pairs $\{(\alpha_j, i_j)\}_{j=1}^{l}$ such that W_{α_1} is a component of V_1', $W_{\alpha_l}^{(i_l)} = W_\alpha^{(i)}$ and $W_{\alpha_j}^{(i_j)} \supset W_{\alpha_{j-1}}$ for all $j = 2, ..., l$. Now $d(C \setminus (V_1' \setminus V_1''))$ is less than $d(C)$, therefore, we may assume again by induction on d that we are able to construct the rest of V_i' and V_i''.

Lemma 3 *Let X be a variety and $f : X \to Y$ a map into any finite set Y. Then $f^{-1}(y)$ is constructible for every $y \in Y$ iff for every closed irreducible subvariety $X' \subset X$ there is an open $U \subset X'$ such that $f|_U$ is a constant function.*

Proof. Assume the second part holds. Take any $y \in Y$ and let us prove that $Z = f^{-1}(y)$ is constructible. Let $n = \dim X$ and assume the lemma is proved for dimensions less then n. First of all, since X is a finite union of its irreducible components, and a subset of X is constructible iff its intersection with every irreducible component of X is, we may proceed assuming that X is irreducible. Let U be an open subset of X such that $f(u) = y'$ for all $u \in U$. There are two possibilities:

(i) if $y' \neq y$ then $Z \subset X \backslash U$, which has dimension less than n and, therefore, Z is constructible by the induction assumption applied to the map $f|_{X\setminus U} : X \setminus U \to Y$.

(ii) in case $y = y'$ the set $(Z \setminus U) \subset (X \setminus U)$ is constructible by the induction assumption again, hence so is $Z = U \cup (Z \setminus U)$.

It remains to check the case $\dim X = 0$, in which $f^{-1}(y)$ is a finite set of points and is certainly constructible.

Conversely, assume that $f^{-1}(y)$ is constructible for every $y \in Y$. Let $X' \subset X$ be a closed irreducible subvariety. Then $X' = \bigcup_{y \in Y} X'_y$, where $X'_y = (f^{-1}(y) \cap X')$, and, since Y is a finite set and X' is irreducible, there exist y such that the closure of X'_y is equal to X'. But X'_y is constructible, hence it contains a nonempty open subset of X'. ■

2.1.2 Parametric Gröbner Bases

Here we describe an approach to computing parametric Gröbner bases in Weyl algebras. For a discussion of parametric Gröbner bases, which leads to the notion of comprehensive Gröbner bases, see [38] for the commutative case and [21] for the case of solvable algebras. However, everything that is needed for our purposes is stated and proved in this section.

Let $C = k[\bar{a}]$ ($\bar{a} = \{a_1, ..., a_m\}$) be the ring of parameters and $R = C\left\langle \bar{y}, \bar{x}, \bar{\partial} \right\rangle$ be the ring of non-commutative polynomials in $\bar{y} = \{y_1, ..., y_l\}$, $\bar{x} = \{x_1, ..., x_n\}$ and $\bar{\partial} = \{\partial_1, ..., \partial_n\}$ with coefficients in C, where $\bar{x}$ and $\bar{\partial}$ satisfy the same relations as in a Weyl algebra and $\bar{y}$ is contained in the center of R.

Definition 4 *For a prime P in C, we shall call the natural map $C \to k(P)$ as well as the induced map $R = C\left\langle \bar{y}, \bar{x}, \bar{\partial} \right\rangle \to k(P)\left\langle \bar{y}, \bar{x}, \bar{\partial} \right\rangle$, where $k(P)$ is the residue field at P, the* specialization *at the point P and denote both maps by σ_P.*

The next result is similar to Oaku's Proposition 7 in [28].

Let $<$ be an order on monomials in $\bar{a}$, $\bar{y}$, $\bar{x}$ and $\bar{\partial}$ such that every a_i is $<<$ than any of x_j, y_j or ∂_j (i.e. the order $<$ eliminates x_j, y_j and ∂_j). Assume G is a finite Gröbner basis of an ideal I of R, then we claim that $\sigma_P(G) = \{\sigma_P(g) \mid g \in G\}$ is a Gröbner basis of $\sigma_P(I)$ in $\sigma_P(R)$ for "almost" every $P \in \operatorname{Spec} C$.

In order to make this statement precise, we need to make some definitions. For a polynomial f let $inM(f)$ be the initial monomial, $inC(f)$ the initial coefficient such that $in(f) = inC(f) \cdot inM(f)$ is the initial term of f. Also for $f \in R$ let $inM_*(f) \in \left\langle \bar{y}, \bar{x}, \bar{\partial} \right\rangle$ and $inC_*(f) \in C$ be the initial monomial and the initial coefficient of f viewed as a polynomial in x, y, ∂ with coefficients in C with respect to $\prec$, the restriction of $<$ to $\langle \bar{y}, \bar{x}, \bar{\partial} \rangle$.

One obvious observation is that a specialization $\sigma_P : (R, <) \to (\sigma_P(R), \prec)$ preserves the order.

Lemma 5 *Let Q be an ideal contained in I and let $h = \prod_{g \in G \setminus Q} inC_*(g) \in C$. Then $\sigma_P(G \setminus Q)$ is a Gröbner basis of $\sigma_P(I)$ for every prime $P \supset Q$ not containing h.*

Proof. Notice that if any $g \in G \setminus Q$ has $inC_*(g) \in Q$ then the statement of the lemma becomes trivial.

Assume $inC_*(g) \notin Q$ for all $g \in G \setminus Q$. Consider any prime $P \supset Q$ not containing h. Take a polynomial $f' = \sum_{g \in G \setminus Q} \frac{[\alpha_g]}{[\beta_g]} \sigma_P(g)$ in the ideal of $\sigma_P(R)$ generated by $\sigma_P(G)$, where $\alpha_g, \beta_g \in C$, $\beta_g \notin P$ and [...] stands for an equivalence class in C/P. Set $\gamma = \prod_{g \in G} \beta_g$ then $(\sum_{g \in G \setminus Q} \frac{\gamma \alpha_g}{\beta_g} g)$ is in the ideal I of R. Let f be the latter sum where all terms with coefficients in Q are set to zero. Then $f' = \frac{1}{[\gamma]} \sigma_P(f)$ and $inM_*(f) = inM(f')$. We have $inM(g) | inM(f)$ for some $g \in G \setminus Q$, which means that $inM_*(g) | inM_*(f)$. Now, $inM(\sigma_P(g)) = inM_*(g)$, because $inC_*(g) \notin P$. Thus $inM(\sigma_P(g)) | inM(\sigma_P(f))$, which proves that $\sigma_P(G)$ is a Gröbner basis. ■

Remark 6 *(i) The statement of the lemma is true for reduced Gröbner bases as well. The proof works almost verbatim.*

(ii) Clearly, $inC_*(g) \notin Q$ for all $g \in G \setminus Q$. Thus if Q is prime, then $h \notin Q$, hence the set of primes containing Q but not containing h is nonempty.

The lemma leads to the following

Algorithm 7

Input:	F':	a finite set of generators for a prime ideal $Q \subset C$.
	F:	a finite set of generators of a left ideal $I \subset R$ containing Q,
Output:	G:	a (reduced) Gröbner basis in R with respect to $<$,
	h:	an *exceptional polynomial* in $C \setminus Q$,
		such that for any $P \in \mathrm{Spec}(k[a_1, ..., a_m])$, $P \supset Q$ and $h \notin P$ the ideal $\sigma_P(I) \subset \sigma_P(R)$ has a $\sigma_P(G)$ as a (reduced) Gröbner basis with respect to $\prec$.

1. Compute a Gröbner basis G of $I + QR$ (which is generated by $F \cup F'$) .

2. Return G and $h = \prod_{g \in G \setminus Q} inC_*(g)$.

Remark 8 *If all polynomials in F' and all C-coefficients of all elements of F are homogeneous, then so is the exceptional polynomial h, because all operations preserve homogeneity.*

2.1.3 Oaku's Algorithm

The original algorithm of Oaku for computing the Bernstein-Sato polynomial appeared in [28]. However there exist several modifications of the algorithm (see [31] for example). For our needs a version of the algorithm described in [37] will be utilized.

Let $f \in R_n(k)$. Denote by $\mathrm{Ann} f^s$ the ideal of all elements in $A_n(k)[s]$ annihilating f^s. The following algorithm is Algorithm 4.4. from [37] with $L = (\partial_1, ..., \partial_n)$.

Algorithm 9

Input:	f:	a polynomial in $R_n(k)$,
Output:	$\{P'_j\}$:	generators of $\mathrm{Ann} f^s$

1. Set $Q = \{\partial_i + \frac{df}{dx_i}\partial_t, t - f\}$.

2. Introduce new variables y_1 and y_2 and the weight w such that $w(t) = w(y_1) = 1$, $w(\partial_t) = w(y_2) = -1$, $w(x_i) = w(\partial_i) = 0$. Homogenize all $q_i \in Q$ $(i = 1, ..., n+1)$ using y_1 with respect to the weight w. Denote the homogenized elements q_i^h.

3. Compute a Gröbner basis for the ideal generated by q_1^h, ... , q_{n+1}^h, $1 - y_1 y_2$ in $A_{n+1}[y_1, y_2]$ with respect to an order eliminating y_1, y_2.

4. Select the operators $\{P_j\}_1^b$ in this basis which do not contain y_1, y_2.

5. For each P_j, if $w(P_j) > 0$ then replace P_j by $P'_j = \partial_t^{w(P_j)} P_j$ else replace P_j by $P'_j = t^{-w(P_j)} P_j$.

6. Return the operators $\{P'_j\}_1^b$.

The following is Algorithm 4.6 in [37].

Algorithm 10

Input:	f :	a polynomial in $R_n(k)$,
Output:	$b_f(s)$	the Bernstein-Sato polynomial of f.

1. Determine $\mathrm{Ann} f^s$ following Algorithm 9.

2. Find a reduced Gröbner basis for the ideal $\mathrm{Ann} f^s + A_n[s] \cdot f$ using an order that eliminates x and ∂.

3. Return the unique element in the basis contained in $k[s]$.

2.1.4 Properties of the Bernstein-Sato polynomial

As was mentioned above roots of the Bernstein-Sato polynomial are of particular importance for computing localization, but also there is a connection of these roots with the local monodromy operator established by Malgrange [26].

For isolated singularities Bernstein-Sato polynomial makes a stronger invariant than local monodromy (see [39, §15]), on the other hand the classification of isolated singularities that identifies two singularities if one is an analytical deformation of the other (see Arnol'd [1, 2]) is finer than the one provided by Bernstein-Sato polynomials.

The first paper that discussed the rationality of the roots of the Bernstein-Sato polynomials was [26] by B. Malgrange. Using resolution of singularities, Kashiwara in [20] proved that the roots of local Bernstein-Sato polynomials are rational when $k = \mathbb{C}$. In [27, Prop. 4.2.1] it is proved that the Bernstein-Sato polynomial $b_f(s)$ is the lowest common multiple of the local Bernstein-Sato polynomials. Hence the roots of $b_f(s)$ are rational if $k = \mathbb{C}$.

In particular, it follows that $b_f(s) \in \mathbb{Q}[s]$. The fact that the roots are rational for every k is well-known to experts, but we have not been able to find a published proof, so we prove it in the next proposition.

Proposition 11 *Let k be a field, char $k = 0$. Then for every $f \in R_n(k)$ the roots of the Bernstein-Sato polynomial $b_f(s)$ are rational.*

Proof. The crucial fact is that if $K \subset k$ is a subfield containing all the coefficients of f, then the coefficients of $b_f(s)$ computed over k belong to K. This is because upon examining every step of Oaku's algorithm one sees that all calculations are done in K. Let K be a finite extension of $\mathbb{Q}$ containing the coefficients of f. Since one can embed K into $\mathbb{C}$, the Bernstein-Sato polynomial of f over K is the same as over $\mathbb{C}$. Now we are done by Kashiwara's result in conjunction with [27]. ∎

We can define the local Bernstein-Sato polynomial of a polynomial f at a point P by considering a functional equation similar to equation 2 in the definition of the (global) Bernstein-Sato polynomial with the only difference that everything is tensored over $R_n(k)$ with $R_n(k)_P$, which is our polynomial ring localized at the maximal ideal corresponding to the point P:

$$b(s)(f^s \otimes \bar{1}) = Q(x, \partial, s) \cdot (f \cdot f^s \otimes \bar{1}),$$

where $\bar{1} \in R_n(k)_P$.

It follows from the definition that the global Bernstein-Sato polynomial is the least common multiple of the local ones for all points P.

Also, it is true that local Bernstein-Sato polynomial at P is invariant under an analytic change of variables fixing P. This fact makes the local Bernstein-Sato polynomial an invariant useful to study singularities of hypersurfaces.

2.2 The Main Results

Consider $\mathbb{P}(n, d; k)$ with the coordinate ring $C = k[\bar{a}]$, where $\bar{a} = \{a_\alpha : |\alpha| \leq d\}$. Let $f = \sum_{|\alpha| \leq d} a_\alpha x^\alpha$.

Definition 12 *Let $b(s) \in B(n, d)$. The set $S(b(s), k) \subset \mathbb{P}(n, d; k)$ is defined as the set of all the points $P \in \mathbb{P}(n, d; k)$ such that $b_{\sigma_P(f)}(s) = b(s)$. (We view points in $\mathbb{P}(n, d; k)$ as homogeneous primes in C. See Definition 4 for $\sigma_P(f)$.)*

Lyubeznik's proof that $B(n, d)$ is finite can be summarized as follows. Let the space of parameters $X = \operatorname{Spec} A$ be an irreducible variety with A a quotient ring of $\mathbb{Q}[\bar{a}]$. We consider the Bernstein-Sato polynomial $b(s)$ of polynomial f from the beginning of this section seen as a polynomial over the field of fractions of A. Using the rationality of $b(s)$ and clearing the denominators, we obtain a functional equation

$$Q(a, x, \partial, s) f^{s+1} = h(a) b(s) f^s.$$

This shows that outside the zeroes of $h \in A$, the Bernstein-Sato polynomial divides $b(s)$. Hence there are only finitely many possibilities for $b(s)$ outside the zeros of h, while the set of zeros of h has a smaller dimension than X. This allows an induction argument for the finiteness of $B(n,d)$. But to prove constructibility one needs to find such h that, outside its zeroes, the Bernstein-Sato polynomial equals $b(s)$. This is the idea of the proof.

Let Q be a homogeneous prime in C. Then $\sigma_Q(f)$ is a polynomial with coefficients in a field, hence $b_{f_Q}(s)$ may be computed. What would happen if we run Algorithm 10 trying to compute $b_{f_Q}(s)$ "lifting from $k(Q)$, the fraction field of C/Q, to C" at every single step of the algorithm? Notice that $\sigma_Q : C \to k(Q)$ has C/Q as its image. Since the steps of the algorithm that do not involve Gröbner bases computation do not involve division either, we have to worry only about the two steps that deal with Gröbner bases. Suppose for these two steps we used Algorithm 7 with F' generating Q, in particular we obtained the exceptional polynomials h_1 and h_2, both in C. Set $h = h_1 h_2 \in C$, then the output, which is going to be $b_{\sigma_Q(f)}(s)$, is also the Bernstein-Sato polynomial of $\sigma_P(f)$ for every $P \supset Q$ such that $h \notin P$. Thus we have

Algorithm 13

Input:	f:	a polynomial in $R_n(C)$,
	F':	generators of a homogeneous prime ideal Q,
Output:	$b(s)$:	a polynomial in $\mathbb{Q}[s]$,
	H:	generators of a homogeneous ideal in C such that
		$b(s) = b_{\sigma_P(f)}(s)$ for every point $P \in V' \backslash V'' \neq 0$,
		where $V' = V(Q)$ and $V'' = V(H)$ ($V'' \subset V' \subset \mathbb{P}(n,d;k)$).

1. Compute the polynomial $b(s)$ and the exceptional polynomial h as described above.

2. Return $b(s)$ and $\{h\} \cup F'$.

Proposition 14 *If we consider $C' = C \otimes_k k'$ and $f \otimes_k 1 \in R_n(C')$, where k' is an extension of k, then $b(s)$ is the Bernstein-Sato polynomial for any point in the set $(V' \otimes_{Spec\ k} Spec\ k') \setminus (V'' \otimes_{Spec\ k} Spec\ k')$.*

Proof. Let Q be as above, then QC' may not be prime anymore. Nevertheless, assume the computation above was done for $f \otimes 1 \in R_n(C')$ and QC' as input. This computation "stays within k", i.e. no operation introduces an element outside the old ring. The output of the algorithm would be the same as before, and we claim that for every prime $P' \supset QC'$ not containing $h \otimes 1$ the Bernstein-Sato polynomial of $\sigma_P(f \otimes 1)$ is equal to $b_{\sigma_Q(f)}(s)$. This is guaranteed by Lemma 5. $\blacksquare$

Remark 15 *One can prove easily that* $(V' \otimes_{Spec\ k} Spec\ k') \setminus (V'' \otimes_{Spec\ k} Spec\ k')$ *is nonempty (although we will not use this fact in the sequel) by showing that*

(i) QC' *is the intersection of its associated primes* $\{Q_i\} \subset Spec\ C'$*,*

(ii) no Q_i *contains* $h \otimes 1$*, for otherwise* $Q_i \cap C = Q$ *contains* h*.*

The next theorem gives an affirmative answer to Lyubeznik's question about the constructibility of the set $S(b(s), k)$ of Definition 12.

Theorem 16 *The set* $S(b(s), k)$ *is constructible for every* $b(s)$*.*

Proof. The proof follows from the algorithm. For the function $\phi : \mathbb{P}(n, d; k) \to B(n, d)$, $\phi(P) = b_{\sigma_P(f)}(s)$ the following is true. For every projective $V' \subset \mathbb{P}(n, d; k)$ there is an open set $U = V' \setminus V'' \subset V'$ such that $f|_U$ is a constant function. Therefore we may apply Lemma 3. $\blacksquare$

Algorithm 13 leads to the main algorithm of this chapter.

Algorithm 17 *Input:* $n, d \in \mathbb{N}$*.*

Output: The set of pairs $L = \{(b(s), S(b(s)))|\, b(s) \in B(n, d)\}$*, where* $S(b(s)) = S(b(s), \mathbb{Q}) \subset \mathbb{P}(n, d; \mathbb{Q})$*.*

1. Set $L := \emptyset$, $f := \sum_{|\alpha| \leq d} a_\alpha x^\alpha$.

2. Define the recursive procedure $\mathbf{BSP}(Q)$, where $Q \in \mathrm{Spec}(\mathbb{Q}[\bar{a}])$.

BSP$(Q) := \{$

Apply Algorithm 13 to $V(Q)$ and f

to get an ideal I in C and $b(s) \in \mathbb{Q}[s]$;

IF there is a pair $(b(s), S) \in L$

THEN replace it by $(b(s), S \cup (V(Q) \setminus V(I)))$

ELSE $L := L \cup \{(b(s), V(Q) \setminus V(I))\}$;

IF $V(I) \neq \emptyset$ THEN $\{$

Find the minimal primes $\{Q_i\}$ associated to I;

FOR each Q_i DO **BSP**(Q_i) ;

$\}$

$\}$

3. Run BSP(0).

Remark 18 *This algorithm returns some presentations for constructible sets $S(b(s), \mathbb{Q})$, the canonical presentations for which may be obtained by using the algorithm discussed in Remark 2.*

Corollary 19 *The set $S(b(s), k)$ is defined over $\mathbb{Q}$, i.e. there exist ideals $I_i \subset \mathbb{Q}[\bar{a}]$ and $J_i \subset \mathbb{Q}[\bar{a}]$ $(i = 1, ..., m)$ such that for any field k*

$$S(b(s), k) = \bigcup_i (V_i' \setminus V_i''),$$

where $V_i' = V(k[\bar{a}]I_i)$ is the zero set of the extension of I_i and $V_i'' = V(k[\bar{a}]J_i)$ is the zero set of the extension of J_i.

Proof. Since the core part of algorithm above is Algorithm 13, the statement of the corollary follows from Proposition 14 applied to the extension k of $\mathbb{Q}$. ■

Remark 20 *Given a polynomial with parameters one can use a similar approach to compute the stratification of the parameter space corresponding to the set of all possible Bernstein-Sato polynomials (see examples 24 and 25).*

The annihilators $\mathrm{Ann}(f^s)$ are computed using Algorithm 9 and the same technique as in the algorithm above. The output is a set of pairs $\{(I_i, V_i)\}$, where I_i are the ideals in $A_n(k)[\bar{a}][s]$ and V_i are locally closed sets, such that for any polynomial f with coefficients in k that corresponds to a point $P \in V_i$ the ideal $\mathrm{Ann}(f^s)$ equals $\sigma_P(I_i)$, the ideal I_i specialized to P.

After doing the above steps, the real life algorithm that produces Example 28 compresses its output in the following way. If (I_i, V_i) and (I_j, V_j) are two different pairs such that $\sigma_P(I_i) = \sigma_P(I_j)$ for all $P \in V_j$ then these two are replaced by the pair $(I_i, V_i \cup V_j)$.

Remark 21 *The stratification of the parameter space constructed by such computation is not unique. This is so because the annihilators, as opposed to Bernstein-Sato polynomials, depend on the parameters, making it possible to slice the space of parameters in many ways.*

2.3 Examples

Our algorithms have been implemented as scripts written in the Macaulay 2 programming language (see [15]). In this section we give some examples of actual computations and discuss possible uses of the results of computation.

Example 22 *If $n = 2$ and $d = 2$ then*

$$f = a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00},$$

so $P(2, 2; k)$ is the set of the k-rational points of the projective space $\mathbb{P}(2, 2; k) = \mathbb{P}^5_k$ with the homogeneous coordinate ring $k[a_{ij}]$, $i, j = 0, 1, 2$. In a matter of minutes our program produces

$$B(2, 2) = \{1,\ s + 1,\ (s + 1)^2,\ (s + 1)(s + \frac{1}{2})\}$$

and gives a description of the corresponding constructible sets of polynomials from $B(2, 2)$ which is essentially equivalent to the following:

- $b_f(s) = 1$ *iff* $f \in V_1 = V_1' \setminus V_1''$, *where* $V_1' = V(a_{1,1}, a_{0,1}, a_{0,2}, a_{1,0}, a_{2,0})$, *while* $V_1'' = V(a_{0,0})$,
- $b_f(s) = s + 1$ *iff* $f \in V_2 = (V_2' \setminus V_2'') \cup (V_3' \setminus V_3'')$, *where* $V_2' = V(0)$, $V_2'' = V(\gamma_1)$, $V_3' = V(\gamma_2, \gamma_3, \gamma_4)$, $V_3'' = V(\gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7, \gamma_8)$,
- $b_f(s) = (s + 1)^2$ *iff* $f \in V_4' \setminus V_4''$, *where* $V_4' = V(\gamma_1)$, $V_4'' = V(\gamma_2, \gamma_3, \gamma_4)$,
- $b_f(s) = (s + 1)(s + \frac{1}{2})$ *iff* $f \in V_5' \setminus V_5''$, *where* $V_5' = V(\gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7, \gamma_8)$, *while* $V_5'' = V(a_{1,1}, a_{0,1}, a_{0,2}, a_{1,0}, a_{2,0})$,

where γ_i may be looked up in this list:

$\gamma_1 = a_{0,2}a_{1,0}^2 - a_{0,1}a_{1,0}a_{1,1} + a_{0,0}a_{1,1}^2 + a_{0,1}^2 a_{2,0} - 4a_{0,0}a_{0,2}a_{2,0}$,

$\gamma_2 = 2a_{0,2}a_{1,0} - a_{0,1}a_{1,1}$,

$\gamma_3 = a_{1,0}a_{1,1} - 2a_{0,1}a_{2,0}$,

$\gamma_4 = a_{1,1}^2 - 4a_{0,2}a_{2,0}$,

$\gamma_5 = 2a_{0,2}a_{1,0} - a_{0,1}a_{1,1}$,

$\gamma_6 = a_{0,1}^2 - 4a_{0,0}a_{0,2}$,

$\gamma_7 = a_{0,1}a_{1,0} - 2a_{0,0}a_{1,1}$,

$\gamma_8 = a_{1,0}^2 - 4a_{0,0}a_{2,0}$.

It is not hard to see that this computation agrees with the well-known result that $b_f(s) = 1$ iff f is constant, $b_f(s) = s+1$ iff f is non-constant and non-singular, and $b_f(s) = (s+1)^2$ (resp. $b_f(s) = (s+1)(s+\frac{1}{2})$) iff f can be reduced to xy (resp. x^2) by a linear change of variables.

Example 23 (Cubic polynomials in 2 variables) *If $n = 2$ and $d = 3$ then*

$$\begin{aligned} f &= a_{3,0}x^3 + a_{2,1}x^2y + a_{1,2}xy^2 + a_{0,3}y^3 \\ &+ a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 + a_{1,0}x + a_{0,1}y + a_{0,0}, \end{aligned}$$

so $P(2,3;k)$ is the set of the k-rational points of $\mathbb{P}(2,3;k) = \mathbb{P}^9_k$ with the homogeneous coordinate ring that involves 10 variables. Our program exhausts all available memory, 128Mb, of the computer after about 3 hours and stops without producing an answer. However, a somewhat creative use of our program enables us to give a complete list of all the elements of $B(2,3)$ (but not the explicit descriptions of the constructible sets corresponding to each element of $B(2,3)$):

Since for any nonsingular polynomial its Bernstein-Sato polynomial is equal to $s+1$, it remains to consider the case where our $f \in \mathbb{P}(2,3;k)$ possesses a singularity at some point (x_0, y_0). Keeping in mind that the Bernstein-Sato polynomial is stable under any linear substitution of variables, we may get rid of its linear part via the substitution $x \mapsto x - x_0$, $y \mapsto y - y_0$, i.e. f takes the form

$$f = (ax^3 + bx^2y + cxy^2 + dy^3) + (a'x^2 + b'xy + c'y^2).$$

Now it is easy to see that by homogeneous linear transformation the quadratic part may be shaped to one of the forms 0, xy, x^2. Therefore it is enough to compute the Bernstein-Sato polynomial for the following polynomials:

$$\begin{aligned}
f_1 &= ax^3 + bx^2y + cxy^2 + dy^3, \\
f_2 &= (ax^3 + bx^2y + cxy^2 + dy^3) + xy, \\
f_3 &= (ax^3 + bx^2y + cxy^2 + dy^3) + x^2.
\end{aligned}$$

In each of the three cases our program produces the set of possible Bernstein-Sato polynomials and an explicit description of the corresponding constructible set in $\mathbb{A}_k^4$ (each f_i contains 4 indeterminate coefficients) for each element $b(s) \in B_{f_i}$. We omit these and list only the Bernstein-Sato polynomials:

$$\begin{aligned}
B_{f_1} &= \{\, (s+1)^2(s+\frac{2}{3})(s+\frac{4}{3}), \\
&\quad (s+1)^2(s+\frac{1}{2}), \\
&\quad (s+1)(s+\frac{2}{3})(s+\frac{1}{3})\,\}; \\
B_{f_2} &= \{\, (s+1)^2\,\}; \\
B_{f_3} &= \{\, (s+1)(s+\frac{7}{6})(s+\frac{5}{6}), \\
&\quad (s+1)^2(s+\frac{3}{4})(s+\frac{5}{4}), \\
&\quad (s+1)^2(s+\frac{1}{2}), \\
&\quad (s+1)(s+\frac{1}{2})\,\}.
\end{aligned}$$

Thus

$$
\begin{aligned}
B(2,3) \quad = \quad & \{\, (s+1)^2(s+\frac{2}{3})(s+\frac{4}{3}), \\
& (s+1)^2(s+\frac{1}{2}), \\
& (s+1)(s+\frac{2}{3})(s+\frac{1}{3}), \\
& (s+1)^2, \\
& (s+1)(s+\frac{7}{6})(s+\frac{5}{6}), \\
& (s+1)^2(s+\frac{3}{4})(s+\frac{5}{4}), \\
& (s+1)(s+\frac{1}{2}), \\
& s+1, \\
& 1\,\}.
\end{aligned}
$$

The efficiency of the algorithm and the current efficiency of computer hardware and software obstruct us from getting a complete description of the constructible sets that correspond to the polynomials above.

Here are a couple of examples of the computation for polynomials with parameters.

Example 24 *Let $f = x^3 + ax + b + cy^4 + y^2$, then*

- $b(s) = (s+1)$ *for* $V(0) \setminus (V(a^3c^2 + \frac{27}{4}b^2c^2 - \frac{27}{8}bc + \frac{27}{64}) \cup V(4a^3 + 27b^2))$,
- $b(s) = (s+1)^2$ *for* $(V(a^3c^2 + \frac{27}{4}b^2c^2 - \frac{27}{8}bc + \frac{27}{64}) \cup V(4a^3 + 27b^2)) \setminus (V(a, 4bc-1) \cup V(a,b))$,
- $b(s) = (s+1)(s+\frac{5}{6})(s+\frac{7}{6})$ *for* $V(a, 4bc-1) \cup V(a,b)$.

Example 25 *Let $f = x^2 + axy + by^2 + z^3 + cx^4$, then*

- $b(s) = (s+1)(s+\frac{4}{3})(s+\frac{5}{3})$ *for* $V(0) \setminus V(a^2 - 4b)$,
- $b(s) = (s+1)(s+\frac{4}{3})(s+\frac{5}{3})(s+\frac{13}{12})(s+\frac{17}{12})(s+\frac{19}{12})(s+\frac{23}{12})$ *for* $V(a^2-4b) \setminus V(c, a^2-4b)$,
- $b(s) = (s+1)(s+\frac{5}{6})(s+\frac{7}{6})$ *for* $V(a,b) \cup V(c, a^2-4b)$.

Example 26 (Cubic polynomials in 3 variables) *This example (studied in collaboration with Josep Alvarez Montaner) generalizes Example 23, hence, there is even less hope for being able to compute $B(3,3)$ directly.*

Normal form	Type	Bernstein-Sato polynomial(s)
$x^2+y^2+z^2$	A_1	$(s+1)(s+\frac{3}{2})$
$x^2+y^2+z^3$	A_2	$(s+1)(s+\frac{4}{3})(s+\frac{5}{3})$
$x^2+y^2+z^4$	A_3	$(s+1)(s+\frac{5}{4})(s+\frac{3}{2})(s+\frac{7}{4})$
$x^2+y^2+z^5$	A_4	$(s+1)(s+\frac{6}{5})(s+\frac{7}{5})(s+\frac{8}{5})(s+\frac{9}{5})$
$x^2+y^2+z^6$	A_5	$(s+1)(s+\frac{7}{6})(s+\frac{4}{3})(s+\frac{3}{2})(s+\frac{5}{3})(s+\frac{11}{6})$
$x^2+z(y^2+z^2)$	D_4	$(s+1)(s+\frac{7}{6})(s+\frac{3}{2})(s+\frac{11}{6})$
$x^2+z(y^2+z^3)$	D_5	$(s+1)(s+\frac{9}{8})(s+\frac{11}{8})(s+\frac{3}{2})(s+\frac{13}{8})(s+\frac{15}{8})$
$x^2+y^3+z^4$	E_6	$(s+1)(s+\frac{13}{12})(s+\frac{4}{3})(s+\frac{17}{12})(s+\frac{19}{12})(s+\frac{5}{3})(s+\frac{23}{12})$
$x^3+y^3+z^3+3\lambda xyz$, where $\lambda^3 \neq -1$	$\tilde{E}_6$	$(s+1)^2(s+\frac{4}{3})(s+\frac{5}{3})(s+2),\ (s+1)^3$

Table 2.1: Isolated singularities on cubic surfaces

Since the Bernstein-Sato polynomial (global b-function) of $f \in R_3 = k[x,y,z]$ is the least common multiple of the local b-functions at singular points of the surface $f=0$ and these do not change under analytic transformations, in order to simplify the task of computing $B(3,3)$ it would be helpful to classify all cubic surfaces by their singularities.

Since the end of 19th century such classification has been known. A modern treatise of this problem from the point of view of the singularity theory can be found in [8]. It turns out that there are 9 types of isolated singularities (up to a local analytic transformation of coordinates) possible on cubic surfaces. The complete list of the normal forms of these singularities is in Table 2.1.

It is also known which combinations of singularities are possible (see Cases A,B,C,D in [8]). For each such combination the computation of the corresponding Bernstein-Sato polynomials would amount to taking the least common multiple of the corresponding local Bernstein-Sato polynomials in Table 2.1. All possible combinations of singularity types that may and do coexist on one cubic surface together with the corresponding Bernstein-Sato polynomials are collected in Table 2.2.

The remaining cases are surfaces with non-isolated singularities:

- **Irreducible.** This corresponds to Case E in [8], however, we have to be careful since there the projective case is considered. Using affine transformations changes of coordinates and technique similar to [8] we boil down this case to one of the polynomials in Table 2.3.

Combination(s)	Bernstein-Sato polynomial(s)
A_1	$(s+1)(s+\frac{3}{2})$
A_2	$(s+1)(s+\frac{4}{3})(s+\frac{5}{3})$
A_3 and A_1A_3	$(s+1)(s+\frac{5}{4})(s+\frac{3}{2})(s+\frac{7}{4})$
A_4	$(s+1)(s+\frac{6}{5})(s+\frac{7}{5})(s+\frac{8}{5})(s+\frac{9}{5})$
A_5 and A_1A_5	$(s+1)(s+\frac{7}{6})(s+\frac{4}{3})(s+\frac{3}{2})(s+\frac{5}{3})(s+\frac{11}{6})$
D_4	$(s+1)(s+\frac{7}{6})(s+\frac{3}{2})(s+\frac{11}{6})$
D_5	$(s+1)(s+\frac{9}{8})(s+\frac{11}{8})(s+\frac{3}{2})(s+\frac{13}{8})(s+\frac{15}{8})$
E_6	$(s+1)(s+\frac{13}{12})(s+\frac{4}{3})(s+\frac{17}{12})(s+\frac{19}{12})(s+\frac{5}{3})(s+\frac{23}{12})$
$\tilde{E}_6$	$(s+1)^2(s+\frac{4}{3})(s+\frac{5}{3})(s+2),\ (s+1)^3$
A_1A_2	$(s+1)(s+\frac{4}{3})(s+\frac{3}{2})(s+\frac{5}{3})$
A_1A_4	$(s+1)(s+\frac{6}{5})(s+\frac{7}{5})(s+\frac{3}{2})(s+\frac{8}{5})(s+\frac{9}{5})$

Table 2.2: Possible combinations of isolated singularities on cubic surfaces

f	$b_f(s)$
$xyz+x^3+y^3$	$(s+1)^3(s+\frac{4}{3})(s+\frac{5}{3})$
$xyz+ax^3+y^3+x^2$	$(s+1)^3(s+\frac{4}{3})(s+\frac{3}{2})(s+\frac{5}{3})$
$xyz+ax^3+by^3+x^2+y^2$	$(s+1)^2(s+\frac{3}{2})$
x^2z+y^3	$(s+1)(s+\frac{5}{6})(s+\frac{7}{6})(s+\frac{4}{3})(s+\frac{5}{3})$
$x^2z+axy^2+by^3+y^2$	$(s+1)^2(s+\frac{3}{2})$
$x^2z+xy+y^3$	$(s+1)^2$
irreducible polynomials in two variables	see Example 24

Table 2.3: Irreducible cubic surfaces

f	possible $b_f(s)$
$x(x+y^2+z^2+x(ax+by+cz))$	$(s+1)^3(s+\frac{3}{2})$
$x(x+y^2+x(ax+by+cz))$	$(s+1)^2(s+\frac{3}{4})(s+\frac{5}{4})$
$x(x+x(ax+by+cz))$	$(s+1)^2(s+\frac{1}{2})$ $(s+1)(s+\frac{1}{2})$
$x(y+z^2+ax^2+bxy+cy^2)$	$(s+1)^2$ $(s+1)^2(s+\frac{3}{2})$
$x(y+zx+y(ay+bz))$	$(s+1)^3(s+\frac{3}{2})$ $(s+1)^2$
x^3	$(s+1)(s+\frac{2}{3})(s+\frac{1}{3})$
$x(y^2+x(ax+by+cz))$	$(s+1)^2(s+\frac{5}{4})(s+\frac{3}{4}),,$ $(s+1)^2(s+\frac{4}{3})(s+\frac{2}{3})$ $(s+1)^2(s+\frac{1}{2})$
$x(yz+x(ax+by+cz))$	$(s+1)^3(s+\frac{4}{3})(s+\frac{5}{3})$ $(s+1)^3$

Table 2.4: Reducible cubic surfaces

- **Reducible.** This corresponds to Case F in [8]. If the surface is reducible then at least one of its irreducible components has to be a plane. Assume this plane is $x = 0$, then one may subdivide reducible surfaces in five families each parameterized by three variables as in Table 2.4.

In the latter case a finer stratification is possible, however, what we have is already fine enough in order to be treated by our algorithm.

Remark 27 *We covered all possible cubic surfaces in above discussion, therefore, $B(3,3)$ is computed. Though we know that each strata in $\mathbb{P}(3,3;k)$ corresponding to a fixed Bernstein-Sato polynomial is constructible, it seems to be impossible (at least at the moment) to compute these strata with our direct algorithm, since the coordinate ring of $\mathbb{P}(3,3;k)$ is a polynomial ring in 20 variables.*

*Given a cubic polynomial a more practical approach would be to compute its singular locus and try to classify its singularities. The former is computationally easy, the latter is doable at least in case of isolated singularities via one of the **Singular** [16] packages.*

Using a technique similar to that for computing Bernstein-Sato polynomials, we constructed an algorithm for computing of $\mathrm{Ann} f^s$, the annihilator ideal of f^s in $A_n(k)[s]$, for all $f \in P(n,d;k)$. By this we mean an explicit subdivision of $\mathbb{P}(n,d;k)$ into a finite union of constructible subsets and for each such subset V, an explicit finite set of elements $\beta_1, \beta_2, ... \in A_n(k)[a_{i_1 \dots i_n}][s]$ with $i_1 + ... + i_n \leq d$, such that $\mathrm{Ann}(f^s) = (\beta_1', \beta_2', ...)$ for every $f \in V$, where β_i' is the image of β_i under the specialization of the $a_{i_1 \dots i_n}$ to the corresponding coefficients of f.

Example 28 *To make the results obtained for $P(2,2;k)$ compact we need the following polynomials:*

$\beta_1 = a_{1,1}x_1\partial_1 + 2a_{0,2}x_2\partial_1 - 2a_{2,0}x_1\partial_2 - a_{1,1}x_2\partial_2 + a_{0,1}\partial_1 - a_{1,0}\partial_2,$

$\beta_2 = a_{1,1}a_{2,0}x_1^2\partial_1 + a_{1,1}^2x_1x_2\partial_1 + a_{0,2}a_{1,1}x_2^2\partial_1 - 2a_{2,0}^2x_1^2\partial_2 - 2a_{1,1}a_{2,0}x_1x_2\partial_2$
$-2a_{0,2}a_{2,0}x_2^2\partial_2 - a_{1,1}^2sx_2 + 4a_{0,2}a_{2,0}sx_2 + a_{1,0}a_{1,1}x_1\partial_1 + a_{0,1}a_{1,1}x_2\partial_1$
$-2a_{1,0}a_{2,0}x_1\partial_2 - 2a_{0,1}a_{2,0}x_2\partial_2 - a_{1,0}a_{1,1}s + 2a_{0,1}a_{2,0}s + a_{0,0}a_{1,1}\partial_1 - 2a_{0,0}a_{2,0}\partial_2,$

$\beta_3 = a_{2,0}x_1^2\partial_2 + a_{1,1}x_1x_2\partial_2 + a_{0,2}x_2^2\partial_2 - a_{1,1}sx_1 - 2a_{0,2}sx_2 + a_{1,0}x_1\partial_2$
$+a_{0,1}x_2\partial_2 - a_{0,1}s + a_{0,0}\partial_2,$

$\beta_4 = a_{1,1}^2x_1\partial_1 - 4a_{0,2}a_{2,0}x_1\partial_1 + a_{1,1}^2x_2\partial_2 - 4a_{0,2}a_{2,0}x_2\partial_2 - 2a_{1,1}^2s$
$+8a_{0,2}a_{2,0}s - 2a_{0,2}a_{1,0}\partial_1 + a_{0,1}a_{1,1}\partial_1 + a_{1,0}a_{1,1}\partial_2 - 2a_{0,1}a_{2,0}\partial_2,$

$\beta_5 = a_{1,1}\partial_1 - 2a_{2,0}\partial_2,$

$\beta_6 = 2a_{2,0}x_1\partial_2 + a_{1,1}x_2\partial_2 - 2a_{1,1}s + a_{1,0}\partial_2,$

$\beta_7 = \partial_1,$

$\beta_8 = 2a_{0,2}x_2\partial_2 - 4a_{0,2}s + a_{0,1}\partial_2,$

$\beta_9 = \partial_2,$

$\beta_{10} = 2a_{2,0}x_1\partial_1 - 4a_{2,0}s + a_{1,0}\partial_1,$

$\beta_{11} = a_{2,0}x_1^2\partial_1 - 2a_{2,0}sx_1 + a_{1,0}x_1\partial_1 - a_{1,0}s + a_{0,0}\partial_1,$

$\gamma_1 = a_{0,2}a_{1,0}^2 - a_{0,1}a_{1,0}a_{1,1} + a_{0,0}a_{1,1}^2 + a_{0,1}^2a_{2,0} - 4a_{0,0}a_{0,2}a_{2,0},$

$\gamma_2 = 2a_{0,2}a_{1,0} - a_{0,1}a_{1,1},$

$\gamma_3 = a_{1,0}a_{1,1} - 2a_{0,1}a_{2,0},$

$\gamma_4 = a_{1,1}^2 - 4a_{0,2}a_{2,0},$

$\gamma_5 = a_{0,1}^2 - 4a_{0,0}a_{0,2},$

$\gamma_6 = a_{0,1}a_{1,0} - 2a_{0,0}a_{1,1},$

$\gamma_7 = a_{1,0}^2 - 4a_{0,0}a_{2,0}.$

Here are all the possible annihilators together with their strata:

- *$Ann(f^s) = (\beta_1, \beta_2, \beta_3)$ for $f \in (V_1' \backslash V_1'') \cup (V_2' \backslash (V_{2,1}'' \cup V_{2,2}''))$, where $V_1' = V(0)$, $V_1'' = V(\gamma_1)$, $V_2' = V(\gamma_2, \gamma_3, \gamma_4)$, $V_{2,1}'' = V(a_{1,1},\ a_{0,2},\ a_{0,1})$ and $V_{2,2}'' = V(\gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7)$;*

- *Ann*$(f^s) = (\beta_1, \beta_4)$ *for* $f \in V_3' \setminus V_3''$, *where* $V_3' = V(\gamma_1)$, *while* $V_3'' = V(\gamma_2, \gamma_3, \gamma_4)$;
- *Ann*$(f^s) = (\beta_5, \beta_6)$ *for* $f \in V_4' \setminus (V_{4,1}'' \cup V_{4,2}'')$ *where* $V_4' = V(\gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7)$, $V_{4,1}'' = V(a_{1,0},\, a_{2,0},\, a_{1,1},\, \gamma_5)$ *and* $V_{4,2}'' = (a_{1,1},\, a_{0,2},\, a_{0,1},\, \gamma_7)$;
- *Ann*$(f^s) = (\beta_7, \beta_8)$ *for* $f \in V_5' \setminus V_5''$, *where* $V_5' = V(a_{1,0},\, a_{2,0},\, a_{1,1},\, \gamma_5)$, *while* $V_5'' = V(a_{1,1},\, a_{0,1},\, a_{0,2},\, a_{1,0},\, a_{2,0})$;
- *Ann*$(f^s) = (\beta_9, \beta_{10})$ *for* $f \in V_6' \setminus V_6''$, *where* $V_6' = V(a_{1,1},\, a_{0,2},\, a_{0,1},\, \gamma_7)$, *while* $V_5'' = V(a_{1,1},\, a_{0,1},\, a_{0,2},\, a_{1,0},\, a_{2,0})$;
- *Ann*$(f^s) = (\beta_9, \beta_{11})$ *for* $f \in (V_7' \setminus V_7'') \cup V_8'$, *where* $V_7' = V(a_{1,1},\, a_{0,2},\, a_{0,1})$, $V_7'' = V(a_{1,1},\, a_{0,2},\, a_{0,1},\, \gamma_7)$ *and* $V_8' = V(a_{1,1},\, a_{0,1},\, a_{0,2},\, a_{1,0},\, a_{2,0})$;

Chapter 3

On minimal generation of holonomic D-modules and Weyl algebra ideals

One of the distinctive properties of holonomic modules, is their finite length. Below we shall show that this property implies that **every holonomic module can be generated by one element**.

Another striking fact, which is very simple to state, but quite hard to prove, is that **for every left ideal of A_n there exist 2 elements that generate it**.

Both statements were proved by Stafford in [33]; also these results appear in [5]. Unfortunately, the arguments given by Stafford cannot be converted to algorithms straightforwardly. There are several obstacles to this, many of which one can overcome with the theory of Gröbner bases for Weyl algebras. However, the main difficulty is that both proofs contain an operation of taking an irreducible submodule of an A_n-module. To our best knowledge, there does not exist an algorithm for this; moreover, even if such algorithm is invented one should expect it to be quite involved.

We were able to modify the original proofs in such a way that computations are possible and implemented the corresponding algorithms in the computer algebra system *Macaulay 2* [15].

We have to mention that in their recent paper [18] Hillebrand and Schmale construct another effective modification of Stafford's proof which leads to an algorithm. We shall discuss the

k	is	a (commutative) field of characteristic 0,
A_r	$=$	$A_r(k) = k\langle x_1, ..., x_r, \partial_1, ..., \partial_r \rangle$,
A	is	a simple ring of infinite length as a left module over itself,
D	is	a skew field of characteristic 0,
K	is	a commutative subfield of D,
S	$=$	$D(x)\langle \partial \rangle$,
$S^{(m)}$	$=$	$S\varepsilon_1 + ... + S\varepsilon_m$, a free S-module of rank m,
$\delta_1, ..., \delta_m$	is	a finite set of K-linearly independent elements in $K\langle x, \partial \rangle$,
$\sigma(\alpha, f)$	$=$	$\sum_{i=1}^{m} \alpha \delta_i f \varepsilon_i \in S^{(m)},\ (\alpha \in S, f \in K\langle x, \partial \rangle)$,
$P(\alpha, f)$	$=$	$S\sigma(\alpha, f),$ submodule of $S^{(m)}$,
$\mathcal{D}_r$	is	the quotient ring of A_r,
$\mathcal{R}_r$	$=$	$\mathcal{D}_r(x_{r+1}, ..., x_n)\langle \partial_{r+1}, ..., \partial_n \rangle$,
$\mathcal{S}_r$	$=$	$\mathcal{D}_r(x_{r+1}, ..., x_n)\langle \partial_{r+1} \rangle$.

Table 3.1: Notation table for Chapter 3

differences of their and our approaches in the last section.

For the convenience of the reader we provide the notation lookup table. All of the symbols listed below show up sooner or later in this chapter along with more detailed definitions. With exception of some minor changes we tried to stick to the notation in [5].

3.1 Preliminaries

Several useful properties of Weyl algebras are discussed in this section. Also, we introduce a few rings that will come handy later on.

3.1.1 A_n is simple

To see that A_n is simple, i.e. has no nontrivial two-sided ideals, we notice that, for $f = \sum_i x^{\alpha_i} \partial^{\beta_i} \in A_n \setminus \{0\}$ in the standard form, $df/dx_r = \partial_r f - f \partial_r$ for $r = 1, ..., n$, where $\partial f / \partial x_r$ is the formal derivative of the above expression of f with respect to x_r. Similarly, $df/d\partial_r = fx_r - x_r f$ for the formal derivative with respect to ∂_r. Note that these formal derivatives as well as all the multiple derivatives of f belong to the two-sided ideal $A_n f A_n$.

Now assume $x^\alpha \partial^\beta$ is the leading term of f with respect to some total degree monomial ordering. We are going to perform $|\alpha| + |\beta|$ differentiations: for all $i = 1, ..., n$ differentiate f α_i times with respect to x_i and β_i times with respect to ∂_i. Under such operation the leading term becomes equal to $\prod_{i=1}^{n} \alpha_i!\beta_i!$ and all the other terms vanish. Since the derivatives of f don't leave $A_n f A_n$, we showed that there is a simple algorithm to find such $s_i, r_i \in A_n$ that

$$\sum_{i=1}^{m} s_i f r_i = 1.$$

Hence, $A_n f A_n = A_n$, so A_n is simple.

3.1.2 A_n is an Ore domain

Proposition 29 *A_n is an Ore domain, i.e. $A_n f \cap A_n g \neq 0$ and $f A_n \cap g A_n \neq 0$ for every $f, g \in A_n \setminus \{0\}$.*

Proof. See the proof of Proposition 8.4 in Björk [5]. ■

Let us point out that using Gröbner bases methods (see next subsection) we can find a left(right) common multiple of $f, g \in A_n \setminus \{0\}$, in other words we can find a nontrivial solution to the equations $af = bg$ and $fa = gb$ where a and b are unknowns.

3.1.3 More rings

There is a quotient ring D associated to every Ore domain A. The ring D is a skew field that can be constructed both as the ring of left fractions $a^{-1}b$ and as the ring of right fractions cd^{-1}, where $a, b, c, d \in A$. There is a detailed treatment of this issue in [5].

Let D be a skew field, we will be interested in the ring $S = D(x)\langle\partial\rangle$, which is a ring of differential operators with coefficients in $D(x)$. It is easy to see that S is simple.

Since the Weyl algebra A_r is an Ore domain, we can form its quotient ring, which we denote by $\mathcal{D}_r$. The S we are going to play with is $\mathcal{S}_r = \mathcal{D}_r(x_{r+2}, ..., x_n)(x_{r+1})\langle\partial_{r+1}\rangle$. Let us state without proof a proposition which shall help us to compute Gröbner bases in $\mathcal{S}_r$.

Proposition 30 *Let $F = \{f_1, ..., f_k\} \subset A_n$ is a generating set of left ideal I of $\mathcal{S}_r$. Compute a Gröbner basis $G = \{g_1, ..., g_m\}$ of $A_n \cdot F$ with respect to any monomial ordering eliminating ∂_{r+1}. Then G is contained in $\mathcal{S}_r \cap A_n$ and is a Gröbner basis of I.*

3.2 Holonomic modules are cyclic

In this section we consider a simple ring A of infinite length as a left module over itself. Note that A_n is such a ring.

Theorem 31 *Every left A-module M of finite length is cyclic. In particular every holonomic A_n module is cyclic.*

Suppose we know how to compute a cyclic generator for every module M' of length less than l. For length 0 such a generator would be 0.

Consider a module M of length l. Take $0 \neq \alpha \in M$. If $M = A\alpha$ then we are done. If not then since $l(M/A\alpha) < l$ by induction we can find β such that its image in $M/A\alpha$ is a cyclic generator. Now $M = A \cdot \{\alpha, \beta\}$ and what we need to prove is

Lemma 32 *Let M be a left A-module of finite length and $\alpha, \beta \in M$. Then there exists $\gamma \in M$ such that $A\gamma = A\alpha + A\beta$.*

Proof. Define two functions l_1 and l_2 for pair (α, β).

$$\begin{aligned} l_1(\alpha, \beta) &= \text{length}(A\beta) \\ l_2(\alpha, \beta) &= \text{length}\left((A\alpha + A\beta)/A\alpha\right). \end{aligned}$$

Let also introduce an order $<$ on the set of pairs $(\alpha, \beta) \in M \times M$:

$$\begin{aligned} (\alpha', \beta') < (\alpha, \beta) &\Leftrightarrow (l_1(\alpha', \beta'), l_2(\alpha', \beta')) <_{lex} (l_1(\alpha, \beta), l_2(\alpha, \beta)) \\ &\Leftrightarrow l_1(\alpha', \beta') < l_1(\alpha, \beta) \\ &\quad\ \text{OR } (l_1(\alpha', \beta') = l_1(\alpha, \beta) \text{ AND } l_2(\alpha', \beta') < l_2(\alpha, \beta)) \end{aligned}$$

Suppose for any pair $(\alpha', \beta') < (\alpha, \beta)$, we can find $\gamma' \in M$ such that $A\gamma' = A \cdot \{\alpha', \beta'\}$.

Let the ideals $L(\alpha)$ and $L(\beta)$ in A be the annihilators of α and β respectively. Since length$(A) = \infty$, we know that $L(\alpha) \neq 0$; pick any element $0 \neq f \in L(\alpha)$. Since A is simple we can find $s_i, r_i \in A$, $I = 1, ..., M$ such that

$$\sum_{i=1}^{m} s_i f r_i = 1. \tag{3.-5}$$

Consider two cases:

1. There is some $r = r_i$ such that $L(\beta) + L(\alpha)r = A$.

2. The opposite is true.

Case 1. We can write $1 = E_\alpha r + E_\beta$ for some $E_\alpha, E_\beta \in A$ such that $E_\alpha \alpha = 0$ and $E_\beta \beta = 0$. Let $\gamma = \alpha + r\beta$.

Now we can get β from γ:

$$\beta = (E_\alpha r + E_\beta)\,\beta = E_\alpha r\beta = E_\alpha \alpha + E_\alpha r\beta = E_\alpha \gamma.$$

Hence $\beta \in A\gamma$ and since $\alpha = \gamma - r\beta$ the module $M = A\alpha + A\beta$ is indeed generated by γ.

Case 2. From (3.2) it follows that $\sum L(\beta) + Afr_i = A$, hence, $\sum A(fr_i\beta) = A\beta$, so there is $r = r_i$ such that

$$A(fr\beta) \nsubseteq A\alpha. \tag{3.-5}$$

Since we are not in case 1, $L(\beta) + Afr \subset L(\beta) + L(\alpha)r \neq A$. Take this modulo $L(\beta)$ to get

$$A(fr\beta) \cong (L(\beta) + Afr)\,/L(\beta) \subsetneq A/L(\beta) \cong A\beta, \tag{3.-5}$$

so $A(fr\beta)$ is proper in $A\beta$.

The last statement implies $l_1(\alpha, fr\beta) < l_1(\alpha, \beta)$, hence, $(\alpha, fr\beta) < (\alpha, \beta)$, so by induction hypothesis we can find $\gamma' \in M$ such that $A\gamma' = A(fr\beta) + A\alpha$.

Now (3.2) guarantees that $l_2(\gamma', \beta) < l_2(\alpha, \beta)$, and by induction we can find γ for which

$$A\gamma = A\gamma' + A\beta = A(fr\beta) + A\alpha + A\beta = A\alpha + A\beta.$$

■

Remark 33 *There is an algorithm that finds a cyclic generator for a holonomic left module over a Weyl algebra, since every step in the proof of the Lemma 32 is computable. The most non-trivial and time consuming operation is producing the annihilators $L(\alpha+r\beta)$ and $L(fr\beta)$ in the proof of Lemma 32 provided $L(\alpha)$ and $L(\beta)$. This is done using Gröbner bases techniques.*

For computational purposes a finitely generated D-module is usually presented as a quotient of a free module A_n^m by the image of a D-matrix. Using such a presentation it is not hard to compute the initial data, namely, the annihilators $L(e_i)$ for each element e_i of the standard basis of A_n^m.

We have programmed the algorithm corresponding to the proof of Theorem 31 using *Macaulay 2*.

Example. Let us view the ring of polynomials $k[x]$ as an A_1-module under the natural action of differential operators. It has an irreducible module, because starting with a nonzero polynomial f we can obtain a nonzero constant by differentiating it $\deg(f)$ times. The module $M = k[x]^3$ is the direct sum of 3 copies of $k[x]$, is holonomic (length$(M) = 3$) and is generated by the vectors $(1,0,0),(0,1,0),(0,0,1)$. Our algorithm produces a cyclic generator $\gamma = (x^2, x, 1)$ and its A_1-annihilator $L(\gamma) = A_1\partial^3$.

```
i1 : load "D-modules.m2";

i2 : R = QQ[x, dx, WeylAlgebra => {x=>dx}];

i3 : M = matrix {{dx, 0, 0}, {0, dx, 0}, {0, 0, dx}}

o3 = | dx 0  0  |
     | 0  dx 0  |
     | 0  0  dx |

              3       3
o3 : Matrix R  <--- R

i4 : -- find a cyclic presentation of coker M
     h = makeCyclic M
                                  3
o4 = HashTable{AnnG => ideal dx    }
                    Generator => | x2 |
                                 | x  |
                                 | 1  |
```

3.3 Ideals are 2-generated

In this section we give an effective proof of

Theorem 34 *Every left ideal of the Weyl algebra A_n can be generated by two elements.*

Proof for A_1. In this case the theorem follows from the fact that module A_1/J is holonomic for any nonzero ideal J of A_1.

Indeed, let I be a left ideal of A_1. Pick $f \in I$ and set $J = A_1 f$. Then I/J is a submodule of the holonomic module A_1/J, hence, is holonomic. By Theorem 31 there is $\bar{g} \in I/J$ such that $A_1\bar{g} = I/J$. Find a lifting $g \in A_1$ such that $\bar{g} = g \mod J$. Elements f and g generate I. ∎

Although the algorithm that follows from the proof seems to be simple, it inherits the complexity of the algorithm for finding a cyclic generator. There is an easier approach to finding 2 generators for an ideal in case of 1 variable suggested by Briançon [7, Prop. 5]. In general, it is easy to show that for any monomial ordering the 2 elements of a Gröbner basis for a given ideal I such that their leading monomials are at the ends of the staircase corresponding to the initial ideal $\mathrm{in}(I)$, in fact, generate I.

Theorem 34 for $n > 1$ presents a significantly tougher challenge, which is met by the rest of the paper.

3.3.1 Lemmas for S

Let us explore some properties of $S = D(x)\langle\partial\rangle$, the ring of linear differential operators with coefficients in rational expressions in x over a skew field D.

Let K be a commutative subfield of D, let $\delta_1, ..., \delta_m$ be a finite set of K-linearly independent elements in $K\langle x, \partial\rangle \subset S$, and let $S^{(m)} = S\varepsilon_1 + ... + S\varepsilon_m$ be a free S-module of rank m.

Also define $\sigma(\alpha, f) \in S^{(m)}$ to be the following sum $\sigma(\alpha, f) = \sum_{i=1}^{m} \alpha\delta_i f \varepsilon_i$, and $P(\alpha, f) = S\sigma(\alpha, f)$ the submodule of $S^{(m)}$ generated by $\sigma(\alpha, f)$. Note that $\sigma(\alpha, f)$ is S-linear in α and respects addition in f.

Lemma 35 *Let $0 \neq \alpha \in S$ and let M be an S-submodule of $S^{(m)}$ generated by $\{\sigma(\alpha, f) | f \in K\langle x, \partial\rangle\}$. Then $M = S^{(m)}$.*

Proof. Without loss of generality let us assume that $\alpha \in D\langle x, \partial\rangle$: if not we can always find such $p \in D[x]$ that $p\alpha \in D\langle x, \partial\rangle$.

Fix a monomial ordering that respects the total degree in x and ∂. For vector $v = \sum v_i \varepsilon_i \in (D\langle x, \partial\rangle)^{(m)}$ denote by $\mathrm{lm}(v)$ the largest of the the leading monomials of the components v_i of v in this ordering and denote $\mathrm{supp}(v) := \{i : v_i \neq 0\}$.

Now start with vector $v = v^{(0)} = \sigma(\alpha, 1)$; its components $v_i = \alpha\delta_i$ are D-linearly independent. Note that computing expressions $\pi(v) = \partial v - v\partial$ and $\chi(v) = vx - xv$ has an effect of differentiating each component of v formally with respect to x and ∂ respectively. These operations lower the total degree of v by 1 if the differentiation is done with respect to a variable that is present in $\mathrm{lm}(v)$. Also, it is not hard to see that they keep us in module M; for example, for $v^{(0)}$ we have $\pi(v^{(0)}) = \partial v^{(0)} - v^{(0)}\partial = \partial\sigma(\alpha, 1) - \sigma(\alpha, \partial)$.

Run the following algorithm: initialize $v := v^{(0)}$, while $\mathrm{lm}(v)$ contains an x set $v := \pi(v)$, then while $\mathrm{lm}(v)$ contains a ∂ we set $v := \chi(v)$. Since each step lowers the total degree of v by 1, this procedure terminates producing vector $w \in M$ of total degree 0.

Hence, $w = w_{i_1}\varepsilon_{i_1} + ... + w_{i_t}\varepsilon_{i_t}$ where $0 \neq w_{i_j} \in D$ for $j = 1, ..., t$ where $\{i_1, ..., i_j\} \subset \mathrm{supp}(v)$. Via multiplying on the left by the inverse of w_{i_1} we can get the relation

$$\varepsilon_{i_1} = a_2\varepsilon_{i_2} + ... + a_t\varepsilon_{i_t} \mod M \tag{3.-5}$$

with $a_j \in D$ for $j = 2, ..., t$.

Now take $v^{(0)}$ and reduce it using (3.3.1). We get vector $v^{(1)}$ whose i_1-th component is 0, therefore $|\mathrm{supp}(v^{(1)})| < |\mathrm{supp}(v^{(0)})|$, and the remaining components are D-linearly independent, since the components of $v^{(0)}$ are.

Repeat the above algorithm for $v = v^{(1)}$ and so on. At the end we get a vector which is a scalar multiple of ε_i for some i, hence $e_i \in M$. Using relations (3.3.1) we see that all basis vectors ε_j, for $j = 1, ..., m$, are in M. ■

Remark 36 *From the proof it follows that given a submodule M of $S^{(m)}$ and $\alpha \in S$ one can find $f \in K\langle x, \partial\rangle$ such that $\sigma(\alpha, f) \notin M$ algorithmically.*

The next lemma is central in the proof of the result. Note that every step of the proof of the lemma can be carried out algorithmically.

Lemma 37 *Let M be an S-submodule of $S^{(m)} = S\varepsilon_1 + ... + S\varepsilon_m$ such that length$(S^{(m)}/M) < \infty$. For every $\alpha \in S$, we can find $f \in K\langle x, \partial\rangle$ such that $S^{(m)} = M + P(\alpha, f)$.*

Proof. Let $l = \mathrm{length}(S^{(m)}/M)$. Assume the assertion is proved for all M' such that $\mathrm{length}(S^{(m)}/M') < l$. Remark 36 says that we can find an $f \in K\langle x, \partial\rangle$ such that $\sigma(\alpha, f)$ doesn't belong to M.

For $t \in S$, $g \in K\langle x, \partial\rangle$ let us define two S-modules

$$
\begin{aligned}
N_1 &= M + P_1, \text{ where } P_1 = P(\alpha, g), \\
N_2 &= M + P_2, \text{ where } P_2 = P(t\alpha, g).
\end{aligned}
$$

Claim. *There is a module M' such that $M \subset M' \subset M + P(\alpha, f)$, $t \in S$, and $g \in K\langle x, \partial\rangle$ for which*

$$
\begin{aligned}
&t\sigma(\alpha, f) \in M, \\
&M' + P(t\alpha, g) = S^{(m)}, \\
&N_1 = N_2.
\end{aligned}
$$

To prove this we employ a second induction on length(M'/M). We start with $M' = M + P(\alpha, f)$. We can find $0 \neq t \in S$ such that $t\alpha \sum \delta_i f \varepsilon_i \in M$; it follows from S being Ore. By the first induction hypothesis, for M' and $t\alpha$ there exists $g \in K\langle x, \partial\rangle$ such that $M' + P(t\alpha, g) = S^{(m)}$. Notice that $N_1 \supset N_2$ and $M' + P_i = S^{(m)}$ for $i = 1, 2$. Also for $i = 1, 2$ we have

$$S^{(m)}/N_i = (M' + P_i)/(M + P_i) = M'/(M + M' \cap P_i).$$

If $\mathrm{length}(S^{(m)}/N_1) = \mathrm{length}(S^{(m)}/N_2)$ then $N_1 = N_2$ and we are done. We are done as well if $N_1 = S^{(m)}$. If both conditions above fail, by looking at the right hand side of 3.3.1 we determine that $M'' = M + M' \cap P_1$ both contains M and is contained in M' properly, plus $\mathrm{length}(M''/M) < \mathrm{length}(M'/M)$. Set $M' := M''$ and repeat the above procedure.

To finish the proof of the lemma we take M', t, g as in the claim and assert that $N' = M + P(\alpha, f + g)$ equals $S^{(m)}$. Indeed, $\sigma(t\alpha, f + g) = t\sigma(\alpha, f) + \sigma(t\alpha, g) = \sigma(t\alpha, g)$ modulo M, so $N_2 \subset N'$. But $N_1 = N_2$, thus $\sigma(\alpha, g) \in N'$, hence, $\sigma(\alpha, f) = \sigma(\alpha, f + g) - \sigma(\alpha, g) \in N'$. Now we see that $M' \subset N'$ and $P_2 \subset N'$. Since $M' + P_2 = S^{(m)}$, we proved $N' = S^{(m)}$. ■

3.3.2 Lemmas for $\mathcal{R}_r$

At this stage we shall specify the components in the definition of $S = D(x)\langle\partial\rangle$. We set $D = \mathcal{D}_r(x_{r+2}, ..., x_n)$, $x = x_{r+1}$ and $\partial = \partial_{r+1}$, so that new S is equal to $\mathcal{S}_r = \mathcal{D}_r(x_{r+1}, x_{r+2}, ..., x_n)\langle\partial_{r+1}\rangle$ which is a subring of $\mathcal{R}_r$. Also the commutative subfield K of D that showed up before is replaced by the k, the coefficient field from the definition of $A_n = A_n(k)$.

Proposition 38 *Let $\delta_1, ..., \delta_m$ be a finite set of k-linearly independent elements in $k\langle x_{r+1}, \partial_{r+1}\rangle$ and let $0 \neq \rho \in A_{r+1}[x_{r+2}, ..., x_n]$. Let $S^{(m+1)} = S\varepsilon_0 + S\varepsilon_1 + ... + S\varepsilon_m$ be a free S-module of rank $m+1$ And let $S^{(m+1)}\rho \subset S^{(m+1)}$ be its S-submodule generated by $\{\rho\varepsilon_0, \rho\varepsilon_1, ..., \rho\varepsilon_2\}$. Then there exists some $f \in k\langle x_{r+1}, \partial_{r+1}\rangle$ such that*

$$S^{(m+1)} = S^{(m+1)}\rho + S(\varepsilon_0 + \delta_1 f \varepsilon_1 + ... + \delta_m f \varepsilon_m).$$

Proof. Follows from Lemma 37 ■

Lemma 39 *Let $q \in A_r[x_{r+1}, ..., x_n]$ and let $a_1, ..., a_t$ be a finite set in A_n.*

Then there exists some $0 \neq \rho \in A_r[x_{r+1}, ..., x_n]$ such that $\rho a_j \in A_n q$ for all j.

Proof. See the proof of Lemma 8.5 in Björk [5]. ■

Let us point out that once we know that the statement of the lemma is true, we can compute the required ρ by finding a Gröbner basis of the module of syzygies of the columns of the matrix

$$\begin{pmatrix} a_1 & q & 0 & ... & 0 \\ a_2 & 0 & q & ... & 0 \\ ... & ... & ... & ... & ... \\ a_t & 0 & 0 & ... & q \end{pmatrix}$$

with respect to a monomial order that eliminates $\partial_{r+1}, ..., \partial_n$ and such that $\varepsilon_1 > \varepsilon_2 > ... > \varepsilon_{t+1}$ where $\varepsilon_1, \varepsilon_2, ..., \varepsilon_{t+1}$ is the basis (ε_i corresponds to the i-th column) of the free module A_n^{t+1} containing our submodule of syzygies. Such a Gröbner basis is guaranteed (by Lemma 39) to contain some syzygy producing the relation $\rho\varepsilon_1 + b_2\varepsilon_2 + ... + b_{t+1}\varepsilon_{t+1} = 0$ where $\rho \in A_r[x_{r+1}, ..., x_n]$, $b_i \in A_n$ for $i = 2, ..., n$. It is not hard to see that this is the ρ we need.

Lemma 40 *Let $0 \neq q \in A_{r+1}[x_{r+2}, ..., x_n]$ and let $u, v \in A_n$ with $v \neq 0$. Then there is some $f \in A_n$ such that $\mathcal{R}_r = \mathcal{R}_r q + \mathcal{R}_r(u + vf)$.*

Proof. Consider the following subring of A_n obtained by "removing" x_{r+1} and ∂_{r+1}:

$$A_{\widehat{r+1}} = k\langle x_1, ...x_r, x_{r+2}, ..., x_n, \partial_1, ..., \partial_r, \partial_r + 2, ..., \partial_n\rangle.$$

Now $A_n = A_{\widehat{r+1}} \otimes_k k\langle x_{r+1}, \partial_{r+1}\rangle$, so we can write $v = \delta_1 g_1 + ... + \delta_m g_m$ where $\delta_1, ..., \delta_m$ are elements of $k\langle x_{r+1}, \partial_{r+1}\rangle$ linearly independent over k and $g_1, ..., g_m \in A_{\widehat{r+1}}$. The ring $A_{\widehat{r+1}}$ is simple, since it is a Weyl algebra, thus we can find such $h_1, ..., h_l \in A_{\widehat{r+1}}$ that

$$A_{\widehat{r+1}} = \sum_{i=1}^{m} \sum_{j=0}^{l} A_{\widehat{r+1}} g_i h_j.$$

Since $A_{\widehat{r+1}}$ is a subring of $\mathcal{R}_r$ it means that $\mathcal{R}_r = \sum\sum \mathcal{R}_r g_i h_j$.

Sublemma. *For any $b_1, ..., b_m \in A_{\widehat{r+1}}$ there exists some $f \in k\langle x_{r+1}, \partial_{r+1}\rangle$ such that*

$$\mathcal{R}_r q + \mathcal{R}_r u + \mathcal{R}_r b_1 + ... + \mathcal{R}_r b_m = \mathcal{R}_r q + \mathcal{R}_r(u + \delta_1 f b_1 + ... + \delta_m f b_m).$$

Proof. It follows from Lemma 39 that there is $0 \neq \rho \in A_r[x_{r+1}, ..., x_n]$ such that $\rho b_1, ..., \rho b_m \in A_n q$ as well as $\rho u \in A_n q$. With the help from Proposition 38 we get $f \in k\langle x_{r+1}, \partial_{r+1}\rangle$ such that

$S^{(m+1)} = S^{(m+1)}\rho + S(\varepsilon_0 + \delta_1 f\varepsilon_1 + ... + \delta_m f\varepsilon_m)$ and since S is a subring of $\mathcal{R}_r$ we have

$$\mathcal{R}_r^{(m+1)} = \mathcal{R}_r^{(m+1)}\rho + \mathcal{R}_r(\varepsilon_0 + \delta_1 f\varepsilon_1 + ... + \delta_m f\varepsilon_m). \tag{3.-10}$$

Now map $\varepsilon_0 \mapsto u$ and $\varepsilon_i \mapsto b_i$ for all i; this map from $\mathcal{R}_r^m$ to $\mathcal{R}_r$ has its image equal to $\mathcal{R}_r q + \mathcal{R}_r u + \mathcal{R}_r b_1 + ... + \mathcal{R}_r b_m$ and maps the right hand side of (3.3.2) to a subset of $\mathcal{R}_r q + \mathcal{R}_r(u + \delta_1 f b_1 + ... + \delta_m f b_m)$, because $\rho u, \rho b_1, ..., \rho b_m \in A_n q$. Moreover these two expressions are equal, since it is easy to see that the latter is contained in the former as well. ∎

Proof of lemma continued. We apply our Sublemma to $b_i = g_i h_1$ $(i = 1, ..., m)$ to get $f_1 \in k\langle x_{r+1}, \partial_{r+1}\rangle$ such that

$$\mathcal{R}_r q + \mathcal{R}_r u + \sum_{j=1}^{m} \mathcal{R}_r g_i h_1 = \mathcal{R}_r q + \mathcal{R}_r(u + \sum_{j=1}^{m} \delta_i f_1 g_i h_1).$$

Since $v = \delta_1 g_1 + ... + \delta_m g_m$ and since f_1 commutes with all g_i, the last equation transforms into

$$\mathcal{R}_r q + \mathcal{R}_r u + \sum \mathcal{R}_r g_i h_1 = \mathcal{R}_r q + \mathcal{R}_r(u + v f_1 h_1).$$

Now reapply the Sublemma with u replaced by $u + v f_1 h_1$ and $b_i = g_i h_2$ $(i = 1, ..., m)$. As in the first step we get

$$\begin{aligned} &\mathcal{R}_r q + \mathcal{R}_r u + \sum \mathcal{R}_r g_i h_1 + \sum \mathcal{R}_r g_i h_2 \\ &= \mathcal{R}_r q + \mathcal{R}_r(u + v f_1 h_1) + \sum \mathcal{R}_r g_i h_2 \\ &= \mathcal{R}_r q + \mathcal{R}_r(u + v f_1 h_1 + f_2 h_2) \end{aligned}$$

for some $f_2 \in k\langle x_{r+1}, \partial_{r+1}\rangle$. After l many steps we arrive at

$$\mathcal{R}_r = \mathcal{R}_r q + \mathcal{R}_r u + \sum_{i=1}^{m}\sum_{j=1}^{l} \mathcal{R}_r g_i h_j = \mathcal{R}_r q + \mathcal{R}_r(u + v\sum_{j=1}^{l} f_i h_i),$$

which proves the lemma with $f = f_1 h_1 + ... + f_l h_l$. ∎

The following lemma follows from the previous one.

Lemma 41 *Let $0 \le r \le n-1$ and let $0 \ne q \in A_{r+1}[x_{r+2}, ..., x_n]$ and let $u, v \in A_n$ with $v \ne 0$. Then there is some $f \in A_n$,$q' \in A_r[x_{r+1}, ..., x_n]$ such that $q' \in A_n q + A_n(u + vf)$.*

Proof. It is easy to see that this lemma is equivalent to the previous one. ∎

3.3.3 Final chords

Proposition 42 (r) *Let $0 \leq r \leq n$, there is some $q_r \in A_r[x_{r+1}, ..., x_n]$ and $d_r, e_r \in A_n$ such that $q_r c \in A_n(a + d_r c) + A_n(b + e_r c)$.*

Proof. The statement is true for $r = n$, since A_n is Ore and $A_n c \cap (A_n a + A_n b)$.

Fix r. Assume that the statement is true for $r + 1, ..., n$, then there exist $q_{r+1}, d_{r+1}, e_{r+1}$ such that $q_{r+1}c \in A_n a' + A_n b'$, where $a' = a + d_{r+1}c$ and $b' = b + e_{r+1}c$. Hence we can write $q_{r+1}c = h_1 a' + h_2 b'$, where we can take $h_1 h_2 \neq 0$ since $A_n a' \cap A_n b' \neq 0$. Also since $h_1 A_n \cap h_2 A_n \neq 0$ we can also find g_1, g_2 satisfying $h_1 g_1 + h_2 g_2 = 0$, and since $A_n q_{r+1} c \cap A_n b' \neq 0$ there are s, t such that $s q_{r+1} c = t b'$. Using Lemma 41 to $q = q_{r+1}$ with $u = 0$ and $v = t g_2$, we get $q_r = q'$ and f such that $q_r = p_1 q_{r+1} + p_2 t g_2 f$ for some p_1, p_2. Summarizing, there exist such $h_1, h_2, g_1, g_2, s, t, p_1, p_2 \in A_n \setminus \{0\}$ that

$$q_r = p_1 q_{r+1} + p_2 t g_2 f$$
$$q_{r+1} c = h_1 a' + h_2 b'$$
$$h_1 g_1 + h_2 g_2 = 0$$
$$s q_{r+1} c = t b'$$

Using these 4 equations, make the following calculation: (In each section the underlined terms sum up to 0.)

$$\begin{aligned}
q_r c &= p_1 q_{r+1} c + p_2 t g_2 f c \\
&= p_1 q_{r+1} c - \underline{p_2 s q_{r+1} c} \\
&+ p_2 t g_2 f c + \underline{p_2 t b'} \\
&= (p_1 - p_2 s) q_{r+1} c + p_2 t (b' + g_2 f c) \\
&= (p_1 - p_2 s)(h_1 a' + h_2 b') + p_2 t (b' + g_2 f c) \\
&= (p_1 - p_2 s) h_1 a' + \underline{(p_1 - p_2 s) h_1 g_1 f c} \\
&+ (p_1 - p_2 s) h_2 b' + \underline{(p_1 - p_2 s) h_2 g_2 f c} + p_2 t (b' + g_2 f c) \\
&= (p_1 - p_2 s) h_1 (a' + g_1 f c) + ((p_1 - p_2 s) h_2 + p_2 t)(b' + g_2 f c).
\end{aligned}$$

Thus, with $d_r = d_{r+1} + g_1 f c$ and $e_r = e_{r+1} + g_2 f c$ the conclusion of the proposition holds. ■

The proposition above (for $r = 0$) shows that by "elimination" of variables ∂_i one at a time we can get such $d, e \in A_n$ that $q_0 c \in A_n(a + dc) + A_n(b + ec)$ where $q_0 \in k[x_1, ..., x_n]$. This proves a 50% version of Theorem 34:

Theorem 43 *Every ideal of $k(x_1, ..., x_n)\langle\partial_1, ..., \partial_n\rangle$ can be generated by two elements.*

To go the other 50% of the way one has to do a similar kind of "elimination" of x_i-s. This amounts to making copies of all lemmas that we stated for a slightly different set of rings. The trickiest part is considering ring $\mathcal{S}_r{}' = k(x_1, ..., x_r)\langle x_{r+1}, \partial_{r+1}\rangle$ instead of $\mathcal{S}_r$. In other words instead of a ring of type $D(x)\langle\partial\rangle$ where D is a skew field, we have to consider the first Weyl algebra $A_1(\mathcal{K})$ where $\mathcal{K}$ is a (commutative) field. Fortunately, analogues of Lemmas 35 and 37 for the latter ring can be effectively proved along the same lines.

Example 1. Consider A_3. For $a = \partial_1$, $b = \partial_2$, $c = \partial_3$ one can show that $A_3 \cdot \{a, b, c\} = A_3 \cdot \{a, b + x_1 c\}$. Indeed, the following calculation displays it:

$$c = (-x_1\partial_3 - \partial_2)a + \partial_1(b + x_1 c).$$

On the other hand our implementation of the algorithm in *Macaulay 2* produces the following:

```
i2 : R = QQ[x_1..x_3,D_1..D_3,WeylAlgebra=>{x_1=>D_1, ...

i3 : a = D_1; b = D_2; c = D_3;

i6 : I = stafford ideal (a,b,c)

                     2
o6 = ideal (D , x x D  + x D  + D )
             1   1 3 3    1 3    2

o6 : Ideal of QQ [x , x , x , D , D , D , WeylAlgebra => ...
                   1   2   3   1   2   3                 ...

i7 : ideal(a,b,c)== I

o7 = true
```

The second generator in the output is more involved than the one we had found before, nevertheless, this answer is valid as shown by the last line of the script.

Example 2. Let $a = \partial_1\partial_3^2$, $b = \partial_1\partial_2$, $c = \partial_2\partial_3^2$. Then our implementation of the algorithm shows that

$$A_3 \cdot \{a, b, c\} = A_3 \cdot \{a, b + (x_1^3 x_3^2 + x_1 x_3^2 + x_1 x_3 + 1)c\}.$$

3.4 Conclusion

The implementations of the algorithms constructed along the lines of the proofs of Theorems 31 and 34 in *Macaulay 2* work only on rather small examples for quite obvious reason: the complexity of the Gröbner bases computations in the Weyl algebra.

Though, for this reason, both the algorithm of Hillebrand and Schmale [18] and ours for finding two generators can not be considered as practical, we shall try to comment on the differences between the two.

A subroutine [18, Algorithm 3.6], which is central in the former algorithm, includes a step that goes through a certain set of polynomials in one variable, checking a certain property. At least one of polynomials in this – possibly large – set is guaranteed to satisfy this property, however the performed check is nontrivial and requires Gröbner bases.

In our algorithm, on the other hand, the main subroutines corresponding to Lemmas 35 and 37 take a more constructive approach.

Finally, we did not attempt to analyze the complexity of any of these algorithms. Since all of them break down rather fast, the question of complexity of the output is interesting only from a theoretical point of view. Using the arguments in this paper, one can construct a very rough bound as soon as a bound for the complexity of Gröbner basis computations in the Weyl algebra is known. To the best of our knowledge, the latter is an open problem.

Chapter 4

Parallel computation of Gröbner bases

In this chapter we describe a coarse-grain parallelization of Buchberger algorithm for computing Gröbner bases in algebras of linear differential operators. The obtained results are experimental rather than theoretical; this study belongs more to the area of computer science than to mathematics. The punch line of this piece of work is that we can benefit more from parallelization of Buchberger-type algorithms in a noncommutative setting than in a commutative one.

4.1 Introduction

Algorithms for computing Gröbner bases have become a standard hard problem for computer scientists, since their complexity is proved to have a sharp double exponential bound. That is in the case of (commutative) polynomials; if one considers, for example, the Weyl algebra, this complexity is not known, though it is guaranteed to be worse.

As parallel computation becomes standard and supercomputing facilities more accessible, we turn our attention to parallelization of Gröbner bases computations. This topic has been explored both by mathematicians and computer scientists (see [3], [30], [32], [9]) for Gröbner bases in rings of (commutative) polynomials.

At the beginning of the project, on top of aiming at constructing a practical implementation of a parallel Buchberger algorithm, we were especially interested in computing Gröbner bases in algebras of linear differential operators. For these the elementary operations (e.g. multiplication of differential operators) are more expensive than in case of polynomials, also there are no elaborate techniques (e.g. see [11]) developed for eliminating the unnecessary s-pairs (ones that reduce to 0), hence, optimizing the Buchberger. These observations made us believe that parallelization in the noncommutative case would yield better results compared to the commutative one for two reasons:

1. The cost of a reduction step dominates the cost of communication.

2. Updates to the basis are not often because of a larger number of 0-reductions.

The instincts have not betrayed us. Our implementation – coded in C++ using MPI for communication interface – displays speedups that on average are better in a noncommutative setting than in a commutative one.

4.2 Preliminaries

In this chapter we shall deal with associative algebras of polynomial type over a field k. In constructing those we use the variables $x_i, \partial_i, s_i, t_i$ where $i \in \mathbb{Z}$ that satisfy the following relations: $[\partial_i, x_i] = 1, [s_i, t_i] = t_i$, where $[a, b] = ab - ba$, and all the pairs of variables that are not mentioned above commute.

Using these variables we can describe the Weyl algebra $A_n = k\langle x_1, ..., x_n, \partial_1, ..., \partial_n\rangle$. Another algebra we used for our experiments is the so-called PBW algebra (see [6])

$$P_{n,p} = A_n\langle s_1, ..., s_p, t_1, ..., t_p\rangle = k\langle s_1, ..., s_p, t_1, ..., t_p, x_1, ..., x_n, \partial_1, ..., \partial_n\rangle.$$

If we fix the order of the variables, each element of any algebra allowed by the above description can be written uniquely as a polynomial with monomials with variables written in the specified order: we call this polynomial the *normal form*, for computational purposes we would assume that we always operate with algebra elements in the normal form. According to [19] our algebras are of *solvable type*, i.e. are eligible for Gröbner bases techniques similar to the ones in the ring of polynomials.

In what follows we describe a simple version of Buchberger algorithm, which is a (sequential) completion algorithm used to compute Gröbner bases.

Let $A = k\langle z\rangle = k\langle z_1, ..., z_n\rangle$ be an associative algebra where variables z_i have names from the list at the beginning of the section and are subject to the corresponding relations. Let f be an element of A having the normal form

$$f = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} a_\alpha z^\alpha,$$

where all but a finite number of a_α equal 0. Denote by $\mathrm{inM}(f)$ and $\mathrm{inC}(f)$ the *initial monomial* and *initial coefficient* of f, call $\mathrm{in}(f) = \mathrm{inC}(f)\mathrm{inM}(f)$ the *initial term* of f. For a left ideal $I \subset A$ we define an *initial ideal* $\mathrm{in}(I) = A \cdot \{\mathrm{in}(f) \mid f \in I\}$.

A generating set G of a left ideal $I \subset A$ is called a Gröbner basis if $\mathrm{in}(I) = A\,\{\mathrm{in}(g) \mid g \in G\}$.

Given a polynomial $f \in A$ and a finite subset $B \subset A$ we can perform reduction as follows:

Algorithm 44 $f' = \mathbf{REDUCE}(f, B)$

$f' := f$

WHILE there is $g \in B$ such that inM(g) divides inM(f')

Set $f' := f' - (in(f')/in(g))g$

END WHILE

For two polynomials $f, g \in A$ we define the s-polynomial

$$\mathrm{sPoly}(f, g) = \mathrm{inC}(g)\frac{l}{\mathrm{inM}(f)}f - \mathrm{inC}(f)\frac{l}{\mathrm{inM}(g)}g, \text{ where } l = \mathrm{lcm}(\mathrm{inM}(f), \mathrm{inM}(g)).$$

Then an alternative definition could be given as follows: A generating set G of a left ideal $I \subset A$ is a Gröbner basis if $\mathbf{REDUCE}(\mathrm{sPoly}(f, g), G) = 0$.

The following is a sketch of the simplest completion algorithm known as Buchberger algorithm.

Algorithm 45 $G = \mathbf{BUCHBERGER}(B)$

$G := B,\ S := \{(f, g) | f, g \in B,\ \ f \neq g\}$

WHILE $S \neq \emptyset$ do

Pick $(f, g) \in S$ according to the "strategy", $S := S \setminus \{(f, g)\}$

$h := \mathbf{REDUCE}(sPoly(f, g), G)$

IF $h \neq 0$

$S' := \{(h, g) | g \in G\}$

Eliminate redundant s-pairs from S and S' according to "criteria"

$S := S \cup S'$ and $G := G \cup \{h\}$

END IF

END WHILE

Here "strategy" refers to an algorithm of determining which s-pair to consider next; the most popular strategies are sorting s-pairs either by total degree or by "sugar" degree (see [14]).

"Criteria" are sets of rules helping to eliminate redundant s-pairs, i.e. the ones that are guaranteed to reduce to 0 under selected strategy. Most commonly used criteria are the ones you may find in Gebauer-Möller [13]. Several of these criteria generalize for the noncommutative setting, however, there are some that work only in the commutative case. A simple example is the so-called T-criterion: if $\gcd(\mathrm{inM}(f), \mathrm{inM}(g)) = 1$ then $\mathbf{REDUCE}(sPoly(f,g), \{f,g\}) = 0$, which does not hold for $f = \partial$ and $g = x$ in the Weyl algebra A_1.

4.3 Parallel Buchberger algorithm

As you may see, though we have a loop in the above algorithm and the **REDUCE** tasks at each iteration seem routine, these tasks are not independent of each other, since the reduction basis G may change from one step to another.

On the other hand, it is not hard to imagine many s-pairs in a row reducing to 0: if we knew ahead of time that this would happen we could have assigned these routine 0-reductions to different processors. This point, in our opinion, is the starting motivation for coarse-grain parallelization of Buchberger algorithm.

What we developed is an algorithm similar to Attardi-Traverso [3]. It uses master-slave paradigm and the distributed computing in the following way. Let $n+1$ threads be at our disposal, each assigned to a node with one processor. Make one of the the master and the rest the slaves; the master can communicate with each of the slaves, however, there is no communication between the slaves. Each slave maintains a local copy of reduction basis $B \in A$ and is capable of completing two tasks: updating for the basis upon reception of an update message from the master and reducing a polynomial $h \in A$ supplied by the master with respect to B.

Algorithm 46 SLAVE

$B := \emptyset$

LOOP

IF $\mathbf{RECEIVE}(\mathbf{MASTER}, h, \texttt{update})$ *THEN* $B := B \cup \{h\}$

IF $\mathbf{RECEIVE}(\mathbf{MASTER}, h, \texttt{reduce})$ *THEN*

$\mathbf{SEND}(\mathbf{MASTER}, \mathbf{REDUCE}(h, B), \texttt{slaveDone})$

END IF

END LOOP

Here **SEND**($THREAD, DATA, TAG$) sends message tagged with TAG containing $DATA$ to $THREAD$ and **RECEIVE**($THREAD, DATA, TAG$) receives message with TAG containing $DATA$ from $THREAD$. **SEND** is nonblocking and **RECEIVE** returns $TRUE$ if the message has been received. Note that, in practice, a slave is terminated by a separate message from the master, but here we prefer to use an infinite loop for simplicity.

Let us describe the master now. In what follows an s-pair shall be represented by a quadruple $(f, g, h, \texttt{status})$ containing two additional elements: h is a partially reduced s-polynomial sPoly(f, g) and `status` is either `red` or `nonRed` depending on whether the s-polynomial has been reduced completely or h may still be reduced with respect to the current basis G. The variable m_i stores the s-pair that is being reduced by $\mathbf{SLAVE}_i$, $i = 1, ..., n$.

Algorithm 47 $G = \mathbf{MASTER}(B)$. *Computes a Gröbner basis of the left ideal generated by $B \subset A$ using threads $\mathbf{SLAVE}_i$, $i = 1, ..., n$.*

FOR $i = 1, ..., n$

  $m_i := \emptyset$

  FOR every $g \in B, \mathbf{SEND}(\mathbf{SLAVE}_i, g, \texttt{update})$

END FOR

$G := B$

$S := \{(f, g, sPoly(f, g), \texttt{nonRed}) | f, g \in B,\ f \neq g\}$

WHILE $S \neq \emptyset$ *do*

  IF $\mathbf{RECEIVE}(\mathbf{SLAVE}_i, h, \texttt{slaveDone})$ *THEN*

    Let $(f, g, ..., \texttt{nonRed}) = m_i$

    Replace m_i *with* $(f, g, h, \texttt{red})$ *in* S

    $m_i := \emptyset$

  END IF

  IF $\exists m_i = \emptyset$ *and* $\exists sp = (f, g, h, \texttt{nonRed}) \in S$ *THEN*

    $\mathbf{SEND}(\mathbf{SLAVE}_i, h, \texttt{reduce})$

    $m_i := sp$

  END IF

  Let $sp = (f, g, h, \sigma)$ *be the first s-pair in* S

  IF $\sigma = \texttt{red}$ *THEN*

    Remove sp *from* S

    IF $h \neq 0$ *THEN*

      $S' := \{(h, g, sPoly(f, g), \texttt{nonRed}) | g \in G\}$

Apply "criteria" to S and S'

$S := S \cup S'$ *and* $G := G \cup \{h\}$

Reorder S according to the "strategy"

FOR every $(f, g, h', \texttt{red}) \in S$

IF h divides h' THEN $\texttt{status} = \texttt{nonRed}$

END FOR

FOR $i = 1, ..., n$ $\mathbf{SEND}(\mathbf{SLAVE}_i, h, \texttt{update})$

END IF

END IF

END WHILE

Notice that the presented algorithm clearly preserves the strategy, since no modifications of G and S are possible before the first s-pair in the queue S is completely reduced. Of course, one may experiment with a variation of this algorithm that takes a whatever pair in the queue with its s-polynomial completely reduced to a nonzero and moves on with these modifications, however, our test runs show that in such case results are highly unpredictable and the output may heavily depend (in a random fashion) on the number of slaves, architecture of the used hardware, as well as random events in the system. Such an approach pays off very rarely and, when it does, it is extremely hard to reproduce the obtained good results consistently.

4.4 Experimental results

We have implemented the algorithm in C++ using MPI libraries for message passing. Though our implementation needs further optimization, at this point it is already competitive with the current computer algebra systems. Let us also point out that for now computations are done only for algebras over $\mathbb{Z}/p\mathbb{Z}$ for a large prime number p. All our experiments were conducted on a shared-memory SGI Origin 3800 supercomputer equipped with 500 MHz R14000 processors.

We tried to test our software for the "natural" input, i.e. problems that arose naturally in our research. Here are typical ideals that we computed Gröbner bases for:

- The elimination ideal in algebra $A = k\langle u, v, t, \partial_t, x, y, z, \partial_x, \partial_y, \partial_z\rangle$ with a monomial order eliminating commutative variables u and v that leads to the annihilator of f^s in $A_3[s]$ for

$f = xyz(x+y)(x+z)$ via Algorithm 9:

$$\begin{aligned} I_1 = A \cdot (\quad & -ux^3yz - ux^2y^2z - ux^2yz^2 - uxy^2z^2 + t, \\ & 3u\partial tx^2yz + 2u\partial txy^2z + 2u\partial txyz^2 + u\partial ty^2z^2 + \partial x, \\ & u\partial tx^3z + 2u\partial tx^2yz + u\partial tx^2z^2 + 2u\partial txyz^2 + \partial y, \\ & u\partial tx^3y + u\partial tx^2y^2 + 2u\partial tx^2yz + 2u\partial txy^2z + \partial z, \\ & uv - 1, 5t\partial t + x\partial x + y\partial y + z\partial z + 5 \quad) \end{aligned}$$

- The elimination ideal in the PBW-algebra $A = k\langle t, s, x, y, z, \partial_x, \partial_y, \partial_z \rangle$ with a product monomial order that eliminates t and then s, which leads to the same annihilator ideal via route laid out in [6]:

$$\begin{aligned} I_2 = (\quad & tx^3yz + tx^2y^2z + tx^2yz^2 + txy^2z^2 + s, \\ & 3tx^2yz + 2txy^2z + 2txyz^2 + ty^2z^2 + \partial x, \\ & tx^3z + 2tx^2yz + tx^2z^2 + 2txyz^2 + \partial y, \\ & tx^3y + tx^2y^2 + 2tx^2yz + 2txy^2z + \partial z \quad) \end{aligned}$$

To compute **BUCHBERGER**(I_1) it took approximately 1 minute with 1 slave and 11 seconds with 10 slaves. For **BUCHBERGER**(I_2) it was 15 and 2.5 seconds respectively, which provides a verification of Ucha-Castro results in [36] on the comparison of two different methods of computing the annihilator above.

For the commutative case we used such popular benchmarks as `cyclic6` (the ideal of $k[a,b,c,d,e,f]$ generated by polynomials $abcdef - 1, abcde + abcdf + abcef + abdef + acdef + bcdef, abcd + bcde + abcf + abef + adef + cdef, abc + bcd + cde + abf + aef + def, ab + bc + cd + de + af + ef, a + b + c + d + e + f$) and `cyclic7` (the ideal of $k[a,b,c,d,e,f,g]$ generated by polynomials $abcdefg - 1, abcdef + abcdeg + abcdfg + abcefg + abdefg + acdefg + bcdefg, abcde + bcdef + abcdg + abcfg + abefg + adefg + cdefg, abcd + bcde + cdef + abcg + abfg + aefg + defg, abc + bcd + cde + def + abg + afg + efg, ab + bc + cd + de + ef + ag + fg, a + b + c + d + e + f + g$) as well as the family of ideals of $k[x,y,z]$ generated by x^m, y^m, z^m and a random polynomial.

All the examples that we used as benchmarks produce between 100 and 2000 s-pairs during the computation, which is a relatively small number. However, our objective was simply to see how our implementation behaves for equally intense (in terms of the number of s-pairs)

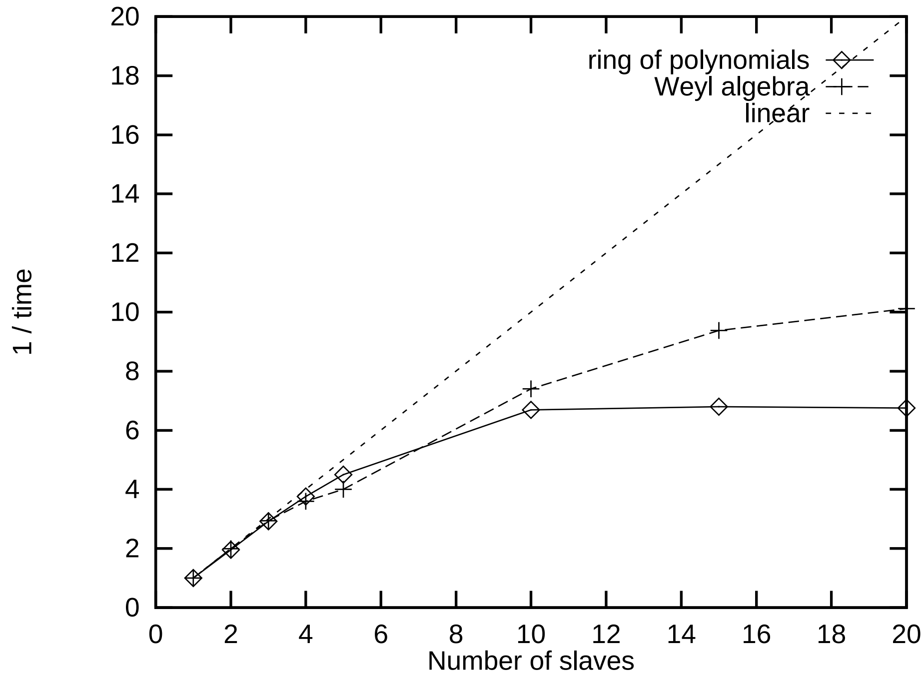

Figure 4.1: Speedups comparison for commutative and noncommutative setting

computations in commutative and noncommutative settings.

Figure 4.1 shows that computing with a small number of processors (slaves) results in good speedups in both cases. These fall behind what would be considered perfect – the linear speedup, though not by a lot.

As you see from the diagram using more than 5 slaves does not pay off much for both cases and using more than 10 slaves makes no sense in the commutative case, although there is still some progress observed if things do not commute.

Figure 4.2 provides an explanation why this behavior occurs: we look at the distribution of **REDUCE** requests sent to slaves. When a decision on where to send a request is made, the algorithm chooses the idling slave with the least number. Therefore, the last slave probabilistically has a chance for more rest than the first. But how bad is the distribution of the load? Figure 4.2 shows that whenever we cease to get a speedup close to linear (we have chosen 10 slaves to prove the point) it is quite far from uniform, however, the situation is slightly better in the noncommutative case.

Another factor that slows downs the algorithm is the communication overhead. This can be measured by the total number of times the **REDUCE** request is issued. For $n = 1$ this number is just the number of s-pairs reduced during the computation, however, for $n > 1$ by the time a

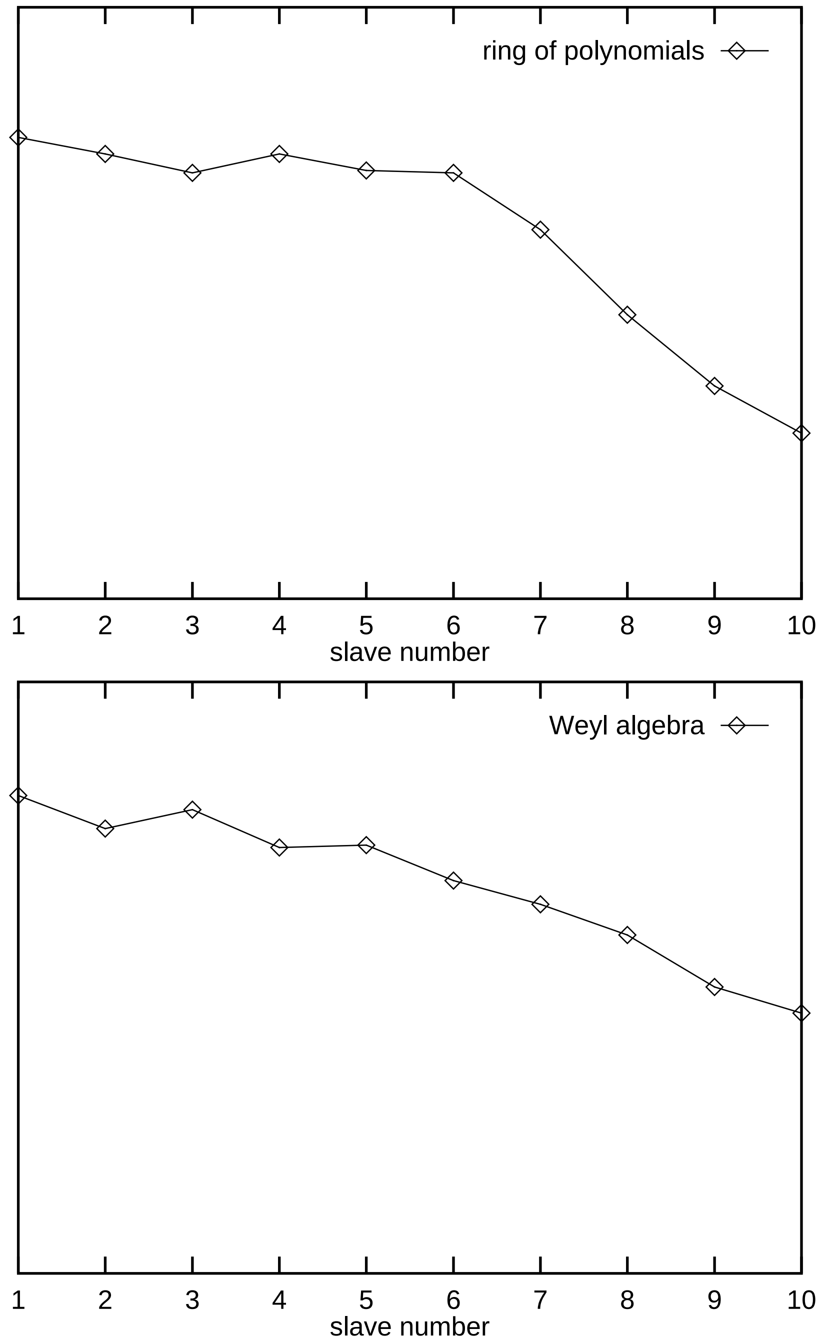

Figure 4.2: Load distribution for 10 slaves in commutative and noncommutative setting

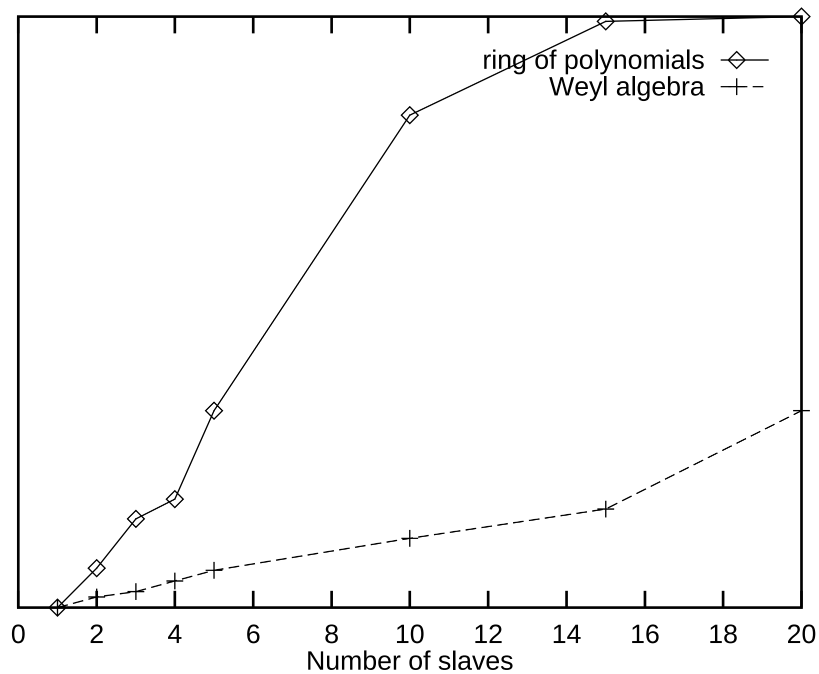

Figure 4.3: The number of times **REDUCE** task is sent

slave finishes its job and sends the reduced s-polynomial back to the master the master's copy of the reduction basis might grow, making a further reduction possible; then the master proceeds by sending an "extra" **REDUCE**. Usually, the number of **REDUCE**s grows with the number of slaves used, though this growth is noticeably steeper in the commutative setting as opposed to noncommutative (see Figure 4.3).

To summarize, we would like to say that, in general, the experimental results that we obtained confirm the results of previous research on this subject: linear (superlinear) speedups are not possible, at least using the strategy preserving approach, however, using small number of CPUs for parallel computing of Gröbner bases is quite efficient. This also have strengthened our believe that the payoff is even bigger if the same technique is used in the noncommutative setting due to better ratio of (communication overhead)/(time spent in reduction routine), plus due to the lack of elaborate s-pair selection techniques in the noncommutative case.

Bibliography

[1] V. Arnol'd. Normal forms of functions near degenerate critical points, the Weyl groups A_k, D_k, E_k and Lagrangian singularities. (Russian). *Funkcional. Anal. i Priložen.*, 6(4):3–25, 1972.

[2] V. Arnol'd. Normal forms of functions in the neighborhood of degenerate critical points. (Russian). *I. Uspehi Mat. Nauk*, 29(2(176)):11–49, 1974.

[3] G. Attardi and C. Traverso. Strategy-accurate parallel Buchberger algorithms. *J. Symbolic Comput.*, 21(4-6):411–425, 1996.

[4] I. Bernstein. Analytic continuation of generalized functions with respect to a parameter. *Functional Anal. Appl.*, 6:273–285, 1972.

[5] J.-E. Björk. *Rings of differential operators*, volume 21 of *North-Holland Mathematical Library.* North-Holland Publishing Co., Amsterdam-New York, 1979.

[6] J. Briançon and P. Maisonobe. Remarques sur l'ideal de Bernstein associé à des polynomes. Preprint.

[7] J. Briançon and P. Maisonobe. Idéaux de germes d'opérateurs différentiels à une variable. *Enseign. Math. (2)*, 30(1-2):7–38, 1984.

[8] J. Bruce. A stratification of the space of cubic surfaces. *Math. Proc. Camb. Phil. Soc.*, 87:427–441, 1980.

[9] S. Chakrabarti and K. Yelick. Distributed data structures and algorithms for Gröbner basis computation. *Lisp and Symbolic Computation*, 7:1–27, 1994.

[10] F. Chyzak. Mgfun Project.
pauillac.inria.fr/algo/chyzak/mgfun.html.

[11] J. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of ISSAC 2002*, pages 75–83, 2002.

[12] A. Galligo. Some algorithmic questions on ideals of differential operators. *Lecture Notes in Comput. Sci.*, 204:413–421, 1985.

[13] R. Gebauer and H. M. Möller. On an installation of Buchberger's algorithm. *J. Symbolic Comput*, 6(2-3):275–286, 1988.

[14] A. Giovini, T. Mora, G. Niesi, L. Robbiano, and C. Traverso. One sugar cube, please. In *Proc. ISSAC '91*, pages 49–54, 1991.

[15] D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[16] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001. `http://www.singular.uni-kl.de`.

[17] R. Hartshorne. *Algebraic geometry.* Number 52 in Graduate Texts in Mathematics. Springer-Verlag, New York-Heidelberg, 1977.

[18] A. Hillebrand and W. Schmale. Towards an effective version of a theorem of Stafford. *J. Symbolic Comput*, 32(6):699–716, 2001.

[19] A. Kandri-Rody and V. Weispfenning. Non-commutative Gröbner bases in algebras of solvable type. *J. Symbolic Computation*, 9:1–26, 1990.

[20] M. Kashiwara. B-functions and holonomic systems. Rationality of roots of B-functions. *Invent. Math.*, 38(1):33–53, 1976/77.

[21] H. Kredel and W. V. Parametric Gröbner bases for non-commutative polynomials. In *Proc. IV. International Conference on Computer Algebra in Physical Research, Joint Institute for Nuclear Research Dubna, USSR, May 1990*, pages 236–244. World Scientific, Singapore, 1991.

[22] A. Leykin. Constructibility of the set of polynomials with a fixed Bernstein-Sato polynomial: an algorithmic approach. *J. Symbolic Comput.*, 32(6):663–675, 2001.

[23] A. Leykin. D-modules for Macaulay 2. *Mathematical Software: ICMS 2002, World Scientific*, pages 169–179, 2002.

[24] A. Leykin and H. Tsai. D-modules for Macaulay 2. www.math.umn.edu/∼leykin/Dmodules.

[25] G. Lyubeznik. On Bernstein-Sato Polynomials. *Proc. of the AMS*, 125(7):1941–1944, 1997.

[26] B. Malgrange. Le polynôme de Bernstein d'une singularité isolée. (French). In *Fourier integral operators and partial differential equations (Colloq. Internat., Univ. Nice, Nice, 1974)*, volume 459 of *Lecture Notes in Math.*, pages 98–119. Springer, Berlin, 1975.

[27] Z. Mebkhout and L. Narváez-Macarro. La théorie du polynôme de Bernstein-Sato pour les algébres de Tate et de Dwork-Monsky-Washnitzer. *Ann. Sci. École Norm. Sup.*, (4) 24(2):227–256, 1991.

[28] T. Oaku. Algorithm for the b-function and D-modules associated with polynomial. *J. Pure Appl. Algebra*, 117/118:495–518, 1997.

[29] T. Oaku and N. Takayama. Algorithms for D-modules – restriction, tensor product, localization, and local cohomology groups. *J. Pure Appl. Algebra*, 156(2-3):267–308, 2001.

[30] B. Reinhard, G. Manfred, and W. Küchlin. A fine-grained parallel completion procedure. In *ISSAC 1994*, pages 269–277, 1994.

[31] M. Saito, B. Sturmfels, and N. Takayama. *Gröbner deformations of hypergeometric differential equations*, volume 6 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2000.

[32] K. Siegl. A Parallel Factorization Tree Gröbner Basis Algorithm. Technical Report 94-51, RISC-Linz, Johannes Kepler University, Linz, Austria, 1994. Published in Proceedings, PASCO'94, World Scientific Publ. Comp.

[33] J. T. Stafford. Module structure of Weyl algebras. *J. London Math. Soc. (2)*, 18(3):429–442, 1978.

[34] N. Takayama. `kan/sm1`: a computer algebra system for algebraic analysis. www.math.sci.kobe-u.ac.jp/KAN/.

[35] N. Takayama. Gröbner basis and the problem of contiguous relation. *Japan Journal of Appl. Math.*, 6:147–160, 1989.

[36] J. Ucha and F. Castro-Jiménez. Bernstein-Sato ideals associated to polynomials. *Preprint*, 2003.

[37] U. Walther. Algorithmic computation of local cohomology modules and the local cohomological dimension of algebraic varieties. *J. Pure Appl. Algebra*, 139(1-3):303–321, 1999.

[38] V. Weispfenning. Comprehensive Gröbner bases. *J. Symbolic Comput.*, 14:1–29, 1992.

[39] T. Yano. On the theory of b-functions. *Publ. Res. Inst. Math. Sci.*, 14(1):111–202, 1978.