Effective Methods for the Computation of Bernstein-Sato polynomials for Hypersurfaces and Affine Varieties

Daniel Andres\textsuperscript{a}, Viktor Levandovskyy\textsuperscript{a}, Jorge Martín-Morales\textsuperscript{b}

\textsuperscript{a}Lehrstuhl D für Mathematik, RWTH Aachen, Templergraben 64, 52062 Aachen, Germany
\textsuperscript{b}Department of Mathematics-I.U.M.A., University of Zaragoza, C/ Pedro Cerbuna, 12 - 50009, Zaragoza, Spain

Abstract

This paper is the widely extended version of the publication, appeared in Proceedings of ISSAC’2009 conference (Andres, Levandovskyy, and Martín-Morales, 2009). We discuss more details on proofs, present new algorithms and examples. We present a general algorithm for computing an intersection of a left ideal of an associative algebra over a field with a subalgebra, generated by a single element. We show applications of this algorithm in different algebraic situations and describe our implementation in SINGULAR. Among other, we use this algorithm in computational $D$-module theory for computing e. g. the Bernstein-Sato polynomial of a single polynomial with several approaches. We also present a new method, having no analogues yet, for the computation of the Bernstein-Sato polynomial of an affine variety. Also, we provide a new proof of the algorithm by Briançon-Maisonobe for the computation of the $s$-parametric annihilator of a polynomial. Moreover, we present new methods for the latter computation as well as optimized algorithms for the computation of Bernstein-Sato polynomial in various settings.

1. Introduction

This paper extends Andres, Levandovskyy, and Martín-Morales (2009) by many details, proofs, algorithms and examples. In this paper we continue reporting (Levandovskyy, 2006; Levandovskyy and Martín-Morales, 2008; Andres et al., 2009) on our advances in constructive $D$-module theory both in theoretical direction and also in the implementation, which we create in SINGULAR.

Our work on the implementation of procedures for $D$-modules started in 2003, motivated among other factors by challenging elimination problems in non-commutative

Email addresses: Daniel.Andres@math.rwth-aachen.de (Daniel Andres), Viktor.Levandovskyy@math.rwth-aachen.de (Viktor Levandovskyy), jorge@unizar.es (Jorge Martín-Morales).
algebras, which appear e. g. in algorithms for computation of Bernstein-Sato polynomials. We reported on solving several challenges in Levandovskyy and Martín-Morales (2008). A non-commutative subsystem SINGULAR:PLURAL (Greuel et al., 2006) of the computer algebra system SINGULAR provides a user with possibilities to compute numerous Gröbner bases-based procedures in a wide class of non-commutative G-algebras (Levandovskyy and Schönenmann, 2003). It is natural to use this functionality in the context of computational D-module theory.

As of today, the D-module suite in SINGULAR consists of three libraries: dmod.lib, dmodapp.lib and bfun.lib. Moreover, gmsing.lib (Schulze, 2004) contains some sophisticated (and hence fast) and useful procedures, e. g. bernstein for the computation of the local Bernstein-Sato polynomial of an isolated singularity at the origin. There are many useful and flexible procedures for various aspects of D-module theory. These libraries are freely distributed together with SINGULAR (Greuel et al., 2005) since the version 3-1-0, which was released in April 2009. More libraries are currently under development, among them procedures for computing the restriction, integration and localization of D-modules.

There are several implementations of algorithms for D-modules, namely the experimental program kan/sm1 by N. Takayama (Takayama, 2003), the bfct package in RISA/ASIR (Noro et al., 2006) by M. Noro (Noro, 2002) and the package Dmodules.m2 in MACAULAY2 by A. Leykin and H. Tsai (Tsai and Leykin, 2006). To the best of our knowledge, there is ongoing work by the CoCoA Team (2009) to develop some D-module functionality as well. We aim at creating a D-module suite, which will combine flexibility and rich functionality with high performance, being able to treat more complicated examples.

We continue comparing our implementation (cf. Section 7.1) with the ones in the systems ASIR and MACAULAY2, see Levandovskyy and Martín-Morales (2008) for earlier results.

In this paper, we address the following computational problems:

- s-parametric annihilator of $f \in \mathbb{K}[x_1, \ldots, x_n]$
- Bernstein-Sato ideals for $f = f_1 \cdots f_m$
- $b$-function with respect to weights for an ideal in $D$
- Global Bernstein-Sato polynomial of $f$
- Bernstein-Sato polynomial for a variety

In Section 3.3, we give a new proof for the algorithm by Briançon-Maisonobe for computing $\Ann_{D[s]} f^s$, announced in Levandovskyy and Martín-Morales (2008). Moreover, using the same technique we design a new algorithm for the computation of the Bernstein-Sato polynomial of an affine variety, following the paper Budur et al. (2006), and prove its correctness.

We develop the method of principal intersection 4.11 in the general context of $\mathbb{K}$-algebras and discuss its improvements. This algorithm is especially useful for problems of D-module theory, since it allows to replace a generally hard elimination with Gröbner bases by the search for a $\mathbb{K}$-linear dependence of a sequence of normal forms. The algorithm is applied in Section 5 to two main methods for computing Bernstein-Sato polynomials as well as to solving 0-dimensional systems in commutative rings and to the computation of central characters in Section 4.2.1. Moreover, we describe a folklore method for computing Bernstein-Sato polynomial via annihilator (using, however, principal intersection instead of Gröbner-based elimination) and prove in Lemma 6.4, that is more efficient than the usual one.
The generalization of principal intersection approach to the case of more general sub-algebras we discuss in Section 4.3.

1.1. Notations

Throughout the article $\mathbb{K}$ stands for a field of characteristic zero. By $R$ we denote the polynomial ring $\mathbb{K}[x_1,\ldots,x_n]$ and by $f \in R$ a non-constant polynomial.

We consider the $n$-th Weyl algebra as the algebra of linear partial differential operators with polynomials coefficients. That is $D_n = D_n(R) = \mathbb{K}(x_1,\ldots,x_n,\partial_1,\ldots,\partial_n \mid \{\partial_x i = x_i\partial_i + 1,\partial_x j = x_j\partial_j, i \neq j\})$. We denote by $D_n[s] = D_n(R) \otimes_{\mathbb{K}} \mathbb{K}[s_1,\ldots,s_n]$ and drop the index $n$ depending on the context.

The ring $R$ is a natural $D_n(R)$-module with the action

$$x_i \cdot f(x) = x_i \cdot f(x), \partial_i \cdot f(x) = \frac{\partial f(x)}{\partial x_i}.$$

Working with monomial orderings in elimination, we use the notation $x \gg y$ for “$x$ is greater than any power of $y$”.

Given an associative $\mathbb{K}$-algebra $A$ and some monomial well-ordering on $A$, we denote by $\text{lm}(f)$ (resp. $\text{lc}(f)$) the leading monomial (resp. the leading coefficient) of $f \in A$. Given a left Gröbner basis $G \subset A$ and $f \in A$, we denote by $\text{NF}(f,G)$ the normal form of $f$ with respect to the left ideal $\langle G \rangle$. We also use the shorthand notation $h \rightarrow_H f$ (and $h \rightarrow f$, if $H$ is clear from the context) for the reduction of $h \in A$ to $f \in A$ with respect to the set $H$. If not specified, under ideal we mean left ideal in a $\mathbb{K}$-algebra, and by Gröbner basis a left Gröbner basis. For $a,b$ in some $\mathbb{K}$-algebra $A$, we use the Lie bracket notation $[a,b] := ab - ba$ as well as skew Lie bracket notation $[a,b]_k := ab - k \cdot ba$ for $k \in \mathbb{K}^*$.

We say, that a proper subalgebra $S$ of an associative $\mathbb{K}$-algebra $A$ is a principal subalgebra, if there exists $g \in A \setminus \mathbb{K}$, such that $S = \mathbb{K}[g]$.

Let $M$ be an $A$-module, then we denote the Gel’fand-Kirillov dimension of $M$ (see McConnell and Robson (2001) for the details and Bueso et al. (2003) for algorithms) by $\text{GK.dim}(M)$. Recall, that a module $M$ is called holonomic, if $\text{GK.dim}(A/\operatorname{Ann} M) = 2 \cdot \text{GK.dim}(M)$. We prefer this general definition, since it concides with the classical way of defining holonomy in Weyl algebras and it is incomparably more general to the latter. In particular, armed with this general definition we can speak on holonomic modules over any $G$-algebra.

2. Preliminaries

It is convenient to treat the algebras we deal with in the bigger framework of $G$-algebras of Lie type.

Definition 2.1. Let $A$ be the quotient of the free associative algebra $\mathbb{K}(x_1,\ldots,x_n)$ by the two-sided ideal $I$, generated by the finite set $\{x_ix_i - x_i x_j - d_{ij} \mid 1 \leq i < j \leq n\}$, where $d_{ij} \in \mathbb{K}[x_1,\ldots,x_n]$. The algebra $A$ is called a $G$-algebra of Lie type (Levandovskyy and Schönemann, 2003), if

- $\forall 1 \leq i < j < k \leq n$ the expression $d_{ij}x_k - x_kd_{ij} + x_jd_{ik} - d_{ik}x_j + d_{jk}x_i - x_id_{jk}$ reduces to zero modulo $I$ and
- there exists a monomial ordering $\prec$ on $\mathbb{K}[x_1,\ldots,x_n]$, such that $\text{lm}(d_{ij}) \prec x_ix_j$ for each $i < j$.
These algebras were also studied in Kandri-Rody and Weispfenning (1990) and Bueso et al. (1998) by the names PBW algebras and algebras of solvable type.

Recall the algorithm for computing the preimage of a left ideal under a homomorphism of $G$-algebras from Levandovskyy (2006).

**Theorem 2.2** (Preimage of a Left Ideal, Levandovskyy (2006)). Let $A, B$ be $G$-algebras of Lie type, generated by $\{x_i \mid 1 \leq i \leq n\}$ and $\{y_j \mid 1 \leq j \leq m\}$ respectively, subject to finite sets of relations $R_A, R_B$ as in Def. 2.1. Let $\phi : A \rightarrow B$ be a homomorphism of $\mathbb{K}$-algebras. Define $I_\phi$ to be the $(A, A)$-bimodule $A(\{x_i - \phi(x_i) \mid 1 \leq i \leq n\})_A \subset A \otimes_{\mathbb{K}} B$. Suppose, that there exists an elimination ordering for $B$ on $A \otimes_{\mathbb{K}} B$, satisfying the following conditions

$$1 \leq i \leq n, 1 \leq j \leq m, \quad \text{lm}(y_j \phi(x_i) - \phi(x_i)y_j) \prec x_i y_j.$$

Then there are the following statements.

1) Define $A \otimes_{\mathbb{K}}^\phi B$ to be the $\mathbb{K}$-algebra generated by $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ subject to the finite set of relations composed of $R_A, R_B$ and $\{y_j x_i - x_i y_j - y_j \phi(x_i) + \phi(x_i)y_j\}$. Then $A \otimes_{\mathbb{K}}^\phi B$ is a $G$-algebra of Lie type.

2) Let $J \subset B$ be a left ideal, then

$$\phi^{-1}(J) = (I_\phi + J) \cap A \subset A \otimes_{\mathbb{K}}^\phi B \cap A.$$ Moreover, this computation can be done by means of elimination.

The following proposition is a reformulation of Theorem 2.2, adopted to the situation, which is often encountered in context of $D$-modules.

**Proposition 2.3.** Let $A_1, B_1, C$ be $G$-algebras of Lie type and $\varphi : A_1 \rightarrow B_1$ be a homomorphism of $\mathbb{K}$-algebras. Consider the following data:

$$A = C \otimes_{\mathbb{K}} A_1, \quad B = C \otimes_{\mathbb{K}} B_1, \quad \phi = 1_C \otimes \varphi : A \rightarrow B,$$

$$E = A \otimes_{\mathbb{K}}^\phi B, \quad E' = C \otimes_{\mathbb{K}} (A_1 \otimes_{\mathbb{K}}^\varphi B_1).$$

Then $A \subset E' \subset E$ and for a left ideal $J \subset B$ we have:

1) $(EI_{\varphi} + EJ) \cap E' = E'I_{\varphi} + E'J$.

2) $\phi^{-1}(J) = (E'I_{\varphi} + E'J) \cap A$.

Moreover, the second intersection can be computed using Gröbner bases, provided there exists an elimination ordering for $B_1$ on $E'$ compatible with the $G$-algebra structure of $E'$.

In the proofs we quite often use the following.

**Lemma 2.4** (Generalized Product Criterion, Levandovskyy and Schöñemann (2003)). Let $A$ be a $G$-algebra of Lie type and $f, g \in A$. Suppose that $\text{lm}(f)$ and $\text{lm}(g)$ have no common factors, then $\text{spoly}(f, g)$ reduces to $[f, g]$ with respect to the set $\{f, g\}$.

3. $s$-parametric Annihilator of $f$

Recall Malgrange’s construction for $f = f_1 \cdot \ldots \cdot f_p \in R = \mathbb{K}[x_1, \ldots, x_n]$. Let $D_n = D_n(R), T = \mathbb{K}[t_1, \ldots, t_p]$ and $D'_p := D_p(T)$. Consider the $(p+n)$-th Weyl algebra $D_{p+n} = \ldots$
$D_n \otimes_K D'_p$. Moreover, consider the following left ideal in $D_{p+n}$, called the Malgrange ideal

$$I_f := \{ \{ t_j - f_j, \partial_i + \sum_{j=1}^{p} \frac{\partial f_j}{\partial x_i} \partial t_j, 1 \leq j \leq p, 1 \leq i \leq n \} \}.$$

Then for $s = (s_1, \ldots, s_p)$ we denote $f^s := f_1^{s_1} \cdots f_p^{s_p}$. Let us compute

$$I_f \cap K[(t_j \partial t_j)] \{ x_i, \partial x_i \mid [\partial x_i, x_i] = 1 \} \subset D_n\{ \{ t_j \partial t_j \} \} \subset D_{p+n}$$

and furthermore, replace $t_j \partial t_j$ with $-s_j - 1$. The result is known (e.g. Saito et al. (2000)) to be exactly $\text{Ann}_{D[s]} f^s \subset D[s]$.

There exist several methods for the computation of the $s$-parametric annihilator of $f^s$.

3.1. Oaku and Takayama

The algorithm of Oaku and Takayama (Oaku, 1997a,b,c; Saito et al., 2000) was developed in a wider context and uses homogenization. With notations as above, let $H := D_n \otimes_K D'_p \otimes_K K[u_1, \ldots, u_p, v_1, \ldots, v_p]$. Moreover, let $I$ below be the $(u,v)$-homogenized Malgrange ideal, that is the left ideal in $H$

$$I = \left( \{ t_j - u_j f_j, \partial_i + \sum_{k=1}^{p} \frac{\partial f_k}{\partial x_i} u_k \partial t_j, u_j v_j - 1 \} \right).$$

Oaku and Takayama proved, that $\text{Ann}_{D_n[s]}(f^s)$ can be obtained in two steps. At first the $\{ u_j, v_j \}$ are eliminated from $I$ with the help of Gröbner bases, thus yielding $I' = I \cap (D_n \otimes_K K)$.

3.2. Briançon and Maisonobe

Consider $S_p = K[(\partial t_j, s_j) \mid \partial t_j s_k = s_k \partial t_j - \delta_j k \partial t_j]$ (the $p$-th shift algebra) and $B = D_n \otimes_K S_p$. Moreover, consider the following left ideal in $B:

$$I = \left( \{ s_j + f_j \partial t_j, \partial_i + \sum_{k=1}^{p} \frac{\partial f_k}{\partial x_i} \partial t_k \} \right).$$

Briançon and Maisonobe proved in Briancon and Maisonobe (2002) that $\text{Ann}_{D_n[s]}(f^s) = I \cap D_n[s]$ and hence the latter can be computed via the left Gröbner basis with respect to an elimination ordering for $\{ \partial t_j \}$.

3.3. A new proof for Briançon-Maisonobe

By using the Preimage Theorem 2.2 we give a new, completely computer-algebraic proof for the method of Briançon-Maisonobe 3.2.

Let $A := D_n[s] = K((s_j, x_i, \partial_i) \mid \partial x_i = x_i \partial_i + 1)$, and $B := K((t_j, \partial t_j, x_i, \partial_i) \mid \partial x_i = x_i \partial_i + 1, \partial t_j t_j = t_j \partial t_j + 1)$.

Thus in the notations of Proposition 2.3, $C = D_n, A_1 = K[s], B_1 = K((t_j, \partial t_j) \mid \partial t_j t_j = t_j \partial t_j + 1)$ and $A = C \otimes A_1, B = C \otimes B_1$. Consider the algebraic Mellin transform (cf. Saito et al. (2000)) $\phi : A_1 \to B_1, \ s_j \mapsto -t_j \partial t_j - 1$. 

5
Hence $I_\varphi = \langle \{ t_j \partial t_j + s_j + 1 \} \rangle \subset A \otimes_B B_1 =: E'$. Since $[t_k, s_j] = \delta_{jk} t_j$ and $[\partial t_k, s_j] = -\delta_{jk} \partial t_j$, the ordering conditions of Theorem 2.2 take the form $t_j \prec s_j t_j, \partial t_j \prec s_j \partial t_j$, which are satisfied if and only if $1 \leq t_j, \partial t_j, s_j$.

By Proposition 2.3, for any $L \subset B$, $\phi^{-1}(L) = (I_\varphi + L) \cap A$. Hence,

$$I_\varphi + L = \langle \{ t_j - f_j, \partial_i + \sum_{j=1}^p \frac{\partial f_j}{\partial x_i} \partial t_j, t_j \partial t_j + s_j + 1 \} \rangle$$

$$= \langle \{ t_j - f_j, \partial_i + \sum_{j=1}^p \frac{\partial f_j}{\partial x_i} \partial t_j, f_j \partial t_j + s_j \} \rangle$$

because $t_j \partial t_j + s_j + 1$ reduces to

$$t_j \partial t_j + s_j + 1 - \partial t_j \cdot (t_j - f_j) = f_j \partial t_j + s_j \in I_\varphi + L.$$

**Lemma 3.1.** Consider an ordering $\prec_T$, which satisfies the property $\{ t_j \} \gg \{ x_i \}$, $\{ \partial_i, s_j \} \gg \{ x_i, \partial t_j \}$. Moreover, set

$$g_i := \partial_i + \sum_{k=1}^p \frac{\partial f_k}{\partial x_i} \partial t_k, \ S_1 := \{ t_j - f_j, g_i \}, \ S_2 := S_1 \cup \{ s_j + f_j \partial t_j \} \subset E'.$$

Then $S_1$ and $S_2$ are left Gröbner bases with respect to $\prec_T$.

**Proof.** We run Buchberger’s algorithm by hand. There are only three kinds of critical pairs we have to consider. Due to the ordering property, for each pair the generalized Product Criterion is applicable. Hence, we need to compute just the Lie brackets of members of pairs.

1. $[t_i - f_i, t_k - f_k] = 0$.
2. For pairs $(g_i, g_k)$ computing $[g_i, g_k]$ yields

$$\sum_j \partial t_j [\partial_i, \frac{\partial f_j}{\partial x_k}] + \sum_j \partial t_j [\frac{\partial f_j}{\partial x_i}, \partial_k] = \sum_j \partial t_j ([\partial_i, \frac{\partial f_j}{\partial x_k}] - [\partial_k, \frac{\partial f_j}{\partial x_i}]).$$

Since $[\partial_i, \frac{\partial f_j}{\partial x_k}] = \frac{\partial^2 f_j}{\partial x_i \partial x_k}$, $[\partial_k, \frac{\partial f_j}{\partial x_i}]$, spoly$(g_i, g_k)$ reduces to zero.
3. For mixed pairs $(t_k - f_k, g_i)$ we have

$$[t_k - f_k, g_i] = \sum_j \frac{\partial f_j}{\partial x_i} [t_k, \partial t_j] - [f_k, \partial_i] = 0.$$

Hence $S_1$ is a left Gröbner basis. Now, in $S_2$ there are three new kinds of critical pairs to consider and for all of them we can apply the generalized Product Criterion.

4. $[t_k - f_k, s_j + f_j \partial t_j] = [t_k, s_j] + [f_j [t_k, \partial t_j] - [f_k, s_j] - [f_k, f_j \partial t_j] = \delta_{jk} (t_k - f_j) \rightarrow 0$.
5. $[s_i + f_i \partial t_i, s_j + f_j \partial t_j] = f_j [s_i, \partial t_j] - f_i [s_j, \partial t_i] = 0.$
6. Finally,

\[ [s_j + f_j \partial t_j, \partial_i + \sum_{k=1}^{p} \frac{\partial f_k}{\partial x_i} \partial t_k] \]

\[ = [s_j, \partial_i] + \partial t_j [f_j, \partial_i] + \sum_{k=1}^{p} \frac{\partial f_k}{\partial x_i} [s_j, \partial t_k] + [f_j \partial t_j, \sum_{k=1}^{p} \frac{\partial f_k}{\partial x_i} \partial t_k] \]

\[ = \frac{\partial f_j}{\partial x_i} \partial t_j - [\partial_i, f_j] \partial t_j = 0. \]

So, \( S_2 \) is a left Gröbner basis.

We want to eliminate both \( \{ t_j \} \) and \( \{ \partial t_j \} \) from \( I_\phi + L \). As we see above, by using an elimination ordering for \( \{ t_j \} \) we proved above that \( S_2 \) is a Gröbner basis. So, the elimination ideal is generated by \( S_3 := S_2 \setminus \{ t_i \} \). Hence we can proceed with eliminating \( \{ \partial t_j \} \) from \( S_3 \), which is exactly the statement of the Briançon-Maisonobe algorithm in Section 3.2.

3.4. Bernstein-Sato ideals for \( f = f_1 \cdot \ldots \cdot f_m \)

Comparing the effectiveness of the algorithms, Gago-Vargas et al. (2005) concluded that the method of Briançon-Maisonobe is the best for the computation of \( s \)-parametric annihilators. In Levandovskyy and Martín-Morales (2008) we gave experimental results for the case \( f = f_1 \) and showed, that the algorithm of Briançon-Maisonobe is faster than the LOT method, which in turn is faster than the algorithm of Oaku and Takayama.

Because of the structure of annihilators in the situation \( f = f_1 \cdot \ldots \cdot f_p \), \( p > 1 \), basically the same principles stand behind the corresponding algorithms.

Let \( s = (s_1, \ldots, s_p) \), then a Bernstein-Sato ideal in \( K[s] \), which is defined as

\[ B(f) = (\text{Ann}_{D[s_1, \ldots, s_p]} f_1^{s_1} \cdot \ldots \cdot f_p^{s_p} + \langle f_1 \cdot \ldots \cdot f_p \rangle) \cap K[s_1, \ldots, s_p], \]

can be computed with the help of \( \text{Ann}_{D[s]} f^s \subset D[s] \). See Bahloul (2001) for algorithms. In contrary to the case \( f = f_1 \), the ideal \( B(f) \) need not be principal in general. However, it is an open question to give a criterion for the principality of \( B(f) \). Armed with such a criterion, one can apply the method of Principal Intersection 4.11 and thus replace expensive elimination above by the computation of a minimal polynomial. As in the case \( f = f_1 \) it is an open question, which strategy and which orderings should one use in the computation of the annihilator in order to achieve better performance.

3.5. Implementation

Due to the comparison above, we decided to implement only Briançon-Maisonobe method for the \( (s_1, \ldots, s_p) \)-parametric annihilator \( \text{Ann}_{D[s]} f^s \subset D[s] \) in the case of \( p > 1 \).

The corresponding procedure of \texttt{dmod.lib} is called \texttt{annfsBMI}. It computes both annihilator and the Bernstein-Sato ideal.

We reported in Levandovskyy and Martín-Morales (2008) on several computational challenges, which have been solved with the help of our implementation.

We use the following acronyms in addressing functions in the implementation: \texttt{OT} for Oaku and Takayama, \texttt{LOT} for Levandovskyy’s modification of Oaku and Takayama (Levandovskyy and Martín-Morales, 2008) and \texttt{BM} for Briançon-Maisonobe. Moreover,
it is possible to specify the desired Gröbner basis engine (std or slimgb) via an optional argument.

For the classical situation where \( f = f_1 \), there are \texttt{SannfsOT}, \texttt{SannfsLOT}, \texttt{SannfsBM} procedures implemented, each along the lines of the corresponding algorithm. Moreover, there is a procedure \texttt{Sannfs}(f), computing \( \text{Ann}_{D[s]} f^s \subset D[s] \) using a “minimal user knowledge” principle.

**Example 3.2.** We demonstrate, how to compute the \( s \)-parametric annihilator with \texttt{Sannfs}. This procedure takes a polynomial in a commutative ring as its argument and returns back a Weyl algebra of the type \texttt{ring} together with an object of the type \texttt{ideal} called LD.

LIB "dmod.lib";
ring r = 0,(x,y),dp; // set up commutative ring
poly f = x^3 + y^2 + x*y^2; // define polynomial
def D = Sannfs(f); // call Sannfs
setring D; LD; // activate ring D, print \( \text{Ann}(f^s) \)

3. \( x \)-fuinction with respect to degrees for an ideal

Let \( 0 \neq w \in \mathbb{R}^n \geq 0 \) and consider the \( V \)-filtration \( V = \{ V_m \mid m \in \mathbb{Z} \} \) on \( D \) with respect to \( w \), where \( V_m \) is spanned by \( \{ x^\alpha \partial^\beta \mid -w\alpha + w\beta \leq m \} \) over \( K \). That is, \( x_i \) and \( \partial_i \) get weights \( -w_i \) and \( w_i \) respectively. Note, that with respect to such weights the relation \( \partial_i x_i = x_i \partial_i + 1 \) is homogeneous of degree 0. It is known that the associated graded ring \( \bigoplus_{m \in \mathbb{Z}} V_m / V_{m-1} \) is isomorphic to \( D \), which allows us to identify them.

From now on we assume, that \( I \) is an ideal such that \( D/I \) is a holonomic module. Since holonomic \( D \)-modules are cyclic (e. g. Coutinho (1995)), for each holonomic \( D \)-module \( M \) there exists an ideal \( I_M \) such that \( M \cong D/I_M \) as \( D \)-modules.

**Definition 4.1.** Let \( 0 \neq w \in \mathbb{R}^n_{\geq 0} \). For non-zero

\[
p = \sum_{\alpha,\beta \in \mathbb{N}^n_0} c_{\alpha,\beta} x^\alpha \partial^\beta \in D
\]

we put \( m = \max_{\alpha,\beta} \{ -w\alpha + w\beta \mid c_{\alpha,\beta} \neq 0 \} \in \mathbb{R} \) and define the initial form of \( p \) with respect to the weight \( w \) as follows:

\[
\text{in}_{(-w,w)}(p) := \sum_{\alpha,\beta \in \mathbb{N}^n_0: -w\alpha + w\beta = m} c_{\alpha,\beta} x^\alpha \partial^\beta.
\]

For the zero polynomial, we set \( \text{in}_{(-w,w)}(0) := 0 \). Additionally, we call the graded ideal \( \text{in}_{(-w,w)}(I) := K \cdot \{ \text{in}_{(-w,w)}(p) \mid p \in I \} \) the initial ideal of \( I \) with respect to the weight \( w \).
Definition 4.2. Let $0 \neq w \in \mathbb{R}^n_{\geq 0}$ and $s := \sum_{i=1}^{n} w_i x_i \partial_i$. Then in$_{(w,w)}(I) \cap \mathbb{K}[s]$ is a principal ideal in $\mathbb{K}[s]$. Its monic generator $b(s)$ is called the global $b$-function of $I$ with respect to the weight $w$.

Theorem 4.3. The global $b$-function of $I$ is nonzero.

We will give yet another proof of this theorem (Saito et al., 2000) in Section 4.2.

Note, that by setting the weight vector in an appropriate way, one can compute $b$-functions of holonomic $D$-modules $D/I$, which are usually referred as $b$-function for restriction, integration, localization etc. These special $b$-functions play an important role in the computation of the corresponding restriction, integration, localization modules, see Oaku (1997c); Saito et al. (2000).

Following its definition, the computation of the global $b$-function of $I$ with respect to $w$ can be done in two steps:
1. Compute the initial ideal $I'$ of $I$ with respect to $w$.
2. Compute the intersection of $I'$ with the subalgebra $\mathbb{K}[s]$.

We will discuss both steps separately, starting with the initial ideal.

4.1. Computing the initial ideal

In order to compute the initial ideal, the method of weighted homogenization has been proposed in Noro (2002). A more general approach on homogenization of differential operators can be found in Castro-Jiménez and Narváez-Macarro (1997).

Let $u, v \in \mathbb{R}^n_{>0}$. The associative $\mathbb{K}$-algebra $D^{(h)}_{(u,v)}$ is a $G$-algebra in the variables $x_1, \ldots, x_n, \partial_1, \ldots, \partial_n, h$ which commute pairwise except for $\partial_j x_i = x_i \partial_j + \delta_{ij} h^{u_i+v_j}$. $D^{(h)}_{(u,v)}$ is called the $n$-th weighted homogenized Weyl algebra with weights $u, v$, i.e. $x_i$ and $\partial_i$ get weights $u_i$ and $v_i$ respectively.

For $p = \sum_{\alpha, \beta} \alpha \beta x^\alpha \partial^\beta \in D$ one defines the weighted homogenization of $p$ as follows:

$$H_{(u,v)}(p) = \sum_{\alpha, \beta} \alpha \beta h^{\deg_{(u,v)}(p) - (u \alpha + v \beta)} x^\alpha \partial^\beta.$$ 

This definition naturally extends to a set of polynomials. Here, $\deg_{(u,v)}(p)$ denotes the weighted total degree of $p$ with respect to weights $u, v$ for $x, \partial$ and weight 1 for $h$.

For a monomial ordering $\prec$ in $D$, which is not necessarily a well-ordering, we define an associated homogenized global ordering $\prec^{(h)}$ in $D^{(h)}_{(u,v)}$ by setting $h \prec^{(h)} x_i, h \prec^{(h)} \partial_i$ for all $i$ and,

$$p \prec^{(h)} q \text{ if } \deg_{(u,v)}(p) < \deg_{(u,v)}(q)$$

$$\text{or } \deg_{(u,v)}(p) = \deg_{(u,v)}(q) \text{ and } p|_{h=1} \prec q|_{h=1}.$$ 

Note that for $u = v = (1, \ldots, 1)$ this is exactly the standard homogenization as in Saito et al. (2000). Analogue statements of the following two theorems can be found in Saito et al. (2000) and Noro (2002) respectively. Due to our different conception of Gröbner bases (we require well-orderings), we give new proofs for them.

Theorem 4.4. Let $F$ be a finite subset of $D$ and $\prec$ a global ordering. If $G^{(h)}$ is a Gröbner basis of $(H_{(u,v)}(F))$ with respect to $\prec^{(h)}$, then $G^{(h)}|_{h=1}$ is a Gröbner basis of $(F)$ with respect to $\prec$. 

9
Proof. For any \( f \in (F) \) with \( \text{lm}(H_{(u,v)}(f)) = h^\lambda x^\alpha \partial^\beta \), there exists \( g^{(h)} \in G^{(h)} \) with \( \text{lm}(g^{(h)}) = h^\lambda x^\gamma \partial^\delta \) satisfying \( \text{lm}(g^{(h)}) | \text{lm}(f) \). Then \( \text{lm}(g^{(h)})_{|h=1} = x^\gamma \partial^\delta \ | \ x^\alpha \partial^\beta = \text{lm}(f)_{|h=1} \), which proves the claim. \( \square \)

**Theorem 4.5.** Let \( \prec \) be a global monomial ordering on \( D \) and \( \prec_{(-w,w)} \) the non-global ordering defined by

\[
  x^\alpha \partial^\beta \prec_{(-w,w)} x^\gamma \partial^\delta \quad \text{if} \quad -w\alpha + w\beta < -w\gamma + w\delta
\]

or \( -w\alpha + w\beta = -w\gamma + w\delta \) and \( x^\alpha \partial^\beta \prec x^\gamma \partial^\delta \).

If \( G^{(h)} \) is a Gröbner basis of \( H_{(u,v)}(I) \) with respect to \( \prec_{(-w,w)} \), then \( \{ \text{in}_{(-w,w,0)}(g) | g \in G^{(h)} \} \) is a Gröbner basis of \( \text{in}_{(-w,w,0)}(H_{(u,v)}(I)) \) with respect to \( \prec^{(h)} \).

**Proof.** Let \( f' \in \text{in}_{(-w,w,0)}(H_{(u,v)}(I)) \) be \((-w, w, 0)\)-homogeneous. There exist elements \( f \in H_{(u,v)}(I), g \in G^{(h)} \) such that \( f' = \text{in}_{(-w,w,0)}(f) \) and \( \text{lm}_{\prec_{(-w,w)}}(g) | \text{lm}_{\prec_{(-w,w)}}(f) \).

Since \( f, g \) are \((u, v)\)-homogeneous, we have

\[
  \text{lm}_{\prec_{(-w,w)}}(p) = \text{lm}_{\prec_{(-w,w)}}(\text{in}_{(-w,w,0)}(p)) \quad \text{for} \quad p \in \{ f, g \},
\]

which finishes the proof. \( \square \)

Summarizing the results from this section, we obtain the following algorithm to compute the initial ideal.

**Algorithm 4.6 (InitialIdeal).**

**Input:** \( I \subset D \) such that \( D/I \) is holonomic, \( \prec \) a global ordering on \( D \), \( 0 \neq w \in \mathbb{R}^n_0 \), \( u, v \in \mathbb{R}^n_0 \).  

**Output:** A Gröbner basis \( G \) of \( \text{in}_{(-w,w)}(I) \) with respect to \( \prec \) 
\(-w,w)\:= the homogenized ordering as in Theorem 4.5  
\( G^{(h)} \):= a Gröbner basis of \( H_{(u,v)}(I) \) with respect to \( \prec^{(h)} \)  
**return** \( G = \text{in}_{(-w,w)}(G^{(h)}_{|h=1}) \)

**4.2. Intersecting an ideal with a principal subalgebra**

We will now consider a much more general setting than needed to compute the global \( b \)-function. Let \( A \) be an associative \( \mathbb{K} \)-algebra. We are interested in computing the intersection of a left ideal \( J \subset A \) with the subalgebra \( \mathbb{K}[s] \) of \( A \) where \( s \in A \setminus \mathbb{K} \). We would like to find the monic polynomial \( b \in A \) such that

\[
  \langle b \rangle = J \cap \mathbb{K}[s].
\]

For this section, we will assume that there is a monomial ordering on \( A \) such \( J \) has a finite left Gröbner basis \( G \).

Then we can distinguish between the following four situations:

1. No leading monomials of elements in \( G \) divide the leading monomial of any power of \( s \).
2. There is an element in \( G \) whose leading monomial divides the leading monomial of some power of \( s \). In this situation, we have the following sub-situations.
2.1. \( J \cdot s \subset J \) and \( \dim_K(\text{End}_A(A/J)) < \infty \).

2.2. One of the two conditions in 2.1. does not hold.

2.2.1. The intersection is zero.

2.2.2. The intersection is not zero.

We now consider the first case.

**Lemma 4.7.** If there exists no \( g \in G \) such that \( \text{lm}(g) \) divides \( \text{lm}(s^k) \) for some \( k \in \mathbb{N}_0 \), then \( J \cap K[s] = \{ 0 \} \).

**Proof.** Let \( 0 \neq b \in J \cap K[s] \). Then \( \text{lm}(b) = \text{lm}(s^k) \) for some \( k \in \mathbb{N}_0 \). Since \( b \in J \), there exists \( g \in G \) such that \( \text{lm}(g) \mid \text{lm}(b) = \text{lm}(s^k) \). \( \square \)

In the second situation however, we cannot in general state whether the intersection is trivial or not as the following example illustrates.

**Remark 4.8.** The converse of the previous lemma does not hold. For instance, consider \( K[x, y] \) and \( J = \langle y^2 + x \rangle \). Then \( J \cap K[y] = \{ 0 \} \) while \( \{ y^2 + x \} \) is a Gröbner basis of \( J \) for any ordering.

In situation 2.1. though, the intersection is not zero as the following lemma shows, inspired by the sketch of the proof of Theorem 4.3 in Saito et al. (2000).

**Lemma 4.9.** Let \( J \cdot s \subset J \) and \( \dim_K(\text{End}_A(A/J)) < \infty \). Then \( J \cap K[s] \neq \{ 0 \} \).

**Proof.** Consider the right multiplication with \( s \) as a map \( A/J \to A/J \) which is a well-defined \( A \)-module endomorphism of \( A/J \) as \( a - a' \in J \) implies that \( (a - a')s \in J \cdot s \subset J \), which holds by assumption for all \( a, a' \in A \). Since \( \text{End}_A(A/J) \) is finite dimensional, linear algebra guarantees that this endomorphism has a well-defined non-zero minimal polynomial \( \mu \). Moreover, \( \mu \) is precisely the monic generator of \( J \cap K[s] \) as \( \mu(s) = [0] \) in \( A/J \), hence \( \mu(s) \in J \cap K[s] \), and \( \deg(\mu) \) is minimal by definition. \( \square \)

**Remark 4.10.** In particular, the lemma holds if \( A/J \) itself is a finite dimensional \( A \)-module. In the case where \( A \) is a Weyl algebra and \( A/J \) is a holonomic module, we know that \( \dim_K(\text{End}_A(A/J)) < \infty \) holds (e. g. Saito et al. (2000)).

By the proof of the lemma, we have reduced our problem of intersecting an ideal with a subalgebra generated by one element to a problem from linear algebra, namely to the one of finding the minimal polynomial of an endomorphism.

**Proof of Theorem 4.3.** Let \( 0 \neq w \in \mathbb{R}_{\geq 0}^n, I \subset D \) such that \( D/I \) is a holonomic module, \( J := \text{in}_{(-w,w)}(I) \) and \( s := \sum_{i=1}^n w_i x_i \partial_i \). Without loss of generality let \( 0 \neq p = \sum c_{\alpha, \beta} x^\alpha \partial^\beta \in J \) be \((-w, w)\)-homogeneous. Then we obtain for every monomial in \( p \) by using the Leibniz rule

\[
x^\alpha \partial^\beta x_i \partial_i = x^{\alpha+\epsilon_i} \partial^{\beta+\epsilon_i} + \beta_i x^\alpha \partial^\beta
\]

\[
= (\partial_i x_i^{\alpha+1} - (\alpha_i + 1) x_i^{\alpha}) \frac{x^\alpha}{x_i^{\alpha}} \partial^\beta + \beta_i x^\alpha \partial^\beta
\]

\[
= (\partial_i x_i - (\alpha_i + 1) + \beta_i) x^\alpha \partial^\beta = (x_i \partial_i - \alpha_i + \beta_i)x^\alpha \partial^\beta.
\]
Put \( m = -w_\alpha + w_\beta \) for some term \( c_{\alpha, \beta} x^\alpha \partial^\beta \) in \( p \) where \( c_{\alpha, \beta} \) is non-zero. Since \( p \) is \((-w, w)\)-homogeneous, \( m \) does not depend on the choice of this term. Hence,

\[
p \cdot s = p \sum_{i=1}^{n} w_i x_i \partial_i = \sum_{i=1}^{n} w_i \sum_{\alpha, \beta} (x_i \partial_i - \alpha_i + \beta_i) c_{\alpha, \beta} x^\alpha \partial^\beta
\]

\[
= s \cdot p + \sum_{i=1}^{n} \sum_{\alpha, \beta} w_i (-\alpha_i + \beta_i) c_{\alpha, \beta} x^\alpha \partial^\beta = (s + m) \cdot p \in J.
\]

Therefore, \( J \cdot s \subset J \) holds. Since \( D/J \) is holonomic (Saito et al., 2000), Remark 4.10 and Lemma 4.9 yield the claim. □

If one knows in advance that the intersection is not zero, the following algorithm can be used for computing.

**Algorithm 4.11 (PrincipalIntersect).**

**Input:** \( s \in A, J \subset A \) a left ideal such that \( J \cap K[[s]] \neq \{0\} \).

**Output:** \( b \in K[[s]] \) monic such that \( J \cap K[[s]] = \langle b \rangle \)

\[ G := \text{a finite left Gröbner basis of } J \text{ (assume it exists)} \]

\[ i := 1 \]

**loop**

\[ \text{if there exist } a_0, \ldots, a_{i-1} \in K \text{ such that} \]

\[ \text{NF}(s^i, G) + \sum_{j=0}^{i-1} a_j \text{NF}(s^j, G) = 0 \] then

\[ \text{return } b := s^i + \sum_{j=0}^{i-1} a_j s^j \]

else

\[ i := i + 1 \]

**end if**

**end loop**

Note that because \( \text{NF}(s^i, G) + \sum_{j=0}^{i-1} a_j \text{NF}(s^j, G) = 0 \) is equivalent to \( s^i + \sum_{j=0}^{i-1} a_j s^j \in J \), the algorithm searches for a monic polynomial in \( K[[s]] \) that also lies in \( J \). This is done by going degree by degree through the powers of \( s \) until there is a linear dependency. This approach also ensures the minimality of the degree of the output. The algorithm terminates if and only if \( J \cap K[[s]] \neq \{0\} \). Note that this approach works over any field.

The check whether there is a linear dependency over \( K \) between the computed normal forms of the powers of \( s \) can be done by means of linear algebra.

**4.2.1. Applications**

Apart from computing global \( b \)-functions, there are various other applications of Algorithm 4.11.

**4.2.1.1. Solving zero-dimensional systems** Recall that an ideal \( I \subset K[x_1, \ldots, x_n] \) is called zero-dimensional if \( K[x_1, \ldots, x_n]/I \) is finite dimensional as a \( K \)-vector space. It is known (e.g. by Lemma 4.9) that in this case there exist \( 0 \neq f_i \in I \cap K[x_i] \) for each \( 1 \leq i \leq n \), which implies that the cardinality of the zero-set of \( I \) is finite.

In order to compute this zero-set, one can use the classical triangularization algorithms. These algorithms require to compute a Gröbner basis with respect to some elimination ordering (like lexicographic one), which might be very hard.
By Algorithm 4.11, a generator of $I \cap \mathbb{K}[x_i]$ can be computed without these expensive orderings. Instead, any ordering, hence a better suited one, may be freely chosen.

A similar approach is used in the celebrated FGLM algorithm (Faugère et al., 1993). See also Noro and Yokoyama (1999) for a different approach.

### 4.2.1.2. Computing central characters

Let $A$ be an associative $\mathbb{K}$-algebra. The intersection of a left ideal with the center of $A$, which is isomorphic to a commutative polynomial ring, is important for many algorithms, among other for the computation of the central character decomposition of a finitely presented module (cf. Levandovskyy (2005b)). In the situation, where the center of $A$ is generated by one element (which is not seldom), we can apply Algorithm 4.11 to compute the intersection (known to be often quite non-trivial) without engaging much more expensive Gröbner basis computations, which use elimination.

**Example 4.12.** Consider the universal enveloping algebra of the Lie algebra $\mathfrak{sl}_2$, $A = U(\mathfrak{sl}_2, \mathbb{K}) = \mathbb{K}\langle e, f, h \mid [e, f] = h, [h, e] = 2e, [h, f] = -2f \rangle$. It is known, that over a field of characteristic 0, the center of $A$ is $\mathbb{K}[4efh^2 - 2h]$. Consider a left ideal $L$ and a two-sided ideal $T$, both generated by $G = \{e^{11}, f^{12}, h^5 - 10h^3 + 9h \} \subset A$. Then consider $A$-modules $M_L = A/L$ and $M_T = A/T$, which turn out to be finite-dimensional over $\mathbb{K}$. We are interested in intersecting $L, T$ with $Z(A)$ and factorizing the output polynomial in one variable. The implementation of the Algorithm 4.11 in the library bfun.lib is described in Section 7.

```
LIB "ncalg.lib"; LIB "central.lib"; LIB "bfun.lib";
def A = makeUsl(2); setring A; // U(sl_2,Q)
ideal Z = center(2); // generators of deg <= 2
poly z = Z[1]; // we know there is just 1 generator
ideal I = e^11,f^12,(h-3)*(h-1)*h*(h+1)*(h+3);
ideal L = std(I); // left GB of I
vdim(L); // K-dimension of A/I
==> 559
vector vL = pIntersect(z,L); // L \cap K[z]
ideal T = twostd(I); // twosided GB of I
vdim(T); // K-dimension of A/T
==> 21
vector vT = pIntersect(z,T); // T \cap K[z]
ring r = 0,z,dp; // commutative univariate ring
// pretty-print factorization of polynomials:
print(matrix(factorize(vec2poly(imap(A,vT)),1)));
==> z-3,z-15
print(matrix(factorize(vec2poly(imap(A,vL)),1)));
==> z-3,z-440,z-8,z-48,z-168,z-15,z-99,z-120,
z-255,z-483,z-575,z+1,z-399,z-143,z-195,z-63,
z-80,z-288,z-360,z-224,z-323,z-35,z-24
```

Note, that all the computations, thanks to Algorithm 4.11, were completed in a couple of seconds, while the Gröbner-driven approach was still running after 20 minutes.
4.3. Intersecting an ideal with a multivariate subalgebra

We now consider the case where we intersect $J$ with the subalgebra $K[s] = K[s_1, \ldots, s_r]$ of an associative $K$-algebra $A$ for nonconstant, pairwise commuting $s_1, \ldots, s_r \in A$.

The following result is a consequence of a well-known characterization of zero-dimensional ideals.

Lemma 4.13. The ideal $J \cap K[s]$ is zero-dimensional if and only if for all $1 \leq i \leq r$ there exist $f_i \in J$ such that $\text{lm}(f_i) = s_i^{d_i}$ for some $d_i \in \mathbb{N}_0$.

Lemma 4.14. For a finite left Gröbner basis $G$ of $J$,

$$\text{GK.dim}(K[s]) \geq \text{GK.dim}(K[s]/(J \cap K[s])) \geq \text{GK.dim}(K[s]/(L(G) \cap K[s])).$$

Proof. For all $f \in J \cap K[s]$ there exists $g \in G$ such that $\text{lm}(g) | \text{lm}(f)$, which implies $\text{lm}(g) \in K[s]$ and thus, the claim follows. \qed

Note that the first inequality is strict if and only if $J \cap K[s] \neq \{0\}$.

We give a generalization of Algorithm 4.11 to compute a partial Gröbner basis of $J \cap K[s]$ up to a specified bound $k \in \mathbb{N}$.

Algorithm 4.15 (IntersectUpTo).

Input: $s_1, \ldots, s_r \in A$ pairwise commuting, $J \subset A$ a left ideal, $k \in \mathbb{N}$ an upper degree bound

Output: a GB for $J \cap K[s_1, \ldots, s_r]$ up to degree $k$

$G :=$ a partial left Gröbner basis of $J$ consisting of elements up to degree $k$

$d := 0$

$B := \emptyset$

while $d \leq k$ do

$M_d := \{s^\alpha \mid |\alpha| \leq d\}$

if there exist $a_m \in K$, not all 0, such that $\sum_{m \in M_d} a_m \text{NF}(m, G) = 0$ then

if $\sum_{m \in M_d} a_m m \notin \langle B \rangle$ then

$B := B \cup \{\sum_{m \in M_d} a_m m\}$

end if

end if

$d := d + 1$

end while

return $B$

A couple of improvements can be made to speed up the computation time.

If $p \in B$ with $\text{lm}(p) = m$ has been found, any monomial which is a multiple of $m$ can be discarded in the following iterations.

Let $G$ be a Gröbner basis of $J$ with respect to some fixed ordering $\prec$. By using $p \in J \cap K[s]$ if and only if $\text{lm}(p) \in L(G) \cap K[s]$, one may disregard $\{m \in M_d \mid \max_{\prec}(m') \in L(G) \cap M_d \prec m\}$.

Further note that $\text{NF}(m, G) = m$, if $m \notin L(G) \cap K[s]$.
Using these improvements and choosing \(<\) to be a degree ordering and the elements in \(B\) to be monic, the output of the algorithm equals the reduced Gröbner basis of \(J \cap \mathbb{K}[s]\) with respect to \(<\) up to degree \(k\). However, in general no termination criterion is known to us yet, that is a priori we do not know when we already have the complete needed basis of the intersection. Nevertheless, the termination is predictable if \(J \cap \mathbb{K}[s]\) is a principal ideal in \(\mathbb{K}[s]\). This situation often arises in the computation of Bernstein-Sato ideals, see Section 3.4. Moreover, another possibility for the algorithm to stop will be when the set of monomials we consider becomes empty on some step, which is the case if and only if \(J \cap \mathbb{K}[s]\) is zero-dimensional.

As one can see, the results above can be generalized by replacing the commutativity condition for a subalgebra \(S\) with the condition, that \(S\) is a \(G\)-algebra in a \(K\)-algebra \(A\). This and further generalizations will be studied in the next articles. Note, that under some extra requirements the algorithm will terminate after finally many steps without setting an explicit degree bound. Hence, in such cases a generally complicated elimination with Gröbner bases can be replaced by much easier and predictable Gröbner-free approach. The latter will, of course, allow to solve harder computational problems.

As it was noted in Levandovskyy (2006), even the existence of a certain elimination ordering in \(G\)-algebras is not guaranteed. Consider the algebra \(B = \mathbb{K}\langle x, y \mid yx = xy + y^2 \rangle\). Then the ordering condition of Def. 2.1 says \(x > y\) must hold for any ordering. Hence, we cannot use Gröbner basis in this \(G\)-algebra for computing the intersection of an ideal \(I\) with the subalgebra \(\mathbb{K}[x]\), since the latter requires the use of ordering with \(x < y\). One possibility would be to consider \(B\) as a \(K\)-algebra with the ordering \(x < y\) modulo the two-sided ideal, generated by \(y^2 − yx + xy\). But this ideal has infinite two-sided Gröbner basis, hence doing the elimination via passing to \(K\)-algebra setting is problematic, since it depends on the input ideal \(I\).

Despite these complications, it is obvious, that the preimage of an ideal in a subalgebra does exist. Hence, Algorithm 4.15 is indeed the only computational possibility to get some information about such a preimage.

5. Bernstein-Sato Polynomial of \(f\)

Let \(f \in \mathbb{K}[x_1, \ldots, x_n]\). One possibility to define the Bernstein-Sato polynomial of \(f\) is to apply the global \(b\)-function for specific weights.

**Definition 5.1.** Let \(B(s)\) denote the global \(b\)-function of the univariate Malgrange ideal \(I_f\) of \(f\) (cf. Section 3) with respect to the weight vector \(w = (1, 0, \ldots, 0) \in \mathbb{R}^{n+1}\), that is the weight of \(\partial t\) is 1. Then \(b(s) := B(-s − 1)\) is called the global \(b\)-function or the *Bernstein-Sato polynomial* of \(f\).

By Theorem 4.3, \(b(s) \neq 0\) holds. Moreover, it is well known that \(-1\) is always a root of the Bernstein-Sato polynomial for nonconstant \(f\) and Kashiwara proved that all its roots are negative rational numbers (Kashiwara, 1976/77).

The following version of Bernstein’s theorem (Bernstein, 1971) gives us another option to define the Bernstein-Sato polynomial.

**Theorem 5.2** (Saito et al. (2000)). The Bernstein-Sato polynomial \(b(s)\) of \(f\) is the unique monic polynomial of minimal degree in \(\mathbb{K}[s]\) satisfying the identity

\[
P \cdot f^{s+1} = b(s) \cdot f^s \quad \text{for some operator } P \in D[s].
\]
Since \( P : f - b(s) \in \Ann(f^s) \) holds, the Bernstein-Sato polynomial is the monic polynomial of minimal degree in \( K[s] \) that also lies in \( \Ann(f^s) + \langle f \rangle \), hence \( b(s) \) is the monic generator of this intersection.

**Remark 5.3.** There are the following choices for computing the Bernstein-Sato polynomial:

1. Compute a Gröbner basis either of
   
   a) \( J = \in(\{-w, w\}(I_f)) \), which amounts to 1 Gröbner basis computation in \( D_n \) or
   
   b) \( J = \Ann(f^s) + \langle f \rangle \), which requires 2 Gröbner basis computations in \( D_n[s] \).

2. Intersect \( J \) with \( K[\xi] \) where \( \xi \) is either \( s \) or \( \sum_i w_i x_i \partial_i \)
   
   a) the classical elimination-driven approach (needs 1 tough Gröbner basis computation) or
   
   b) using Algorithm 4.11 with no Gröbner basis computation.

It is very interesting to investigate the new approach for the computation of Bernstein-Sato polynomials, arising as the combination of the two methods

1. \( \Ann(f^s) \) via Briançon-Maisonobe (cf. Section 3 and Levandovskyy and Martín-Morales (2008)),

2. \( (\Ann(f^s) + \langle f \rangle) \cap K[s] \) via Algorithm 4.11.

For the computation of \( \in(\{-w, w\}(I_f)) \) using the method of weighted homogenization as described in Section 4.1, the following choice of weights is proposed in Noro (2002) for an efficient Gröbner basis computation:

\[
\begin{align*}
\hat{u} &= (\deg_\partial(f), \hat{u}_1, \ldots, \hat{u}_n), \\
v &= (1, \deg_\partial(f) - \hat{u}_1 + 1, \ldots, \deg_\partial(f) - \hat{u}_n + 1)
\end{align*}
\]

such that the weight of \( t \) is \( \deg_\partial(f) \) and the weight of \( \partial_i \) is 1. Here, \( \hat{u} \in \mathbb{R}^n_{\geq 0} \) is an arbitrary vector and \( \deg_\partial(f) \) denotes the weighted total degree of \( f \) with respect to \( \hat{u} \). The vector \( \hat{u} \) may be chosen heuristically in accordance to the shape of \( f \) or by default, one can set \( \hat{u} = (1, \ldots, 1) \).

6. **Enhancements to steps of algorithms**

6.1. **Enhanced computation of** \( \Ann_{D_n[s]}(f^s) \)

Consider the set of generators \( G := \{ f \partial t + s, \{ f_i \partial t + \partial_i \mid 1 \leq i \leq n \} \} \) of an ideal \( J \), coming from the Briaçon-Maisonobe method. According to the latter, we have to eliminate \( \partial_i \) from \( J \), that is to compute \( J \cap D_n[s] = \Ann_{D_n[s]}(f^s) \).

Since any element \( h \) from \( J \) has a presentation as

\[
h = a_0(f \partial t + s) + \sum_{i=1}^{n} a_i(f_i \partial t + \partial_i) = (a_0 f + \sum_{i=1}^{n} a_i f_i) \partial t + (a_0 s + \sum_{i=1}^{n} a_i \partial_i),
\]

then for all \( (a_0, a_1, \ldots, a_n) \in \text{Syz}(\{f, f_1, \ldots, f_n\}) \cap \mathbb{K}[x, s]^{n+1} \) we obtain that \( a_0 s + \sum_{i=1}^{n} a_i \partial_i \in J \cap D_n[s] \).

Moreover, it is known, that indeed the above elements generate the \( \mathbb{K}[x] \)-submodule of all the elements in \( J \cap D_n[s] \), which total degree in \( \partial_i \) does not exceed 1.

Consider the set \( T_f = \{ f, f_1, \ldots, f_n \} \subset \mathbb{K}[s] \subset D_n[s] \), a left ideal \( D_n[s] T_f \subset D_n[s] \) and an ideal \( \mathbb{K}[x] T_f \subset \mathbb{K}[x] \). Then a Gröbner basis of \( \mathbb{K}[x] T_f \) is a Gröbner basis for \( D_n[s] T_f \).
as well. Denote by $S_f$ a set of generators of the module $\text{Syz}(T_f) \subset \mathbb{K}[x]^{n+1}$. By e. g. generalized Schreyer’s theorem (Levandovskyy, 2005a), it follows that the module of left syzygies $\text{LeftSyz}_{D_n}[s](T_f) = D_n[s]S_f$.

Let $\prec_1$ be a monomial module ordering on $\mathbb{K}[x]^{n+1}$, which is a position-over-term ordering, which gives preference to the 1st component. Since degree of $f$ is always by 1 bigger than the degree of $\frac{\partial f}{\partial x_i}$, the cofactors to $f$ have respectively smaller degree.

**Algorithm 6.1** (SannfsBMSyz).

**Input:** $f \in \mathbb{K}[x]$

**Output:** $\text{Ann}_{D_n[s]}(f^s)$

$T_f := \{ f, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \} \subset \mathbb{K}[x]$

$S_f := \text{Syz}(T_f) \subset \mathbb{K}[x]^{n+1}$

$S_f := \text{GRÖBNERBasis}(S_f)$ with respect to $\prec_1$

create ring $D_n[s]$

form $S_a := \{ a_0s + \sum_{i=1}^n a_i \partial_i \}$ for every gen $a$ of $S_f$

$S_a := \text{GRÖBNERBasis}(S_a) \in D[s]$ with respect to an ordering

$G := \{ f \partial t + s, \frac{\partial f}{\partial x_i} \partial t + \partial_i, \ldots, \frac{\partial f}{\partial x_n} \partial t + \partial_n \} \subset D[\partial t, s]$

$G := \text{GRÖBNERBasis}(G \cup S_a)$ with respect to an elimination ordering for $\partial t$

return $(G \cap D[s])$

**Remark 6.2.** One of major difficulties in the computation of Gröbner basis (especially with respect to an elimination ordering) is the need to compute numerous intermediate polynomials (of usually high degree and with big coefficients) in order to come to a polynomial in the answer, which is often of small degree with coefficients of moderate size. Actually the set of generators $S_a$, which we compute in the syzygy-driven algorithm, generates already a part of the answer, though the corresponding ideal is, in general, not yet the complete answer.

Computing a Gröbner basis of $S_a$ and adding it to the original set of generators $G$ allows to avoid at first place the discovery of elements of $S_a$ in the Gröbner basis computation of $G \cup S_a$ and hence allows to decrease the number of intermediate unpleasant polynomials, which are needed in such computation. This is important, since in the answer there are no polynomials of degree zero with respect to $\partial_i$, that is $\text{Ann}_{D_n[s]}(f^s) \cap \mathbb{K}[x, s] = 0$ due to the fact, that the only element from the ring $\mathbb{K}[x, s]$, annihilating $f^s$, is zero. Hence with $S_a$ we add the set of elements of smallest possible total degree in $\partial_i$ that is of degree 1. Such elements are, in general, very hard to compute via the Gröbner-driven elimination.

However, it is very interesting to derive conditions, under which the above algorithm is more efficient than the one of Briançon-Maisonobe. We observe that it is not true for a couple of examples. See section 7.1.

6.2. Enhanced computation of $b_f(s)$

In the following Lemma we collect folklore results and supply them with short proofs for the completeness of exposition.

**Lemma 6.3.** Let, as before, $f \in \mathbb{K}[x] \setminus \{0\}$.

1. $\forall 1 \leq i \leq n$ we have $f \partial_i - s \frac{\partial f}{\partial x_i} \in \text{Ann}_{D[s]} f^s$ and $\frac{\partial f}{\partial x_i} \partial_j - \frac{\partial f}{\partial x_j} \partial_i \in \text{Ann}_{D[s]} f^s$

2. For $f \in \mathbb{K}$, $b_f(s) = 1$. For $f \in \mathbb{K}[x] \setminus \{ \mathbb{K} \}$, $(s + 1) | b_f(s)$.
Hence these elements can be reduced to an inclusion $I \subseteq \mathbb{K}[s]$ that indeed will in general keep the factor $(s+1)$, thus operating with larger polynomials of higher degree. Hence the claim. □

Proof. We use shortcut $f_i := \frac{\partial f}{\partial x_i}$. Consider surjective $\mathbb{K}$-alg homomorphism $\pi_\alpha : D[s] \to D$, $s \mapsto \alpha$ and apply it to the inclusion $D[s]/(P(s)f - b(s)) \subset \text{Ann}_D f^\alpha$. Then we have an inclusion

$$\pi_\alpha((P(s)f - b(s))) = (P(\alpha)f - b(\alpha)) \subset (\text{Ann}_D f^\alpha) \mid_{s=\alpha} = \text{Ann}_D f^\alpha$$

(1) Direct calculation.

(2) By using $\pi_{-1}$ from above, we obtain $P(-1) = b(-1)f^{-1}$ modulo $\text{Ann}_D f^0 = \langle \partial_1, \ldots, \partial_n \rangle$. Hence $P(-1) = p(x) \in \mathbb{K}[x]$ and $p(x) = b(-1)f^{-1}$, which can be true only in two cases:
1. $f \in \mathbb{K}^*$, then $P(s) = f^{-1} \in \mathbb{K}$ and $b(s) = 1$,
2. $f \notin \mathbb{K}^*$, then $b(-1) = 0$.

(3) Let us write $P(s) = \sum_i P_i \partial_i + P_0$ for $P_0 \in \mathbb{K}[x,s]$ and $P_i \in D[s]$. Computing modulo $\text{Ann}_D f^s$ and using (1) we can present $P(s)f = P_0f + \sum_i P_i \partial_i f = P_0f + (s + 1)\sum_i P_i f_i$. By (2) $b(-1) = 0$, hence by specializing $s$ to $-1$ in Bernstein’s equation we get $P(-1) \cdot 1 = b(-1)f^{-1} = 0$. Thus $P(-1) \in \text{Ann}_D f(1) = \langle \partial_1, \ldots, \partial_n \rangle$. In particular, $P_0(-1) = 0$ and hence $s + 1 \mid P_0 \in \mathbb{K}[x,s]$. Moreover,

$$\sum_i P_i f_i + \frac{P_0}{s+1}f - \frac{b(s)}{s+1} \in \text{Ann}_D f^s$$

and the claim follows.

(4) Since $I = I(\text{Sing}(V(f))) = \langle f, f_1, \ldots, f_n \rangle \subset \mathbb{K}[x]$, smoothness takes place when $1 \in I$, hence by (3) we have $1 \in \langle \frac{b(s)}{s+1} \rangle$ and thus, $b(s) = s + 1$. □

Lemma 6.4. For a fixed algorithm, which computes $\text{Ann}_D f^s$, let us consider two ideals $I_1 = \text{Ann}_D f^s + \langle f \rangle$ and $I_2 = \text{Ann}_D f^s + \langle f, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \rangle$ (note that $I_1, I_2 \subset D[s]$). There are two following algorithms, which take an ideal and a polynomial as input and return Bernstein-Sato polynomial, namely

Algorithm 1. $b(s) = \text{PINTERSECT}(I_1, s)$.

Algorithm 2. $b(s) = (s+1) \cdot \text{PINTERSECT}(I_2, s)$.

Then the Algorithm 2 is more efficient than the Algorithm 1.

Proof. Performing the principal intersection, the Algorithm 2 will compute one normal form less (of an element of high degree) than the Algorithm 1. Moreover, the normal forms in Algorithm 2 are taken with respect to a bigger ideal, what makes respective computations easier as well. By Lemma 6.3 (1) we know that $f \partial_i - sf_i \in \text{Ann}_D f^s$. Hence these elements can be reduced to $(s+1)\cdot f_i \in I_1 = \text{Ann}_D f^s + \langle f \rangle$. Meanwhile in $I_2 = \text{Ann}_D (f^s) + \langle f, f_1, \ldots, f_n \rangle$ we reduce $f \partial_i - sf_i$ automatically to zero. Note, that indeed $(s+1)I_2 \subset I_1 \subset I_2$ holds and hence, in the process of computing a Gröbner basis of $I_1$ (Algorithm 1), the operations with commutative elements of the kind $(s+1)f_i$ will in general keep the factor $(s+1)$, thus operating with longer polynomials of higher degree. Hence the claim. □
6.3. Enhanced computation of normal forms

When computing normal forms of the form $\text{NF}(s^i, J)$ like in Algorithm 4.11 we can speed up the reduction process by making use of the previously computed normal forms.

**Lemma 6.5.** Let $A$ be a $K$-algebra, $J \subset A$ a left ideal and let $f \in A$. For $i \in \mathbb{N}$ put
\[ r_i = \text{NF}(f^i, J), \quad q_i = f^i - r_i \in J \quad \text{and} \quad c_i = \frac{b(f^i, r_i)}{e(r_i, q_i)} \quad \text{provided} \ r_1 q_i \neq 0. \]
For $r_1 q_i = 0$ we put $c_i = 0$. Then we have for all $i \in \mathbb{N}$
\[ r_{i+1} = \text{NF}(fr_i, J) = \text{NF}([f^i - r_i, r_1]_{c_i} + r_i r_1, J). \]

**Proof.** It holds that $f^{i+1} = f q_i + f r_i \rightarrow f r_i$, which shows the first equation. On the other hand, $f^{i+1} = q_i f + r_i f = q_i q_1 + r_i (q_1 + r_1) = q_i q_1 + q r_1 + r_1 q_1 + r_i r_1 \rightarrow q_i r_1 + r_i r_1 = (f^i - r_i) r_1 + r_i r_1 \rightarrow \lfloor f^i - r_i, r_1 \rfloor_{c_i} + r_i r_1$, which proves the second equation. \( \square \)

As a direct consequence, we obtain the following result for some $K$-algebras of special importance.

**Corollary 6.6.** If $A$ is a $G$-algebra of Lie type (e. g. a Weyl algebra), then
\[ r_{i+1} = \text{NF}(fr_i, J) = \text{NF}([f^i - r_i, r_1] + r_i r_1, J) \quad \text{holds}. \]
If $A$ is commutative, we have
\[ r_{i+1} = \text{NF}(r_i r_1, J) = \text{NF}(r_1^{i+1}, J). \]

Note, that computing Lie bracket $[f, g]$ both in theory and in practice is easier and faster, than to compute $[f, g]$ as $f \cdot g - g \cdot f$, see e. g. Levandovskyy and Schönemann (2003).

**Remark 6.7.** We work on enhanced algorithms for the computation of the Bernstein operator from Theorem 5.2 as well and will report on the progress in forthcoming articles.

7. Implementation

In Noro (2002), M. Noro proposed methods of modular change of ordering and modular solving of linear equations to be used in his approach, which is based on a kind of Algorithm 4.11. In our implementation we decided to develop, test and enhance first purely characteristic 0 methods, thus having the possibility to adjoin modular methods later.

For the computation of $b$-functions and Bernstein-Sato polynomials, we offer the following procedures in the SINGULAR library $\text{bfun.lib}$:

- $\text{bfct}$ computes $\text{in}_{[-w, w]}(I_f)$ using weighted homogenization with weights $u, v$ for an optional weight vector $\hat{u}$ (by default $\hat{u} = (1, \ldots, 1)$) as described above, and then uses Algorithm 4.11 with the enhancement from Corollary 6.6 for the intersection, where the occurring systems of linear equations are solved by means of linear algebra.

- $\text{bfctSyz}$ computes $\text{in}_{[-w, w]}(I_f)$ as in $\text{bfct}$ and then uses Algorithm 4.11, where the linear equations are treated as polynomial ones and then solved by computing syzygies.

- $\text{bfctAnn}$ computes $\text{Ann}(f^s)$ via Algorithm 6.1 and then computes the intersection of $\text{Ann}(f^s) + \langle f, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \rangle$ with $K[s]$ analogously to $\text{bfct}$.
bfct computes the initial ideal and the intersection at once using a homogenized elimination ordering (see also Hartillo-Hermoso (2001)). For the global $b$-function of an ideal $I$, bfctIdeal computes $\text{in}_{(-w,w)}(I)$ using standard homogenization, i.e. weighted homogenization where all weights are equal to 1, and then proceeds the same way as bfct. Recall that $D/I$ must be holonomic as in Saito et al. (2000).

All these procedures work as the following example illustrates for bfct and the hyperplane arrangement $xyz(y - z)(y + z)$.

```
LIB "bfun.lib";
ring r = 0,(x,y,z),dp;
poly f = x*y*z*(y-z)*(y+z);
bfct(f);
==> [1]:
==> _[1]=-1
==> _[2]=-5/4
==> _[3]=-3/4
==> _[4]=-3/2
==> _[5]=-1/2
==> [2]:
==> 3,1,1,1,1
```

7.1. Comparison

We use the polynomials in Table 1 for test examples, where we measure the total running time of each call to a system in a batch mode. In this time the initialization of a system, loading of an example file, the actual computation and the writing of an output are included.

| Example | Input |
|---------|-------|
| ab23    | $(z^2 + w^3)(2xz + 3yw^2)$ |
| cnu6    | $(xz + y)(x^6 - y^6)$ |
| cnu7    | $(xz + y)(x^7 - y^7)$ |
| tt43    | $x^4 + y^4 + z^4(xyz)^3$ |
| xyzcusp45 | $(xz + y)(x^4 + y^5)$ |
| uw18    | $xyz(x - z)(z - y)(y + z)(2x + 2y - z)$ |
| uw22    | $xyz(x + z)(x - y)(y - z)$ |
| uw27    | $xyz(x + y)(-x + 2y + z)(x + y + z)(y + z)$ |
| uw28    | $xyz(x + z)(-x + y + z)(x + y)(y + z)$ |
| uw29    | $xyz(x + z)(x + y)(-3x - y + 2z)(y + z)$ |
| uw30    | $xyz(x - 2z)(-x + y + z)(x - y)(y + z)$ |
The running times in the tables below are given in “[hours][h]:[minutes]:[seconds]” format. We use the shortcuts \( t^* \) when we have stopped the process after the time \( t \) and \( t^\dagger \) when the process ran out of memory after the time \( t \).

The tests were performed on a machine with 4 Dual Core AMD Opteron 64 Processor 8220 (2800 MHz) (only one processor could be used at a time) equipped with 32 GB RAM (at most 16 GB were allowed to us) running openSUSE 11 Linux.

We first request the computation of \( \text{Ann}_{D_n}(f^s) \) and the Bernstein-Sato polynomial comparing the different algorithms from Section 6. We use the notation from Lemma 6.4.

### Table 2. Comparison of the algorithms from Section 6

| Example | \( \text{Ann}_{D_n}(f^s) \) | Bernstein-Sato polynomial |
|---------|-----------------|--------------------------|
|         | \( \text{SannfsBMSyz} \) | \( \text{SannfsBM} \) based | \( \text{SannfsBMSyz} \) based |
|         | Alg. 1 | Alg. 2 | Alg. 1 | Alg. 2 |
| ab23    | 0:01 | 0:02 | 0:05 | 0:03 | 0:06 | 0:03 |
| cnu6    | 0:01 | 0:01 | 0:01 | 0:01 | 0:01 | 0:01 |
| cnu7    | 0:09 | 0:12 | 0:11 | 0:11 | 0:19 | 0:12 |
| tt43    | 0:01 | 0:01 | 0:03 | 0:01 | 0:03 | 0:01 |
| xyzcusp45 | 0:56 | 1:22 | 1:30 | 1:16 | 1:20 | 1:10 |
| uw18    | 3:38 | 0:04 | 17:36 | 12:38 | 16:24 | 11:15 |
| uw22    | 2h:04:01\dagger | 0:04 | 2h:17:36\dagger | 2h:16:55\dagger | 2h:28:07\dagger | 2h:28:53\dagger |

Further, we compare our implementations for the computation of the Bernstein-Sato polynomial with the existing ones in the computer algebra systems RISA/ASIR and MACAULAY2.

We have used RISA/ASIR version 20071022, MACAULAY2 version 1.1 with version 1.0 of Dmodules.m2 and SINGULAR 3-1-0 with bfun.lib version 1.13.

We would like to stress, that in our implementation of bfun.lib we have restricted ourselves to the use of characteristic zero methods, in order to see what can we achieve with them. The implementation of ASIR by M. Noro (Noro, 2002) uses the methods in prime characteristic, which can be applied to our implementation as well. However, the values in the table above indicate, that the difference in timings is not devastating for our cause.

As the timings in Table 3 suggest, the approach via the initial ideal seems to be specially well suited for hyperplane arrangements, while it looks like that the performance of the annihilator based method is better for other kind of input (we took non-quasi-homogeneous singularities). See Walther (2005) for details about generic arrangements.
Table 3. Comparison of different systems

| Example       | ASIR bfunction | ASIR bfct | MACAULAY2 globalBFunction | MACAULAY2 bfct | SINGULAR bfct | SINGULAR bfctAnn |
|---------------|----------------|-----------|---------------------------|----------------|---------------|-----------------|
| ab23          | 0:23           | 0:17      | 0:27                      | 0:17           | 0:04          |                 |
| cmu6          | 1:39           | 0:54      | 14:03                     | 0:01           | 0:01          |                 |
| cmu7          | 7:32           | 4:46      | 4h:03:39\(^x\)           | 0:06           | 0:20          |                 |
| tt43          | 0:07           | 0:05      | 0:05                      | 0:17           | 0:01          |                 |
| xyzcusp45     | 1:52           | 1:10      | 4h:18:35\(^x\)           | 3:05           | 3:01          |                 |
| uw18          | 7:22           | 29h:35:54\(^x\) | 4h:08:16\(^x\)     | 6:21           | 12:27         |                 |
| uw22          | 2:12           | 4h:04:05\(^x\) | 4h:01:43\(^x\)     | 2:24           | 2h:42:02      |                 |
| uw27          | 2:37           | 3h:05:14\(^x\) | 11h:45:18\(^x\)     | 4:40           | 6h:55:35\(^x\) |                 |
| uw28          | 1:36           | 10h:23:40\(^x\) | 3h:03:00\(^x\)     | 3:10           | 3h:03:32\(^x\) |                 |
| uw29          | 1:48           | 3h:51:14\(^x\) | 10h:23:42\(^x\)     | 2:52           | 3h:01:30\(^x\) |                 |
| uw30          | 1:58           | 5h:14:18\(^x\) | 3h:06:57\(^x\)     | 3:09           | 3h:00:13\(^x\) |                 |

8. Bernstein-Sato Polynomial for a Variety

In the paper of Budur, Mustață and Saito (Budur et al., 2006), using the theory of V-filtrations of Kashiwara (Kashiwara, 1983) and Malgrange (Malgrange, 1983), the theory of the Bernstein-Sato polynomial of an arbitrary variety has been developed. We present here the construction of the Bernstein-Sato polynomial of an affine algebraic variety.

Given two positive integers \(n\) and \(r\), for the rest of this section we fix the indices \(i, j, k, l\) ranging between 1 and \(r\) and an index \(m\) ranging between 1 and \(n\).

Let \(f = (f_1, \ldots, f_r)\) be an \(r\)-tuple in \(K[x]^r\). Consider a free \(K[x, s, 1/f]\)-module of rank one generated by the formal symbol \(f^s\) and denote it by \(M = K[x, s, 1/f] \cdot f^s\). Here, \(s = (s_1, \ldots, s_r)\), \(f = \frac{1}{f_r} \cdots \frac{1}{f_1}\) and \(f^s = f_1^{s_1} \cdots f_r^{s_r}\). Moreover, we denote by \(K\langle S\rangle\) the universal enveloping algebra \(U(\mathfrak{gl}_r)\), generated by the set of variables \(S = (s_{ij})\), \(i, j = 1, \ldots, r\) subject to relations:

\[
[s_{ij}, s_{kl}] = \delta_{jk}s_{il} - \delta_{il}s_{kj}.
\]

Then, we denote by \(D_n\langle S\rangle := D_n \otimes_K K\langle S\rangle\), which is a \(G\)-algebra of Lie type by e. g. Levandovskyy and Schönemann (2003).

The module \(M\) has a natural structure of left \(D_n\langle S\rangle\)-module when the variables \(s_{ij}\) act in the following way (\(i \leq j\)):

\[
s_{ij} \cdot (G(s) \cdot f^s) = s_i \cdot G(s + \epsilon_j - \epsilon_i) \frac{f_j}{f_i} \cdot f^s \in M,
\]

where \(G(s)\) is an element in \(K[x, s, 1/f]\) and \(\epsilon_j\) stands for the \(j\)-th basis vector.

One can easily observe that the action of \(s_{ii}\) on \(M\) coincides with the multiplication by \(s_i\) from the left.
Following the ideas by Malgrange, one can also consider $M$ as a $D_n(R) \otimes_k D_r(T)$-module, with $T = k[t], t = (t_1, \ldots, t_r), \partial t = (\partial t_1, \ldots, \partial t_r)$ and the action
\[
t_i \cdot (G(s) \cdot f^s) = G(s + \epsilon_i) f_j \cdot f^s,
\]
\[
\partial t_i \cdot (G(s) \cdot f^s) = -s_i G(s - \epsilon_i) \frac{1}{f_j} \cdot f^s.
\]
Observe that the action of $s_{ij}$ above corresponds to the action of $-\partial t_i \cdot t_j$.

**Theorem 8.1** (Budur, Mustaţă, and Saito (2006)). For every $r$-tuple $f = (f_1, \ldots, f_r) \in k[x]^r$ there exists a non-zero polynomial in one variable $b(s) \in k[s]$ and $r$ differential operators $P_1(S), \ldots, P_r(S) \in D_n(S)$ such that
\[
\sum_{k=1}^r P_k(S) f_k \cdot f^s = b(s_1 + \cdots + s_r) \cdot f^s \in M.
\]

The Bernstein-Sato polynomial $b_f(s)$ of $f = (f_1, \ldots, f_r)$ is defined to be the monic polynomial of the lowest degree in the variable $s$ satisfying the equation (2). It is demonstrated in Budur et al. (2006), that every root of the Bernstein-Sato polynomial is rational. Let $I$ be the ideal generated by $f_1, \ldots, f_r$ and $Z$ the (not necessarily reduced) algebraic variety associated with $I$ in $k^n$. Then it can be verified that $b_f(s)$ is independent of the choice of a system of generators of $I$, and moreover that $b_f(s) = b_f(s - \text{codim} Z + 1)$ depends only on $Z$. For instance, the Bernstein-Sato polynomial of $f(x, y) \in k[x, y]$ and Bernstein-Sato polynomial of the variety defined by the ideal $(f(x, y), z)$ coincide. However, due to the codimension, there is a shift between $b_f(s)$ and $b_{(f(x, y), z)}(s)$.

Now, let us denote by $\text{Ann}_{D_n(S)}(f^s)$ the left ideal of all elements $P(S) \in D_n(S)$ such that $P(S) \cdot f^s = 0$. We call this ideal the annihilator of $f^s$ in $D_n(S)$. From the definition of the Bernstein-Sato polynomial it is clear that
\[
(\text{Ann}_{D_n(S)}(f^s) + \langle f_1, \ldots, f_r \rangle) \cap k[s_1 + \cdots + s_r] = \langle b_f(s_1 + \cdots + s_r) \rangle.
\]

Since the final intersection can be computed with the Principal Intersection method 4.11, the above formula provides an algorithm for computing the Bernstein-Sato polynomial of affine algebraic varieties, once we know a Gröbner basis of the annihilator of $f^s$ in $D_n(S)$. The rest of this section is dedicated to the solving of this problem.

8.1. The annihilator of $f^s$ in $D_n(S)$

Consider the generalization of Malgrange’s ideal $I_f$ associated with $f = (f_1, \ldots, f_r),
\[
I_f = \left\langle t_i - f_i, \partial_m + \sum_{j=1}^r \frac{\partial f_j}{\partial x_m} \partial t_j \middle| 1 \leq i \leq r \right\rangle \subset D_n(t, \partial t)
\]
Here we give a computer-algebraic proof to the following Lemma, whose assertion is expected as in Saito et al. (2000) (for instance).

**Lemma 8.2.** $I_f$ is a maximal ideal in $D_n(R) \otimes_k D_r(T)$ and $I_f = \text{Ann}_{D_n(R) \otimes_k D_r(T)} f^s$.

**Proof.** $(t_i - f_i) \cdot f^s = f_i f^s - f_i f^s = 0$. For any $m$, observe that
\[
\partial_m \cdot (f_1^{s_1} \cdots f_r^{s_r}) = \sum_{j=1}^r \partial_m \cdot (f_j^{s_j})(f_1^{s_1} \cdots f_r^{s_r})(f_j^{s_j})^{-1} = \sum_{j=1}^r s_j \frac{\partial f_j}{\partial x_m} (f_j^{-1})(f_1^{s_1} \cdots f_r^{s_r})
\]
Since $\partial t_k$ acts on $f^s$ by the multiplication with $-s_jf_j^{-1}$, the generators of the second type annihilate $f^s$, so $I_f \subseteq \text{Ann}_{D_n(R) \otimes K D_r(T)} f^s$.

By Lemma 3.1, the set of generators of $I_f$ is the same as the set $S_1$ in the Lemma and hence there is a monomial ordering, such that $S_1$ is a Gröbner basis. In particular, $I_f$ is a proper ideal. The set of leading monomials of $S_1$ is then $L = \{t_j, \partial_m\}$. Since any monomial ordering on $\mathbb{N}^{2r+2n}$ can be presented as weighted degree ordering with the weight vector $w$ with strictly positive entries (see e. g. Bueso et al. (2003)), we see that $\text{GK. dim}(D_n(R) \otimes K D_r(T))/I_f = \text{GK. dim } \mathbb{K}\langle t_j, \partial_m \rangle / \langle L \rangle = r + n$.

Assume the left ideal $I_f$ is not maximal, then there exists $p \notin I_f$, such that $I_f \subseteq I_f + \langle p \rangle \subset D_n(R) \otimes K D_r(T)$. In particular, $\text{im}(p)$ does not include the elements of $L$ above. If $I_f + \langle p \rangle$ is a proper ideal, its set of leading monomials strictly includes $L$ and has at least one element more. But then the dimension argument as above shows, that $\text{GK. dim}(D_n(R) \otimes K D_r(T))/(I_f + \langle p \rangle) < r + n$, what contradicts Bernstein’s inequality. Hence $I_f$ is maximal and it is equal to the annihilator. □

**Theorem 8.3.** Let $f = (f_1, \ldots, f_r)$ be an $r$-tuple in $\mathbb{K}[x]^r$ and $D_n(\partial t, S)$ the $\mathbb{K}$-algebra generated by $D_n, \partial t$ and $S$ with the corresponding non-commutative relations. Then the following ideal of $D_n(S)$ coincides with the annihilator of $f^s$ in $D_n(S)$:

$$[D_n(\partial t, S) \langle s_{ij} + \partial t_1f_j, \partial_m + \sum_{k=1}^r \frac{\partial f_k}{\partial x_m} \partial t_k \mid 1 \leq i, j \leq r \rangle] \cap D_n(S).$$

**Proof.** Let $\phi : D_n(S) \hookrightarrow D_n \otimes K D_r(T)$ be the $\mathbb{K}$-algebra homomorphism given by $\phi(s_{ij}) = -t_j \partial t_i$, $\phi(t_j) = t_j$ and $\phi(P) = P$ for all $P$ in $D_n$. In view of Lemma 8.2, we observe that $\text{Ann}_{D_n(S)}(f^s) = D_n(S) \cap I_f = \phi^{-1}(I_f)$.

The morphism $\phi$ can be written as $\phi = 1_{D_n} \otimes \varphi$, where $\varphi : \mathbb{K}(S) \hookrightarrow D_r(T) = \mathbb{K}(t, \partial t)$, $s_{ij} \mapsto t_j \partial t_i - \delta_{ij}$. Thus we can apply Proposition 2.3 to obtain that $\text{Ann}_{D_n(S)}(f^s) = (E'I_\varphi + E'I_f) \cap D_n(S)$, with $I_\varphi = \langle \{s_{ij} + t_j \partial t_i + \delta_{ij} \mid 1 \leq i, j \leq r\} \rangle$ and $E' = D_n(\{t_j, \partial t_i, s_{ij}\})$ subject to relations

$$[s_{ij}, s_{kl}] = \delta_{jk}s_{il} - \delta_{il}s_{jk}, [\partial t_k, t_j] = \delta_{jk}, [s_{ij}, t_k] = -\delta_{ik}t_j, [s_{ij}, \partial t_k] = \delta_{jk}\partial t_i.$$  

By Theorem 2.2, $E'$ is a G-algebra, if there exists an elimination ordering for $\{\partial t_1, \ldots, \partial t_r\}$ on $D_n(\partial t, S)$, obeying the conditions

$$\text{im}(\delta_{jk}s_{il} - \delta_{il}s_{jk}) < s_{ij}s_{kl}, \ t_j < s_{ij}t_i, \ \text{and} \ \partial t_i < s_{ij}t_j.$$  

It is clear, that such orderings exist.

Now, we proceed with the elimination of $\{t_i, \partial t_i \mid 1 \leq i \leq r\}$ from $(I_f + I_\varphi)$ in $E'$. By taking a monomial ordering with the property $\{t_j\} \gg \{x_i\}, \{\partial t_i, s_{ij}\} \gg \{x_i, \partial t_j\}$, we start with eliminating $\{t_j\}$ first.

By Lemma 3.1, the generators $G_1$ of $I_f$ form a Gröbner basis. The ideal $I_\varphi$ in the current situation is generated by $G_2 = \{s_{ij} + t_j \partial t_i + \delta_{ij}\}$. In order to prove, that $G_1 \cup G_2$ is a a Gröbner basis, we apply the generalized Product Criterion (Lemma 2.4). At first we apply reduction process by $G_1$, thus obtaining $G'_2 = \{s_{ij} + f_j \partial t_i\}$. Then

$$[s_{ij} + f_j \partial t_i, s_{kl} + f_l \partial t_k] = \delta_{jk}(s_{il} + f_i \partial t_l) - \delta_{il}(s_{kj} + f_j \partial t_k)$$
which clearly reduces to zero. The next kind of pairs

\[ [s_{ij} + f_j \partial t_i, t_k - f_k] = -\delta_{ik} t_j + \delta_{ik} f_j = -\delta_{ik}(t_j - f_j) \]

again reduces to zero. It remains to consider

\[ [s_{ij} + f_j \partial t_i, \partial m + \sum_{k=1}^r \frac{\partial f_k}{\partial x_m} \partial t_k] = \sum_{k=1}^r \frac{\partial f_k}{\partial x_m} \delta_{jk} \partial t_i - \partial t_i[\partial m, f_j] = 0, \]

since \([\partial m, f_j] = \frac{\partial f_j}{\partial x_m}\).

Hence, \(G_1 \cup G'_2\) is a a Gröbner basis and hence, by the Elimination Lemma, \(G_3 = (G_1 \cup G'_2) \backslash \{t_j - f_j\}\) is a Gröbner basis of \((I_f + I_\partial) \cap D_n(\partial t_k, s_{ij}). Thus, it follows, that

\[
\text{Ann}_{D(S)}(f^*) = \left< s_{ij} + \partial t_i f_j, \partial m + \sum_{k=1}^r \frac{\partial f_k}{\partial x_m} \partial t_k \right> \cap D_n(S). \quad \square
\]

Indeed, the result we have proved is a natural generalization of the algorithm for computing the annihilator of \(f^*\) in \(D_n[s]\) (cf. 3.2) given by Briançon-Maisonobe in Briancon and Maisonobe (2002). Finally, the algorithm for the computation of \(\text{Ann}_{D_n(S)} f^*\) looks as follows:

**Algorithm 8.4 (SannfsVar).**

**Input:** \(f = (f_1, \ldots, f_r),\) an \(r\)-tuple in \(\mathbb{K}[x]^r\)

**Output:** \(\{G_1(S), \ldots, G_e(S)\},\) a Gröbner basis of \(\text{Ann}_{D_n(S)}(f^*)\)

Let \(D_n(\partial t, S)\) be the algebra in Corollary 8.3, with non-commutative relations

\[ [\partial_i, x_i] = 1, \quad [s_{ij}, \partial t_k] = \delta_{jk} \partial t_i, \quad [s_{ij}, s_{kl}] = \delta_{jk} s_{il} - \delta_{il} s_{kj}. \]

\[ J_1 := \langle \{s_{ij} + \partial t_i f_j \mid 1 \leq i, j \leq r\} \rangle \]
\[ J_2 := \langle \{\partial_m + \sum_{k=1}^r \frac{\partial f_k}{\partial x_m} \partial t_k \mid 1 \leq m \leq n\} \rangle \]
\[ J := J_1 + J_2 \quad \triangleright J \subseteq D_n(\partial t, S) \]
\[ H := \text{G.B. of } J \text{ w.r.t. a compatible elim. ordering for } \partial t_1, \ldots, \partial t_r \]
\[ H \cap D_n(S) := \{G_1(S), \ldots, G_e(S)\} \]

**return** \(\{G_1(S), \ldots, G_e(S)\}\)

8.2. Elimination orderings in \(D_n(\partial t, S)\)

One of the bottlenecks of the presented algorithm for computing the Bernstein-Sato polynomial for varieties is to calculate the corresponding annihilator. An elimination term ordering for \(\{\partial t_1, \ldots, \partial t_r\}\) in \(D_n(\partial t, S),\) which has \(2n + r + r^2\) variables, has to be considered. In addition, due to the structure of the \(G\)-algebra, this ordering \(<\) has to be chosen with the following extra restrictions

\[ \partial t_i < s_{ij} \partial t_j, \quad \text{Im}(\delta_{ik} s_{il} - \delta_{il}s_{kj}) < s_{ij} s_{kl} \]

for all indices \(i, j, k, l\) where the expression makes sense. The efficiency of the method strongly depends on the selected ordering. Therefore it is worth analysing it in detail.
Assume that $<$ is such an ordering and let us consider the first two rows of the matrix representing the ordering in this way.

\[
\begin{array}{c|ccc}
\partial t_1 & \cdots & \partial t_r & S \times \partial_x \\
p_1 & \cdots & p_r & a \ b \ c \\
q_1 & \cdots & q_r & \alpha \ \beta \ \gamma \\
\hline
<'
\end{array}
\]

The vectors $a$, $b$ and $c$ must be zero, since $<$ is an elimination ordering for $\{\partial t_i\}$. The conditions $\partial t_i < s_{ij}\partial t_j$, imply $p_i \leq p_j$ for all $i,j$. Thus all $p_1, \ldots, p_r$ are equal and can be taken as 1.

From computational point of view, since the variables $\{s_{ij}\}$ do not commute with $\{\partial t_i\}$, these two blocks must be together in the elimination ordering, namely $\beta = \gamma = 0$, otherwise Gröbner bases computation may be slow.

In the implementation we have taken $\alpha_{ii} = 2$ and $\alpha_{ij} = 1$ for $i \neq j$, and $q = 0$. However, in some examples we have observed that lexicographical orderings are also useful, see Example 8.8 below.

In this section we have described an algorithm for computing the Bernstein-Sato polynomial of affine algebraic varieties without any homogenization but passing through the computation of the annihilator of $f^s$ in $D_n(S)$. Now, other methods are illustrated.

8.3. Another approach

As Budur et. al. point out in (Budur et al., 2006, p. 794), the Bernstein-Sato polynomial for varieties coincides, up to shift of variables, with the $b$-function in (Saito et al., 2000, p. 194), if the weight vector is chosen appropriately. Let us describe this algorithm more carefully.

Let $I_f = \text{Ann}_{D_n(t,\partial t)}(f^s)$ be the Malgrange ideal associated with $f = (f_1, \ldots, f_r)$ and consider the weight vector $w = ((0, \ldots, 0), (1, \ldots, 1)) \in \mathbb{Z}^n \times \mathbb{Z}^r$ which gives weight 0 to $\partial_m$ and weight 1 to $\partial t_i$. Consider the $V$-filtration $V = \{V_k \mid k \in \mathbb{Z}\}$ on $D_n(t,\partial t)$ with respect to $w$, where $V_k$ is spanned by $\{t^\alpha \cdot \partial t^\beta \mid -|\alpha| + |\beta| \leq k\}$ over $\mathbb{K}$. Note that the associated graded ring $\oplus_{k \in \mathbb{Z}} V_k / V_{k-1}$ is isomorphic again to the $(n+r)$-Weyl algebra $D_n(t,\partial t)$ and the homogeneous parts are the following.

$$V_k / V_{k-1} = \begin{cases} D_n(t_i \cdot \partial t_j)\partial^\beta, & |\beta| = k > 0; \\ D_n(t_i \cdot \partial t_j), & k = 0; \\ D_n(t_i \cdot \partial t_j)t^\alpha, & -|\alpha| = k < 0. \end{cases}$$

Denote by $B(s)$ the $b$-function of the holonomic ideal $I_f$ with respect to $w$. Recall that $B(s)$ is the monic generator of the ideal $\text{in}_{(-w,w)}(I_f) \cap \mathbb{K}[t_1 \partial t_1 + \cdots + t_r \partial t_r]$.

As in the classical case, i.e. $r = 1$, the following result holds.

**Lemma 8.5.** $b_f(s) = (-1)^{\text{deg} B(s)}B(-s - r)$.

**Proof.** Consider $P_1(S), \ldots, P_k(S) \in D_n(S)$ differential operators satisfying the functional equation $\sum_{k=1}^r P_k(S)f_k \bullet f^s = b_f(s_1 + \cdots + s_r) \bullet f^s$. Then $b_f(s_1 + \cdots + s_r) - \sum_{k=1}^r P_k(S)f_k$ is an element in $\text{Ann}_{D_n(S)}(f^s)$ and hence applying the Mellin transform,
or equivalently making the substitution $s_{ij} \mapsto -t_j \partial t_i - \delta_{ij}$, one obtains the following element in $I_f$.

$$b_f(-t_1 \partial t_1 - \cdots - t_r \partial t_r - r) - \sum_{k=1}^r P_k(-t_j \partial t_i - \delta_{ij})f_k \in I_f$$

Modulo $I_f$ the polynomials $f_k$ in the above expression can be replaced by $t_k$, since $t_k - f_k \in I_f$. Finally, taking initial parts one concludes that $b_f(-t_1 \partial t_1 - \cdots - t_r \partial t_r - r) \in \text{in}_{(-w,w)}(I_f)$, which means that $B(s)$ divides $b_f(-s - r)$.

Conversely, by definition there exists a differential operator $P(t, \partial t) \in I_f \subset D_n(t, \partial t)$ such that $B(t_1 \partial t_1 + \cdots + t_r \partial t_r) = \text{in}_{(-w,w)}(P(t, \partial t))$. In particular $P(t, \partial t)$ has $V$-degree zero and hence it can be decomposed into $V$-homogeneous parts as follows

$$P(t, \partial t) = B(t_1 \partial t_1 + \cdots + t_r \partial t_r) + \sum_{|\alpha| \geq 1} Q_\alpha(t_1 \partial t_1) t^\alpha.$$

Since $P(t, \partial t) \in I_f$, making left reduction of $P(t, \partial t)$ with respect to $\{t_i - f_i\}$ we arrive at $B(t_1 \partial t_1 + \cdots + t_r \partial t_r) + \sum_{|\alpha| \geq 1} Q_\alpha(t_1 \partial t_1) f^\alpha \in I_f \cap D_n(t_1 \partial t_1)$. After applying the substitution $t_i \partial t_j \mapsto -s_{ij} - \delta_{ij}$, we conclude that $B(-s_1 - \cdots - s_r - r)$ belongs to the ideal $\text{Ann}_{D_2(s)}(f^\alpha) + \langle f_1, \ldots, f_r \rangle$ and the proof is complete. □

Algorithms for computing this $b$-function, which use the homogenization technique in the Weyl algebra, are given in Section 4, see also Saito et al. (2000). We describe the complete algorithm for computing Bernstein-Sato polynomials using initial parts.

**Algorithm 8.6 (bfctVar).**

**Input:** $f = (f_1, \ldots, f_r)$, an $r$-tuple in $\mathbb{K}[x]^r$; $Z$, variety associated with $f$

**Output:** $b_Z(s) = b_f(s - \text{codim } Z + 1)$, Bernstein-Sato polynomial of $Z$

Let $D_n(t, \partial t) = D_n \otimes D_r$ be the $(n + r)$-Weyl algebra.

$I := \langle \{t_i - f_i\}_{i=1}^r, \{\partial_m + \sum_{j=1}^r \frac{\partial f_j}{\partial t_j} \partial t_j\}_{m=1}^n \rangle \quad \triangleright I \subseteq D_n(t, \partial t)$

$w := ((0, \ldots, 0), (1, \ldots, 1)) \in \mathbb{Z}^n \times \mathbb{Z}^r$

$J := \text{InitialIdeal}(I, w) \quad \triangleright \text{Algorithm 4.6}$

$s := -(\partial t_1 \cdot t_1 + \cdots + \partial t_r \cdot t_r)$

$b(s) := \text{pIntersect}(s, J) \quad \triangleright \text{Algorithm 4.11}$

**return** $b(s - \text{codim } Z + 1)$

As for the Bernstein-Sato polynomial of a polynomial (indicated in Remark 5.3), we have two different methods for the computation of Bernstein-Sato polynomial of an affine algebraic variety, namely

- minimal polynomial for $s_1 + \cdots + s_r$ in $D/\text{in}_{(-w,w)}(I_f)$,
- $(\text{Ann } f^s + \langle f_1, \ldots, f_r \rangle) \cap \mathbb{K}[s_1 + \cdots + s_r]$, where the intersection can be done either with the 4.11 method, than by using Gröbner basis elimination.

It is important to investigate the connection of these methods and especially their applicability to different classes of varieties. Our experience shows, that no method is clearly superior to the other one in general. Thus it is desired to have both of them in any package for $D$-modules.

**Remark 8.7.** Very recently, the authors have realized another approach for computing Bernstein-Sato polynomials for varieties. In Shibuta (2008), Shibuta modifies the definition of Budur-Mustaţă-Saito’s Bernstein-Sato polynomials to determine a system of
generators of the multiplier ideals of a given ideal. Then he obtains an algorithm for computing Bernstein-Sato polynomials, which gives an algorithm for computing multiplier ideals and jumping coefficients. His methods are based on the theory of Gröbner bases in Weyl algebras and corresponds to the natural generalization given by Oaku and Takayama, hence they need homogenization techniques.

We conclude this section showing several examples calculated with our experimental implementation.

**Example 8.8.** Let \( TX = \mathcal{V}(x_0^2 + y_0, 2x_0x_1 + 3y_0^2y_1) \subset \mathbb{C}^4 \) the tangent bundle of \( X = \mathcal{V}(x^2 + y^3) \subset \mathbb{C}^2 \). The annihilator of \( f^s \) in \( D(S) \) and the Bernstein-Sato polynomial of \( TX \) using the previous approach can be computed with the SINGULAR commands `SannfsVar` and `bfcfVarAnn`.

\[
\text{LIB "bfunVar.lib";}
\text{ring R = 0,(x0,x1,y0,y1),Dp;}
\text{LIB "bfunVar.lib";}
\text{ideal F = x0^2+y0^3, 2*x0*x1+3*y0^2*y1;}
\text{bfcfVarAnn(F);}
\]

The output is lengthy, hence we supress it. We obtain an ideal called \( \text{bfctVarAnn}(F) \); with 15 generators and the Bernstein-Sato polynomial for \( TX \), which looks as follows

\[
b_{TX}(s) = (s + 1)^2(s + \frac{1}{3})^2(s + \frac{2}{3})^2(s + \frac{1}{2})(s + \frac{5}{6})(s + \frac{7}{6}).
\]

Analogously, one can consider the tangent bundle of \( V(x^4 + y^3) \). In this case the Bernstein polynomial has degree 42 and it equals

\[
\left(s + \frac{1}{5}\right)^2\left(s + \frac{3}{5}\right)^2\left(s - \frac{1}{5}\right)^2\left(s + \frac{2}{5}\right)^2\left(s + \frac{1}{15}\right)^2\left(s + \frac{4}{15}\right)^2\left(s + \frac{1}{6}\right)^2
\]
\[
\left(s + \frac{13}{20}\right)^2\left(s - \frac{1}{10}\right)^2\left(s + \frac{1}{20}\right)^2\left(s + \frac{1}{2}\right)^2\left(s + \frac{7}{20}\right)^2\left(s + \frac{4}{15}\right)^2\left(s + \frac{5}{15}\right)^2
\]
\[
\left(s + \frac{2}{3}\right)^2\left(s + \frac{12}{15}\right)^2\left(s + \frac{5}{12}\right)^2\left(s + \frac{1}{12}\right)^2\left(s + \frac{7}{15}\right)^2\left(s + \frac{3}{20}\right)^2\left(s + \frac{3}{10}\right)^2\left(s + \frac{3}{4}\right)^2
\]
\[
\left(s + \frac{1}{10}\right)^2\left(s + \frac{11}{20}\right)^2\left(s + \frac{9}{20}\right)^2\left(s + \frac{1}{12}\right)^2\left(s + \frac{3}{4}\right)^2.
\]

The result was not able to be obtained using the elimination ordering given in section 8.2. Instead the following monomial ordering (in SINGULAR format) has been taken:

\"(a(1,1), a(0,0,2,1,1,2), (dp(6), rp)\".

Note that if \( TX \) is the variety defined by \( f = (f_1, f_2) \), then \( b_f(s) \) has always negative roots. As this examples shows the same is not true for \( b_Z(s) \), due to the codimension.

**Example 8.9.** Let \( Z \) be the algebraic variety defined by \( f = (x_1^2 - x_2x_3, x_2^2 - x_1x_3, x_3^2 - x_1^2x_2) \). Then

\[
b_{Z}(s) = (s + 1)^2s + \frac{7}{9})(s + \frac{8}{9})(s + \frac{10}{9})(s + \frac{4}{9}).
\]

This is actually Example 5.10 in Shibuta (2008) and it corresponds to the space of monomial curve \( \text{Spec} \mathbb{C}[T^3, T^4, T^5] \subset \mathbb{C}^3 \). Note that the \( b \)-function coincides with the one that appears in Shibuta (2008), since we are computing \( b_Z(s) \) instead of \( b_f(s) \).
Example 8.10. Let $Z$ be the so-called Hirzebruch-Jung singularity of type $(5, 2)$. It is a cyclic quotient singularity and can be seen as the algebraic variety associated with the ideal $\langle z_3^2 - z_2 z_4, z_2^2 z_3 - z_1 z_4, z_2^2 - z_1 z_3 \rangle \subset \mathbb{C}[z_1, z_2, z_3, z_4]$. Then
$$b_Z(s) = (s + 1)^3(s + \frac{4}{3})(s + \frac{5}{3})(s + \frac{3}{2}).$$

Example 8.11. As an intractible example we would like to mention the following one. Let $Z$ be the cyclic quotient singularity of type $(6; 1, 2, 3)$. It can also be seen as the toric variety associated with the matrix
$$A := \begin{pmatrix} 6 & 4 & 2 & 0 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \end{pmatrix}.$$

The corresponding ideal can be taken to have 9 generators in $\mathbb{C}[z_1, \ldots, z_7]$. We are not yet able to compute the Bernstein-Sato polynomial even trying several elimination orderings.

9. Conclusion and Future Work

From our recent articles there follow some important conclusions.

1. In Levandovskyy and Martín-Morales (2008) we proved, that for the computation of Bernstein-Sato polynomial of a hypersurface, the homogenization, used in the method of Oaku and Takayama is superfluous. This, however, does not apply to the situation of computing a $b$-function with respect to weights. By several steps in Levandovskyy and Martín-Morales (2008); Andres et al. (2009) and in this article we have shown, that the method by Briançon and Maisonobe can be seen as natural refinement of the method of Oaku and Takayama.

2. From our investigations it follows, that we cannot in general prove, that either initial-based method or annihilator-based one for the computation of Bernstein-Sato polynomial is definitely more efficient than the other one. Instead, on numerous examples we see that roughly the domain of better performance of initial-based method includes hyperplane arrangements, while for other singularities annihilator-based method scores distinctly better. It is important to continue these investigations and derive more classes of polynomials, when possible, in order to use this information in attempts to estimate at least the practical complexity of $D$-module computations.

3. Also for the syzygy-driven method to compute $\text{Ann}_{D[s]}(f^s)$ we do not have yet a proof of its superiority over the method by Briançon and Maisonobe. Due to the reasons we explain in this paper one could achieve such superiority. But on the other hand, there are examples, which show the contrary. Even if there are only a few examples of such kind, it is interesting to investigate this phenomenon deeper.

4. We pay so much attention to the algorithms for the case of a hypersurface due to many reasons. According to our proofs for the case of a variety, we use indeed the same technology. We expect to generalize all of enhancements we have described to the case of an affine variety in a similar way we presented the generalization of algorithms for annihilator and Bernstein-Sato polynomial. It is very important
to stress, that working in the case of a variety is a priori much more involved computationally, hence more attention on very effective algorithms need to be paid.

Acknowledgements

We would like to thank Francisco Castro-Jiménez, José-María Ucha, Gert-Martin Greuel, Enrique Artal and José-Ignacio Cogolludo for their constant support and motivation in our work over years.

Special thanks go to Hans Schönenemann and Michael Brickenstein, whose continuous support in numerous aspects of SINGULAR and discussions on computer algebra we deeply appreciate.

We are grateful to Masayuki Noro and Anton Leykin for their help and careful explanations, concerning the computer algebra systems ASIR and MACAULAY2.

We also thank Uli Walther for fruitful discussions.

References

Andres, D., Levandovskyy, V., Martín-Morales, J., July 28-31 2009. Principal intersection and Bernstein-Sato polynomial of an affine variety. In: May, J. P. (Ed.), Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC’09). ACM Press, Seoul, Republic of Korea, pp. 231–238.

Bahloul, R., 2001. Algorithm for computing Bernstein-Sato ideals associated with a polynomial mapping. J. Symbolic Comput. 32 (6), 643–662, effective methods in rings of differential operators.

Bernštejn, I. N., 1971. Modules over a ring of differential operators. An investigation of the fundamental solutions of equations with constant coefficients. Functional Anal. Appl. 5 (2), 89–101.

Briançon, J., Maisonobe, P., 2002. Remarques sur l’idal de Bernstein associ des polynomes. Preprint no. 650, Univ. Nice Sophia-Antipolis.

Budur, N., Mustaţă, M., Saito, M., 2006. Bernstein-Sato polynomials of arbitrary varieties. Compos. Math. 142 (3), 779–797.

Bueso, J., Gómez-Torrecillas, J., Verschoren, A., 2003. Algorithmic methods in non-commutative algebra. Applications to quantum groups. Kluwer Academic Publishers.

Bueso, J. L., Gómez Torrecillas, J., Lobillo, F. J., Castro, F. J., 1998. An introduction to effective calculus in quantum groups. In: Rings, Hopf algebras, and Brauer groups (Antwerp/Brussels, 1996). Vol. 197 of Lecture Notes in Pure and Appl. Math. Dekker, New York, pp. 55–83.

Castro-Jiménez, F., Narváez-Macarro, L., 1997. Homogenising differential operators. Pre-publicación no. 36, Universidad de Sevilla.

CoCoA Team, 2009. CoCoA: a system for doing Computations in Commutative Algebra. Available at http://cocoa.dima.unige.it.

Coutinho, S., 1995. A primer of algebraic D-modules. Cambridge Univ. Press.

Faugere, J. C., Gianni, P., Lazard, D., Mora, T., 1993. Efficient computation of zero-dimensional gröbner bases by change of ordering. J. Symbolic Computation 16 (4), 329–344.
Gago-Vargas, J., Hartillo-Hermoso, M., Ucha-Enriquez, J., 2005. Comparison of theoretical complexities of two methods for computing annihilating ideals of polynomials. J. Symbolic Computation 40 (3), 1076–1086.

Greuel, G.-M., Levandovskyy, V., Schönemann, H., 2006. PLURAL. a SINGULAR 3.0 subsystem for computations with non–commutative polynomial algebras. Centre for Computer Algebra, University of Kaiserslautern.
URL http://www.singular.uni-kl.de

Greuel, G.-M., Pfister, G., Schönemann, H., 2005. SINGULAR 3.0. a computer algebra system for polynomial computations. Centre for Computer Algebra, University of Kaiserslautern.
URL http://www.singular.uni-kl.de

Hartillo-Hermoso, M. I., 2001. About an algorithm of T. Oaku. In: Ring theory and algebraic geometry (León, 1999). Vol. 221 of Lecture Notes in Pure and Appl. Math. Dekker, New York, pp. 241–250.

Kandri-Rody, A., Weispfenning, V., 1990. Noncommutative Gröbner bases in algebras of solvable type. J. Symbolic Comput. 9 (1), 1–26.

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

Kashiwara, M., 1983. Vanishing cycle sheaves and holonomic systems of differential equations. In: Algebraic geometry (Tokyo/Kyoto, 1982). Vol. 1016 of Lecture Notes in Math. Springer, Berlin, pp. 134–142.

Levandovskyy, V., 2005a. Non-commutative computer algebra for polynomial algebras: Gröbner bases, applications and implementation. Doctoral Thesis, Universität Kaiserslautern.
URL http://kluedo.ub.uni-kl.de/volltexte/2005/1883/

Levandovskyy, V., 2005b. On preimages of ideals in certain non–commutative algebras. In: Pfister G., C. S., Ufnarovski, V. (Eds.), Computational Commutative and Non-Commutative Algebraic Geometry. IOS Press.

Levandovskyy, V., 2006. Intersection of ideals with non-commutative subalgebras. In: Dumas, J.-G. (Ed.), Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC’06). ACM Press, pp. 212–219.

Levandovskyy, V., Martín-Morales, J., 2008. Computational D-module theory with SINGULAR, comparison with other systems and two new algorithms. In: Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC’08). ACM Press.

Levandovskyy, V., Schönemann, H., 2003. Plural — a computer algebra system for non-commutative polynomial algebras. In: Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC’03). ACM Press, pp. 176 – 183.

Malgrange, B., 1983. Polynômes de Bernstein-Sato et cohomologie évanescente. In: Analysis and topology on singular spaces, II, III (Luminy, 1981). Vol. 101 of Astérisque. Soc. Math. France, Paris, pp. 243–267.

McConnell, J., Robson, J., 2001. Noncommutative Noetherian rings. With the cooperation of L. W. Small. Graduate Studies in Mathematics. 30. Providence, RI: American Mathematical Society (AMS).

Noro, M., 2002. An efficient modular algorithm for computing the global \(b\)-function. In: Mathematical software (Beijing, 2002). World Sci. Publ., River Edge, NJ, pp. 147–157.
Noro, M., Shimoyama, T., Takeshima, T., 2006. Risa/Asir, an open source general computer algebra system. URL http://www.math.kobe-u.ac.jp/Asir

Noro, M., Yokoyama, K., 1999. A modular method to compute the rational univariate representation of zero-dimensional ideals. J. Symbolic Comput. 28 (1-2), 243–263, polynomial elimination—algorithms and applications.

Oaku, T., 1997a. An algorithm of computing $b$-functions. Duke Math. J. 87 (1), 115–132.

Oaku, T., 1997b. Algorithms for $b$-functions, restrictions, and algebraic local cohomology groups of $D$-modules. Adv. in Appl. Math. 19 (1), 61–105.

Oaku, T., 1997c. Algorithms for the $b$-function and $D$-modules associated with a polynomial. J. Pure Appl. Algebra 117/118, 495–518.

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

Schulze, M., 2004. A SINGULAR 3.0 library to compute invariants related to the Gauss-Manin system of an isolated hypersurface singularity gmssing.lib.

Shibuta, T., 2008. An algorithm for computing multiplier ideals. Available at http://arxiv.org/abs/math/0807.4302.

Takayama, N., 2003. KAN/sm1, a Gröbner engine for the ring of differential and difference operators. URL http://www.math.kobe-u.ac.jp/KAN/index.html

Tsai, H., Leykin, A., 2006. D-modules package for MACAULAY 2 – algorithms for D-modules. URL http://www.ima.umn.edu/~leykin/Dmodules

Walther, U., 2005. Bernstein-Sato polynomial versus cohomology of the Milnor fiber for generic hyperplane arrangements. Compos. Math. 141 (1), 121–145.