Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Symbolic Computation 37 (2004) 629–639
Journal of Symbolic Computation
www.elsevier.com/locate/jsc

On the computation of Bernstein–Sato ideals

J.M. Ucha*, F.J. Castro-Jiménez

Depto. Álgebra, Universidad de Sevilla, Apdo. 1160, E-41080 Sevilla, Spain

Received 21 May 2002; accepted 27 July 2003

Abstract

In this paper we compare the approach of Briançon and Maisonobe for computing Bernstein–Sato ideals—based on computations in a Poincaré–Birkhoff–Witt algebra—with the readily available method of Oaku and Takayama. We show that it can deal with interesting examples that have proved intractable so far.

Keywords: Bernstein–Sato

1. Introduction

Let $X = \mathbf{C}^n$ be the complex affine space of dimension n, A_n be the complex Weyl algebra of order n and $(f_1, \ldots, f_p)$ be polynomial functions on X, that is $f_i \in \mathbf{C}[x] = \mathbf{C}[x_1, \ldots, x_n]$. Let us consider the algebra $A_n[s_1, \ldots, s_p] = A_n \otimes_{\mathbf{C}} \mathbf{C}[s_1, \ldots, s_p]$ with the trivial action of the elements of $\mathbf{C}[s_1, \ldots, s_p]$. We will write $b(s)$, $P(s)$ for elements $b(s_1, \ldots, s_p)$, $P(s_1, \ldots, s_p)$ in $\mathbf{C}[s] = \mathbf{C}[s_1, \ldots, s_p]$ and $A_n[s] = A_n[s_1, \ldots, s_p]$ respectively.

Let $\mathcal{B}$ be the *Bernstein–Sato ideal* of $(f_1, \ldots, f_p)$ consisting of polynomials $b(s) \in \mathbf{C}[s]$ such that there exists a differential operator $P(s) \in A_n[s]$ satisfying

$$b(s)\, f_1^{s_1} \cdots f_p^{s_p} = P(s)\, f_1^{s_1+1} \cdots f_p^{s_p+1}.$$

In a similar way, other Bernstein–Sato ideals can be defined, namely

* Corresponding address: Universidad de Sevilla, Algebra Facultad de Matematicas, Apdo. 1160, E-41080 Sevilla, Spain. Tel.: +34-954-55-61-83; fax: +34-954-55-69-38.

E-mail address: ucha@us.es (J.M. Ucha).

0747-7171/$ - see front matter
doi:10.1016/j.jsc.2003.07.005

- $\mathcal{B}_j = \{b(s) \in \mathbf{C}[s] \mid b(s)F^s \in A_n[s] f_j F^s\}$ for $j \in \{1, \ldots, p\}$
- $\mathcal{B}_\Sigma = \{b(s) \in \mathbf{C}[s] \mid b(s)F^s \in \sum_{j=1}^{p} A_n[s] f_j F^s\}$

where F^s denotes $f_1^{s_1} \cdots f_p^{s_p}$. If $p = 1$ then $\mathcal{B} = \mathcal{B}_1 = \mathcal{B}_\Sigma$ and the monic generator of the principal ideal $\mathcal{B}$ is called the Bernstein–Sato polynomial, or simply the b-function, associated with f_1; see Bernstein (1972). Sabbah proved (see Sabbah, 1987) in the analytic setting that $\mathcal{B}$ is not zero, generalizing previous works of Bernstein (1972) and Björk (1979), both in the algebraic case and for $p = 1$.

The explicit calculation of these ideals is of major interest: many conjectures about how the important properties of the case $p = 1$ appear in the general case remain open. For example: when are the Bernstein–Sato ideals principal? Also, what can be said about their primary decomposition? (In the case $p = 1$ the roots of the Bernstein–Sato polynomial are rational numbers (Kashiwara, 1976).) See Maynadier (1997) for more details.

As far as we know there are three ways of computing Bernstein–Sato ideals in the algebraic case:

- The method due to Oaku and Takayama for calculating $\mathcal{B}$ is described in Procedure 2.2. of Oaku and Takayama (1999). It needs the calculation of Gröbner bases in a polynomial algebra over a Weyl algebra.
- The method proposed by Bahloul in Bahloul (2001). It provides algorithms for computing $\mathcal{B}_\Sigma$ and $\mathcal{B}_j$ for $j \in \{1, \ldots, p\}$ by adapting a previous work of Oaku (see Oaku, 1997). As it uses Gröbner bases with respect to non-well-ordered cases, the calculations are made in the *homogenized* Weyl algebra (Castro-Jiménez and Narváez-Macarro, 1997).
- The method recently proposed by Briançon and Maisonobe in Briançon and Maisonobe (2002) that computes the three types of Bernstein–Sato ideal. Their approach is used to show the constructability of the Bernstein–Sato ideals with respect to the space of parameters. The calculations are made in an intermediate algebra that appears as a natural generalization of the one used by Malgrange and Kashiwara in early works for the case $p = 1$ (see for example Kashiwara, 1976).

We will refer to the first method as the OT method and the third one as the BM method.

The aim of this paper is to compare the BM method as a computational (not just theoretical) alternative to the Oaku–Takayama algorithm, checking for a selected family of difficult examples in concrete implementations. As we will detail, both methods start with the calculation of the annihilator ideal $\mathrm{Ann}_{A_n[s]}(F^s)$. We will provide experimental evidence for the former approach being a better alternative.

In many examples the bottleneck of the calculation is not the above-mentioned annihilator but an elimination problem which occurs in a second step in both methods. Anyway, a shortcut to the annihilator leads in practice to the possibility of finding some of the Bernstein–Sato ideals (see the example in 3.4). Moreover, we think that there are interesting advantages of the BM method from the complexity point of view that are not simply related to the number of variables. They need to be brought to light in future works (see the example in 3.3).

2. Preliminaries

In this section we will recall the general setting of the PBW algebras and explain how the computation of the Bernstein–Sato ideals can be performed in this context.

2.1. $R_{n,p}$ as a PBW algebra

As in Briançon and Maisonobe (2002), we will work in the non-commutative algebra

$$R_{n,p} = A_n[s_1, \ldots, s_p, t_1, \ldots, t_p] = A_n[s, t],$$

an extension of the Weyl algebra A_n where the new variables s, t satisfy the relations $[s_i, t_j] = \delta_{ij} t_i$. As we have mentioned it is, in fact, a ring analogous to the one introduced by Malgrange and Kashiwara for $p = 1$.

The elements of $R_{n,p}$ can be represented as polynomials in a finite number of variables due to the following lemma:

Lemma 2.1. *In the algebra $R_{n,p}$ the following formulas hold:*

1. *For the set of variables $x_1, \ldots, x_n$ and $\partial_1, \ldots, \partial_n$ we have*

$$\partial_i^\beta x_i^\alpha = \sum_{k=0}^{\beta} \binom{\beta}{k} \partial_i^k (x_i^\alpha) \partial_i^{\beta-k},$$

for $i \in \{1, \ldots, n\}$, where $\partial_i(a)$ stands for the partial derivative of a with respect to x_i.

2. *For the set of variables $s_1, \ldots, s_p$ and $t_1, \ldots, t_p$ we have*

$$s_i^\lambda t_i^\mu = \sum_{j=0}^{\lambda} \binom{\lambda}{j} \mu^j t_i^\mu s_i^{\lambda-j},$$

for $i \in \{1, \ldots, p\}$.

Proof. The property (1) is very well known as a special case of the *Leibnitz formula*. The proof of (2) is very easy—by induction taking into account the easier formula

$$s_i t_i^\beta = t_i^\beta s_i + \beta t_i^\beta. \quad \square$$

Roughly speaking, you can choose a normal form for the elements of $R_{n,p}$ using monomials with the x, t variables to the left and ∂, s to the right, so the set

$$\mathcal{M} = \{x_1^{\alpha_1} \cdots x_n^{\alpha_n} \partial_1^{\beta_1} \cdots \partial_n^{\beta_n} t_1^{\lambda_1} \cdots t_p^{\lambda_p} s_1^{\mu_1} \cdots s_p^{\mu_p} = x^\alpha \partial^\beta t^\lambda s^\mu \mid \text{with } (\alpha, \beta, \lambda, \mu) \in \mathbf{N}^{2n+2p}\}$$

forms a basis of $R_{n,p}$ as a vector space over $\mathbf{C}$.

On the other hand, you can consider the total degree $<$ (the sum of all of the exponents for every variable) in the exponents of the monomials of $\mathcal{M}$ to ensure that

$$x^\alpha \partial^\beta t^\lambda s^\mu \cdot x^{\alpha'} \partial^{\beta'} t^{\lambda'} s^{\mu'} = x^{\alpha+\alpha'} \partial^{\beta+\beta'} t^{\lambda+\lambda'} s^{\mu+\mu'} + M,$$

where $M \in R_{n,p}$ is a sum of monomials with exponents less than $(\alpha+\alpha', \beta+\beta', \lambda+\lambda', \mu+\mu')$ with respect to $<$.

The last two properties—the existence of a **C**-basis and the good behaviour of the product of monomials—are the conditions needed to define a *Poincaré–Birkhoff–Witt algebra*. The work (Kandri-Rody and Weispfenning, 1990) (see also Bueso et al., 1998) is a good introduction to the subject of effective calculus in this very general family of rings which contains, for example, the *iterated Ore extensions*. In particular it is proved there that basic topics of computational commutative algebra such as Gröbner bases, the Buchberger algorithm and elimination orders can be developed in these PBW algebras.

2.2. $Ann_{A_n[s]}(F^s)$ and the Bernstein–Sato ideals from $R_{n,p}$

Let us consider the left A_n-module $M = \mathbf{C}[x][s, \frac{1}{F}]F^s$ where F is the product of the f_i. M has a natural structure of a left $R_{n,p}$-module where s_i acts by multiplication and the action of t_i is defined by

$$t_i a(x,s)F^s = -a(x, s-\epsilon_i)s_i \frac{1}{F^{\epsilon_i}} F^s,$$

where $a(x,s)$ is an element of $\mathbf{C}[x][s, \frac{1}{F}]$ and ϵ_i is the ith element of the canonical basis in $\mathbf{N}^p$.

Following Briançon and Maisonobe (2002), and like in Malgrange (1975), considering the annihilating ideal of F^s in the ring $R_{n,p}$ is the main point. It is generated by the family

$$\left\{ s_j + f_j t_j, \partial_i + \sum_l \frac{\partial f_l}{\partial x_i} t_l \;\middle|\; i = 1, \ldots, n;\ j = 1, \ldots, p \right\}.$$

So the annihilator of F^s in $A_n[s]$ is $\mathrm{Ann}_{R_{n,p}}(F^s) \cap A_n[s]$. Once you have the annihilator, the Bernstein–Sato ideal $\mathcal{B}$ can be calculated by eliminating the variables x_i, ∂_i, t_j, i.e. computing

$$\mathcal{B} = ((\mathrm{Ann}_{R_{n,p}}(F^s) \cap A_n[s]) + A_n[s]\langle F\rangle) \cap \mathbf{C}[s]. \tag{1}$$

Of course the formulas are analogous for $\mathcal{B}_\Sigma$ and $\mathcal{B}_j$, $j = 1, \ldots, p$:

$$\mathcal{B}_\Sigma = ((\mathrm{Ann}_{R_{n,p}}(F^s) \cap A_n[s]) + A_n[s]\langle f_1, \ldots, f_p\rangle) \cap \mathbf{C}[s], \tag{2}$$

$$\mathcal{B}_j = ((\mathrm{Ann}_{R_{n,p}}(F^s) \cap A_n[s]) + A_n[s]\langle f_j\rangle) \cap \mathbf{C}[s]. \tag{3}$$

2.3. The algorithm in $R_{n,p}$

We have in fact presented the following algorithm in the preceding section. The orderings $<_t$ (respectively $<_s$) denote any elimination orderings that consider the variables t (respectively s) greater than the others.

INPUT: $f_1, \ldots, f_p \in \mathbf{C}[x]$.
OUTPUT: The ideals $\mathcal{B}, \mathcal{B}_\Sigma$ and $\mathcal{B}_j$ for $j = 1, \ldots, p$.

(Step 1) $I := \mathrm{Ann}_{R_{n,p}}(F^s) = \langle s_j + f_j t_j, \partial_i + \sum_l \frac{\partial f_l}{\partial x_i} t_l, 1 \le i \le n, 1 \le j \le p\rangle$.
Compute a Gröbner basis $\mathcal{G}$ of I with respect to $<_t$.
$J := I \cap A_n[s] = \langle \mathcal{G} \cap A_n[s]\rangle$.

(Step 2) $K := J + \langle f_1 \cdots f_p \rangle$,
$K_\Sigma := J + \langle f_1, \ldots, f_p \rangle$,
$K_j := J + \langle f_j \rangle$ for $j = 1, \ldots, p$.
Compute Gröbner bases $\mathcal{G}_K, \mathcal{G}_\Sigma, \mathcal{G}_1, \ldots, \mathcal{G}_p$ of $K, K_\Sigma, K_1, \ldots, K_p$ with respect to $<_s$.

(Step 3) $\mathcal{B} = K \cap \mathbf{C}[s] = \langle \mathcal{G}_K \cap \mathbf{C}[s] \rangle$,
$\mathcal{B}_\Sigma = K_\Sigma \cap \mathbf{C}[s] = \langle \mathcal{G}_\Sigma \cap \mathbf{C}[s] \rangle$,
$\mathcal{B}_j = K_j \cap \mathbf{C}[s] = \langle \mathcal{G}_j \cap \mathbf{C}[s] \rangle$ for $j = 1, \ldots, p$.

See Briançon and Maisonobe (2002) to check the correctness of the algorithm. To finish this section, we recall how the computation of step 1 of the algorithm presented above is carried out in the Oaku–Takayama method. It needs the Weyl algebra $A_{p+n} = \mathbf{C}[t_1, \ldots, t_p, \partial_{t_1}, \ldots, \partial_{t_p}, x_1, \ldots, x_n, \partial_1, \ldots, \partial_n]$:

1. Compute

$$\left\langle t_j - f_j, \frac{\partial f_j}{\partial x_i} \partial_{t_j} + \partial_i, i = 1, \ldots, n, j = 1, \ldots, p \right\rangle \bigcap \mathbf{C}[t_1 \partial_{t_1}, \ldots, t_p \partial_{t_p}] \times \langle x, \partial_x \rangle,$$

as explained in Procedure 4.1. of Oaku and Takayama (1999). This elimination uses $2n + 4p$ variables because it is made in $A_{p+n}[u_1, \ldots, u_p, v_1, \ldots, v_p]$. The ideal considered is the one generated by

$$t_j - u_j f_j,$$
$$\partial_i + \sum_{l=1}^{p} \frac{\partial f_l}{\partial x_i} u_l \partial_{t_l}, \qquad i = 1, \ldots, n,$$
$$1 - u_j v_j, \qquad j = 1, \ldots, p.$$

To obtain the required intersection, eliminate the variables of type u, v.
2. Replace each $t_j \partial_{t_j}$ by $-s_j - 1$, for $j = 1, \ldots, p$, in the generators obtained.

3. Examples and comparisons

We present in this section examples of annihilators of f^s and b-functions that have proved intractable so far (here $p = 1$). We also include a comparison of running times between OT and BM methods computing Bernstein–Sato ideals for two functions.

We have used three different implementations to check how good the computations are in $R_{n,p}$ compared to the alternative homogenized Weyl algebra:

- The software **kan/sm1**. It is taken into account that the ring $\mathbf{C}\langle s, t \rangle$ is isomorphic to the ring of difference operators

$$\mathbf{C}\langle n, E_n^{-1} \rangle, \qquad E_n^{-1} n = (n-1) E_n^{-1}$$

by the correspondence $t = E_n^{-1}, s = n$. E_n^{-1} acts on a space of functions of n via $E_n^{-1} \bullet f(n) = f(n-1)$. The system **kan/sm1** (Takayama, 1991) provides some tools for this family of rings.

- The package **Plural** (Levandovskyy, 2002), designed by Levandovskyy as a part of the celebrated **Singular** (Greuel et al., 2001). It provides an excellent setting for non-commutative calculation of Gröbner bases.
- A tailored implementation in CLISP designed by the authors to compare CPU times of many examples in a not very ambitious environment. The first evidence that we found of the BM method being better was the possibility for our humble system to compute examples intractable to the more powerful programs using the algorithms of Oaku and Takayama.

3.1. Some selected annihilators of f^s

All the examples in this section have been treated using our CLISP prototype. It is not optimized, so the timing data must be taken into account only in order to compare the different methods.

The following table contains five interesting examples with some details of the computation of $\mathrm{Ann}_{A_n[s]}(f^s)$, for a polynomial f—namely: CPU time, number of elements (N.E.) in the reduced Gröbner basis (before the truncation of the elimination, of course), maximum number of monomials (N.M.) in the elements of the basis and maximum total degree (T.D.) of the elements of the basis.

Remark. In all these examples, the ordering used is the following one: to compare two exponents first look at the exponent of the variable t (in order to do the elimination). Second, look at the exponent of the variable s. To break ties, finally use a reverse graded lexicographical ordering with $x > \partial_x > t > s$.

f	CPU time		N.E.		N.M.		T.D.	
	OT (s)	BM (s)	OT	BM	OT	BM	OT	BM
$x^6 + y^4 + z^3$	1.21	0.17	15	6	5	4	6	7
$(x^3 + y^2)(x^2 + y^3)$	101.66	9.56	26	15	48	39	13	7
$xyz(x + y)(x + z)$	13 248.8	7565.22	76	26	118	150	12	10
$x^7 + y^7 + x^4y^4$	131.22	19.1	27	15	63	43	16	10
$x^7 + y^7 + z^7 + x^2y^2z^2$	5 514.5	2995.75	46	63	165	139	14	13

The computations of the Bernstein–Sato polynomial for the last two examples have been treated in Briançon et al. (1989) by a different method. Here we give generators of $\mathrm{Ann}_{A_n[s]}(f^s)$ for two examples from the table. The calculations are rather far from being trivial: the powerful system **kan/sm1** cannot manage their calculation using the OT method. The annihilators, however, look harmless:

- The non-generic arrangement of hyperplanes (taken from Walther, 2002) ($f = xyz(x + y)(x + z) = 0) \subset \mathbf{C}^3$:

$$\begin{aligned}\mathrm{Ann}_{A_3[s]}(f^s) = \langle & -5s + x\partial_x + y\partial_y + z\partial_z, x^2\partial_x + 2xz\partial_x + xy\partial_y + 2yz\partial_y \\ & - 4xz\partial_z - 3z^2\partial_z, -2xy\partial_x + 2xz\partial_x + 5xy\partial_y + 3y^2\partial_y \\ & + 2yz\partial_y - 5xz\partial_z - 2yz\partial_z - 3z^2\partial_z, -4y^2z\partial_x\partial_y \\ & - 2xz^2\partial_x\partial_y - 3xy^2\partial_y^2 - 3y^3\partial_y^2 - 5xyz\partial_y^2 - y^2z\partial_y^2\end{aligned}$$

$$- 2yz^2\partial_y^2 + 2xz^2\partial_x\partial_z + 4yz^2\partial_x\partial_z + 5xyz\partial_y\partial_z + 8y^2z\partial_y\partial_z + 5xz^2\partial_y\partial_z - yz^2\partial_y\partial_z 3z^3\partial_y\partial_z - 2xz^2\partial_z^2 - 4yz^2\partial_z^2 + 4xz\partial_x - 2xy\partial_y - 5y^2\partial_y - 5xz\partial_y - 6yz\partial_y - 2z^2\partial_y - 3xz\partial_z + z^2\partial_z, -x^2y\partial_y - xy^2\partial_y - 2xyz\partial_y - 2y^2z\partial_y + x^2z\partial_z + 2xyz\partial_z + xz^2\partial_z + 2yz^2\partial_z\rangle.$$

- The semi-quasi-homogeneous curve $(f = x^7 + y^7 + x^4y^4 = 0) \subset \mathbf{C}^2$:

$$\mathrm{Ann}_{A_2[s]}(f^s) = \langle 12xy^3s - \tfrac{147}{4}y^2s - \tfrac{9}{7}x^2y^3\partial_x - \tfrac{12}{7}xy^4\partial_y - \tfrac{3}{4}x^4\partial_y + \tfrac{21}{4}xy^2\partial_x + \tfrac{21}{4}y^3\partial_y, 12y^4s + 21x^3s - \tfrac{9}{7}xy^4\partial_x - \tfrac{12}{7}y^5\partial_y - 3x^4\partial_x - 3x^3y\partial_y - \tfrac{28}{3}x^4s - \tfrac{49}{3}y^3s + \tfrac{4}{3}x^5\partial_x + x^4y\partial_y + \tfrac{7}{3}xy^3\partial_x + \tfrac{7}{3}y^4\partial_y - \tfrac{28}{3}x^3ys + \tfrac{343}{12}x^2s + \tfrac{4}{3}x^4y\partial_x + x^3y^2\partial_y + \tfrac{7}{12}y^4\partial_x - \tfrac{49}{12}x^3\partial_x - \tfrac{49}{12}x^2y\partial_y, \tfrac{768}{49}xys^2 - 48s^2 - \tfrac{192}{49}x^2y\partial_x s - \tfrac{192}{49}xy^2\partial_y s + \tfrac{96}{7}x\partial_x s + \tfrac{96}{7}y\partial_y s - \tfrac{48}{7}s + \tfrac{576}{2401}x^3y\partial_x^2 + \tfrac{1200}{2401}x^2y^2\partial_x\partial_y + \tfrac{576}{2401}xy^3\partial_y^2 + \tfrac{576}{2401}x^2y\partial_x - \tfrac{48}{49}x^2\partial_x^2 + \tfrac{576}{2401}xy^2\partial_y - \tfrac{96}{49}xy\partial_x\partial_y - \tfrac{48}{49}y^2\partial_y^2, 4x^4y^3\partial_x - 4x^3y^4\partial_y + 7y^6\partial_x - 7x^6\partial_y\rangle.$$

3.2. *Their b-functions*

The timing information in this section is taken from a Pentium III, 1 GHz.

As far as we know, none of the available implementations of Oaku's algorithm can manage the following two examples. However, their b-functions can be obtained in **kan/sm1** using the PBW algebra $R_{n,p}$ and the BM algorithm.

Equation	b-function	Running time (s)
$xyz(x+y)(x+z)$	$(5s+4)(5s+3)(3s+4)(5s+7)(5s+6)(3s+2)(s+1)^3$	13
$x^7+y^7+x^4y^4$	$(7s+10)(7s+9)(7s+8)(7s+4)(7s+6)(7s+2)(7s+5)$ $(7s+3)(s+1)^2$	504

3.3. *Towards a mathematical explanation: a paradigmatic example*

There is a whole family of examples that does not appear in the table of the last section: the curves[1] $x^a + y^b + xy^{b-1}$, with $b \geq a + 1 \geq 5$: the *Reiffen family* (Reiffen, 1972). These examples defeated our prototype with the Oaku–Takayama method but were easily managed by the Briançon–Maisonobe method. In the case $a = 4, b = 5$, for example, the system took 139.51 s of CPU time for a basis with ten elements; maximum number of monomials = 107.

[1] And surfaces obtained from constructions over the curves.

The annihilators corresponding to these curves have been widely studied with the use of **Plural**. The results are summarized in the next table for the case $b = 5, a = 4$:

f	CPU time		N.E.		N.M.		T.D.	
	OT	BM	OT	BM	OT	BM	OT	BM
$x^4 + y^5 + xy^4$	94 min	<1 s	33	9	1240	71	15	10

The calculations in the first case have been done with respect to a typical elimination ordering: first give weight 1 to the variables u, v and to break ties use degree reverse lexicographical ordering. In the second case the ordering for BM was simply a lexicographical ordering with s, t greater than the others.

The explanation of the enormous difference between the two methods in this example is not as easy as considering the number of variables, 8 and 6 respectively. More precisely:

- The bounds of Grigoriev (see Grigoriev, 1991) for the calculations in the Weyl algebra are applicable to $R_{n,p}$ but neither of the two calculations is a worst case (double exponential) in the sense of total degree, the usual measurement of the complexity for Gröbner bases. Of course for the general case (more than one function) the difference between $2n + 2p$ and $2n + 4p$ becomes significant.
- Nevertheless, the number of variables is important from the point of view of the *number of monomials*. In this non-commutative setting the ingredients of the Buchberger algorithm—S-polynomials and reductions—produce many more monomials than in the commutative case. As the number of possible monomials of total degree $\leq d$ in n variables is

$$\binom{n+d}{d}$$

the comparison of the two methods relies on the ratio

$$\binom{2n+2p+d}{d} \Big/ \binom{2n+4p+d}{d}.$$

If you reach, say, total degree 15, you could have elements of about

$$\binom{15+7}{7}$$

monomials in the algebra of Oaku and Takayama, but

$$\binom{15+5}{5}$$

in $R_{2,1}$. And you have to consider the total degrees not only in the final result but also in the intermediate calculations! This consideration influences the duration of the calculation as well as the total amount of memory used.
- Another important factor is the growth of the coefficients of the monomials. This is a very well known problem in the commutative setting but is a more difficult matter in the Weyl algebra or in $R_{n,p}$, because of the binomial coefficients appearing in

the Leibnitz rule that is repeatedly applied. Coefficients of more than 30 digits are obtained in the example that we are studying in the case of the Oaku–Takayama method. A lot of computation becomes much slower due to these coefficients.

Remark. The ordering selected seems to be the best option for each case. It is a little surprising that the fastest option in the commutative case, elimination orderings like those used in the OT method, defeated the calculations for the BM method.

Remark. The annihilator of $(x^a + y^b + xy^{b-1})^s$ defeats **kan/sm1** and **Plural** for $b \geq a + 1 > 12$. It seems that this is a really hard example!

3.4. Bernstein–Sato ideals

In this section we compare the OT method to the BM method for the computation of Bernstein–Sato ideals for $p = 2$. We present two examples in **kan/sm1**.

- Take $\{f_1 = x^3 + y^2,\ f_2 = y^3 + x^2\}$, two transverse cuspids. The table of running times in the computation of $\mathcal{B}_\Sigma$ is

Method	Time for the step 1 (s)	Time for the step 2 (s)	Total (s)
OT	2.4	0.02	2.42
BM	0.03	0.07	0.1

Steps 1 and 2 are the successive eliminations in each method as explained in 2.3. The Bernstein–Sato ideal $\mathcal{B}_\Sigma$ is as follows:

$$\mathcal{B}_\Sigma = \langle s_2 + 1, s_1 + 1\rangle \cap \langle g\rangle,$$

where

$$g = (4s_1 + 6s_2 + 5)(6s_1 + 4s_2 + 5)(6s_1 + 4s_2 + 7)(4s_1 + 6s_2 + 7).$$

The calculation of the ideal $\mathcal{B}$ in this case is rather hard. At the moment it seems to be intractable with any method. It was first proposed in Bahloul (2001).

- Take $\{f_1, f_2\} = \{x^2 + y^2(1 + y), y^3 + x^2\}$.

Method	Time for the step 1 (s)	Time for the step 2 (s)	Total (s)
OT	16.4	0.02	16.42
BM	1.78	0.06	1.84

In this case we have

$$\mathcal{B}_\Sigma = \langle s_2 + 1, s_1 + 1\rangle \cap \langle g\rangle,$$

where

$$g = (s_1 + s_2 + 1)(2s_1 + 2s_2 + 3)(4s_1 + 6s_2 + 5)(4s_1 + 6s_2 + 7).$$

4. Conclusions and challenges

We have tried to explain how the different number of variables required by the Oaku–Takayama and Briançon–Maisonobe methods, respectively $2n + 4p$ and $2n + 2p$, produce very different effects in the calculations of annihilators and, hence, of the Bernstein–Sato ideals. As we have mentioned, in the non-commutative setting the role of the number of variables has a more intense influence in the Buchberger algorithm. A more complete explanation of the apparently much lower complexity of the BM method is beyond the scope of this work.

From the point of view of the limits of the available methods and systems, the calculation of $\mathcal{B}$ for two cuspids remains open. We hope that an optimized implementation of the BM method will solve this problem. Then, perhaps, the next step would be the calculation of the Bernstein–Sato ideals for two transverse Reiffen curves and other families of non-quasi-homogeneous plane curves.

A rich source of hard examples is the hyperplane arrangements. The calculation of the corresponding annihilators is beyond the current limits of computation, already in dimension four with, say, a dozen hyperplanes.

Acknowledgements

We are deeply grateful to Professor Nobuki Takayama and Viktor Levandovskyy for very useful suggestions and especially for the implementation of the BM method in **kan/sm1** and **Plural**, as well as the computation of the examples. We also thank the referees for a very careful reading of the manuscript and for their helpful suggestions. Both authors were partially supported by DGESIC BFM-2001-3164 and FQM-333.

References

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

Bernstein, I.N., 1972. The analytic continuation of generalized functions with respect to a parameter. Funct. Anal. 6, 273–285.

Bjork, J.E., 1979. Rings of Differential Operators. North-Holland, Amsterdam.

Briançon, J., Maisonobe, Ph., 2002. Remarques sur l'idéal de Bernstein associé à des polynômes (preprint).

Briançon, J., Granger, M., Maisonobe, Ph., Miniconi, M., 1989. Algorithme de calcul du polynôme de Bernstein: cas non dégénéré. Ann. Inst. Fourier (Grenoble) 39, 553–610.

Bueso, J.L., Castro, F.J., Gómez-Torrecillas, J., Lobillo, F.J., 1998. An introduction to effective calculus in quantum groups. In: Rings, Hopf Algebras and Brauer Groups. Lecture Notes in Pure and Applied Mathematics, vol. 197. Marcel-Dekker, New York, pp. 55–83.

Castro-Jiménez, F.J., Narváez-Macarro, L., 1997. Homogenising differential operators. Prepub. n. 36, Dpto. Álgebra de la Universidad de Sevilla. Junio 1997. Available from http://www.us.es/da/prepubli/prepub36.pdf.

Greuel, G.-M., Pfister, G., Schönemann, H., 2001. Singular 2.0. A Computer Algebra System for Polynomial Computations. Centre for Computer Algebra, University of Kaiserslautern. Available from http://www.singular.uni-kl.de.

Grigoriev, D., 1991. Complexity of solving systems of linear equations over the rings of differential operators. In: Effective Methods in Algebraic Geometry (Castiglioncello, 1990), Progr. Math., vol. 94. Birkhauser, Boston, MA, pp. 195–202.

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

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

Levandovskyy, V., 2002. On Gröbner bases for noncommutative G-algebras. In: Kredel, H., Seiler, W.K. (Eds.), Proceedings of the 8th Rhine Workshop on Computer Algebra. Mannheim, Germany.

Malgrange, B., 1975. Le polynôme de Bernstein d'une singularité isolée. In: Fourier Integral Operators and Partial Differential Equations (Colloq. Internat., Univ. Nice, Nice, 1974), Lecture Notes in Mathematics, vol. 459. Springer-Verlag, New York, pp. 98–119.

Maynadier, H., 1997. Polynômes de Bernstein–Sato associées à une intersection complète quasi-homogène à singularité isolée. Bull. Soc. Math. France 125, 547–571.

Oaku, T., 1997. An algorithm of computing b-functions. Duke Math. J. 87, 115–132.

Oaku, T., Takayama, N., 1999. An algorithm for de Rham cohomology groups of the complement of an affine variety via D-module computation. J. Pure Appl. Algebra 139, 201–233.

Reiffen, H.J., 1972. Das Lemma von Poincaré für holomorphe Differential-formen auf komplexen Räumen (German). Math. Z. 101, 269–284.

Sabbah, C., 1987. Proximité évanescente II. Equations fonctionelles pour plusieurs fonctions analytiques. Compositio Math. 64, 213–241.

Takayama, N., 1991. Kan: a system for computation in algebraic analysis. Source code available for Unix computers from ftp://ftp.math.kobe-u.ac.jp.

Walther, U., 2002. Bernstein–Sato polynomial versus cohomology of the Milnor fiber for generic arrangements. Available from arXiv:math.AG/0204080.