

Risa/Asir における Weyl Algebra 上のグレブナ基底 計算およびその応用

神戸大学 野呂 正行(Masayuki NORO) *

1 Weyl Algebra

さまざまな計算機代数システム上で Weyl Algebra に関する演算が実装されている。代表的なものとして, Kan/sm1 [9], Macaulay2 [1][3], Maple Ore algebra package, Singular [2] などがある。以下では Risa/Asir における Weyl Algebra 関連機能の実装について述べるが, ここで述べられている改良その他は, 文献として参照することはできないものの, 上記システムそれぞれにおいて採り入れられていると考えられる。

1.1 Leibnitz rule

体 K 上の n 次元 Weyl Algebra

$$D_n = K\langle x_1, \dots, x_n, \partial_1, \dots, \partial_n \rangle$$

における積の計算は, よく知られているように Leibnitz rule で計算できる。すなわち, $2n$ 変数多項式環 $K[x, \xi] = K[x_1, \dots, x_n, \xi_1, \dots, \xi_n]$ から D_n への K 線形写像 Ψ を $\Psi(x^\alpha \xi^\beta) = x^\alpha \partial^\beta$ で定義するとき,

$$\Psi(f)\Psi(g) = \Psi\left(\sum_{k_1, \dots, k_n \geq 0} \frac{1}{k_1! \cdots k_n!} \frac{\partial^{k_1} f}{\partial \xi^{k_1}} \frac{\partial^{k_2} g}{\partial x^{k_2}}\right)$$

$(f, g \in K[x, \xi])$ となる。Weyl Algebra における計算の効率化のためには積の効率化が重要であり, そのためには Leibnitz rule をどう使うかが鍵となる。特に, グレブナ基底計算を考える場合, monomial \times polynomial の高速化が必要である。

*norou@math.kobe-u.ac.jp

1.2 Asir における実装

Asir における初期の実装では、これを

$$m : \text{monomial}, f = \sum f_j (f_j : \text{monomial}) \text{ に対し } mf = \sum mf_j$$

として計算していた. ここで monomial どうしの積は

$$x^k D^l \cdot x^a D^b = (x_1^{k_1} (D_1^{l_1} x_1^{a_1}) D_1^{b_1}) \cdots (x_n^{k_n} (D_n^{l_n} x_n^{a_n}) D_n^{b_n})$$

として, 右辺の $D_i^{k_i} x_i^{a_i}$ を Leibnitz rule により

$$D_i^{l_i} x_i^{a_i} = \sum_{j=0}^{\text{Min}(l_i, a_i)} j! \binom{l_i}{j} \binom{a_i}{j} x_i^{a_i-j} D_i^{l_i-j}$$

とし, これらの積を, 通常の可換な多項式として展開する, という方法を取っていた. しかし, この monomial の積自体が多項式となるため, これらを足し合わせる際に項の比較およびリストのつなぎ換えが多数生じ, 特に f の項数が多い場合に効率が悪いことが分かった.

1.3 改良

Leibnitz rule に現れる微分を $f \in K[x, \xi]$ に作用させても monomial の順序は変わらないことに注意すれば, 左からかける monomial m を固定したとき, 前項の monomial どうしの積において

1. $D_i^{l_i} x_i^{a_i}$ を変形して得られる和を常に $\sum_{j=0}^{l_i}$ とする. $l_i > \text{Min}(l_i, a_i)$ の場合, 係数は 0 としておく.
2. 適当な順序を定めて, 積を構成する monomial を整列し, 対応する位置の monomial どうしの和をまず作る. これは既に整列されている f_j と同順でよい.
3. 最後にそれらを足し合わせる.

要するに, Leibnitz rule において, 微分に関する和を先に計算するということだが, これで m の ∂ に関する次数がよほど高くない限り効率化する. (これは, Kan/sml で既に行われていた.) さらに, 次の改良が考えられる.

- $(x_1^{k_1} (D_1^{l_1} x_1^{a_1}) D_1^{b_1}) \cdots (x_n^{k_n} (D_n^{l_n} x_n^{a_n}) D_n^{b_n})$ の計算を incremental に行う.

これは $\frac{1}{k_1! \cdots k_n!}$ に現れる共通の部分積を重複して計算しないということであり, 次数が高い場合に有効かもしれない.

- l_i, a_i が 0 の場合の shortcut.

一般に次数が 0 になる部分が多いでこれは重要である.

- exponent vector, monomial などを表す構造体のメモリ管理を自前で行う。

これは, Weyl に限らず, 有限体上の場合には一般に用いている. これも効率に大きく影響する.

以上の改良を加えることで, 初期実装に比較して十分効率化することができた.

1.4 Weyl Algebra における Buchberger アルゴリズム

多項式環上の Buchberger アルゴリズム実装においては, 種々の criteria および selection strategy が考案され実用化されているが, Weyl Algebra においても Buchberger's criterion (S-polynomial を構成する多項式の頭項の GCD が 1 の場合にはその S-polynomial は 0 に正規化される) 以外はそのまま使える. よって, あとは積を前節のものに換えれば, 多項式環用のドライバ, サブルーチンが使える, 容易に Weyl Algebra 上の Buchberger アルゴリズムの実装が得られる.

2 b-function の計算

2.1 b-function

$D = D_n = K\langle x_1, \dots, x_n, \partial_1, \dots, \partial_n \rangle$ ($\partial_i = \partial/\partial x_i$), $f \in K[x_1, \dots, x_n]$ とするとき $P(s)f^{s+1} = b_f(s)f^s$ なる $P(s) \in D[s]$ が存在するような最小次数の $b_f(s) \in K[s]$ を f の (global) b -function と呼ぶ¹⁾. 大阿久 [5] により b -function は以下のように計算できる. まず, $n+1$ 次元の Weyl Algebra $D_{n+1} = K\langle t, x_1, \dots, x_n, \partial_t, \partial_1, \dots, \partial_n \rangle$ を考える.

定義 1 D_{n+1} の元 $P = \sum c_{\mu\nu\alpha\beta} t^\mu x^\alpha \partial_t^\nu \partial^\beta$ に対し,

$$\text{ord}_F(P) := \max\{\nu - \mu \mid \text{ある } \alpha, \beta \text{ に対し } c_{\mu\nu\alpha\beta} \neq 0\}$$

$$\hat{\sigma}(P) := \sum_{\nu - \mu = \text{ord}_F(P)} c_{\mu\nu\alpha\beta} t^\mu x^\alpha \partial_t^\nu \partial^\beta$$

定義 2 $\text{ord}_F(P) = m$ なる $P \in D_{n+1}$ に対し, $\psi(P) \in D[s]$ を

$$\psi(P)(t\partial_t) = \begin{cases} \hat{\sigma}(t^m P) & (m \geq 0) \\ \hat{\sigma}(\partial_t^{-m} P) & (m < 0) \end{cases}$$

で定義する.

¹⁾本来は $\tilde{b}(s)$ と書くべきだが, 本稿では $b(s)$ と書くことにする.

定理 3

$$I = \text{Id}(t - y_1 f, \partial_1 + y_1(\partial f/\partial x_1)\partial_t, \dots, \partial_n + y_1(\partial f/\partial x_n)\partial_t)$$

に対し, G_1 を $I_1 = I \cap D$ のグレブナ基底とする. この時,

$$\text{Id}(\psi(G_1)) \cap K[s] = \text{Id}(b(-s-1))$$

大阿久 [5] では

$$(\text{Id}(\psi(G_1)) \cap K[x, s]) \cap K[s]$$

なる二段階の elimination による計算を提案している. これは, local な b-function の計算に $(\text{Id}(\psi(G_1)) \cap K[x, s])$ が必要となるためと考えられるが, global b-function のみを求める場合には, 直接 $K[s]$ との交わりを求めて構わない. よって, 任意の順序に関する $\text{Id}(\psi(G_1))$ のグレブナ基底が求まっていれば, $b(s)$ は可換多項式環の場合と同様に, 未定係数法より求めることができる.

2.2 \mathbb{Q} 上の Weyl Algebra における最小多項式の modular 計算

D を \mathbb{Q} 上の Weyl Algebra, J を D の ideal, $P \in D$ かつ P は整数係数とし, $J \cap \mathbb{Q}[P] \neq \{0\}$ とする. この時, $J \cap \mathbb{Q}[P] = \text{Id}(b(P))$ とすれば $b(s)$ は D/J における P の \mathbb{Q} 上の最小多項式となる. ここで, $b(s) \in \mathbb{Z}[s]$ かつ \mathbb{Z} 上原始的と取れる.

J の, 順序 $<$ に関するグレブナ基底で, 各元の頭係数が 1 であるものを G とし, G の各元の係数が $\mathbb{Z}_{(p)} = \{a/b \mid a \in \mathbb{Z}, b \notin p\mathbb{Z}\}$ に属するような p を選ぶ. ϕ_p を $\mathbb{Z}_{(p)}$ から $GF(p)$ への標準的射影 (およびその D への拡張) とする.

補題 4 $\phi_p(G)$ は $\text{Id}(\phi_p(G))$ の $<$ に関するグレブナ基底で, $\phi_p(b(P)) \in \text{Id}(\phi_p(G))$.

証明 G から作られる S-polynomial の G による 0 への reduction が $\mathbb{Z}_{(p)}$ 上で行える. よって, その ϕ_p による像がそのまま $\phi_p(G)$ による reduction になるから, $\phi_p(G)$ は $\text{Id}(\phi_p(G))$ のグレブナ基底となる. 後半も同様である. ■

仮定により $\phi_p(b(s))$ は 0 でないから, $\phi_p(P)$ の $\phi_p(D)/\text{Id}(\phi_p(G))$ における最小多項式 $b_p(s)$ が存在して $b_p(s) \mid \phi_p(b(s))$ が成り立つ. 特に $\deg(b_p(s)) \leq \deg(b(s))$.

定理 5 $b_p(s)$ を $\phi_p(D)/\text{Id}(\phi_p(G))$ における $\phi_p(P)$ の最小多項式とするとき, $\deg(f(s)) = \deg(b_p(s))$, $f(P) \in \text{Id}(G)$ なる $f \in \mathbb{Z}[s]$ が存在すれば, $f(s) = b(s)$.

証明 仮定より $b(s) \mid f(s)$. 一方で $\deg(f(s)) = \deg(b_p(s)) \leq \deg(b(s))$ だから $b(s) = f(s)$. ■

アルゴリズム 6

Input: 項順序 \prec, \prec に関するグレブナ基底 $G \subset D$ (頭係数が 1),

$Id(G) \cap \mathbb{Q}[P] \neq \{0\}$ なる $P \in D$ (整数係数)

Output: $Id(G) \cap \mathbb{Q}[P] = Id(b(P))$ なる $b(s) \in \mathbb{Q}[s]$

do {

$p \leftarrow G \in \mathbb{Z}_{(p)}$ なる未使用の素数

$b_p \leftarrow \phi_p(P)$ の $\phi_p(D)/Id(\phi_p(G))$ における最小多項式

$d \leftarrow deg(b_p)$

$c \leftarrow NF(P^d, G) + \sum_{i=0}^{d-1} c_i NF(P^i, G)$ (c_i は未定係数)

$C \leftarrow \{C_t(c_0, \dots, c_{d-1}) = 0 \mid C_t \text{ は } c \text{ の各 monomial の係数}\}$

if C が解 $\{c_0 = a_0, \dots, c_{d-1} = a_{d-1}\}$ を持つ {

return $s^d + \sum_{i=0}^{d-1} a_i s^i$

}

}

法 p での最小多項式は, $NF(\phi_p(P^k), Id(\phi_p(G)))$ を順に計算して線形関係を探ること得られる. $b(s) \in \mathbb{Z}[s]$ とした時の主係数を割らない p に対しては定理 5 の f は存在するから, アルゴリズム 6 は停止する. また, C が解を持つなら, 法 p での一意性よりそれは一意的に決まる. C の求解は, 法 p での一意性を利用して, Hensel 構成により効率よく求めることができる. 詳細は [4] を参照.

2.3 ホロノミックな ideal に対する b -function

ideal $I \subset D_n, w \in \mathbb{R}^n \setminus \{0\}$ に対し, weight $(-w, w)$ に関する initial ideal $in_{(-w, w)}(I)$ が, Weyl Algebra におけるグレブナ基底計算により求まる ([8] Theorem 1.1.6). I の characteristic variety の次元が n であるとき I はホロノミックと呼ばれるが, このとき [8] Theorem 5.1.2 により

$$in_{(-w, w)}(I) \cap K[s] = Id(b(s))$$

($s = w_1\theta_1 + \dots + w_n\theta_n, \theta_i = x_i\partial_i$) なる 0 でない多項式 $b(s)$ が存在する. この $b(s)$ を, I の (global) b -function と呼ぶ. 特に, 多項式 f に対し,

$$I_f = Id(t - f, \partial_1 + (\partial f / \partial x_1)\partial_t, \dots, \partial_n + (\partial f / \partial x_n)\partial_t)$$

とおく時, I_f はホロノミックで, t を先頭の変数とする時 I_f の $w = (1, 0, \dots, 0)$ に関する global b -function を $B(s)$ とすれば, $b_f(s) = B(-s - 1)$ が成り立つ. よって, $D/in_{(-w, w)}(I_f)$

における $t\partial_t$ の最小多項式から $b_f(s)$ を求めることもできる. ここで, $in_{(-w,w)}(I_f)$ の計算には non-term order によるグレブナ基底計算が必要となるが, [8] で述べられている斉次化によりある term order のもとのグレブナ基底計算に帰着できる.

2.4 タイミングデータ

Section 2 で紹介した, 二段階の消去による方法 (方法 1) と, $Id(\psi(G_1)) \cap K[s]$ を Section 2.2 で述べたように最小多項式として求める方法 (方法 2), および Section 2.3 で述べた方法 (方法 3) による計算時間をさまざまな多項式に対して比較する. いずれも, b -function の計算は modular 計算で最小多項式求める方法により行った. 例題は [5], [10] から採った. 表 2 の $x^a + xy^{b-1} + y^b$ に関しては, 本講究録中の大阿久, 高山両氏による稿を参照. 計算は, PentiumIII 1GHz 上で行った. 単位は秒でガーベッジコレクション時間は除いてある. “-” は, 他の方法と比較して時間がかかり過ぎるため中断したことを意味する. 表 1 では計算が比較的容易なものが扱われており, 各方法で大差はないが, 表 2 では, 方法 1 では計算が困難なものが, 方法 2, 3 により計算できている. また, 次数が上がるにつれ, 方法 3 が優位となる. これは, 方法 2 に現れるグレブナ基底計算が, 方法 3 における $in_{(-w,w)}(I_f)$ の計算に比べて困難となるためである. これは表 3 においてさらに顕著となり, 方法 2 では計算できない例も出て来る. 実際には, 方法 3 では, 最小多項式の計算が dominant となる例もしばしば見られる.

	$\deg(b(s))$	方法 1	方法 2	方法 3
$x^5 + x^3y^3 + y^5$	7	1.4	2.4	1.3
$x^3 + y^3 + z^3 + x^2y^2z^2 + xyz$	5	3.5	3.9	10
$x^6 + y^4 + z^3$	18	0.4	0.7	0.2
$x^4 + y^4 + z^3 + xyz$	8	0.3	0.6	0.3
$(x^3 - y^2z^2)^2$	14	0.7	1.5	0.2
$y(x^5 - y^2z^2)$	18	0.8	0.5	0.1
$y((y+1)x^3 - y^2z^2)$	11	0.6	1.1	0.7

表 3: [5] からの例題

3 おわりに

Risa/Asir における, Weyl Algebra 関連機能の実装および, その応用として b -function の計算方法の改良について述べた. b -function 計算は Kan/sml, Macaulay 2 にも実装されているが, 本稿で述べたような, 最小多項式を未定係数法で求める方法を用いた例はないようである.

	$\deg(b(s))$	方法 1	方法 2	方法 3
$x^4 + xy^4 + y^5$	13	135	5.0	3.4
$x^4 + xy^5 + y^6$	13	15	2.9	2.1
$x^4 + xy^6 + y^7$	19	719	5.1	6.9
$x^4 + xy^7 + y^8$	11	19	7.3	4.0
$x^4 + xy^8 + y^9$	25	–	14	11
$x^4 + xy^9 + y^{10}$	23	–	22	14
$x^5 + xy^5 + y^6$	21	–	158	21
$x^5 + xy^6 + y^7$	25	–	32	19

表 4: [10] からの例題

	$\deg(b(s))$	方法 2	方法 3
$x^6 + y^{12}z^8$	50	71	1.8
$x_1x_2^3 + x_3x_4^5 + x_5x_6^7$	106	404	14
$(x_1x_2)^3 + (x_3x_4)^3 + (x_5x_6)^3$	23	–	27
$(x_1x_2)^2 + (x_3x_4)^2 + (x_5x_6)^2 + (x_7x_8)^2$	16	–	61

表 5: [10] からの例題 (non-isolated singularities)

る. 一方で b -function は f の局所モノドロミーと関係することが知られているが, Singular においては, 全く異なる立場から isolated singularity でのモノドロミー行列を求める機能を提供している. これについて, 効率の面からの比較も必要と考えられるが, 得られる結果が異なることもありまだ詳細な比較は行っていない.

本稿で述べた方法により, より広い範囲の多項式およびイデアルに対して b -function が計算できるようになったことは確かである. しかし, 既に他の方法で結果が知られているものでも計算不可能な問題は存在し, またいわゆる多重 b -function に対しては, 最小多項式による方法は無力である. これらに対処するためにはさらなる改良, あるいは新しい方法が必要であろう.

参 考 文 献

- [1] Grayson, D., Stillman, M.: Macaulay 2, a software system for research in algebraic geometry. <http://www.math.ucuc.edu/Macaulay2>.
- [2] Greuel, G.-M., Pfister, G., Schönemann, H.: SINGULAR, A Computer Algebra System for Polynomial Computations. <http://www.singular.uni-kl.de/>.

- [3] Leykin, A., Tsai, H.: *D*-module package for Macaulay 2.
<http://www.math.cornell.edu/~tsai>.
- [4] Noro, M., Yokoyama, K.: A Modular Method to Compute the Rational Univariate Representation of Zero-Dimensional Ideals. *J. Symb. Comp.*, **28**,1 (1999) 243–263.
- [5] Oaku, T.: Algorithm for the *b*-function and *D*-modules associated to a polynomial. *J. Pure Appl. Algebra*, **117** & **118** (1997) 495–518.
- [6] Oaku, T., Takayama, N.: An algorithm for de Rham cohomology groups of the complement of an affine variety via *D*-module computation. *J. Pure Appl. Algebra*, **139** (1999), 201–233.
- [7] Oaku, T., Takayama, N.: Algorithms for *D*-modules —restriction, tensor product, localization, and local cohomology groups. *J. Pure Appl. Algebra* (in press).
- [8] Saito, M., Sturmfels, B., Takayama, N.: Gröbner Deformations of Hypergeometric Differential Equations. *Algorithms and Computation in Mathematics* **6**, Springer (2000).
- [9] Takayama, N.: Kan — A system for doing algebraic analysis by computer.
<http://www.math.kobe-u.ac.jp/KAN>.
- [10] Yano, T.: On the theory of *b*-functions. *Publ. RIMS Kyoto Univ.* **14** (1978), 111–202.