数理解析研究所講究録
第 1666 巻 2009 年 106-116

局所 b 関数に付随する stratification アルゴリズムの実装および応用

西山 絢太
KENTA NISHIYAMA
神戸大学大学院理学研究科*

野呂 正行
MASAYUKI NORO
神戸大学大学院理学研究科†

Abstract

n 変数多項式 $f \in \mathbb{C}[x]$ の local b 関数 $b_{f,p}(s)$ は各 stratum で一定であるという性質で $\mathbb{C}^n$ を stratify する. Oaku [12] は準素イデアル分解を用いて, このような stratification と各 stratum の上の $b_{f,p}(s)$ を計算するアルゴリズムを提案した. 我々は準素イデアル分解を用いない新たなアルゴリズムを提案し, その実装および応用について詳しく述べる.

1 はじめに

$\mathcal{D}$ を n 変数 $x = (x_1, \dots, x_n)$ の微分作用素環とするとき, 多項式 $f(x) \in \mathbb{C}[x] = \mathbb{C}[x_1, \dots, x_n]$ の b 関数は,

$$L(x, \partial_x, s) f^{s+1} = b(s) f^s$$

を満たす微分作用素 $L \in \mathcal{D}[s]$, 1 変数多項式 $b(s)$ のペアに対し, $b(s)$ の次数が最小のものとして定義される. ここで, $\mathcal{D}$ が Weyl algebra $D = \mathbb{C}\langle x, \partial_x \rangle$ の場合は global b 関数と呼び $b_f(s)$ と書く. また, $\mathcal{D}$ が,

$$\mathbb{C}[x]_p = \{f(x)/g(x) \mid f(x), g(x) \in \mathbb{C}[x], g(p) \neq 0\}$$

(p における局所化) を係数とする微分作用素環 $\mathbb{C}[x]_p\langle \partial_x \rangle$ の場合, p における local b 関数と呼び, $b_{f,p}(s)$ と書く. 定義から直ちに, 任意の p に対し $b_{f,p}(s) | b_f(s)$ であり, $b_f(s) = \mathrm{LCM}(b_{f,p}(s); p \in \mathbb{C}^n)$ が成り立つことが分かる.

2 b 関数の計算アルゴリズム

この章では, 既に知られている b 関数の計算アルゴリズムについて簡単に述べる.

global b 関数の計算は $D[s]$ 上で $\mathrm{Ann}_{D[s]} f^s := \{P \in D[s] \mid P f^s = 0\}$ を計算し, $(\mathrm{Ann}_{D[s]} f^s + D[s] f) \cap \mathbb{C}[s]$ の生成元として $b_f(s)$ を計算する方法 [13] と, 新たな変数 t を追加した $n+1$ 変数の Weyl algebra $D\langle t, \partial_t \rangle$ 上で $I_f := \langle t - f, \partial_1 + \frac{\partial f}{\partial x_1}\partial_t, \dots \partial_n + \frac{\partial f}{\partial x_n}\partial_t \rangle$ の $(t, x_1, \dots, x_n \mid \partial_t, \partial_1, \dots, \partial_n)$ に対する weight $(-w, w) := (-1, 0, \dots, 0 \mid 1, 0, \dots, 0)$ に関する先頭項イデアル $\mathrm{in}_{(-w,w)}(I_f)$ を計算し, $\mathrm{in}_{(-w,w)}(I_f) \cap \mathbb{C}[t\partial_t]$ の生成元として $b_f(-t\partial_t - 1)$ を計算する方法 [16] の 2 つの方法がある. これらの方法は次の定理による密接な関係がある.

*nisiyama@math.kobe-u.ac.jp
†noro@math.kobe-u.ac.jp

定理 1 ([15] 定理 5.3.4.)
$D[s]$ における f^s の零化イデアル $\mathrm{Ann}_{D[s]}f^s$ は $I_f \cap D[t\partial_t]$ において $t\partial_t \mapsto -s-1$ と変換したものに等しい.

与えられた1点における local b 関数を計算するアルゴリズムは Oaku [11] による方法と Nakayama [7] による Mora division を用いる方法, 近似割り算を用いる方法の合わせて3種類が知られているが, ここでは深く言及しない. 一方, local b 関数 $b_{f,p}(s)$ を $p \in \mathbb{C}$ の関数として与える方法は Oaku [12] により与えられた. それは次のようなアルゴリズムである.

アルゴリズム 1

$L_f \leftarrow (\mathrm{Ann}_{D[s]}f^s + D[s]f) \cap \mathbb{C}[x,s]$
L_f の準素イデアル分解 $L_f = Q_1 \cap \cdots \cap Q_r$ を計算
$V_i \leftarrow V(Q_i \cap \mathbb{C}[x])$ $(i=1,\ldots,r)$
$b_i(s) \leftarrow Q_i \cap \mathbb{C}[s]$ $(i=1,\ldots,r)$
$S \leftarrow \emptyset$
for each $Z \subset \{1,\ldots,r\}$ do
  $V_Z \leftarrow \bigcap_{i\in Z} V_i \setminus \bigcup_{i\notin Z} V_i$
  if $V_Z \neq \emptyset$ then
    $b_Z(s) \leftarrow \mathrm{LCM}(b_i(s); i \in Z)$
    $S \leftarrow S \cup \{(V_Z, b_Z(s))\}$
  end if
end for
return S

このアルゴリズムは $(V_Z, b_Z(s))$ というペアの集合を返し, 点 $p \in V_Z$ に対して $b_{f,p}(s) = b_Z(s)$ であることを意味する. また, 各 V_Z は $\mathbb{C}^n$ の構成可能集合になっている. この方法には次のようなボトルネックになり得る点がある.

- L_f の計算
  $\mathrm{Ann}_{D[s]}f^s + D[s]f$ を計算し, その消去イデアルとして L_f を計算しているが, 定理1により $\mathrm{in}_{(-w,w)}(I_f)$ の計算をしてから消去イデアルの計算をしてもよい. 実際に $\mathrm{Ann}_{D[s]}f^s + D[s]f$ の計算より $\mathrm{in}_{(-w,w)}(I_f)$ の計算の方が速く計算できることが多かったことから後者を採用した方が効率面で有利であると思われる.

- 準素イデアル分解
  最終的に点 $p \in \mathbb{C}$ における local b 関数 $b_{f,p}(s)$ を定めるためには p が $V(Q_i \cap \mathbb{C}[x])$ に属するかどうかを調べる必要がある. これには零点の情報があれば十分であるが, 準素分解は零点の情報より多くの情報を持っている. 実際に, 準素分解による方法は, 同一の local b 関数を与える成分を更に分解した結果を出力する場合がある. また, 準素分解はアルゴリズム, 実装ともに複雑であり, 入力によっては計算が困難になる場合もある.

3 新しい計算アルゴリズム

以下では $J_f = \mathrm{in}_{(-w,w)}(I_f)$ とする.

定理 2
$p \in \mathbb{C}^n$ と $b(s) \in \mathbb{C}[s]$ に対して以下は同値.

1. $Lf^{s+1} = b(s)f^s$ を満たすような $L \in \mathbb{C}[x]_p\langle\partial_x\rangle$ が存在する.

2. $a(x) \in (J_f \cap \mathbb{C}[x, t\partial_t]) : b(-t\partial_t - 1))$, $a(p) \neq 0$ を満たす $a(x) \in \mathbb{C}[x]$ が存在する.

証明 $L(s, x, \partial_x) \in \mathbb{C}[x]_p\langle\partial_x\rangle$ から $L = 1/a(x) \cdot \tilde{L}$ を満たす $\tilde{L}(s, x, \partial_x) \in D$ と $a(x) \in \mathbb{C}[x]$ $(a(p) \neq 0)$ が存在して,

$$\begin{aligned}
1/a(x) \cdot \tilde{L}(s, x, \partial_x)f^{s+1} = b(s)f^s \quad &\Leftrightarrow \quad \tilde{L}(s, x, \partial_x)f^{s+1} = a(x)b(s)f^s \\
&\Leftrightarrow \quad \tilde{L}(s, x, \partial_x)f - a(x)b(s) \in \mathrm{Ann}_{D[s]}f^s \\
&\Leftrightarrow \quad \tilde{L}(-t\partial_t - 1, x, \partial_x)t - a(x)b(-t\partial_t - 1) \in I_f \\
&\Leftrightarrow \quad a(x)b(-t\partial_t - 1) \in \mathrm{in}_{(-w,w)}(I_f) \\
&\Leftrightarrow \quad a(x) \in \mathrm{in}_{(-w,w)}(I_f) : b(-t\partial_t - 1).
\end{aligned}$$

3 行目の同値は定理 1 と $t \equiv f \bmod I_f$ から, 4 行目の同値は $(-w, w)$ が t, ∂_t にのみ $-1, 1$ という重みが付いた weight であることからそれぞれ従う. ∎

この定理と local b 関数の定義から次の系が従う.

系 3
次の 2 つは同値.

1. $c(s) \in \mathbb{C}[s]$ が $b_{f,p}(s) \mid c(s)$ を満たす.

2. $a(p) \neq 0, a(x) \in (J_f \cap \mathbb{C}[x, t\partial_t]) : c(-t\partial_t - 1))$ を満たす $a(x) \in \mathbb{C}[x]$ が存在する.

global b 関数は柏原の定理により $\mathbb{Q}$ 上で 1 次式の積に因数分解されることが知られている. そこで $b_f(s)$ が $b_f(s) = \prod_{i=1}^{r} g_i(s)^{d_i}$ と分解されているとしよう. このとき, local b 関数 $b_{f,p}(s)$ は $b_{f,p}(s) \mid b_f(s)$ なので $b_{f,p}(s) = \prod_{i=1}^{r} g_i(s)^{e_i(p)} \quad (0 \leq e_i(p) \leq d_i)$ と書ける.

定義 4
$i = 1, \ldots, r$, $j = 0, \ldots, d_i$ に対して $b_{i,j}(s) := b_f(s)/g_i(s)^{d_i - j}$ と定義する. 更に, イデアル $I_{i,j}$ と, 多様体 $V_{i,j}$ を

$$I_{i,j} := ((J_f \cap \mathbb{C}[x, t\partial_t]) : b_{i,j}(-t\partial_t - 1)) \cap \mathbb{C}[x],\ I_{i,-1} := \{0\},\ V_{i,j} := V(I_{i,j}) \subset \mathbb{C}^n$$

で定義する.

定理 5

1. $\mathbb{C}[x] = I_{i,d_i} \supset I_{i,d_i-1} \supset \cdots \supset I_{i,0} \supset I_{i,-1} = \{0\}$

2. $\emptyset = V_{i,d_i} \subset V_{i,d_i-1} \subset \cdots \subset V_{i,0} \subset V_{i,-1} = \mathbb{C}^n$

3. $p \in V_{i,j-1} \setminus V_{i,j} \Leftrightarrow e_i(p) = j$.

証明 (3) のみ示す. $p \in V_{i,j-1}$ とすると, $\tilde{L}f^{s+1} = a(x)b_{i,j-1}(s)f^s$ を満たす $\tilde{L} \in D[s]$, $a(x) \in \mathbb{C}[x]$ が存在すれば $a(p) = 0$ となる. 系 3 より $b_{f,p} \not| b_{i,j-1}$. $p \notin V_{i,j}$ とすると, $a(x) \in \mathbb{C}[x]$ ($a(p) \neq 0$) が存在して $a(x) \in I_{i,j}$. よって, 系 3 より $b_{f,p} \mid b_{i,j}$. 故に $e_i(p) = j$. 逆に $e_i(p) = j$ とすると, $b_{f,p}|b_{i,j}$ よりある $a(x) \in \mathbb{C}[x]$, $a(p) \neq 0$ が存在して $a(x) \in I_{i,j}$. よって $p \notin V_{i,j}$. $b_{f,p} \not| b_{i,j-1}$ より $a(x) \in I_{i,j-1}$ ならば $a(p) = 0$ すなわち $p \in V_{i,j-1}$. ∎

系 6
$j = (j_1, \ldots, j_r)$ $(0 \leq j_i \leq d_i)$ に対して, $V^j = (\bigcap_{i=1}^{r} V_{i,j_i-1})$, $V_j = (\bigcup_{i=1}^{r} V_{i,j_i})$ とおく. このとき,

$$p \in V^j \setminus V_j \Leftrightarrow b_{f,p}(s) = \prod_{i=1}^{r} g_i(s)^{j_i}.$$

以上の議論から次のアルゴリズムが得られたことになる.

アルゴリズム 2 (local b 関数に付随する stratification)
Input : 多項式 $f \in \mathbb{C}[x]$
Output : local b 関数に付随する stratification $\{(C_j, b_j(s))\}$
$J_f \leftarrow \mathrm{in}_{(-w,w)}(I_f)$; $L_f \leftarrow J_f \cap \mathbb{C}[x, t\partial_t]$
$b_f(s) = \prod_{i=1}^{r} g_i(s)^{d_i} \leftarrow f$ の global b 関数
for $i = 1, \ldots, r$ do
  for $j = 0, \ldots, d_i$ do
    $b_{i,j}(s) \leftarrow b_f(s)/g_i(s)^{d_i - j}$; $I_{i,j} \leftarrow (L_f : b_{i,j}(-t\partial_t - 1)) \cap \mathbb{C}[x]$
  end for
end for
$S \leftarrow \emptyset$
for all $j = (j_1, \ldots, j_r)$ $(0 \leq j_i \leq d_i)$ do
  $V^j \leftarrow \bigcap_{i=1}^{r} V(I_{i,j_i-1})$; $V_j \leftarrow \bigcup_{i=1}^{r} V(I_{i,j_i})$.
  if $V^j \setminus V_j \neq \emptyset$ then
    $b_j(s) \leftarrow \prod_{i=1}^{r} g_i(s)^{j_i}$; $S \leftarrow S \cup \{(V^j \setminus V_j, b_j(s))\}$
  end if
end for

注意 1
1 点 $p \in \mathbb{C}^n$ における local b 関数を計算したいときは $I_{i,j}$ を全て計算する必要はなく, 各 i について $p \in V_{i,j-1} \setminus V_{i,j}$ となる j を見つけた時点で計算を止めればよい.

4 実装について

アルゴリズム 2 の各ステップの実装の詳細や改良点を述べる.

4.1 J_f および $b_f(s)$ の計算

J_f は $(-w, w)$ が負の数を含む weight であるため homogenization を行ってグレブナー基底を計算をする必要がある. ここでは, ある特殊な homogenization を行うことにより modular change of ordering を利用した高速化について述べる.

4.1.1 modular change of ordering

$\mathbb{Z}_{\langle p\rangle} := \{a/b \mid a, b \in b \not\in p\mathbb{Z}\}$, $\phi_p : \mathbb{Z}_{\langle p\rangle} \to \mathbb{F}_p$ を標数 p の有限体 F_p への標準射影とする.

アルゴリズム 3 (trace lifting によるグレブナー基底計算 [8])

Input :モニックな多項式を含む $<_0$ に関するグレブナ基底 $G_0 \subset \mathbb{Q}\langle x, \partial_x\rangle$
　　項順序 $<$
Output : $<$ に関する簡約グレブナー基底 $\langle G_0\rangle$
do
　　$p \leftarrow G_0 \subset Z_{\langle p\rangle}\langle x, \partial_x\rangle$ を満たす新たな素数
　　$G_p \leftarrow <$ に関する簡約グレブナー基底
　　If $\phi_p(G) = G_p$ を満たす $G \subset \langle G_0\rangle$ が存在 then
　　　return G
end do

このアルゴリズムにおいて, G の計算を trace algorithm [17] により行うと, $\mathbb{Q}$ 上の 0-簡約の手間と G がグレブナー基底であるかどうかの最終チェックの手間が, 有限体上のグレブナ基底計算の手間に置き換わる. これは計算途中に巨大な係数が現れる場合に非常に効果的である.

4.1.2 Weighted homogenization

Noro [9] は global b 関数の計算に適する weighted homogenization を次のように定義している. $\mathrm{tdeg}_\gamma(f)$ を f の weight $\gamma = (\gamma_1, \ldots, \gamma_n) \in \mathbb{R}^n_{>0}$ に関する全次数とし, $\delta = (\mathrm{tdeg}_\gamma(f) - \gamma_1 + 1, \ldots, \mathrm{tdeg}_\gamma(f) - \gamma_n + 1)$ とする. $\hat{\gamma} = (\mathrm{tdeg}_\gamma(f), \gamma)$, $\hat{\delta} = (1, \delta)$ をそれぞれ (t, x) , (∂_t, ∂_x) の weight ベクトルとし $d = \mathrm{tdeg}_\gamma(f) + 1$ とおく. このとき, weighted homogenized Weyl algebra $D\langle t, \partial_t, h\rangle$ を $x_1, \ldots, x_n$, $\partial_1, \ldots, \partial_n$, t, ∂_t, h で生成され, $\partial_i x_i - x_i\partial_i = h^d$ $(i = 1, \ldots, n)$, $\partial_t t - t\partial_t = h^d$ 以外は全て可換な基本関係を持つものとする.

$P = \sum c_{k,\alpha,l,\beta} t^k x^\alpha \partial_t^l \partial_x^\beta \in D\langle t, \partial_t\rangle$ の weighted homogenization $H_{(\hat{\gamma},\hat{\delta})}(P)$ を

$$H_{(\hat{\gamma},\hat{\delta})}(P) = \sum c_{k,\alpha,l,\beta} t^k x^\alpha \partial_t^l \partial_x^\beta h^{\deg_{(\gamma,\delta)}(P) - (k\cdot \mathrm{tdeg}_\gamma(f) + l + \gamma\alpha + \delta\beta)}$$

で定義し, $D\langle t, \partial_t\rangle$ における単項式順序 $\prec$ に対して $D\langle t, \partial_t, h\rangle$ における単項式順序 $\prec^h_{(\hat{\gamma},\hat{\delta})}$ を

$$uh^k \prec^h_{(\hat{\gamma},\hat{\delta})} vh^l \Leftrightarrow \mathrm{tdeg}_{(\hat{\gamma},\hat{\delta})}(uh^k) < \mathrm{tdeg}_{(\hat{\gamma},\hat{\delta})}(vh^l) \text{ or }$$
$$(\mathrm{tdeg}_{(\hat{\gamma},\hat{\delta})}(uh^k) = \mathrm{tdeg}_{(\hat{\gamma},\hat{\delta})}(vh^l) \text{ and } u \prec v)$$

で定義する.

命題 7
$F = \{t - f, \partial_1 + \frac{\partial f}{\partial x_1}\partial_t, \ldots, \partial_n + \frac{\partial f}{\partial x_n}\partial_t\}$, $\hat{f} = H_{(\hat{\gamma},\hat{\delta})}(f)$ とする. $\mathrm{in}_{\prec_0}(F) = \{t, \partial_1, \ldots, \partial_n\}$ となる任意の項順序 $\prec_0$ に対して, $H_{(\hat{\gamma},\hat{\delta})}(F) := \{t - \hat{f}, \partial_1 + \frac{\partial \hat{f}}{\partial x_1}\partial_t, \ldots, \partial_n + \frac{\partial \hat{f}}{\partial x_n}\partial_t\}$ は $\langle H_{(\hat{\gamma},\hat{\delta})}(F)\rangle$ のグレブナー基底.

この命題により, 上の weighted homogenization を用いれば J_f の計算にアルゴリズム 3 を適用することができる. いくつかの実験によれば, weight γ を適切に選べば J_f の計算に限らず, この後の計算も効率的に行えることが分かった. 今のところ γ はヒューリスティックに選択するしかないが, f が weighted homogeneous である場合はそうなるような weight を選択すると良いようである.

4.1.3 $b_f(s)$ の計算

$J_f \cap \mathbb{C}[t\partial_t]$ の生成元の計算はグレブナー基底計算による消去法ではなく, 冪を順に上げて未定係数法を行う方法で J_f の最小多項式として計算する.

アルゴリズム 4 (最小多項式の計算アルゴリズム)
Input : イデアル $J \subset R$ のグレブナー基底 G , $u \in R$
Output : u の J に関する最小多項式
$i \leftarrow 0$
do
  If $\mathrm{NF}_G(u^i) + \sum_{j=0}^{i-1} a_j \mathrm{NF}_G(u^j) = 0$ となる $a_{i-1}, \ldots, a_0 \in \mathbf{K}$ が存在 then
    return $s^i + \sum_{j=0}^{i-1} a_j s^j$
  else
    $i \leftarrow i + 1$
end do

4.2 $J_f \cap \mathbb{C}[x, t\partial_t]$ の計算

$L = J_f \cap \mathbb{C}[x, t\partial_t]$ を計算するために, 一旦グレブナー基底計算による消去法で J_f から変数 $\partial_1, \ldots, \partial_n$ を消去したイデアル $\tilde{L} = J_f \cap \mathbb{C}[x]\langle t, \partial_t\rangle$ を計算する. その後, Oaku [13] による方法で L を $\tilde{L} \cap \mathbb{C}[x, t\partial_t]$ として計算する. $\mathrm{ord}_{(-w,w)}(h) = d$ を満たす $(-w, w)$-homogeneous な元 $h \in \mathbb{C}[x]\langle t, \partial_t\rangle$ に対して $\psi(h)$ を次で定義する.

$$\psi(h) = \begin{cases} t^d h & d > 0 \\ \partial_t^{-d} h & d \leq 0 \end{cases}$$

このとき, $\psi(h)$ は $(-w, w)$-homogeneous かつ $\mathrm{ord}_{(-w,w)}(\psi(h)) = 0$ であるので $\psi(h) \in \mathbb{C}[x, t\partial_t]$ である.

定理 8
$\{g_1,\ldots,g_r\}$ を $\tilde{L}$ の生成元とし, 各 g_i が $(-w,w)$-homogeneous かつ $\mathrm{ord}_{(-w,w)}(g_i)=d_i$ を満たしているとする. このとき, $\tilde{L}\cap\mathbb{C}[t\partial_t]=\langle\psi(g_1),\ldots,\psi(g_r)\rangle$.

4.3 $I_{i,j}$ の計算

$L=J_f\cap\mathbb{C}[x,t\partial_t]$ を $s=t\partial_t$ とおいて $\mathbb{C}[x,s]$ のイデアルとみなす事で, これ以降の計算は可換環 $\mathbb{C}[x,s]$ 上で行うことができる. コロンイデアル $I_{i,j}=L:b_{i,j}(-s-1)$ の生成元は $L\cap\langle b_{i,j}(-s-1)\rangle$ の生成元を $b_{i,j}(-s-1)$ で割ることによって得ることができる. 更に, $b_{i,j+1}(s)=g_i(s)b_{i,j}(s)$ であったので

$$L:b_{i,j+1}(-s-1)=(L:b_{i,j}(-s-1)):g_i(-s-1)$$

という関係が成り立つ. 従って, $L:b_{i,0}$ から計算を始めてインクリメンタルに $L:b_{i,d_i}$ まで計算すれば効率的であろう.

4.4 V^j と V_j の計算

各 i に対して次のような多様体の列 T_i が得られる.

$$T_i:\emptyset=V_{i,d_i}\subset V_{i,d_i-1}\subset\cdots\subset V_{i,0}\subset V_{i,-1}=\mathbb{C}^n$$

このとき, 2 つの列 T_i と T_k が一致したならば, 各 local b 関数に現れる因子 g_i と g_k の指数がいつも一致することが分かるので探索の組み合わせを減らすことができる. 2 つの多様体の一致, $V(I)=V(J)$ を調べるには双方向の包含関係を見ればよい. 例えば, $V(I)\subset V(J)$ を調べたいなら J の全ての生成元が $\sqrt{I}$ の元であるかどうかを調べればよい. つまり, 根基所属判定を行えばよい.

この操作を行った後, 各列から 1 つずつ選んでイデアルの和と交わりを取って順に V^j と V_j を構成していく. このとき, 交わりや差集合 $V^j\setminus V_j$ が空集合であるかどうかを調べて空集合でないもののみ残していけば更に組み合わせを減らすことができる. また, $V^j\setminus V_j=\emptyset\Leftrightarrow V^j\subset V_j$ なので, これも根基所属判定によって分かる.

4.5 $b_{f,p}(s)$ を与える作用素の計算

各 stratum の情報から $Lf^{s+1}=b_{f,p}(s)f^s$ を満たすような作用素 $L\in\mathbb{C}[x]_p\langle\partial_x\rangle[s]$ を以下のアルゴリズムで計算することができる.

アルゴリズム 5 (local b 関数を与える作用素の計算アルゴリズム)
Input : $p\in\mathbb{C}^n$ における f の local b 関数 $b_{f,p}(s)=\prod_{i=1}^{r}g_i(s)^{e_i(p)}$, イデアル $I_p=\bigcap_{i=1}^{r}I_{i,e_i(p)}$
Output : $L(x,\partial_x,s)f^{s+1}=b_{f,p}(s)f^s$ を満たす作用素 $L(x,\partial_x,s)\in\mathbb{C}[x]_p\langle\partial_x\rangle[s]$
$a(x)\leftarrow a(p)\neq 0$ を満たす I_p に含まれる多項式
$G\leftarrow I_f$ の $\prec_{(-w,w)}$ に関するグレブナー基底 $\{g_1,\ldots,g_k\}$
$a(x)b_{f,p}(-t\partial_t-1)=\sum_{i=1}^{k}h_i\mathrm{in}_{(-w,w)}(g_k)$ を満たす $(-w,w)$-homogeneous な $h_i\in D\langle t,\partial_t\rangle$ を計算.

$R \leftarrow a(x)b_{f,p}(-t\partial_t - 1) - \sum_{i=1}^{k} h_i g_k$

R を $\sum_{k\geq 1, l\geq 0} u_{k,l}(x,\partial_x)(t\partial_t)^l t^k$ という形に書き直す.

$\hat{L} \leftarrow \sum_{k\geq 1, l\geq 0} u_{k,l}(x,\partial_x)(-s-1)^l f^{k-1}$

retun $1/a(x) \cdot \hat{L}$

このアルゴリズムにおける, G や I_p はアルゴリズム 2 の実行中であれば保持している情報である. したがって, 1 つの作用素を計算するために生じる新たな手間は h_i を計算すること, つまり $D\langle t, \partial_t\rangle$ 上の割り算を 1 回実行する程度である.

5 実験

アルゴリズム 2 を Risa/Asir [10] において実装した. この章ではいくつかの例について計算を行った結果を示す. 使用したのは Linux OS, Intel Xeon X5450 (3GHz) 搭載マシンである.

5.1 計算例

入力として $f(x,y,z) = x^3 - y^2z^2$ を用いると (Oaku [12] の例) アルゴリズム 2 は次のような結果を返す.

$$(\langle x^2, -y, -z\rangle, \langle 1\rangle, (s+1)(3s+4)(3s+5)(6s+5)^2(6s+7)^2)$$
$$(\langle zy, x^2\rangle, \langle -x^2, -z^2, -y^2\rangle, (s+1)(6s+5)(6s+7))$$
$$(\langle x^3 - z^2y^2\rangle, \langle -x^2, -z^2y, zy^2\rangle, s+1)$$
$$(\langle 0\rangle, \langle x^3 - z^2y^2\rangle, 1).$$

各 $(I, J, b(s))$ は $p \in V(I) \setminus V(J)$ における local b 関数が $b(s)$ であることを表している. 出力されたこれらのイデアルの根基を計算して整理すると,

$$\mathbb{C}^3 = \{(0,0,0)\} \cup S_1 \cup S_2 \cup (\mathbb{C}^3 \setminus V(f))$$

という stratification が得られる. ただし, $S_1 = (V(x,z) \cup V(y,z)) \setminus \{(0,0,0)\}$, $S_2 = V(f) \setminus (V(x,z) \cup V(y,z))$ とする. また, 各 stratum に対応する local b 関数 $b_{f,p}(s)$ は

$$b_{f,p}(s) = \begin{cases} (s+1)(3s+4)(3s+5)(6s+5)^2(6s+7)^2 & , p = (0,0,0) \\ (s+1)(6s+5)(6s+7) & , p \in S_1 \\ (s+1) & , p \in S_2 \\ 1 & , p \notin V(f) \end{cases}$$

であることが分かる. 更に, $k \leq 6$ の f^k についても計算を行った. f^6 の stratification の計算には 340 秒を要した. f は weight $(4,3,3)$ に関して weighted homogeneous であり, この weight を設定すると 194 秒で計算が完了した.

5.2 A_k 型特異点 x^{k+1} の versal deformation

1 点における local b 関数は解析的な不変量であり, local b 関数に付随する stratification は超曲面における特異点の分類の情報を与えている.

$f_k(x, u_1, \ldots, u_k)$ を A_k 型特異点 x^{k+1} の versal deformation

$$f_k(x, u_1, \ldots, u_k) = x^{k+1} + u_1 + u_2 x + \cdots + u_k x^{k-1} \quad (k \geq 1)$$

とし, その判別式 $\mathrm{disc}(f_k) = \mathrm{resultant}_x(f_k(x), f_k'(x))$ を考える. これは f_k が重根をもつ条件を表しており, $V(\mathrm{disc}(f_k))$ は根の重複の様子によって自然に stratify される. $k = 3$ の場合を考えてみる.

$$\mathrm{disc}(f_3) = 256u_1^3 - 128u_1^2u_3^2 + 144u_1u_2^2u_3 + 16u_1u_3^4 - 27u_2^4 - 4u_2^2u_3^3$$

は $\mathbb{R}^3$ の曲面で, とくに swallowtail と呼ばれるものである. $I_{2,2} = \langle u_2, 4u_1 - u_3^2 \rangle$, $I_{3,1} = \langle 27u_2^2 + 8u_3^3, 12u_1 + u_3^2 \rangle$ とおく. このとき, $V(I_{2,2}) \cap V(I_{3,1}) = \{(0,0,0)\}$ であり, $V(\mathrm{disc}(f_3))$ の local b 関数に付随する stratification は

$$\mathbb{C}^3 = S_4 \cup S_{2,2} \cup S_{3,1} \cup S_{2,1,1} \cup S_{1,1,1,1}$$

で与えられる. ただし, $S_4 = \{(0,0,0)\}$, $S_{2,2} = V(I_{2,2}) \setminus \{(0,0,0)\}$, $S_{3,1} = V(I_{3,1}) \setminus \{(0,0,0)\}$, $S_{2,1,1} = V(\mathrm{disc}(f_3)) \setminus (V(I_{2,2}) \cup V(I_{3,1}))$, $S_{1,1,1,1} = \mathbb{C}^3 \setminus V(\mathrm{disc}(f_3))$ である. 各 S の添字は f_3 の重根の取り方と対応するようにとってある. 例えば, $S_{3,1}$ は $(u_1, u_2, u_3) \in S_{3,1}$ なら f_3 は 3 重根を 1 つ, 単根を 1 つ持つという意味である. 計算機による実験で $k = 5$ までのとき, 重根の取り方による stratification と一致することが確かめられた. また, この例では $(u_1, u_2, \ldots, u_k)$ に対する weight を $(k+1, k, \ldots, 2)$ と取ると $\mathrm{disc}(f_3)$ は weighted homogeneous となる. この weight を用いない場合は計算が非常に困難になる. 例えば, $k = 5$ のときは $\mathrm{in}_{(-w,w)}(\mathrm{disc}(f_5))$ のグレブナー基底計算が 1 時間以上経っても終わらなかった.

5.3 D_k 型特異点 $x^2y - y^{k-1}$ の versal deformation

次に, D_k 型特異点 $x^2y - y^{k-1}$ の versal deformation

$$f_k(x, y, u_1, \ldots, u_k) = x^2y - y^{k-1} + u_1 + u_2x + u_3x^2 + u_4y + \cdots + u_ky^{k-3} \quad (k \geq 4)$$

について考える. f_k を $(u_1, \ldots, u_k)$ をパラメータとする (x, y)-平面の曲線と見なして, 特異点集合のパラメータ空間への射影を考えると, それから $\langle f_k, \frac{\partial f_k}{\partial x}, \frac{\partial f_k}{\partial y} \rangle \cap \mathbb{C}[u_1, \ldots, u_k] = \langle g_k \rangle$ を満たすような多項式 $g_k(u_1, \ldots, u_k)$ が得られる. この g_k を入力として用いる.

g_4 に対して stratification を計算すると $\mathbb{C}^4$ は 7 つの stratum に stratify される. 計算にかかる時間は (u_1, u_2, u_3, u_4) に対する weight を $(6,4,2,4)$ と設定すれば 4 秒程度しかかからない.

g_5 に対しては $\mathbb{C}^5$ は 12 個の stratum に stratify される. $(u_1, u_2, u_3, u_4, u_5)$ に対する weight を $(8,5,2,6,4)$ とすると 8937 秒を要した. 表 1 はこの計算の各ステップにかかった時間を示したものである.

$J_f = \mathrm{in}_{(-w,w)}(I_f)$	$J_f \cap \mathbb{C}[x, t\partial_t]$	$I_{i,j}$	V^j, V_j	Total
4053	2122	1407	1355	8937

表 1: $V(g_5)$ に対する stratification の計算時間 (秒)

6 まとめ

local b 関数に付随する stratification を計算する従来の方法は準素分解を必要とするものであったが, その計算は困難である場合も多い. ここで提案した方法は準素分解を用いず, コロンイデアルの計算やイデアルの包含関係を調べるといった基本的な計算のみを用いるものである. 2 つの方法における L_f と $J_f \cap \mathbb{C}[x, t\partial_t]$ の計算は本質的に同じものであるから相互に入れ替えることができる. そのため, 計算効率を考えるにはその後の計算で比較するのが妥当である. 例えば, $V(g_5)$ に対しては $I_{i,j}$ と V^j, V_j の計算に 2762 秒を要する. 一方, SINGULAR [2] の 関数 `primdecGTZ` によって準素分素を計算させたところ, 5 日以上経過しても結果は返ってこなかった. また, $I_{i,j}$ の計算は各 i に対して独立であるので, マルチコア環境であれば比較的簡単に並列計算を行うこともできる.

我々のアルゴリズムは local b 関数が global b 関数を割り切るという性質を用いて, 有限個の候補に対してチェックを行うというものである. 最近, Oaku, Bahloul [14] によって local Bernstein-Sato イデアル に付随する stratification を計算するアルゴリズムが提案された. local Bernstein-Sato イデアルは local b 関数の一般化であり, その計算アルゴリズムも準素分解を用いる方法を拡張したものである. しかし, 今のところ local Bernstein-Sato イデアルの候補を有限個にとる方法は分かっていない. 従って, 今回提案したアルゴリズムを適用することはできない.

参 考 文 献

[1] Briançon, J., Granger, M., Maisonobe, Ph., Miniconi, M., Algorithme de calcul du polynôme de Bernstein: cas non dégénéré. *Ann. Inst. Fourier* **39**, 553–610.

[2] Greuel, G.-M., Pfister, G., Schönemann, H., 2001. SINGULAR 3.0 - A Computer Algebra System for Polynomial Computations. In M. Kerber and M. Kohlhase: Symbolic Computation and Automated Reasoning, The Calculemus-2000 Symposium, 227–233.

[3] Greuel, G.-M., Lossen, C., Shustin, E., 2007. Introduction to Singularities and Deformations. Springer Monographs in Mathematics.

[4] Kashiwara, M., 1976. B-functions and holonomic systems, *Invent. Math.* **38**, 33–53.

[5] Levandovskyy, V., Morales, J. M., 2008. Computational D-module Theory with SINGULAR, Comparison with Other Systems and Two New Algorithms. Proceedings of ISSAC 2008, ACM Press, 173–180.

[6] Malgrange, B., 1974. Intégrales asymptotiques et monodromie, *Ann. Sci. ENS* **4**, 405–430.

[7] Nakayama, H., 2008. Algorithm computing the b function by an approximate division algorithm in $\hat{D}$, To appear in *J. Symb. Comp.*.

[8] Noro. M., Yokoyama, K., 1999. A Modular Method to Compute the Rational Univariate Representation of Zero-Dimensional Ideals, *J. Symb. Comp.* **28**, 1, 243-263.

[9] Noro. M., 2002. An Efficient Modular Algorithm for Computing the Global BFunction, Mathematical Software, Proceedings of the first international congress of mathematical software, Beijing, Edited by A. M. Cohen, X. S. Gao, N. Takayama, World Scientific, 147–157.

[10] Noro, M., 2002. A Computer Algebra System: Risa/Asir, Algebra, Geometry and Software, M. Joswig and N. Takayama (eds.), Springer, 147–162.

[11] Oaku, T., 1997. Algorithms of computing b-functions. *Duke Math.J.* **87**, 115–132.

[12] Oaku, T., 1997. Algorithms for b-Functions, Restrictions, and Algebraic Local Cohomology Groups of D-Modules. *Advances in Applied Mathematics* **19**, 61–105.

[13] Oaku, T., 1997. Algorithm for the b-function and D-modules associated to a polynomial. *J. Pure Appl. Algebra* **117** & **118**, 495–518.

[14] Oaku, T., Bahloul, R., 2008. Algorithm for Computing Local Bernstein-Sato Ideals, preprint arXiv:math/0806.4869v1, 2008.

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

[16] Sato, M., Kashiwara, M., Kimura, T., Oshima, T., 1980. Micro-local analysis of prehomogeneous vector spaces, *Invent. Math.* **62**, 117–179.

[17] Traverso, C., 1988. Groebner trace algorithms, Proceedings of ISSAC'88, LNCS **358**, Springer-Verlag, 125–138.

[18] Yano, T., 1978. On the Holonomic System of f^s and b-functions. *Publ. RIMS Kyoto Univ.* **12** Suppl., 469–480.

[19] Yano, T., 1978. On the theory of b-functions, *Publ. RIMS Kyoto Univ.* **14**, 111–202.