

積分の満たす非斉次微分方程式系を与えるアルゴリズム

An algorithm of computing inhomogeneous differential equations for definite integrals

中山 洋将

NAKAYAMA HIROMASA

神戸大学大学院理学研究科 / JST CREST

DEPARTMENT OF MATHEMATICS, GRADUATE SCHOOL OF SCIENCE, KOBE UNIVERSITY *

西山 絢太

NISHIYAMA KENTA

神戸大学大学院理学研究科 / JST CREST

DEPARTMENT OF MATHEMATICS, GRADUATE SCHOOL OF SCIENCE, KOBE UNIVERSITY †

Abstract

我々はパラメータ付き定積分の満たす非斉次微分方程式系を求める新しいアルゴリズムを与えた. このアルゴリズムは大阿久による D -加群における積分アルゴリズムを改良したのもであり, 微分作用素環上のグレブナ基底の理論が用いられている. このアルゴリズムを Risa/Asir のパッケージ `nk_restriction.rr` として実装した. これを用いた様々な計算例についても掲載している.

Abstract

We give an algorithm to compute inhomogeneous differential equations for definite integrals with parameters. The algorithm is based on the integration algorithm for D -modules by Oaku. Main tool in the algorithm is the Gröbner basis method in the ring of differential operators. We implement our algorithm as the package `nk_restriction.rr` on the Risa/Asir and give some examples.

1 イントロダクション

変数の組 $t = (t_1, \dots, t_m)$, $x = (x_1, \dots, x_n)$ に対する微分作用素を $\partial_t = (\partial_{t_1}, \dots, \partial_{t_m})$, $\partial_x = (\partial_{x_1}, \dots, \partial_{x_n})$ と書くことにし, $m+n$ 変数と m 変数の微分作用素環をそれぞれ

$$D = K\langle t, x, \partial_t, \partial_x \rangle, \quad D' = K\langle t, \partial_t \rangle$$

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

†nisiyama@math.kobe-u.ac.jp

と定義する. このとき, ホロノミックな左 D イデアル I の t についての積分イデアル J とは

$$J = (I + \partial_{t_1} D + \cdots + \partial_{t_m} D) \cap D'$$

なる左 D' イデアルのことであり, 積分イデアルを計算するアルゴリズムは Oaku [6] により与えられた.

我々はこの積分アルゴリズムを改良することにより, 積分イデアル J の生成元を求めるだけでなく, 各生成元 $P \in J$ について,

$$P = P_0 + \partial_{t_1} P_1 + \cdots + \partial_{t_m} P_m$$

となる $P_0 \in I, P_1, \dots, P_m \in D$ を計算するアルゴリズムを与えた. このような $P_1, \dots, P_m$ を P の非斉次部分と呼ぶことにする. 非斉次部分を計算することにより, パラメータを含む積分の満たす非斉次微分方程式系を得られる.

例えば, x, t を 1 変数 (すなわち $m = n = 1$) として, 積分 $A(x) = \int_a^b e^{-t-xt^3} dt$ の満たす微分方程式を考える. 被積分関数 $f(t, x) = e^{-t-xt^3}$ の D における零化イデアル $I = \langle \partial_t + 1 + 3xt^2, \partial_x + t^3 \rangle$ の t に関する積分イデアルは $J = \langle 27x^3 \partial_x^2 + 54x^2 \partial_x + 6x + 1 \rangle = \langle P \rangle$ である. J の生成元 P の非斉次部分は $P_1 = -(\partial_t^2 + 3\partial_t + 3)$ となる. この P を積分に作用させることで

$$\begin{aligned} P \cdot A(x) &= \int_a^b \partial_t (P_1 \cdot e^{-t-xt^3}) dt = \left[P_1 \cdot e^{-t-xt^3} \right]_{t=a}^{t=b} \\ &= - \left[(9x^2 t^4 - 3xt^2 - 6xt + 1) e^{-t-xt^3} \right]_{t=a}^{t=b} \end{aligned}$$

が得られ, 積分 $A(x)$ の満たす非斉次微分方程式が得られる. 今までの積分アルゴリズムでは, 積分イデアルの生成元だけしか得られず, 非斉次部分を計算できないので, 上のような計算はできなかった.

この論文では, 積分イデアルの非斉次部分の計算アルゴリズムとその適用例について述べる. また, 積分の満たす非斉次微分方程式系を計算する他のアルゴリズムとして, Almkvist-Zeilberger アルゴリズム ([1]) や Chyzak アルゴリズム ([3],[4]), Oaku-Shiraki-Takayama アルゴリズム ([7]) などがあるが, それらアルゴリズムと我々のアルゴリズムとの比較についても述べる. 更に, 我々はこの論文で述べたアルゴリズムを, 数式処理ソフト Risa/Asir ([11]) において実装し, パッケージ `nk_restriction.rr` ([14]) として公開している.

2 D 加群の積分アルゴリズム

この節では, D 加群における積分アルゴリズムを解説する.

微分作用素環 $D = K\langle x_1, \dots, x_m, x_{m+1}, \dots, x_n, \partial_1, \dots, \partial_m, \partial_{m+1}, \dots, \partial_n \rangle$, とその部分環 $D' = K\langle x_1, \dots, x_m, \partial_1, \dots, \partial_m \rangle$ を定めておく. ($m \leq n$)

D の環同型写像 $\mathcal{F}$ を

$$\mathcal{F}(x_i) = \begin{cases} -\partial_i & (1 \leq i \leq m) \\ x_i & (m < i \leq n) \end{cases}, \mathcal{F}(\partial_i) = \begin{cases} x_i & (1 \leq i \leq m) \\ \partial_i & (m < i \leq n) \end{cases}$$

なるものとし, Fourier 変換と呼ぶ.

ホロノミック左 D イデアル I の $x_1, \dots, x_m$ についての積分イデアル J とは

$$J = (I + \partial_1 D + \dots + \partial_m D) \cap D'$$

なる左 D' イデアルである.

アルゴリズム 1 (D 加群の積分アルゴリズム, [6], [8])

入力: ホロノミック左 D イデアル I の生成元.

重みベクトル $w = (w_1, \dots, w_m, w_{m+1}, \dots, w_n)$ で $w_1, \dots, w_m > 0, w_{m+1} = \dots = w_n = 0$ を満たすもの.

出力: I の $x_1, \dots, x_m$ についての積分イデアルの生成元.

1. $\mathcal{F}(I)$ に対し w についての制限加群の計算. 具体的には次のように行う.

- (a) $\mathcal{F}(I)$ の $\langle_{(-w, w)}$ についてのグレブナ基底 $\{h_1, \dots, h_l\}$ を計算.
- (b) $\mathcal{F}(I)$ の $(-w, w)$ についての generic b 関数 $b(s)$ を計算. s_0 を $b(s)$ の非負最大整数根とする.
- (c) s_0 がない時, 制限加群は 0 であり, 積分イデアルは D' となり, アルゴリズムを終了.
- (d) $m_i = \text{ord}_{(-w, w)}(h_i)$,
 $B_d = \{\partial_1^{i_1} \dots \partial_m^{i_m} \mid i_1 w_1 + \dots + i_m w_m \leq d\}$,
 $r = \#\{(i_1, \dots, i_m) \mid i_1 w_1 + \dots + i_m w_m \leq s_0\}$ とおく.
- (e) $\tilde{B} = \bigcup_{i=1}^l \{\tilde{h}_{i\beta} := \partial^\beta h_i = \sum c_{u,v} x^u \partial^v \mid \partial^\beta \in B_{s_0 - m_i}\}$ とし,
 $B = \{h_{i\beta} := \tilde{h}_{i\beta} \mid x_1 = \dots = x_m = 0 \mid \tilde{h}_{i\beta} \in \tilde{B}\}$ を返す.

2. $\mathcal{F}^{-1}(B)$ を左 D' 加群 $(D')^r$ の元とみなしたものを M とおく.

3. M に対し, x^0 に対応する位置が最小になるような POT 項順序について, グレブナ基底を計算する. その生成元の中で, x^0 に対応する成分以外が全て 0 になっているものの x^0 成分からなる集合が I の積分イデアルの生成系.

I を被積分関数の満たす微分方程式系としたとき, その積分イデアルは積分の満たす斉次微分方程式系に対応している. まず, ホロノミックな関数 $f(t, x)$ の t に関する定積分

$$A(x) = \int_R f(t, x) dt, \quad R = \prod_{i=1}^m [a_i, b_i]$$

を考える. $I = \text{Ann}_D f$ の t に関する積分イデアルを J とすると, 任意の $p \in J$ に対して,

$$p - \sum_{i=1}^m \partial_{t_i} p_i \in I$$

となるような $p_1, \dots, p_m \in D$ が存在し, 微分方程式

$$p \cdot A(x) = \int_R p \cdot f dt = \int_R \sum_{i=1}^m (\partial_{t_i} p_i) \cdot f dt = \sum_{i=1}^m \int_R \partial_{t_i} (p_i \cdot f) dt \quad (1)$$

を満たす. つまり, 積分アルゴリズムの出力 p は, $A(x)$ の積分領域として式 (1) の最右辺が 0 となるようなものを取ったとき, $p \cdot A(x) = 0$ という斉次微分方程式を満たすものとみなすことができる.

3 積分の満たす非斉次微分方程式系を与えるアルゴリズム

この章では、前節の積分アルゴリズムを改良して、積分イデアルの非斉次部分の計算アルゴリズムを与える。また、このアルゴリズムの応用として、積分が満たす非斉次な微分方程式系を得られることを述べる。

前節の最後の議論から、式 (1) の右辺が 0 とならないような積分領域の場合は、 p_i ($1 \leq i \leq m$) を具体的に求める必要がある。

定理 1

ホロノミックな左 D -イデアル I の積分イデアルを $J \subset D'$ とする。このとき、任意の $p \in J$ に対して

$$p - \sum_{i=1}^m \partial_i p_i \in I \quad (2)$$

を満たす微分作用素 $p_i \in D$ ($1 \leq i \leq m$) を計算するアルゴリズムが存在する。

証明 アルゴリズム 1 を適用して積分イデアル J の生成系 $\{g_1, \dots, g_t\}$ を得る。この各生成元 g_j ($1 \leq j \leq t$) に対して示せば十分である。アルゴリズム 1 の (3) より、 $q_{ji\beta} \in D'$ を用いて $g_j = \sum q_{ji\beta} \mathcal{F}^{-1}(h_{i\beta})$ と表現される。また、この $q_{ji\beta}$ は グレブナ基底の計算における履歴を参照することで計算できる。従って、

$$\begin{aligned} I \ni \sum q_{ji\beta} \mathcal{F}^{-1}(\tilde{h}_{i\beta}) &= g_j - \left(g_j - \sum q_{ji\beta} \mathcal{F}^{-1}(\tilde{h}_{i\beta}) \right) \\ &= g_j - \sum q_{ji\beta} \left(\mathcal{F}^{-1}(h_{i\beta}) - \mathcal{F}^{-1}(\tilde{h}_{i\beta}) \right) \\ &= g_j - \sum q_{ji\beta} \left(\mathcal{F}^{-1}(\tilde{h}_{i\beta}|_{x_1=\dots=x_m=0}) - \mathcal{F}^{-1}(\tilde{h}_{i\beta}) \right) \\ &= g_j - \sum q_{ji\beta} \mathcal{F}^{-1}(\tilde{h}_{i\beta}|_{x_1=\dots=x_m=0} - \tilde{h}_{i\beta}) \end{aligned}$$

と変形できる。ここで、 $\tilde{h}_{i\beta}|_{x_1=\dots=x_m=0} - \tilde{h}_{i\beta}$ の各項は $x_1, \dots, x_m$ のいずれかで左から割り切れるので、 $\mathcal{F}^{-1}(\tilde{h}_{i\beta}|_{x_1=\dots=x_m=0} - \tilde{h}_{i\beta})$ の各項は $\partial_1, \dots, \partial_m$ のいずれかで左から割り切れる。従って、

$$\sum q_{ji\beta} \mathcal{F}^{-1}(\tilde{h}_{i\beta}|_{x_1=\dots=x_m=0} - \tilde{h}_{i\beta}) = \sum_{i=1}^m \partial_i p_{ij}$$

と書き直せばよい。 ■

アルゴリズム 2 (非斉次部分の計算アルゴリズム)

入力：ホロノミック左 D イデアル I の生成元。

重みベクトル $w = (w_1, \dots, w_m, w_{m+1}, \dots, w_n)$ で $w_1, \dots, w_m > 0, w_{m+1} = \dots = w_n = 0$ を満たすもの。

出力： I の $x_1, \dots, x_m$ についての積分イデアルの生成元 $\{g_1, \dots, g_t\}$ 。

各生成元 g_j ($1 \leq j \leq t$) に対して $g_j - \sum_{i=1}^m \partial_i p_{ij} \in I$ を満たす $p_{ij} \in D$ 。

1. アルゴリズム 1 を適用。

2. $g_j = \sum q_{ji\beta} \cdot \mathcal{F}^{-1}(h_{i\beta})$ を満たす $q_{ji\beta}$ を計算.

$R_j = g_j - \sum q_{ji\beta} \mathcal{F}^{-1}(\tilde{h}_{i\beta})$ を $R_j = \sum_{i=1}^m \partial_i p_{ij}$ と書き直す.

3. p_{ij} を出力.

定理 2 (多重積分の満たす微分方程式系を求めるアルゴリズム)

多重積分

$$F(x_{m+1}, \dots, x_n) = \int_{a_1}^{b_1} \cdots \int_{a_m}^{b_m} f(x_1, \dots, x_n) dx_1 \cdots dx_m \quad (m \leq n)$$

を考える. 被積分関数 $f(x_1, \dots, x_n)$ の満たす ホロノミックな微分方程式系が得られれば, 多重積分 $F(x_{m+1}, \dots, x_n)$ の満たす非斉次微分方程式が得られる.

証明 [概略] $m = 1$ のときは, アルゴリズム 2 を適用する. $m \geq 2$ のときは, アルゴリズム 2 を用いて, m 個の $m - 1$ 重積分に分解できるので, それを繰り返せばよい. ここでは, 簡単のため $m = 2$ の場合について, 多重積分

$$F(x_3, \dots, x_n) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x_1, \dots, x_n) dx_1 dx_2 \quad (2 \leq n)$$

の満たす微分方程式系を求めるアルゴリズムについて説明する.

被積分関数 $f(x_1, \dots, x_n)$ の満たすホロノミックな微分方程式系を I とし, I の x_1, x_2 についての積分イデアル

$$J = (I + \partial_1 D + \partial_2 D) \cap D' \quad (D' = K\langle x_3, \dots, x_n, \partial_3, \dots, \partial_n \rangle)$$

を計算する. J のある元 P をとってくると, $P = P_0 + \partial_1 P_1 + \partial_2 P_2$ ($P_0 \in I, P_1, P_2 \in D$) と表すことができる. P を積分 F に作用させると,

$$P \cdot F = \int_{a_2}^{b_2} (P_1 \cdot f|_{x_1=b_1} - P_1 \cdot f|_{x_1=a_1}) dx_2 + \int_{a_1}^{b_1} (P_2 \cdot f|_{x_2=b_2} - P_2 \cdot f|_{x_2=a_2}) dx_1$$

となる. ここで, 右辺の第 1 項目の積分を F_1 , 被積分関数を f_1 とし, 右辺の第 2 項目の積分を F_2 , 被積分関数を f_2 とする.

積分 F_1 の満たす微分方程式系を計算するために, 被積分関数 f_1 の満たすホロノミックな微分方程式系 I_1 を計算する. Oaku のアルゴリズムを用いて f_1 の零化イデアルを計算してもよいが, I から次のようにして計算することもできる.

イデアル商 $I : P_1$ は $P_1 \cdot f$ を零化するホロノミックイデアルである. ($Q \in I : P_1$ とすれば $QP_1 \in I$ すなわち $(QP_1) \cdot f = 0$. また I がホロノミックより $I : P_1$ はホロノミック.) $P_1 \cdot f|_{x_1=b_1}$ を零化する ホロノミックイデアル J_1 を得るには, $I : P_1$ の $x_1 = b_1$ についての制限イデアルを計算してやればよい. 同様に, $P_1 \cdot f|_{x_1=a_1}$ を零化する ホロノミックイデアル J_2 も得られる. $f_1 = P_1 \cdot f|_{x_1=b_1} - P_1 \cdot f|_{x_1=a_1}$ を零化するホロノミックイデアル I_1 は $J_1 \cap J_2$ で得られる.

I_1 の x_2 についての積分イデアル

$$K_1 = (I_1 + \partial_2 D_1) \cap D' \quad (D_1 = K\langle x_2, x_3, \dots, x_n, \partial_2, \partial_3, \dots, \partial_n \rangle)$$

を計算する. K_1 のある元 $P^{(1)}$ をとってくると, $P^{(1)} = P_0^{(1)} + \partial_2 P_2^{(1)}$ ($P_0^{(1)} \in K_1, P_2^{(1)} \in D_1$) と表すことができ, $P^{(1)}$ を積分 F_1 に作用させると,

$$P^{(1)} \cdot F_1 = P_2^{(1)} \cdot f_1|_{x_2=b_2} - P_2^{(1)} \cdot f_1|_{x_2=a_2}$$

となる. 積分 F_2 についても同様に, f_2 の零化イデアル I_2, I_2 の x_1 についての積分イデアル K_2 を計算する.

上の結果から

$$P^{(1)} \cdot P \cdot F = P^{(1)} \cdot F_1 + P^{(1)} \cdot F_2$$

なる式が得られ, 右辺の第 1 項目の積分は計算できる. 第 2 項目の積分を計算するために, $K_2 : P^{(1)}$ を計算し, その中の元 $P^{(2)}$ をとれば, $P^{(2)}P^{(1)} \in K_2$ となり, $P^{(2)}P^{(1)} \cdot F_2$ で計算ができる. 結局,

$$P^{(2)}P^{(1)}P \cdot F = P^{(2)}P^{(1)} \cdot F_1 + P^{(2)}P^{(1)} \cdot F_2$$

により, 積分 F についての非斉次な微分方程式が得られたことになる. ■

注意 1

非斉次微分方程式系

$$\ell_1 \cdot f = g_1, \dots, \ell_p \cdot f = g_p \quad (\ell_i \in D', g_i \text{ はホロノミック関数.})$$

において, $\langle \ell_1, \dots, \ell_p \rangle$ が D' におけるホロノミックイデアルとなると, この非斉次微分方程式系を, 非斉次ホロノミック系と呼ぶことにする. $m = 1$ のとき, 2 の出力は非斉次ホロノミック系であるが, $m \geq 2$ のとき, 出力が常に非斉次ホロノミック系であるかどうかは未解決である. しかし, 得られた微分方程式系が非斉次ホロノミック系でないときは, 後に解説する Oaku-Shiraki-Takayama アルゴリズム ([7]) を用いて, 非斉次ホロノミック系にすることができる.

例 1 (多重積分の満たす微分方程式系)

次の多重積分

$$F(s) = \int_0^1 \int_0^1 (x+y)^s dx dy$$

のについて, 上のアルゴリズムに沿って計算した結果を示す. $f = (x+y)^s$ の零化イデアルは

$$\langle (x+y)\partial_y - s, -\partial_x + \partial_y \rangle$$

となり, パラメータ s についての f の満たす差分方程式系は $E_s - (x+y)$ である. パラメータ s について Mellin 変換 (すなわち s, E_s を $-p\partial_p, p$ に変換) を零化イデアルと差分方程式系に行うと, 微分作用素環 $K\langle x, y, p, \partial_x, \partial_y, \partial_p \rangle$ におけるイデアル I

$$I = \langle (x+y)\partial_y + p\partial_p, -\partial_x + \partial_y, p - (x+y) \rangle$$

が得られる.

I の x, y についての積分イデアル J を計算すれば, $J = D' = K\langle p, \partial_p \rangle$ となり, $P = 1 \in J$ を F に作用させると

$$P \cdot F = \int_0^1 f_1 dy + \int_0^1 f_2 dx$$

なる 2 つ積分の和となる. 右辺の 2 つの積分をそれぞれ F_1, F_2 とおいておく. f_1 の満たす微分方程式系 I_1 は

$$I_1 = \langle -y + p - 1, p\partial_y + p\partial_a - 1 \rangle$$

f_2 の満たす微分方程式系 I_2 は

$$I_2 = \langle x^2 + (-2p+1)x + p^2 - p, p\partial_x^2 + (2p\partial_p - x - p - 1)\partial_x + p\partial_p^2 + (-x - p - 1)\partial_p + 4, \\ (p^2 - p)\partial_x + (p^2 - p)\partial_p + x - 3p + 2, -px\partial_x - px\partial_p + x + p \rangle$$

と計算できる.

積分 F_1 を計算するため, I_1 の y についての積分イデアル J_1 を計算すれば, $J_1 = \langle p\partial_p - 1 \rangle$ となり, $P^{(1)} = p\partial_p - 1 \in J_1$ を F_1 に作用させると

$$P^{(1)} \cdot F_1 = G_1$$

となり, G_1 の満たす微分方程式系 K_1 は

$$K_1 = \langle -p^2 + 3p - 2 \rangle$$

である. 同様に積分 F_2 を計算すれば, I_2 の x についての積分イデアルは $J_2 = \langle -p^2\partial_p^2 + 2p\partial_p - 2 \rangle$ となり, $P^{(2)} = -p^2\partial_p^2 + 2p\partial_p - 2 \in J_2$ を F_2 に作用させると

$$P^{(2)} \cdot F_2 = G_2$$

G_2 の満たす微分方程式系 K_2 は

$$K_2 = \langle (2p^3 - 8p^2 + 10p - 4)\partial_p - 5p^3 + 28p^2 - 45p + 22, p^4 - 6p^3 + 13p^2 - 12p + 4 \rangle$$

となる.

最終的な計算結果を見るために, $P, P^{(1)}, P^{(2)}, K_1, K_2$ をそれぞれ逆 Mellin 変換して s の差分作用素に戻せば,

$$\tilde{P} = 1, \tilde{P}^{(1)} = -s - 1, \tilde{P}^{(2)} = -(s+1)(s+2)$$

$$\tilde{K}_1 = \langle -E_s^2 + 3E_s - 2 \rangle$$

$$\tilde{K}_2 = \langle -5E_s^3 + (-2s+24)E_s^2 + (8s-37)E_s + (-10s+22) + (4s-4)E_s^{-1}, E_s^4 - 6E_s^3 + 13E_s^2 + 4 \rangle$$

この場合は $\tilde{P}^{(2)} = (s+2)\tilde{P}^{(1)}$ だから $\tilde{P} \cdot F = F_1 + F_2$ の式に $\tilde{P}^{(2)}$ を左から作用させて,

$$\tilde{P}^{(2)}\tilde{P} \cdot F = (s+2)\tilde{P}^{(1)}F_1 + \tilde{P}^{(2)}F_2 = (s+2)G_1 + G_2$$

となり, すなわち $-(s+1)(s+2)F = (s+2)G_1 + G_2$ であり, G_1, G_2 の満たす差分方程式系がそれぞれ $\tilde{K}_1, \tilde{K}_2$ である. さらに右辺 $G = (s+2)G_1 + G_2$ の満たす差分方程式系は $\tilde{K}_1, \tilde{K}_2$ から計算でき,

$$L = \langle (E_s - 2)^2(E_s - 1)^2 \rangle$$

である.

直接積分を計算すれば $F = \frac{1}{(s+1)(s+2)}(2^{s+2} - 2)$ であり, $-(s+1)(s+2)F$ に L の元を作用させると 0 になることが確かめられる.

4 既存のアルゴリズムとの比較

4.1 Almkvist-Zeilberger アルゴリズム

2 変数の hyperexponential function $F(t, x)$ について, 定積分 $\int_a^b F(t, x) dt$ の満たす x の非斉次微分方程式を計算するアルゴリズムとして, Almkvist-Zeilberger アルゴリズム (AZ アルゴリズム) ([1]) がある. ここで, hyperexponential function とは, 各変数について $\log$ 微分が有理関数となるような関数のことである.

アルゴリズム 3 (AZ アルゴリズム, [1])

入力: hyperexponential function $F(t, x)$

出力: x の微分作用素 $\bar{S}(x, \partial_x)$ と hyperexponential function $G(t, x)$ で次の微分方程式を満たすもの.

$$\bar{S}(x, \partial_x) \cdot F(t, x) = \partial_y \cdot G(t, x)$$

さらに $G(t, x) = (\text{有理関数})F(t, x)$ も成り立つ.

連続版の Gosper アルゴリズムと未定係数法を繰り返し用いて計算を行っている.

例 2

$$\int_0^\infty e^{-t-x^3} dt$$

の被積分関数 $F(t, x) = e^{-t-x^3}$ に対して AZ アルゴリズムを適用してやると,

$$\bar{S}(x, \partial_x) \cdot F(t, x) = \partial_t \cdot G(t, x)$$

が成り立つ微分作用素 $\bar{S}(x, \partial_x)$ と hyperexponential function $G(t, x)$ が得られる.

$$\bar{S}(x, \partial_x) = 27x^3\partial_x^2 + 54x^2\partial_x + 6x + 1$$

$$G(t, x) = (-9t^4x^2 + 3t^2x + 6tx - 1)F(t, x)$$

多項式 $H(t, x) = (-9t^4x^2 + 3t^2x + 6tx - 1)$ とおき, 上の式を変形すれば,

$$(\bar{S}(x, \partial_x) - \partial_y H(t, x)) \in \text{Ann}_{D_2} F(t, x)$$

$\bar{S}(x, \partial_x)$ が, 零化イデアル $\text{Ann}_{D_2} F(t, x)$ の t についての積分イデアル

$$(\text{Ann}_{D_2} F(t, x) + \partial_y D_2) \cap D_1$$

に含まれることがわかる. (ただし, $D_2 = K\langle x, t, \partial_x, \partial_t \rangle$, $D_1 = K\langle x, \partial_x \rangle$ としておく.) この場合は, 積分アルゴリズムによる計算結果と全く同じ結果を返している. (例 8 参照.)

```
[1481] gosper_int_op(exp(-t-x*t^3), x, t);
[27*x^3*dx^2+54*x^2*dx+6*x+1,
-9*exp(-t^3*x-t)*t^4*x^2+(3*exp(-t^3*x-t)*t^2+6*exp(-t^3*x-t)*t)*x-
exp(-t^3*x-t)]
```

例 3 ([1] からの計算例)

$F(x, y) = e^{-x^2/y^2 - y^2}$ に対して AZ アルゴリズムを適用してやると,

$$(\partial_x^2 - 4) \cdot F(x, y) = \partial_y \cdot \left(\frac{2}{y} F(x, y) \right)$$

なる式が得られる.

積分アルゴリズムとの結果と比較するために, 上の式を変形していく.

$$((\partial_x^2 - 4) - \partial_y \frac{2}{y}) \cdot F(x, y) = 0$$

と変形できる. 有理関数の部分 $\frac{2}{y}$ を消して, 多項式係数の微分作用素 D_2 における式にするために, $-\frac{y}{x} \partial_x \cdot F(x, y) = \frac{2}{y} F(x, y)$ を使うと,

$$((\partial_x^2 - 4) - \partial_y (-\frac{y}{x} \partial_x)) \cdot F(x, y) = 0$$

$$(x(\partial_x^2 - 4) + \partial_y(y \partial_x)) \cdot F(x, y) = 0$$

となり, $x(\partial_x^2 - 4) + \partial_y(y \partial_x) \in \text{Ann}_{D_2} F(x, y)$ がわかる. だから $x(\partial_x^2 - 4)$ が, 零化イデアル $\text{Ann}_{D_2} F(x, y)$ の y についての積分イデアルに含まれる.

```
[1491] gosper_int_op(exp(-x^2/y^2-y^2),x,y);
[dx^2-4,(2*exp((-x^2-y^4)/(y^2)))/(y)]
```

$F(x, y)$ を零化する元として, $y^2 \partial_x + 2x, y^3 \partial_y - 2x^2 + 2y^4$ をとり, それらの生成する D_2 のイデアルを I とおく. このイデアル I の y についての積分イデアル J を計算すれば,

$$J = \langle x^2 \partial_x^3 + 3x \partial_x^2 - 4x^2 \partial_x - 12x, x^3 \partial_x - 4x^3 \rangle$$

となる. J の生成元 $P = x^3 \partial_x - 4x^3$ について, 非斉次部分に対応する微分作用素は $Q = -\partial_y(-yx^2 \partial_x)$ である. $(P - Q) \in I \subset \text{Ann}_{D_2} F(x, y)$ より,

$$(P - Q) \cdot F(x, y) = x^2((x \partial_x - 4x) + \partial_y(y \partial_x)) \cdot F(x, y) = 0$$

となり, AZ アルゴリズムから得られた計算結果と同じものを返していることがわかる.

```
[1488] Id=[y^2*dx+2*x,y^3*dy-2*x^2+2*y^4];
[y^2*dx+2*x,y^3*dy-2*x^2+2*y^4]
[1489] integration_ideal(Id,[y,x],[dy,dx],[1,0]|inhomo=1);
-- nd_weyl_gr :0sec(0.0004811sec)
-- weyl_minipoly :0sec(0.0003979sec)
-- generic_bfct_and_gr :0sec(0.0004039sec)
generic bfct : [[1,1],[s,1],[s-1,1]]
S0 : 1
B_{S0} length : 2
-- fctr(BF) + base :0sec(0.0004849sec)
-- integration_ideal_internal :0sec(0.0007648sec)
[[x^2*dx^3+3*x*dx^2-4*x^2*dx-12*x,x^3*dx^2-4*x^3],
[[[dy,-y*x^2*dx],1],[[dy,-y*x^2*dx],1]]]
```

例 4 ([1] からの計算例 2)

$F(x, y) = e^{-x^6/y^4 - y^2}$ に対して AZ アルゴリズムを適用してやると,

$$(4x^2\partial_x^3 - 12x\partial_x^2 + 7\partial_x + 216x^5) \cdot F = \partial_y\left(\frac{1}{y}(-216x^{11} + (-108y^6 + 54y^4)x^5) \cdot F\right)$$

```
[1525] gopher_int_op(exp(-x^6/y^4-y^2),x,y);
[4*x^2*dx^3-12*x*dx^2+7*dx+216*x^5,
(-216*exp((-x^6-y^6)/(y^4))*x^11+(-108*exp((-x^6-y^6)/(y^4))*y^6+
54*exp((-x^6-y^6)/(y^4))*y^4)*x^5)/(y^7)]
```

$F(x, y)$ を零化する元として, $y^4\partial_x + 6x^5, y^5\partial_y - 4x^6 + 2y^6$ をとり, それらの生成する D_2 のイデアルを I とおく. このイデアル I の y についての積分イデアル J を計算すれば,

```
[1526] Id=[y^4*dx+6*x^5,y^5*dy-4*x^6+2*y^6];
[y^4*dx+6*x^5,y^5*dy-4*x^6+2*y^6]
[1527] integration_ideal(Id,[y,x],[dy,dx],[1,0]|inhomo=1);
-- nd_weyl_gr :0.004001sec(0.003166sec)
-- weyl_minipoly :0sec(0.0008359sec)
-- generic_bfct_and_gr :0sec(0.0008459sec)
generic bfct : [[1,1],[s,1],[s-3,1],[s-2,1],[s-1,1]]
S0 : 3
B_{S0} length : 4
-- fctr(BF) + base :0.004sec(0.002611sec)
-- integration_ideal_internal :0.004sec(0.003038sec)
[[-4*x^8*dx^3+12*x^7*dx^2-7*x^6*dx-216*x^11,
-4*x^7*dx^8-148*x^6*dx^7- ...],
[[[dy,(-6*x^7*dy+24*y*x^7)*dx^2+(-9*y*x^6*dy^2-15*x^6*dy-66*y*x^6)*dx]],2],
[[[dy,(-2*x^7*dy+4*y*x^7)*dx^8+(-3*y*x^6*dy^2+(-6*y^2-89)*x^6*dy+ ... ]]]]
```

積分イデアルの生成元の $-4x^8\partial_x^3 + 12x^7\partial_x^2 - 7x^6\partial_x - 216x^{11}$ は, AZ アルゴリズムで得られた x の微分作用素の $-x^6$ 倍になっていることが確認できる.

4.2 Chyzak アルゴリズム

Chyzak アルゴリズム ([3], [4], [5]) は未定係数法と Ore algebra におけるグレブナ基底の理論を用いたアルゴリズムであり, Ore algebra に含まれる様々なクラスを扱うことができる. 例えば, 和や積分の満たす差分方程式や微分方程式, それらの q -アナログ等を扱える. 有理関数体係数の微分作用素環 $K(x)\langle\partial_x\rangle$ においては, AZ アルゴリズムの適用範囲をホロノミック関数にまで拡張したとなっている. 以下が Chyzak アルゴリズムである.

アルゴリズム 4 (Chyzak アルゴリズム ([4], Algorithm 2.))

入力: $f(x, t)$ を零化するホロノミックイデアルの生成系 B .

出力: $P(\partial_x) \cdot f = \sum_i \eta_i \partial_x^i f = \partial_t[Q(x, \partial_x, \partial_t) \cdot f]$ を満たす作用素のペア (P, Q) .

1. B のグレブナ基底 G を計算して, $D/\text{Ann}f$ の基底 $\{\partial_x^\alpha \partial_t^\beta\}_{(\alpha,\beta) \in I}$ を得る.
2. $L = 0, 1, 2, \dots$ として以下を繰り返す.

(a) 未定係数 $\phi_{\alpha,\beta} \in \mathbf{C}(x)$, $\eta_i \in \mathbf{C}$ を導入して,

$$\partial_t \sum_{(\alpha,\beta) \in I} \phi_{\alpha,\beta} \partial_x^\alpha \partial_t^\beta - \sum_{i=0}^L \eta_i \partial_t^i$$

を G で割り算して, 基底 $\{\partial_x^\alpha \partial_t^\beta\}_{(\alpha,\beta) \in I}$ の一次結合に書き直す.

(b) $\phi_{\alpha,\beta}$, η_i の連立系と見て解を計算する.

(c) もし, 解ければ (P, Q) を返す. ただし, $P = \sum_{i=0}^L \eta_i \partial_t^i$, $Q = \sum_{(\alpha,\beta) \in I} \phi_{\alpha,\beta} \partial_x^\alpha \partial_t^\beta$.

このアルゴリズムには Chyzak 自身による実装があり, Maple のパッケージ Mgfund ([12]) として公開されている. このパッケージ中の関数 `creative telescoping` は, デフォルトでは, アルゴリズム中のステップ 2 を更に繰り返して, 得られた作用素 P たちによって生成されるイデアルがホロノミックになるまで繰り返される.

計算時間を比較してみると, 一般には, グレブナ基底計算の方が未定係数法より計算コストが大きいので, 多くの問題において, Chyzak アルゴリズムの方が我々のアルゴリズムより速い. しかし, 出力の微分作用素の階数が高く項の数が少ないような問題では, 我々のアルゴリズムの方が早くなることがある. 以下にそのような例を示す.

例 5

定積分

$$F(x, y) = \int_a^b \frac{1}{xt + y + t^{10}} dt.$$

の満たす非斉次微分方程式系を計算する. 以下は, 我々のアルゴリズムによって計算した時の出力であり, 計算にはおよそ 1.3 秒を要した.

```
[2345] load("nk_restriction.rr");
[2545] F=x*t+y+t^10$
[2546] Ann=ann(F)$ /* annihilating ideal of F`s */
0.052sec(0.0485sec)
[2547] Id=map(subst, Ann, s, -1)$ /* substitute s=-1 in Ann */
0sec(4.411e-05sec)
[1569] nk_restriction.integration_ideal(Id, [t,x,y], [dt,dx,dy], [1,0,0]
|inhomo=1);
-- nd_weyl_gr : 0.012sec + gc : 0.008001sec(0.02009sec)
-- weyl_minipoly : 0sec(0.001189sec)
-- generic_bfct_and_gr : 0.016sec + gc : 0.008001sec(0.02358sec)
generic bfct : [[1,1],[s,1],[s-9,1]]
S0 : 9
B_{S0} length : 10
-- fctr(BF) + base : 0.044sec + gc : 0.024sec(0.0674sec)
-- integration_ideal_internal : 0.8321sec + gc : 0.236sec(1.071sec)
[[9*x*dx+10*y*dy+9,-10*dx^9-x*dy^9,-9*dx^10+y*dy^10+9*dy^9],
[[[dt,-t]],1],[[dt,-dy^8]],1],[[dt,-t*dy^9]],1]]
0.9081sec + gc : 0.28sec(1.19sec)
```

以下は, Maple12 上で Chyzak アルゴリズム (パッケージ Mgfuns [12]) によって計算したときの出力であり, 計算にはおよそ 50 秒を要した.

```
with(Mgfuns):
f:=1/(x*t+y+t^10):
ts:=time():
creative_telescoping(f,[x::diff,y::diff], t::diff):
time()-ts;

49.583
```

これらの計算は共に, Intel Xeon 5450 (3.00GHz), 32 GB のメモリを搭載した計算機で行った.

4.3 Oaku-Shiraki-Takayama アルゴリズム

Oaku-Shiraki-Takayama によるアルゴリズム (以降, OST アルゴリズム) は, 各有限区間 $[a, b]$ における積分を Heaviside 関数を用いて $\mathbf{R}$ 上の積分とみなして積分アルゴリズムを適用し, 積分の満たす斉次微分方程式を得るアルゴリズムである. つまり, 今 $Y(t)$ を $Y(t) = 0 (t < 0), Y(t) = 1 (t \geq 0)$ で定義される関数とすると,

$$\int_a^b u(t, x) dt = \int_{-\infty}^{\infty} Y(t-a)Y(b-t)u(t, x) dt$$

であるから, 左辺の満たす斉次微分方程式を得るために, 右辺の非積分関数の満たす微分方程式系を入力として積分アルゴリズムを適用するということである. 彼らは Heaviside 関数と $u(t, x)$ の積の満たす微分方程式系を与える方法として以下の 2 種類を与えている.

- (a) Heaviside 関数の性質を利用する方法
- (b) D 加群のテンソル積の計算を用いる方法

前者は入力の微分方程式系を多項式倍する程度の処理で済むので計算は直ちに終了するが, その出力がホロノミックであるかどうかは分かっていない. つまり, その後の積分アルゴリズムの停止性は保証されない. 後者は $u(t, x)$ の満たすホロノミック系を入力とすれば, 積の満たすホロノミック系を得ることができるので, その後の積分アルゴリズムの停止性は保証される. しかし, テンソル積の計算のコストは非常に大きい. ここでは 前者を用いて計算する方法を OST アルゴリズム a, 後者を用いて計算する方法を OST アルゴリズム b と呼ぶことにする. 詳しくは [7, Chap 5] を参照.

例 6 ([7], Example 5.1.)

$$v(x) = \int_0^{\infty} e^{(-t^3+t)x} dt$$

以下は, $v(x)$ の満たす微分方程式系を OST アルゴリズムによって計算した結果である.

```
[1486] B = [dt+(3*t^2-1)*x,dx+t^3-t];
[dt+(3*t^2-1)*x,dx+t^3-t]
[1487] ost_integration_ideal(B,[t,x],[dt,dx],[1,0],[0],["inf"]);
-- nd_weyl_gr :0sec(0.014sec)
-- weyl_minipoly :0.004sec(0.000402sec)
-- generic_bfct_and_gr :0.004sec(0.0004091sec)
generic bfct : [[1,1],[s,1],[s-2,1]]
S0 : 2
B_{S0} length : 3
-- fctr(BF) + base :0sec(0.000891sec)
-- integration_ideal_internal :0sec(0.000793sec)
[-27*x^3*dx^3-54*x^2*dx^2+(4*x^3+3*x)*dx+4*x^2-3,
27*x^2*dx^4+135*x*dx^3+(-4*x^2+105)*dx^2-16*x*dx-8]
```

また、以下は我々のアルゴリズムによって計算した結果である。

```
[1567] B = [dt+(3*t^2-1)*x,dx+t^3-t];
[dt+(3*t^2-1)*x,dx+t^3-t]
[1568] INT=integration_ideal(B,[t,x],[dt,dx],[1,0]|inhomo=1);
-- nd_weyl_gr :0.004001sec(0.004164sec)
-- weyl_minipoly :0sec(0.002094sec)
-- generic_bfct_and_gr :0.008002sec(0.007762sec)
generic bfct : [[1,1],[s,1],[s-2,1]]
S0 : 2
B_{S0} length : 3
-- fctr(BF) + base :0.004sec(0.003639sec)
-- integration_ideal_internal :0.008sec(0.005342sec)
[[-27*x^2*dx^2-27*x*dx+4*x^2+3],[[dt,-9*t*x*dx+(-6*t^2+4)*x+3*t]],1]]
```

我々のアルゴリズムの出力からは $v(x)$ の満たす非斉次微分方程式系として

$$(-27x^2\partial_x^2 - 27x\partial_x + 4x^2 + 3) \cdot v(x) = -4x \quad (3)$$

が得られる。一方、 $-4x$ の零化イデアルは $\langle x\partial_x - 1, \partial_x^2 \rangle$ であるから、この 2 つの生成元を (3) の両辺に左から作用させると $v(x)$ の満たす斉次微分方程式系が得られ、OST アルゴリズムによる出力と一致する。しかし、OST アルゴリズムの出力からは右辺の非斉次部分を直接計算することは出来ない。つまり、我々のアルゴリズムは OST アルゴリズムよりも多くの情報を含んでいる。

表 1 は

$$v_n(x) = \int_0^\infty u_k(t, x) dt, \quad \bar{v}_n(x) = \int_0^1 u_k(t, x) dt \quad (k = 1, \dots, 4)$$

$$\text{ただし, } u_k(t, x) = \exp\left(-tx \prod_{i=1}^k (t^2 - i^2)\right)$$

に対して、我々のアルゴリズム (Algorithm 2) と OST アルゴリズム a, b を適用したときの計算時間を表している。また、比較の為に アルゴリズム 1 の計算時間も示した。計算環境は CPU:Xeon X5570 (2.93GHz), Memory:48GB である。記号 † は v_i の結果を利用できるので、改めて計算する必要がないことを示している。

	Alg 2			OST a	OST b			Alg 1
input	int	Ann	total	total	prod	int	total	total
v_1	0.0042	0.0014	0.0056	0.0062	0.11	0.012	0.12	0.0039
v_2	0.15	0.019	0.17	0.25	5.10	0.16	5.26	0.075
v_3	19.91	0.45	20.36	96.14	24.54	95.24	119.8	13.58
v_4	26724	28.33	26752	> 1 day	1726	> 1 day	—	24003
$\bar{v}_1$	†	0.0015	0.0057	0.0071	0.47	0.0050	0.48	n/a
$\bar{v}_2$	†	0.027	0.18	1.56	18230	1.19	18231	n/a
$\bar{v}_3$	†	1.62	21.53	3769	848	2802	3650	n/a
$\bar{v}_4$	†	294	27018	> 1 day	16231	> 1 day	—	n/a

表 1: OST アルゴリズムとの計算時間 (sec) の比較

計算効率の面では、アルゴリズム 2 はアルゴリズム 1 と比べて、元々計算しているグレブナ基底の中間情報を用いて非斉次部分を構成するコストが増える。一方、OST アルゴリズムは、非積分関数に Heaviside 関数をかけることで積分アルゴリズムの入力イデアルが複雑になり計算コストが増加する。コストの増加分は、単に微分作用素の積や和をとるだけの計算であるアルゴリズム 2 の方が小さいが、OST アルゴリズムの出力のように斉次微分方程式を導出する為には、非斉次部分の零化イデアルの計算が必要である。零化イデアルの計算は重たい場合もあり、そのコストも含めると両者のどちらが計算に有利であるかは問題に依存する。

同一の非積分関数について積分区間を変えたものの満たす微分方程式を得たい場合、OST アルゴリズムでは初めから計算をやり直す必要があるが、我々のアルゴリズムは積分区間に依存しないのでその必要はない。

5 適用例

この節では、我々のアルゴリズムの適用例を挙げる。また、それらを Risa/Asir において実行したときの入力や出力等の経過も掲載した。

例 7 (不完全 Gauss 超幾何積分)

$$\int_p^q t^{b-1}(1-t)^{c-b-1}(1-xt)^{-a} dt$$

の満たす微分方程式系を求める。被積分関数 $f(x, t) = t^{b-1}(1-t)^{c-b-1}(1-xt)^{-a}$ の $K(x, t, \partial_x, \partial_t)$ での零化イデアルを計算すれば、

$$I_f = \langle (-x^2 + x)\partial_x^2 + ((-t+1)\partial_t + (-a-b-1)x + c-1)\partial_x - ab, \\ (-t+1)x\partial_x + (t^2-t)\partial_t + (-c+2)t + b-1, (tx-1)\partial_x + at \rangle$$

となる。(Oaku の多項式の零化イデアルの計算アルゴリズム) これに対し t についての積分イデアルを計算してやれば、

$$\langle (-x^2 + x)\partial_x^2 + ((-a-b-1)x + c)\partial_x - ab \rangle$$

が得られ, これは Gauss の超幾何微分方程式に出てくる微分作用素である. この微分作用素 P の非斉次部分は

$$\partial_t(-t+1)\partial_x$$

となり, P を上の積分に作用させると

$$P \cdot \int_p^q f(x,t)dt = \int_p^q (\partial_t(t-1)\partial_x) \cdot f(x,t)dt = \left[(t-1) \frac{\partial f}{\partial x}(x,t) \right]_p^q$$

```
[1543] L=ndbf.ann_n([t,1-t,1-x*t]);
[(-x^2+x)*dx^2+((-t+1)*dt+(s2-s0-2)*x+s1+s0+1)*dx+(s0+1)*s2,
(-t+1)*x*dx+(t^2-t)*dt+(-s1-s0)*t+s0,(t*x-1)*dx-s2*t]
[1544] L=map(subst, L, s0, b-1,s1,c-b-1,s2,-a);
[(-x^2+x)*dx^2+((-t+1)*dt+(-a-b-1)*x+c-1)*dx-b*a,
(-t+1)*x*dx+(t^2-t)*dt+(-c+2)*t+b-1,(t*x-1)*dx+a*t]
[1545] integration_ideal(L,[t,x],[dt,dx],[1,0]|param=[a,b,c],
s0=0,inhomo=1);
-- nd_weyl_gr :0sec(0.000319sec)
skip computation of generic b-function
-- generic_bfct_and_gr :0sec(0.0004299sec)
generic bfct : [[1,1],[s,1]]
S0 : 0
B_{S0} length : 1
-- fctr(BF) + base :0sec(0.0001419sec)
-- integration_ideal_internal :0sec(0.000402sec)
[[(-x^2+x)*dx^2+((-a-b-1)*x+c)*dx-b*a],[[dt,(-t+1)*dx]],1]]
```

例 8 (指数関数の積分)

$$\int_0^\infty e^{-t-xt^3} dt$$

被積分関数 $f(t,x) = e^{-t-xt^3}$ の $K\langle t,s,\partial_t,\partial_s \rangle$ での零化イデアルは

$$I_f = \langle \partial_t + 1 + 3xt^2, \partial_x + t^3 \rangle$$

このイデアルに対して t についての積分アルゴリズムを適用すると, 積分イデアルは

$$\langle 27x^3\partial_x^2 + 54x^2\partial_x + 6x + 1 \rangle$$

となり, この微分作用素 P の非斉次部分は

$$-\partial_t(\partial_t^2 + 3\partial_t + 3)$$

となる. 微分作用素 P を積分に作用させると

$$\begin{aligned} P \cdot \int_0^\infty e^{-t-xt^3} dt &= - \int_0^\infty (\partial_t(\partial_t^2 + 3\partial_t + 3)) \cdot e^{-t-xt^3} dt \\ &= - \left[(\partial_t^2 + 3\partial_t + 3) \cdot e^{-t-xt^3} \right]_0^\infty \\ &= - \left[(-6xt + (1 + 3xt^2)^2 - 3 - 9xt^2 + 3) e^{-t-xt^3} \right]_0^\infty \\ &= \left[(-9x^2t^4 + 3xt^2 + 6xt - 1) e^{-t-xt^3} \right]_0^\infty = 1 \end{aligned}$$

```
[1553] integration_exp(1,-t-x*t^3,[t,x],[dt,dx],[1,0]|inhomo=1);
-- nd_weyl_gr :0sec(0.000427sec)
-- weyl_minipoly :0sec(0.0002291sec)
-- generic_bfct_and_gr :0sec(0.000246sec)
generic bfct : [[1,1],[s,1]]
S0 : 0
B_{S0} length : 1
-- fctr(BF) + base :0sec(0.000165sec)
-- integration_ideal_internal :0sec(0.000236sec)
[[27*x^3*dx^2+54*x^2*dx+6*x+1],[[[[dt,dt^2+3*dt+3]],1]]]
```

例 9 (ベキ関数の積分)

$$\int_a^b t^s(1-t)^s dt$$

被積分関数 $f(t, s) = t^s(1-t)^s$ の $K\langle t, s, \partial_t, \partial_s \rangle$ での零化イデアルは

$$I_f = \langle (-t^2 + t)\partial_t + (2t - 1)s \rangle$$

p を s についてのシフト作用素 (i.e. $p \cdot (t^s(1-t)^s) = t^{s+1}(1-t)^{s+1}$) とえば,

$$(p - t(1 - t)) \cdot f(t, s) = 0$$

が成り立つ. 零化イデアル I_f を Mellin 変換 ($-p\partial_p - 1 \leftrightarrow s$) で $K\langle t, p, \partial_t, \partial_p \rangle$ のイデアルに変換すれば,

$$\langle (-t^2 + t)\partial_t + (2t - 1)(-p\partial_p - 1) \rangle$$

結局,

$$I'_f = \langle p - t(1 - t), (-t^2 + t)\partial_t + (2t - 1)(-p\partial_p - 1) \rangle$$

を被積分関数の満たす微分方程式系として, 積分アルゴリズムを適用する. 積分イデアルは

$$\langle (-4p^2 + p)\partial_p - 2p \rangle$$

となり, この微分作用素 P の非斉次部分は

$$\partial_t(-4p^2 + (-4t^2 + 2t + 1)p)$$

となる. 微分作用素 P を積分に作用させると

$$\begin{aligned} P \cdot \int_a^b t^s(1-t)^s dt &= \int_a^b (\partial_t(4p^2 + (4t^2 + 2t + 1)p)) \cdot t^s(1-t)^s dt \\ &= [(4p^2 + (4t^2 + 2t + 1)p) \cdot t^s(1-t)^s]_a^b \\ &= [4t^{s+2}(1-t)^{s+2} + (4t^2 + 2t + 1)t^{s+1}(1-t)^{s+1}]_a^b \\ &= [(6t + 1)t^{s+1}(1-t)^{s+1}]_a^b \end{aligned}$$

```

[1554] L=ann(t*(1-t));
[(-t^2+t)*dt+(2*t-1)*s]
[1555] L=map(subst, L, s, -p*dp-1);
[(-2*t+1)*p*dp+(-t^2+t)*dt-2*t+1]
[1556] L=cons(p-t*(1-t), L);
[p+t^2-t, (-2*t+1)*p*dp+(-t^2+t)*dt-2*t+1]
[1557] integration_ideal(L, [t,p], [dt,dp], [1,0] | inhom=1);
-- nd_weyl_gr :0sec(0.00033sec)
-- weyl_minipoly :0sec(0.0003438sec)
-- generic_bfct_and_gr :0sec(0.0003538sec)
generic bfct : [[1,1],[s,1],[s-1,1]]
S0 : 1
B_{S0} length : 2
-- fctr(BF) + base :0sec(0.0004201sec)
-- integration_ideal_internal :0sec(0.000452sec)
[[(-4*p^2+p)*dp-2*p],[[dt,-4*p^2+(-4*t^2+2*t+1)*p]],1]]

```

例 10 (Bessel 関数)

$$\int_0^{\infty} e^{-ax} J_0(bx) dx \left(= \frac{1}{\sqrt{a^2 + b^2}} \right)$$

の積分区間を $[c, d]$ にした積分を考える.

$$F(a, b) = \int_c^d e^{-ax} J_0(bx) dx$$

0 次 Bessel 関数 $J_0(bx)$ を零化するホロノミックイデアルは, $I = \langle b\partial_b^2 + \partial_b + bx^2, x\partial_x^2 + \partial_x + xb^2 \rangle$ で与えられる (ν 次 Bessel 関数 $J_\nu(x)$ は $(\partial_x^2 + \frac{1}{x}\partial_x + (1 - \frac{\nu^2}{x^2})) \cdot J_\nu(x) = 0$ を満たすことから). e^{-ax} を零化するホロノミックイデアルは $J = \langle \partial_x + a, \partial_a + x \rangle$ で与えられる. $K\langle a, b, x, \partial_a, \partial_b, \partial_x \rangle$ で考えるので, I, J を改めて

$$I' = \langle \partial_a, b\partial_b^2 + \partial_b + bx^2, x\partial_x^2 + \partial_x + xb^2 \rangle, J' = J = \langle \partial_x + a, \partial_a + x \rangle$$

とおく. I', J' より $e^{-ax} J_0(bx)$ を零化するホロノミックイデアルは

$$K = \langle b\partial_b\partial_x^2 + 2ab\partial_b\partial_x + (a^2b + b^3)\partial_b + b^2, -b\partial_b\partial_x + b^2\partial_a - ab\partial_b, -b\partial_a^2 - b\partial_b^2 - \partial_b, \\ \partial_a\partial_x + a\partial_a + b\partial_b + 1, -\partial_a - x \rangle$$

である. K の x に関する積分イデアルは

$$\langle P_1 = a\partial_a + b\partial_b + 1, P_2 = -b\partial_a^2 - b\partial_b^2 - \partial_b, P_3 = b^2\partial_a - ab\partial_b, P_4 = (-a^2b - b^3)\partial_b - b^2 \rangle$$

であり, P_i の非斉次部分 Q_i は

$$Q_1 = \partial_x(-\partial_a), Q_2 = 0, Q_3 = \partial_x(b\partial_b), Q_4 = \partial_x(b^2\partial_a + ab\partial_b)$$

である.

```

[2073] I=[x*dx^2+dx+x*b^2,b*db^2+db+b*x^2];
[x*dx^2+dx+b^2*x,b*db^2+db+b*x^2]
[2074] J=[dx+a,da+x];
[dx+a,da+x]
[2075] I=cons(da,I);
[da,x*dx^2+dx+b^2*x,b*db^2+db+b*x^2]
[2076] J=cons(db,J);
[db,dx+a,da+x]
[2077] K=ann_mul(I,J,[a,b,x],[da,db,dx]);
[[da+da_,x*dx^2+(2*dx_*x+1)*dx+(b^2+dx_^2)*x+dx_,
b*db^2+(2*db_*b+1)*db+b*x^2+db_^2*b+db_,-db_,a-a_-dx_,x-x_-da_],
[a_,b_,x_,a,b,x],[da_,db_,dx_,da,db,dx],[1,1,1,0,0,0]]
option:[0,-1]
-- nd_weyl_gr :0.008sec(0.004608sec)
-- weyl_minipoly :0.004sec(0.01127sec)
-- generic_bfct_and_gr :0.012sec(0.01751sec)
generic bfct : [[1,1],[s,1]]
S0 : 0
B_{S0} length : 1
-- fctr(BF) + base :0sec(0.006942sec)
-- restriction_ideal_internal :0sec(0.0009649sec)
[b*db*dx^2+2*b*a*db*dx+(b*a^2+b^3)*db+b^2,-b*db*dx+b^2*da-b*a*db,
-b*da^2-b*db^2-db,da*dx+a*da+b*db+1,-da-x]
[2079] integration_ideal(K,[x,a,b],[dx,da,db],[1,0,0]|inhomo=1);
-- nd_weyl_gr :0sec(0.000289sec)
-- weyl_minipoly :0sec(0.000268sec)
-- generic_bfct_and_gr :0.004sec(0.0009289sec)
generic bfct : [[1,1],[s,1]]
S0 : 0
B_{S0} length : 1
-- fctr(BF) + base :0sec(0.0002501sec)
-- integration_ideal_internal :0.004sec(0.01079sec)
[[a*da+b*db+1,-b*da^2-b*db^2-db,b^2*da-b*a*db,(-b*a^2-b^3)*db-b^2],
[[[dx,-da]],1],[[dx,0]],1],[[dx,b*db]],1],[[dx,b^2*da+b*a*db]],1]]

```

積分 $F(b) = \int_c^d J_0(bx)dx$ について考える. 積分イデアルは $\langle P = -b\partial_b - 1 \rangle$ となり, P の非斉次部分は

$$Q = \partial_x(bx^2\partial_b + x^2)\partial_x - b^2x\partial_b^2 - 2bx\partial_b - x$$

だから $F(b)$ は

$$P \cdot F(b) = [Q \cdot J_0(bx)]_{x=c}^{x=d}$$

$Q \cdot J_0(bx) = -xJ_0(bx)$ だから

$$P \cdot F(b) = -dJ_0(bd) + cJ_0(bc)$$

となる.

```

[2084] I;
[da,x*dx^2+dx+b^2*x,b*db^2+db+b*x^2]
[2085] II=cdr(I);
[x*dx^2+dx+b^2*x,b*db^2+db+b*x^2]
[2086] integration_ideal(II,[x,b],[dx,db],[1,0]|inhomo=1);
-- nd_weyl_gr :0sec(0.000536sec)
-- weyl_minipoly :0sec(0.0007172sec)
-- generic_bfct_and_gr :0sec(0.001731sec)
generic bfct : [[1,1],[s,2],[s-1,1]]
S0 : 1
B_{S0} length : 2
-- fctr(BF) + base :0sec(0.01755sec)
-- integration_ideal_internal :0.004sec(0.001466sec)
[[-b*db-1],[[[[dx,(b*x^2*db+x^2)*dx-b^2*x*db^2-2*b*x*db-x]],1]]]

```

参 考 文 献

- [1] G. Almkvist, D. Zeilberger, The method of differentiating under the integral sign, *Journal of Symbolic Computation* **10**, 571-591, 1990.
- [2] M. Apagodu, D. Zeilberger, Multi-variable Zeilberger and Almkvist-Zeilberger algorithms and the sharpening of Wilf-Zeilberger theory, *Advances in Applied Mathematics* **37**, 139-152, 2006.
- [3] F. Chyzak, Gröbner Bases, Symbolic Summation and Symbolic Integration, *London Mathematics Lecture Notes Series*, vol.251, 32-60, 1998.
- [4] F. Chyzak, An Extension of Zeilberger's Fast Algorithm to General Holonomic Functions, *Discrete Mathematics* **217**, 115-134, 2000.
- [5] F. Chyzak, B. Salvy, Non-commutative Elimination in Ore Algebras Proves Multivariate Holonomic Identities, *Journal of Symbolic Computation* **26**, 187-227, 1998.
- [6] T. Oaku, Algorithms for b -functions, restrictions, and algebraic local cohomology groups of D -modules, *Advances in Applied Mathematics* **19**, 61-105, 1997.
- [7] T. Oaku, Y. Shiraki, N. Takayama, Algebraic Algorithm for D -modules and numerical analysis, *Computer mathematics (Proceedings of ASCM 2003)*, 23-39, *Lecture Notes Ser. Comput.*, 10, World Sci. Publ., River Edge, NJ, 2003.
- [8] M. Saito, B. Sturmfels, N. Takayama, *Gröbner Deformations of Hypergeometric Differential Equations*, Springer, 2000.
- [9] N. Takayama, An Approach to the Zero Recognition Problem by Buchberger Algorithm, *Journal of Symbolic Computation* **14**, 265-282, 1992.
- [10] A. Tefera, MultInt, a Maple package for multiple integration by the WZ method, *Journal of Symbolic Computation* **34**, 329-353, 2002.
- [11] M. Noro, et al: Risa/Asir, <http://www.math.kobe-u.ac.jp/Asir>

- [12] F. Chyzak: Mgfund,
<http://algo.inria.fr/chyzak/mgfund.html>
- [13] C. Koutschan: HolonomicFunctions,
<http://www.risc.jku.at/research/combinat/software/HolonomicFunctions/>
- [14] H. Nakayama, K. Nishiyama: nk_restriction.rr,
http://www.math.kobe-u.ac.jp/~nakayama/nk_restriction.rr