数理解析研究所講究録
第 1785 巻 2012 年 146-153

和の満たす非斉次差分方程式系を与えるアルゴリズム

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 UN IVERSITY *

1 Introduction

2 変数 x, t の多項式係数微分作用素環 (Weyl 代数) を $D = \mathbb{Q}\langle t, x, \partial_t, \partial_x\rangle$ とし, その部分環である 1 変数 x の微分作用素環を $D' = \mathbb{Q}\langle x, \partial_x\rangle$ とおく. ここで, ∂_x は x についての微分作用素を表す. このとき, 左 D イデアル I の t についての制限イデアル J とは, $J = (I + tD) \cap D'$ なる左 D' イデアルのことであり, 制限イデアルを計算するアルゴリズムは Oaku ([5]) により与えられた.

我々はこの制限アルゴリズムを変更することにより, 制限イデアル 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, $P = P_0 + tP_1$ となる $P_0 \in I, P_1 \in D$ を計算するアルゴリズムを与えた ([4]). このような P_1 を P の非斉次部分と呼ぶ. 制限イデアルと Mellin 変換を利用して, パラメータを含む和の満たす斉次差分方程式系を計算するアルゴリズムが, Oaku, Shiraki, Takayama ([6]) により与えられている. これをもとにして, 制限イデアルの非斉次部分を計算するアルゴリズムを使い, 我々はパラメータを含む和の満たす非斉次差分方程式系を計算するアルゴリズムを得た.

例えば, 和 $F(n) = \sum_{k=a}^{b} \binom{n}{k}$ ($\binom{n}{k}$ は 2 項係数) の満たす非斉次差分方程式系を考えよう. 被和関数 $f(k, n) = \binom{n}{k}$ の満たす差分方程式系は,

$$((n-k+1)E_n - (n+1)) \cdot f(k,n) = 0, \quad ((k+1)E_k - (n-k)) \cdot f(k,n) = 0$$

である. ここで, E_k, E_n は k, n についてのシフト演算子で, $f(k, n)$ に作用させると

$$E_k \cdot f(k,n) = f(k+1,n), \quad E_n \cdot f(k,n) = f(k,n+1)$$

となるものである. この差分方程式系を入力として, 我々のアルゴリズムを適用すると, 次の非斉次差分方程式系が出力される.

$$(E_n - 2) \cdot F(n) = -\binom{n+1}{b} + \binom{n}{b+1} + \binom{n+1}{a} - \binom{n}{a}$$

制限イデアルの非斉次部分を使うことで, 上の式の非斉次部分 (右辺) が計算できるようになる.

この論文では, 制限イデアルの非斉次部分の計算アルゴリズムの詳細と, その応用である和の満たす非斉次差分方程式系を計算する方法ついて述べる. 和の満たす非斉次差分方程式系を計算する他のアルゴリズムとして, 被和関数が超幾何的な関数 (すなわち各変数について 1 階差分方程式を満たす関数) なものをを入力とする Zeilberger アルゴリズム (creative telescoping) ([9], [10]) や, 被和関数がホロノミック関数なもの

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

を入力とする Chyzak アルゴリズム ([2]) などがある. 我々のアルゴリズムは入力は Chyzak アルゴリズムと同じ範囲であるが計算方法が異なる.

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`, `ost_sum.rr` ([14]) として公開している.

2 D 加群の制限アルゴリズム

和の非斉次差分方程式系を計算する時に必要となる D 加群の制限アルゴリズムについて概要を述べる. n 変数多項式係数微分作用素環を

$$D = \mathbb{Q}\langle x_1, \ldots, x_m, x_{m+1}, \ldots x_n, \partial_1, \ldots, \partial_m, \partial_{m+1}, \ldots, \partial_n \rangle$$

とし, その部分環である $n-m$ 変数多項式係数微分作用素環を

$$D' = \mathbb{Q}\langle x_{m+1}, \ldots, x_n, \partial_{m+1}, \ldots, \partial_n \rangle$$

とおく。ホロノミック左 D イデアル I の $x_1, \cdots, x_m$ についての制限イデアル J とは

$$J = (I + x_1 D + \cdots + x_m D) \cap D'$$

なる左 D' イデアルのことである. ホロノミック左 D イデアル I を関数 $f(x_1, \ldots, x_n)$ の満たす微分方程式系とした時, その $x_1, \ldots, x_m$ についての制限イデアル I は関数 $f(0, \ldots, 0, x_{m+1}, \ldots, x_n)$ の満たす微分方程式系に対応している. 制限イデアルについて, D におけるグレブナー基底を用いた計算方法が知られている.

Algorithm 1 (D **加群の制限アルゴリズム** [5], [7])

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 $w = (w_1, \ldots, w_m, w_{m+1}, \ldots, w_n)$ で, $w_1, \ldots, w_m > 0, w_{m+1} = \cdots = w_n = 0$ を満たすもの.

出力 : I の $x_1, \ldots, x_m$ についての制限イデアルの生成元.

1. I の w についての制限加群 $M = D/(I + x_1 D + \cdots + x_m D)$ の計算. 具体的には次のように行う.
   (a) I の $<_{(-w,w)}$ についてのグレブナー基底 $\{h_1, \cdots, h_l\}$ を計算.
   (b) 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 がなければ $M = 0$ を返す.
   s_0 がある場合,
   $m_i = \mathrm{ord}_{(-w,w)}(h_i)$,
   $\mathcal{B}_d = \{\partial_1^{i_1} \cdots \partial_m^{i_m} \mid i_1 w_1 + \cdots + i_m w_m \le d\}$,
   $r = \#\mathcal{B}_{s_0} (= \#\{(i_1, \ldots, i_m) \mid i_1 w_1 + \cdots + i_m w_m \le s_0\})$ とおく.
   (d) $\tilde{\mathcal{B}} = \bigcup_{i=1}^{l} \{\tilde{h}_{i\beta} := \partial^\beta h_i = \sum c_{u,v} x^u \partial^v \mid \partial^\beta \in \mathcal{B}_{s_0 - m_i}\}$ とし,
   $\mathcal{B} = \{h_{i\beta} := \tilde{h}_{i\beta}|_{x_1 = \cdots = x_m = 0} \mid \tilde{h}_{i\beta} \in \tilde{\mathcal{B}}\}$ を返す.
   (e) $\mathcal{B}$ を左 D' 加群 $(D')^r$ (基底は $\mathcal{B}_{s_0}$) の元とみなし, それで生成される左 D' 加群を M とおく.

2. M に対し, ∂^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, グレブナー基底を計算する. その生成元の中で, ∂^0 に対応する成分以外が全て 0 になっている元たちを取り出し, その元の ∂^0 成分からなる集合が I の制限イデアルの生成系.

Theorem 1 (制限イデアルの非斉次部分の計算 [4])
ホロノミックな 左 D イデアル I の制限イデアルを $J \subset D'$ とする. このとき, 任意の $P \in J$ に対して

$$P - x_1 P_1 - \cdots - x_m P_m \in I \tag{1}$$

を満たす微分作用素 $P_1, \ldots, P_m \in D$ を計算するアルゴリズムが存在する. この $P_1, \ldots, P_m$ を, 制限イデアルの元 P の非斉次部分と呼ぶ.

(proof) Algorithm 1 を適用して制限イデアル J の生成系 $\{g_1, \ldots, g_t\}$ を得る. この各生成元 g_j $(1 \le j \le t)$ に対して示せば十分である. Algorithm 1 の Step 2. より, $q_{ji\beta} \in D$ を用いて $g_j = \sum_{i,\beta} q_{ji\beta} h_{i\beta}$ と表される. また, この $q_{ji\beta}$ はグレブナー基底計算における履歴を参照することで計算できる. 従って,

$$\begin{aligned} I \ni \sum_{i,\beta} q_{ji\beta} \tilde{h}_{i\beta} &= g_j - \left(g_j - \sum_{i,\beta} q_{ji\beta} \tilde{h}_{i\beta} \right) \\ &= g_j - \sum_{i,\beta} q_{ji\beta} \left(h_{i\beta} - \tilde{h}_{i\beta}) \right) \\ &= g_j - \sum_{i,\beta} q_{ji\beta} \left(\tilde{h}_{i\beta}|_{x_1 = \cdots = x_m = 0} - \tilde{h}_{i\beta} \right) \end{aligned}$$

と変形できる. ここで, $\tilde{h}_{i\beta}|_{x_1=\cdots=x_m=0} - \tilde{h}_{i\beta}$ の各項は $x_1, \ldots, x_m$ のいずれかで左から割り切れるので,

$$\sum_{i,\beta} q_{ji\beta} (\tilde{h}_{i\beta}|_{x_1=\cdots=x_m=0} - \tilde{h}_{i\beta}) = \sum_{i=1}^{m} x_i p_{ij} \quad (p_{ij} \in D)$$

と書き直すことができる. ■

Algorithm 2 (制限イデアルの非斉次部分の計算, [4])

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 $w = (w_1, \ldots, w_m, w_{m+1}, \ldots, w_n)$ で, $w_1, \ldots, w_m > 0, w_{m+1} = \cdots = w_n = 0$ を満たすもの.

出力 : I の $x_1, \ldots, x_m$ についての制限イデアルの生成元 $\{g_1, \ldots, g_t\}$.
各生成元 g_j $(1 \le j \le t)$ に対して $g_j - \sum_{i=1}^{m} x_i p_{ij} \in I$ を満たす $p_{ij} \in D$.

1. *Algorithm 1* を適用.
2. $g_j = \sum_{i,\beta} q_{ji\beta} h_{i\beta}$ を満たす $q_{ji\beta} \in D$ を計算.
3. $R_j = g_j - \sum_{i,\beta} q_{ji\beta} \tilde{h}_{i\beta}$ を $R_j = \sum_{i=1}^{m} x_i p_{ij}$ と書き直したときの $p_{ij} \in D$ を出力.

3 和の満たす非斉次差分方程式系を与えるアルゴリズム

差分作用素環

$$S = \mathbb{Q}\langle k_1, \ldots, k_n, E_1, \ldots, E_n, E_1^{-1}, \ldots, E_n^{-1}\rangle$$

とおく. ここで, E_i は変数 k_i のシフト演算子を表している. また, その部分環

$$S' = \mathbb{Q}\langle k_2, \ldots, k_n, E_2, \ldots, E_n, E_2^{-1}, \ldots, E_n^{-1}\rangle$$

とおく. 差分作用素を微分作用素に写す Mellin 変換 $\mathcal{M}$ を次のように定義する.

$$\text{環準同型 } \mathcal{M} : S \to D, \quad E_i \mapsto x_i, \quad k_i \mapsto -x_i\partial_i, \quad -(k_i - 1)E_i^{-1} \mapsto \partial_i \quad (1 \le i \le n)$$

この変換により, 差分作用素環 S は微分作用素環 D に同型である.

関数 $f(k_1, k_2, \ldots, k_n)$ と, それの満たす差分方程式系 I (左 S イデアル) が与えられているとする. さらに, I を Mellin 変換した左 D イデアル $J = \mathcal{M}(I)$ が, ホロノミックであると仮定する. ここで考えるのは, 和 $F(k_2, \ldots, k_n) = \sum_{k_1=a}^{b} f(k_1, k_2, \cdots, k_n)$ の満たす非斉次差分方程式系を計算することである. そのためには, 差分作用素環 S のイデアル

$$J = (I + (E_1 - 1)S) \cap S'$$

が計算できればよい. これを和イデアルと呼ぶことにする. 和イデアル J の元 P について, $P = P_0 + (E_1 - 1)P_1, P_0 \in I, P_1 \in S$ と表すことができ, これを和 $F(k_2, \ldots, k_n)$ に作用させれば,

$$\begin{aligned}
P \cdot F(k_2, \ldots, k_n) &= P \cdot \sum_{k_1=a}^{b} f(k_1, k_2, \ldots, k_n) = \sum_{k_1=a}^{b} P \cdot f(k_1, k_2, \ldots, k_n) \\
&= \sum_{k_1=a}^{b} (E_1 - 1)P_1 \cdot f(k_1, k_2, \ldots, k_n) \\
&= \sum_{k_1=a}^{b} ((P_1 \cdot f)(k_1 + 1, k_2, \ldots, k_n) - (P_1 \cdot f)(k_1, k_2, \ldots, k_n)) \\
&= (P_1 \cdot f)(b + 1, k_2, \ldots, k_n) - (P_1 \cdot f)(a, k_2, \ldots, k_n)
\end{aligned}$$

となり, 和 $F(k_2, \ldots, k_n)$ の満たす非斉次差分方程式系が得られる. 和イデアル J を Mellin 変換すれば,

$$\mathcal{M}(J) = (\mathcal{M}(I) + (x_1 - 1)D) \cap D'$$

であるから, I についての和イデアルを計算するには, D イデアル $\mathcal{M}(I)$ の $x_1 = 1$ についての制限イデアルが計算できればよい.

このような D 加群の制限アルゴリズムと Mellin 変換を使って, 和イデアルを計算する方法は [6] で与えられている. この方法では和イデアルの生成系のみを計算し, 和の満たす斉次微分方程式系を与えている. ここでは, [6] のアルゴリズムをもとに, 制限イデアルの非斉次部分を計算するアルゴリズムを使い, 和の満たす非斉次微分方程式系を計算する方法を紹介する.

Algorithm 3 (和の非斉次差分方程式系を与えるアルゴリズム)

入力 : 関数 $f(k_1, k_2, \ldots, k_n)$ の満たす差分方程式系 I *(左 S イデアル)*

出力 : 和 $F(k_2, \ldots, k_n) = \sum_{k_1=a}^{b} f(k_1, k_2, \cdots, k_n)$ の満たす非斉次差分方程式系

1. 左 S イデアル I を *Mellin* 変換して, 左 D イデアル $\mathcal{M}(I)$ を得る.

2. 左 D イデアル $\mathcal{M}(I)$ がホロノミックかどうかを判定. ホロノミックでなければ, 制限アルゴリズムを実行できるとは限らないので, ここで終了する.

3. 左 D イデアル $\mathcal{M}(I)$ に対して, 変数 x_1 について, $x_1 \to x_1 + 1$ と変数変換する.

4. 左 D イデアル $\mathcal{M}(I)$ の x_1 についての制限イデアル $J = D' \cdot \{P_1, \ldots, P_l\}$ と, 各生成元 $P_1, \ldots, P_l \in D'$ に対応する非斉次部分 $Q_1, \ldots, Q_l \in D$ を計算する. *(Algorithm 1, 2* を使う.*)*

5. 非斉次部分 $Q_1, \ldots, Q_l \in D$ に対して, 変数 x_1 について, $x_1 \to x_1 - 1$ と変数変換する.

6. D' の元 $P_1, \ldots, P_l$ とその非斉次部分 $Q_1, \ldots, Q_l$ を逆 *Mellin* 変換し, S' の元 $\mathcal{M}^{-1}(P_1), \ldots, \mathcal{M}^{-1}(P_l)$ と $\mathcal{M}^{-1}(Q_1), \ldots, \mathcal{M}^{-1}(Q_l)$ を得る.
$\mathcal{M}^{-1}(P_1), \ldots, \mathcal{M}^{-1}(P_l)$ が和イデアルの生成元であり, $\mathcal{M}^{-1}(Q_1), \ldots, \mathcal{M}^{-1}(Q_l)$ が非斉次部分に対応する. よって, 和 $F(k_2, \ldots, k_n)$ の満たす非斉次差分方程式系

$$\mathcal{M}^{-1}(P_i) \cdot F(k_2, \ldots, k_n) = \mathcal{M}^{-1}(Q_i) \cdot F(b+1, k_2, \ldots, k_n) - \mathcal{M}^{-1}(Q_i) \cdot F(a, k_2, \ldots, k_n)$$

が得られた.

Example 1 (和の非斉次差分方程式系を与えるアルゴリズムの計算例)

和 $F(n) = \sum_{k=a}^{b} \binom{n}{k}$ *(*$\binom{n}{k}$ は *2* 項係数*)* の満たす非斉次差分方程式系を計算する. ここでは,

$$S = \mathbb{Q}\langle k, n, E_k, E_n, E_k^{-1}, E_n^{-1} \rangle, \quad D = \mathbb{Q}\langle x, y, \partial_x, \partial_y \rangle$$

で計算を行う.

被和関数 $f(k, n) = \binom{n}{k}$ の満たす差分方程式系は,

$$((n-k+1)E_n - (n+1)) \cdot f(k,n) = 0, \quad ((k+1)E_k - (n-k)) \cdot f(k,n) = 0$$

である. この *2* つの差分作用素から生成されるイデアルを

$$I = \langle (n-k+1)E_n - (n+1), (k+1)E_k - (n-k) \rangle$$

とおき, *Algorithm 3* を実行する.

1. I を *Mellin* 変換

$$E_k \to x, k \to -x\partial_x, E_n \to y, n \to -y\partial_y$$

した結果, 左 D イデアル

$$\mathcal{M}(I) = \langle xy\partial_x + (-y^2 + y)\partial_y - 1, (-x^2 - x)\partial_x + y\partial_y \rangle$$

を得る.

2. $\mathcal{M}(I)$ はホロノミック D イデアル.

3. $\mathcal{M}(I)$ を $x \to x + 1$ と変数変換.

$$\mathcal{M}(I) = \langle (x+1)y\partial_x + (-y^2 + y)\partial_y - 1, -(x+1)(x+2)\partial_x + y\partial_y \rangle$$

4. $\mathcal{M}(I)$ の x についての制限イデアル J と非斉次部分を計算. 制限イデアルは $J=\langle(y^2-2y)\partial_y+2\rangle$ であり, $P=(y^2-2y)\partial_y+2$ の非斉次部分は $Q=-((y^2-y)\partial_y+1)$.

5. 非斉次部 Q を $x\to x-1$ と変数変換. $Q=-((y^2-y)\partial_y+1)$ となる.

6. P,Q の逆 *Mellin* 変換は,

$$\mathcal{M}^{-1}(P)=-(n+1)(E_n-2),\quad \mathcal{M}^{-1}(Q)=-(n+1)(-E_n+1)$$

よって

$$-(n+1)(E_n-2)\cdot F(n)=-(n+1)(-E_n+1)\cdot f(b+1,n)+(n+1)(-E_n+1)\cdot f(a,n)$$

さらに計算して,

$$(E_n-2)\cdot F(n)=-\binom{n+1}{b}+\binom{n}{b+1}+\binom{n+1}{a}-\binom{n}{a}$$

が得られる.

4 計算例

Example 2 (Algorithm 3 の実装) 関数 $f(k,n)$ を次の差分方程式系を満たすようなものとする.

$$f(k+10,n)-(k+n)f(k,n)=0,\quad f(k,n+10)-(k+n)f(k,n)=0$$

例えばこのような差分方程式系を満たすような関数として, 多重階乗関数 $m!10(m\in\mathbb{N})$ がある. ここで,

$$m!10=\begin{cases}(m-10)(m-20)\cdots 10 & ((m-10)\equiv 0 \pmod{10}),\\ (m-10)(m-20)\cdots 1 & ((m-10)\equiv 1 \pmod{10}),\\ \cdots & \\ (m-10)(m-20)\cdots 9 & ((m-10)\equiv 9 \pmod{10}).\end{cases}$$

と定義されるものである. $f(k,n)=(k+n-10)!10$ とすれば, これは上の差分方程式系を満たす.

和 $F(n)=\sum_{k=a}^{b}f(k,n)$ の満たす差分方程式系を *Algorithm 3* を使い求めれば, 非斉次差分方程式

$$\begin{aligned}-F(n+10)+F(n)&=\sum_{k=a}^{b}(E_k-1)\cdot(-f(k+9,n)-f(k+8,n)-\cdots-f(k,n))\\ &=-f(b+10,n)-\cdots-f(b+1,n)+f(a+9,n)+\cdots+f(a,n)\end{aligned}$$

が得られる. 我々は *Algorithm 3* を数式処理ソフト *Risa/Asir* に実装した *(*プログラム `ost_sum.rr` *[14])*. 実行例は次のようになる.

```
[1661] load("ost_sum.rr");
[1697] Id=[ek^10-(k+n), en^10-(k+n)];
[ek^10-k-n,en^10-k-n]
[1698] L=ost_sum(Id,[k,n],[ek,en],[x,y],[dx,dy],[1,0]);
... 計算経過略
[[-en^10+1],[[[[ek-1,-ek^9-ek^8-ek^7-ek^6-ek^5-ek^4-ek^3-ek^2-ek-1]],1]]]
0.004sec(0.006489sec)
[1699] show_eqs(L,[k,n],[ek,en],[n],[en]);
0 :
+(-1)*F(n+10)+(1)*F(n)
ek-1
+(-1)*f(k+9,n)+(-1)*f(k+8,n)+(-1)*f(k+7,n)+(-1)*f(k+6,n)+(-1)*f(k+5,n)+
(-1)*f(k+4,n)+(-1)*f(k+3,n)+(-1)*f(k+2,n)+(-1)*f(k+1,n)+(-1)*f(k,n)
```

この計算にはほとんど時間がかからず *0.01* 秒程度の計算で済む *(Linux machine with Xeon5450(3.00GHz) and 32 GB memory)*.

和の非斉次差分方程式系を求める他のアルゴリズムとして *Chyzak* アルゴリズム *([2])* がある．これは *Maple* 上に実装されており，パッケージ `Mgfun` のコマンド `creative_telescoping` *([12])* で実行できる．これを上の例に適用するとかなり時間がかかり *1000* 秒程度の時間がかかる．

この場合，*Chyzak* アルゴリズムは 100 個の一階線形差分方程式からなる差分方程式系の有理関数解を計算する必要があり，その計算に非常に時間がかかるためにこのようなことが起こった．被和関数が高階の差分方程式系を満たすときにこのようなことが起こる．

しかしながら，一般には *Chyzak* アルゴリズムの方が我々のアルゴリズムよりも高速である *(*様々な計算例については，`http://www.math.kobe-u.ac.jp/~nakayama/i-sum/index.html` を参照*)*. また *Zeilberger* アルゴリズム *([9, 10])* も和の非斉次差分方程式系を求める方法であり，非常に高速に計算できるが，基本的に計算できるのは超幾何和の場合でありこの場合は計算できない．

参 考 文 献

[1] F.Chyzak, Gröbner Bases, Symbolic Summation and Symbolic Integration, London Mathematics Lecture Notes Series, vol.251, 32-60, 1998.

[2] F.Chyzak, An Extension of Zeilberger's Fast Algorithm to General Holonomic Functions, Discrete Mathematics **217**, 115-134, 2000.

[3] F.Chyzak, B. Salvy, Non-commutative Elimination in Ore Algebras Proves Multivariate Holonomic Identities, Journal of Symbolic Computation **26**, 187-227, 1998.

[4] H.Nakayama, K.Nishiyama, An algorithm of computing inhomogeneous differential equations for definite integrals, Lecture Notes in Comput. Sci. 6327 (2010), 221–232.

[5] T.Oaku, Algorithms for b-functions, restrictions, and algebraic local cohomology groups of D-modules, *Advances in Applied Mathematics* **19**, 61–105, 1997.

[6] T.Oaku, Y.Shiraki, and 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.

[7] M.Saito, B.Sturmfels, and N.Takayama, *Gröbner Deformations of Hypergeometric Differential Equations*, Springer, 2000.

[8] N.Takayama, An Approach to the Zero Recognition Problem by Buchberger Algorithm, Journal of Symbolic Computation **14**, 265–282, 1992.

[9] D.Zeilberger: A fast algorithm for proving terminating hypergeometric identities, Discrete Math. **80**, 207–211, 1990.

[10] D.Zeilberger: The method of creative telescoping, Journal of Symbolic Computation **11**, 195–204, 1991.

[11] M.Noro, et al: `Risa/Asir`, `http://www.math.kobe-u.ac.jp/Asir`

[12] F.Chyzak: `Mgfun`, `http://algo.inria.fr/chyzak/mgfun.html`

[13] C.Koutschan: HolonomicFunctions,
http://www.risc.jku.at/research/combinat/software/HolonomicFunctions/

[14] H.Nakayama, K.Nishiyama: nk_restriction.rr, ost_sum.rr,
http://www.math.kobe-u.ac.jp/~nakayama/nk_restriction.rr
http://www.math.kobe-u.ac.jp/~nakayama/ost_sum.rr