

超幾何系の隣接関係式を求めるアルゴリズム

中山洋将 (日本大学), 高山信毅 (神戸大学*), 13:30–14:15 (11/18)

1. D 加群の同型とは?
2. 問題: Horn 型超幾何系の同型射 (隣接関係式) を求めよ.
3. 同型射の制限を求めるアルゴリズム.
4. 応用例: Horn 型超幾何系の隣接関係式を求める

プログラムは

<https://www.math.kobe-u.ac.jp/HOME/taka/2025/prog-rest>

スライドは Nobuki Takayama 検索

<https://www.math.kobe-u.ac.jp/HOME/taka/index-ja.html>

プレプリントは <https://arxiv.org/abs/2510.05104>

D 加群の同型とは?

$D = \mathbb{C}\langle x, \partial_x \rangle$, $\partial_x x = x\partial_x + 1$. 結合法則や分配法則 OK.

Example 1

$$L(a) = x\partial_x + a$$

$a \in \mathbb{Z}$ に対して左 D 加群 $D/L(a)$ を同型なもので分類せよ.

$$D/L(a) \ni \ell \xrightarrow{\cdot x} \ell x \in D/L(a+1)$$

は D 左準同型. $L(a)x = (x\partial_x + a)x = x(x\partial_x + a+1) = xL(a+1)$ なので well-defined.

$$D/L(a+1) \ni \ell \xrightarrow{\cdot \partial_x} \ell \partial_x \in D/L(a)$$

も D 左準同型. $L(a+1)\partial_x = (x\partial_x + a+1)\partial_x = \partial_x(x\partial_x + a)$ なので well-defined.

この2つの準同型を合成すると

$$D/L(a) \ni \ell \mapsto \ell x \partial_x \in D/L(a)$$

だが

$$x \partial_x \equiv -a \pmod{L(a)}$$

となる. $a \neq 0$ なら x, ∂_x は同型射を与える. しかし $a = 0$ では $D/L(0)$ と $D/L(1)$ をつなぐ同型射はつくれていない. $-a$ が超幾何 b 函数のもっとも単純な例.

∂_x, x は解の **contiguity relation** (隣接関係) を与える. つまり, $f = x^{-(a+1)}$ を $L(a+1) \bullet f = 0$ なる解とすると, xf は $L(a)$ の解. $f = x^{-a}$ を $L(a) \bullet f = 0$ なる解とすると, $\partial_x \bullet f$ は $L(a+1)$ の解. 以下, 超幾何 D -加群の上記のような同型射も **contiguity relation** とよぶことにする.

問題: パラメータが整数シフトした Horn 型超幾何系の同型射 (contiguity relation) を構成せよ

1. Gauss の超幾何系

$$L(a, b, c) = x(1-x)\partial_x^2 + (c - (a+b+1)x)\partial_x - ab \quad (1)$$

は (a, b, c) をパラメータに持つ. たとえば $D/L(a, b, c)$ と $D/L(a+1, b, c)$ の間の同型射を作れ (解の変換公式を作れ).

答え: $f(a+1, b, c) = (x\partial_x + a) \bullet f(a, b, c)$, $a(c-a-1) \neq 0$

- Appell F_1 の 2 変数超幾何系は (a, b, b', c) をパラメータに持つ.
- 一般に定理 1 の $A = (E, *)$ からきまる Horn 型超幾何系 (1,2,3 はこの場合に含まれる).

GKZ 超幾何系 (A 超幾何系)

GKZ 超幾何系¹ $H_A(\beta)$ は $d \times n$ の整数成分の行列 $A = (a_{ij})$ とパラメータベクトル $\beta \in \mathbb{C}^d$ で決まる n 変数 $x_1, \dots, x_n$ の holonomic 左イデアル.

n 変数の Weyl algebra: $D = D_n = \mathbb{C}\langle x_1, \dots, x_n, \partial_1, \dots, \partial_n \rangle$.

$$\sum_{j=1}^n a_{ij} x_j \partial_j - \beta_i, \quad i = 1, \dots, d$$

$$\partial^u - \partial^v, \quad Au = Av, u, v \in \mathbb{N}_0^n \quad (\text{toric ideal})$$

が Weyl algebra D_n で生成するイデアル $H_A(\beta)$ が GKZ 超幾何系.

¹1980 年代終わりに Gel'fand, Kapranov, Zelevinsky により導入された. toric ideal, regular triangulation 等の導入の動機でもある. 

定理 1.

1. E を $d \times d$ 単位行列とする. $A = (E, *)$ の時の GKZ 超幾何を $x_1 = \cdots = x_d = 1$ へ制限つまり

$$J := (H_A(\beta) + (x_1 - 1)D_n + \cdots + (x_d - 1)D_n) \\ \cap \mathbb{C}\langle x_{d+1}, \dots, x_n, \partial_{d+1}, \dots, \partial_n \rangle$$

を求めることで対応する Horn 型超幾何系が得られる^a. さらに $D/H_A(\beta)$ と $D/H_A(\beta')$ が D 加群として同型なら対応する Horn 型超幾何系も同型.

2. GKZ 超幾何としての同型射を制限して Horn 型超幾何の同型射を構成するアルゴリズムあり.

^a定義としてもよい

2 の実装は 2025-04-01-contiguity.rr.

Example 2

$$A = \left(\begin{array}{ccc|c} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right), \quad \beta = (c-1, -a, -b)^T, \text{ Gauss HG } {}_2F_1$$

$$A = \left(\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{array} \right), \quad \beta = (-a, -b, -b', c-1)^T, \text{ Appell } F_1$$

```
F=sm1.gkz([A=[[1,0,0,-1],[0,1,0,1],[0,0,1,1]],Beta=[c-1,-a,-b]]);
[[-x4*dx4+x1*dx1-c+1,x4*dx4+x2*dx2+a,x4*dx4+x3*dx3+b,
-dx1*dx4+dx2*dx3], ...]
import("nk_restriction.rr");
F0=base_replace(F[0],[[x1,x1+1],[x2,x2+1],[x3,x3+1]]);
nk_restriction.restriction_ideal(F0,[x1,x2,x3,x4],[dx1,dx2,dx3,dx4],
[1,1,1,0]);
[(x4^2-x4)*dx4^2+((a+b+1)*x4-c)*dx4+b*a] //Gauss HG
```

先行研究

1. Mutsumi Saito, Isomorphism classes of A -hypergeometric systems, *Compositio Mathematica* 128 (2001), 323–338.
2. Berkesch, Christine; Matusevich, Laura Felicia; Walther, Uli, Torus equivariant D -modules and hypergeometric systems, *Advances in Mathematics* 350 (2019), 1226–1266.

1 は GKZ 超幾何系 (A -超幾何系) について同型射の構成, 同型分類の方法を与えた.

GKZ 超幾何系と Gauss, Appell などの Horn 型超幾何との対応を 2 が与えた.

記号

$$u, v \in \mathbb{R}^n, u + v \geq 0,$$

$$\text{ord}_{(u,v)}(x^\alpha \partial^\beta) := \langle u, \alpha \rangle + \langle v, \beta \rangle.$$

$L \in D_n, w \in \mathbb{R}^n, \text{in}_{(-w,w)}(L)$: L の中で $\text{ord}_{(-w,w)}$ が最大である項の和.

例: $n = 1,$

$$\text{in}_{(-1,1)}(x^2 \partial_x^2 - x(x \partial_x + 1/2)) = x^2 \partial_x^2$$

$$\text{in}_{(-1,1)}(x^2 \partial_x^2 + 3 - x(x \partial_x + 1/2)) = x^2 \partial_x^2 + 3$$

定理 1 (1) の証明.

$\theta_i + \sum_{j>d} a_{ij}\theta_j - \beta_i \in H_A(\beta)$. 変数変換 $C : x_i \rightarrow x_i + 1$,
 $i = 1, \dots, d$ で

$$\theta_i + \partial_i + \sum_{j>d} a_{ij}\theta_j - \beta_i.$$

$(-w, w) = (-1_d, 0_{n-d}, 1_d, 0_{n-d})$ での initial term は ∂_i .
 $\partial_1, \dots, \partial_d \in \text{in}_{(-w, w)}(H_A(\beta)|_C)$.

$$\mathbb{C}[\theta_1 + \dots + \theta_d] \cap \text{in}_{(-w, w)}(H_A(\beta)|_C)$$

は $s = \theta_1 + \dots + \theta_d$, $\theta_i := x_i \partial_i$ を含む. つまり $b(s) = s$. よって
 大阿久の制限アルゴリズムにより

$$\frac{D_n}{(H_A(\beta) + (x_1 - 1)D_n + \dots + (x_d - 1)D_n)} \simeq D_{n-d}/J,$$

$$D_{n-d} = \mathbb{C}\langle x_{d+1}, \dots, x_n, \partial_{d+1}, \dots, \partial_n \rangle.$$

定理 1 (2) の証明に使うもの.

1. M.Saito (2001).

$$E\partial^u - b(s)\partial^v \in H_A(s) \quad (2)$$

$$E\partial^{u-v} \bullet f(s; x) - b(s) \bullet f(s; x) = 0 \quad (3)$$

2. Ir-reduction の導入 (left ideal と right ideal の和による reduction).

同型射の制限, 準備

$D = D_n$: n 変数の Weyl algebra.

I : D の左 holonomic イデアル

I の b -関数 $b(\theta_n)$: $\text{in}_{(-w,w)}(I) \cap \mathbb{C}[\theta_n]$ の生成元で monic なもの.

ここで $w = (0, \dots, 0, 1)$, $\theta_n = x_n \partial_n$.

k_0 : $b(s) = 0$ の最大の非負整数根.

$$F_{k_0} := \sum_{k=0}^{k_0} D_{n-1} \partial_n^k \simeq D_{n-1}^{k_0+1}. \quad (4)$$

T.Oaku の制限アルゴリズム² は $D/(I + x_n D) \simeq F_{k_0}/D_{n-1}G$ (as the left D_{n-1} module) となる $G \subset F_{k_0}$ を与える. キーとなる公式は

$$b(\theta_n) - r \in I, \text{ord}_{(-w,w)}(r) \leq -1$$

$$\partial_n^j b(\theta_n) = b(j) \partial_n^j + (b(\theta_n + j) - b(j)) \partial_n^j + \partial_n^j r \pmod{I}. \quad (5)$$

²<https://doi.org/10.1006/aama.1997.0527> Algorithms for b -function, restrictions, and algebraic local cohomology groups (1997).

同型射の制限, 一般

$f \in D$ の $D/(I + x_n D)$ における normal form :

1. f 内の ∂_n^j ($j > k_0$) と x_n を (5) を用いて modulo $I + x_n D$ で除く. その結果 $\tilde{f}$ は F_{k_0} に属する.
2. $\tilde{f}$ の F_{k_0} における normal form を G のグレブナー基底で求める. その normal form を $\bar{f}$ と書く.

$$D/I \ni [f] \mapsto [f\ell] \in D/I'$$

$$D/(I + x_n D) \ni [f] \mapsto [f\ell] \in D/(I' + x_n D) \quad (6)$$

$k'_0 : I'$ の b -関数の非負最大整数根.

命題 1

$k_0 = k'_0 = 0$ とする. この時, 制限における準同型 (6) は

$$D_{n-1}/D_{n-1}G \ni [f] \mapsto [f\bar{\ell}] \in D_{n-1}/D_{n-1}G' \quad (7)$$

で与えられる. ここで normal form $\bar{\ell}$ は $I' + x_n D$ で計算する.

同型射の制限, GKZ 系

$$\ell_i := -(\theta_i + \sum_{j>d} a_{ij}\theta_j - \beta_i), \quad \partial_i - \ell_i \in I := H_A(\beta)|_{x_k \rightarrow x_k+1, k=1, \dots, d}.$$

アルゴリズム 1 (LR(left-right)-reduction)

- ▶ 入力: $\ell \in D$. Rules $\partial_i \rightarrow \ell_i \in D, i = 1, \dots, d$.
- ▶ 出力: normal form $\bar{\ell}$, これは $\bar{\ell} = \ell \bmod I + \sum_{i=1}^d x_i D$ をみたす.

Repeat

$$\ell \leftarrow \ell|_{x_1=\dots=x_d=0};$$

ℓ の中で $t := cx^\alpha \partial^\beta \partial_i, c \in K$ なる形の元 ($\partial_i, i = 1, \dots, d$ を含む元) を選び次のように書き換える:

$$\ell \leftarrow cx^\alpha \partial^\beta \ell_i + (\ell - t)$$

until (there is no term divided by $\partial_i, i = 1, \dots, d$)

ℓ を $\bar{\ell}$ として出力

$$A = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \quad x_1 = x_2 = x_3 = 1 \text{ への } \partial_1 \text{ の制限.}$$

```
[2622] import("2025-04-01-contiguity.rr");
[2652] A=[[1,0,0,-1], [0,1,0,1], [0,0,1,1]];
[2653] lr_red(dx1,A,[x1,x2,x3,x4],[b1,b2,b3],0);
// LR reduction の右辺.
Aop=[x4*dx4-x1*dx1+b1,-x4*dx4-x2*dx2+b2,-x4*dx4-x3*dx3+b3]
lr_red1(L=dx1)
lr_red1_top(L=dx1)
lr_red1_top(dx1)-->x4*dx4+b1
lr_red1(L=0)
check_witness: Xvar=[x1,x2,x3], Xvar_all=[x1,x2,x3,x4]
  L[dx1](orig) = L1[x4*dx4+b1](reduced)+Witness[[ 1 0 0 ]] [dxi-Aop[i]]
                mod Xvar*D
  check_witness: OK.
x4*dx4+b1
```

$x_1, x_2, x_3, \partial_1, \partial_2, \partial_3$ を消去して $\bar{\partial}_1 = x_4 \partial_4 + \beta_1$.

アルゴリズム 2 (GKZ 系の E, b を計算)

- ▶ 入力: $H_A(\beta)^3$ の生成元 $\ell_1, \dots, \ell_m$. ∂^u, ∂^v , ここで $u, v \in \mathbb{N}_0^d$ で, これらの台は disjoint で $D/H_A(\beta - Au)$ と $D/H_A(\beta - Av)$ は同型と仮定.
- ▶ 出力: $E\partial^u = b\partial^v$ modulo $H_A(\beta)$ である $E \in D$, $b \in \mathbb{C}[\beta]$.
- 1. $F \subset D^{m+2}$ (F は次のページ) の POT 順序で F のグレブナー基底 G を計算. ここで POT 順序の tie breaker $\prec$ は $\beta \prec (\partial_i \text{'s in the support of } \partial^v) \prec (\text{その他の変数})$. とする.
- 2. $(b\partial^w, c_0, c_1, \dots, c_m)^T$, $\partial^w | \partial^v$ なる形の元を G 内で探す.
- 3. $E = \partial^{v-w} c_0$ において return $[E, b]$.

³以下 β も変数とみなす.

F は次の元の集合.

$$\begin{pmatrix} \partial^u \\ 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \begin{pmatrix} \ell_1 \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \begin{pmatrix} \ell_2 \\ 0 \\ 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \ell_m \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} \in D^{m+2} \quad (8)$$

M.Saito (2001) の Alg 4.2 は E, b を計算するより能率のよいアルゴリズムを与えているが複雑. この方法は効率が劣り $b(s)$ に余分な因子を持つ可能性あり, しかしよりシンプル.

例.

$$A = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \quad (9)$$

$u = (1, 0, 0, 0)^T, v = (0, 1, 0, 0)^T$. パラメータのシフトは
 $Au - Av = (1, 0, 0)^T - (0, 1, 0)^T$.

$$F = \{x_1\partial_1 - x_4\partial_4 - \beta_1, x_2\partial_2 + x_4\partial_4 - \beta_2, x_3\partial_3 + x_4\partial_4 - \beta_3, \partial_1\partial_4 - \partial_2\partial_3\}$$

は $H_A(\beta)$ の生成元. D^6 での $(\partial_1, 1, 0, 0, 0, 0), \{(\ell_k, 0, e_k) \mid \ell_k \in F\}$
のグレブナー基底 (POT order such that
 $x_1, x_2, x_3, x_4, \partial_1, \partial_3, \partial_4 \succ \partial_2 \succ \beta_1, \beta_2, \beta_3$) は以下の元を含む.

$$\begin{aligned} & [(\beta_1 + \beta_3)(\beta_1 + \beta_2), c_0 := x_2x_3\partial_4 + x_1x_2\partial_2 + (\beta_1 + \beta_3)x_1, \\ & -(x_2\partial_2 + \beta_1 + \beta_3), -(\beta_1 + \beta_3), -x_2\partial_2, x_2x_3] \end{aligned}$$

したがって, $b = (\beta_1 + \beta_3)(\beta_1 + \beta_2), \partial^w = 1, E = \partial_2 c_0$.

```

[2622] import("2025-04-01-contiguity.rr");;
[2652] A=[[1,0,0,-1], [0,1,0,1], [0,0,1,1]];
[2685] L=uv_contiguity([[1,0,0,0],[0,1,0,0]],A,[b1,b2,b3]);
      [1,1,1,1,1,0,1,1]
      G[15]=[-b1^2+(-b2-b3)*b1-b3*b2,-x2*x3*dx4-x1*x2*dx2+(-b1-b3)*x1,
              x2*dx2+b1+b3,b1+b3,x2*dx2,-x2*x3]

      [(-b1^2+(-b2-b3)*b1-b3*b2)*dx2,
        (-x2*x3*dx2-x3)*dx4-x1*x2*dx2^2+(-b1-b3-1)*x1*dx2,x1,x2,x3,x4]
[2686] fctr(red(L[0]/dx2));
      [[-1,1],[b1+b3,1],[b1+b2,1]]

```

実装: Risa/Asir(Weyl 代数) と sagemath(多面体—同型分類用)

例: Gauss HG の contiguity

```
sage: load("2025-07-23-representative.py")
sage: ring=PolynomialRing(QQ,'x,dx,a,b,c')
....: x,dx,a,b,c=ring.gens()
....: dic={'x':x,'dx':dx,'a':a,'b':b,'c':c}
....:
sage: gauss_contiguity([a,b,c],[a+1,b,c],ring)
[-x*dx - a,
 1,
 '-x*dx-a,[[a,b,c],->,[a+1,b,c]], [dx2,dx2,1,dx2]]',
 Multivariate Polynomial Ring in x, dx, a, b, c over Rational Field]

sage: ans=gauss_contiguity([a+1,b,c],[a,b,c],ring)
sage: ans
[x^2*dx - x*dx + x*b + a - c + 1,
 -a^2 + a*c - a,
 '(x^2-x)*dx+b*x+a-c+1,[[a+1,b,c],->,[a,b,c]], [-x*x1*dx3-x*x2*dx+(a-c+1
 Multivariate Polynomial Ring in x, dx, a, b, c over Rational Field]
sage: factor(ans[1])
(-a + c - 1) * a
```

例: Appell F_1 の contiguity

```
sage: ring=PolynomialRing(QQ,'x,dx,y,dy,a,b,bp,c')
....: x,dx,y,dy,a,b,bp,c=ring.gens()
....: dic={'x':x,'dx':dx,'y':y,'dy':dy,'a':a,'b':b,'bp':bp,'c':c}
....:
sage: f1_contiguity([a,b,bp,c],[a+1,b,bp,c],r=ring)
[-x*dx - y*dy - a,
 1,
 '[[a,b,bp,c], -> ,[a+1,b,bp,c]], [dx1,dx1,1,dx1]]',
 Multivariate Polynomial Ring in x, dx, y, dy, a, b, bp, c over Rational
sage: f1_contiguity([a,b,b,c],[a,b+1,b+1,c],r=ring)
[(x^2*dx*b - y^2*dy*b + x*b^2 - y*b^2)/(x - y),
 1,
 '[[a,b,b,c], -> ,[a,b+1,b+1,c]], [dx2*dx3,dx2*dx3,1,dx2*dx3]]',
 Multivariate Polynomial Ring in x, dx, y, dy, a, b, bp, c over Rational
```

例: Appell F_2

```
sage: load("2025-07-23-representative.py")
sage: ring=PolynomialRing(QQ,'x,dx,y,dy,a,b,bp,c,cp')
....: x,dx,y,dy,a,b,bp,c,cp=ring.gens()
....: dic={'x':x,'dx':dx,'y':y,'dy':dy,'a':a,'b':b,'bp':bp,'c':c,'cp':cp}
....:
sage: f2_contiguity([c,c,-1,c,cp],[c+1,c+1,-1,c+1,cp],ring)
[-dx,
 -1,
 '[c,c,-1,c,cp], -> ,[c+1,c+1,-1,c+1,cp]],[-dx6,-dx1*dx2,dx4,dx1*dx2]]',
Multivariate Polynomial Ring in x, dx, y, dy, a, b, bp, c, cp over Rational Field]
sage: f2_contiguity([c,c,-1,c,cp],[c-1,c-1,-1,c-1,cp],ring)
[-x*y^2*dy^2 - y^3*dy^2 + x*y*dy^2 + 2*y^2*dy^2 - x^2*dx*c + x^2*dx*cp + x^2*dx - y^2*dy - y*dy^2 + x*dx*
 c^3 - c^2*cp - 2*c^2 + 2*c*cp + c - cp,
 '[c,c,-1,c,cp], -> ,[c-1,c-1,-1,c-1,cp]],[-x1*x2*x7*dx4*dx7+(-x4^2*x6+x1*x2*x4)*dx4^2+(-x2*x5*x7*dx3+(-
Multivariate Polynomial Ring in x, dx, y, dy, a, b, bp, c, cp over Rational Field]
```

例: $F_2, c \mapsto c + 2$

```
sage: load("2025-07-23-representative.py")
sage: ring=PolynomialRing(QQ,'x,dx,y,dy,a,b,bp,c,cp')
....: x,dx,y,dy,a,b,bp,c,cp=ring.gens()
....: dic={'x':x,'dx':dx,'y':y,'dy':dy,'a':a,'b':b,'bp':bp,'c':c,'cp':cp}
....:
sage: ans=f2_contiguity([c/2-1,b,-1,c,-c/2],[(c+2)/2-1,b,-1,c+2,-(c+2)/2],ring)
[(x^2*y*dy^2*c + 3*x*y^2*dy^2*c + 2*y^3*dy^2*c - 1/2*x^3*dx*c^2 - 1/2*x^2*y*dy*c^2 + 2*x^2*y*dy^2 + 2*x*y^
-b^2*c + 2*b*c^2 - c^3 - 2*b^2 + 5*b*c - 3*c^2 + 2*b - 2*c,
'[[1/2*c-1,b,-1,c,-1/2*c], -> ,[1/2*c,b,-1,c+2,-1/2*c-1]], [(-2*x1*x2^2*dx5-2*x2^2*x7*dx3)*dx6^2+(2*b-4*c
Multivariate Polynomial Ring in x, dx, y, dy, a, b, bp, c, cp over Rational Field]
sage: factor(ans[1])
(-1) * (c + 2) * (-b + c) * (-b + c + 1)
```

例: パラメータの飛びが大きいと時間がかかる

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & -2 & -3 \\ 0 & 1 & 3 & 4 \end{pmatrix}$$

```
sage: load("2025-07-23-representative.py")
sage: gb_verbose(1)
sage: ring=PolynomialRing(QQ,'x,dx,y,dy,a,b')
....: x,dx,y,dy,a,b=ring.gens()
....: dic={'x':x,'dx':dx,'y':y,'dy':dy,'a':a,'b':b}
....:
sage: f0134_contiguity([-3*a-1,4*a+1],[-3*a-4,4*a+5],ring)
[(46656*x^6*y*dy^2*a^3 + 69984*x^6*y*dy^2*a^2 + 7884*x^5*y*dy^2*a^3 + 10368*x^4*y^2*dy^2*a^3 + 331776*x^3*
9216*a^5 + 29952*a^4 + 32832*a^3 + 11952*a^2 - 1224*a - 1080,
'[-3*a-1,4*a+1], -> ,[-3*a-4,4*a+5]],[((3888*a^2+6804*a+2916)*x3^3*dx2+((128*a-32)*x1^2*x4^2+(-16*a+4)*
Multivariate Polynomial Ring in x, dx, y, dy, a, b over Rational Field]
```

Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, で 218m49.79s (実時間).
アルゴリズム 2 の GB 計算に時間がかかっている。

```
load("2025-04-01-contiguity.rr");;
dp_gr_print(1);;
uv_contiguity(UV=[[0,4,0,0],[4,0,0,0]],A=[[1,0,-2,-3],[0,1,3,4]],
              Beta=[-3*a-1,4*a+5]);
// degree 30 の GB.
```