

ns_twistedlog.rr

ns_twistedlog.rr User's Manual
Edition 1.0
February 2012

by Keisuke Nishitani

1 ns_twistedlog.rr

1.1 ns_twistedlog.rr について

‘ns_twistedlog.rr’ は twisted logarithmic cohomology 群の計算, およびそれに基づいて, 多項式ベキの積分から定まるあるクラスの超幾何積分の満たす差分方程式系の計算と, 指数関数と多項式ベキの積分から定まるあるクラスの超幾何積分の満たす微分方程式系の計算を行うためのパッケージである.

1.2 Twisted logarithmic cohomology 群の計算に関する関数

1.2.1 ns_twistedlog.twisted_log_cohomology

`ns_twistedlog.twisted_log_cohomology(FL, PL, VL)`
 :: Twisted logarithmic cohomology 群の middle cohomology 群の基底を返す.

FL 多項式のリスト

PL パラメータのリスト

VL 変数のリスト

- *PL* に属する変数は, 係数体に属する整数でない不定元と見なして計算される. 従って, これらの変数に依存する generic *b*-関数の根は最大整数根とはならず, またグレブナ基底の計算において, これらの変数を係数に含む先頭項は 0 にはならない. このような事情により, パラメータが変数の場合と整数の場合とでは出力結果が異なる場合がある.
- 3 変数以上の場合だと, twisted logarithmic cohomology 群からくる *D*-加群がホロノミックでない場合がある. そのような場合には twisted logarithmic cohomology 群は計算できない.

```
[1] ns_twistedlog.twisted_log_cohomology([x,y,1-x-y],[a,b,c],[x,y]);
-- nd_weyl_gr :0.003848sec(0.008291sec)
-- weyl_minipoly_by_elim :0.006988sec(0.007177sec)
-- generic_bfct_and_gr :0.01325sec(0.02175sec)
generic bfct : [[-1,1],[s,1],[s+a+b+c-1,1]]
S0 : 0
B_S0 length : 1
-- fctr(BF) + base :0.001454sec(0.005543sec)
dimension : 1
[1]

[2] ns_twistedlog.twisted_log_cohomology([x,y,1-x-y],[-1,-2,-3],[x,y]);
-- nd_weyl_gr :0.001845sec(0.001838sec)
-- weyl_minipoly_by_elim :0.003972sec(0.003971sec)
-- generic_bfct_and_gr :0.007363sec(0.007584sec)
generic bfct : [[-1,1],[s,1],[s-7,1]]
S0 : 7
B_S0 length : 36
-- fctr(BF) + base :0.02438sec(0.03323sec)
dimension : 3
```

```
[y^2*x^5,y^7,1]
```

```
[3] ns_twistedlog.twisted_log_cohomology([x*z+y,x^4+y^5+x*y^4],[0,0],[x,y,z]);
-- nd_weyl_gr :0.004sec(0.0028sec)
weyl_minipoly_by_elim : b-function does not exist
stopped in weyl_minipoly_by_elim2 at line 378 in file "/usr/local/ox/OpenXM/src/
asir-contrib/packages/src/nk_restriction.rr"
378          error("weyl_minipoly_by_elim : b-function does not exist");
(debug)
```

参照 ns_twistedlog.twisted_log_cohomology(option)

1.2.2 ns_twistedlog.twisted_log_cohomology(option)

ns_twistedlog.twisted_log_cohomology(... | exp = f, check = n, s0 = m, excp
= v)

:: ns_twistedlog.twisted_log_cohomology のオプションの説明

f 多項式
n 0 または 1
m 整数
v 0 または 1

- *exp* を指定すると、外微分の twisted の項に指数関数 $\exp(f)$ が含まれる場合の twisted logarithmic cohomology 群の計算を行う。
- *n* が 0 でないとき、twisted logarithmic cohomology 群からくる *D*-加群がホロノミックであるか否かの判定を行う。
- *s0* を指定すると、*s-m* を generic *b*-関数として計算を行う。本来の generic *b*-関数の計算は行わない。
- *v* が 0 でないとき、除外集合の計算を行う。パラメータに現れる変数は整数でないと仮定しており、その情報は出力されない。excp を指定するとグレブナ基底計算において Buchberger アルゴリズムがそのまま用いられるため、計算は格段に遅くなる。

```
[4] ns_twistedlog.twisted_log_cohomology([x,y,1-x-y],[a,b,c],[x,y]|exp = x+y);
-- nd_weyl_gr :0.004sec + gc : 0.004sec(0.006156sec)
-- weyl_minipoly_by_elim :0sec(0.001558sec)
-- generic_bfct_and_gr :0.004sec + gc : 0.004sec(0.008213sec)
generic bfct : [[1,1],[s,1],[s-1,1],[s+a+b-1,1]]
S0 : 1
B_S0 length : 3
-- fctr(BF) + base :0sec(0.000469sec)
dimension : 2
[y,1]
```

```
[5] ns_twistedlog.twisted_log_cohomology([x*z+y,x^4+y^5+x*y^4],[0,0],[x,y,z]|ch
eck = 1);
Hilbert polynomial : 1/24*x^4+65/12*x^3-529/24*x^2+727/12*x-51
holonomic : No
```

```

-- nd_weyl_gr :0.004001sec(0.002876sec)
weyl_minipoly_by_elim : b-function does not exist
stopped in weyl_minipoly_by_elim2 at line 378 in file "/usr/local/ox/OpenXM/src/
asir-contrib/packages/src/nk_restriction.rr"
378          error("weyl_minipoly_by_elim : b-function does not exist");
(debug)

[6] ns_twistedlog.twisted_log_cohomology([x*z+y,x^4+y^5+x*y^4],[0,0],[x,y,z]|s0 = 1
);
dimension : 3
[y,z,1]

[7] ns_twistedlog.twisted_log_cohomology([x,y,1-x-y],[a,b,c],[x,y]|excp = 1);
generic bfct : [[-1,1],[s,1],[s+a+b+c-1,1]]
S0 : 0
B_S0 length : 1
dimension : 1
[[1],[a+b+c-1,a,b]]

```

1.2.3 ns_twistedlog.difference_equation

`ns_twistedlog.difference_equation(FL,PL,VL)`
 :: 多項式ベキの積分から定まる超幾何関数の満たす差分方程式系を返す.

FL 多項式のリスト

PL 多項式のベキのリスト(パラメータに対応)

VL 積分を行う変数のリスト

- Twisted logarithmic cohomology 群の基底を用いて, 多項式ベキの積分から定まる超幾何積分の満たす齊次差分方程式系を計算する.
- 差分方程式が計算できるのは, パラメータが多項式のベキのみに存在する場合に限る. パラメータに現れる変数は整数でない不定元として扱われる. さらに, 多項式のベキの形には以下を仮定している.
 1. 多項式のベキは有理数であるか, またはパラメータを表す 1 文字+有理数の形をしている.
 2. パラメータに現れる変数は全て相異なる.
 3. パラメータに現れる変数の係数は 1 である.

この仮定を満たさない入力に対しては正しい出力が得られない.

- 3 変数以上の場合には, twisted logarithmic cohomology 群からくる D -加群がホロノミックとならないために, 計算できない場合がある.

以下は, $p(a,b,c) = \int_C x^{a-1}y^{b-1}(1-x-y)^{c-1} dx dy$ の満たす差分方程式系を計算した例である.

```

[8] ns_twistedlog.difference_equation([x,y,1-x-y],[a,b,c],[x,y]);
-- nd_weyl_gr :0sec(0.000421sec)
-- weyl_minipoly_by_elim :0sec(0.001051sec)
Order : 1

```

```
[(-ea+1)*a-ea*b-ea*c,eb*a+(eb-1)*b+eb*c,ec*a+ec*b+(ec-1)*c]
```

以下のような入力に対しては正しく動かない.

```
[9] ns_twistedlog.difference_equation([x,y,1-x-y],[a,b,a-b],[x,y]);
-- nd_weyl_gr :0sec(0.0003741sec)
-- weyl_minipoly_by_elim :0.004sec + gc : 0.004sec(0.006554sec)
Order : 1
[-ea,eb,1]
```

```
[10] ns_twistedlog.difference_equation([x,y,1-x-y],[-a,-b,2*c],[x,y]);
-- nd_weyl_gr :0sec(0.0003951sec)
-- weyl_minipoly_by_elim :0sec(0.001059sec)
Order : 1
[(ea-1)*a+ea*b-2*ea*c,-eb*a+(-eb+1)*b+2*eb*c,ec*a+ec*b+(-2*ec+2)*c]
```

参照 ns_twistedlog.difference_equation(option)

1.2.4 ns_twistedlog.difference_equation(option)

```
ns_twistedlog.difference_equation(... | exp = f, check = n, inhom = h,
shift = p,
order = m, excp = v)
```

:: ns_twistedlog.difference_equation のオプションの説明.

f	多項式
n	0 または 1
h	0 または 1
p	パラメータ
m	整数
v	0 または 1

- exp を指定すると, 被積分関数に指数関数 $\text{exp}(f)$ がある場合の差分方程式系を計算する.
- n が 0 でないとき, twisted logarithmic cohomology 群からくる D -加群がホロノミックであるか否かの判定を行う.
- h が 0 でないとき, 非斉次項部分の計算を行う.
- shift を指定すると, 指定されたパラメータに関する差分方程式のみを計算する. これは差分方程式系から消去法を行うよりも, 効率のよい計算である.
- $s0$ を指定すると, $s-m$ を generic b -関数として計算を行う. 本来の generic b -関数の計算は行わない.
- v が 0 でないとき, 除外集合の計算を行う. パラメータに現れる変数は整数でないと仮定しており, その情報は出力されない. excp を指定するとグレブナ基底計算において Buchberger アルゴリズムがそのまま用いられるため, 計算は格段に遅くなる. excp は inhomo , shift , order と併用できない.

```
[11] ns_twistedlog.difference_equation([x,y,1-x-y],[a,b,c],[x,y]|inhomo = 1);
-- nd_weyl_gr :0sec(0.0003991sec)
-- weyl_minipoly_by_elim :0sec(0.001058sec)
```

```

Order : 1
[[(-ea+1)*b*a-ea*b^2-ea*c*b,[(y^2-y)*dy+b*x+(b+c)*y-b)*dx+(-y^2+y)*dy^2+((-a-b-c)*y+b)*dy,(-a-b-c)*x+(-b-c)*y]], [eb*a+(eb-1)*b+eb*c,[(y^2-y)*dy+b*x+(b+c)*y-b)*dx+(-y^2+y)*dy^2+((-a-b-c)*y+b)*dy,-y]], [ec*b*a+ec*b^2+(ec-1)*c*b,[(y^2-y)*dy+b*x+(b+c)*y-b)*dx+(-y^2+y)*dy^2+((-a-b-c)*y+b)*dy,(-a-b-c)*x-c*y]]]

[12] ns_twistedlog.difference_equation([x,y,1-x-y],[a,b,c],[x,y]|shift = a);
-- nd_weyl_gr :0.004sec(0.0004289sec)
-- weyl_minipoly_by_elim :0sec(0.001042sec)
Order : 1
[(ea-1)*a+ea*b+ea*c]

```

1.2.5 ns_twistedlog.differential_equation

`ns_twistedlog.differential_equation(EXP,FL,PL,TVL,XVL)`
 :: 指数関数と多項式ベキの積分から定まる超幾何関数の満たす微分方程式系を返す.

EXP 多項式
FL 多項式のリスト
PL 多項式のベキのリスト
TVL 積分を行う変数のリスト
XVL パラメータの変数のリスト

- Twisted logarithmic cohomology 群の基底を用いて, 指数関数と多項式ベキの積分から定まる超幾何関数の満たす斉次微分方程式系を計算する.
- 微分方程式が計算できるのは, パラメータが指数関数に存在する場合に限る. 多項式のベキに変数が存在しても構わないが, その変数は固定されたものと見なされる. パラメータに現れる変数および多項式のベキに現れる変数は, 整数でない不定元として扱われる.
- 3 変数以上の場合には, twisted logarithmic cohomology 群からくる D -加群がホロノミックとならないために, 計算できない場合がある.

以下は $f(x_1, x_2) = \int_C \exp(x_1 t_1 + x_2 t_2) t_1^{a-1} t_2^{b-1} dt_1 dt_2$ の満たす微分方程式系を計算した例である.

```

[13] ns_twistedlog.differential_equation(x1*t1+x2*t2,[t1,t2],[a,b],[t1,t2],[x1,x2])
;
-- nd_weyl_gr :0sec(0.0004089sec)
-- weyl_minipoly_by_elim :0sec(0.000495sec)
Order : 1
[x1*dx1+a,-x2*dx2-b]

```

参照 `ns_twistedlog.differential_equation(option)`

1.2.6 ns_twistedlog.differential_equation(option)

`ns_twistedlog.differential_equation(... | check = n, inhom = h, diff = p, order = m, excp = v)`
 :: `ns_twistedlog.differential_equation` のオプションの説明

n 0 または 1
 h 0 または 1
 p パラメータ
 m 整数
 v 0 または 1

- n が 0 でないとき, twisted logarithmic cohomology 群からくる D -加群がホロノミックであるか否かの判定を行う.
- h が 0 でないとき, 非斉次項部分の計算を行う.
- `diff` を指定すると, 指定されたパラメータに関する微分方程式のみを計算する. これは微分方程式系から消去法を行うよりも, 効率のよい計算である.
- `s0` を指定すると, $s-m$ を generic b -関数として計算を行う. 本来の generic b -関数の計算は行わない.
- v が 0 でないとき, 除外集合の計算を行う. パラメータに現れる変数は整数でないと仮定しており, その情報は出力されない. `excp` を指定するとグレブナ基底計算において Buchberger アルゴリズムがそのまま用いられるため, 計算は格段に遅くなる. `excp` は `inhomo`, `diff`, `order` と併用できない.

```
[14] ns_twistedlog.differential_equation(x1*t1+x2*t2,[t1,t2],[a,b],[t1,t2],[x1,x2] |
diff = x1);
-- nd_weyl_gr :0sec(0.0007901sec)
-- weyl_minipoly_by_elim :0sec + gc : 0.008sec(0.006175sec)
Order : 1
[x1*dx1+a]
```

1.3 その他の関数

1.3.1 ns_twistedlog.twisted_deRham

`ns_twistedlog.twisted_deRham(F,P,VL)`
 :: Twisted de Rham cohomology 群の middle cohomology 群の基底を返す.

F 多項式
 P パラメータ
 VL 変数のリスト

- P が変数の場合, 係数体に属する整数でない不定元と見なして計算される. 従って, これらの変数に依存する generic b -関数の根は最大整数根とはならず, またグレブナ基底の計算において, これらの変数を係数に含む先頭項は 0 にはならない. このような事情により, パラメータが変数の場合と整数の場合とでは出力結果が異なる場合がある.

```
[15] ns_twistedlog.twisted_deRham(x*y*(1-x-y),a,[x,y]);
-- nd_weyl_gr :0sec(9.489e-05sec)
-- weyl_minipoly :0sec(0.0002191sec)
-- generic_bfct_and_gr :0sec(0.000423sec)
generic bfct : [[1,1],[s,1]]
S0 : 0
```



```

B_S0 length : 1
-- fctr(BF) + base :0sec(6.008e-05sec)
dimension : 0
[]

[16] ns_twistedlog.twisted_deRham(x*y*(1-x-y),-1,[x,y]);
-- nd_weyl_gr :0sec(0.0001891sec)
-- weyl_minipoly :0sec(0.000247sec)
-- generic_bfct_and_gr :0sec(0.0006139sec)
generic bfct : [[1,1],[s,1],[s-1,1]]
S0 : 1
B_S0 length : 3
-- fctr(BF) + base :0.004sec(0.0002241sec)
dimension : 3
[x,y,1]

```

1.3.2 ns_twistedlog.holonomic

`ns_twistedlog.holonomic(Id, VL, DVL)`
 :: D の左イデアル Id がホロノミックならば標準単項式のリストを返す. ホロノミックでないならば -1 を返す.

Id イデアルの生成元のリスト

VL 変数のリスト

DVL 変数のリスト (*VL* に対応する微分作用素の方の変数)

```

[17] ns_twistedlog.holonomic([x*dy,y*dx],[x,y],[dx,dy]);
Hilbert polynomial : x^2+1
holonomic : Yes
holonomic rank : 1
[1]

[18] ns_twistedlog.holonomic([(x^3-y^2)*dx+3*x^2,(x^3-y^2)*dy-2*y],[x,y],[dx,dy]);
Hilbert polynomial : 1/2*x^3+2*x^2+1/2*x+2
holonomic : No
-1

```

Index

(インデックスがありません)

(インデックスがありません)

簡単な目次

1	ns_twistedlog.rr	1
	Index	8

目次

1	ns_twistedlog.rr	1
1.1	ns_twistedlog.rr について	1
1.2	Twisted logarithmic cohomology 群の計算に関する関数	1
1.2.1	ns_twistedlog.twisted_log_cohomology	1
1.2.2	ns_twistedlog.twisted_log_cohomology(option)	2
1.2.3	ns_twistedlog.difference_equation	3
1.2.4	ns_twistedlog.difference_equation(option)	4
1.2.5	ns_twistedlog.differential_equation	5
1.2.6	ns_twistedlog.differential_equation(option)	5
1.3	その他の関数	6
1.3.1	ns_twistedlog.twisted_deRham	6
1.3.2	ns_twistedlog.holonomic	7
	Index	8

