1　微分作用素環のグレブナー基底とその応用

例題 1 $\partial_x^2 x \partial_x$ を紙と鉛筆で計算し, Macaulay2, Singular, Risa/Asir での計算と比較せよ.

解答 1 非可換関係 $\partial_x x = x\partial_x + 1$ に注意しながら, まずは手計算してみよう.

紙と鉛筆: 微分作用素の積

$$\partial_x^2 x \partial_x = \partial_x (x\partial_x + 1)\partial_x = (x\partial_x + 1)\partial_x^2 + \partial_x^2 = x\partial_x^3 + 2\partial_x^2.$$

Macaulay2, (Singular) の多項式環の宣言については二日目に学んだが, 第 6 章では主に微分作用素環を用いる. まず, これらのシステムでの微分作用素環の定義の仕方を説明する.

Macaulay2: 微分作用素環の宣言と微分作用素の積

```
i1 : R = QQ[x,dx,WeylAlgebra => {x=>dx}];
i2 : dx^2*x*dx
          3      2
o2 = x*dx  + 2dx
o2 : R

i3 : loadPackage "Dmodules";
i4 : R1=QQ[x];
i5 : D1=makeWA R1;
i6 : dx^2*x*dx
          3      2
o6 = x*dx  + 2dx
o6 : D1
```

Macaulay2 では, 多項式環の宣言とほぼ同様であるが, 変数を並べたあとに `WeylAlgebra => {x=>dx}` と書いて `x` に関する微分作用素が `dx` であることを明示する. このように宣言しておけば, 通常の積の演算子 `*` で, 交換関係を考慮して扱われる.

また, パッケージ `Dmodules` に定義されている `makeWeylAlgebra` (省略形 `makeWA`) という関数で現在の多項式環に対応する微分作用素環を定義することができる. 各変数に `d` がついたものが, 対応する微分作用素である. 環の定義の長い入力が必要なくなるので便利ではあるが, 後で登場する項順序の指定などの細かい設定を要するときは, 前述の方法を用いなければならないことも多い.

Singular: 微分作用素環の宣言と微分作用素の積

```
> LIB "dmod.lib";
> ring r=0,(x,dx),dp;
> def D=Weyl();
> setring D;
> poly f = dx^2*x*dx;
> f;
x*dx^3+2*dx^2
```

Singular における微分作用素環の宣言は, 非可換環を扱うためのライブラリ `nctools.lib` で定義されている関数 `Weyl` を用いるのが簡単である. しかし, この本では常に `dmod.lib` を読み込むことにする. このライブラリを読み込むと `nctools.lib` を含む多くの D 加群に関するライブラリが読み込まれて便利だからである.

まず, 変数と微分作用素に相当する変数を持つ多項式環を宣言する. その後, `Weyl` というコマンドを用いてワイル代数を定義する. 引数なし, または引数 0 でこの関数を呼ぶと, 現在自分が扱っている環の後ろ半分の変数を前半分の変数に対する微分作用素とするワイル代数を生成する. 非ゼロな引数を与えると, 前から順に (変数, 微分作用素) のペアと見なした環を定義する. つまり, $(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n})$ という変数を持つ多項式環を定義した状態で `Weyl()`, または `Weyl(0)` と呼べば, x_i $(1 \leq i \leq n)$ に関する微分作用素が x_{i+n} であるような微分作用素環が定義され, `Weyl(1)` と呼べば, x_i (i : 奇数) に関する微分作用素が x_{i+1} であるような微分作用素環が定義される. どちらを使ってもよいのだが, 関数の中には変数が前者のように宣言されていることを仮定しているものがあるので, この本では常に前者, つまり `Weyl()` を用いることにする.

Risa/Asir: 微分作用素の積

```
[1371] P=dx^2;
dx^2
[1372] Q=x*dx;
x*dx
[1373] V=[x,dx];
[x,dx]
[1374] DP=dp_ptod(P,V);
(1)*<<0,2>>
[1375] DQ=dp_ptod(Q,V);
(1)*<<1,1>>
[1376] DPQ=dp_weyl_mul(DP,DQ);
(1)*<<1,3>>+(2)*<<0,2>>
[1377] PQ=dp_dtop(DPQ,V);
x*dx^3+2*dx^2
```

Risa/Asir では, あらかじめ基礎環の宣言を行わないので微分作用素として何かを計算するときには, 専用の関数を用いることが多い. 今の場合は, 微分作用素として積を計算する関数 `dp_weyl_mul` を用いる. もちろん, 第 1 引数の作用素が第

2 引数の作用素の左から掛かる．この関数は入力として分散表現しか受け付けないので, 先に変換を行わなければならない．また, 分散表現の指数部の前半分と後半分がそれぞれ変数と対応する微分作用素となっていなければならない．つまり, $V = [v_1, \ldots, v_n, v_{n+1}, \ldots, v_{2n}]$ という変数リストを用いて分散表現に変換して, dp_weyl_mul を施すと, v_{i+n} が v_i に関する微分作用素であるように扱われる．微分作用素環に関する関数の変数の入力には, このような仮定がなされていることが多い.

例題 2 $A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$, $\beta = (1,2,3)^T$ に付随する A-超幾何系 $H_A(\beta)$ を求めよ.

解答 2

Risa/Asir: A-超幾何系の計算

```
[1631] A=[[1,1,0,0],[0,0,1,1],[0,1,0,1]];
[[1,1,0,0],[0,0,1,1],[0,1,0,1]]
[1632] B=[1,2,3];
[1,2,3]
[1633] sm1.gkz([A,B]);
[[x2*dx2+x1*dx1-1,x4*dx4+x3*dx3-2,x4*dx4+x2*dx2-3,-dx1*dx4+dx2*dx3],
[x1,x2,x3,x4]]
```

Macaulay2: A-超幾何系の計算

```
i1 : loadPackage "Dmodules"
i2 : A=matrix{{1,1,0,0},{0,0,1,1},{0,1,0,1}};
              3        4
o2 : Matrix ZZ  <--- ZZ
i3 : b={1,2,3};
i4 : gkz(A,b)
o4 = ideal (D D  - D D , x D  + x D  - 1, x D  + x D  - 2, x D  + x D  - 3)
             2 3    1 4   1 1    2 2       3 3    4 4       2 2    4 4
o4 : Ideal of QQ[x , x , x , x , D , D , D , D ]
                  1   2   3   4   1   2   3   4
```

Singular: A-超幾何系の計算

```
> LIB "ncalg.lib";
> intmat A[3][4]=
1,1,0,0,
0,0,1,1,
0,1,0,1
> def D=GKZsystem(A,"lp","ect");
> setring D;
> GKZid;
GKZid[1]=x(1)*d(1)+x(2)*d(2)+(-b(1))
GKZid[2]=x(3)*d(3)+x(4)*d(4)+(-b(2))
GKZid[3]=x(2)*d(2)+x(4)*d(4)+(-b(3))
GKZid[4]=d(1)*d(4)-d(2)*d(3)
> subst(GKZid,b(1),1,b(2),2,b(3),3);
_[1]=x(1)*d(1)+x(2)*d(2)-1
_[2]=x(3)*d(3)+x(4)*d(4)-2
_[3]=x(2)*d(2)+x(4)*d(4)-3
_[4]=d(1)*d(4)-d(2)*d(3)
```

例題 3 定理 6.9.3 を用いて, $I = \langle \partial_x^2 + y^2, \partial_y^2 + x^2 \rangle$ のグレブナー基底計算を行え.

解答 3 $\partial_x \succ \partial_y \gg x \succ y$ なるブロック順序で各ブロックの比較は (次数) 逆辞書式順序を用いて計算することにする. 各システムにおける計算は以下のように行う. 各システムに共通することであるが, 2 日目に学んだブロック順序の指定法が使えないことに注意しよう. その方法では, 変数リストの後ろにあるもの優先のブロック順序は指定できないからである. こういう時に便利なのが行列による項順序 (matrix order) の指定である. どのように matrix order を指定しているか注意してほしい. この例で用いる行列は

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

である.

Macaulay2: R におけるグレブナー基底

```
i1 : loadPackage "Dmodules";
i2 : R=QQ[x,y,dx,dy,WeylAlgebra=>{x=>dx,y=>dy},
MonomialOrder=>{Weights=>{0,0,1,1},Weights=>{0,0,0,-1},
                Weights=>{1,1},Weights=>{0,-1}}];
i3 : I=ideal(dx^2-y^2,dy^2-x^2);
o3 : Ideal of R
i4 : gens gb I
o4 = | xdx-ydy dy^2-x2 ydxdy-dx-xy2 dx^2-y2 |
```

Macaulay2 では Weights を用いる. 初めの重みで引き分けた場合は, 2 番目の重みで比較, 2 番目の重みで引き分けた場合は, 3 番目の重みで比較, というように順に比較が行われるので, 行列の各行を並べれば matrix order を指定したことになる. 重みの長さが変数の長さより短い場合は, 残りの変数に対する重みは 0 とされる.

Singular: R におけるグレブナー基底

```
> LIB "dmod.lib";
> intmat ORD[4][4]=
. 0,0,1,1,
. 0,0,0,-1,
. 1,1,0,0,
. 0,-1,0,0;
> ring r=0,(x,y,dx,dy),M(ORD);
> def D=Weyl();
> setring D;
> ideal I=dx^2+y^2,dy^2+x^2;
> groebner(I);
_[1]=x*dx-y*dy
_[2]=dy^2+x^2
_[3]=y*dx*dy-dx+x*y^2
_[4]=dx^2+y^2
```

Singular では, intmat 型の変数で整数行列を作っておいて, 環の定義で M() の中に行列を入れると, matrix order が定義できる.

Risa/Asir: R におけるグレブナー基底

```
[1437] L1=dx^2+y^2$
[1438] L2=dy^2+x^2$
[1439] V=[x,y,dx,dy]$
[1440] M=newmat(4,4,[[0,0,1,1],[0,0,0,-1],[1,1],[0,-1]]);
[ 0 0 1 1 ]
[ 0 0 0 -1 ]
[ 1 1 0 0 ]
[ 0 -1 0 0 ]
[1441] G=nd_weyl_gr([L1,L2],V,0,M);
[x*dx-y*dy,dy^2+x^2,(y*dy-1)*dx+y^2*x,dx^2+y^2]
```

Risa/Asir ではグレブナー基底計算の時に順序を指定するが, その順序に行列を用いることができる.

例題 4 $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \end{pmatrix}$ の定義する A-超幾何系に対して, $\beta = (1,0)^T$, のときの rank を計算せよ.

解答 4

Macaulay2: rank の計算

```
i1 : loadPackage "Dmodules";
i2 : A=matrix{{1,1,1,1},{0,1,3,4}};
i3 : b={1,0};
i4 : I=gkz(A,b);
o4 : Ideal of QQ[x , x , x , x , D , D , D , D ]
                 1   2   3   4   1   2   3   4
i5 : I
i6 : R=I.ring;
i7 : S=newRing(R,MonomialOrder=>
          {Weights=>{0,0,0,0,1,1,1,1},
           Weights=>{0,0,0,0,0,0,0,-1},
           Weights=>{0,0,0,0,0,0,-1},
           Weights=>{0,0,0,0,0,-1},
           Weights=>{1,1,1,1},
           Weights=>{0,0,0,-1},
           Weights=>{0,0,-1},
           Weights=>{0,-1}})
i8 : J=(map(S,R))(I);
o8 : Ideal of S
i9 : leadTerm(J);
(略)
i12 : holonomicRank I
o12 = 4
```

Risa/Asir: rank の計算

```
[1631] A=[[1,1,1,1],[0,1,3,4]];
[[1,1,1,1],[0,1,3,4]]
[1632] B=[1,0];
[1,0]
[1633] G=sm1.gkz([A,B]);
[[x4*dx4+x3*dx3+x2*dx2+x1*dx1-1,4*x4*dx4+3*x3*dx3+x2*dx2,
-dx1*dx4+dx2*dx3,-dx2^2*dx4+dx1*dx3^2,dx1^2*dx3-dx2^3,-dx2*dx4^2+dx3^3],
[x1,x2,x3,x4]]
[1636] sm1.rank(G);
4
```

例題 5 R_2 において正規形を求める関数を書き, R_2 における Buchberger アルゴリズムを実装せよ. (参考 [2, 第 16 章].)

解答 5 実装例として Risa/Asir でのプログラム `nf_r2.rr` を紹介する. プログラムを簡単にするために, 変数は `x`, `y`, `dx`, `dy` に固定してある.

下の計算例では, R_2 の元 $x\partial_x\partial_y + x$ と $\frac{y}{x}\partial_x^2$ の積を計算している. この結果から積が, $y\partial_x^3\partial_y + \partial_x^3 - \frac{y}{x}\partial_x^2\partial_y + \frac{xy-1}{x}\partial_x^2$ であることがわかる.

Risa/Asir: nf_r2.rr による R_2 での積の計算

```
[1251] load("nf_r2.rr");
[1275] d_mult2(x*dx*dy + x, (y/x)*dx^2);
((y*x*dy+x)*dx^3+(-y*dy+y*x-1)*dx^2)/(x)
```

下の計算例では, $F = \frac{1}{x}\partial_x^2\partial_y^2 + (\frac{1}{x}+y)\partial_x^3 + \partial_x + \partial_y + 1$ に対して, $\partial_x \succ \partial_y$ なる純辞書式順序 $\prec$ についての先頭項 $\mathrm{in}_\prec(F) = (\frac{1}{x}+y)\xi_x^3$, 先頭単項式 ξ_x^3, 先頭係数 $(\frac{1}{x}+y)$ を求めている.

Risa/Asir: nf_r2.rr による先頭項, 先頭単項式, 先頭係数の計算

```
[1278] load("nf_r2.rr");
[1302] F = 1/x*dx^2*dy^2+(1/x+y)*dx^3+dx+dy+1;
((y*x^2+x)*dx^3+x*dy^2*dx^2+x^2*dx+x^2*dy+x^2)/(x^2)
[1303] in2(F, [dx,dy], 2);
((y*x+1)*dx^3)/(x)
[1304] in2_(F, [dx,dy], 2);
dx^3
[1305] c_in2(F, [dx,dy], 2);
(y*x+1)/(x)
```

下の計算例では, $F = \partial_x\partial_y^3$ の $G_1 = \partial_x\partial_y + 1, G_2 = 2y\partial_y^2 - \partial_x + 3\partial_y + 2x$ による, $\partial_x \succ \partial_y$ なる (次数) 逆辞書式順序 $\prec$ についての正規形 $-\frac{1}{2y}\partial_x + \frac{3}{2y}\partial_y + \frac{1}{y}x$ を求めている. さらに, 商の出力から, F, G_1, G_2 について

$$F = \partial_y^2 G_1 + \left(-\frac{1}{2y}\right) G_2 + \left(-\frac{1}{2y}\partial_x + \frac{3}{2y}\partial_y + \frac{1}{y}x\right)$$

が成り立つことがわかる.

Risa/Asir: nf_r2.rr による正規形の計算

```
[1306] load("nf_r2.rr");
[1330] F = dx*dy^3;
dy^3*dx
[1331] G = [dx*dy + 1, 2*y*dy^2-dx+3*dy+2*x];
[dy*dx+1,-dx+2*y*dy^2+3*dy+2*x]
[1332] normal_form2(F, G, [dx,dy], 0);
[(-1/2*dx+3/2*dy+x)/(y),[ dy^2 (-1/2)/(y) ]]
```

下の計算例では, R_2 の元 $f_1 = \partial_x^2 + y^2$, $f_2 = \partial_y^2 + x^2$ について, $\partial_x \succ \partial_y$ なる (次数) 逆辞書式順序 $\prec$ に関する S 多項式 $\mathrm{sp}(f_1, f_2)$ の計算と, R_2 のイデアル $\langle f_1, f_2\rangle$ のグレブナー基底 $G = \{\partial_x^2 + y^2, \partial_y^2 + x^2, -4x\partial_x + 4y\partial_y\}$ を計算している.

Risa/Asir: nf_r2.rr によるグレブナー基底計算

```
[1333] load("nf_r2.rr");
[1357] F1 = dx^2 + y^2$
[1358] F2 = dy^2 + x^2$
[1359] sp2(F1, F2, [dx,dy], 0);
-x^2*dx^2-4*x*dx+y^2*dy^2+4*y*dy
[1360] buchberger2([F1, F2], [dx,dy], 0);
[dx^2+y^2,dy^2+x^2,-4*x*dx+4*y*dy]
```

例題 6 $I = \langle \partial_x^2 + y^2, \partial_y^2 + x^2 \rangle$ を yang.rr を用いて Pfaffian 系に変換せよ.

解答 6 オイラー作用素 ($\theta_x = x\partial_x$) や差分作用素, q-差分作用素からなる環でのグレブナー基底計算などを行う Risa/Asir のパッケージとして yang.rr がある.

まず, load("yang.rr"); でパッケージを読み込む[1]. yang.rr では, ここまでの計算とは異なり, 環 $R = Q\langle \theta_x, \theta_y \rangle$ 上の計算であることに注意する. (実行結果の表記 dx, dy は x, y についてのオイラー作用素 θ_x, θ_y を意味している.) また, Q として有理数体 $\mathbf{Q}$ 以外に, 下の例のように $\mathbf{Q}(\alpha, \beta, \beta', \gamma)$ などをとることができる. 計算を行う前に, コマンド yang.define_ring を用いて環の定義を行わなければならない. R でのグレブナー基底 G をコマンド yang.gr で計算する. グレブナー基底 G を入力として, コマンド yang.stdmon を用いて標準単項式の集合 $S = \{1, \theta_y\}$ を計算できる. グレブナー基底 G と標準単項式の集合 S から, コマンド yang.pf により Pfaffian 系が得られる.

[1] yang.rr は Asir Contrib で定義されている関数を用いているので, Asir Contrib が読み込まれていない場合には, エラーメッセージが表示されることがある. このような場合は import("names.rr"); と実行すればよい.

Risa/Asir: yang.rr による計算

```
[1567] load("yang.rr");
[2120] yang.define_ring([x,y])$
[2121] L=[dx^2-y^2,dy^2-x^2]$
[2122] LL=yang.util_pd_to_euler(L,[x,y]);
[(dx^2-dx-y^2*x^2)/(x^2),(dy^2-dy-y^2*x^2)/(y^2)]
[2123] G=yang.gr(LL);
[dx-dy,dy^2-dy-y^2*x^2]
[2124] S=yang.stdmon(G);
[dy,1]
[2125] PF=yang.pf(reverse(S),G);
[ [ 0 (1)/(x) ]
[ y^2*x (1)/(x) ] [ 0 (1)/(y) ]
[ y*x^2 (1)/(y) ] ]
[2126] PF[0];
[ 0 (1)/(x) ]
[ y^2*x (1)/(x) ]
[2127] PF[1];
[ 0 (1)/(y) ]
[ y*x^2 (1)/(y) ]
```

この結果は, $I \bullet f = 0$ を満たす関数 f について, $F = \begin{pmatrix} f \\ y\frac{\partial f}{\partial y} \end{pmatrix}$ と置いたとき,

$$\frac{\partial F}{\partial x} = \begin{pmatrix} 0 & \frac{1}{x} \\ y^2 x & \frac{1}{x} \end{pmatrix} F, \quad \frac{\partial F}{\partial y} = \begin{pmatrix} 0 & \frac{1}{y} \\ yx^2 & \frac{1}{y} \end{pmatrix} F$$

を満たすことを意味している.

参考文献

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

[2] 野呂正行, 高山信毅, Risa/Asir ドリル, 2010, http://www.math.kobe-u.ac.jp/Asir/