数理解析研究所講究録
1199 巻 2001 年 90-99

D-加群における (u,v)-極小自由分解とその応用

東京女子大学　大阿久　俊則(Toshinori OOAKU) *
神戸大学　高山　信毅(Nobuki TAKAYAMA) †

D-加群における (u,v)-極小自由分解の説明をするには, LaScala の極小自由分解の構成アルゴリズムを説明する必要がある. 講演では, LaScala の極小自由分解の構成アルゴリズムをくわしく説明したのち, (u,v)-極小自由分解の説明およびデモをおこなった. (u,v)-極小自由分解とその例については, 数理研講究録 1171 “D 加群のアルゴリズム” 128–155 [7] で日本語で詳しく説明したので, ここでは, 主に前半部分の LaScala のアルゴリズムについて解説し, 極小自由分解の応用としての de Rham コホモロジ群の計算例等を掲載する.

1　自由分解の定義とシュライアーの構成法

以下 R で多項式環 $\mathbf{k}[x_1,\ldots,x_n]$, 微分作用素環 $D=\mathbf{k}\langle x_1,\ldots,x_n,\partial_1,\ldots,\partial_n\rangle$, または同次化微分作用素環 $D^{(h)}=\mathbf{k}\langle h,x_1,\ldots,x_n,\partial_1,\ldots,\partial_n\rangle$ をあらわす.

P を R の元を成分とする行列で, q 個の行ベクトル $f_1,\ldots,f_q\in R^p$ がならんだものとみなすことにする:

$$P=\overbrace{\begin{array}{|c|}\hline f_1 \\ \hline f_2 \\ \hline \cdots \\ \cdots \\ \cdots \\ \hline f_q \\ \hline \end{array}}^{p}$$

$\mathrm{Ker}\,P$ で P の syzygy module すなわち

$$\{h=(h_1,\ldots,h_q)\in R^q \mid hP=h_1f_1+\cdots+h_qf_q=0\}$$

を表すものとする. このとき $\mathrm{Ker}\,P$ は左 R-加群となる.

*oaku@twcu.ac.jp
†takayama@math.kobe-u.ac.jp

$f_1, \ldots, f_q$ がすでにグレブナ基底のときには, syzygy は次のように容易にわかる. $F = \{f_i\}$ がグレブナ基底なので,

$$\mathrm{sp}(f_i, f_j) \longrightarrow^* 0 \quad \text{by } F.$$

したがって, 次の式をみたす s^i_{jk}, q^k_{ij} が存在する

$$s^i_{ij} f_i + s^j_{ij} f_j + \sum q^k_{ij} f_k = 0.$$

e_i を i 番目の単位ベクトルとして,

$$s^i_{ij} e_i + s^j_{ij} e_j + \sum q^k_{ij} e_k$$

を $h^{ij}(P) \in R^q$ と書く.

定理 1 (よく知られている) Ker (P) は $h^{ij}(P)$, $i \neq j$ で生成される.

R-加群の自由分解というのは通常完全系列を用いて定義するが, 計算の立場では, 次のような性質をみたす行列の集合として十分である.

行列 P_0 が与えられているとする. 行列の集合

$$\{P_m, P_{m-1}, \ldots, P_0\}$$

が P_0 の自由分解 (free resolution) とは, 次の条件を満たすこと.

1. 積 $P_{i+1}P_i$ が定義できる, つまり P_{i+1} の行の長さと, P_i の列の長さが一致する.

2. Ker P_i は P_{i+1} の各行で R-加群として生成されている.

m を自由分解の長さとよぶ.

例 1.1
$R = \mathbf{Q}[x]$, $P_0 = \begin{pmatrix} x^2 \\ x^4 \end{pmatrix}$ のとき,

$$\left\{ (x^2, -1), \begin{pmatrix} x^2 \\ x^4 \end{pmatrix} \right\}$$

が P_0 の自由分解である.

自由分解は Ker を, 定理 1 を用いて順に計算していけば, 計算することが可能である. 自由分解は, 一意的でない.

さて, 1970 年代の終りごろ Schreyer は自由分解の計算には, グレブナ基底の計算は実質一回でよくて, あとは reduction の繰り返しのみであることを示した.

定理 2 (Schreyer) P の各列にあらわれている $f_1, \ldots, f_q \in R^p$ が順序 $\prec$ についてのグレブナ基底ならば, $\{h^{ij}(P) \mid i \neq j\}$ は Ker P の順序 $\prec'$ についてのグレブナ基底である. ここで, R^q での順序 $\prec'$ を次のように定める.

$$
\begin{aligned}
& ce_i \prec' de_j \\
\Leftrightarrow\ & cf_i \prec df_j \\
& \text{or } (cf_i = df_j \text{ かつ } i > j)
\end{aligned}
$$

ここで c, d は R の任意の元であり, "=" は, 順序が等しいという意味で用いている.

この定理をもちいて構成した自由分解を Schreyer 自由分解とよぶ.

2 LaScala のアルゴリズム

さて, $R = \mathbf{k}[x_1, \ldots, x_n]$ または $R = D^{(h)}$ と仮定し, P_0 の各成分はすべて同次式と仮定する. このとき P_0 の自由分解が "極小" (minimal) であるとは, 任意の i に対して P_i の成分が 1 を含まないことと定義する.

次の定理が知られている.

定理 3 (良く知られてる) P_0 の極小自由分解が存在する. 自由分解の長さは一意的であり, また, 極小自由分解に出現する, 行列 P_i のサイズも一意的である.

極小自由分解を計算するには, 各 Ker P_i の極小な個数の生成元をもとめればよい. さてこの極小自由分解を効率的に計算するのが LaScala のアルゴリズムである (たとえば, LaScala, Stillman の論文 [3] を参照). このアルゴリズムの大事な部分を説明しよう.

LaScala のアルゴリズムでは, $P_0, P_1, \ldots$ と順番に計算するのではなく, あるストラテジを決めて P_i の要素を一斉にもとめていく. 左側でアルゴリズムを, 右側で実例を説明する.

ここでは, 次の例を LaScala のアルゴリズムの説明のために使用しよう.

$$
\begin{aligned}
f_1 &= h\partial_x - (x\partial_x + y\partial_y) \\
f_2 &= h\partial_y - (x\partial_x + y\partial_y) \\
f_3 &= x\partial_x^2 - x\partial_x\partial_y + y\partial_x\partial_y - y\partial_y^2
\end{aligned}
$$

f_1, f_2, f_3 は, weight vector $(\overbrace{-1}^{x}, \overbrace{-1}^{y}, \overbrace{1}^{\partial_x}, \overbrace{1}^{\partial_y}, \overbrace{0}^{h})$ できまる部分順序を $\partial_x \succ \cdots \succ x_2 \succ h$ できまる lexicographic order で細分した全順序に関する, $D^{(h)}$ のグレブナ基底になっている.

Step 1. P_0 の各行がグレブナ基底であるように変更し, Schreyer 自由分解の S-pair のみ求める. これを skelton と呼ぶことにする. たとえば, 右の例では, $\hat{P}_1$ の $(1,3)$ は f_1 と f_3 の S-pair を作るための係数である.

$$\hat{P}_2$$

$$\begin{array}{cc} (1,2) & \begin{pmatrix} \underline{\partial_y} & -x\partial_x & 0 \end{pmatrix} \end{array}$$

level = 3

$$\hat{P}_1 \qquad\qquad\qquad \hat{P}_0$$

$$\begin{array}{cc} (1,3) & \\ (1,2) & \\ (2,3) & \end{array} \begin{pmatrix} \underline{x\partial_x} & 0 & -h \\ \underline{\partial_y} & -\partial_x & 0 \\ 0 & \underline{x\partial_x^2} & -h\partial_y \end{pmatrix} \qquad \begin{pmatrix} h\partial_x \\ h\partial_y \\ x\partial_x^2 \end{pmatrix}$$

level = 2 level = 1

Step 2. te_k, $t \in R$ を $\hat{P}_i$ の行の成分とするとき,

$$\text{s-degree}(te_k) = \deg(t) + \text{s-degree}(tg_k)$$

で再帰的に s-degree を定義する. ここで, $\deg(t)$ は t の全次数であり, g_k は $\hat{P}_{i-1}$ の k 番目の行とする. Skelton の 各 S-pair の strategy (str) を計算. ここで,

$$\text{strategy}(f) := \text{s-degree}(f) - \text{level}.$$

$$\begin{array}{cc} \text{s-deg} & \text{str} \end{array}$$
$$\begin{pmatrix} 1+4=5 & 2 \end{pmatrix}$$

$$\begin{array}{cc} \text{s-deg} & \text{str} \end{array} \qquad \begin{array}{cc} \text{s-deg} & \text{str} \end{array}$$
$$\begin{pmatrix} 2+2=4 & 2 \\ 1+2=3 & 1 \\ 3+2=5 & 3 \end{pmatrix} \quad \begin{pmatrix} 2 & 1 \\ 2 & 1 \\ 3 & 2 \end{pmatrix}$$

Step 3. strategy の小さい S-pair s より "すでにもとまった" グレブナ基底のみで reduction する. s の余りを r とおく.

if (r != 0) {
 (A) r を新しい グレブナ基底の元として加える.
 (B) Reduction の過程から, syzygy も get できる.
 (C) その syzygy の r の係数は 1 なので, この syzygy は極小自由分解には不要と印を付ける.
} else {
 Reduction の過程から syzygy を get.
}

level 1 には strategy が 1 の元は最初の 2 個である. これらをそのままグレブナ基底として加える. 次に level 2 の strategy 1 の元 (1,2) をもとに f_1 と f_2 の S-pair s を計算する.

$$s = (\partial_y - h)f_1 + (-\partial_x + h)f_2 - r$$

なる元を得る. r を新しいグレブナ基底として加える. r は f_3 に等しい. この reduction の過程より f_i の syzygy $(\partial_y - h, -\partial_x + h, -1)$ を得る. これは極小自由分解には不要である. 以下, strategy 2, 3 の元を処理する.

このようにして Schreyer 自由分解を計算したのちに, 極小自由分解を構成する.

3 (u,v)-極小分解の計算

(u,v)-極小分解の概念については, [7] を参照. $(u,v) \in \mathbf{Z}^{2n}$ にたいして, $D^{(h)}$ における行列 P_0 の (u,v)-極小分解を求めるには, 前節のアルゴリズムの Step 3. (A)(B)(C) の部分を以下のように変更すればよい.

if $(r$!= 0) {
  (A) r を新しい グレブナ基底の元として加える.
  (B) Reduction の過程から, syzygy L も get できる.
  if (r $\in F_{\text{s-ord}_{(u,v)}(s)-1}$) {
    L は (u,v)–極小分解に必要.
  } else {
    (C) その syzygy の $\mathrm{gr}_{(u,v)}(D^{(h)})^b$ での r の係数は 1 なので,
      この syzygy は極小自由分解には不要と印を付ける.
  }
}

ここで s-ord は, s-degree のように再帰的に定義した, (u,v) に関する次数である:

$$\text{s-ord}_{(u,v)}(ce_i) = \mathrm{ord}_{(u,v)}(c) + \text{s-ord}_{(u,v)}(g_i)$$

と定義する. ここで, ce_i が P_{k+1} にある行ベクトルに現れる要素としたとき, g_i は P_k の第 i 行である. 前節の Step 3 の例では, s-ord$(s) = 2$ だが, s-ord$(r) = 1$ である. したがって, この元は極小自由分解には不必要だが, (u,v)-極小自由分解には必要である.

4 応用の実例

4.1 (u,v)-極小自由分解の計算

まず, 前の章の実例の $(-1,-1,1,1)$ 極小自由分解を Kan/k0 で計算してみよう.

```
% pwd
/usr/home/nobuki/OpenXM/src/k097/lib/minimal
% k0
sm1>macro package : dr.sm1,   9/26,1995 --- Version 12/10, 2000.
sm1>macro package : module1.sm1, 1994 -- Nov 8, 1998
This is kan/k0 Version 1998,12/15
WARNING: This is an EXPERIMENTAL version
sm1>var.sm1 : Version 3/7, 1997

In(1)=Loading startup files (startup.k)   2000, 1/3.
```

```
sm1 version = 3.001203
Default ring is Z[x,h].
WARNING(sm): You rewrited the protected symbol pushVariables.
WARNING(sm): You rewrited the protected symbol popVariables.

In(2)=load["minimal.k"];;
cohom.sm1 is the top of an experimental package to compute restrictions
    略
In(3)=Sweyl("x,y",[["x",-1,"y",-1,"Dx",1,"Dy",1]]);
In(4)=rr = Sminimal([Dx-(x*Dx+y*Dy),Dy-(x*Dx+y*Dy)]);
Automatic homogenization.
tower=[    [    Dx*h , Dy*h , x*Dx^2 ]  , [    x*Dx , Dy , es*x*Dx^2 ]  ,
           [    Dy ]  ]
reductionTable= [
   [    1 , 1 , 2 ]
   [    2 , 1 , 3 ]
   [    2 ]
 ]
1112223
BettiTable ------
  [    0 , 1 , 0 ]
略
In(9)=Pmat(rr[0]);
 [
  [
    [    Dx*h-x*Dx-y*Dy ]
    [    Dy*h-x*Dx-y*Dy ]
    [    x*Dx^2-x*Dx*Dy+y*Dx*Dy-y*Dy^2 ]
  ]
  [
    [    x*Dx-x*Dy+y*Dy+x*h , -y*Dy-x*h , -h+x ]
    [    -Dy+h , Dx-h , 1 ]
  ]
 ]
In(10)=
```

4.2 de Rham コホモロジの計算

Kan/k0 ((u,v)-極小自由分解の計算, Sminimal) および ox_asir (b-関数, f^s の 零化イデアルの計算, bfct, generic_bfct) を用いて, $X = \mathbf{C}^2 \setminus V(xy^9 + y^{10} + x^4)$ のコホモロジ群の次元を計算する例を収録する. 次元は $H^0(X,\mathbf{C})$ が 1, $H^1(X,\mathbf{C})$ が 1, $H^2(X,\mathbf{C})$ が 6 となる. プログラムは, [4] より配布中. コホモロジ群の計算アルゴリズムについては, [5] を参照.

本講究録の野呂氏の原稿にあるように, b-関数の計算の高速化のおかげで, 過去計算できなかった多くの例が計算できるようになった.

```
bash$ pwd
/home/nobuki/OpenXM/src/k097/lib/restriction
bash$ k0
sm1>macro package : dr.sm1,   9/26,1995 --- Version 12/10, 2000.
sm1>macro package : module1.sm1, 1994 -- Nov 8, 1998
This is kan/k0 Version 1998,12/15
```

```
WARNING: This is an EXPERIMENTAL version
sm1>var.sm1 : Version 3/7, 1997

In(1)=Loading startup files (startup.k)   2000, 1/3.
sm1 version = 3.001203
Default ring is Z[x,h].
WARNING(sm): You rewrited the protected symbol pushVariables.
WARNING(sm): You rewrited the protected symbol popVariables.
In(2)=load("demo.k");;
cohom.sm1 is the top of an experimental package to compute restrictions
of all degrees based on restall.sm1 and restall_s.sm1
      略
In(3)=nonquasi2(4,10);
f=x*y^9+y^10+x^4
Step 1: Annhilating ideal (II)
[    [  -2187*x^2*y^6*Dx^2+2268*y^8*Dx^2-1458*x*y^7*Dx*Dy-1296*y^8*Dx*Dy
 -243*y^8*Dy^2-15309*x*y^6*Dx-7776*y^7*Dx-4617*y^7*Dy-17496*y^6
 +12000*x^2*Dx^2+8000*x*y*Dx^2-1008*x^2*Dx*Dy+6720*x*y*Dx*Dy
 +3200*y^2*Dx*Dy+432*x^2*Dy^2-432*x*y*Dy^2+960*y^2*Dy^2+57600*x*Dx
 +40000*y*Dx-2880*x*Dy+9120*y*Dy-9600 ,
 -9*x*y^8*Dx-10*y^9*Dx+y^9*Dy+4*x^3*Dy ,
 -729*x*y^7*Dx-486*y^8*Dx-243*y^8*Dy-2916*y^7+4000*x*y*Dx+216*x^2*Dy
 -240*x*y*Dy+1600*y^2*Dy+16000*y ,
 9*x^2*Dx+10*x*y*Dx+3*x*y*Dy+4*y^2*Dy+36*x+40*y ]  ]
Homogenize_vec is automatically set to 0. grade is set to module1v
Warning: (mmLarger) (tower) switch_function is executed.
Automatic homogenization.
Warning: (mmLarger) (tower) switch_function is executed.
tower=[    [    9*x*Dx^2 , -729*x*Dx*Dy^7 , 1944*x^2*Dy^8 , -324*x*Dy^9 ,
  -2916*y*Dx*Dy^9 ]  ,
[  8*es*x*Dy , -9*es^3*y*Dx , -es^2*Dy , -4*es*Dy^2 , -81*Dy^7 ]  ,
[  -Dy ]  ]
reductionTable= [
   [    2 , 8 , 9 , 9 , 10 ]
   [    9 , 10 , 9 , 9 , 8 ]
   [    9 ]
 ]
2889999991010
BettiTable ------
  [    2 , 1 , 0 ]
321
------------ Note -----------------------------
To get shift vectors, use Reparse and SgetShifts(resmat,w)
To get initial of the complex, use Reparse and Sinit_w(resmat,w)
0: minimal resolution, 3: Schreyer resolution
------------ Resolution Summary  --------------
Betti numbers : [    1 , 3 , 2 ]
Betti numbers of the Schreyer frame: [    1 , 5 , 5 , 1 ]
-----------------------------------------------
Step2: (-1,1)-minimal resolution (Res0)
 [
  [
    [    9*x*Dx^2+10*x*Dx*Dy+3*y*Dx*Dy+4*y*Dy^2-15*Dx*h^2-22*Dy*h^2 ]
    [    -729*x*Dx*Dy^7-486*x*Dy^8-243*y*Dy^8+243*Dy^7*h^2+216*y*Dx^2*h^6
     +4000*x*Dx*Dy*h^6-240*y*Dx*Dy*h^6+1600*y*Dy^2*h^6-240*Dx*h^8-8800*Dy*h^8 ]
    [    1944*x^2*Dy^8-1944*x*y*Dy^8-8748*x*Dx*Dy^6*h^2-3888*x*Dy^7*h^2
     -2916*y*Dy^7*h^2+5832*Dy^6*h^4+648*y^2*Dx^2*h^6+24000*x^2*Dx*Dy*h^6
     -5280*x*y*Dx*Dy*h^6-288*y^2*Dx*Dy*h^6+9600*x*y*Dy^2*h^6
```

```
      -1536*y^2*Dy^2*h^6+48960*x*Dx*h^8-720*y*Dx*h^8-52800*x*Dy*h^8
      +28608*y*Dy*h^8-131040*h^10 ]
  ]
  [
    [    -729*y*Dy^7+2916*Dy^6*h^2-8000*x*Dy*h^6+5568*y*Dy*h^6-16512*h^8 ,
         -9*y*Dx+8*x*Dy-12*y*Dy , 3*Dx+2*Dy ]
    [    729*x*Dy^7-216*y*Dx*h^6-4000*x*Dy*h^6+384*y*Dy*h^6+144*h^8 ,
         9*x*Dx+4*x*Dy-12*h^2 , Dy ]
  ]
 ]

Step3: computing the cohomology of the truncated complex.
Roots and b-function are [    [    0 , 1 , 2 , 3 , 4 , 5 , 6 , 8 ]  ,
[    729*s^15-45927*s^14+1323621*s^13-23116833*s^12+273152682*s^11
 -2308298256*s^10+14373627488*s^9-66928083024*s^8+233870259744*s^7
 -609340017792*s^6+1162915401216*s^5-1572310714368*s^4+1418654187520*s^3
 -760843468800*s^2+181665792000*s ]  ]
[    6 , 6 , 6 ]
i = 0
dim of the i-th truncated complex = 45
i = 1
dim of the i-th truncated complex = 45
i = 2
dim of the i-th truncated complex = 6
Completed (GB with sugar).
Answer is [    6 , 1 , 1 ]
In(4)=
```

いくつかの例題についてタイミングデータを示す. 問題は $\mathbf{C}^n \setminus V(f)$ のコホモロジの次元の計算である. 上の例からわかるように, 計算は 3 ステップより構成されている. Step 1 の $D[1/f]$ と同型になる D-加群の計算には, `ox_asir` を利用, Step 2 の $(-\mathbf{1}, \mathbf{1})$-自由分解の計算には `kan/k0` を利用, Step 3 の restriction のための b-関数の計算には `ox_asir` を利用, 最終的な ker/im の計算には, `kan/k0` を利用している.

問題 1: $n = 2, f = x^p + y^q + xy^{q-1}$ (この問題系列は F.Castro-Jimenéz 氏が計算代数システムのテスト用にもおもしろいと提案したものである). 計測は OpenXM の

```
  $OpenXM: OpenXM/src/kan96xx/Kan/stackmachine.c,v
    1.6 2001/01/27 05:48:46 takayama Exp $
```

時の head version を用い,

```
FreeBSD
CPU: Pentium III/Pentium III Xeon/Celeron (552.07-MHz 686-class CPU)
real memory  = 536854528 (524272K bytes)
avail memory = 518086656 (505944K bytes)
```

なる環境で実行した.

p	q	dim of H^0, H^1, H^2	resolution by `k0`	b-functions by `asir`	Total time
4	5	$[1,1,1]$	$2.37s$	$2.30s$	$4.67s$
4	6	$[1,1,2]$	$2.53s$	$1.35s$	$3.88s$
4	7	$[1,1,3]$	$3.04s$	$3.14s$	$6.18s$
4	8	$[1,1,4]$	$3.49s$	$3.33s$	$6.82s$
5	6	$[1,1,1]$	$5.85s$	$7.40s$	$13.25s$
5	7	$[1,1,2]$	$6.16s$	$10.71s$	$16.87s$
5	8	$[1,1,3]$	$6.63s$	$12.11s$	$18.74s$

この例は, 以前は計算できない系列の問題であったが, `Asir` に最近実装された高性能な b-関数の計算アルゴリズムにより, 簡単に計算できるようになった. 自由分解の betti 数自体は小さい.

問題 2: 次は特異点に関連した 3 変数の多項式をいくつかためしてみる. '?' は不明であることを示す.

f_i	dim of H^0, H^1, H^2, H^3	resolution by `k0`	b-functions by `asir`	Total time
$i=1$	$[1,1,0,0]$	$8.74s$	$0.16s$	$8.90s$
$i=2$	$[1,1,2,2]$	$1263.15s$	$0.40s$	$1263.55s$
$i=3$	?	memory exhaustion	a few seconds	

ここで,

$$\begin{aligned} f_1 &= x^3 - y^2z^2, \\ f_2 &= x^2z + y^3 + y^2z + z^3, \\ f_3 &= yz^2 + x^3 + x^2y^2 + y^6 \end{aligned}$$

である. $(-\mathbf{1},\mathbf{1})$-極小自由分解および Schreyer 分解の Betti 数はそれぞれ,

$$\begin{array}{ll} i=1 & [1,4,5,2],[1,8,16,11,2] \\ i=2 & [1,4,5,2],[1,14,56,96,84,40,10,1] \\ i=3 & ?,[1,53,371,987,1350,1063,499,136,19,1] \end{array}$$

であった. これらの例題の場合は, 自由分解の計算がボトルネックとなっており, 自由分解の計算をより高速化, 省メモリー化することにより, よりおおきな問題に挑戦できると期待できる.

参 考 文 献

[1] Eisenbud, D.: Commutative Algebra with a View toward Algebraic Geometry. Springer, New York, 1995.

[2] D.Grayson, M.Stillman, Macaulay2, a software system for research in algebraic geometry, available at `http://www.math.uiuc.edu/Macaulay2`.

[3] La Scala, R., Stillman, M.: Strategies for computing minimal free resolutions. J. Symbolic Computation **26** (1998), 409–431.

[4] Maekawa, M., Ohara, K., Noro, M., Tamura, K., Takayama, N.: OpenXM project, `http://www.openxm.org`, 2000.

[5] Oaku, T., Takayama, N.: An algorithm for de Rham cohomology groups of the complement of an affine variety via D-module computation. J. Pure Appl. Algebra, **139** (1999), 201–233.

[6] Oaku, T., Takayama, N.: Algorithms for D-modules —restriction, tensor product, localization, and local cohomology groups. J. Pure Appl. Algebra (in press).

[7] Oaku, T., Takayama, N.: D 加群の極小自由分解, 数理研講究録 1171 "D 加群のアルゴリズム" 128–155.

[8] Oaku, T., Takayama, N.: Minimal Free Resolutions of Homogenized D-Modules, preprint.

[9] Saito, M., Sturmfels, B., Takayama, N.: *Gröbner Deformations of Hypergeometric Differential Equations.* Springer, 1999.

[10] Schapira, P.: Microdifferential Systems in the Complex Domain. Springer-Verlag, Berlin, 1985.

[11] A.Leykin, H.Tsai, D-module package for Macaulay 2. `http://www.math.cornell.edu/~htsai`