数理解析研究所講究録
729巻 1990年 140-161

Generators, Relations and Groebner basis

電気通信大学　情報　大久保謙二郎 (Kenjiro Okubo)

1. It has already been 20 years since I proposed a set of first order ordinary differential operators whose group can be determined algebraically. However, except for the extremal cases of Jordan-Pochhammer equations and Generalized Hypergeometric equations, no one has computed the group generators explicitly except SASAI.

The reason for the state of the matter is naturally of great computational complexity of determining a set of parameters where complexity is enhanced by the presence of arbitariness of choosing the parameters within group automorphisms.

The object of this small note is to show by examples, how the algebraic method of Groebner basis is efficient in this type of computations. The use of Groebner package in REDUCE3.3. is by no means a straight forward process. It is my common experience, if the package should be used automatically then an output of several hundreds kilo-bytes comes out.

We need some sort of insight and a great number of experiments to achieve the actual computations. And hence the note will give no explcitly stated theorems, but some messy sample outputs.

2. Throughout this paper we stick to the non-logarithmic cases for the system of hypergeometric equations;

$[tI-B]x'=Ax$

where x is unknown vector of dimension d, and A and B are diagonalizable matrices with eigenvaluse with various multiplicities. We assume necessary conditions upon A and B which will enable use that the system is irreducible and logarithmic free. What matters in the following samples, is the way these multiplicities are arranged.

By {m1,m2m,,,mf;n1,n2,...,nl} we mean that the sums:

$$m1 + m2 + ...+ mf = d;$$

$$n1 + n2 + \quad ..+ nl = d;$$

with the accesory parameter free condition:

$$m1^2 + m2^2 +.....+mf^2 + n1^2 + ...+ nl^2 = d^2 -d -2$$

where d is the dimension of the unknown vector.
With some irreducibility conditions which I would not state here, I can pick up a fundamental set of solutions for which generators have some specific forms described below, with unknown parameters (u1,u2,...,;v1,v2,....) and known characteristic exponents; (e1,e2,....;f1,f2,....) with Riemann-Fuchs relation:

e1*e2*... =f1*f2*.....

3. To be more specific, in the case of third order generalized hypergeometric differential equations, we have:

```
% The hypergoematric series of order 3.Second case in 3rd order.
% There are only two generators with given exponents e1,e1,e2.
% Eigenvause are separated with exponesnts f1,f2,f3.

genr1:= mat ( ( e1, 0, (e1-1)*u1 ),
              ( 0 ,e2, (e2-1)*u2 ),
              ( 0,  0,  1 )
             )$


genr2:= mat ( ( 1,  0,  0 ),
              ( 0,  1,  0 ),
              ( (e3-1)*v1, (e3-1)*v2, e3)
            )$
% There are only four parameters.

gmat:=genr1*genr2$

% We have 3 equations:
 identity:=mat ( (1,0,0),
                 (0,1,0),
                 (0,0,1)
               )$

gg:=det ( gmat-f*identity );
```

```
gg0:=coeffn(gg,f,0);
gg1:=coeffn(gg,f,1);
gg2:=coeffn(gg,f,2);
% Very well known result is that:

let u1*v1=p1$
let u2*v2=p2$

% is solvable as a system of first order equations.

load "groebner"$

groebner({gg1,gg2},{p1,p2});

end$
end;
```

This is actually a program in REDUCE3.3. with the outcome:

```
GG :=  - (E1*E2*E3*F*U1*V1 + E1*E2*E3*F*U2*V2 - E1*E2*E3 - E1*E2*F*U1

                                                 2
         *V1 - E1*E2*F*U2*V2 + E1*E2*F - E1*E3*F *U1*V1 - E1*E3*F*U2

                          2           2                    2
         *V2 + E1*E3*F + E1*F *U1*V1 - E1*F  + E1*F*U2*V2 - E2*E3*F
```

```
                                            2           2
*U2*V2 - E2*E3*F*U1*V1 + E2*E3*F + E2*F *U2*V2 - E2*F  + E2

             2          2           2   3   2
*F*U1*V1 + E3*F *U1*V1 + E3*F *U2*V2 - E3*F  + F  - F *U1*

      2
V1 - F *U2*V2)

gg0:=coeffn(gg,f,0);

% The first result is the Riemann-Fuchs relation:

GG0 := E1*E2*E3

gg1:=coeffn(gg,f,1);

GG1 :=  - (E1*E2*E3*U1*V1 + E1*E2*E3*U2*V2 - E1*E2*U1*V1 - E1*E2*U2*

           V2 + E1*E2 - E1*E3*U2*V2 + E1*E3 + E1*U2*V2 - E2*E3*U1*V1

           + E2*E3 + E2*U1*V1)

gg2:=coeffn(gg,f,2);


GG2 := E1*E3*U1*V1 - E1*U1*V1 + E1 + E2*E3*U2*V2 - E2*U2*V2 + E2 - E3
```

```
      *U1*V1 - E3*U2*V2 + E3 + U1*V1 + U2*V2


groebner({gg1,gg2},{p1,p2});


                                    3     2
                                  E1  - E1  - E1*E2*E3 + E2*E3
{(E1*E3 - E1 - E3 + 1)*P1 + ------------------------------,
                                       2
                                     E1  - E1*E2 - E1 + E2

        2          2             2    2          2       2
 P2 + (E1 *E3 - E1*E2  - E1*E3 + E2 )/(E1 *E2*E3 - E1 *E2 - E1 *E3 +

     2        2          2                 2        2
   E1  - E1*E2 *E3 + E1*E2  + E1*E3 - E1 + E2 *E3 - E2  - E2*E3 + E2

   )}
```

This output says the following:

(A) The quantities P1 and P2 which were made from the unknown parameter u1,u2,v1,v2 can be expressed rationally in terms of e's and f's.

(B) These terms cannot be determined if either of the divisors;

(e1-e2)*f2, and f1*f2*f3 = e1*e2*e3

is zero. These conditons are actually the conditions we imposed upon for the irreducibility and logarithmic free conditions.

(C) The configuration for the multiplicity is (2,1;1,1,1). There were four unknowns but there are two arbitraries. That is, we can determine unknown parameters within the diagonal transformations of generators.

(D) The generators is in GL(2,K) where K is the field extended from rational number field by the addition of {e1,e2,f1,f2} with the condition e1*e2*e3 = f1*f2*f3.

The results (D) had been anticipated, but was not proved, and remains unproved yet for the general situation.

4. We have done similar computations for all the possible accessory parameter-free systems of equations up to order 5. Up to this time, there seems to be no general principle to take parameters except to name those pairs which are diagonally symmetric with the similar name, and name their product with new names will greatly depress the size of outputs.
To explain the situation, we take another equation up in third order. The computation can be carried out on PC9801 if you take factor off.

It has been carried out on NEWS with 4MB memory for Reduce3.3..

```
% THE FIRST THIRD ORDER EQUATION Jordan-Pochhammer.
% MULTIPLICITY IS 2,1;1,1,1
ARRAY SCOL(2),SROW(2);
MATRIX SMAT(2,2),GMAT(3,3);
PROCEDURE MINOR;
 BEGIN
  scalar j,k,sdim;
  sdim:=2;
FOR J:=1:sdim
  DO << FOR K:=1:SDIM
  do   SMAT(J,K):= GMAT( SCOL(J),SROW(K) ) >>;
 RETURN DET(SMAT)
END;

%genr1:= mat( ( e1, (e1-1)*u1, (e1-1)*u2 ),
%             ( 0,  1,  0  ),
%             ( 0,  0,  1  )
%             )$
%genr2:=mat ( ( 1,  0, 0  ),
%             ( (e2-1)*v1, e2, (e2-1)*u3),
%             ( 0, 0, 1  )
%             )$
%genr3:= mat ( ( 1, 0, 0  ),
%              ( 0, 1,  0 ),
%              ((e3-1)*v2,(e3-1)*v3, e3 )
%             )$
```

```
%gmat:=genr1*genr2*genr3$
% The expression is not compact. We use the product formula Theorem.4.,
% of reference [1].


lmat:= mat(
            ( e1-1, 0, 0 ),
            ((e2-1)*v1,e2-1,0),
            ((e3-1)*v2,(e3-1)*v3,e3-1)
            )$
umat:= mat (
             ( 0, (e1-1)*u1,(e1-1)*u2),
             ( 0,   0, (e2-1)*u3),
             ( 0, 0, 0)
           )$

identity:= mat (
                 (1,0,0),
                 (0,1,0),
                 (0,0,1)
                  )$

gmat:=((lmat+f*umat)-(f-1)*identity)$

% The leading coefficient of
 gg0:=det(gmat)$
% is just simply -1 in f.
% the constant term, namely the determinat at f=0 is Riemann-Fuchs relation.
```

```
% we know of two relations significant; linear term and the term of second
% degree.
coeff(gg0,f);

gg01:=coeffn(gg0,f,1)$
gg02:=coeffn(gg0,f,2)$

scol(1):=1$ scol(2):=2$
srow(1):=1$ srow(2):=2$
gg1:=minor()$

scol(1):=1$ scol(2):=2$
srow(1):=2$ srow(2):=3$
gg2:=minor()$
gg2:=v2*gg2$

scol(1):=2$ scol(2):=3$
srow(1):=1$ srow(2):=2$
gg3:=minor()$
gg3:=u2*gg3$

scol(1):=2$ scol(2):=3$
srow(1):=2$ srow(2):=3$
gg4:=minor()$

% from the looks of the output it is obvious that if we introduce

let u1*v1=p1$
```

```
let u2*v2=p2$
let u3*v3=p3$
let v1*v3*u2=q1$
let u1*u3*v2=q2$

%then we know gg01,gg02,gg2,gg3 are linear in 5 unknowns p1,p2,p3,q1,q2.
%where p1,p2 are already solvable in gg1 and gg4.
offnat$
on factor$
on allfac$

gg1;
gg4;

load "groebner"$
groebner({gg01,gg02,gg2},{q1,q2,p2});

end$
end;

% Output is lengthy mixed up expressions. Yet we do have very neat
% informations for the unsovability conditions:
% First coefficient of p1 and p3 in gg1 and gg4 respectively are
% (e1-f)*(e2-f) and (e3-f)*(e2-f)
% which are exactly the conditions we excluded.
%
% Secondly, from the Groebner basis, we know q1 has coefficient:
% (e1-1)*(e2-1)*(e3-1), the non-logarithmic condition.
```

```
%
% q2 has similarly coefficient (e1-f)*(e2-f) with a numerator
% e2*(e1-1)*(e1-1)*(e2-1)*(e3-1).
%
% p2 is linear in p1 and p3 with the same numerator as q2.
% In this way, we can assure us with the irreducibity condition.
%
% The output with grobner package with random choice of equations
% with 'torder' totaldegree in u's and v's goes up to tragic output
% of size 57KB.
% End of jp order 3.
```

The following Program is run on News1450 with 4MB reduce3.3. The output is similar to the above, yet one can see more clearly the validity of underlying theory. There are 12 variables to be determined, and we have essentially 10 equations. One relation is just a result of Riemann-Fuchs relation. Among these 12 variables we do have 3=4-1 redundancy of diagonal transformations.

We may derive other relations but they must be trivially satisfied. One may test the following program to confirm the comments stated here.

```
% THE FIRST fourth ORDER EQUATION Jordan-Pochhammer.
% MULTIPLICITY IS 3,1;1,1,1,1
ARRAY SCOL(2),SROW(2);
MATRIX SMAT(2,2),GMAT(4,4);

PROCEDURE MINOR;
 BEGIN
```

```
  scalar j,k,sdim;
  sdim:=2;
FOR J:=1:sdim
  DO << FOR K:=1:SDIM
  do   SMAT(J,K):= GMAT( SCOL(J),SROW(K) ) >>;
 RETURN DET(SMAT)
END;

Procedure lsolve(leqn,var0);
 Begin
  scalar leqn,var0;
  leqn:=num(leqn);
  var0:=-coeffn(leqn,var0,0)/coeffn(leqn,var0,1);
 return var0
 End;

off nat$

% This is the monodromy matrix of infinity.
gmat:= mat (
            (e1-f,    f*(e1-1)*u1,f*(e1-1)*u2,f*(e1-1)*u3),
            ((e2-1)*v1,e2-f,      f*(e2-1)*u4,f*(e2-1)*u5),
            ((e3-1)*v2,(e3-1)*v4, e3-f,       f*(e3-1)*u6),
            ((e4-1)*v3,(e4-1)*v5, (e4-1)*v6,  e4-f        )
           )$

gg0:=det(gmat)$
gg00:=coeffn(gg0,f,0)$
```

```
gg01:=coeffn(gg0,f,1)$
gg02:=coeffn(gg0,f,2)$
gg03:=coeffn(gg0,f,3)$

let u1*v1=p1$
scol(1):=1$
scol(2):=2$
srow(1):=1$
srow(2):=2$
gg1:=minor();

% p1 is determined rationally.
p1:=lsolve(gg1,p1);


let u2*v2=p2$
scol(1):=1$
 scol(2):=3$
srow(1):=1$
 srow(2):=3$
gg2:=minor();

% p2 is rationally expressed.
 p2:=lsolve(gg2,p2);


let u3*v3=p3$
scol(1):=1$
```

```
 scol(2):=4$
srow(1):=1$
 srow(2):=4$
gg3:=minor(););
% p3 is rational.
 p3:=lsolve(gg3,p3);

let u4*v4=p4$
scol(1):=2$
 scol(2):=3$
srow(1):=2$
 srow(2):=3$
gg4:=minor();
% p4 rational.
 p4:=lsolve(gg4,p4);

let u5*v5=p5$
scol(1):=2$
 scol(2):=4$
srow(1):=2$
 srow(2):=4$
gg5:=minor();
% p5 is rational.
 p5:=lsolve(gg5,p5);

let u6*v6=p6$
scol(1):=3$
 scol(2):=4$
```

```
srow(1):=3$
 srow(2):=4$
gg6:=minor();
% p6 is rational.
 p6:=lsolve(gg6,p6);
%To check consistency of the system;

% Instead of the procedure Lsolve, we may do as well by the lines:
%load groebner$
%groebner({gg1,gg2,gg3,gg4,gg5,gg6},{p1,p2,p3,p4,p5,p6});

let u1*v2*u4=q1$
scol(1):=1$
 scol(2):=2$
srow(1):=2$
 srow(2):=3$
gg7:=v2*minor();

% q1 is linear in p2
 q1:=lsolve(gg7,q1);

let u2*u5=q2$
let u3*u4=q2$
srow(1):=3$
 srow(2):=4$
gg8:=minor();
% u2u5=u3u4 are the same.
```

```
let u4*u6*v5=q4$
scol(1):=2$
 scol(2):=3$
srow(1):=3$
 srow(2):=4$
gg9:=v5*minor();
% q4 is linear in p5.
 q4:=lsolve(gg9,q4);

%To make things simpler we set x=u1,y=u2,z=u3.

let u1=x;
let u2=y;
let u3=z;

v1:=p1/x;
v2:=p2/y;
v3:=p3/z;
u4:=q1/u1/v2;
v4:=p4/u4;
u5:=u4*u3/u2;
v5:=p5/u5;

%We know q4=u4*(p5/u5)*u6 and u4/u5=u2/u3=y/z
%consequently, we solve u6 in terms of y,z;

u6:=q4/u4/v5;
v6:=p6/u6;
```

```
% We have 3 equations in 3 unknowns x,y,z, by gg01,gg02 and gg03.

f1:=e1*e2*e3*e4/f/f/f$
gg01:=num(gg01+f*f*f+3*f1*f*f);
gg02:=num(gg02-3*f1*f-3*f*f);
gg03:=num(gg03+3*f+f1);

%Since the ouputs of all of these are trivial we see u1,u2,u3 are
%arbitrarily definabale constants.

end$
end;
----------------------------------------------------------------------
*** Tuneup Lisp/Sony/News  (Ver. 6.7 Mon Aug 15 16:33:24 1988) : 3670016 byt
es
Copyright (c) 1988 Forbs system Co. Ltd. All rights reserved.
REDUCE 3.3, 15-Jan-88 ...

1: in "e412.red"$

     MINOR

     LSOLVE

     GG1 := F**2 - F*P1*E2*E1 + F*P1*E2 + F*P1*E1 - F*P1 - F*E2 - F
     *E1 + E2*E1$
```

```
P1 := (F**2 - F*E2 - F*E1 + E2*E1)/(F*(E2*E1 - E2 - E1 + 1))$

GG2 := F**2 - F*P2*E3*E1 + F*P2*E3 + F*P2*E1 - F*P2 - F*E3 - F
*E1 + E3*E1$

P2 := (F**2 - F*E3 - F*E1 + E3*E1)/(F*(E3*E1 - E3 - E1 + 1))$

GG3 := F**2 - F*P3*E4*E1 + F*P3*E4 + F*P3*E1 - F*P3 - F*E4 - F
*E1 + E4*E1$

P3 := (F**2 - F*E4 - F*E1 + E4*E1)/(F*(E4*E1 - E4 - E1 + 1))$

GG4 := F**2 - F*P4*E3*E2 + F*P4*E3 + F*P4*E2 - F*P4 - F*E3 - F F
*E2 + E3*E2$

P4 := (F**2 - F*E3 - F*E2 + E3*E2)/(F*(E3*E2 - E3 - E2 + 1))$

GG5 := F**2 - F*P5*E4*E2 + F*P5*E4 + F*P5*E2 - F*P5 - F*E4 - F
*E2 + E4*E2$

P5 := (F**2 - F*E4 - F*E2 + E4*E2)/(F*(E4*E2 - E4 - E2 + 1))$

GG6 := F**2 - F*P6*E4*E3 + F*P6*E4 + F*P6*E3 - F*P6 - F*E4 - F
*E3 + E4*E3$

P6 := (F**2 - F*E4 - F*E3 + E4*E3)/(F*(E4*E3 - E4 - E3 + 1))$

GG7 := (F**3 + F**2*Q1*E3*E2*E1 - F**2*Q1*E3*E2 - F**2*Q1*E3*
```

```
E1 + F**2*Q1*E3 - F**2*Q1*E2*E1 + F**2*Q1*E2 + F**2*Q1
*E1 - F**2*Q1 - F**2*E3 - F**2*E2 - F**2*E1 + F*E3*E2
+ F*E3*E1 + F*E2*E1 - E3*E2*E1)/(E3 - 1)$

Q1 :=  - (F**3 - F**2*E3 - F**2*E2 - F**2*E1 + F*E3*E2 + F*E3*
E1 + F*E2*E1 - E3*E2*E1)/(F**2*(E3*E2*E1 - E3*E2 -
E3*E1 + E3 - E2*E1 + E2 + E1 - 1))$

GG8 := 0$

GG9 := (F**3 + F**2*Q4*E4*E3*E2 - F**2*Q4*E4*E3 - F**2*Q4*E4*
E2 + F**2*Q4*E4 - F**2*Q4*E3*E2 + F**2*Q4*E3 + F**2*Q4
*E2 - F**2*Q4 - F**2*E4 - F**2*E3 - F**2*E2 + F*E4*E3
+ F*E4*E2 + F*E3*E2 - E4*E3*E2)/(E4 - 1)$

Q4 :=  - (F**3 - F**2*E4 - F**2*E3 - F**2*E2 + F*E4*E3 + F*E4*
E2 + F*E3*E2 - E4*E3*E2)/(F**2*(E4*E3*E2 - E4*E3 -
E4*E2 + E4 - E3*E2 + E3 + E2 - 1))$

V1 := (F**2 - F*E2 - F*E1 + E2*E1)/(F*X*(E2*E1 - E2 - E1 + 1))
$

V2 := (F**2 - F*E3 - F*E1 + E3*E1)/(F*Y*(E3*E1 - E3 - E1 + 1))
$

V3 := (F**2 - F*E4 - F*E1 + E4*E1)/(F*Z*(E4*E1 - E4 - E1 + 1))
$
```

```
U4 :=  - (Y*(F - E2))/(F*X*(E2 - 1))$

V4 :=  - (X*(F**2 - F*E3 - F*E2 + E3*E2))/(Y*(F*E3 - F - E3*E2
 + E2))$

U5 :=  - (Z*(F - E2))/(F*X*(E2 - 1))$

V5 :=  - (X*(F**2 - F*E4 - F*E2 + E4*E2))/(Z*(F*E4 - F - E4*E2
 + E2))$

U6 :=  - (Z*(F**2 - F*E3 - F*E2 + E3*E2))/(F*Y*(F*E3 - F - E3*
E2 + E2))$

V6 :=  - (Y*(F**3 - F**2*E4 - F**2*E3 - F**2*E2 + F*E4*E3 + F*
E4*E2 + F*E3*E2 - E4*E3*E2))/(Z*(F**2*E4 - F**2 -
 F*E4*E3 - F*E4*E2 + F*E3 + F*E2 + E4*E3*E2 - E3*E2))$

GG01 := 0$

GG02 := 0$

GG03 := 0$

 *** end of run
% Isn't it beautiful?
```

Hence the outputs of the similar procedure are clumsy and suppressed. We have used similar procedures for all the systems with order 5.

Except for the Generalized hypergeometric equations, we have not obtained a neat format which is self-explanatory for dimension higher than 5.

Groebner Package is NOT sufficient for the automatic computation for these cases. We had to go interactive sessions with intermediate outputs all the time.

Reference

[1] K.Okubo: On the group of Fuchsian Equations. Seminar Note, Tokyo Metropolitan University, 1987.