数理解析研究所講究録
第 612 巻 1987 年 25-37

On solving a system of algebraic equations by using a Gröbner basis

Shuichi Moritsugu (森継 修一)

Dept. of Information Science, Faculty of Science, Univ. of Tokyo

Abstract

This paper proposes a new algorithm to reduce a system of algebraic equations by using a Gröbner basis with total degree ordering. The reduced system by this method is suitable for numerical calculation, reducing the accumulation of errors largely. Timing data are presented, showing the superiority of the new method to the conventional method using Gröbner bases with lexicographic ordering.

1. Introduction

The Gröbner basis method is very powerful for solving a system of algebraic equations([3]), and a number of packages have been implemented([1],[4]). The reported results of computation show, however, that problems which can be solved by the present computers are restricted to small-sized ones and that the algorithm must be improved.

This paper considers how to reduce a given system of algebraic equations and how to solve more complicated problems. In the next section, we review the well-known methods and discuss advantages and disadvantages of them. In section 3, we present a new algorithm which meets the requirements for computational efficiency and numerical computation. The timing data show that the new method is actually superior to the straightforward calculation of a Gröbner basis with lexicographic ordering when the problem size is not small.

2. Conventional methods

2.0 Preparation

We assume that the polynomials are in a ring $R = K[x_1, x_2, \cdots, x_n]$, with K a number field, and that the variable ordering is $x_1 > x_2 > \cdots > x_n$. We abbreviate "lexicographic ordering" to "$>_L$" and "total degree ordering" to "$>_T$", and we follow to [3] for other basic notations and definitions.

2.1 Trinks' algorithm

The following theorem[9] gives an essential property of Gröbner bases with $>_L$.

<u>Theorem</u> [Trinks]

Let G be a Gröbner basis with $>_L$, then

$$\text{ideal}(G) \cap K[x_i, \cdots, x_n] = \text{ideal}(G \cap K[x_i, \cdots, x_n]) \qquad (1 \leq \forall i \leq n)$$

provided $x_1 > x_2 > \cdots > x_n$.

□

This theorem shows that the variables are "triangulated" in G and that G has only one polynomial that belongs to $K[x_n]$. On the basis of this theorem, the following algorithm is obtained as "Method 6.10" in [3]. (Algorithms in this paper are written in a REDUCE-like language.)

<u>Algorithm 2.1</u> [Trinks] [Solving a system of algebraic equations]

```
% input : polynomials {f_1, ..., f_s}  in K[x_1, ..., x_n] ;
% output : all solutions of the system of algebraic equations ;
%                 { f_1 = f_2 = ... = f_s = 0 } ;
G := a reduced Gröbner basis of (f_1, ..., f_s) with >_L;
p(x_n) := the polynomial in G ∩ K[x_n];
X_n := {(a) | p(a) = 0};
for i := n down to 2 do
  begin X_{i-1} := ∅;
      for all (a_i, ..., a_n) ∈ X_i do
         begin H :={g(x_{i-1}, a_i, ..., a_n) |
                    g ∈ G∩(K[x_{i-1}, ..., x_n]−K[x_i, ..., x_n])};
               p(x_{i-1}) := GCD(H);
               X_{i-1} := X_{i-1} ∪ {(a,a_i, ..., a_n) | p(a) = 0 };
         end;
  end;
return X_1;
```

□

This method is very elegant and simple but has disadvantages.

(i) Numerical solution a_i has an error which propagates and accumulates as the computation proceeds.

(ii) The complexity of computation of Gröbner basis with $>_L$ is very large, therefore, only small-sized problems can be solved.

Table 1 shows the computation time of Gröbner bases with $>_L$ using the package implemented by the author in SLISP-REDUCE3.0[5][6][7] on FACOM M380.

Table 1 CPU time (milliseconds) with $>_L$

Problem	Order(1)	Order(2)	Order(3)	Order(4)
KATSURA1	20	19	19	20
KATSURA2	61	52	58	51
KATSURA3	>3,600,000	924	>3,600,000	944
KATSURA4	******	******	******	******

The Problems are taken from [1][7]. "******" shows overflow of memory storage (8 M bytes). Variable ordering is decided as follows.

(i) Order(1) : $\cdots > U_5 > U_4 > U_3 > U_2 > U_1 > U_0$

(ii) Order(2) : $U_0 > \cdots > U_5 > U_4 > U_3 > U_2 > U_1$

(iii) Order(3) : $U_0 > U_1 > U_2 > U_3 > U_4 > U_5 > \cdots$

(iv) Order(4) : "Heuristically optimized" following [1].

KATSURA1 : $U_1 > U_0$, KATSURA2 : $U_2 > U_0 > U_1$, KATSURA3 : $U_3 > U_0 > U_2 > U_1$,

KATSURA4 : $U_4 > U_0 > U_3 > U_2 > U_1$, KATSURA5 : $U_5 > U_0 > U_4 > U_3 > U_2 > U_1$

N.B. For KATSURA4, $U_4 > U_0 > U_2 > U_3 > U_1$ is a good ordering([4]) and a Gröbner basis was obtained in 1,083,899 milliseconds.

2.2 Buchberger's algorithm

If we can choose freely the ordering in the Gröbner bases construction, we better choose $>_T$ when the size of problem is not small. Timing data of computation of Gröbner bases with $>_T$ for the same problems as above are shown in *Table* 2.

Table 2 CPU time (milliseconds) with $>_T$

Problem	Order(1)	Order(2)	Order(3)	Order(4)
KATSURA1	21	19	19	21
KATSURA2	66	60	61	57
KATSURA3	804	695	813	688
KATSURA4	28,407	31,096	36,584	29,849
KATSURA5	2,292,003	>3,600,000	2,633,969	>3,600,000

Given a Gröbner basis, where the ordering may be any one, we can calculate a univariate polynomial belonging to the ideal by the method of undetermined coefficients[2]. The following algorithm is an improvement of "Method 6.11" in [3], because a system of linear equations has to be solved only once.

Algorithm 2.2 [Finding a univariate polynomial $p \in I$]

% input : a Gröbner basis G of $I = (f_1, \cdots, f_s)$;

% output : a univariate polynomial $p \in I$ with the least degree ;

$N := \prod_{i=1}^{s} deg(f_i)$; % upper bound of $deg(p)$ by Bézout's theorem;

for i := 0 : N do p_i := $NormalForm(G, x_n^i)$;

solve the equation $d_0p_0 + \cdots + d_Np_N = 0$;

$p := d_0 + d_1x_n + \cdots + d_Nx_n^N$;

if the solution $(d_0, \cdots, d_N)$ has arbitrary constants
then determine them so that the degree of p may be minimal;

return p;

□

Using Algorithm 2.2, the following algorithm is obtained. (Cf. "Method

6.12" in [3].)

Algorithm 2.3 [Buchberger] [Solving a system of algebraic equations]

% input : polynomials $\{f_1, \cdots, f_s\}$ in $K[x_1, \cdots, x_n]$;

% output : all solutions of the system of algebraic equations ;

% $\{ f_1 = f_2 = \cdots = f_s = 0 \}$;

G := a reduced Gröbner basis of $(f_1, \cdots, f_s)$ with $>_T$;

$p(x_n)$:= the polynomial in $ideal(G) \cap K[x_n]$ obtained by Algorithm 2.2;

$X_n := \{(a) \mid p(a) = 0\}$;

for $i := n$ down to 2 do

begin $X_{i-1} := \phi$;

for all $(a_i, \cdots, a_n) \in X_i$ do

begin $H := \{g(x_1, \cdots, x_{i-1}, a_i, \cdots, a_n) \mid g \in G\}$;

H := a Gröbner basis of H with $>_T$;

p := the polynomial in $ideal(H) \cap K[x_{i-1}]$;

$X_{i-1} := X_{i-1} \cup \{(a, a_i, \cdots, a_n) \mid p(a) = 0\}$;

end;

end;

return X_1;

□

From the viewpoint of computational efficiency, this algorithm is much better than Algorithm 2.1 as *Tables* 1 and 2 show. Nevertheless, this method has also disadvantages.

(i) Numerical accuracy is not maintained. This is the same in Algorithm 2.1 also.

(ii) Nested calculation of Gröbner bases with $>_T$ in Algorithm 2.3 is wasteful.

3. New method

In the new algorithm to be stated below, the above disadvantages are removed. That is,

(i) The reduced system is suitable for numerical calculation, that is, it does not suffer from error propagation and error accumulation.

(ii) Calculation of a Gröbner basis with $>_T$ is performed only once.

Unfortunately, we must set a restriction in applying the new algorithm, which we think is not so serious because most problems can be transformed to satisfy it by changing the order of variables.

Assumption

Let the solutions of an n-variable system of algebraic equations be $(c_1^{(i)},\cdots,c_n^{(i)})$ $(1\leqq i\leqq \exists l)$, where multiple roots are counted only once. Then, we assume $\forall i\neq j$, $c_n^{(i)}\neq c_n^{(j)}$.

□

If this assumption is satisfied, we can reduce the original system of equations into the following form:

$$\{\ x_1 - h_1(x_n) = 0\ ,\ \cdots\ ,\ x_{n-1} - h_{n-1}(x_n) = 0\ ,\ h_n(x_n) = 0\ \}$$

$$h_i \in K[x_n] \qquad (1\leqq i\leqq n)\ .$$

When this system is passed into numerical calculation, error occurs at only two steps : (i) solving the univariate equation $h_n(x_n)=0$; (ii) evaluating the polynomials $h_i(x_n)(1\leqq i\leqq n-1)$. Therefore, it is much more advantageous than Algorithm 2.1 and 2.3 from the viewpoint of numerical accuracy. A Gröbner basis with $>_L$ under proper variable ordering may be of such a form, but we construct this by means of a Gröbner basis with $>_T$.

The polynomial $h_n(x_n)$ is obtained by Algorithm 2.2, and polynomials $x_i - h_i(x_n)$ can be computed by the method of undetermined coefficients like Algorithm 2.2.

Algorithm 3.1 [Calculating a polynomial p of the form $x_i - h_i(x_n)$]

% input : a Gröbner basis G of I, index i of variable x_i ;

% output : a polynomial $p = x_i - h_i(x_n)$;

$M := deg(h_n(x_n)) - 1$;

for j := 1 : M do $p_j := NormalForm(G\ ,\ x_n^j)$;

solve the equation $NormalForm(G\ ,\ x_i) + d_0 p_0 + \cdots + d_M p_M = 0$;

$p := x_i + d_0 + d_1 x_n + \cdots + d_M x_n^M$;
return p;

□

The new algorithm is described as follows.

Algorithm 3.2 [New method for solving a system of algebraic equations]

% input : polynomials $\{f_1, \cdots, f_s\}$ in $K[x_1, \cdots, x_n]$;

% output : all solutions of the system of algebraic equations ;

% $f_1 = f_2 = \cdots = f_s = 0$;

G := a reduced Gröbner basis of $(f_1, \cdots, f_s)$ with $>_T$;

$h_n(x_n)$:= the polynomial in $G \cap K[x_n]$ computed by Algorithm 2.2;

for i := 1 : n-1 do compute $x_i - h_i(x_n)$ by Algorithm 3.1;

$X := \{(h_1(a), \cdots, h_{n-1}(a), a) \mid h_n(a) = 0\}$;
return X;

□

Example

Let the system of equations be $\{x^2 + y^2 - 2 = 0\ ,\ x^2 - y = 0\}$. A Gröbner basis with $>_T$ and $x>y$ is $\{x^2 - y\ ,\ y^2 + y - 2\}$.

Since y does not satisfy the Assumption, we construct a univariate polynomial of x. Applying Algorithm 2.2,

$d_4x^4 + d_3x^3 + d_2x^2 + d_1x + d_0 = 0 \rightarrow d_4(-y + 2) + d_3xy + d_2y + d_1x + d_0 = 0$,
therefore, $d_1 = d_3 = 0$, $d_2 = d_4 = 1$, $d_0 = -2$, and we obtain

$x^4 + x^2 - 2 \cdots$ (i).

Applying algorithm 3.1,

$y + d_3x^3 + d_2x^2 + d_1x + d_0 = 0 \rightarrow y + d_3xy + d_2y + d_1x + d_0 = 0$,
therefore, $d_0 = d_1 = d_3 = 0$, $d_2 = -1$, and we obtain $y - x^2 \cdots$ (ii).

Numerical solutions are computed as follows. Solving (i) = 0, $x = \pm 1$, $\pm\sqrt{2}i$. Substituting them to (ii), we obtain $(x , y) = (\pm 1 , 1) , (\pm\sqrt{2}i , -2)$.

□

Timing data for the same problems (Order(1)) as in section 2.1 are given in *Table* 3. Systems of linear equations were solved by the SOLVE package of REDUCE3.0.

Table 3 CPU time (milliseconds) of New method

Problem	G-Basis	Transform	Total
KATSURA1	21	166	187
KATSURA2	66	324	390
KATSURA3	804	1,834	2,638
KATSURA4	28,407	69,056	97,463
KATSURA5	2,292,003	20,091,180	22,383,183

"Transform" means computation of Algorithm 2.2 and 3.1.

4. Conclusion

It is concluded from timing data that Gröbner bases with $>_T$ are more advantageous than $>_L$ for large-sized problems. Comparing *Table* 1 with 3, it is confirmed that the new algorithm has more practical efficiency than the straightforward calculation of Gröbner bases with $>_L$.

Acknowledgements

I would like to express my sincere gratitude to Dr. T.Sasaki, Professor H.Kobayashi, Mr. A.Furukawa, for valuable and helpful discussions and assistance. I also would like to express special thanks to Professor B.Buchberger for reading an early manuscript and comments.

References

[1] Böge,W., Gebauer,R., Kredel,H., Some Examples for Solving Systems of Algebraic Equations by Calculating Gröbner Bases, J.Symbolic Comput., Vol.2,1, 1986, pp.83-98.

[2] Buchberger,B., Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems, Aequationes Mathematicae, Vol.4,3, 1970, pp.374-383.

[3] Buchberger,B., Gröbner bases : An algorithmic method in polynomial ideal theory, N.K.Bose, ed., Recent Trends in Multidimensional Systems Theory, D.Reidel Publ.Comp., 1985, Chapter 6.

[4] Czapor,S.R., Geddes,K.O., On Implementing Buchberger's Algorithm for Gröbner bases, Proc. SYMSAC '86, N.Y., ACM, 1986, pp.233-238.

[5] Fitch,J., MANUAL for STANDARD LISP on IBM SYSTEM 360 AND 370, University of Utah, Symbolic Computation Group, Technical Report, No. TR-6, 1978.

[6] Hearn,A.C., REDUCE USER'S MANUAL Version 3.0, Santa Monica, The Rand Corporation, 1983.

[7] Katsura,S., Fukuda,W., The solution of integral equation at T=0 of the effective field of spin glass, REDUCE Programming Reference Book, Vol.2, Computer Centre of University of Tokyo, 1985, pp.337-346 (In Japanese).

[8] Marti,J.B., Hearn,A.C., Griss,M.L., Griss,C., STANDARD LISP REPORT, Utah Computational Physics, UCP-60, 1978.

[9] Trinks,W.L., Über Buchbergers Verfahren, Systeme algebraischer Gleichungen

zu lösen, J.Number Theory, Vol.10,4, 1978, pp.475-488.

Appendix

The results of Algorithm 3.2 for a series of Katsura's problems are shown. Long expressions are omitted leaving first several terms.

```
%---------------------------------------------------------------------;
%%%%% KATSURA'S PROBLEM (1) %%%%%
===== INPUT POLYNOMIALS =====
         2      2
F(1)=2*U1  + U0  - U0
F(2)=2*U1 + U0 - 1
===== REDUCED SYSTEM =====
        2
FF(0)=U0  - 4/3*U0 + 1/3
FF(1)=U1 + 1/2*U0 - 1/2
%---------------------------------------------------------------------;
%%%%% KATSURA'S PROBLEM (2) %%%%%
===== INPUT POLYNOMIALS =====
         2       2      2
F(1)=2*U2  + 2*U1  + U0  - U0
F(2)=2*U2*U1 + 2*U1*U0 - U1
F(3)=2*U2 + 2*U1 + U0 - 1
===== REDUCED SYSTEM =====
        4            3            2
FF(0)=U0  - 46/21*U0  + 34/21*U0  - 10/21*U0 + 1/21
                         3           2
FF(1)=U1 + 21/2*U0  - 71/4*U0  + 17/2*U0 - 5/4
                         3           2
FF(2)=U2 - 21/2*U0  + 71/4*U0  - 8*U0 + 3/4
%---------------------------------------------------------------------;
%%%%% KATSURA'S PROBLEM (3) %%%%%
===== INPUT POLYNOMIALS =====
         2       2       2      2
F(1)=2*U3  + 2*U2  + 2*U1  + U0  - U0
F(2)=2*U3*U2 + 2*U2*U1 + 2*U1*U0 - U1
                                        2
F(3)=2*U3*U1 + 2*U2*U0 - U2 + U1
F(4)=2*U3 + 2*U2 + 2*U1 + U0 - 1
===== REDUCED SYSTEM =====
        8            7              6                  5
FF(0)=U0  - 332/77*U0  + 5492/693*U0  - 151772/18711*U0  + 25786/
             4                   3                2
      5103*U0  - 109940/56133*U0  + 956/2079*U0  - 3340/56133*U0 +
      181/56133
                               7                      6
FF(1)=U1 + 51602265/3424*U0  - 188085645/2996*U0  + 2600257707/
              5                      4                        3
      23968*U0  - 300968973/2996*U0  + 1284187269/23968*U0  -
                          2
      49010579/2996*U0  + 62902081/23968*U0 - 505167/2996
                               7                          6
FF(2)=U2 - 45940851/1712*U0  + 1220593131/11984*U0  - 1917921807/
              5                      4                       3
      11984*U0  + 201995835/1498*U0  - 786765859/11984*U0  +
                            2
```

```
      220481943/11984*U0   - 32737193/11984*U0 + 984531/5992
                                  7                      6
FF(3)=U3 + 40279437/3424*U0  - 468250551/11984*U0  + 1235585907/
              5                     4                       3
      23968*U0  - 103022697/2996*U0  + 289344449/23968*U0  -
                                 2
      24439627/11984*U0  + 2584289/23968*U0 + 22807/5992
%-------------------------------------------------------------------------;
%%%%%% KATSURA'S PROBLEM (4) %%%%%%
===== INPUT POLYNOMIALS =====
         2        2        2        2     2
F(1)=2*U4  + 2*U3  + 2*U2  + 2*U1  + U0  - U0
F(2)=2*U4*U3 + 2*U3*U2 + 2*U2*U1 + 2*U1*U0 - U1
                                              2
F(3)=2*U4*U2 + 2*U3*U1 + 2*U2*U0 - U2 + U1
F(4)=2*U4*U1 + 2*U3*U0 - U3 + 2*U2*U1
F(5)=2*U4 + 2*U3 + 2*U2 + 2*U1 + U0 - 1
===== REDUCED SYSTEM =====
        16                15                      14
FF(0)=U0   - 1948/231*U0   + 466850/14157*U0   -  .....
FF(1)=U1 + 913120042346651247135568357300894441/
                                        15
      68905785701353855630672*U0   -  .....
FF(2)=U2 - 317790599551725189190145389478624691/
                                         15
      137811571402707711261344*U0   +  .....
FF(3)=U3 - 289746854195772712626213908934181 83/
                                        15
      34452892850676927815336*U0   +  .....
FF(4)=U4 + 251065332760704024813517281592118541/
                                         15
      137811571402707711261344*U0   -  .....
%-------------------------------------------------------------------------;
%%%%%% KATSURA'S PROBLEM (5) %%%%%%
===== INPUT POLYNOMIALS =====
         2        2        2        2        2    2
F(1)=2*U5  + 2*U4  + 2*U3  + 2*U2  + 2*U1  + U0  - U0
F(2)=2*U5*U4 + 2*U4*U3 + 2*U3*U2 + 2*U2*U1 + 2*U1*U0 - U1
                                                      2
F(3)=2*U5*U3 + 2*U4*U2 + 2*U3*U1 + 2*U2*U0 - U2 + U1
F(4)=2*U5*U2 + 2*U4*U1 + 2*U3*U0 - U3 + 2*U2*U1
                                              2
F(5)=2*U5*U1 + 2*U4*U0 - U4 + 2*U3*U1 + U2
F(6)=2*U5 + 2*U4 + 2*U3 + 2*U2 + 2*U1 + U0 - 1
===== REDUCED SYSTEM =====
        32                 31                          30
FF(0)=U0   - 73352/4389*U0   + 2744754641/20369349*U0   -  .....
FF(1)=U1 -
715831228408419183465540687163370318923414432889690393804894089511583395
489495554311355176213824591407652333505032925565469766379414904218328822
726521791893184868923393035800434037522491735002202636755730567251412107
627114695977183535758569543756237238588063613870941774417324208226132303
96243020244071318225902713644140937500
                                                                      /

350566325837750222023658221340099528551232377119923000910014304646571507
915523398504471786506882853098109813792116300412242229813924606909770564
342038683439329991140635224109929769829286247437927108023959550038032808
859942989891398738953783867057481185362417973969068518756383653564849522
29026402789
                                             31
```

```
                                        *UO    +   .....
FF(2)=U2 +
700886663193952602670810062701644073106854212956940016738122574692802059
337800867730387096183411166931025714010260206300589509036628558996452036
903670634721664537903374269461978220170788558054356149767517010846569757
522054551532276121258019394038039219003495083453189015971080170799969175
2504830025356632194967058184025000000
                                                                 /
480227843613356468525559207315204833631825174136880823164403157050097956
048662189732153132201209387805629881907008630701701684676609050561329540
194573538957986289233746882342369547711351023887571380854739109641140834
054716424508765395827101187749974226523860238313792491447100895294314414
096252093
                                          31
                                      *UO    -    .....
FF(3)=U3 +
684297581391937492991725756940802671000833115526608474104482947025947223
177045373540067647257604557278211276223032790829504633412968986366821955
717659972751027921035367265778720121340071695860307525192551133664014006
139643992519139217246729346148917408778160025755797590228138794660794439
60093961553917111658523333817677398437
                                                                    /
701132651675500444047316442680199057102464754239846001820028609293143015
831046797008943573013765706196219627584232600824484459627849213819541128
684077366878659982281270448219859539658572494875854216047919100076065617
719885979782797477907567734114962370724835947938137037512767307129699044
58052805578
                                           31
                                        *UO    -   .....
FF(4)=U4 -
792769319326811158534195254365693638819854581518923246704620861693105400
519865082010599963290102954449701655346262743416658301248324523712284260
114323607337859765265091981811923857886980796070060117055046597125015567
427566512203394364401019338025887457214208665529530693761450832124747442
8183819436937746332165682087264843 75
                                                                 /
960455687226712937051118414630409667263650348273761646328806314100195912
097324379464306264402418775611259763814017261403403369353218101122659080
389147077915972578467493764684739095422702047775142761709478219282281668
109432849017530791654202375499948453047720476627584982894201790588628828
192504186
                                          31
                                      *UO    +   .....
FF(5)=U5 -
292794314112856760857779817546936503127950779794751120388916351062591475
252225751918982966536872007684689117146362350778344034623265304245282071 8
785972951853045103460114793643905846177379440111344303581222051103602316
395114331228283517598705928625666951329210176291059545515932873603628457
5346411057657948809775687405691118750 00
                                                                    /
350566325837750222023658221340099528551232377119923000910014304646571507
915523398504471786506882853098109813792116300412242229813924606909770564
342038683439329991140635224109929769829286247437927108023959550038032808
859942989891398738953783867057481185362417973969068518756383653564849522
29026402789
                                           31
                                        *UO    +   .....
%-------------------------------------------------------------------------;
```

□