数理解析研究所講究録
941 巻 1996 年 84-91

11.

Generalized Shape Lemma の Hensel 構成による計算

富士通情報研 野呂 正行

11.1 概要

0 次元イデアルの零点を数値的に求める場合, 辞書式順序によるグレブナ基底を求めることが有力な方法の一つであるが, 計算途中における係数膨張はもとより, 結果自体も非常に大きくなる場合がしばしばあり, 実用的には問題がある. これに比較して, 結果の有理式による表現である GSL によれば, 結果に現れる係数の大きさが, グレブナ基底に比較して大幅に小さくなることが多くの実例でわかっている. GSL の計算法としては, 対称式による方法が提案されているが, ここでは モジュラグレブナ基底を template として線形代数および Hensel 構成により GSL を計算する方法について述べる.

11.2 Generalized Shape Lemma (GSL) について

Shape Lemma は, 0 次元 radical イデアルの辞書式順序におけるグレブナ基底が, "大部分の" 変数変換により

$$\{x_1 - f_1(x_n), \cdots, x_{n-1} - f_{n-1}(x_n), f_n(x_n)\} \cdots (SL)$$

という形になることを示している. イデアルの基底が (SL) の形で求まれば, $f_n(x_n)$ の根を求めることにより, $x_1, \cdots, x_{n-1}$ は代入により求めることができるため, 辞書式順序でのグレブナ基底を求めることが, 0 次元イデアルで表される方程式系の解を求める一つの有力な方法と考えられてきた.

しかし, 有理数係数での辞書式順序グレブナ基底計算においては, (SL) の形の基底において, $f_1, \cdots, f_{n-1}$ の係数が, f_n の係数に比較して非常に大きくなる場合がしばしば生ずる. このことは, グレブナ基底を Buchberger アルゴリズムで計算する場合でも, change-of-ordering により計算する場合でも, 途中の計算量の増大を招き, グレブナ基底の計算を困難にしている.

Generalized Shape Lemma (GSL) は, radical イデアルの基底として, 変数変換後,

$$\{f_n'(x_n)x_1 - g_1(x_n), \cdots, f_n'(x_n)x_{n-1} - g_{n-1}(x_n), f_n(x_n)\} \cdots (GSL)$$

という形のものがとれることを主張する. ここで, f_n は (SL) に現れる f_n と一致する無平方な一変数多項式であり, $GCD(f_n, f_n') = 1$ が成り立つ. よって, (GSL) で表されるイデアルの零点は,

$$\{(\frac{g_1(\alpha)}{f_n'(\alpha)}, \cdots, \frac{g_{n-1}(\alpha)}{f_n'(\alpha)}, \alpha) | f_n(\alpha) = 0\}$$

と表すことができる. この時, 注目すべきことは, 多くの実例において, g_i の各係数が, f_n の係数と同程度の大きさに押えられることがわかっていることである. この場合, (GSL) による零点の表現は, (SL) による表現に比べてコンパクトであり, 計算時間も小さくできることが期待される.

11.3 対称式による GSL の計算

GSL は, [1] により与えられた. その効率的な計算法として, [3], [7] により, 対称式を用いた計算法が提案されている.

定義 1 K を体, $R = K[X]$ $(X = (x_1, \cdots, x_n))$ とし, $I \subset R$ を 0 次元イデアルとすると, $A = K[x_1, \cdots, x_n]/I$ は K 上有限次元のベクトル空間となる. この時, $f \in R$ に対し, $M_f : A \to A$ を, $M_f(g \bmod I) = fg \bmod I$ で定義する.

簡単のため, 変数変換により既に (GSL) の形の基底を持つとする. すると, (GSL) における f_n の係数は $M_{x_n^k}$ の trace により表現できる. 同様に, g_i の係数は, $M_{x_i x_n^k}$ の trace により表現できる. これらの trace から係数を求める際に, 積和から基本対称式への変換が用いられる.

この方法において計算効率をあげるためには, M_f および trace の計算を効率良く行わなければならない. PoSSo library においては, M_f および trace の計算に必要な, A の K-基底間の積の normal form の table の計算を重複なく効率良く行うことにより, M_f, trace の計算効率をあげている. 実際, この table が計算できれば, あとの計算は比較的容易である. しかし, A の K-次元 m が大きくなると, m^2 個の normal form を計算する必要があり, この部分がボトルネックとなる.

11.4 Hensel 構成による GSL の計算

我々は, [6] において, モジュラグレブナ基底を template として, 有理数上のグレブナ基底を未定係数法により直接求める change-of-ordering 法を提案した. この方法は, GSL の計算にも応用できる. $R = \mathbf{Q}[X]$ $(X = (x_1, \cdots, x_n))$, $\mathbf{Z}_p$ を $\mathbf{Z}$ の, 素数 p による局所化とし, $\phi_p : \mathbf{Z}_p \to GF(p)$ を標準的射影とする.

定義 2 $F \subset \mathbf{Z}[X]$, 素数 p に対し, $\phi_p(Id(F) \cap \mathbf{Z}[X]) = Id(\phi_p(F))$ が成り立つとき, p は F に関し *compatible* であるという.

定義 3 $F \subset \mathbf{Z}[X]$, 項順序 $<$ に対し, 素数 p が F の各元の $<$ に関する *head coefficient* を割り切らないとき, p は $(F, <)$ に関し *permissible* であるという.

補題 4 $G \subset \mathbf{Z}[X]$ が項順序 $<$ に関し $Id(G)$ のグレブナ基底で, p が $(G, <)$ に関し *permissible* ならば, p は G に関し *compatible*.

補題 5 p が $F \subset \mathbf{Z}[X]$ に関し *compatible* とする. $\overline{G}$ を $Id(\phi_p(F))$ の, 項順序 $<$ に関する *reduced* グレブナ基底とし, その各元を $<$ に関して順序の小さい順に並べたものを $\{\overline{g}_1, \cdots, \overline{g}_l\}$ とする. この時, $g_1, \cdots, g_m \in \mathbf{Z}_p[X]$ が存在して, $\phi_p(g_i) = \overline{g}_i (i = 1, \cdots, m)$ が成り立つならば, $\{g_1, \cdots, g_m\}$ は, F の *reduced* グレブナ基底の一部となる.

定義 6 *0* 次元イデアル $I \subset R$ に対し, $\mathbf{Q}[x_i]$ の単項イデアル $I \cap K[x_i]$ の生成元を I における x_i の最小多項式と呼ぶ.

定義 7 *0* 次元イデアル $I \subset R$ に対し, $S = \{x_1 - f_1(x_n), \cdots, x_{n-1} - f_{n-1}(x_n), f_n(x_n)\} \subset I$ が存在して, $I = Id(S)$ と書けるとき, S は x_n に関する I の *Shape Basis* であるという.

補題 8 *0* 次元イデアル $I \subset R$ に対し, x_n の最小多項式 m の次数が $dim_Q R/I$ と一致するならば, I の, x_n を最小順序とする辞書式順序グレブナ基底は *Shape Basis* となる.

以上の補題により, モジュラグレブナ基底から GSL あるいは通常のグレブナ基底を求めるアルゴリズムが次のように構成できる.

アルゴリズム 9

tolex_gsl(F, V)

Input : 多項式集合 F, 変数リスト V

Output : *GSL* 形式の F の零点, または $Id(F)$ の辞書式順序グレブナ基底

$G_0 \leftarrow$ ある項順序 $<$ に関する F のグレブナ基底

again:

$p \leftarrow (G_0, <)$ に関して *permissible* な未使用の素数

$\overline{G} \leftarrow Id(\phi_p(G_0))$ の変数順序 V に関する辞書式順序 *reduced* グレブナ基底
if $\overline{G}$ が x に関する *Shape Basis then*
$\overline{f} \leftarrow x$ の一変数多項式 $\in \overline{G}$
(1) *if* $\exists f \in Id(F)$ *s.t.* $\phi_p(f) = \overline{f}$ *then* $l \leftarrow \{f\}$
else goto again:
for each $v \in V \setminus \{x\}$ *do* {
$\overline{g} \leftarrow NF_{lex}(\overline{f}'v, \overline{G})$ *(辞書式順序での normal form)*
(2) *if* $\exists g \in \mathbf{Q}[x]$ *s.t.* $f'v - g \in Id(F)$ かつ $\phi_p(g) = \overline{g}$ *then* $l \leftarrow l \cup \{v - g/f'\}$
else goto again:
}
return l
else
(3) *if* $\exists G \subset Id(G)$ *s.t.* $\phi_p(G) = \overline{G}$ *then return* G
else goto again:

あらかじめ, F をある項順序によるグレブナ基底に変換しておくことにより, compatibility, permissibility に関する補題が適用可能になる. (1) において f が存在することが, 補題 5, 補題 8 により $\mathbf{Q}$ 上のグレブナ基底が Shape Basis となることを保証する. モジュラグレブナ基底が Shape Basis でない場合には, 通常のグレブナ基底を求めることになる.

(1) および (2) は, 実際には未定係数の導入および G_0 による normal form 計算により, 未定係数に対する線形方程式を解くことに帰着される. この線形方程式は, その構成法により, 法 p で一意解を持つ. このことから, この線形方程式が一般に過剰決定系であり, 有理数体上一意解を持つか, または解がないかのいずれかであることがわかる. さらに, 未定係数と同一の数の方程式を選んで, その方程式が法 p および有理数体上一意解を持つようにできることもわかる.

すなわち, 解くべき問題は次のようになる.

問題 10 M, B : $n \times n$, $n \times 1$ 整数行列; X : 未定係数を要素とする $n \times 1$ 行列とする時, $det(\phi_p(M)) \neq 0$ の元で $MX = B$ を $\mathbf{Q}$ 上で解け

この方程式を法 p での解から出発して Hensel 構成により解くことができる.

アルゴリズム 11

solve_linear_equation_by_hensel(M, B, p)
$R \leftarrow \phi_p(M)^{-1}$; $c \leftarrow B$; $x \leftarrow 0$; $q \leftarrow 1$; $count \leftarrow 0$
do {
$t \leftarrow \phi_p^{-1}(R\phi_p(c))$; $x \leftarrow x + qt$; $c \leftarrow (c - Mt)/p$;
$q \leftarrow qp$; $count \leftarrow count + 1$

```
    if count = Predetermined_Constant then {
      count ← 0; X ← inttorat(x, q)
      if X ≠ nil then return X
    }
}
```

inttorat() は整数を分数に変換する函数である. この方法においては, あらかじめ定められた段数ごとに, 整数-分数変換を行い, 解となっているかをチェックする方法を採用している. これにより, termination が, 解の大きさに依存してきまる, というメリットが生ずる. これに比べて, fraction-free 法では, 最終的に, M の行列式を計算してしまうことになるため, 解の大きさに関わらず, 一定の手間が必要となる. 我々の実験によれば, 最悪の場合, 即ち, 解の分母に行列式が現れる場合でも, Hensel 構成による方法が優位であった.

11.5 タイミングデータ

表は, 0 次元イデアルに対し, 全次数辞書式順序のグレブナ基底からスタートして, trace-lifting および Hensel 構成により辞書式順序グレブナ基底を求めた場合の CPU 時間, および, GSL 形式の零点を求めた場合の CPU 時間を示す. マシンは Sparc20/61 (60MHz supersparc), 単位は秒. カッコ内の数字は, 剰余環の, **Q**-次元, 即ち解の個数を示す.

C_6 : cyclic 6-roots [2] $\cup\{c-(c_0-c_1+2c_2-3c_3-4c_4+3c_5)\}$ (c について Shape Basis)

Mod : modified cyclic 5-roots [2] $\cup\{z-(a-b+2c+d-e))\}$ (z について Shape Basis)

K_n : Katsura-n [4]

GB-TL : trace-lifting と homogenization によるグレブナ基底計算

GB-Hensel : Hensel 構成 によるグレブナ基底計算

GSL-Hensel : Hensel 構成による GSL の計算

GB-PoSSo : 対称式による GSL の計算 (RealSolving in PoSSo library)

GSL-Hensel-para : Hensel 構成の並列化による仮想的な (通信時間を含まない, 理想的な) elapsed time

	C_6 (156)	Mod (64)	K_5 (32)	K_6 (64)	K_7 (128)
GB-TL	579	1584	78	7200	12 days
GB-Hensel	262	518	45	1570	1 day
GSL-Hensel	32	49	14	157	2308
GSL-PoSSo	349	43	9	106	2140
GSL-Hensel-para	17	21	6	57	773

0 次元イデアルの零点の計算

イデアルが Shape Basis を持つ場合, グレブナ基底, GSL に含まれる一変数多項式 $f_n(x_n)$ は同一であるため, 数値的に零点を求める際の手間は本質的には同一である. 計算時間の比較を見ればわかるように, これらの例においては, GSL の計算を行った方が明らかに少い手間で数値解を求める前処理を行うことができる. これは, これらの例においては, Shape Basis に含まれる, $f_n(x_n)$ 以外の元の係数が, $f_n(x_n)$ の係数に比較して極端に大きいためである. Hensel 構成による方法では, 結果の大きさにより計算時間が決まるため表のような結果となった.

次に, PoSSo library の対称式による方法と, 我々の Hensel 構成による方法を比較した場合, C_6 以外では PoSSo が有利に見えるものの, C_6 において, Hensel による方法が大幅に高速に GSL を計算できている. この問題において, PoSSo library における計算時間の内訳をみると, 大部分が table の計算に使われている. これは C_6 に対する剰余環の次元が高いためであり, そのような場合に Hensel 構成による方法が有利である可能性を示している.

11.6 分散並列計算

グレブナ基底を並列計算する試みは, 古くからさまざまの形で行われてきた. 特に Buchberger アルゴリズムは, 一見容易に並列化できそうな構造を持つため, その試みの対象となることが多かった. しかし, Buchberger アルゴリズムの安易な並列化は strategy を破壊し, 中間式膨張を招き, 結果として効率をあげることは難しい.

これに比較して, 我々の方法においては, 計算の大部分を占める線形方程式の求解が template ごとに完全に並列に行える. 即ち, 十分な数のプロセッサがあれば, 方程式の求解の部分は, 実時間において, もっとも時間のかかる方程式の計算時間程度に押えられることになる. これらの方法は, 全て **Risa/Asir** [5] にインプリメントされ, UNIX 上でネットワークを用いた分散計算により実行することができる. 以下は, **Asir** 上での分散計算の例である.

```
amulet: asir
This is Asir, Version 950831.
Copyright (C) FUJITSU LABORATORIES LIMITED.
3 March 1994. All rights reserved.
0
0sec
[141] M=["amulet","sofie","geisha","tohoho","parvarti"]$ /* マシン名 */
0sec
[142] S=map(tcpinit,M,"asir_server","noxlog");          /* サーバの起動 */
[0,1,2,3,4]
0.01sec
```

```
[143] P=map(spawn,S);                                    /* slave Asir の起動 */
[5,6,7,8,9]
0sec
[144] load("katsura")$
0sec
[149] K=katsura(5)$
0.03001sec
[150] V=[u5,u4,u3,u2,u1,u0]$
0sec
[151] G=gr(K,V,0)$                                       /* DRL G-basis の計算 */
3.341sec + gc : 2.29sec
[152] T=time()[3]$ tolex_gsl_d(G,V,0,V,P)$ print(time()[3]-T)$ /* 分散計算 */
0sec
[153]
5.371sec + gc : 1.1sec                                   /* master 上の CPU time */
[154] 12.4795                                            /* elapsed time */
0sec
[155] T=time()[3]$ tolex_gsl(G,V,0,V)$ print(time()[3]-T)$  /* 逐次計算 */
0sec
[156]
13.52sec + gc : 2.841sec                                   /* CPU time */
[157] 17.2832                                              /* elapsed time */
```

対称式による方法では, table の計算がいかに効率良く計算できるかが効率を上げる鍵となるが, monomial の ordering などを利用して, 既に計算した値を用いて次の値を計算しているとすれば, 並列化は困難と思われる. Hensel 構成による方法では, 全体の計算の大部分を占める, Hensel 構成の部分が容易に並列化できるため, 台数効果がより得易い. 現在の実現においては, 一つの線形方程式を, 一つのプロセスで, Hensel 構成により解いているが, これを Chinese Remainder Theorem と組合せることで, 更に並列化することができる. 現在 Fujitsu AP1000 上への **Risa/Asir** の移植を進めており, 数百プロセスを用いた並列計算により更に効率を上げることができると思われる.

11.7 今後の課題

ここで示した方法は, 与えられたイデアルが, ある変数に関して Shape Basis を持つことを要求す

るため, 重複度を持つイデアルに対してはそのまま適用できない. そのようなイデアルに対しても, ラディカルを考えればGSL形式の零点表現は可能であるが, それをいかにしてモジュラ計算により効率的に求めるかが今後の課題の一つである. また, 係数にパラメタを含む問題は, 有理函数体上の問題として扱うことができるが, このような問題は, 準素分解においても, また, 工学的な応用においても重要である. このような問題に対しても, Hensel構成による方法は, 多項式-有理式変換を用いることにより原理的には適用可能である. しかし, その効率化は, 実際にシステム上に実現する際に大きな問題となると考えられる.

参考文献

[1] Alonso, M. E., Becker, E., Roy, M. F., Wörmann, T., Zeros, Multiplicities and Idempotents for Zerodimensinal Systems. To appear in Proc. MEGA 94.

[2] Faugère, J.C., Gianni, P., Lazard, D., Mora, T., Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. J. Symb. Comp. 16/4(1993), 329-344.

[3] Gonzalez-Vega, L., Trujillo, G., Using Symmetric Functions to Describe the Solution Set of a Zero Dimensional Ideal. Proc. AAECC-11 (LNCS 948), 232-247.

[4] Katsura, S., Fukuda, W., Inawashiro, S., Fujiki, N. M. and Gebauer, R., Distribution of Effective Field in the Ising Spin Glass of the $\pm J$ Model at $T = 0$. Cell Biophysics Vol.11(1987), 309-319.

[5] Noro, M., Takeshima, T., Risa/Asir – A Computer Algebra System. Proc. ISSAC '92, 387-396. Binaries for various platforms are available from `endeavor.fujitsu.co.jp [164.71.1.131]:/pub/isis/asir`.

[6] Noro, M., Yokoyama, K., New methods for the change-of-ordering in Gröbner basis computation. ISIS Research Report, ISIS-RR-95-8E (1995).

[7] Rouillier, F., PoSSo - RealSolving (Zero-dimensional systems of polynomials). Proc. the POSSO Workshop on Software (1995).

[8] Traverso, C., The PoSSo test suits. Available from `gauss.dm.unipi.it [131.114.6.55]`.