# 限界性能を追求した計算機代数 vs 和算

木村欣司
KINJI KIMURA
京都大学大学院情報学研究科
GRADUATE SCHOOL OF INFORMATICS, KYOTO UNIVERSITY

## 1 はじめに

非線形連立代数方程式を解くことは簡単ではない. そのテクニックは多岐にわたる. ここでは, 現在われわれが知っているすべてのテクニックをあますところなく紹介することを試みる. これらの手法の成果として, 日本数学史の問題を解いたのであり問題を解くことが最終目標ではない. 手法の確立のためのテスト問題と位置づけている.

## 2 基本的な道具

### 2.1 数式処理

1. Characteristic set (Wu 先生の方法)
2. Gröbner basis (Buchberger 先生のアルゴリズム,F4 アルゴリズム)
3. 終結式, multipolynomial resultant
4. real root finding
5. 倍写像行列

### 2.2 数値計算

1. Krawczyk 法 (精度保証付き数値計算法)
2. ホモトピー法 (精度保証もおこなう)
3. Bézout の定理, Bernstein の定理
4. Gerschgorin の定理

それぞれの詳細は, 後に述べる.

# 3 数式処理

終結式は, 行列式を用いて計算することもできる. よって, 行列式から始める.

## 3.1 行列式

### 3.1.1 Fraction-free Gaussian elimination

$a_{i,j}$ を多変数多項式とする.

$$\begin{pmatrix} . & . & . & . & . & . & . \\ 0 & a_{k-1,k-1} & . & . & . & . & . \\ 0 & 0 & a_{k,k} & . & a_{k,j} & . & . \\ 0 & 0 & . & . & . & . & . \\ 0 & 0 & . & . & . & . & . \\ 0 & 0 & a_{i,k} & . & a_{i,j} & . & . \\ 0 & 0 & . & . & . & . & . \\ 0 & 0 & . & . & . & . & . \end{pmatrix}$$

次の漸化式に従って計算する.

$$a_{i,j}^{k+1} \leftarrow \frac{a_{i,j}^k a_{k,k}^k - a_{i,k}^k a_{k,j}^k}{a_{k-1,k-1}^{k-1}}$$

肩の添字は, 消去法が第何行目まで進んだかをあらわす. $a_{i,j}^k a_{k,k}^k - a_{i,k}^k a_{k,j}^k$ は必ず $a_{k-1,k-1}^{k-1}$ で割り切れる. [5]

$$\begin{pmatrix} a & b & c \\ f & g & h \\ k & l & m \end{pmatrix} \rightarrow \begin{pmatrix} a & b & c \\ 0 & \begin{vmatrix} a & b \\ f & g \end{vmatrix} & \begin{vmatrix} a & c \\ f & h \end{vmatrix} \\ 0 & \begin{vmatrix} a & b \\ k & l \end{vmatrix} & \begin{vmatrix} a & c \\ k & m \end{vmatrix} \end{pmatrix} \rightarrow \begin{pmatrix} a & b & c \\ 0 & \begin{vmatrix} a & b \\ f & g \end{vmatrix} & \begin{vmatrix} a & c \\ f & h \end{vmatrix} \\ 0 & 0 & X = \det \end{pmatrix}$$

$$X = \frac{\begin{vmatrix} a & c \\ k & m \end{vmatrix} \begin{vmatrix} a & b \\ f & g \end{vmatrix} - \begin{vmatrix} a & b \\ k & l \end{vmatrix} \begin{vmatrix} a & c \\ f & h \end{vmatrix}}{a} = \begin{vmatrix} a & b & c \\ f & g & h \\ k & l & m \end{vmatrix}$$

つぎのように書き直してみる.

$$\begin{vmatrix} a & c \\ k & m \end{vmatrix} \begin{vmatrix} a & b \\ f & g \end{vmatrix} - \begin{vmatrix} a & b \\ k & l \end{vmatrix} \begin{vmatrix} a & c \\ f & h \end{vmatrix} = \begin{vmatrix} a & b & c \\ f & g & h \\ k & l & m \end{vmatrix} a$$

これを, **Jacobi の恒等式**という. 行列式は $a_{n,n}^{(n)}$ に現れる.

### 3.1.2 小行列式展開

[15] を参照されたい.

### 3.1.3 ラプラス展開

$4 \times 4$ の行列まで, この方法では 1 箇所も計算結果の使い回しができないことがわかる. $5 \times 5$ の行列 $A = (a_{i,j})$ の行列式を例にどの程度計算結果の使い回しができるかをみる.

$$
\begin{aligned}
L_1 &= a_{4,4}a_{5,5} - a_{4,5}a_{5,4}, L_2 = a_{4,3}a_{5,5} - a_{4,5}a_{5,3}, L_3 = a_{4,3}a_{5,4} - a_{4,4}a_{5,3}, \\
L_4 &= a_{4,2}a_{5,5} - a_{4,5}a_{5,2}, L_5 = a_{4,2}a_{5,4} - a_{4,4}a_{5,2}, L_6 = a_{4,2}a_{5,3} - a_{4,3}a_{5,2}, \\
L_7 &= a_{4,1}a_{5,2} - a_{4,2}a_{5,1}, L_8 = a_{4,1}a_{5,5} - a_{4,5}a_{5,1}, L_9 = a_{4,1}a_{5,4} - a_{4,4}a_{5,1}, \\
L_{10} &= a_{4,1}a_{5,3} - a_{4,3}a_{5,1} \\
\det(A) &= +(a_{1,1}a_{2,2} - a_{1,2}a_{2,1})(a_{3,3}L_1 - a_{3,4}L_2 + a_{3,5}L_3) \\
&\quad -(a_{1,1}a_{2,3} - a_{1,3}a_{2,1})(a_{3,2}L_1 - a_{3,4}L_4 + a_{3,5}L_5) \\
&\quad +(a_{1,1}a_{2,4} - a_{1,4}a_{2,1})(a_{3,2}L_2 - a_{3,3}L_4 + a_{3,5}L_6) \\
&\quad -(a_{1,1}a_{2,5} - a_{1,5}a_{2,1})(a_{3,2}L_3 - a_{3,3}L_5 + a_{3,4}L_6) \\
&\quad +(a_{1,2}a_{2,3} - a_{1,3}a_{2,2})(a_{3,1}L_1 - a_{3,4}L_8 + a_{3,5}L_9) \\
&\quad -(a_{1,2}a_{2,4} - a_{1,4}a_{2,2})(a_{3,1}L_2 - a_{3,3}L_8 + a_{3,5}L_{10}) \\
&\quad +(a_{1,2}a_{2,5} - a_{1,5}a_{2,2})(a_{3,1}L_3 - a_{3,3}L_9 + a_{3,4}L_{10}) \\
&\quad +(a_{1,3}a_{2,4} - a_{1,4}a_{2,3})(a_{3,1}L_4 - a_{3,2}L_8 + a_{3,5}L_7) \\
&\quad -(a_{1,3}a_{2,5} - a_{1,5}a_{2,3})(a_{3,1}L_5 - a_{3,2}L_9 + a_{3,4}L_7) \\
&\quad +(a_{1,4}a_{2,5} - a_{1,5}a_{2,4})(a_{3,1}L_6 - a_{3,2}L_{10} + a_{3,3}L_7)
\end{aligned}
$$

一般に, $N \times N (N > 4)$ の行列のラプラス展開においては ($N/2$ の切り上げの数 $-1$) $\times$ ($N/2$ の切り上げの数 $-1$) 以下のサイズの行列式の計算結果の使い回しができる. 計算結果の使い回しをするためには, 大量のメモリが必要である. もちろん, $N$ が大きくなれば組み合わせ爆発もおきる.

### 3.1.4 補間法

この方法を紹介する前に, 基本定理の確認をする.

**整数を要素とする行列の行列式の評価公式**

$n \times n$ の行列 $A$ に対して $\det(A)$ の評価は以下のようにおこなう.[16]

$$
A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix},
$$

より

$$u_1 = (a_{1,1}, \ldots, a_{1,n}), \ldots, u_n = (a_{n,1}, \ldots, a_{n,n}),$$
$$v_1 = (a_{1,1}, \ldots, a_{n,1}), \ldots, v_n = (a_{1,n}, \ldots, a_{n,n}),$$

を定義すると, Hadamard の公式より

$$\det(A) \text{ の絶対値}$$
$$\leq \min(\|u_1\|_2 \|u_2\|_2 \ldots \|u_{n-1}\|_2 \|u_n\|_2, \|v_1\|_2 \|v_2\|_2 \ldots \|v_{n-1}\|_2 \|v_n\|_2) \equiv H,$$

は整数を要素とする行列の行列式を計算するときに重要な役割を果たすことはあらためて述べるまでもない.

**多変数多項式を要素とする行列の行列式の評価公式**

多変数多項式の 1 ノルムは, 係数の絶対値の総和と定義する.

$$\begin{pmatrix} ||a_{1,1}||_1 & \cdots & ||a_{1,n}||_1 \\ \vdots & & \vdots \\ ||a_{n,1}||_1 & \cdots & ||a_{n,n}||_1 \end{pmatrix}$$

として Hadamard の公式を適用する. そのときの値を $H_1$ とすると

$$\text{多変数多項式を要素とする行列の行列式の係数の絶対値最大} \leq H_1$$

が成立する. 詳しくは, [6] を参照されたい. これらの基本知識を踏まえた上で, 補間により行列式を計算する. 詳細は, [9] を参照されたい.

## 3.2 終結式

### 3.2.1 終結式の行列式による表現

終結式を行列式で表現する場合に, 現在知られている 3 つの代表的な方法を紹介する. これ以外にも, 佐々木らの方法 [14] もある.

1. Sylvester の方法
2. 関孝和の方法
3. 擬剰余を用いて構成する方法

はじめの 2 については, あまりにも有名でありここでは特に触れることはしない. [5] 最後の 1 つは, [3] に解説がある.

最後の 1 つについて, ここでは具体例をあげてその特徴をみることとする.

$$f(x) = b_0 + b_1 x + b_2 x^2,$$
$$g(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3.$$

$f(x)$ と $g(x)$ の終結式を計算したかったとする. $g(x)$ を法として, $f(x)$ についての擬剰余を計算して行列式を構成すると

$$\mathrm{res}_x(f(x),g(x)) = a_3^{-1} \times \begin{vmatrix} b_0 & b_1 & b_2 \\ -a_0b_2 & a_3b_0 - a_1b_2 & a_3b_1 - a_2b_2 \\ a_2a_0b_2 - a_3a_0b_1 & (a_2a_1 - a_3a_0)b_2 - a_3a_1b_1 & (a_2^2 - a_3a_1)b_2 + a_3^2b_0 - a_3a_2b_1 \end{vmatrix}$$

関孝和の方法をもちいても, $3\times3$ の行列式が構成され, かつ $a_3^{-1}$ という余剰な因子を取り除くための割り算が存在しないのでこの定義を用いる意味はまったくない. $f(x)$ を法として, $g(x)$ についての擬剰余を計算して行列式を構成すると

$$\mathrm{res}_x(f(x),g(x)) = a_3^{-1} \times \begin{vmatrix} a_3b_0b_1 + a_0b_2^2 - a_2b_0b_2 & a_3b_1^2 - a_2b_2b_1 + a_1b_2^2 - a_3b_0b_2 \\ (a_2b_2 - a_3b_1)b_0b_1 + (a_3b_0 - a_1b_2)b_0b_2 & (a_2b_2 - a_3b_1)b_1^2 + 2a_3b_0b_2b_1 + (a_0b_2 - a_2b_0 - a_1b_1)b_2^2 \end{vmatrix}$$

となり, $2\times2$ の行列式が構成される. この具体例では顕著ではないが, 500 次の式と 3 次の式の終結式を計算したいときなど, この定義がきわめて有効に働くときがある. $3\times3$ の行列式の計算問題に帰着するからである. $3\times3$ の行列式は, サラスの方法ではなくラプラス展開を用いて計算すればよい.

### 3.2.2 終結式の計算

行列式による定義をもちいて, 行列式として計算をおこなうこともできが, 終結式の特徴を生かして計算する方法に以下のものがある.

**Colins の subresultant PRS(polynomial remainder sequence)**

$\deg(A(x)) \geq \deg(B(x))$ の $A(x), B(x)$ という環 $R$ の元を係数とする 1 変数多項式が与えられたとき,

1. $R_0(x) = A(x), R_1(x) = B(x)$
2. $\alpha_i R_{i-1}(x) = Q_i(x)R_i(x) + \beta_i R_{i+1}(x) \quad \mathrm{with}\alpha_i, \beta_i \in R$
3. $\mathrm{prem}(R_{k-1}, R_k)$=0

という条件をみたす多項式列 $R_1(x), ...., R_k(x)$ を PRS という. $\alpha_i, \beta_i$ には不定性がある. $r_i = \text{leading coeff.}(R_i(x)), \delta_i = \deg(R_{i-1}(x)) - \deg(R_i(x))$ として

$$\alpha_i = r_i^{\delta_i+1}, \beta_1 = (-1)^{\delta_1+1}, \beta_i = -r_{i-1}\psi_i^{\delta_i} \quad \text{for} \quad 2 \leq i \leq k,$$
$$\psi_1 = -1, \psi_i = (-r_{i-1})^{\delta_{i-1}}\psi_{i-1}^{1-\delta_{i-1}} \quad \text{for} \quad 2 \leq i \leq k$$

と選ぶと終結式が計算できる.[5]

補間法も有効に働く, 補間を用いるための準備としてつぎの結果を紹介する.

**有限体 GF(p) を係数とする多項式の終結式の計算法**

$R_1 = f, R_2 = g$ とする. ユークリッドの互除法により $R_3, ...R_k$ を計算する, $n_i = \deg(R_i)$ として

$$\mathrm{res}(R_1, R_2) = \mathrm{lcoeff}(R_k)^{n_{k-1}} \prod_{i=1}^{k-2} (-1)^{n_i n_{i+1}} \mathrm{lcoeff}(R_{i+1})^{n_i - n_{i+2}}.$$

を計算すると, 有限体 GF(p) を係数とする多項式の終結式が計算できる. 終結式の補間による計算のためには各変数への数値の代入が必須となるが, そのときに主係数が消えた場合の取り扱いには十分に注意しなければならない. 具体的には, 上記の計算の後その結果への補正が必要となる. 詳しくは, [5] を参照されたい.

#### 3.2.3 終結式計算の計算量削減のための公式

$$\mathrm{res}_x(A_1 \times A_2, B) = \mathrm{res}_x(A_1, B) \times \mathrm{res}_x(A_2, B), \tag{1}$$

$$\mathrm{res}_x(A, B_1 \times B_2) = \mathrm{res}_x(A, B_1) \times \mathrm{res}_x(A, B_2). \tag{2}$$

$$\mathrm{res}_x(m \text{ の倍数の次数のみ係数をもつ多項式 } A, m \text{ の倍数の次数のみ係数をもつ多項式 } B)$$
$$= (\mathrm{res}_x(A(x^m \to x), B(x^m \to x))^m \tag{3}$$

これらの定理を, 上手に使って終結式の計算を高速化する. また, 与えられた終結式の問題を行列式で表現してみると, 計算を効率化するための指針を得られることもある.

### 3.3 multipolynomial resultant

multipolynomial resultant は, 新しいアイディアを導入しない限り Sylvester の定義を多変数拡張することとなり行列式のサイズが大きくなる. そこで, 多少サイズを小さくするような試みがありその代表が Macaulay によるものである. しかし, それでも十分に大きいためさらなる改良が図られた. それが Gelfand-Kapranov-Zelevinsky によるものである. [4] これらは, その出発点から明らかなように $m$ 次の式と $n$ 次の式から 1 変数を消去する場合, その行列式は Sylvester によるものそのものが構成される. 一方で, 関孝和先生の定義の多変数への拡張も図られている. まず, はじめに触れておかなければならないことは $m$ 次の式と $n$ 次の式において $m = n$ の場合に関孝和先生により構成される $m \times m$ の行列式には, Cayley による別の構成法が存在することである.[16, pp.355] $m \neq n$ の場合には, Cayley による構成法では余分な因子が入り込むことになるが, それを気にしなければこの Cayley の見方は優れており多変数への拡張という点では非常に都合がよい. そのことに注目して, Dixon はこの Cayley による構成法の多変数への自然な拡張を提案している. [18] 各変数の次数が高い場合には, Gelfand-Kapranov-Zelevinsky によるものは Dixon

による行列式よりもサイズが小さい, 逆に, 変数の数が多いものについては, サイズが小さいくなるという傾向があるためDixonによる行列式を利用したほうがよい場合が多い.

### 3.4 グレブナー基底

グレブナー基底の定義, 計算法やその高速化につながるAutomatic wight generator[10]についてまで記述するほどの紙面に余裕がない. ここでは, 日本数学史の問題を解く上で最も重要な定理 [12, pp.240] をひとつ紹介するのみとする.

**定理**

$G_0$ は, 項順序 $\prec_0$ に関する簡略グレブナー基底とし, $p$ を $G_0 \subset \mathbf{Q}_{<p>}[X]$ となる素数とする. $f \in \subset \mathbf{Q}_{<p>}[X]$ に対し, $m_p(t) \in \mathrm{GF}(p)[t]$ を $\phi_p(f)$ の $< \phi_p(G_0) >$ に関する最小多項式とする. このとき, モニックな多項式 $m(t) \in \mathbf{Q}_{<p>}[t]$ が存在して,

$$\deg(m(t)) = \deg(m_p(t)) \quad \text{かつ} \quad m(f) \in < G_0 >$$

ならば, $m(t)$ は $f$ の $< G_0 >$ に関する最小多項式となる. 定理は, さらに拡張される. $G_0$ にパラメータ含まれている場合には, 頭係数が消えないようなパラメータへの数値の代入は許される.

この定理は, 終結式とグレブナー基底とを結ぶものである. すなわち, 終結式を計算することで $m(f) \in < G_0 >$ なる $m(f)$ を作ることは容易である. 一方, グレブナー基底が計算できれば $\deg(m_p(t))$ の計算も容易にできる. 一般に, 終結式の計算は必要条件変形であり非線形連立代数方程式の解法に応用すると偽因子 (偽の根) が混入する場合ある. 運よく偽因子が混入しない場合には, この定理が有効に働きその計算が必要十分条件変形であったことが証明されるわけである. では, 不運にも偽因子が混入した場合すなわち $\deg(m(t)) \neq \deg(m_p(t))$ の場合には, どのように対処したらよいであろうか?

具体例を用いて, その対処法を述べる. 日本数学史の問題の中に次のような問題がある. $a, b, c, d, e, f, g, h$ は, パラメータ

$$uv + wx + yz + s - a = 0, st + wx + yz + u - b = 0, st + uv + yz + w - c = 0$$
$$st + uv + wx + y - d = 0, u + w + y + t - e = 0, s + w + y + v - f = 0$$
$$s + u + y + x - g = 0, s + u + w + z - h = 0$$

の式から $y, w, u, z, x, v, t$ を消して $s$ だけの式にせよ!

方程式系からDixon multipolynomial resultant行列をつくり, その行列を計算すると $s$ についての15次の多項式 $s^3$ ($s$ の12次の多項式) ができる. しかし, $\deg(m_p(s))$ を計算すると12である. この場合には,

$$uv + wx + yz + s - a = 0, st + wx + yz + u - b = 0, st + uv + yz + w - c = 0$$
$$st + uv + wx + y - d = 0, u + w + y + t - e = 0, s + w + y + v - f = 0$$
$$s + u + y + x - g = 0, s + u + w + z - h = 0, \underline{s = 0}$$

についてのグレブナー基底を計算し, $\{1\}$ を確認すると上記の定理に持ち込むことができる.

また, $m(f)$ はグレブナー基底の計算後, 連立一次方程式を用いても構成できる. 詳しくは, [12, pp.240-248] を参照されたい.

### 3.5 real root finding

上記の定理をもちいると最小多項式が計算できる. 最小多項式の実根の探索は, 以下のようにおこなう.

まず, 補助的な役割を果たす定理について述べる. 実係数の 1 変数代数方程式 $f(x) = a_k x^k + \cdots + ax + a_0 = 0, a_k > 0, k \geq 1$ の正の実根のみの存在限界は, $a_k$ を除いた負の係数を $a_\alpha, a_\beta, a_\gamma, \cdots$ とすれば $2\max\left(|a_\alpha/a_k|^{1/\alpha}, |a_\beta/a_k|^{1/\beta}, |a_\gamma/a_k|^{1/\gamma}, \cdots\right)$ となる. この公式を Johson の限界とよぶことにする. [17] $x \to 1/x$ により正の実根の最小を下から抑えられる.

重根をもたない実係数の 1 変数代数方程式 $f(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0$ の実根を計算機代数によって分離することを考える. 係数列 $a_0, a_1, \cdots, a_k$ の符号の変りの数 (0 になるものは飛ばして数える) を $W$ とすれば, $f(x)$ の正根 (0 を入れない) の数は $W$ またはそれよりも偶数だけ少ない. これを Descarte の符号律という. よって, $W$ が 0 ならば正の実根が非存在が保証される. また, $W$ が 1 ならば正の実根が 1 根あることが保証される. もし $W$ が 2 以上の場合には, 領域を分割して再度 $W$ を計算して調べればよい. $x \to 1/(x+1)$ と変数変換しさらに分母を消去し多項式化することにより, 新しい $x$ の $(0, \infty)$ の実根の数を調べることは, 元の世界では $(0, 1)$ の実根の数を調べることに対応する. $x \to x + 1$ と変数変換することにより, 新しい $x$ の $(0, \infty)$ を実根の数を調べることは, 元の世界では $(1, \infty)$ を調べることに対応する. $x = 1$ が根の場合には, $x \to x + 1$ の後, 定数項が消えていないかを check することで対応する. 以上の操作を繰り返す. $x \to x + 1$ であるから根が巨大な整数の場合アルゴリズムは極端に遅くなるようにに思えるが, Johson の限界により原点移動がおこなえるためそのような問題は起きない. この方法を, Uspensky の方法と呼ぶ. 詳しくは, [17] を参照されたい. なお, 代数方程式系の実根の数を数えるときは最小多項式の変数は**分離化元**にとらなければならない.

### 3.6 倍写像行列

以下では, 具体例として

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 2, \quad f_2(x_1, x_2) = 3x_1x_2 - 2$$

を考える. 項順序は $x_2 \prec x_1$ として全次数逆辞書式順序を用いてグレブナー基底を計算する.

$$G_1 = \{3\underline{x_1x_2} - 2, \underline{x_1^2} + x_2^2 - 2, -3\underline{x_2^3} - 2x_1 + 6x_2\}$$

項順序は $x_2 \prec x_1$ として辞書式順序を用いてグレブナー基底を計算する.

$$G_2 = \{9\underline{x_2^4} - 18x_2^2 + 4, -2\underline{x_1} - 3x_2^3 + 6x_2\}$$

辞書式順序では, $x_2$ のみの式が現れるために数値計算において都合が良いように思われるが1変数にしたことにより係数の桁が全次数逆辞書式順序に比べて膨張する傾向がある.

$G_1$ または $G_2$ の下線の部分は頭項を現している. その頭項を利用して次のような図を書く.

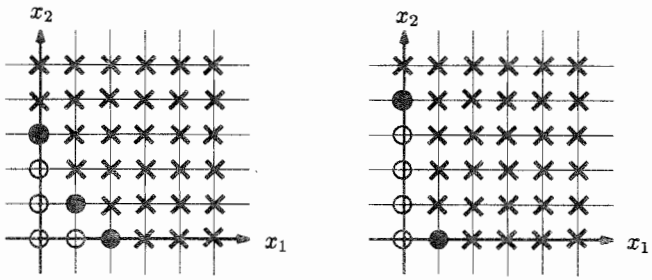



図 1: $G_1$ と $G_2$ の頭項による割り切れる項と割ることのできない単項の識別図

塗りつぶされた円が $G_1$ と $G_2$ のそれぞれの頭項を表している. 中抜きの円は, 頭項では割れない項を表し, ×は頭項で割ることのできる項を表している. 正規化の対象となる多項式は, その多項式の頭項が×のものだけでなくその多項式の頭項がグレブナー基底の頭項に一致するものも含まれる. 例えば, 単項式 $x_1^2$ は $G_1$ では, $G_1$ では中抜きの円に属しているためこれ以上の正規化は望めない. しかし, $G_2$ においては頭項であり正規化によって $-x_2^2+2$ となる. 天下りであるが, グレブナー基底 $G_1$ に対してその頭項で割れない項を利用して次のベクトル $u$ を用意する.

$$\begin{aligned} u &= (1, x_1, x_2, x_2^2)^{\mathrm{T}} \\ x_1 u &= (x_1, x_1^2, x_1x_2, x_1x_2^2)^{\mathrm{T}} \end{aligned}$$

となるが, グレブナー基底を法とする正規化が実行できるものも存在し

$$\begin{aligned} x_1 u &\equiv (x_1, -x_2^2 + 2, 2/3, 2/3x_2)^{\mathrm{T}} \mod G_1 \\ &\equiv \begin{pmatrix} 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & -1 \\ 2/3 & 0 & 0 & 0 \\ 0 & 0 & 2/3 & 0 \end{pmatrix} u \mod G_1 \end{aligned}$$

$x_1 = \lambda$ とすると, 非線形連立代数方程式はグレブナー基底を通して標準固有値問題に変換される. [12, pp.131-146]

# 4 数値計算

ホモトピー法とその精度保証については, [13, pp.146-158] を参照されたい. ここでは, Krawczyk 法とそれに関連して Bézout の定理と Bernstein の定理を述べることとする.

## 4.1 Krawczyk 法と有界領域の全根探索

### 4.1.1 Krawczyk 法 [13, pp.141-146]

非線形連立方程式 $f(x) = 0$ を考える. この方法は, 代数方程式以外にも適用できることを注意する. $U \subset \mathrm{R}^n, f : U \to \mathrm{R}^n$ を $C^1$ 級とする. $U$ の内部におけるある区間 $T$ とし $c = \mathrm{Mid}(T)$ を区間 $T$ の中心として, 区間行列 $M$ を

$$M = E - L^{-1}F'(T),$$

を考える.

ここで, $E$ は単位行列であり $F'(T)$ は写像 $f$ のヤコビ行列の各要素を定義域 $T$ として区間包囲したものである. $L^{-1}$ は, 構成法としては通常 $c = \mathrm{Mid}(T)$ においてヤコビ行列の各要素を近似的に評価しその近似逆行列を計算して得るが条件 (4)(5) を満たせば数学的にはどのような構成法を用いてもよい. 区間写像 $K$ を

$$K(T) = c - L^{-1}f(c) + M(T - c) \quad (c = \mathrm{Mid}(T))$$

で定義する. ただし, $f(c)$ は点に対してその値域を無誤差で計算することは一般にはできないため $x_j = [c_i, c_i]$ という区間における $f(x)$ の評価とする. ここで,

$$||M||_\infty < 1 \tag{4}$$

$$K(T) \subset T, \tag{5}$$

がみたされるならば, $K(T)$ に方程式 $f(x) = 0$ の解 $x^*$ がただ一つ存在する. **ただし, 重根の場合にはこの存在判定は絶対に成立しない.** ここで, $\subset$ は $K(T)$ が $T$ の境界上に辺を持たない $T$ の真部分集合と定義する. $K(T)$ が $T$ の境界上に辺を持つことはここでは認めない.

### 4.1.2 有界領域の全根探索

$f(x) = 0$ と有界超直方体領域 $S$ が与えられたとき,

1. $G = \{\}, H = \{S\}$ とする
2. $H = \{\}$ ならば end
3. $T \leftarrow \mathrm{head}(H)$
4. $f(T) \not\ni 0$ ならば 2 へ

5. $T$ に根が唯一存在するかを Krawczyk 法で判定. 条件をみたすならば $G$ に $T$ を追加. 2へ

6. $T$ のもっとも長さの長い辺を 1/2 する. それを $T_1, T_2$ とすると $T_1, T_2$ を $H$ の最後に追加. 2へ

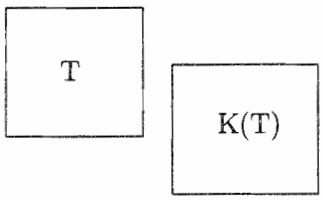
$G$ は根を含む超直方体のリストとなる.

もし, $K(T)$ が $T$ の境界上に辺を持つ場合にも根の存在を認めてしまうと分割領域の境界上に根が存在する場合には同じ根を異なった領域で2回数えるあるいは根として判定することとなる. そのような判定条件の場合, 上記のアルゴリズムの後再度連結した区間について再計算をしなければならないという問題が起きる. 逆に, 根の存在の際に $K(T)$ が $T$ の境界上に辺を持たないとすると偶然にも分割領域の境界上に根が存在した場合アルゴリズムはその根について存在判定をすることができずかつその根を含む領域では非存在判定もできない. よって, アルゴリズムは停止しない. そのような場合には, 初期領域 $S$ をわずかに大きくすることで回避できる. プログラムの簡便性なども考慮すると $K(T)$ が $T$ の境界上に辺を持たないとするのが妥当である.

#### 4.1.3 有界領域の全根探索の speed up

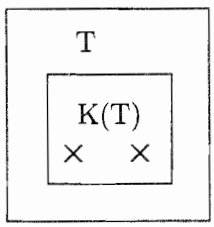
有界領域における全実根の探索の改良として, 以下3つをあげることができる.

1. 非存在判定の強化



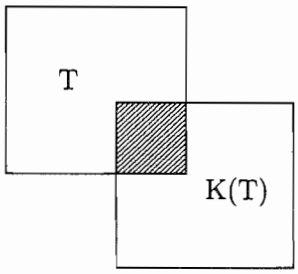

このような場合には, $T$ に根がないことが証明できる.

2. 根が, 唯一でない場合の取り扱い




$K(T) \subset T$ であるが, K(T) になかに2つ以上の根があるその場合には, $T$ ではなく $K(T)$ を二分割する.

3. 共通領域を利用した探索範囲の縮小




この場合には, 斜線の部分のみを調べればよいことになる,

#### 4.1.4 無限領域探索問題の有界領域探索問題への帰着

ホモトピー法により Bernstein の定理 [2] から導かれる個数分の複素根を求めた後, 得られた根を Krawczyk 法により実と複素に分離すれば無限領域における実根をすべてもとめることも可能であるが, ここでは別のアプローチを模索することにする.

上記のような有界領域探索問題の解法が与えられると, 次のような方法で, 無限領域探索問題を扱うことができるように誤解を招くのであらかじめ注意をする.

$$\begin{cases} x_1^2 + 2x_1 + x_2^2 - 2x_2 - 7 &= 0 \\ x_1^2 - 6x_1 + x_2^2 - 4x_2 + 9 &= 0 \end{cases}$$

が与えられたとき, はじめに Krawczyk 法を用いて $-1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1$ のすべての根を調べる. 次に, $x_1 \to 1/y_1$ とする.

$$\begin{cases} y_1^2 x_2^2 - 2y_1^2 x_2 - 7y_1^2 + 2y_1 + 1 &= 0 \\ y_1^2 x_2^2 - 4y_1^2 x_2 + 9y_1^2 - 6y_1 + 1 &= 0 \end{cases}$$

$-1 \leq y_1 \leq 1, -1 \leq x_2 \leq 1$ のすべての根を調べる. さらに, $x_2 \to 1/y_2$ とする.

$$\begin{cases} x_1^2 y_2^2 + 2x_1 y_2^2 - 7y_2^2 - 2y_2 + 1 &= 0 \\ x_1^2 y_2^2 - 6x_1 y_2^2 + 9y_2^2 - 4y_2 + 1 &= 0 \end{cases}$$

$-1 \leq x_1 \leq 1, -1 \leq y_2 \leq 1$ のすべての根を調べる.
最後に,

$$\begin{cases} -7y_1^2 y_2^2 + 2y_1 y_2^2 + y_2^2 - 2y_1^2 y_2 + y_1^2 &= 0 \\ 9y_1^2 y_2^2 - 6y_1 y_2^2 + y_2^2 - 4y_1^2 y_2 + y_1^2 &= 0 \end{cases} \tag{6}$$

の $-1 \leq y_1 \leq 1, -1 \leq y_2 \leq 1$ のすべての根を調べる. 以上より, 有界な領域のすべての根を調べることができるように思える. しかし, それは間違いである. 最後の方程式に注目する. 式 (6) は, 原点 $(y_1, y_2) = (0, 0)$ に重根をもつ. もちろん, それは偽の解である. 偽の解を得ることがこの手法の問題なのではなく, $(y_1, y_2) = (0, 0)$ が**重根**であることがこの手法の問題なのである. Krawczyk 法は, **重根**には非対応である. よって, アルゴリズムが停

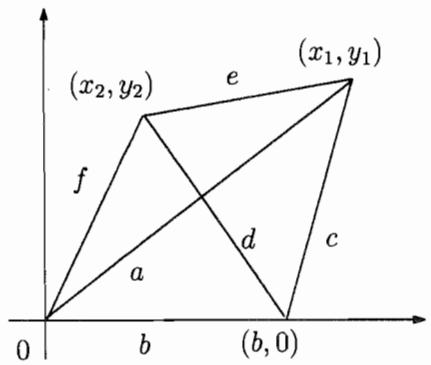
止しないためこのような手法を用いることはできない. 計算手順から明らかなように無限遠方の根がこの手法の邪魔をする. 非線形連立代数方程式には, 複素根まで含めて根の個数の上界を計算できる定理がある. Bézout の定理と Bernstein の定理 [2] である. Bézout の定理では, 無限遠方の解まで含めて根の個数を数える. 一方で, Bernstein の定理は数えない. 少なくとも, 現時点でいえることは**Bézout の定理と Bernstein の定理が異なった数を出すときこの手法は使えない**という否定的な結果のみである. より確実な手法を次に述べる.

#### 4.1.5 Gerschgorin の定理 [1]

グレブナー基底の計算の後, 倍写像行列を変数の種類分構成し以下の定理を用いることで根の存在範囲を構成できる. 与えられた $n$ 次正方行列 $A = (a_{i,j})$ の固有値の位置を事前に推定する方法がある. いま, $r_i = \sum_{j=1(j\neq i)}^{n} |a_{i,j}|, c_j = \sum_{i=1(i\neq j)}^{n} |a_{i,j}|$ とおく. 1) $A$ の固有値は複素平面の $a_{i,i}$ を中心とする半径 $r_i$ の円 $R_i = \{z : |z - a_{i,i}| \leq r_i\}$ のユニオン $\cup R_i$ にある. 2) $A$ の固有値は複素平面の $a_{j,j}$ を中心とする半径 $c_i$ の円 $C_j = \{z : |z - a_{j,j}| \leq c_j\}$ のユニオン $\cup C_j$ にある. この定理を用いると無限領域探索問題は, 有限領域探索問題に常に帰着できる. これ以外にも方法がある. 終結式を用いて最小多項式を因子として含む式を高速に計算できた場合, Johson の限界を用いれば実根の限界が計算できる.

## 5 日本数学史の問題

### 5.1 6 角形の辺の長さの関係の導出

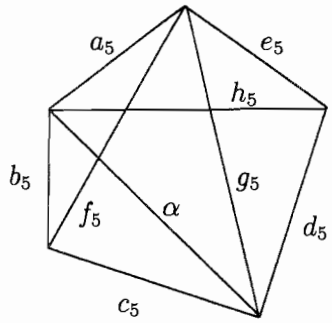



$(x_1 - b)^2 + y_1^2 = c^2, (x_1 - x_2)^2 + (y_1 - y_2)^2 = e^2, x_2^2 + y_2^2 = f^2, x_1^2 + y_1^2 = a^2, (x_2 - b)^2 + y_2^2 = d^2$

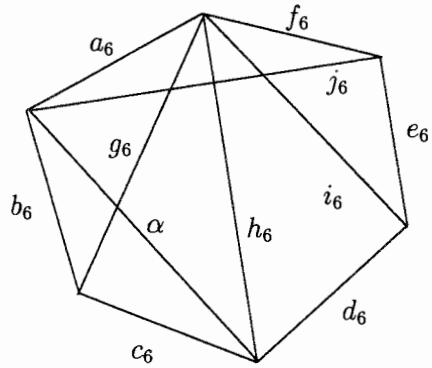
これより, グレブナー基底を用いて

$$a^2d^2(b^2 + c^2 + e^2 + f^2) - a^2d^4 - a^4d^2 + b^2e^2(a^2 + c^2 + d^2 + f^2) - b^2e^4 - b^4e^2 - c^2f^4$$
$$+c^2f^2(a^2 + b^2 + e^2 + d^2) - c^4f^2 - a^2b^2c^2 - a^2e^2f^2 - b^2f^2d^2 - c^2d^2e^2 = 0 \qquad (7)$$

という22項の式を得る．左辺を$F(a,b,c,d,e,f)$とする．




5角形は，$F_1 = F(\alpha, b_5, c_5, f_5, g_5, a_5), F_2 = F(h_5, \alpha, d_5, g_5, e_5, a_5)$ として，$\alpha$ について $F_1$ と $F_2$ の終結式 $G(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5) = \mathrm{res}_\alpha(F_1, F_2)$ を計算する．$F_1 = (F_1(0)) + (F_1(2))\alpha^2 + (F_1(4))\alpha^4, F_2 = (F_2(0)) + (F_2(2))\alpha^2 + (F_2(4))\alpha^4$ より，終結式計算の計算量削減のための公式(3)が利用でき，$F_1' = (F_1(0)) + (F_1(2))\beta + (F_1(4))\beta^2, F_2' = (F_2(0)) + (F_2(2))\beta + (F_2(4))\beta^2$ として $H(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5) = \mathrm{res}_\beta(F_1', F_2')$ とすれば $G(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5) = H(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5)^2$ となる．$H(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5)$ は843項であり，$G(a_5, b_5, c_5, d_5, e_5, f_5, g_5, h_5)$ は29125項である．




6角形は，$U_1 = F(\alpha, b_6, c_6, g_6, h_6, a_6), U_2 = G(a_6, \alpha, d_6, e_6, f_6, h_6, i_6, j_6)$ として $\alpha$ について $U_1$ と $U_2$ の終結式 $K(a_6, b_6, c_6, d_6, e_6, f_6, g_6, h_6, i_6, j_6) = \mathrm{res}_\alpha(U_1, U_2)$ を計算する．$U_3 = H(a_6, \alpha, d_6, e_6, f_6, h_6, i_6, j_6)$ とすると，$K(a_6, b_6, c_6, d_6, e_6, f_6, g_6, h_6, i_6, j_6) = [\mathrm{res}_\alpha(U_1, U_3)]^2$ となる．$U_1, U_3$ について丁寧にみると $U_1 = (U_1(0)) + (U_1(2))\alpha^2 + (U_1(4))\alpha^4, U_3 = (U_3(0)) + (U_3(2))\alpha^2 + (U_3(4))\alpha^4 + (U_3(6))\alpha^6 + (U_3(8))\alpha^8$ という構造をしている．$U_1' = (U_1(0)) + (U_1(2))\beta + (U_1(4))\beta^2, U_3' = (U_3(0)) + (U_3(2))\beta + (U_3(4))\beta^2 + (U_3(6))\beta^3 + (U_3(8))\beta^4$ とすると，終結式計算の計算量削減のための公式(3)が利用でき，$K(a_6, b_6, c_6, d_6, e_6, f_6, g_6, h_6, i_6, j_6) = [\mathrm{res}_\beta(U_1', U_3')]^4$ となる．$\mathrm{res}_\beta(U_1', U_3')$ は，273123項である．この終結式の計算は，関孝和先生の定義とラプラス展開を利用して行った．

[5] K. O. Geddes, Stephen R. Czapor, George Labahn, Keith O. Geddes, S. R. Czapor, G. Labahn, Algorithms for Computer Algebra, Kluwer Academic Pub., United States, 1992.

[6] A.J. Goldstein, R.L. Graham, A Hadamard-type bound on the coefficient of a determinant of polynomials, SIAM Review 16, 394-395, (1974).

[7] 平山諦: 関孝和 その業績と伝記, 恒星社版, 東京,1974.

[8] 広田良吾: 行列式とパフィアン (1), 日本応用数理学会誌「応用数理」, **14(1)**, 2004, pp.62-66.

[9] 木村欣司, 多項式行列の行列式の補間による計算 II, 京都大学数理解析研究所講究録 1514「Computer Algebra-Desingn of Algorithms, Implementations and Applications」, 2005, pp.176-182.

[10] K. Kimura, M. Noro: Automatic weight generator for the Buchberger algorithm, In Proceedings of International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS2006), 2006, pp.33-44.

[11] 野呂正行: private communication.

[12] 野呂正行, 横山和弘: グレブナー基底の計算 基礎篇, 東京大学出版会, 東京, 2003.

[13] 大石進一: 非線形解析入門, コロナ社, 東京, 2000.

[14] 佐々木建昭, 金田康正, 渡辺隼郎: 終結式計算アルゴリズムと判別式の計算, 情報処理学会研究報告「プログラミング」, Vol.1981 No.28, 1981.

[15] 佐々木建昭, 村尾裕一: 記号行列に対する効率的なガウス消去法, 情報処理学会研究報告「プログラミング」, Vol.1979 No.50, 1980.

[16] 高木貞治: 代数学講義 (改訂新版), 共立出版, 東京, 1965.

[17] J.R.Johnson, Algorithms for Polynomial Real Root Isolation, Quantifier Elimination and Cylindrical Algebraic Decomposition, SpringerWienNewYork, Austria, 1998.

[18] Deepak Kapur, Tushar Saxena, Lu Yang: Algebraic and geometric reasoning using Dixon resultants, International Conference on Symbolic and Algebraic Computation archive, Proceedings of the international symposium on Symbolic and algebraic computation, 1994, pp.99-107.

## 5.2 1458 次式の問題

### 5.2.1 1458 次式の導出

式 (7) のほかに,

$$d^3 - b^3 = 271, b^3 - c^3 = 217, c^3 - a^3 = 60.8, a^3 - e^3 = 326.2, e^3 - f^3 = 61 \qquad (8)$$

という条件も課す. 終結式を利用して $f$ のみの式を作ると 1458 次式が得られる. この計算には, さまざまな工夫が考えられる. その中で, 他の問題にも使える終結式計算の効率化のための工夫として, 式 (8) の次数が低いことから擬剰余を用いて $3 \times 3$ の行列を構成し, ラプラス展開を用いてその行列式を展開することで終結式計算する方法をあげる. これ以外にも, この問題に特化したさまざまな工夫が考えられるが, ここでは省略する. [11] 一方で, グレブナー基底を計算し, 有限体上で $f$ の最小多項式を計算すると 1458 次式になるために, この終結式の計算は必要十分条件であったことがわかる. すなわち, 1458 次式の中には偽因子は混入していない.

### 5.2.2 real root finding

倍写像行列を作り, Gerschgorin の定理をもちいて実根の存在範囲を計算すると
$a = [-52589152874765992, 52589152874765992]$
$b = [-42146023482750655, 42146023482750655]$
$c = [-9147118262947078, 9147118262947078]$
$d = [-30423673761003874, 30423673761003874]$
$e = [-14776817925389682, 14776817925389682]$
$f = [-15965294561020909, 15965294561020909]$
とわかる. この範囲において, Krawczyk 法を適用すると実根は 8 つであり, すべてが正の根は唯一であることが証明できる.

## 参 考 文 献


[1] 有本卓, 数値解析 (I), コロナ社, 東京, 1997.

[2] D. N. Bernshtein, "The number of roots of a system of equations," *Functional Analysis and Appl.*, **9(3)**, 1975, pp. 183-185.

[3] David Cox,Donal O ' Shea,John Little: グレブナー基底 1 代数幾何と可換代数におけるグレブナー基底の有効性 (大杉 英史, 日比 孝之, 北村 知徳 訳), シュプリンガー・フェアラーク東京, 東京, 2000.

[4] J. F. Canny, I. Z. Emiris: An Efficient Algorithm for the Sparse Resultant, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, International Conference, 1993.