数理解析研究所講究録 1395 巻 2004 年 64-70

多変数多項式の絶対既約半径の改良？

長坂耕作

KOSAKU NAGASAKA

山口大学メディア基盤センター

MEDIA AND INFORMATION TECHNOLOGY CENTER, YAMAGUCHI UNIVERSITY*

1 取り上げる問題

Kaltofen の 1995 年の論文 [Kal95] で少し触れられていたものに関して，ISSAC2002 の論文 [Nag02] である程度実用的なものを提案した次の問題を取り上げる．なお，2003 年には，ISSAC2002 の枠組みをベースにした新しい 2 つの発表が行われている [KM03, Nag03]．特に，ISSAC2003 での Kaltofen と May の発表 [KM03] は，偏微分方程式による既約条件（Ruppert[Rup99]）を使ったもので，かなり効果的な手法となっている．

Problem 1 *For the given polynomial $f \in \mathbb{C}[x,y]$ which is absolutely irreducible, compute the largest possible value $B(f) \in \mathbb{R}_{>0}$ such that all $\tilde{f} \in \mathbb{C}[x,y]$ with $\|f - \tilde{f}\| < B(f)$ (and $\deg(\tilde{f}) \leq \deg(f)$) must remain absolutely irreducible.* ◁

この値 $B(f)$ を絶対既約半径または分離精度と呼ぶことにする．なお，ISSAC2002 と CASC2003 の私の発表では，$B(f)$ を与式のノルムで割った値 $B(f)/\|f\|$ を使用している．

既存の手法について，その有効性を実験したところ，表 1 のような結果が得られた（実験では，係数を実数区間 $[-1,1]$ からランダムに生成した，次数がそれぞれ 7 次と 6 次の x と y の二変数多項式を使用）．なお，「ISSAC2003(for sparse polynomials)」は，Kaltofen と May の論文 [KM03] に記載されている Gao と Rodrigues による，疎な多項式に対する既約条件式 [GR03] も使ったもので，多項式が疎な場合に結果が改善される．さて，表 1 の結果からも明らかなように，最も効果的なアルゴリズムは Kaltofen と May のもの

使用したアルゴリズム	計算された $B(f)/\|f\|$ の平均値
ISSAC2002(Nagasaka)	2.658×10^{-7}
+CASC2003(Nagasaka)	1.948×10^{-6}
ISSAC2003(Kaltofen and May)	1.021×10^{-2}
+ISSAC2003(for sparse polynomials)	1.026×10^{-2}

表 1: 実験結果

であった．ただし，まだ若干の改良の余地が残されているので，それについて議論することが本稿の目的である．

*nagasaka@yamaguchi-u.ac.jp

2 Kaltofen と May の方法

改良する基となる Kaltofen と May の方法について，より一般化し，実装時の計算速度が向上する形で，説明しておく．既約条件として使用するのは，Ruppert や Gao らの偏微分方程式を用いた次の既約条件式である．

$$f\frac{\partial g}{\partial y} - g\frac{\partial f}{\partial y} + h\frac{\partial f}{\partial x} - f\frac{\partial h}{\partial x} = 0,$$

$$\deg_x g \le \deg_x f - 1,\ \deg_y g \le \deg_y f,\ \deg_x h \le \deg_x f,\ \deg_y h \le \deg_y f - 2.$$

この偏微分方程式から，g と h の係数を未知数とする線形方程式を作り，その係数行列を Ruppert 行列 $R(f)$ とおく．ここで，$n = \deg_x(f)$ かつ $m = \deg_y(f)$ とすれば，Ruppert 行列 $R(f)$ のサイズは，$(4nm) \times (2nm + m - 1)$ となる．与多項式を，係数に変数を導入し，

$$f = \sum c_{i,j} x^i y^j,\ \in \mathbb{C}$$

と表現し，そのときの Ruppert 行列 $R_{\mathrm{var}}(f)$ を次のように定義する．

$$R_{\mathrm{var}}(f) = \sum R_{i,j} c_{i,j},\ R_{i,j} \in \mathbb{Z}^{(4nm)\times(2nm+m-1)}.$$

もともとの Kaltofen らの計算式は，次のように表すことができる．

$$B(f) = \sigma(R(f)) / \max_{i,j} \|R_{i,j}\|_F .$$

ここで $\sigma(A)$ は，A の $(2nm + m - 1)$ 番目に大きい (与式が既約なら最小の) 特異値を，$\|A\|_F$ は，A のフロベニウスノルムを表す．次の項のように，$R_{i,j}$ を使うことで，$\max_{i,j} \|R_{i,j}\|_F$ の値は，元の論文よりも若干速く計算することができる．

2.1 分離精度公式の分母の計算

行列 $R(f)$ を (多少冗長にはなるが)，ブロック成分 G_i と H_i を使って表現すると次のようになる．この行列のサイズは冗長に構成しているので，$4nm \times (2mn + m - 1)$ となっている．

$$\begin{pmatrix}
G_n & 0 & \cdots & 0 & 0\cdot H_n & 0 & \cdots & 0 & 0 \\
G_{n-1} & G_n & \ddots & \vdots & -H_{n-1} & H_n & \ddots & \vdots & \vdots \\
\vdots & G_{n-1} & \ddots & 0 & \vdots & 0\cdot H_{n-1} & \ddots & 0 & \vdots \\
G_1 & \vdots & \ddots & G_n & (1-n)H_1 & \vdots & \ddots & (n-1)H_n & 0 \\
G_0 & G_1 & \ddots & G_{n-1} & -nH_0 & (2-n)H_1 & \ddots & (n-2)H_{n-1} & nH_n \\
0 & G_0 & \ddots & \vdots & 0 & (1-n)H_0 & \ddots & \vdots & (n-1)H_{n-1} \\
\vdots & \ddots & \ddots & G_1 & \vdots & 0 & \ddots & 0\cdot H_1 & \vdots \\
0 & \cdots & 0 & G_0 & 0 & \cdots & 0 & -1H_0 & H_1
\end{pmatrix}$$

各ブロック成分 G_i は次の行列で，サイズは $2m \times (m+1)$.

$$\begin{pmatrix}
0 & 0 & \cdots & 0 & 0 & 0 \\
c_{i,m-1} & -c_{i,m} & \ddots & \vdots & \vdots & 0 \\
2c_{i,m-2} & 0 & \ddots & 0 & \vdots & \vdots \\
\vdots & c_{i,m-2} & \ddots & (2-m)\,c_{i,m} & 0 & \vdots \\
\vdots & \vdots & \ddots & \vdots & (1-m)\,c_{i,m} & 0 \\
m\,c_{i,0} & \vdots & \ddots & \vdots & \vdots & -m\,c_{i,m} \\
0 & (m-1)\,c_{i,0} & \ddots & 0 & \vdots & \vdots \\
\vdots & 0 & \ddots & c_{i,1} & -c_{i,2} & \vdots \\
0 & \vdots & \ddots & 2c_{i,0} & 0 & -2c_{i,2} \\
0 & 0 & \cdots & 0 & c_{i,0} & -c_{i,1}
\end{pmatrix}$$

各ブロック成分 H_i は次の行列で，サイズは $2m \times (m-1)$.

$$\begin{pmatrix}
0 & \cdots & 0 \\
c_{i,m} & \ddots & \vdots \\
c_{i,m-1} & \ddots & 0 \\
\vdots & \ddots & c_{i,m} \\
c_{i,1} & \ddots & c_{i,m-1} \\
c_{i,0} & \ddots & \vdots \\
 & \ddots & c_{i,1} \\
0 & & c_{i,0}
\end{pmatrix}$$

従って，$\max_{i,j} \|R_{i,j}\|_F$ は次のように計算できる.

まず，$\|R_{i,j}\|_F$ の H に関する右部分 $\|R_{i,j}\|_{F_H}$ を計算する.

$$\|R_{i,j}\|_{F_H}^2 = (m-1)(\sum_{l=1}^{i} l^2 + \sum_{l=-n+i}^{-1} l^2) = (m-1) \sum_{l=-n+i}^{i} l^2.$$

次に，$\|R_{i,j}\|_F$ の G に関する左部分 $\|R_{i,j}\|_{F_G}$ を計算する.

$$\|R_{i,j}\|_{F_G}^2 = n(\sum_{l=-j}^{-1} l^2 + \sum_{l=1}^{m-j} l^2) = n \sum_{l=-j}^{m-j} l^2.$$

これらをまとめると，$\|R_{i,j}\|_F$ は次のようになる.

$$\begin{aligned} \|R_{i,j}\|_F^2 &= \|R_{i,j}\|_{F_G}^2 + \|R_{i,j}\|_{F_H}^2 \\ &= n \textstyle\sum_{l=-j}^{m-j} l^2 + (m-1) \sum_{l=-n+i}^{i} l^2. \end{aligned}$$

つまり，最大値は次のように計算できる.

$$\max_{i,j} \|R_{i,j}\|_F^2 = \frac{n(m(m+1)(2m+1) + (m-1)(n+1)(2n+1))}{6}.$$

なお，このとき次の関係が成立していることも明らか.

$$\max_{i,j} \|R_{i,j}\|_F = \|R_{n,m}\|_F = \|R_{n,0}\|_F = \|R_{0,m}\|_F = \|R_{0,0}\|_F .$$

3 若干の改良の余地

取り扱っている偏微分方程式の定義から，g と h の係数を未知数とした線形方程式は，未知数の数よりも方程式（制約式）の数が多い過剰決定系となる．$B(f)$ の主な構成要因である特異値（計算式の分子に対応）の大きさは，過剰決定系の方が一般に大きくなるが，実際には，$\tilde{f}$ の f からの係数の変動上限（計算式の分母に対応）も同時に計算する必要があるので，デメリットも存在することになる．

この点で，実装上，Kaltofen と May の方法には若干の改良の余地が残されている．実際に，ISSAC2003 において，ISSAC2002 との比較のために利用されている次の多項式の例で，改良の余地を示す．

$$f = (x^2 + yx + 2y - 1)(x^3 + y^2x - y + 7) + 0.2x.$$

この多項式から生成される ISSAC2003(Kaltofen and May) の Ruppert 行列において，制約条件である方程式をうまく取り除くと，表 2 のようなより良い結果が得られる．一見すると，方程式の数が減ることに

削除した方程式の数	0	1	2	3	4	10
$B(f)/\|\|f\|\|$ の値（単位は 10^{-5}）	3.868	3.868	3.963	3.993	4.034	4.247

表 2: 行の削除によるメリット

より特異値が小さくなり，$B(f)$ の値も小さくなってしまいそうだが，同時に行列の変動上限も小さくなるので，削除する方程式によっては $B(f)$ の値が大きくなるというメリットが生じている．

3.1 最適な行削除の考察

前項で，Ruppert 行列から制約式を削除することで，分離精度を大きく出来ることがわかった．そこで，どの行をどれくらい削除すれば良いのかについて考察してみる．

行列の行（制約式に対応）の削除を意味する記号を次のように定義しておく．

$$\mathrm{drop}_i(A) = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{i-1}, \boldsymbol{a}_{i+1}, \ldots, \boldsymbol{a}_{k_1})^t, \quad A \in \mathbb{C}^{k_1 \times k_2}.$$

すると，今回の最適な行削除を求める問題は，与えられた $k(\leq 2nm - m + 1)$ に対して，

$$B^{(k)}(f) = \sigma(R^{(k)}(f)) / \max_{i,j} \|R^{(k)}_{i,j}\|_F$$

を最大化するような次の $R^{(k)}(f)$ と $R^{(k)}_{i,j}$ を与える組 $d_1, \ldots, d_k$ を求めることになる．

$$R^{(k)}(f) = \mathrm{drop}_{d_k}(\mathrm{drop}_{d_{k-1}}(\cdots(\mathrm{drop}_{d_1}(R(f))))).$$

$$R^{(k)}_{i,j} = \mathrm{drop}_{d_k}(\mathrm{drop}_{d_{k-1}}(\cdots(\mathrm{drop}_{d_1}(R_{i,j})))).$$

本稿では，この問題に対する完璧な解の提供には至っておらず，とても簡易な方法のみを考察している．

3.2 行削除によるノルムの変化

まず，第 $2md_x + d_y (0 \leq d_x, 1 \leq d_y \leq 2m)$ 行の削除による $\|R_{i,j}\|_F$ の変化を，次式を満たす，$R_{i,j}$ の左側部分の変動 Δ_G と $R_{i,j}$ の右側部分の 変動 Δ_H について分けて計算することで求める．

$$\|\mathrm{drop}_{2md_x+d_y}(R_{i,j})\|_F^2 = \|R_{i,j}\|_F^2 - \Delta_G - \Delta_H.$$

最初に，左側の G_i 部分のノルムについて考える．左側のノルムが減少するには，インデックス i は $n-d_x$ から $2n-d_x-1$ までになければいけない．ただし，0 未満や n より大きいインデックスについては，無視することとする．i を固定して G_i の要素を考えることで，インデックス j は $m-d_y+1$ から $2m+1-d_y$ までになければいけないこともわかる．この場合も，0 未満や m より大きいインデックスについては，無視することとする．

減少するノルムの大きさはインデックス j にのみ依存するので，次のように計算できる．

$$(d_y-1-2(j-(m-d_y+1)))^2=(1-d_y-2j+2m)^2.$$

従って，次を得る．

$$\Delta_G=\begin{cases}0 & (i<n-d_x)\\ 0 & (2n-d_x-1<i)\\ 0 & (j<m-d_y+1)\\ 0 & (2m+1-d_y<j)\\ (2m+1-d_y-2j)^2\end{cases}$$

同様に，右側の H_i 部分のノルムについても考える．右側のノルムが減少するには，インデックス i は $n-d_x$ から $2n-d_x$ までになければいけない．ただし，0 未満や n より大きいインデックスについては，無視することとする．i を固定して H_i の要素を考えることで，インデックス j は $m-d_y+2$ から $2m-d_y$ までになければいけないこともわかる．この場合も，0 未満や m より大きいインデックスについては，無視することとする．

減少するノルムの大きさはインデックス i にのみ依存するので，次のように計算できる．

$$(-d_x+2(i-(n-d_x)))^2=(2i+d_x-2n)^2.$$

従って，次を得る．

$$\Delta_H=\begin{cases}0 & (i<n-d_x)\\ 0 & (2n-d_x<i)\\ 0 & (j<m-d_y+2)\\ 0 & (2m-d_y<j)\\ (2i+d_x-2n)^2\end{cases}$$

3.3 行削除は少なくとも 2 行必要

前項で求めた表を利用して，行削除により分離精度を大きくするには，少なくとも 2 行必要であることを示す．既に示した通り，

$$\max_{i,j}\|R_{i,j}\|_F=\|R_{n,m}\|_F=\|R_{n,0}\|_F=\|R_{0,m}\|_F=\|R_{0,0}\|_F\ .$$

が成立しているので，削除行を表す d_x と d_y の選択は上記インデックスの組 $(i,j)=\{(n,m),(n,0),(0,m),(0,0)\}$ を小さくすることが少なくとも必要になる．そこで，各組によるノルムの減少が 0 にならない条件を調べる．

(n,m) の組について．

$$(0\le d_x\le n-1\ \wedge\ 1<d_y\le m+1)\ \vee\ (0<d_x\le n\ \wedge\ 2\le d_y\le m)$$

$(n,0)$ の組について．

$$(0\le d_x\le n-1\ \wedge\ m+1\le d_y<2m+1)\ \vee\ (0<d_x\le n\ \wedge\ m+2\le d_y\le 2m)$$

$(0,m)$ の組について.

$$(n \leq d_x \leq 2n-1 \ \wedge \ 1 < d_y \leq m+1) \ \vee \ (n \leq d_x < 2n \ \wedge \ 2 \leq d_y \leq m)$$

$(0,0)$ の組について.

$$(n \leq d_x \leq 2n-1 \ \wedge \ m+1 \leq d_y < 2m+1) \ \vee \ (n \leq d_x < 2n \ \wedge \ m+2 \leq d_y \leq 2m)$$

つまり，これら 4 組の条件を同時に満たす d_x と d_y は存在しないことになる．従って，意味のある行削除を行うには，2 行削除，即ち $k \geq 2$ が条件となる．実際，$k=2$ であれば，上記 4 条件式から以下の条件を導出できる．

(n,m) と $(n,0)$，$(0,m)$ と $(0,0)$ で組み合わせる場合.

$$\begin{aligned} &1\text{行目}: \ 0 \leq d_x \leq n-1 \ \wedge \ d_y = m+1 \\ &2\text{行目}: \ n \leq d_x \leq 2n-1 \ \wedge \ d_y = m+1 \end{aligned}$$

(n,m) と $(0,m)$，$(n,0)$ と $(0,0)$ で組み合わせる場合.

$$\begin{aligned} &1\text{行目}: \ d_x = n \ \wedge \ 2 \leq d_y \leq m \\ &2\text{行目}: \ d_x = n \ \wedge \ m+2 \leq d_y \leq 2m \end{aligned}$$

4 インプリメンテーション

これらの考察を基に次のような簡単な 3 つのアルゴリズムを実装してみた.

Algorithm 1 （二行同時削除のみ）

1. 以下を満たす削除候補行 $2md_x + d_y$ の組の生成を行う.

$$\begin{aligned} &\{0 \leq d_x \leq n-1 \ \wedge \ d_y = m+1 \quad , \quad n \leq d_x \leq 2n-1 \ \wedge \ d_y = m+1\} \\ &\{d_x = n \ \wedge \ 2 \leq d_y \leq m \qquad\quad , \quad d_x = n \ \wedge \ m+2 \leq d_y \leq 2m\} \end{aligned}$$

2. 全ての削除候補組で絶対既約半径（分離精度）を計算し，最大値を返す.

◁

Algorithm 2 （二行同時削除の繰り返し）

1. 分離精度が変わらなくなるまで，Algorithm 1 を適用する.

◁

Algorithm 3 （二行同時削除の繰り返し後の三行同時削除の繰り返し）

1. Algorithm 2 を適用する.
2. 以下を満たす削除候補行 $2md_x + d_y$ を三行生成する.

$$\begin{aligned} &\{0 \leq d_x \leq n-1 \ \wedge \ d_y = m+1 \quad , \quad n \leq d_x \leq 2n-1 \ \wedge \ d_y = m+1, \\ &\ d_x = n \ \wedge \ 2 \leq d_y \leq m \qquad\quad , \quad d_x = n \ \wedge \ m+2 \leq d_y \leq 2m\} \end{aligned}$$

使用したアルゴリズム	計算された $B(f)/\|\|f\|\|$ の平均値	改善率
ISSAC2003(Kaltofen and May)	1.021×10^{-2}	-
+ISSAC2003(for sparse polynomials)	1.026×10^{-2}	0.5%
2 行削除	1.037×10^{-2}	1.6%
2 行削除繰り返し	1.070×10^{-2}	4.8%
2 行削除繰返＋3 行削除繰返	1.082×10^{-2}	6.0%

表 3: 実験結果

3. 削除候補組の中で絶対既約半径（分離精度）を計算し，改善される場合は，その三行を削除する．

4. 分離精度が変わらなくなるまで，ステップ 2 と 3 を繰り返す．

◁

これらのアルゴリズムを，冒頭の実験で使用したのと同じ多項式 100 個に対して，実験した結果が表 3 である．実際に，不要行削除の効果が現れていることが読み取れる．

しかしながら，2 行削除で 19%，2 行削除繰り返しで 548%，2 行削除繰返＋3 行削除繰返で 7037% も計算時間が増えてしまった．係数の最大値を最終的な値を除き，$(n,m),(n,0),(0,m),(0,0)$ 次数の係数変数についてのみ計算を行うなど，細かい部分の速度向上は行ってみたものの，計算量が指数関数的なため計算時間の増加は不可避であった．

5 まとめ

適切な不要行を Ruppert 行列から削除することにより，Kaltofen と May による方法を少しは改善（最大で 6%）することが出来たが，計算時間を考慮すると 2 行削除が実用的な限界と思われる（1.6%の向上）．この方法は，Gao らの疎な場合の条件式利用の場合にも有効であり，更なる向上が見込まれるだろう．

参 考 文 献

[GR03] S. Gao and V. M. Rodrigues. Irreducibility of polynomials modulo p via Newton Polytopes. *J. Number Theory*, 101:32–47, 2003.

[Kal95] E. Kaltofen. Effective noether irreducibility forms and applications. *J. Computer and System Sciences*, 50:274–295, 1995.

[KM03] E. Kaltofen and J. May. On Approximate Irreducibility of Polynomials in Several Variables. *ISSAC 2003 Proc.*, pages 161–168, 2003.

[Nag02] K. Nagasaka. Towards Certified Irreducibility Testing of Bivariate Approximate Polynomials. *ISSAC 2002 Proc.*, pages 192–199, 2002.

[Nag03] K. Nagasaka. Neighborhood Irreducibility Testing of Multivariate Polynomials. *CASC 2003 Proc.*, pages 283–292, 2003.

[Rup99] W. M. Ruppert. Reducibility of polynomials $f(x,y)$ modulo p. *J. Number Theory*, 77:62–70, 1999.