疎多項式補間に関するrobustな計算手法及びアルゴリズム

Robust computation methods and algorithms for sparse interpolation of multivariate polynomials

東京理科大学大学院　　近藤 和希 *1
Kazuki Kondo
Graduate School, Tokyo University of Science

東京理科大学　　関川 浩 *2
Hiroshi Sekigawa
Tokyo University of Science

Abstract

There is some risk that large numerical errors occur when solving the Hankel system in the Giesbrecht, Labahn, and Lee algorithm that interpolates a sparse polynomial. We carry out experiments by applying several solving algorithms to the Hankel system and compare the results of interpolation.

1　はじめに

代入することで値が得られるが，どのような式であるかは分からない多項式を black-box 多項式と呼ぶ．疎な (項数が少ない) black-box 多変数多項式 $f(x_1,\ldots,x_n)=\sum_{j=1}^{t}c_j x_1^{d_{j,1}}\cdots x_n^{d_{j,n}}\in\mathbb{C}[x_1,\ldots,x_n]$ $(c_j\neq 0)$ が与えられたとき，適切な入力点とそれに対する出力を用いることで，この多項式の係数と指数を決定する問題を疎多項式補間問題と呼ぶ．本稿では，疎多項式補間を行うアルゴリズムにおける，悪条件な線型方程式を解く手法を Mathematica の内部関数 LinearSolve から変更し，その結果を実験によって調べ，考察を行う．以下，2 節において，先行研究である Ben-Or と Tiwari によるアルゴリズムとその派生手法である Giesbrecht，Labahn，Lee によるアルゴリズムについて述べ，後者の概要を記す．そして，そのアルゴリズムの中に現れる Hankel system を解く手法である Levinson-type のアルゴリズムについて述べる．3 節では，Giesbrecht，Labahn，Lee によるアルゴリズムにおいて，Hankel system を解く手法を種々変更した実験結果を載せ，その考察を行う．最後に 4 節において，まとめを行う．

2　先行研究

先行研究として，Ben-Or と Tiwari によるアルゴリズム [1] (以下 BT と略記)，及び派生手法である Giesbrecht，Labahn，Lee によるアルゴリズム [2] (以下 GLL と略記) が知られている．BT は入力する値に素数のべきを用いて誤差のない計算を行い，GLL は入力する値に 1 のべき根を用いて数値計算を行う．以下では GLL の概要を記述した後に，その手法で現れる Hankel system を解くアルゴリズムについて述べていく．

*1　E-mail: 1418503@alumni.tus.ac.jp
*2　E-mail: sekigawa@rs.tus.ac.jp

2.1 Giesbrecht, Labahn, Lee によるアルゴリズム

本節では，Giesbrecht, Labahn, Lee によるアルゴリズムについて記述する．このアルゴリズムは，入力点に 1 のべき根を用いて数値計算を行う．これによって，black-box から出力される値の絶対値の増加を防ぐことができる．なお，入力点に素数を用いて誤差のない計算を行う BT は，GLL では必要となる $D_k \geq \deg_{x_k}(f)$ なる指数の上界 $(D_1, \ldots, D_n)$ を必要としない．

アルゴリズム 1 (Giesbrecht, Labahn, Lee [2] (2009))

Input : black-box 多変数多項式 $f(x_1, \ldots, x_n)$，f の項数 t，及び指数の上界 $(D_1, \ldots, D_n)$．

Output : $f(x_1, \ldots, x_n) = \sum_{j=1}^{t} c_j x_1^{d_{j,1}} \cdots x_n^{d_{j,n}}$ なる係数 c_j 及び指数 $(d_{j,1}, \ldots, d_{j,n})$ $(1 \leq j \leq t)$．

1. 互いに素で $p_1 > D_1, \ldots, p_n > D_n$ であるような $p_1, \ldots, p_n \in \mathbb{N}$ を選び，$2t$ 個の点における値

$$\alpha_s = f(\omega_1^s, \ldots, \omega_n^s) \qquad (0 \leq s \leq 2t-1)$$

を求める (ただし，$\omega_k = \exp(2\pi i/p_k)$)．また，

$$\begin{aligned} m &= p_1 \cdots p_n \\ \omega &= \exp(2\pi i/m) \\ \beta_j(x_1, \ldots, x_n) &= x_1^{d_{j,1}} \cdots x_n^{d_{j,n}} \\ \omega^{d_j} &= \beta_j(\omega_1, \ldots, \omega_n) \end{aligned}$$

とする．

2.

$$\Lambda(z) = \prod_{j=1}^{t} (z - \omega^{d_j}) = \sum_{h=0}^{t} \lambda_h z^h$$

として (ただし，$\lambda_t = 1$)，以下の Hankel 行列を係数行列とする方程式

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \ldots & \alpha_{t-1} \\ \alpha_1 & \alpha_2 & \ldots & \alpha_t \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{t-1} & \alpha_t & \ldots & \alpha_{2t-2} \end{bmatrix} \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_{t-1} \end{bmatrix} = - \begin{bmatrix} \alpha_t \\ \alpha_{t+1} \\ \vdots \\ \alpha_{2t-1} \end{bmatrix} \tag{1}$$

を解き，$\lambda_0, \ldots, \lambda_{t-1}$ を求める．

3. 1 変数多項式 $\Lambda(z)$ の根 $\omega^{d_1}, \ldots, \omega^{d_t}$ を求める．

4. ω^{d_j} から d_j $(1 \leq j \leq t)$ を求め，

$$d_j = d_{j,1} \cdot \frac{m}{p_1} + \cdots + d_{j,n} \cdot \frac{m}{p_n}$$

であるので，それぞれの指数 $d_{j,1}, \ldots, d_{j,n}$ を求める．

5. 以下の Vandermonde 行列を係数行列とする方程式

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \omega^{d_1} & \omega^{d_2} & \ldots & \omega^{d_t} \\ \vdots & \vdots & \ddots & \vdots \\ (\omega^{d_1})^{t-1} & (\omega^{d_2})^{t-1} & \ldots & (\omega^{d_t})^{t-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{t-1} \end{bmatrix} \tag{2}$$

を解き，係数 $c_1, \ldots, c_t$ を求める．

6. 係数 $c_1, \ldots, c_t$ と，指数 $d_{1,1}, \ldots, d_{t,n}$ を返す．

GLL において，大きな誤差が発生する危険があるのは，式 (1) の Hankel system と式 (2) の Vandermonde system であり，特に Hankel system の解は許容誤差が小さく，その後の影響も大きい．

2.2 Levinson-type のアルゴリズム

本節では，$n \times n$ の Hankel 行列

$$H_n = \begin{bmatrix} h_1 & h_2 & \ldots & h_n \\ h_2 & h_3 & \ldots & h_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_n & h_{n+1} & \ldots & h_{2n-1} \end{bmatrix}$$

を係数行列に持つ Hankel system $H_n \boldsymbol{z} = \boldsymbol{b}$ を解く Levinson-type のアルゴリズム [3] の概要について述べる ($\boldsymbol{b} = (b_i)_{i=1}^n$ は任意のベクトル)．H_k を H_n における次の小行列とする．

$$H_k = \begin{bmatrix} h_1 & h_2 & \ldots & h_k \\ h_2 & h_3 & \ldots & h_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_k & h_{k+1} & \ldots & h_{2k-1} \end{bmatrix} \qquad (1 \le k \le n)$$

ここで，各 H_k $(1 \le k \le n)$ は正則であると仮定する (これを strong nonsingularity 性という)．また $\boldsymbol{x}_k$ を $H_k \boldsymbol{x}_k = \boldsymbol{e}_k$ を満たすベクトルとし (e_k は k 番目の成分が 1 である単位ベクトル)，

$$\begin{aligned} \sigma_k &= \begin{bmatrix} h_{k+1} & \ldots & h_{2k} \end{bmatrix} \boldsymbol{x}_k, \\ \sigma'_k &= \begin{bmatrix} h_{k+2} & \ldots & h_{2k+1} \end{bmatrix} \boldsymbol{x}_k, \\ \tau_k &= \sigma'_k - \sigma'_{k-1} - (\sigma_k - \sigma_{k-1})\sigma_k \end{aligned}$$

とすると，$k = 1, 2, \ldots, n-1$ に対して，$\boldsymbol{x}_k$ は次の再帰関係を満たす．

$$\boldsymbol{x}_{k+1} = \frac{1}{\tau_k} \left(\begin{bmatrix} 0 \\ \boldsymbol{x}_k \end{bmatrix} - (\sigma_k - \sigma_{k-1}) \begin{bmatrix} \boldsymbol{x}_k \\ 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{x}_{k-1} \\ 0 \\ 0 \end{bmatrix} \right). \tag{3}$$

ただし，

$$\begin{aligned} \boldsymbol{x}_1 &= 1/h_1, \\ \sigma_1 &= h_2/h_1, \\ \sigma'_1 &= h_3/h_1, \\ \sigma_0 &= \sigma'_0 = 0 \end{aligned}$$

で，$\boldsymbol{x}_0$ は empty とする．この再帰関係を用いて $\boldsymbol{x}_k$ を求める手法を Levinson-type のアルゴリズムという．Hankel 行列 H_n が strong nonsingularity 性を持つならば，各小行列 H_k は逆行列を持ち，従って $\boldsymbol{x}_k$ が存在する ($\tau_k \neq 0$ である)．さらに，$\boldsymbol{z}_k$ を $H_k \boldsymbol{z}_k = \boldsymbol{b}_k$ を満たすベクトルとし ($\boldsymbol{b}_k = (b_i)_{i=1}^k$)，

$$\beta_k = \begin{bmatrix} h_{k+1} & \ldots & h_{2k} \end{bmatrix} \boldsymbol{z}_k$$

とすると，$k=1,2,\ldots,n-1$ に対して，$\boldsymbol{z}_k$ は次の再帰関係を満たす．

$$\boldsymbol{z}_{k+1}=\begin{bmatrix}\boldsymbol{z}_k\\0\end{bmatrix}+(b_{k+1}-\beta_k)\boldsymbol{x}_{k+1}$$

ただし，$\boldsymbol{z}_{\boldsymbol{1}}=b_1/h_1$ とする．この再帰関係を用いて $\boldsymbol{z}_k$ を求める手法を bordering method という．この手法は Hankel system に限らず，係数行列が strong nonsingularity 性を持つ任意の線型方程式に適用することができる．計算量は，1 回の乗算を 1(M)，1 回の加算を 1(A) と表すと，Levinson-type のアルゴリズムによる計算量は $2n^2(M)+2n^2(A)$ であり，bordering method による計算量は $n^2(M)+n^2(A)$ である．

3　主結果

本節では，GLL における Hankel system を解く手法として

- 数式処理システム Mathematica の内部関数 LinearSolve
- Gauss の消去法
- Levinson-type のアルゴリズム

の 3 つをそれぞれ適用して実験を行い，その後に考察を行う．

3.1　実験結果

本節では，Hankel system を 3 つの手法によって解いた GLL の結果を載せる．実験に用いた計算機は，OS が Windows 10 Pro，CPU が Intel(R) Core (TM) i7-7500U CPU@2.70 GHz – 2.90 GHz，メモリが 16.0 GB である．また，数式処理システムは Mathematica 11.3 を用いる．実験では計算精度は 20 とし，多変数多項式の black-box は係数を -10 から 10 までのランダムな実数，指数を入力された指数の上界未満のランダムな非負整数として与える．このような設定の下で，100 個の多項式に対して，Hankel system を解く手法を変えた GLL をそれぞれ適用し，成功した回数と計算時間を調べる．実験結果は表 1 である．表 1

表 1: Hankel system を解く手法の比較．3 変数，指数の上界 $(100,100,100)$

	LinearSolve		Gauss の消去法		Levinson-type のアルゴリズム	
項数	計算時間 (秒)	成功回数	計算時間 (秒)	成功回数	計算時間 (秒)	成功回数
30	11.0	100	20.6	100	16.5	96
40	23.7	100	42.4	100	31.5	83
50	37.0	97	78.8	97	56.5	63
60	55.7	12	121.6	12	83.9	12

から，Gauss の消去法による計算と Levinson-type のアルゴリズムによる計算を比較すると，成功回数については前者に優位性があり，計算時間においては後者に優位性があることが分かる．

3.2 考察

本節では，前節の実験結果について，計算量が少ない Levinson-type のアルゴリズムが，Gauss の消去法よりも成功回数が少なくなった原因について考える．以下は，前節の実験における，Hankel system によって発生する誤差を示したものである．前節のものと同じ実験環境，実験設定で，100 個の多項式に対して，GLL における Hankel system を解き，その誤差を調べる．誤差は 1 つの Hankel system の解からはその最大値を取り出し，全体では 100 個のデータの最大値と相乗平均を表に表す．後者は，誤差の指数の相加平均を求めるために相乗平均を採用した．実験結果は表 2 である．表の値における注意として，「0.」は

表 2: Hankel system の解の誤差の比較．3 変数，指数の上界 $(100, 100, 100)$

項数	誤差	LinearSolve	Gauss の消去法	Levinson-type のアルゴリズム
30	相乗平均	$0. \times 10^{-10}$	$0. \times 10^{-10}$	$0. \times 10^{-9}$
	最大	5.8×10^{-2}	5.8×10^{-2}	1.0×10^{5}
40	相乗平均	$0. \times 10^{-8}$	$0. \times 10^{-8}$	$0. \times 10^{-6}$
	最大	$1. \times 10^{-6}$	$1. \times 10^{-6}$	3.4×10^{2}
50	相乗平均	$0. \times 10^{-6}$	$0. \times 10^{-6}$	$0. \times 10^{-4}$
	最大	1.3×10^{4}	1.3×10^{4}	5.1×10^{6}
60	相乗平均	0.	0.	0.
	最大	1.8×10^{8}	1.8×10^{8}	1.8×10^{8}

Mathematica において，表示できる有効数字がないことを表している．表 2 から，Gauss の消去法による計算と Levinson-type のアルゴリズムによる計算を比較すると，誤差の相乗平均，最大ともに後者の方が大きくなっている．これは Levinson-type のアルゴリズムにおいて必要な仮定である strong nonsingularity 性を GLL における Hankel 行列が常に満たすとは限らないためであると考えられる．Strong nonsingularity 性が仮定されない Levinson-type のアルゴリズムの計算では，各小行列 H_k が逆行列を持つことが保証されず，従って $\tau_k \neq 0$ が必ずしも満たされない．τ_k の計算において桁落ちが起きると，式 (3) より $\boldsymbol{x}_{k+1}$ 全体の精度に影響するため，この τ_k の桁落ちによって Levinson-type のアルゴリズムによる誤差が大きくなっているのではないかと予想される．以下では，τ_k の桁落ちと GLL の成功の関係を調べる実験を行う．先ほどと同じ実験環境，実験設定で，100 個の多項式に対して，GLL を適用する．

X : Gauss の消去法を適用した補間において失敗した多項式の集合

Y : Levinson-type のアルゴリズムを適用した補間において失敗した多項式の集合

Z : Levinson-type のアルゴリズムにおいて，τ_k による桁落ちが 10 桁以上となった状態で補間された多項式の集合

上記の 3 つの集合を用いて，τ_k の桁落ちと GLL の成功の関係を調べる．実験結果は表 3 である．表 3 より，τ_k による桁落ちが，Levinson-type のアルゴリズムを適用した補間が失敗する要因の 1 つになっていると考えられる．以上の考察から，GLL の Hankel system に適用する手法として，Levinson-type のアルゴリズムは適切ではないことがいえる．

表 3: τ_k と GLL の成功回数の関係．3 変数，指数の上界 $(100, 100, 100)$

項数	$\lvert X\rvert$	$\lvert Y\rvert$	$\lvert Z\rvert$	$\lvert (Y\setminus X)\rvert$	$\lvert (Z\setminus X)\rvert$	$\lvert (Y\setminus X)\cap(Z\setminus X)\rvert$
30	0	6	11	6	11	6
40	0	15	21	15	21	12
50	4	19	21	15	17	11
60	88	88	9	0	0	0

4 おわりに

今回，GLL における Hankel system を解く手法として，今まで用いていた LinearSolve に加え，Gauss の消去法と Levinson-type のアルゴリズムの 2 つを採用し，実験を行った．そして，計算量が少ない後者のアルゴリズムが，前者の手法よりも GLL の成功回数が低くなっている原因を調べた．今後の課題として，今回適用したもの以外の Hankel system を解く手法の実験や，Vandermonde system を解く手法の選択などが挙げられる．

謝　辞

本研究は科研費 18K11172 の助成を受けたものである．

参　考　文　献

[1] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. *20th Annual ACM Symp. Theory Comput.*, 301–309, New York, N.Y., 1988. ACM.

[2] M. Giesbrecht, G. Labahn, and W.-s. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.*, 44:943–959, 2009.

[3] G. Heinig, K. Rost. Fast algorithms for Toeplitz and Hankel matrices. *Linear Algebra and its Applications*, 435:1–59, 2011.