

1変数代数方程式の微小根の上界を用いた 近接根の計算

照井 章 (Akira Terui) *

筑波大学 数学系 (Institute of Mathematics, University of Tsukuba)

1 はじめに

1変数代数方程式が多重度 m の重根や近接根をもつときに、Durand-Kerner 法 ([2], [3]) などの数値解法で計算された近似値の重根・近接根からの誤差は大体 $O(\sqrt[m]{\varepsilon_M})$ 程度 (ε_M は計算機イプシロンを表す) になることが知られているが、これらの近似値に対して Smith の定理 [7] などを用いて計算される誤差上界は、はるかに過大な見積もりになることがあり、見積もりの精度を改善するためのいくつかの方法が提案されている (杉浦 [9] 等を参照)。

佐々木・照井 [8] は、方程式が原点付近に重根や近接根をもつときに、ある条件下でそれらの“微小根”的絶対値の上界や、それ以外の根の絶対値の下界を、方程式の係数から見積もる定理を与えた。本稿ではまず、この定理を応用し、1変数代数方程式の近接根の存在範囲の上界を計算する方法を述べる。

次に、近接根どうしの距離が比較的近い場合や、1個の近接根のまとまり (クラスタ) がさらに複数個の異なる重根・近接根から成るときは、上述の定理をただちに適用できないことがある。本稿では、そのような近接根をもつ多項式因子を与方程式から精度よく分離し、分離した因子に変数変換を行って各根間の距離を“拡大”し、近接根の存在範囲の上界を再帰的に計算する。

近接根をもつ多項式因子の分離には、尾崎・佐々木 [4] による因子分離法を用いるが、この因子分離法は拡張 Euclid 互除法を用いているので、因子の次数が大きいと、因子を精度よく分離するのが困難な場合がある。本稿では、拡張 Euclid 互除法に代えて、因子の係数に関する連立 1 次方程式を数値解法で解くことにより、次数が大きな因子を精度よく分離する方法について述べる。

2 1変数代数方程式の微小根の上界と近接根の計算への応用

2.1 1変数代数方程式の微小根の上界

$P(x)$ を式 (1) で与えられる複素係数 1変数多項式とし、代数方程式 $P(x) = 0$ を考える。

$$P(x) = c_n x^n + \cdots + c_{m+1} x^{m+1} + x^m + \varepsilon_{m-1} x^{m-1} + \cdots + \varepsilon_0, \quad c_n \neq 0. \quad (1)$$

このとき、 ε_j ($j = 0, \dots, m-1$) が c_j ($j = m+1, \dots, n$) に比べて十分小さければ、方程式 $P(x) = 0$ の根の中で絶対値の小さい方から m 個の根は、複素平面上原点の近傍に存在すると考えられる。このような場合の“微小根”的絶対値の上界と、それ以外の(原点から十分離れた位置に存在する)根の絶対値の下界は、次の定理によって与えられる(証明は佐々木・照井 [8] を参照)。

*terui@math.tsukuba.ac.jp

定理 1

e を正数とし、式 (1) における $P(x)$ の係数が次の条件を満たすとする。

$$\begin{cases} \max\{|c_n|, \dots, |c_{m+1}|\} = 1, \\ e = \max\{\sqrt[1]{|\varepsilon_{m-1}|}, \sqrt[2]{|\varepsilon_{m-2}|}, \dots, \sqrt[m]{|\varepsilon_0|}\} \ll 1. \end{cases} \quad (2)$$

また、方程式 $P(x) = 0$ の n 個の根を ζ_1, \dots, ζ_n とし、次式の大小関係が成り立つものとする。

$$|\zeta_1| \leq \dots \leq |\zeta_m| < |\zeta_{m+1}| \leq \dots \leq |\zeta_n|. \quad (3)$$

このとき、 $e \leq 1/9$ ならば、 $|\zeta_m|$ と $|\zeta_{m+1}|$ に対して次の不等式が成り立つ。

$$|\zeta_m| < \frac{1+3e}{4} \left[1 - \sqrt{1 - \frac{16e}{(1+3e)^2}} \right], \quad |\zeta_{m+1}| > \frac{1+3e}{4} \left[1 + \sqrt{1 - \frac{16e}{(1+3e)^2}} \right]. \quad (4)$$

上記の上界と下界は次式のように有理式で表すこともできる。

$$|\zeta_m| < 2e \cdot \left[\frac{1}{1+3e} + \frac{16e}{(1+3e)^3} \right], \quad |\zeta_{m+1}| > \frac{1}{2} \left[1 - \frac{e(1-9e)}{1+3e} - \frac{64e^2}{(1+3e)^3} \right]. \quad (5)$$

2.2 定理 1 を用いた近接根の存在範囲の計算

$F(x)$ を式 (6) で与えられる複素係数 1 变数多項式とする。

$$\begin{aligned} F(x) &= x^n + a_{n-1}x^{n-1} + \dots + a_0 \\ &= (x - \alpha_1) \cdots (x - \alpha_n). \end{aligned} \quad (6)$$

ここで、方程式 $F(x) = 0$ の根のうち $\alpha_1, \dots, \alpha_m$ が近接根であるとする。本節では、このような方程式 $F(x) = 0$ に対し、上述の定理 1 を用いて近接根の存在範囲の上界を計算することを考える。そのためには、基本的に次のような方法をとればよい。

アルゴリズム 1 (微小根の上界を用いた近接根の存在範囲の上界の計算)

[Step 1] 方程式 $F(x) = 0$ の根の近似値を数値解法で求め、 $\alpha_1, \dots, \alpha_n$ に対する近似値をそれぞれ $\bar{\alpha}_1, \dots, \bar{\alpha}_n$ とおく。同時に、 $\bar{\alpha}_1, \dots, \bar{\alpha}_n$ の誤差上界を求める(誤差上界の計算には Smith の誤差上界 [7] 等を用いる)。ここでは、 $\bar{\alpha}_1, \dots, \bar{\alpha}_m$ が m 個の近接根の近似値として検出されたものとする。 $\bar{\alpha} = (\bar{\alpha}_1 + \dots + \bar{\alpha}_m)/m$ とおく。

[Step 2] $F(x)$ に対し、式 (7) による変数変換を行った式 $\bar{F}(x)$ を求める。

$$\bar{F}(x) = F(x + \bar{\alpha}) = x^n + \dots + \bar{a}_m x^m + \bar{a}_{m-1} x^{m-1} + \dots + \bar{a}_0. \quad (7)$$

このとき、方程式 $\bar{F}(x) = 0$ は複素平面上の原点付近に m 個の近接根をもつこと、 $\bar{F}(x)$ の $m-1$ 次以下の係数は高次の係数に比べて微小になっていることに注意されたい。

[Step 3] 方程式 $\bar{F}(x) = 0$ に対して定理 1 を適用する。得られた原点付近の微小根の上界を β とおく。

[Step 4] Step 3 より $\{x \in \mathbb{C} \mid |x - \bar{\alpha}| < \beta\}$ が与方程式 $F(x) = 0$ の $x = \bar{\alpha}$ 付近の近接根の存在範囲となる。■

上記 Step 3 において、 $\bar{F}(x)$ が式 (1) の形でない場合は定理 1 を適用できないが、以下に述べる手順で変数変換を行うことにより、 $\bar{F}(x)$ を式 (1) の形に変換することができる。

[Step 3a] $\bar{F}(x)$ を $1/\bar{a}_m$ 倍した多項式を $\hat{F}(x)$ とおく。

$$\hat{F}(x) = (1/\bar{a}_m)\bar{F}(x) = \hat{a}_n x^n + \cdots + \hat{a}_{m+1} x^{m+1} + x^m + \hat{a}_{m-1} x^{m-1} + \cdots + \hat{a}_0. \quad (8)$$

[Step 3b] $\hat{a} = \max\{\sqrt[n-m]{|\hat{a}_n|}, \dots, \sqrt[1]{|\hat{a}_{m+1}|}\}$ とおく。

[Step 3c] $\hat{F}(x)$ に対し、Step 3b の \hat{a} を用いて式 (9) による変数変換を行った式 $\tilde{F}(x)$ を求める。

$$\begin{aligned} \tilde{F}(x) &= \hat{a}^m \cdot \hat{F}(x/\hat{a}) \\ &= (\hat{a}_n/\hat{a}^{n-m})x^n + \cdots + (\hat{a}_{m+1}/\hat{a})x^{m+1} + x^m \\ &\quad + \hat{a}_{m-1}\hat{a}x^{m-1} + \cdots + \hat{a}_0\hat{a}^m \\ &= \tilde{a}_n x^n + \cdots + \tilde{a}_{m+1} x^{m+1} + x^m + \tilde{a}_{m-1} x^{m-1} + \cdots + \tilde{a}_0. \end{aligned} \quad (9)$$

このとき、 $\tilde{F}(x)$ は式 (1) の形をしており、式 (2) の係数条件を満たしていることに注意されたい。

[Step 3d] 方程式 $\tilde{F}(x) = 0$ に対して定理 1 を適用する。得られた原点付近の微小根の上界を $\tilde{\beta}$ とおく。

[Step 4'] Step 3d より $\{x \in \mathbb{C} \mid |x - \bar{\alpha}| < (\tilde{\beta}/\hat{a})\}$ が与方程式 $F(x) = 0$ の $x = \bar{\alpha}$ 付近の近接根の存在範囲となる。■

3 アルゴリズム 1がうまく適用できない場合

本章以降では、 $F(x)$ がより一般的な形で与えられた場合を考える。

$$\begin{aligned} F(x) &= x^n + a_{n-1}x^{n-1} + \cdots + a_0 \\ &= (x - \alpha_{1,1}) \cdots (x - \alpha_{1,m_1})(x - \alpha_{2,1}) \cdots (x - \alpha_{2,m_2}) \cdots \\ &\quad \cdots (x - \alpha_{k,1}) \cdots (x - \alpha_{k,m_k}), \\ m_1 + \cdots + m_k &= n, \quad 1 \leq m_j \leq n. \end{aligned} \quad (10)$$

ここで $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ ($j = 1, \dots, k$) はそれぞれ单根 ($m_j = 1$) もしくは多重度 m_j の重根か近接根とし、 $F(x)$ は k 個の異なる单根/重根/近接根をもつとする。このとき、数値解法を用いて重根・近接根の多重度 m_j を正確に計算できたとしても、アルゴリズム 1 を適用して重根・近接根の存在範囲を計算できるとは限らない。なぜなら、定理 1 における e の値が $e \leq 1/9$ を満たさなければ、原点付近に移動した重根・近接根の絶対値の上界を見積もることができないからである。

定理 1 における e の値が $e \leq 1/9$ を満たさないような状況は、次の場合に起こり得る。

- 重根・近接根 $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ と他の重根・近接根の距離が比較的近い。
- $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ がさらにいくつかの重根・近接根に分離される。

そこで、我々は、次の方針に従い、重根・近接根の存在範囲の上界を再帰的に計算することを考える。

- $F(x)$ の因子のうち、 $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ を根としてもつ因子を分離する。分離した因子を $\bar{F}_j(x)$ とおく。
- 分離した因子 $\bar{F}_j(x)$ に対し、 $\hat{F}_j(x) = a^{m_j} \cdot \bar{F}_j(ax)$ ($a \ll 1$) なる変数変換を適用させ、座標軸を“拡大”させた多項式 $\hat{F}_j(x)$ を求める。

3. $\hat{F}_j(x)$ に対してアルゴリズム 1 を適用させ、重根・近接根の存在範囲の上界を計算する。

上記と同様に重根・近接根を再帰的に分離する方法は、佐々木・野田 [6] によって提案されている。(佐々木・野田は、重根・近接根の分離に Euclid 互除法に基づく近似無平方分解を用いている。)

4 近接根をもつ因子の分離と因子分離法の改善

本章では、 $F(x)$ の因子から、重根もしくは近接根である $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ を根としてもつ因子 $F_j(x) = (x - \alpha_{j,1}) \cdots (x - \alpha_{j,m_j})$ を分離する方法について述べる。数値解法で求めた $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ の近似値をそれぞれ $\bar{\alpha}_{j,1}, \dots, \bar{\alpha}_{j,m_j}$ とおくと、これらの精度は $\sqrt[m]{\varepsilon_M}$ (ε_M は計算機イプシロン) 程度なので、 $(x - \bar{\alpha}_{j,1}) \cdots (x - \bar{\alpha}_{j,m_j})$ は $F_j(x)$ を精度よく近似しているとは言い難く、何らかの方法で $F_j(x)$ に “より近い” 因子 $\bar{F}_j(x)$ を計算する必要がある。

そこで、我々は、尾崎・佐々木 [4] による因子分離法を用いる。因子分離法の概要は次の通りである。 F, G, H を一般に複素係数 1 变数多項式とし、 G と H は互いに素で $F = GH$ なる関係を満たすとする。そして、 G と H は未知であるが、 G, H に “近い” F の因子としてそれぞれ G_j, H_j が与えられているとする。因子分離法は、以下の手順により G, H に “より近い” F の因子 G_{j+1}, H_{j+1} を構成する反復算法である(詳細は文献 [4] を参照)

アルゴリズム 2 (因子分離法: G_j, H_j から G_{j+1}, H_{j+1} を構成するステップ)

$lc(F) = lc(G_j) = lc(H_j) = 1$ とする。($lc(F)$ は F の主係数を表す。以下同様)

[Step 1] $\Delta_j = F - G_j H_j$ とおく。

[Step 2] 拡張 Euclid 互除法を用いて、式 (11) を満たす A_j, B_j を求める。($\deg(A_j)$ は A_j の次数を表す。以下同様)

以下同様)

$$\begin{cases} A_j G_j + B_j H_j = 1, \\ \deg(A_j) < \deg(H_j), \quad \deg(B_j) < \deg(G_j). \end{cases} \quad (11)$$

[Step 3] $U_j = \text{rem}(\Delta_j B_j, G_j), V_j = \text{rem}(\Delta_j A_j, H_j)$ とおく。($\text{rem}(\Delta_j B_j, G_j)$ は $\Delta_j B_j$ を G_j で割った多項式剰余を表す。以下同様)

[Step 4] $G_{j+1} = G_j + U_j, H_{j+1} = H_j + V_j$ とおく。■

尾崎・佐々木 [4] によると、 G, H を近似する因子 G_0, H_0 が与えられたときに、 G の零点と H の零点が十分離れていれば、 G_0, H_0 にアルゴリズム 2 を繰り返し適用させることにより得られる G_j, H_j はそれぞれ G, H に 2 次収束する。

ところが、我々が尾崎・佐々木 [4] の実験例よりも次数の高い多項式 ($\deg(F) = 30, \deg(G) = 10, \deg(H) = 20$ 程度) に対して因子分離法を適用させたところ、 G と H が互いに素であるにもかかわらず、アルゴリズム 2 を適用させても因子 G_{j+1}, H_{j+1} の精度が改善されない例が見られた。このような例の計算過程を検討した結果、 G_{j+1}, H_{j+1} の精度向上が妨げられている原因として、アルゴリズム 2 の Step 2 および 3 における多項式剰余(列)の計算の結果、修正量 U_j, V_j の係数精度が低下した可能性が考えられた。

このような問題の対策として、これまでに、多項式剰余列を安定に生成させるための方法(例えば大迫・杉浦・鳥居 [10] を参照)が提案されている。一方で、問題を連立 1 次方程式の解法に帰着させ、安定した数値解法を用いて解を精度よく計算する方法も提案されている(例えば Pan は 1 变数多項式の近似 GCD 計算を Toeplitz 行列の解法に帰着させる算法を提案している [5])。修正量 U_j, V_j の係数は、 $G_j, H_j, U_j, V_j, \Delta_j$ の係数に関する連立 1 次方程式を解くことにより求まる。

$G_j, H_j, U_j, V_j, \Delta_j$ を式 (12) のように表す。

$$\begin{cases} G_j(x) = x^k + g_{k-1}x^{k-1} + \cdots + g_0 \\ H_j(x) = x^{n-k} + h_{n-k-1}x^{n-k-1} + \cdots + h_0 \end{cases}, \begin{cases} U_j(x) = u_{k-1}x^{k-1} + \cdots + u_0 \\ V_j(x) = v_{n-k-1}x^{n-k-1} + \cdots + v_0 \end{cases}, \Delta_j(x) = \delta_{n-1}x^{n-1} + \cdots + \delta_0. \quad (12)$$

すると、 U_j, V_j の各係数 u_l, v_l は連立 1 次方程式 (13) を満たす。

$$\begin{pmatrix} 1 & & & & 1 \\ g_{k-1} & \ddots & & & h_{n-k-1} & \ddots \\ \vdots & \ddots & 1 & & \vdots & \ddots & 1 \\ \vdots & \ddots & g_{k-1} & & \vdots & \ddots & h_{n-k-1} \\ g_0 & \ddots & \vdots & h_0 & \ddots & \vdots & \\ \ddots & \vdots & & & \ddots & \vdots & \\ & g_0 & & & & h_0 & \end{pmatrix} \begin{pmatrix} v_{n-k-1} \\ v_{n-k-2} \\ \vdots \\ v_0 \\ u_{k-1} \\ u_{k-2} \\ \vdots \\ u_0 \end{pmatrix} = \begin{pmatrix} \delta_{n-1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \delta_0 \end{pmatrix}. \quad (13)$$

よって、アルゴリズム 2 の Step 2 および 3 を次の計算で置き換える。

[Step 2'] $G_j, H_j, U_j, V_j, \Delta_j$ の係数に関する連立 1 次方程式を解くことにより、 U_j および V_j の係数を求め、 U_j, V_j を構成する。■

我々が行った実験では、NSL (Nara Standard Lisp, SPARC アセンブリ言語版) 上で動作する数式処理システム GAL (General Algebraic Language/Laboratory) に実装された因子分離法のプログラムを用いているが、今回、上記 Step 2' の部分を CLAPACK 3.0 [1] を用いて実装を行い、これまでの因子分離法のプログラムに加えた。その結果、 $\deg(F) = 30, \deg(G) = 10, \deg(H) = 20$ 程度の例で、従来の因子分離法では分離精度 (残差 Δ_j のノルム) が 10^{-3} 以上でしか分離できなかったものが、今回一部変更した因子分離法により、分離精度 10^{-16} 程度 (ε_M と同程度) で分離できるようになった。

以上より、 $F(x)$ から $\alpha_{j,1}, \dots, \alpha_{j,m_j}$ を根としてもつ因子 $F_j(x)$ の近似因子 $\bar{F}_j(x)$ の分離は次の手順にしたがって行う。

アルゴリズム 3 (因子分離法を用いた $\bar{F}_j(x)$ の分離)

[Step 1] $G_0(x), H_0(x)$ を次式で定義する。

$$G_0(x) = (x - \bar{\alpha}_{j,1}) \cdots (x - \bar{\alpha}_{j,m_j}), \quad H_0(x) = \prod_{i=1, i \neq j}^k (x - \bar{\alpha}_{i,1}) \cdots (x - \bar{\alpha}_{i,m_i}). \quad (14)$$

[Step 2] $F(x)$ に対し $G_0(x), H_0(x)$ を初期因子としてアルゴリズム 2 を繰り返し適用させる。

[Step 3] アルゴリズム 2 を l 回実行後、残差 $\Delta_l(x)$ のノルムが十分小さくなつたところでアルゴリズム 2 を終了し、 $\bar{F}_j(x) = G_l(x)$ とおく。■

謝辞 本稿をまとめるにあたり、有益な助言を下さった佐々木建昭教授に感謝します。

参 考 文 献

- [1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, third edition, 1999.
- [2] E. Durand. *Solutions Numériques des Équations Algébriques*, Tome I. Masson, Paris, 1960.
- [3] I. O. Kerner. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math.*, Vol. 8, pp. 290–294, 1966.
- [4] Y. Ozaki and T. Sasaki. Univariate Factor Separation and Separation of Multiple/Close Factors. *数式処理*, Vol. 6, No. 4, pp. 18–34, September 1998.
- [5] V. Y. Pan. Univariate polynomials: Nearly optimal algorithms for factorization and rootfinding. In *Proceedings of ISSAC 2001*, pp. 253–267. ACM Press, 2001.
- [6] T. Sasaki and M.-T. Noda. Approximate Square-free Decomposition and Root-finding of Ill-conditioned Algebraic Equations. *J. Inform. Process.*, Vol. 12, No. 2, pp. 159–168, 1989.
- [7] B. T. Smith. Error Bounds for Zeros of a Polynomial Based Upon Gershgorin's Theorems. *J. ACM*, Vol. 17, No. 4, pp. 661–674, October 1970.
- [8] A. Terui and T. Sasaki. “Approximate zero-points” of real univariate polynomial with large error terms. *情報処理学会論文誌*, Vol. 41, No. 4, pp. 974–989, April 2000.
- [9] 杉浦洋. Durand-Kerner 法と根の精度保証. 第 28 回数值解析シンポジウム講演予稿集, pp. 57–60, June 1999.
- [10] 大迫尚行, 杉浦洋, 鳥居達生. 多項式剩余列の安定な拡張算法. 日本応用数理学会論文誌, Vol. 7, No. 3, pp. 227–255, September 1997.