

新たな条件式の導入による多変数連立代数方程式の解法

沢 田 浩 之†

多変数連立代数方程式の解法においては、数値解に含まれる誤差の評価方法が重要な問題となる。もしも誤差が適切に評価されなければ、数値解として真の解の近似値が得られないばかりか、真の解とは無縁なものも数値解として得られてしまうこともありうる。本論文では、新たな条件式を導入することによって誤差を適切に評価して数値解を得る解法を提案する。本解法の利点は、真の解の近似値をすべて求めるために許容誤差が満足しなければならない条件を明らかにできる点にある。すなわち、従来の解法でも許容誤差を十分に小さく設定すれば真の解の近似値をすべて求めることは可能であるが、そのために許容誤差が満足しなければならない条件は明らかではなく、したがって得られた数値解が真の解の近似値であることは保証されない。これに対して本解法では、真の解の近似値をすべて求めるために許容誤差が満足しなければならない条件がその計算過程において明らかとなる。必要に応じて設定許容誤差を自動的に更新することにより得られた数値解が真の解の近似値であることを保証できる。なお、本論文では解の個数は有限個であるとし、解として実数解のみを対象としているが、変数を実部と虚部に分離することによってすべての複素数解を求めることも可能である。

Solution of Algebraic Equation Systems by Introducing New Equations

HIROYUKI SAWADA†

In the solution of algebraic equation systems, the estimation of numerical errors included in numerical solutions is very important. If numerical errors are not estimated properly, then approximate values to exact solutions may not be obtained or values that do not correspond to any exact solution may be obtained as numerical solutions. This paper proposes a new method of solving algebraic equation systems by introducing new equations to properly evaluate numerical errors. This new method has the merit of clarifying conditions that allowable margins of errors must satisfy to get all the approximate values to exact solutions. In conventional methods, all the approximate values can be found by assuming small enough allowable margins. However, since conditions satisfied by the allowable margins are not clarified, the obtained numerical solutions are not sure to be the approximate values. On the contrary, this new method clarifies those conditions in the process of computation. Thus, obtained numerical solutions are guaranteed to be the approximate values by resetting allowable margins automatically if necessary. In this paper, the given algebraic equation system is assumed to have a finite number of solutions, and the method of obtaining real solutions is discussed.

1. はじめに

多変数連立代数方程式の解法についてはこれまでに多くの研究がなされてきており、様々な解法が提案されている。現在、それらの中ではグレブナ基底による方法が比較的大きな系に対して有効であるとされている³⁾。

従来用いられている方法は、グレブナ基底を用いてある変数についての1変数方程式を導き、その変数についての数値解を求め、それを元の方程式に代入して

変数一つずつ消去することによって解を求めるというものである³⁾。しかしながら、この方法ではある変数についての数値解をもとにして他の変数の解を求めると、誤差が累積されることになる。

誤差の累積を避けるためには、すべての変数についての1変数方程式を導いてそれぞれの変数についての数値解を求め、それらを組み合わせて解を得るという方法が考えられる。この方法を用いた場合、組み合わせによって得られた解の候補点か、元の連立方程式の正当な解であるかどうかを検証する方法が重要な問題となる。数値解には誤差が含まれているために、たとえそれが真の解の近似値であっても元の方程式を完全に満足することはまれである。すべての計算を数式処

† 機械技術研究所
Mechanical Engineering Laboratory

理で行う場合であっても、数値解は有理数として表現されるため、無理数解に含まれる誤差を取り除くことはできない。数値解に含まれるこれらの誤差が、いわゆる「最悪のケース」を考慮して、最大限に見積もられてしまうと、数値解として真の解の近似値を求めることはできるものの、真の解とは無縁なものも近似値として判定してしまう可能性がある⁵⁾。すなわち、この方法を用いる場合には、解の候補点の検証を完全に正しく行う方法が不可欠であると言える。

本論文では、候補点の検証を完全に正しく行う方法として外接球法を提案する。この方法では、まず、与えられた連立代数方程式の数値解の候補点の集合を構成し、各候補点 \mathbf{a} について \mathbf{a} を中心とする半径 r の球を導入する。与えられた連立代数方程式と連立させると、変数 r は候補点 \mathbf{a} と連立代数方程式の真の解との距離に等しくなる。したがって、変数 r の値を評価することによって \mathbf{a} が連立代数方程式の数値解、すなわち真の解の近似値であるか否かを判定することが可能である。本解法では、この判定を正しく行うために許容誤差が満足しなくてはならない条件が、計算の過程で明らかにされるので、得られた数値解が真の解の近似値であることが保証される。また、誤差が累積されることもないので、各数値解に含まれる誤差は設定された許容誤差未満となる。

本論文の構成は以下のとおりである。まず第2章において本論文で提案する解法の基本的な概念について述べ、その定式化を行う。次に、第3章において、与えられた連立代数方程式が複素数解を持つ場合の解法について述べる。第4章では、数値解の候補点集合の構成方法について説明し、最後に第5章では実際の計算機上での実行結果を示し、さらに他の方法との比較を行う。なお、本論文では、与えられた連立代数方程式の解は有限個であると仮定している。

2. 定式化

本節では、実数解のみを持つ連立代数方程式を対象として、本論文で提案する解法の定式化を行う。

与えられた実数係数の連立代数方程式を式 (1) とする。

$$f_1(x_1, \dots, x_n) = \dots = f_m(x_1, \dots, x_n) = 0. \tag{1}$$

方程式 (1) の解が有限個であることは、そのグレブナ基底を計算し、グレブナ基底を構成する各方程式の最大単項を調べることによって検証される²⁾。次に各変数についての1変数方程式 (2) を計算する。1変数方程式の計算方法はいくつかあり、その中では全次数順

序で計算されたグレブナ基底をもとに各1変数方程式の係数を決定する方法が効率的であるとされる²⁾。

$$g_1(x_1) = \dots = g_n(x_n) = 0. \tag{2}$$

各多項式 $g_j(x_j)$ のストルム関数列¹⁾を計算する。各変数の数値解についての許容誤差 $\epsilon_1, \dots, \epsilon_n$ を与えると、ストルム関数列を用いた2分法により、方程式 (2) の数値解 a_j が求められる。また、方程式 (1) の解がすべて実数であることは、式 (2) の解がすべて実数であることによって確認される。ここで許容誤差 ϵ_j の満足すべき条件について考察する。本解法では、方程式 $g_j(x_j) = 0$ の真の解と数値解 a_j がある許容誤差 ϵ_j のもとで1対1に対応するように ϵ_j を設定する。このためには区間 $(a_j - \epsilon_j, a_j + \epsilon_j)$ には方程式 $g_j(x_j) = 0$ の真の解がただ一つだけ存在しなければならない。したがって許容誤差 ϵ_j は次式 (3) を満足する必要がある。

$$\forall (a_j \neq a'_j) \{ |a_j - a'_j| \geq 2\epsilon_j \}. \tag{3}$$

ただし、 a_j および a'_j はいずれも方程式 $g_j(x_j) = 0$ の数値解である。許容誤差 ϵ_j が式 (3) を満足しない場合には ϵ_j を設定し直す必要があるが、本解法ではストルム関数列を用いた2分法によって数値解の計算を行っているので、許容誤差 ϵ_j を数値解の計算過程において式 (3) を満足するように自動的に再設定することが可能である。

このようにして得られた方程式 (2) の数値解をすべて組み合わせることで方程式 (1) の数値解の候補点を構成する。各候補点 $\mathbf{a} = (a_1, \dots, a_n)$ について方程式 (1) の真の解が存在する可能性のある領域 R は、次式 (4) によって与えられる。

$$R = \{ (x_1, \dots, x_n) \mid |x_j - a_j| < \epsilon_j, 1 \leq j \leq n \}. \tag{4}$$

領域 R に存在する真の解の個数は高々一つである。領域 R に真の解が存在すれば、候補点 \mathbf{a} は真の解の近似値であり、存在しない場合には真の解とは無縁のものである。

ここで図1に示すような領域 R の外接球 S を考える。また、領域 R に含まれない点の集合を領域 \bar{R}

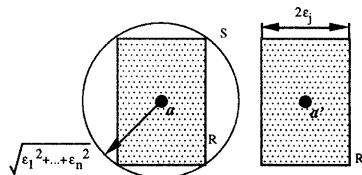


図1 領域 R の外接球 S

Fig. 1 Circumscribed sphere S of region R.

とする.

$$S = \left\{ (x_1, \dots, x_n) \mid \sum_{j=1}^n (x_j - a_j)^2 < \sum_{j=1}^n \epsilon_j^2 \right\},$$

$$\bar{R} = \{ (x_1, \dots, x_n) \mid \exists(j) |x_j - a_j| \geq \epsilon_j \}. \quad (5)$$

領域 $\bar{R} \cap S$ に真の解が存在しないものとする. 領域 R に真の解が存在するか否かは, 外接球 S に真の解が存在するか否かを調べることによって判定することができる. 図 1 に示すように, 領域 $\bar{R} \cap S$ に真の解が存在しないという条件は次式 (6) によって満足される.

$$\forall(j) \forall(a_j \neq a'_j) \left\{ |a_j - a'_j| \geq \epsilon_j + \sqrt{\epsilon_1^2 + \dots + \epsilon_n^2} \right\}. \quad (6)$$

ただし, a_j および a'_j はともに方程式 $g_j(x_j) = 0$ の数値解である. なお, 式 (6) は領域 $\bar{R} \cap S$ に真の解が存在しないための十分条件に過ぎないが, 式 (6) の成立は方程式 (2) の数値解を求めさえすれば判定可能であるので, 本論文ではこれを採用している. このとき, 式 (6) が成立しない場合には, ϵ_j を再設定しなくてはならないが, その方法については後述する.

次に, 外接球 S に真の解が存在するか否かを調べるために, 候補点 a を中心とする半径 r の球を示す式 (7) を導入する.

$$\sum_{j=1}^n (x_j - a_j)^2 = r^2. \quad (7)$$

ここで, 連立方程式 (1)∩(7) の真の解 (x_1, \dots, x_n, r) の持つ意味について考察する.

変数 x_1, \dots, x_n, r に $x_1 < \dots < x_n < r$ なる辞書式順序を与え, その辞書式順序における方程式 (1) のグレブナ基底を GB_0 とする. このとき, GB_0 に式 (7) をつけ加えた GB_1 は方程式 (1)∩(7) の辞書式順序の正規化もしくは非正規化グレブナ基底である.

$$GB_1 = GB_0 \cup \left\{ \sum_{j=1}^n (x_j - a_j)^2 = r^2 \right\}. \quad (8)$$

GB_1 は辞書式順序のグレブナ基底であるので, その真の解 (x_1, \dots, x_n, r) は理論上 x_1 から r まで逐次求められる²⁾. すなわち, r は方程式 (1) の真の解 (x_1, \dots, x_n) を用いて計算されることになる. 変数 x_1, \dots, x_n の値はすべて実数値であるので, 式 (7) により, r^2 は候補点 a と方程式 (1) の真の解 (x_1, \dots, x_n) との距離の自乗を示す. したがって, 次式 (9) を満足する r が存在すれば, 外接球 S に方程式 (1) の真の解 (x_1, \dots, x_n) が存在することになり, 候補点 a は

方程式 (1) の数値解, すなわち真の解の近似値であると結論される. 次式 (9) を満足する r が存在しない場合には候補点 a は方程式 (1) の数値解ではない.

$$0 \leq r^2 < \sum_{j=1}^n \epsilon_j^2. \quad (9)$$

式 (9) を満足する r の存在は, 連立方程式 (1)∩(7) から r についての 1 変数方程式を導き, そのスツルム関数列を計算し, $r = 0$ と $r = \sum_{j=1}^n \epsilon_j^2$ における符号反転回数を調べることによって検証できる.

以上に述べた方法を定式化したものが図 2 であり, この方法を外接球法と呼ぶことにする. 図 2 では, 方程式 (1) は多項式集合 F として与えられ, 最後に数値解集合 N が返される.

許容誤差 ϵ_j が満足すべき条件は式 (3)∩(6) であるが, 式 (6) が満足されれば式 (3) は自動的に満足されるので, 条件式として必要となるのは式 (6) のみであ

```

(1) input x := (x1, ..., xn)
    % x1, ..., xn are input variables.
(2) input F := {f1(x), ..., fm(x)}
    % fj(x) is an input polynomials.
(3) input E := {epsilon1, ..., epsilon_n}
    % epsilon_j is an allowable margin of error for x_j.
(4) for j = 1 to n
(5)   gj(xj) := UniVariatePolynomial(xj, F)
    % gj(xj) is a uni-variate polynomial.
(6)   Aj := NumericalSolutions(gj(xj), epsilon_j)
    % Aj is a set of numerical solutions.
(7) endfor
(8) if exists(j) exists(a_j != a'_j, a_j in Aj, a'_j in Aj)
    { |a_j - a'_j| < epsilon_j + sqrt(sum_{j=1}^n epsilon_j^2) } then
(9)   E := Reset(E) % Reset epsilon_j.
(10)  for j = 1 to n
(11)    Aj := NumericalSolutions(gj(xj), epsilon_j)
(12)  endfor
(13) endif
(14) A := {(a1, ..., an) | a_j in Aj, 1 <= j <= n}
(15) N := empty set
(16) while A != empty set
(17)   (a1, ..., an) := FirstElement(A)
    % (a1, ..., an) is the first element of A.
(18)   A := A \ {(a1, ..., an)}
(19)   f := sum_{j=1}^n (x_j - a_j)^2 - r^2
(20)   g(r) := UniVariatePolynomial(r, F union {f})
(21)   if exists(r) {g(r) = 0 and 0 <= r^2 < sum_{j=1}^n epsilon_j^2} then
    % This condition is checked by a Sturm
    % sequence of g(r).
(22)     N := N union {(a1, ..., an)}
(23)   endif
(24) endwhile
(25) return N
    
```

図 2 外接球法

Fig. 2 Circumscribed sphere method.

る。式(6)が満足されない場合には、許容誤差 ϵ_j は改めて設定し直す必要がある。このとき、次式(10)を満足する ϵ を計算し、 $\forall(j)\{\epsilon_j \leq \epsilon\}$ となるように ϵ_j を設定すれば、式(6)が満足されることが保証される。式(10)を満足する ϵ は方程式(2)の数値解を計算する過程において自動的に求めることが可能である。 $\forall(j)\{\epsilon_j \leq \epsilon\}$ は式(6)が満足されるための十分条件に過ぎないが、各許容誤差 ϵ_j の満足すべき条件が陽に与えられるので、実際のインプリメントにおいてはこの条件を用いて許容誤差 ϵ_j を自動再設定している。

$$\forall(j)\forall(a_j \neq a'_j)\{|a_j - a'_j| \geq (1 + \sqrt{n})\epsilon\}. \quad (10)$$

3. 複素数解が存在する場合の解法

前章では、方程式(1)は実数解のみを持つと仮定しており、その場合には図2に示す外接球法ですべての解を得ることができる。しかしながら、方程式(1)が実数解だけでなく複素数解も持つ場合には、単純に外接球法を適用すると真の解とは無縁なものを数値解として求めてしまう可能性がある。本章ではその問題を明らかにするとともに、その対処方法について述べる。

3.1 複素数解が存在する場合の問題点

一般に複素数解は α_j および β_j を実数として式(11)のように表される。

$$(x_1, \dots, x_n) = (\alpha_1 + i\beta_1, \dots, \alpha_n + i\beta_n). \quad (11)$$

式(11)を式(7)に代入すると式(12)が得られる。

$$\sum_{j=1}^n [(\alpha_j - a_j)^2 - \beta_j^2] + 2i \sum_{j=1}^n (\alpha_j - a_j)\beta_j = r^2. \quad (12)$$

次式(13)を成立させ、かつ、 $\exists(j)\{\beta_j \neq 0\}$ であるような複素数解が方程式(1)の解として存在する場合には式(12)の左辺は実数となるため、候補点 \mathbf{a} が方程式(1)の真の解の近似値ではないにもかかわらず、式(9)を満足する r が存在する可能性があり、真の解とは無縁なものが外接球法によって求められてしまうおそれがある。

$$\sum_{j=1}^n (\alpha_j - a_j)\beta_j = 0. \quad (13)$$

真の解とは無縁なものを排除するために、方程式(1)が複素数解を持つ場合には、各候補点 \mathbf{a} について式(13)を満足し、かつ、 $\exists(j)\{\beta_j \neq 0\}$ であるような複素数解は存在しないことを確認しておく必要がある。そのような複素数解が存在しないことは以下のようにして

確認できる。方程式(1)に式(11)を代入し、各方程式 $f_j = 0$ を実部と虚部に分けて変数 $(\alpha_1, \beta_1, \dots, \alpha_n, \beta_n)$ からなる新たな連立方程式(14)を作成する。

$$\begin{aligned} \operatorname{Re}[f_1(\alpha_1 + i\beta_1, \dots, \alpha_n + i\beta_n)] &= 0, \\ \operatorname{Im}[f_1(\alpha_1 + i\beta_1, \dots, \alpha_n + i\beta_n)] &= 0, \\ &\vdots \\ \operatorname{Re}[f_m(\alpha_1 + i\beta_1, \dots, \alpha_n + i\beta_n)] &= 0, \\ \operatorname{Im}[f_m(\alpha_1 + i\beta_1, \dots, \alpha_n + i\beta_n)] &= 0. \end{aligned} \quad (14)$$

ただし、 $\operatorname{Re}[f_j(\mathbf{x})]$ および $\operatorname{Im}[f_j(\mathbf{x})]$ はそれぞれ $f_j(\mathbf{x})$ の実部および虚部を表すものとする。方程式(14) \cap (13)が $\exists(j)\{\beta_j \neq 0$ (β_j は実数)} である解を持たないことを示せばよい。そのために、まず、方程式(14) \cap (13)のグレブナ基底を計算する。 $\forall(j)\{\beta_j^{b_j} = 0$ (b_j は自然数)} がグレブナ基底に含まれていれば、方程式(14) \cap (13)は $\exists(j)\{\beta_j \neq 0$ (β_j は実数)} である解を持たない。

方程式(14) \cap (13)のグレブナ基底が $\forall(j)\{\beta_j^{b_j} = 0\}$ を要素として含むことは、式(13)を満足する $\exists(j)\{\beta_j \neq 0$ (β_j は実数)} である複素数解が存在しないための十分条件に過ぎない。なぜならば、グレブナ基底が $\forall(j)\{\beta_j^{b_j} = 0\}$ を要素として含まなくても、方程式(14) \cap (13)を満足する実数の組 $(\alpha_1, \dots, \beta_n)$ が存在せず、したがって式(13)を満足する $\exists(j)\{\beta_j \neq 0$ (β_j は実数)} である複素数解が存在しないということがあり得るからである。しかしながら、必要十分条件を追求する場合には、連立方程式(14) \cap (13)のすべての実数解を求めなくてはならないことになり、本質的に問題は解決しない。したがって本論文では、方程式(14) \cap (13)のグレブナ基底が $\forall(j)\{\beta_j^{b_j} = 0\}$ を要素として含むことを示すことによって、式(13)を満足する $\exists(j)\{\beta_j \neq 0\}$ である複素数解が存在しないことを確認している。

このことを確認できない場合には、外接球の中心として候補点 \mathbf{a} の代わりにその近傍の点 \mathbf{a} を用いることによってこの問題を回避するという方法を本論文では採用している。次節では近傍点 \mathbf{a} の選出方法について述べる。

3.2 近傍点の選出方法

方程式(14)の解は有限個なので、任意に選んだ近傍点 \mathbf{a} に関して、方程式(14) \cap (13) $\cap \exists(j)\{\beta_j \neq 0\}$ が成立する可能性は低いと考えられる。しかしながらその可能性が0でない以上、方程式(14) \cap (13) $\cap \exists(j)\{\beta_j \neq 0\}$ が成立しないような近傍点 \mathbf{a} の選出方法を確立しておく必要がある。以下その選出方法について述べるが、本節では議論をより一般化し、 α_j および β_j は複素

数であるとしている.

方程式 (14) の解の集合を C とし, C の要素によって決定される 1 次多項式を要素とする集合 $P(j)$ を導入する.

$$C = \{(\alpha_1, \dots, \beta_n) \mid (\alpha_1, \dots, \beta_n) \text{ satisfies Eq. (14).}\}$$

$$P(j) = \left\{ \sum_{k=1}^n (\alpha_k - x_k) \beta_k \mid \beta_1 = \dots = \beta_j = 0, (\alpha_1, \dots, \beta_n) \in C \right\} \quad (0 \leq j \leq n). \quad (15)$$

このとき, 集合 $P(j)$ には以下の包含関係がある.

$$P(0) \supseteq \dots \supseteq P(n).$$

方程式 (14) \cap (13) が $\exists(j) \{\beta_j \neq 0\}$ なる解を持つことは, 式 (16) が成立することと同値である (付録, 証明 1).

$$\exists(p(\mathbf{x}) \in P(0) \setminus P(n)) \{p(\mathbf{a}) = 0\}. \quad (16)$$

近傍点 \mathbf{a} が満足すべき条件は式 (16) が成立しないこと, すなわち次式 (17) が成立することである.

$$\forall(p(\mathbf{x}) \in P(0) \setminus P(n)) \{p(\mathbf{a}) \neq 0\}. \quad (17)$$

近傍点 \mathbf{a} は以下の手順によって, 候補点 \mathbf{a} の座標成分を順次修正することにより求められる. ある点 $\hat{\mathbf{a}} = (\hat{a}_1, \dots, \hat{a}_n)$ について, 次式 (18) が成立しているものとする.

$$\forall(p(\mathbf{x}) \in P(0) \setminus P(j)) \{p(\hat{\mathbf{a}}) \neq 0\}. \quad (18)$$

このとき, 次式 (19) が成立する点 $\hat{\mathbf{a}}^* = (\hat{a}_1^*, \dots, \hat{a}_n^*)$ を選出することができれば, 同様の操作を繰り返すことによって点 \mathbf{a} を選出することができる.

$$\forall(p(\mathbf{x}) \in P(0) \setminus P(j+1)) \{p(\hat{\mathbf{a}}^*) \neq 0\}. \quad (19)$$

このとき点 $\hat{\mathbf{a}}^*$ を式 (20) のように設定すると, 式 (21) が成立する (付録, 証明 2).

$$\hat{\mathbf{a}}^* = (\hat{a}_1, \dots, \hat{a}_{j+1} + \Delta, \dots, \hat{a}_n). \quad (20)$$

$$\forall(p(\mathbf{x}) \in P(0) \setminus P(j+1)) \forall(\Delta^0 \neq \Delta^1) \{p(\hat{\mathbf{a}}^*)|_{\Delta=\Delta^0} \neq 0 \vee p(\hat{\mathbf{a}}^*)|_{\Delta=\Delta^1} \neq 0\}. \quad (21)$$

$P(0) \setminus P(j+1)$ に含まれる $p(\mathbf{x})$ の数を l とする. 式 (21) により, Δ に $(l+1)$ 個の相異なる値を代入すると, 少なくともそのうちの一つの $\hat{\mathbf{a}}^*$ について式 (19) が成立する. 式 (19) が満足されていることは, 連立方程式 (14) \cap $\{\sum_{k=1}^n (\alpha_k - \hat{a}_k^*) \beta_k = 0\}$ のグレブナ基底の要素に $\forall(k \leq j+1) \{\beta_k^{b_k} = 0\}$ が含まれていることにより検証される. このとき, l の値をあらかじめ知ることはできないが, l の値は有限であるので, これらの操作が無限ループには陥らないことが保証される.

このようにして, 近傍点 \mathbf{a} は候補点 \mathbf{a} の座標成分を順次修正していくことによって得られる. 一つの座標成分を修正するときには, その修正が成功するまで修正量 Δ の値を変化させることになる. その際, Δ の値は, ある範囲内であれば任意に選ぶことができる. 次章では, 修正量 Δ の取りうる値の範囲と, 近傍点 \mathbf{a} を用いることにともなう許容誤差 ϵ_j の設定方法について述べる.

3.3 許容誤差の設定方法

図 3 は候補点 \mathbf{a} を中心とした場合の外接球 S と近傍点 \mathbf{a} を中心として用いた場合の外接球 S^* を示したものである. 図 3 により, 外接球の中心として候補点 \mathbf{a} のかわりに近傍点 \mathbf{a} を用いる場合, 領域 $\bar{R} \cap S^*$ に真の解が存在しないためには, 許容誤差 ϵ_j は式 (6) よりも厳しい条件を満足しなくてはならないことがわかる. しかしながら, 許容誤差 ϵ_j を再設定する場合には, 候補点 \mathbf{a} の計算から近傍点 \mathbf{a} の選出までの作業をあらためて行わなくてはならない. したがって本論文では, 複素数解が存在する場合には, あらかじめ近傍点 \mathbf{a} を選出する範囲を決めておき, その範囲内にあるどの \mathbf{a} を用いても領域 $\bar{R} \cap S^*$ に真の解が存在しないように許容誤差 ϵ_j を設定するという方法を採用する.

$\forall(j) \{|a_j^* - a_j| \leq \Delta_j\}$ とすると, 図 3 に示すように, 領域 $\bar{R} \cap S^*$ に真の解が存在しないという条件は次式 (22) によって満足される.

$$\forall(j) \forall(a_j \neq a_j') \{|a_j - a_j'| \geq \epsilon_j + \Delta_j + \sqrt{\sum_{k=1}^n (\epsilon_k + \Delta_k)^2}\}. \quad (22)$$

ただし, a_j および a_j' はともに方程式 $g_j(x_j) = 0$ の数値解である. 式 (22) は式 (6) と同様, 領域 $\bar{R} \cap S^*$ に真の解が存在しないための十分条件に過ぎないが, 式 (22) の成立は方程式 (2) の数値解を求めさえすれば検証可能であるので, これを採用している. なお, 式 (10) を満足するような ϵ を計算し, $\forall(j) \{\epsilon \geq (\epsilon_j + \Delta_j)\}$ が満足されるように各 ϵ_j, Δ_j を設定すれば式 (22) の

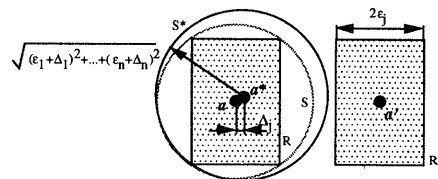


図 3 近傍点 \mathbf{a}^* を中心とする外接球 S^*
Fig. 3 Circumscribed sphere S^* whose center is the near point \mathbf{a}^* .

この絞り込み法では、1変数方程式の導出を $(n-1)$ 回行う必要がある。しかしながら、変数 t_k の定義式 (24) は x_j に関する1次式でありその係数も1であるので、1変数方程式 $h_k(t_k) = 0$ の導出に要する計算コストは、候補点 \mathbf{a} から真の解までの距離を表す変数 r に関する1変数方程式の導出に要する計算コストに比べて低く、本章で提案した方法は絞り込み方法として有効であるものと考えられる。

5. 例

本章では、二つの例題を取り上げる。まず第5.1節で、複素数解が存在する場合の例を示す。次に第5.2節で外接球法と他の方法との比較を、第5.3節で絞り込み方法の有効性の確認を行う。

5.1 複素数解が存在する例

本節では例題 (30) を取り上げる。

$$\begin{aligned} x_1 + x_2 + x_3 - 3 &= 0, \\ x_1^2 + x_2^2 + x_3^2 - 1 &= 0, \\ x_1^3 + x_2^3 + x_3^3 + 3 &= 0. \end{aligned} \tag{30}$$

式 (30) の各変数についての1変数方程式として式 (31) が導かれる。

$$\begin{aligned} x_1^3 - 3x_1^2 + 4x_1 - 2 &= 0, \\ x_2^3 - 3x_2^2 + 4x_2 - 2 &= 0, \\ x_3^3 - 3x_3^2 + 4x_3 - 2 &= 0. \end{aligned} \tag{31}$$

式 (31) の数値解の計算過程において、式 (30) が複素数解を持つことと式 (10) を満たす ϵ が任意であることがわかる。そこで、ここでは $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1/2$, $\Delta_1 = \Delta_2 = \Delta_3 = 1$ として以下の計算を行う。

式 (31) の数値解を組み合わせることにより、実数解の候補点集合 A として式 (32) が求められる。

$$A = \{(1, 1, 1)\}. \tag{32}$$

候補点 $(1, 1, 1)$ に関して方程式 $(14) \cap (13) \cap \exists(j)\{\beta_j \neq 0\}$ が成立するか否かを検証する。式 (30) をその実部と虚部に分割すると式 (33) となる。

$$\begin{aligned} \alpha_1 + \alpha_2 + \alpha_3 - 3 &= 0, \\ \beta_1 + \beta_2 + \beta_3 &= 0, \\ \alpha_1^2 - \beta_1^2 + \alpha_2^2 - \beta_2^2 + \alpha_3^2 - \beta_3^2 - 1 &= 0, \\ \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3 &= 0, \\ \alpha_1^3 - 3\alpha_1\beta_1^2 + \alpha_2^3 - 3\alpha_2\beta_2^2 \\ + \alpha_3^3 - 3\alpha_3\beta_3^2 + 3 &= 0, \\ 3\alpha_1^2\beta_1 - \beta_1^3 + 3\alpha_2^2\beta_2 - \beta_2^3 + 3\alpha_3^2\beta_3 - \beta_3^3 &= 0. \end{aligned} \tag{33}$$

また、式 (13) に $\mathbf{a} = (1, 1, 1)$ を代入することにより次式 (34) が得られる。

$$(\alpha_1 - 1)\beta_1 + (\alpha_2 - 1)\beta_2 + (\alpha_3 - 1)\beta_3 = 0. \tag{34}$$

式 (33) \cap (34) のグレブナ基底を計算すると、そこには、

$\beta_1^{b_1} = \beta_2^{b_2} = \beta_3^{b_3} = 0$ (b_j は自然数) なる要素は含まれておらず、したがって候補点 $(1, 1, 1)$ の近傍点 \mathbf{a}^* を選び、それを代わりに用いなくてはならないことがわかる。

第3.2節で述べた方法にしたがい、 \mathbf{a} の第1成分を修正する。このとき修正量 Δ は、その絶対値が Δ_1 以下の値であれば、任意に選べる。ここでは $\Delta = 1$ として、 $\hat{\mathbf{a}}^*$ を次式 (35) のように設定する。

$$\hat{\mathbf{a}}^* = (2, 1, 1). \tag{35}$$

$\mathbf{a} = \hat{\mathbf{a}}^*$ を式 (13) に代入すると式 (36) が得られる。

$$(\alpha_1 - 2)\beta_1 + (\alpha_2 - 1)\beta_2 + (\alpha_3 - 1)\beta_3 = 0. \tag{36}$$

式 (33) \cap (36) のグレブナ基底には、 $\beta_1 = 0$ は含まれているものの $\beta_2^{b_2} = \beta_3^{b_3} = 0$ は含まれていない。したがって、第1成分の修正はここで終了し、次に第2成分の修正を行う。 $\hat{\mathbf{a}}^*$ を式 (37) のように設定する。

$$\hat{\mathbf{a}}^* = (2, 2, 1). \tag{37}$$

以下、同様の作業を繰り返すことによって最終的に近傍点 \mathbf{a}^* が次式 (38) のように決定される。

$$\mathbf{a}^* = \left(2, \frac{1}{2}, 1\right). \tag{38}$$

なお、近傍点 \mathbf{a}^* を決定するために行った座標成分の修正回数は、以下に示すように5回である。

- (1) $\mathbf{a} = (1, 1, 1) \rightarrow (2, 1, 1)$
(第1成分を \mathbf{a} から +1 修正)
- (2) $(2, 1, 1) \rightarrow (2, 2, 1)$
(第2成分を \mathbf{a} から +1 修正)
- (3) $(2, 2, 1) \rightarrow (2, 0, 1)$
(第2成分を \mathbf{a} から -1 修正)
- (4) $(2, 0, 1) \rightarrow (2, \frac{3}{2}, 1)$
(第2成分を \mathbf{a} から $+\frac{1}{2}$ 修正)
- (5) $(2, \frac{3}{2}, 1) \rightarrow (2, \frac{1}{2}, 1)$
(第2成分を \mathbf{a} から $-\frac{1}{2}$ 修正)

$\hat{\mathbf{a}}^*$ を中心とする半径 r の球の方程式 (39) を作る。

$$r^2 = (x_1 - 2)^2 + (x_2 - \frac{1}{2})^2 + (x_3 - 1)^2. \tag{39}$$

式 (30) \cap (39) から r に関する1変数方程式を導くと、式 (30) \cap (39) は実数解 r を持たないことがわかり、式 (30) は実数解を持たないことがわかる。

5.2 外接球法と他の方法との比較

例題 (40)⁴⁾ を用いて外接球法と以下の二つの方法との比較を行う。

表1 逐次消去法による数値解

Table 1 Numerical solutions by the successive elimination method.

| | | 真の解 a_j^0 | 数値解 a_j | $ a_j - a_j^0 /\epsilon_j$ |
|-----|-------|-------------|------------|----------------------------|
| (1) | x_1 | -1 | -1.0000000 | 0.00 |
| | x_2 | 0 | 0.0000000 | 0.00 |
| (2) | x_1 | -0.7141434 | -0.7142333 | 0.74 |
| | x_2 | -0.6998564 | -0.6470947 | 432.22 |
| (3) | x_1 | -0.5000550 | -0.5001220 | 0.55 |
| | x_2 | -0.5000450 | -0.5289306 | 236.63 |
| (4) | x_1 | -0.0262280 | -0.0262451 | 0.14 |
| | x_2 | 0.9996512 | 1.0093994 | 79.85 |
| (5) | x_1 | -0.0227383 | -0.0228271 | 0.73 |
| | x_2 | -0.9997464 | -0.9488525 | 416.92 |
| (6) | x_1 | 0.5000350 | 0.5001220 | 0.71 |
| | x_2 | 0.5000650 | 0.5333251 | 272.46 |
| (7) | x_1 | 0.6645091 | 0.6644287 | 0.66 |
| | x_2 | 0.7471453 | 0.7864990 | 322.38 |
| (8) | x_1 | 0.9990732 | 0.9991455 | 0.59 |
| | x_2 | 0.0432840 | 0.3499755 | 2512.41 |

表2 単純代入法による数値解

Table 2 Numerical solutions by the simple substitution method.

| | | 真の解 a_j^0 | $(\delta_1 = \delta_2 = 2^{-13})$ | | $(\delta_1 = \delta_2 = 2^{-12})$ | |
|-----|-------|-------------|-----------------------------------|----------------------------|-----------------------------------|----------------------------|
| | | | 数値解 a_j | $ a_j - a_j^0 /\epsilon_j$ | 数値解 a_j | $ a_j - a_j^0 /\epsilon_j$ |
| (1) | x_1 | -1 | -1.0000000 | 0.00 | -1.0000000 | 0.00 |
| | x_2 | 0 | 0.0000000 | 0.00 | 0.0000000 | 0.00 |
| (2) | x_1 | -0.7141434 | -0.7142333 | 0.74 | -0.7142333 | 0.74 |
| | x_2 | -0.6998564 | -0.6998291 | 0.22 | -0.6998291 | 0.22 |
| (3) | x_1 | -0.5000550 | -0.5001220 | 0.55 | -0.5001220 | 0.55 |
| | x_2 | -0.5000450 | -0.5001220 | 0.63 | -0.5001220 | 0.63 |
| (4) | x_1 | -0.0262280 | -0.0262451 | 0.14 | -0.0262451 | 0.14 |
| | x_2 | 0.9996512 | 0.9996337 | 0.14 | 0.9996337 | 0.14 |
| (5) | x_1 | -0.0227383 | - | - | -0.0228271 | 0.73 |
| | x_2 | -0.9997464 | - | - | -0.9996337 | 0.92 |
| (6) | x_1 | 0.5000350 | 0.5001220 | 0.71 | 0.5001220 | 0.71 |
| | x_2 | 0.5000650 | 0.5001220 | 0.47 | 0.5001220 | 0.47 |
| (7) | x_1 | 0.6645091 | 0.6644287 | 0.66 | 0.6644287 | 0.66 |
| | x_2 | 0.7471453 | 0.7471923 | 0.39 | 0.7471923 | 0.39 |
| (8) | x_1 | 0.9990732 | - | - | 0.9991455 | 0.59 |
| | x_2 | 0.0432840 | - | - | 0.0433349 | 0.42 |
| | x_1 | - | -0.0262451 | - | -0.0262451 | - |
| | x_2 | - | -0.9996337 | - | -0.9996337 | - |
| | x_1 | - | - | - | -0.0228271 | - |
| | x_2 | - | - | - | 0.9996337 | - |

$$\begin{aligned}
 &x_2 x_1^3 - \frac{1}{4} x_1^2 \\
 &+ \left(x_2^3 - \frac{9999}{10000} x_2 \right) x_1 - \frac{1}{4} (x_2^2 - 1) = 0, \\
 &x_1^3 - x_2 x_1^2 + \left(x_2^2 - \frac{100001}{100000} \right) x_1 \\
 &- \left(x_2^3 - x_2 + \frac{1}{100000} \right) = 0.
 \end{aligned}
 \tag{40}$$

- (1) 逐次消去法
変数一つずつ消去していくことによって数値解を得る。
- (2) 単純代入法
各変数についての1変数方程式を導き、それらの数値解を組み合わせるによって解の候補点 \mathbf{a} を構成する。次に、各候補点を元の方程式

に代入してそれが解であるか否かを判定する。その場合、判定誤差 δ_j を設定し、次式(41)が満足されれば \mathbf{a} は解であるとされる。

$$|f_1(\mathbf{a})| < \delta_1, \dots, |f_m(\mathbf{a})| < \delta_m. \tag{41}$$

表1から表3に計算結果を示す。実際の計算はすべて有理数で行われているが、計算結果はすべて小数で示してある。また、これらの計算では、許容誤差 ϵ_1 および ϵ_2 は条件式(6)を満足するように以下のように設定してある。

$$\epsilon_1 = \epsilon_2 = 2^{-13}.$$

表1から、逐次消去法では誤差が累積され設定された許容誤差内では数値解を得られないことがわかる。また、単純代入法では真の解とは無縁なものも数値解として求められ、特に $\delta_1 = \delta_2 = 2^{-13}$ の場合には、

表3 外接球法による数値解
Table 3 Numerical solutions by the circumscribed sphere method.

| | | 真の解 a_j^0 | 数値解 a_j | $ a_j - a_j^0 /\epsilon_j$ |
|-----|-------|-------------|------------|----------------------------|
| (1) | x_1 | -1 | -1.0000000 | 0.00 |
| | x_2 | 0 | 0.0000000 | 0.00 |
| (2) | x_1 | -0.7141434 | -0.7142333 | 0.74 |
| | x_2 | -0.6998564 | -0.6998291 | 0.22 |
| (3) | x_1 | -0.5000550 | -0.5001220 | 0.55 |
| | x_2 | -0.5000450 | -0.5001220 | 0.63 |
| (4) | x_1 | -0.0262280 | -0.0262451 | 0.14 |
| | x_2 | 0.9996512 | 0.9996337 | 0.14 |
| (5) | x_1 | -0.0227383 | -0.0228271 | 0.73 |
| | x_2 | -0.9997464 | -0.9996337 | 0.92 |
| (6) | x_1 | 0.5000350 | 0.5001220 | 0.71 |
| | x_2 | 0.5000650 | 0.5001220 | 0.47 |
| (7) | x_1 | 0.6645091 | 0.6644287 | 0.66 |
| | x_2 | 0.7471453 | 0.7471923 | 0.39 |
| (8) | x_1 | 0.9990732 | 0.9991455 | 0.59 |
| | x_2 | 0.0432840 | 0.0433349 | 0.42 |

そのような誤った解が得られている一方で真の解の近似値が排除されていることが表2に示されている。これに対して、表3に示すように、外接球法ではすべての解が設定された許容誤差の範囲内で求められており、また、誤った解を求めていることもない。これらの結果から、方程式の数値解を許容誤差内で求めるといふ点において、外接球法が他の二つの方法に比べて有効であることがわかる。

しかしながら、外接球法では各変数に関する1変数方程式の導出を n 回、さらに候補点 \mathbf{a} から真の解までの距離を表す変数 r に関する1変数方程式の導出を候補点の数だけ行う必要がある。これに対して、逐次消去法では辞書式順序²⁾のグレブナ基底を計算すれば1変数方程式の導出を行う必要はなく、単純代入法でも各変数に関する1変数方程式の導出を n 回行うだけですみ、計算コストの面では外接球法は他の二つの方法に比べて劣る。実際例題(40)では、解の候補点の絞り込みを行わない場合、外接球法は他の二つの方法に比べて20倍程度の計算時間を要している。この欠点を少しでも補うためには第4章で述べた解の候補点 \mathbf{a} の絞り込みが必要である。

5.3 解候補の絞り込み

第4章で提案された解の候補点集合の構成方法を例題(40)に対して適用した例を示す。方程式(40)から計算された1変数方程式 $g_1(x_1) = 0$ および $g_2(x_2) = 0$ はそれぞれ八つの実数解を持つ。したがって単純にこれらの解を組み合わせた場合には、候補点集合 A は64個の候補点を持つことになる。しかしながら第4章で提案された構成方法を用いると、許容誤差 ϵ_j を $\epsilon_1 = \epsilon_2 = 2^{-13}$ とした場合、候補点集合 A には候補点は8個しか含まれず、結果的に真の解とは無縁な候補点はすべて排除される。これにより実際の計算時間

は、絞り込みを行わない場合に比べて約6分の1となり、この構成方法が有効であることが確認された。

6. おわりに

多変数連立代数方程式のすべての実数解を求めるための方法として、外接球法を提案した。さらに、例題として方程式(40)を用いて他の方法との比較を行い、他の方法ではすべての数値解を許容誤差内で求めることは極めて困難であるが、外接球法では確実に求められることを示した。また、外接球法は他の方法に比べて多くの計算コストを要するが、解の候補点を絞り込むことによってある程度それが改善できることを示した。

外接球法の特長は、すべての近似解を求めるために許容誤差が満足すべき条件を計算の過程で明らかにできることである。他の方法でも、許容誤差を十分小さく設定すればすべての数値解を許容誤差内で計算することが可能であろう。しかしながら、他の方法ではその十分小さな許容誤差を決定する方法がない。したがって、許容誤差をどれだけ小さくとっても、すべての数値解が許容誤差内で求められることが保証されない。

外接球法を実行する場合、その計算時間の大部分はグレブナ基底計算によって占められる。したがって、外接球法の計算効率を向上させるにはグレブナ基底の計算効率を向上させることが不可欠となる。グレブナ基底計算の効率向上については多くの研究が続けられており^{6),7)}、今後の研究成果が期待される場所である。

本論文では外接球法によって実数解を得る方法のみを示したが、変数およびもとの方程式をその実部と虚部に分けることで全複素数解を求めることも可能で

ある。

謝辞 本研究にあたり、適切な助言をいただいた機械技術研究所システム工学研究室長の小鍛治繁博士に深く感謝します。

参考文献

- 1) 奥川光太郎: 基礎数学講座 1 巻 代数学, p.194, 共立出版, 東京 (1956).
- 2) Buchberger, B.: Gröbner bases: An Algorithmic Method in Polynomial Ideal Theory, Technical Report, RISC-LINZ (1983).
- 3) 小林英恒: 多変数連立代数方程式の解法, 情報処理, Vol.27, No.4, pp.414-421 (1986).
- 4) 野田松太郎, 佐々木建昭, 鈴木正幸: 数式処理と数値計算の融合による精度保証, 情報処理, Vol.31, No.9, pp.1204-1211 (1990).
- 5) 山本哲郎, 陳小君: 有限次元非線形方程式に対する保証付き数値計算, 情報処理, Vol.31, No.9, pp.1191-1196 (1990).
- 6) 野呂正行: モジュラ演算によるグレブナ基底計算の高速化, 研究集会—数式処理における理論と応用の研究—, 京都大学 数理解析研究所 (1993).
- 7) Sawada, H., Terasaki, S. and Aiba, A.: Parallel Computation of Gröbner Bases on Distributed Memory Machines, *Journal of Symbolic Computation*, Vol.18, pp.207-222 (1994).

付 録

A.1 証 明

証明 1 $\exists(\alpha_1, \dots, \beta_n)\exists(j)\{\text{式(14)} \wedge \text{式(13)} \wedge \beta_j \neq 0\}$ と式(16)が同値であることを示す。

$\exists(\alpha_1, \dots, \beta_n)\exists(j)\{\text{式(14)} \wedge \text{式(13)} \wedge \beta_j \neq 0\}$.

$\leftrightarrow \exists(\alpha_1, \dots, \beta_n)\exists(j)$

$\{(\alpha_1, \dots, \beta_n) \in C \wedge \sum_{k=1}^n (\alpha_k - a_k)\beta_k = 0 \wedge \beta_j \neq 0\}$.

$\leftrightarrow \exists(p(x) \in P(0) \setminus P(n))\{p(a) = 0\}$.

□

証明 2 $p(\hat{a}^*)|_{\Delta=\Delta^0} = p(\hat{a}^*)|_{\Delta=\Delta^1} = 0$ から矛盾が導かれることを示す。

式(15)および式(20)より次の関係式が導かれる。

$$p(\hat{a}^*) = p(\hat{a}) - \beta_{j+1}\Delta.$$

したがって $p(\hat{a}^*)|_{\Delta=\Delta^0} = p(\hat{a}^*)|_{\Delta=\Delta^1} = 0$ より式(42)が得られ、式(43)が導かれる。

$$\begin{aligned} p(\hat{a}^*)|_{\Delta=\Delta^0} &= p(\hat{a}) - \beta_{j+1}\Delta^0 = 0, \\ p(\hat{a}^*)|_{\Delta=\Delta^1} &= p(\hat{a}) - \beta_{j+1}\Delta^1 = 0. \end{aligned} \quad (42)$$

$$\beta_{j+1}(\Delta^0 - \Delta^1) = 0. \quad (43)$$

(1) $p(x) \in P(0) \setminus P(j)$ のとき。

式(43)および $\Delta^0 \neq \Delta^1$ より $\beta_{j+1} = 0$. 式(42)より $p(\hat{a}) = 0$ となり式(18)に矛盾。

(2) $p(x) \in P(j) \setminus P(j+1)$ のとき。

式(43)および $\Delta^0 \neq \Delta^1$ により $\beta_{j+1} = 0$. $P(j)$ の定義式(15)により $\beta_{j+1} \neq 0$ であるので矛盾。

□

(平成7年5月17日受付)

(平成7年10月5日採録)



沢田 浩之 (正会員)

1964年生まれ。1987年、東京大学工学部航空学科卒業。1989年、東京大学大学院工学系研究科航空学専攻課程修士修了。同年、工業技術院機械技術研究所勤務。1990年より92年まで(財)新世代コンピュータ技術開発機構へ休職出向。同年、機械技術研究所に復職。数式処理およびその機械設計支援技術への応用に取り組む。日本航空宇宙学会会員。