

悪条件代数方程式へのハイブリッド計算と関連する試み

野田 松太郎 (愛媛大学工学部)

1 1 変数代数方程式に対するハイブリッド計算

代数方程式を解く場合に、方程式が良条件なら数値解法はきわめて強力で、特に 1 変数の場合には D K 法を用いると全根を同時に困難なく求めることが出来る。また、良条件の多変数連立代数方程式系に対してもニュートン法やその変形アルゴリズムで十分な精度の根を求める得る。

しかし、方程式が悪条件になると状況は大幅に変化する。1 変数の場合の悪条件性は、文献 [1] にもあるように

方程式の係数のわずかな変化が大きな根の変動をもたらす

とまとめられ、代数方程式が重根や近接根を含む場合に相当する。このとき、数値解法の収束は遅くかつ真の根へ収束しない場合もある。このような問題に数値解法を適用しようという試みは古くから行われているが、ほとんど成果をあげていないといえる。これに対して、佐々木、野田 [2, 3] は数式処理で行うような代数計算を浮動小数点係数な場合に拡張して、上のような悪条件問題を解決し高精度な数値解を得る方法を提唱した。

もし、 $F(x)$ が x の多項式とし、代数方程式

$$F(x) = 0 \quad (1.1)$$

が重根を含むなら、2 つの多項式 $F_1 = F(x)$ と $F_2 = dF(x)/dx$ との最大公約多項式 $D = \gcd(F_1, F_2)$ を計算する。これをもとに、 $F(x)$ の無平方分解を行うと多重度を含めて方程式の重根部分を求めることが出来る。 D を求める計算は良く知られたユークリッドの互除法による。しかし、代数方程式の係数に浮動小数が含まれたり、近接根があったりすると代数的な計算は行えなくなる。ほとんどの重根は近似的重根であると見なし得、代数計算を用いようとしても係数膨張などにより不経済性は増すのみとなる。この場合にもユークリッドの互除法をあえて適用し、計算される浮動小数を含む多項式剰余列を忠実に追いついてその収束を調べる。多項式剰余列の各多項式の係数は急速に減少するが、その大きさと近似的重根あるいは近接根の間の根間距離との間に関係があることが分かる。もし多項式剰余列を、多項式の係数の大きさ $< \varepsilon$ (微小量) で打ち切ると、それから近似的 GCD を求められることが出来る。近似的 GCD により近似的無平方分解を行うと、代数計算の場合と同じく悪条件部分を分離することが可能になる。このような解法をハイブリッド計算と呼ぶ。この場合に近似的重根も近接根もともにそれらの重心に位置のある近似的重根と見なされる。したがって次に近似的重根の多重度 (近似的無平方分解で得られる) までの次数のテーラー展開を行って

得られる代数方程式から重心からの各根のずれを高い精度で数値的に求める。すなわち近接根部分の根の値も高い精度で求め得る。以上をまとめると、悪条件単独（1変数）代数方程式に対するハイブリッド計算は

- 数値計算では正しい根を求めにくい悪条件問題にも正しい根を与える。
- 重根、近接根をとともに含んだ場合にもきわめて良い精度の根を与える。
- 数式処理で行う代数計算のように係数膨張などがなく高速である。

などの利点がある。

ハイブリッド計算を多変数の連立代数方程式系に拡張しようとするのは当然である。もし、多変数代数方程式を変数を消去し、主変数の1変数代数方程式に簡単に変換し得るなら、上で述べた方法は容易に適用範囲を広げる。しかし事情はさして簡単ではない。変数消去は数値計算では困難で数式処理のような記号計算で行わなければならない。現在この分野で多項式系の簡単化の決定版のように云われているグレブナー基底 [5] の計算は原理的には1変数化が可能でも計算時間は大変なものになり、多変数で高次の問題には使用不可能である。さらに多変数代数方程式系での悪条件性のおこり方にも色々のタイプがある。したがって、1変数の場合に議論したものとは異なったハイブリッド計算その他を考えなくてはならない。

2 多変数代数方程式へのハイブリッド計算の適用

悪条件多変数連立代数方程式系を取り扱う場合に、上で述べたように悪条件性のタイプを考える必要がある。まず、1変数代数方程式に対して議論したハイブリッド計算の考え方を少し変えると根を得られる場合について考察する。これは越智、野田、佐々木 [3, 4] によって提唱されている。ここで扱う方程式系は

$$\begin{cases} F_1(x_1, \dots, x_n) = D(x_1, \dots, x_n)\tilde{F}_1(x_1, \dots, x_n) + \Delta F_1(x_1, \dots, x_n) = 0 \\ \dots \\ F_n(x_1, \dots, x_n) = D(x_1, \dots, x_n)\tilde{F}_n(x_1, \dots, x_n) + \Delta F_n(x_1, \dots, x_n) = 0 \end{cases} \quad (2.1)$$

のタイプである。ここで、多項式 F_k の数係数の中、絶対値最大のものを **最大係数** といい、 $\text{mmc}(F_k)$ と表す。また ε を正の微小量としたとき、(2.1) で $\text{mmc}(F_k) = 1$, $k = 1, \dots, n$ 、 $\text{mmc}(\Delta F_k) = O(\varepsilon)$ であるとき、 D を **精度 ε の近似的因子** という。特に、 D が次数最大るとき D を **精度 ε の近似的 GCD** といい、

$$D = \gcd(F_1, \dots, F_n; \varepsilon)$$

と表す。この計算も1変数の場合に述べたように主変数に関する多項式剰余列を計算して求める。(2.1) のような代数方程式系の悪条件性は精度 ε の近似的因子が存在した場合におこる。明らかに、(2.1) で $\Delta F_1 = \dots = \Delta F_n = 0$ のときには (2.1) の根は

$\{D=0\}$ と $\{\tilde{F}_1=0, \dots, \tilde{F}_n=0\}$ とに分かれる。一般に、後者の根は有限個だが前者からは無限個の根が出る。これに対して $\Delta F_k \neq 0$ のときは根の数は有限個になる。すなわち係数のわずかな変化により根の数が急激に変化するという悪条件性がある。また根の数が有限個になっても $\{D=0\}$ の近くにある根を高精度で数値的に求めるのはきわめて困難であることも明かである。1変数の場合と同様、多項式 GCD である D を代数計算で求められれば残りの根の計算は容易である。しかし浮動小数係数であると精度 ε の近似的 GCD の計算をしなければならず、このためのハイブリッド計算のアルゴリズムを作らなければならない。文献 [3, 4] に述べられているハイブリッドアルゴリズムを簡単にまとめると次のようになる。

いま、2変数連立代数方程式が

$$\begin{cases} F(x, y) = D(x, y)\tilde{F}(x, y) + \Delta F(x, y) = 0 \\ G(x, y) = D(x, y)\tilde{G}(x, y) + \Delta G(x, y) = 0 \end{cases} \quad (2.2)$$

と分解されるとする。 $D(x, y)$ は $F(x, y)$ と $G(x, y)$ の精度 ε の近似的 GCD として計算される。そこで近似的記号除算を行い、(2.1) を

$$\begin{cases} F(x, y) = 0 \\ H(x, y) = \tilde{G}(x, y)F(x, y) - \tilde{F}(x, y)G(x, y) = 0 \end{cases} \quad (2.3)$$

と変換するのは容易である。(2.2) に含まれた悪条件性は (2.3) ではほとんどの場合解消されている。

通常の数値計算で (2.2) を解こうとすると初期値の設定が大きな問題である。実際にニュートン法などを適用する場合に解への収束・非収束、収束の速さなどにはすべて初期値の設定が強く関与する。上に述べたハイブリッド計算を用いると、今の場合も $\{D=0, H=0\}$ の粗い数値解から $D(x, y)=0$ の近傍に分布する(解きにくい)根の初期値を、 $\{\tilde{F}=0, \tilde{G}=0\}$ から通常の方法でも解きやすい根の初期値を求めることが出来る。

実例として文献でもあげた、上の種類の悪条件代数方程式

$$\begin{cases} F(x, y) = yx^3 - 0.25x^2 + (y^3 - 0.9999y)x - 0.25(y^2 - 1) \\ G(x, y) = x^3 - yx^2 + (y^2 - 1.00001)x - (y^3 - y + 0.00001) \end{cases} \quad (2.4)$$

についてハイブリッド計算の結果を示す。 $\varepsilon = 0.0001$ として、精度 ε の近似的 GCD を求めると

$$D(x, y) = x^2 + y^2 - 1.0$$

である。これを用いて (2.4) を分解すると

$$\begin{cases} F(x, y) = (x^2 + y^2 - 1)(xy - 0.25) + 0.0001xy \\ G(x, y) = (x^2 + y^2 - 1)(x - y) - 0.00001(x + 1) \end{cases}$$

となり、

$$H(x, y) = 0.0001(-1.1x^2y + xy^2 - 0.1xy + 0.025x + 0.025)$$

も得ることが出来る。初期値を $\{D=0, H=0\}$ 及び $\{\tilde{F}=0, \tilde{G}=0\}$ を粗く解いて決定し、悪条件性を無くした方程式 $\{F=0, H=0\}$ を解くと結果が求まる。表 1 に通常の数値計算の結果と上のハイブリッド計算の結果とを比較する。数値計算の初期値は 100 組の乱数を $[-1, 1]$ の範囲で発生させた。表にはこれらの中、各根に収束する個数と平均の収束回数を示した。ハイブリッド計算の初期値の中、根 1 ～ 6 に収束するものは第 1 の組、残りは第 2 の組から得られる。平均の収束回数がやはり示されている。明らかにこの種の問題への通常の数値解法の応用には限界があり、全ての根を計算できないことが分かる。なお、表にある真の根の値は多変数多項式のスツルム列の代数的計算を、根を含む区間を十分小さくするまで繰り返して求めた [8]。ハイブリッド計算での結果はこの代数計算の結果と一致している。これに対して、(2.1) のタイプの問題には 1 変数の場合を拡張したハイブリッド計算は有力であるといえる。しかし、すでに述べたように多変数連立代数方程式系の悪条件問題には分岐問題など多くのタイプがある。しかし本稿で述べてきた計算法が多くの問題にどの程度まで有効かは今後の課題である。

3 曲線追跡法と区間計算

次に、数式処理との結合ではないが、単純な数値計算を少し拡張した方法で悪条件問題を解くことを考える。基本として用いるのは曲線追跡法である。この方法も広く教科書などでも紹介されており、本講究録中にもこの方法の応用が含まれているので詳細は述べない。以下では、本講で取り上げているある種の悪条件代数方程式系 (2.4) に対して曲線追跡法を用いてみる。

この方法で代数方程式系 (2.4) の根を求めるには、

- 1) 曲線追跡の初期値を求め、曲線からのズレの許容値 δ を入力する。

第 1 式 $F(x, y) = 0$ に $y = 0.5$ を代入し解を (x_0, y_0) とする。

- 2) 追跡方向を定め、上の初期値で第 1 式をルンゲクッタ法で追跡する。各ステップで得られた点を $(x_i, y_i), i = 1, \dots$ とする。ただし、

- 3) $F(x_i, y_i) > \delta$ なら、曲線上の点になるように補正をする。

$F(x_{i-1}, y_{i-1})$ を初期値として曲線に対するニュートン法による補正を行い、 $F(x_{i-1}, y) = 0$ を満たす y を y_{i-1} と取り直す。

- 4) $G(x_{i-1}, y_{i-1}) \cdot G(x_i, y_i) < 0$ ならステップ 5) へ、さもなければステップ 2) で追跡を繰り返す。

- 5) (x_i, y_i) を初期値としてニュートン法により根を求める。

- 6) 追跡の初期値を (x_i, y_i) とし同じ追跡方向で全ての根が求まるまで、ステップ 2) ~ 5) を繰り返す。

なるアルゴリズムを用いる。少なくともこのまま実際の計算をするには、初期値と追跡方向に不定性がある。例えば (2.4) で $y = 0.5$ として定めた初期値 (これを I_1 とする) による追跡曲線と $y = -0.5$ として定めた初期値 (I_2) による曲線とには違いがある。それらはを各々図 1₁, 1₂ に示した。このままでは $F(x, y)$ の全ての曲線を追跡できず、当然すべての根を見いだすこともできない。(2.4) の根は表 1 の通りなので、初期値 I_1 では 6 の根が、初期値 I_2 では 5 の根が求められない。このように実数計算のままで曲線追跡した場合に根を得られるか否かの結果を初期値 I_1 に対して表 1 に加えた

さて、この曲線追跡法を区間演算化するとどうなるだろうか。本来区間演算は解の精度の向上などに主眼があり、このような分岐問題への応用はあまり考えられていない。当然図 1 のように追跡すべき曲線は基本的には円と双曲線で自ら交わっているので、交点近くでは結果はおかしくなる。すなわち図 2 のように区間幅が急激に拡大する。したがって一見すると、区間演算のこれ以上の使用は不可能に思える。しかし、区間演算の結果をよく調べると、区間幅の上下限は分岐した曲線を含んでいることが分かる (図 3)。図 3 で追跡すべき曲線が $a \rightarrow$ 分岐 b, c, d であるとする。これに対して、区間演算で追跡してきた a 部の区間の上下限を各々 I_{a^u}, I_{a^l} とする。 b, c, d はこの区間内に入っている。この事実は区間演算の結果と分岐問題との間に強い相関があり、この点に着目すると実数計算のみの結果を拡張し得る可能性があると思われる。

曲線の分岐点ではヤコビアンが 0 になることが知られている。(2.4) では

$$\begin{vmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} \end{vmatrix} = 0$$

であり、このとき

$$\frac{\partial F}{\partial x} = 0, \quad \frac{\partial F}{\partial y} = 0$$

となっている。上の 0 は区間演算では下限が負、上限が正の区間数となり、 $(\partial F / \partial x)^2$ などは下限が 0、上限が正となる。その結果、区間演算での曲線追跡は分岐点付近で 0 での除算を含み続行不可能になる。そこで I_{a^u} を初期値として実数計算で曲線追跡を続行すれば、特別な場合を除き図 3 の b 部を追跡することになる。十分に実数計算の結果が b 部に近づくと、この点を区間の midpoint とした区間数にして再び区間演算での曲線追跡を繰り返す。同様の操作を I_{a^l} を初期値として行うと d 部を区間演算で追跡し得る。この部分のアルゴリズムは次のように書かれる。

アルゴリズム 分岐曲線の区間演算での追跡

入力 点 $x_i = \bar{x}_i$ での区間演算での曲線追跡の区間数 $I(\bar{x}_i)$

(上限: $I_{\bar{x}_i^u}$, 下限: $I_{\bar{x}_i^l}$;

- 方法
1. if 計算続行可能 (ヤコビアン $\neq 0$) then return (曲線追跡を続行)
 else $z_1 := I_{\bar{x}_i^u}; z_2 := I_{\bar{x}_i^l}$ として以下を実行
 2. for $k := 1, 2$ do begin
 3. 初期値 z_k 実数計算での曲線追跡を数回行い、点 $\tilde{z}_k$ を得る;
 4. $\tilde{z}_k$ を中点とする区間数を求める;
 5. 新しく求めた区間数を初期値として区間演算による曲線追跡を行う
 6. end.

このように実数計算と区間演算を併用し修正した曲線追跡のアルゴリズムを区間演算ソフトウェア PASCAL-SC [7] で実行する。結果は図4の経路で (2.4) の第1式 $F(x, y) = 0$ の曲線の全てを追跡することが出来る。したがって、 $G(x_{i-1}, y_{i-1}) \cdot G(x_i, y_i)$ の符号変化を調べると代数方程式系 (2.4) の全根を見いだすことが可能となる。さらに、このアルゴリズムを用いると (2.4) のような特別のタイプの代数方程式系の根を求め得るばかりでなく、たとえば非線形回路の問題から表れる牛田による2個のトンネルダイオードの直列結合回路の駆動点特性 [9] などの曲線追跡も系統的に困難なく行うことが出来る。

4 各種方法の比較

以上述べてきたように、本稿では(特殊なタイプではあるが)悪条件代数方程式系の全ての根を求めるため、一般的ではないいくつかの方法の紹介や提案をしてきた。少なくともニュートン法などの**通常の数値解法**では、ここで取り上げるタイプの問題に対しては

全ての根を求め得るとは限らず、かつ初期値の選択に対する依存性が大きい

といえる。特に (2.4) の根 5 や 6 は初期値を乱数の組にしたニュートン法でも見つけられなかったり、反復回数が増大したりし、曲線追跡を行ってみてもこれらの根の近くの曲線がうまく追跡されなかったりする。すなわち、**通常の数値解法**はこの種の問題にはきわめて不安定であるといえる。これに対し、くわしくは述べなかったがグレブナー基底の計算やスツルム列の正確計算などによる**代数的算法**では (2.4) のように2変数で多項式の次数があまり大きくない場合はそれなりの計算時間で安定な根を求めえても、

変数の個数や多項式の次数が少し大きくなると中間式や係数の膨張が著しく、計算時間の上で実用的ではなくなる

という欠点がある。

一方、近似的 GCD 算法に基づくハイブリッド計算では

全根を少ない反復回数で求められ、変数の増加にもある程度対応し得る

という点で従来の方法より有利である。ただし、この計算法も現在はある意味では研究段階にあり、より広いタイプの問題に使用するためには解決すべき課題が多く残っている。本文では触れなかったが、紹介したハイブリッド計算法に似た発想から非線形代数方程式系の根を求めるために Auzinger と Stetter が開発した手法がある [6]。これは基本的には **終結式理論に基づく方法** であるが、多変数終結式の計算には数式処理が必要なので、数値計算のみで行うため代数方程式中の累乗積 (Power Product) を基底にした行列を作成し (要素は各累乗積の数係数のみになる)、これを分解して簡単な固有値問題に置き換えるものである。この方法を (2.4) に適用すると

全根を求めることが出来て、FORTRAN 計算も容易である

という利点がある。重根を含む場合は適用できないが、もし数式处理的に最初に無平方分解の操作などを施せば、この手法も有力になり得る。ただし、(2.4) に適用した場合にも含まれる累乗積は yx^3 など 1 2 通りになる。実際の計算ではより大次元の行列計算が必要になる。すなわちこの方法にも

重根には対応しきれず、複雑な問題には大次元行列の操作が煩雑である

という欠点がある。

区間演算の曲線追跡への使用は一見区間演算の本来の使用目的とも異なり、無駄な計算をしているようだが、分岐点でヤコビアンが 0 になり、区間演算が継続できなくなることを逆用して実数計算と併用して曲線追跡を続行することが可能になる。その結果、問題 (2.4) に対しては $F(x, y) = 0$ の全曲線を追跡することが出来る。曲線上の各値をもとに $G(x, y)$ の符号変化を調べ、変化する点でニュートン法を用いれば (この段階で、始めて区間演算らしい使い方になる) 精度良く全ての根を求めることが出来る。また、この方法は (2.4) のような特殊な問題に限らず、より幅広いタイプの分岐問題などの非線形代数方程式系の根を計算し得る可能性が大きい。よって、区間演算による曲線追跡に基づく方法は

(2.4) の全根を求めることが可能で、かつ他の色々のタイプの悪条件問題へも応用の可能性が大きい

という特徴を有する。しかし、この手法には

- 計算法としての正当性や収束に関する問題
- 分岐点近くで実数計算に引き渡したときの実数計算に対する保証
- 計算の手間は大規模な問題に対して、どの程度増すか

- 初期値の選択法と曲線探索の方向をいかに確立するか
- どの程度までの非線形問題に応用し得るか

などなど今後検討すべき課題が山積されている。

参考文献

- [1] E.H.Bareiss, The Numerical Solution of Polynomial Equation and the Resultant Procedure, *in: Mathematical Method for Digital Computers - Vol II*, John Wiley (1967).
- [2] T.Sasaki and M.T.Noda, Approximate Square-free Decomposition and Root-finding of ill-conditioned Algebraic Equations, *Jour. Inf. Proc.*, Vol.12, No.2, pp.159-168 (1989).
- [3] 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, *情報処理学会誌*, Vol.31, No.9, pp.1204-1211 (1990).
- [4] M.Ochi, M.T.Noda and T.Sasaki, Approximate GCD of Multivariate Polynomials and Application to ill-conditioned System of Algebraic Equations, *Jour. Inf. Proc.* (*to appear*).
- [5] B.Buchberger, Gröbner bases: an Algorithmic Method in Polynomial Ideal Theory, *in: Multidimensional System Theory* (Ed. N.K.Bose), pp.184-232, D.Reidel Pub.Co. (1985).
- [6] W.Auzinger and H.J.Stetter, An Elimination Algorithm for the Computation of All Zeros of a System of Multivariate Polynomial Equations, *in: Numerical Mathematics Singapore 1988* (Eds. R.P.Agarwal et.al.), pp.11-30, Birkhäuser Verlag (1988).
- [7] U.W.Kurlisch (Ed.), PASCAL-SC : A PASCAL Extension for Scientific Computation, B.G.Teubner (1987).
- [8] J.H.Davenport, Y.Siret and E.Tournier, *Computer Algebra*, Academic Press (1988).
- [9] 池田 智保, 牛田 明夫, 解曲線追跡法による抵抗回路における分岐点の求解法, *電子情報通信学会研究会資料 (NLP 90-13)*, pp.17-23 (1990).

表1 代数方程式の根を求める色々な計算法の比較

$$\text{例題} \begin{cases} F(x, y) = (x^2 + y^2 - 1)(xy - 0.25) + 0.0001xy \\ G(x, y) = (x^2 + y^2 - 1)(x - y) - 0.00001(x + 1) \end{cases}$$

方法

数値計算： $\{F = 0, G = 0\}$ をニュートン法で解く（初期値は 100 組の乱数）

ハイブリッド計算：近似的 GCD を求め、良条件の $\{F = 0, G = 0\}$ に変更

曲線追跡法： $F(x, 0.5) = 0$ の解を $x = x_0$ とすると、 $(x_0, 0.5)$ を初期値とする

計算は次の 2 種

通常の実数計算（初期値は 3 点、探索は両方向）

区間演算を主にしたアルゴリズム（初期値は 1 点、探索は両方向）

なお、真の根はスツルム列を数式処理的に計算したものを基本にして計算 [8]

根	真の根	数値計算		ハイブリッド計算		曲線追跡法	
		収束回数/100	平均反復数	収束?	反復数	実数計算	区間演算
1	$x = 0.9990732$ $y = 0.0432840$	23	23.39	○	2	○	○
2	$x = -0.0262280$ $y = 0.9996512$	13	23.08	○	2	○	○
3	$x = -1.0$ $y = 0.0$	11	16.91	○	1	○	○
4	$x = -0.0227383$ $y = -0.9997464$	22	17.64	○	2	○	○
5	$x = 0.6645091$ $y = 0.7471453$	収束しない		○	3	○	○
6	$x = -0.7141434$ $y = -0.6998564$	7	16.57	○	3	×	○
7	$x = 0.5000350$ $y = 0.5000650$	17	5.53	○	2	○	○
8	$x = -0.5000550$ $y = -0.5000450$	7	6.00	○	2	○	○

図1 実数計算による曲線追跡法と代数方程式系の根

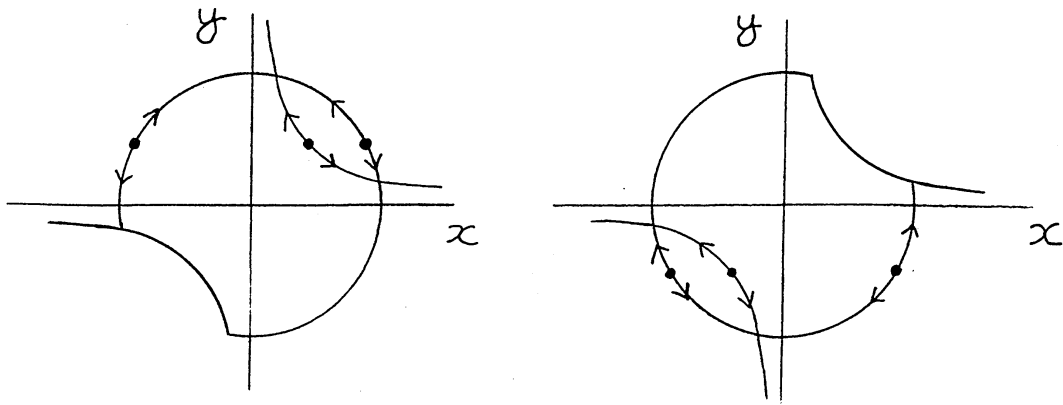


図2 区間演算による曲線追跡

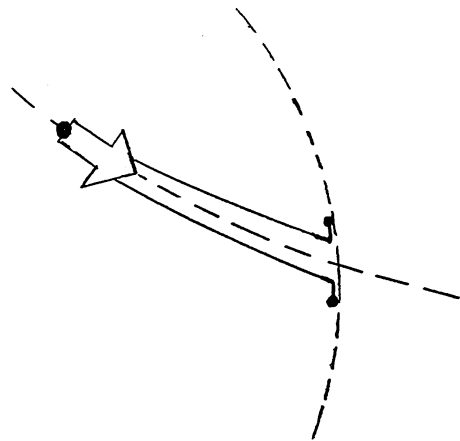


図3 区間演算による曲線追跡と分岐点での振舞い

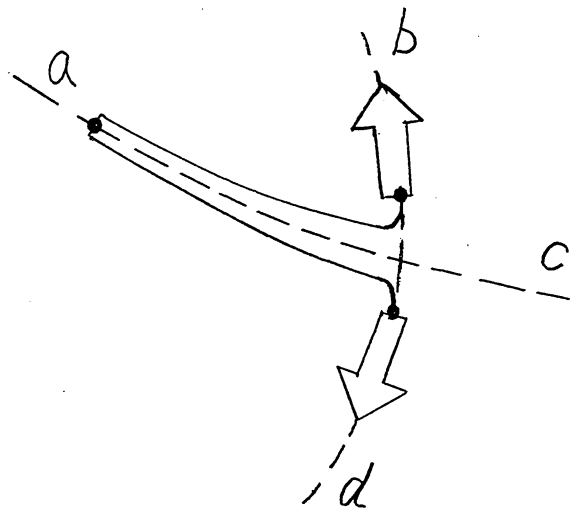


図4 区間演算による曲線追跡法と代数方程式系の根

