数理解析研究所講究録
第722巻 1990年 33-49

悪条件代数方程式と融合アルゴリズム

愛媛大学工学部　野田松太郎　（Matu-Tarow NODA）

1．はじめに

数値計算の幅広い分野への普及とともに、いかに正確な計算を行うかに関する関心が高まっている。容易に考えられる解決策は、計算に数式処理を取り入れ数値計算により生じる誤差を少なくすることである。このような数式処理と数値計算を融合する試みは、提唱はされているものの両者の計算方法やプログラムの稼働システムなどの大きな相違などにより、いまだ一般的ではない[1),2)]。一方、数値計算の枠内で計算精度の劣化を防ぎ数値を区間に拡張して計算する立場に区間演算による自己検証的算法がある[3)]。ここでもPASCAL-SCのようなパソコンでも稼働する道具は開発されているものの応用範囲は限られており、あまり一般的ではない。

本論では、数式処理と数値計算の結合による新しいアルゴリズムの開発とその応用について考察する。特に、ある種の悪条件代数方程式に対し数値計算では良い解（以下、代数方程式を扱うので根という）を得られない場合にも、融合アル

ゴリズムを用いると極めて高精度の根が得られることを示す。この詳細は文献 4)～6) にある。さらにこれらの計算を可能にする計算システムの開発についても言及する。なお、本論の融合アルゴリズム部分は佐々木・野田の文献4、越智・野田・佐々木の文献6の一部をまとめたものであり、システム部分は野田・岩下の文献8を拡張したものである。

2. 数値・数式融合アルゴリズム

（佐々木・野田[4]、越智・野田・佐々木[6]）

代数方程式の求根を考える。主係数 $a_m \neq 0$ で次数（deg）$= m$ の代数方程式を

$$F(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_0 = 0 \qquad (1)$$

とする。(1) が悪条件でないならニュートン法等の数値計算は強力である。しかし、(1) が重根や近接根を含むと、ニュートン法の収束は遅くなるし結果も不安定になる。このような悪条件の場合でも、係数や根が有理数の範囲の重根なら、$G_1, G_2, \cdots, G_k$ を無平方として、

$$F(x) = G_1(x) G_2^2(x) \cdots G_k^k(x) \qquad (2)$$

と無平方分解することによって、根を分離し得ることは明らかである。無平方分解は数式処理を用いて **GCD**（最大公約多項式）演算を行うと得られる。ここに数式処理の割り込む余

地がある。しかし、係数が浮動小数になったり、近接根が存在したりすると、(2)の分解は困難になり数式処理でも十分でない。この場合は(2)のような代数的算法を近似的な代数算法（**近似的代数計算**）に置き換える必要がある。これが数値・数式融合アルゴリズムの出発点である。すなわち**GCD**計算を浮動小数に拡張し**近似的GCD**を求め、それによって近似的無平方分解を達成する。2つの多項式 F(x) と G(x) が近似的共通因子 D(x) を持つなら、それらは $0 < \varepsilon \ll 1$ として

$$F(x) = D(x)\tilde{F}(x) + f(x) = 0 \ , \ f(x) = O(\varepsilon(x))$$

$$G(x) = D(x)\tilde{G}(x) + g(x) = 0 \ , \ g(x) = O(\varepsilon(x)) \qquad (3)$$

と書けるであろう。ここで f(x), g(x) は絶対値最大係数（mmc と略する）が ε 以下の微小係数多項式である。したがって D(x) は精度 ε の共通因子といえる。この中、次数最大のものを**精度εの近似的GCD**といい、GCD(F, G; ε)と表す。悪条件代数方程式に対し、近似的**GCD**を用いて近接根をあたかも重根のように分離し悪条件性を解消することが可能になる。$P_1 = F(x), P_2 = G(x)$として、近似的**GCD**を求めるアルゴリズムは基本的にはユークリッドの互除法であるが、剰余多項式の正則化をして $i = 2, \cdots, k$ に対し、

$$P_{i-1} = Q_i P_i + MAX\{1, mmc(Q_i)\} P_{i+1} \ , \qquad (4)$$

を計算し、多項式剰余列（PRS）を求める。なお、Q_i、P_{i+1}は各々P_{i-1}/P_iの商、剰余である。$\varepsilon \neq 0$であるので、近似的**GCD**があってもPRSは最終的に厳密に0にならない。そこで、多項式の0の判定をかえる必要がある。正則化の演算をPRSを求める各kで行い、係数の絶対値が微小量ε以下のとき、その係数を0とする。この操作をcutoff εという。cutoff εの近似的PRSは

$$(P_1, P_2, \cdots, P_k \neq 0(\text{cutoff } \varepsilon), P_{k+1} = 0(\text{cutoff } \varepsilon)) \quad (5)$$

となる。この近似的多項式剰余列から精度εの近似的**GCD**を次のように定める。

$$GCD(P_1, P_2; \varepsilon) = pp(P_k) \quad (6)$$

近似的**GCD**を計算する場合に、精度εと近接根間の根間距離δとの関係を調べる必要がある。一般に$\varepsilon = O(\delta)$であり、特に$P_2(x) = dP_1(x)/dx$なら$\varepsilon = O(\delta^2)$となることがPRSの解析により知られている[5]。実際にPRSにおいて

$$P_k = \text{const} \times D(x) + O(\varepsilon(x))$$

となるので$\deg(P_k) = \deg(D(x))$なら$P_{k+1} = O(\varepsilon(x))$であることも容易に示し得る。

悪条件代数方程式の求根を行う融合アルゴリズムは、まず精度εの近似的無平方分解

$$F(x) = G_1(x)G_2^2(x)\cdots G_k^k(x) + O(\varepsilon(x)), \quad G_k \neq 0 \quad (7)$$

をする。ここで、多項式 F は正則（主係数・他の各係数の最大値がともに O(1)）であり、因子 G_m は無平方である。(7) の分解は近似的**GCD** を用いて次のように得られる。

アルゴリズム　**近似的無平方分解**

① $P_1 := GCD(F, dF/dx; const \times \delta^2)$; $C_1 := F/P_1$;

② while P_i $(= P_k) \neq 1$ do

for i := 1 to k do

$P_{i+1} := GCD(F, dF/dx; const \times \delta^2)$;

$C_{i+1} := P_i / P_{i+1}$; $G_i := C_i / C_{i+1}$ end;

③ 無平方因子 ← $(G_1, G_2, \cdots, G_{i-1}, C_k)$

ここで、除算は浮動小数係数の多項式の記号除算を意味する。このように近似的無平方分解により、近接根をあたかも重根のように分離することが出来る。近似的重根から近接根を求めるには、F(x) を近似的重根の廻りでその多重度（m）の次数まで Taylor 展開する。こうして得られた Taylor 級数には悪条件性はないので容易に数値計算によって近似的重根の位置（u_0）と各近接根の位置の間の距離を求めることが出来る。そのアルゴリズムは次のように書ける。

アルゴリズム　**近似的GCD から近接根の求根**（近似的重心の位置 = u_0, 多重度 m）

① F(x) を u_0 の廻りで Taylor 展開。ただし、$0 < \delta \ll 1$

とする。

$$(\delta^m/m!)F^{(m)}(u_0)+\cdots+(\delta/1)F^{(1)}(u_0)+F(u_0)=0 \quad (8)$$

ただし、$F^{(k)}(u_0)=d^kF(x)/dx^k|_{x=u_0}$ である。

② ニュートン法により (8) を解き、$\delta_1,\delta_2,\cdots,\delta_m$ を数値的に得る。

③ 各 δ_i に対し、近接根を $u_i \leftarrow u_0+\delta_i$ $(i=1,\cdots,m)$ と求める。

以上が悪条件代数方程式の近似的**GCD**による近接根分離と、近接根計算のための数値数式融合アルゴリズムである。近接根分離では数値計算はあまり高精度である必要はない。しかし、Taylor 級数展開の係数は非常に小さくなることがあり、高精度の数値計算が必要となる。この場合に Taylor 級数の最大、最小の係数の比を変数変換により調節する（正規化）ことが重要になる。

以上の1変数悪条件代数方程式に対する融合算法を多変数悪条件代数方程式に拡張する。この場合は取りあえず代数方程式の系が近似的共通因子を持つ次のようなものに限定する。

$$F=D\tilde{F}+f=0,\quad G=D\tilde{G}+g=0,\quad \cdots \quad (9)$$

ここで、$F,G,\cdots$ は係数が O(1) の多項式、$f, g, \cdots$ は微小係数 (O(ε)) の多項式とする。たとえば2変数の場合、$\{F=0, G=0\}$ は一般に有限個の根しかもたないが、f と g を

0にすると無限個の根をもつ（D = 0 上のすべての点が根となる）。これは（9）の数値解法の不安定性を示唆しており、実際ヤコビアン行列は D = 0 の近傍で 0 に近くなるという悪条件性を持つ。このような代数方程式に通常の反復法に基ずく数値解法を適用するのはやはり不適切である。これに対し、融合算法では少ない反復回数で安定した根を得ることができ、さらに数値計算の初期値の有効な設定法をも与える。

以下に、2変数の場合に対する手法の概略を与える。（9）で与えられる多項式 F と G に対し、f = g = O（ε）なら1変数の場合に確立した方法を多変数に拡張し、（9）のような分解が可能になる。このとき、D は精度εの近似的因子となる。特に D が次数最大であるとき、**精度εの近似的GCD** といい、（gcd(F, G ; ε)）と書く。近似的GCD を用いて、悪条件連立代数方程式（9)を良条件連立方程式に変換する。このため、まず方程式を

$$H \equiv F\,\widetilde{G} - G\,\widetilde{F} = \widetilde{G}\,f - \widetilde{F}\,g = 0, \quad F = 0$$

のように変形し、連立方程式 { F = 0, G = 0 } のかわりに { F = 0, H = 0 } を考える。この方程式は F = 0 のとき

$$\{ F = 0, G = 0 \} \vee \{ F = 0, \widetilde{F} = 0 \}$$

となり、元の方程式の根をすべて含むが、それ以外に { F = 0, $\widetilde{F}$ = 0 } のみを満足する余分な根までも含む。H は F と

Gの微小項のみに比例するので、ほとんどの場合FとGが有していた悪条件性を持たないと期待できる。したがって、{F=0, H=0}においては悪条件性がほとんどの場合に解消されているといえる。この変換は次のように行ない得る。

① 近似的GCD算法によって、Dを求める。

② $\tilde{F}$ = F/D , $\tilde{G}$ = G/D を精度εの数式除算で求める。

③ 数係数の大きさを揃え（正規化）、H = F $\tilde{G}$ - G $\tilde{F}$ を得る。

④ 連立方程式 {F=0, H=0} は、悪条件性を持たないから、通常のニュートン法による数値計算を実行して根を得る。

⑤ 余分な根を除くため、得られた根のチェックをする。

ニュートン法の初期値は、方程式が悪条件性を持たないので、fを無視し{D$\tilde{F}$ = 0, H = 0}の近似値を考察することにより定める。この近似式より余分な根に対する部分を消去するとニュートン法の初期値は

{D=0, H=0}, {$\tilde{F}$ = 0, $\tilde{G}$ = 0}

の連立方程式を粗く解いて決めればよいことがわかる。第一の方程式はD=0の近傍に分布する（通常の方法では解きにくい）根の近似値を与え、第二の方程式は（初期値の選択さえ

よければ）通常の方法でも解き易い根の近似値を与える。ここに述べた初期値の選択法はある種の悪条件連立代数方程式の反復解法での、良い初期値を与える方法として見ることもできる。

他の解法と、ここで述べた融合解法を比較すると

① Sturm の定理を拡張した純粋な代数的解法では計算料が多く時間がかかりすぎる、

② 元の代数方程式へ直接ニュートン法を適用すると初期値の選択に問題があり、すべての根を求めれるとは限らない、また根を得るための反復回数も増大する。

③ 融合算法では D=0 近傍の解きにくい根も得られるし、初期値が良いので反復回数も少ない。

とまとめられる。

3. 融合アルゴリズムによる悪条件代数方程式の求解例

上で求めた融合アルゴリズムを実際の例に当てはめてみる。悪条件代数方程式の例として

$$F(x)=(x-1)*(x-100)*(x-100.001)*(x-200)*(x-1000)=0 \quad (10)$$

を取り上げる。まず、近似的無平方分解を行う。

$P_1 \leftarrow F(x)$, $P_2 \leftarrow dF(x)/dx$, $\varepsilon \leftarrow 0.0001$ として近似的

GCD を求めると

$P_3 = -476.4025*x^3+158527.5389*x^2-15121402.3092*x$

$+4032662 38.9734$

$P_4 = -13981.9996*x^2+2949804.1275*x-155160492.8987$

$P_5 = 579790.0304*x-57979292.9453$

$P_6 = 0$ (cutoff 0.0001)

の多項式剰余列が求められるので

GCD (P_1, P_2, 0.0001) = 579790.0304*(x - 100.0005)

となり、無平方因子として

$Q_1 = x^3 - 1201*x^2 + 201199.99999*x - 199999.99956$

$Q_2 = x - 100.0005$

を得、F(x) は近似的に $F(x) = Q_1*Q_2^2$ と分解される。$Q_1 = 0$ の数値計算は容易で

$x_1 = 1.0000000000$, $x_4 = 200.00000000$,

$x_5 = 1000.0000000$

を得る。Q_2 は近似的重根であることを示し、$u_0 = 100.0005$, m = 2 である。m 次までの Taylor 展開は

$8909995.5*x^2 + 0.00225306*x - 2.22743416$

となる。係数の大きさを揃えるため z = 1000*x の変数変換により正規化すると

$8.9099952*z^2 + 0.000002253 06*z - 2.2274316 = 0$

を得る。正規化をすると例え単精度の FORTRAN 計算でも

$z_1 = -0.49999$ ($\delta_1 = -0.0004999$),

$z_2 = 0.49999$ ($\delta_2 = 0.0004999$)

が求まる。よって

$x_2 = u_0 + \delta_1 = 100.00000001$,

$x_3 = u_0 + \delta_2 = 100.00099999$

と精度良く根を求めることが出来る。

次に2変数悪条件代数方程式に対する融合アルゴリズムの使用例を見る。方程式を

$$F = (x^2+y^2-1)*(x*y-0.25)+0.0001*x*y$$

$$G = (x^2+y^2-1)*(x-y)-0.00001*(x+1) \qquad (11)$$

とする。近似的GCDを求めるための多項式剰余列 ($P_1, P_2, \cdots, P_k \neq 0, P_{k+1} = O(\varepsilon)$) は

$P_1 \leftarrow F(x)$, $P_2 \leftarrow dF(x)/dx$, $\varepsilon \leftarrow 0.0001$ とすると

$$P_3 = x^2*y^2-0.25*x^2+0.00011*x*y+y^4-1.25*y^2+0.00001*y$$
$$+0.25$$

$$P4 = 0.0001*x*y^4-0.00001*x*y^3-0.00002249*x*y^2$$
$$+0.0000025*x*y-0.00000063*x+0.00011*y^5$$
$$-0.0001375*y^3+0.000025*y^2+0.0000275*y-0.00000063$$

P_4 と P_3 の最大係数の比は $O(10^{-4})$ であり、P_3 が精度 10^{-4} の近似的GCDとなる。P_3 の係数を 10^{-4} より若干大きな値で

丸め、原始的部分（pp）を取り、

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

を得る。Dによる精度O(10^{-4})の除算で容易に以下が求まる。

$\tilde{F} = xy - 0.25$, $\tilde{G} = x - y$,

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

表1に結果を示す。ニュートン法の計算は１００組の乱数を初期値として発生させた。D=0 近傍の根の中、NO.5 の根へは1回も収束しない。通常はこの根は見忘れられることになる。また通常の計算でも容易に得られる No.7,8 の根以外への収束するまでの平均反復数も多い。しかし、融合アルゴリズムを用いた計算ではこれらの困難は解消していることがわかる。ここで開発した融合アルゴリズムを（11）を3変数の場合に拡張した悪条件代数方程式に適用すると、悪条件性を解消しないままの数値計算結果は極めて不十分であることが、より明かとなる。結果は文献 6）にある。

4. ハイブリッド計算システム

以上の計算を有効に遂行するためには、それに適したハイブリッド計算システムの完成が望まれる。数式処理システムで FORTRAN出力を行う方向は MACSYMA、REDUCE 等で実現されているが数式処理と数値計算の結合は強力ではない。MAPLE

や MATHEMATICA のように C を開発言語とするシステムでは、浮動小数係数の数式を扱うことも出き、融合計算もそれなりに可能である。一方、GAL[7]では FORTRAN との結合をＯＳの一部を手直しすることによって可能にする方向が示され、より強力なハイブリッドシステムが期待されている。SYNC[8]では簡単な中間言語を設定することにより、FORTRAN との結合を可能にしている。これと似た方法で、SYNC の数式処理と区間演算用の PASCAL-SC の結合も可能になったので、以下この点に付いて簡単に述べる。

SYNC は開発言語に PROLOG を用い、基本的なデータ構造の構成や制御を C で記述したパソコン対象のシステムで、主記憶 640KB の IBM-PC/AT 上で稼働する。SYNC の FORTRAN との融合は

① FORTRAN に類似の中間言語を利用者が作成しておく。ここでは、SYNC からの記号入力、SYNC への数値出力に対応する文（JOINT文）が FORTRAN の文に加えられている。これに属性 HYB をつけファイルに格納する。JOINT文は２つのブロック IN と OUT からなり、COMMON 文と同様の書式を持つ。

（例）　JOINT /IN/ F(X), FD(X), A /OUT/ Z

② SYNC の命令 fortran によって自動的に JOINT 文の IN

ブロック内の数式を FORTRAN の文関数に変換し、FORTRAN のコンパイル、リンク、実行をする。

③出力結果は /OUT/ ブロックを通じ SYNC に戻され、SYNC の変数として用いる。

のステップでなされる。しかし、当然ながら FORTRAN 計算では場合によっては誤差の問題が生じる可能性もあり取扱に注意を要する。一方、PASACL-SC との融合を実現すると適用範囲は制限されるが、精度保証については考慮がシステム内で払われている。精度を要求するような問題では数式処理と精度保証付き数値計算との融合はパソコン版究極的な科学計算システムといえるだろう。SYNC と PASCAL-SC の融合は基本的には FORTRAN とのものと同じだが、PASCAL には COMMON 的な文が無い。また文関数が無いため、/IN/ に定められた数式をすべて別個の関数として宣言する必要がある。PASACAL の基本的な構造を損することなく融合するために、HYB ファイルは、/in/ ブロックと /out/ ブロックを持つ次の例のような、ある種の注釈行を含む。

```
{ /in/ f(x),fd(x),p(y:integer):real,a:real /out/ z}
```

上で f, fd, x のように型指定の無いものは interval の型を持つとする。以下、これを受けて PASCAL-SC の起動以下は SYNC の命令 pascal で自動的に行なう。

このようなシステムを用いて、融合アルゴリズムの中心をなす多項式剰余列の計算のより高速化がはかられれば、さらに幅広い問題が近似的代数計算の枠内にはいることと期待される。

表 1

No	代数的計算で得た根 (x, y)	ニュートン法		融合アルゴリズムによる計算	
		収束回数	平均反復回数	根の値 (x, y)	反復回数
1	0.9990732, 0.0432840	23	23.74	0.9990732, 0.0432840	2
2	-0.0262280, 0.9996512	13	23.08	-0.0262280, 0.9996512	2
3	-1.0000000, 0.0	11	16.91	-1.0000000, 3.428D-22	1
4	-0.0227383, -0.9997464	22	17.64	-0.0227384, -0.9997464	2
5	0.6645091, 0.7471453	0	—	0.6645091, 0.7471453	3
6	-0.7141434, -0.6998564	7	16.57	-0.7141434, -0.6998564	3
7	0.5000350, 0.5000650	17	5.52	0.5000350, 0.5000650	2
8	-0.5000550, -0.5000450	7	6.00	-0.5000550, -0.5000450	2

【参考文献】

1)三井斌友：数値処理と数式処理の界面，情報処理学会誌，27巻 4号，pp.422-430,(1986).

2)伊理正夫、土谷隆、星守:偏導関数計算と丸め誤差推定の自動化と大規模非線形方程式系への応用,情報処理学会論文誌,26巻,11号,pp.1411-1420(1985).

3)L.B.Rall:An Introduction to the Scientific Computing Language Pascal-SC: Trans.Second Army Conference on Applied Mathematics and Computing, pp.117-148(1985).

4)T.Sasaki and M.T.Noda: Approximate Square-free Decomposition and Root-finding of Ill-conditioned Algebraic Equations, J.Inf.Proc.12,pp.159-168(1989).

5)T.Sasaki and M.Sasaki: Analysis of Accuracy Decreasing in Polynomial Remainder Sequence with Floating-point Number Coefficents, J.Inf.Proc.12, No.4(1990).

6)M.Ochi, M.T.Noda and T.Sasaki: Approximate GCD of Multivariate Polynomials and Application to Ill-conditioned System of Algebraic Equations,submitted.

7)M.Suzuki and T.Sasaki: A Hybrid Algebraic-Numeric System ANS and Its Preliminary Implementation: Lect. Notes Comp. Sci. vol.378, pp.163-171(1989).

8)野田松太郎,岩下英俊:パーソナルなハイブリッド処理システム SYNC の設計,情報処理学会論文誌,30巻,4号,pp.419-426(1989).