解　説

4. 数式処理と数値計算の融合による精度保証†

野田松太郎††　佐々木建昭†††　鈴木正幸†††

1. はじめに

数値計算の幅広い分野への普及とともに，計算誤差の問題克服の重要性が認識されるようになり，区間解析に代表される精度保証付き計算が見直されている．単純に考えれば，計算を数式処理で正確に実行することによって，正解あるいはそれに近い近似解を見い出し得ると考えられる．しかしながら，数値計算に比べると，数式処理の算法は著しく融通性に欠け，適用範囲が限られる．そこで，数式処理と数値計算を融合して，両者の長所を生かし短所を補えば，有効な算法を構成できると考えられる．このような融合化は，提唱はされているものの，両者の計算方法の大きな相違などにより，具体的な問題に対して十分にその良さを示すに至っていない．本稿では，まず数値計算と数式処理の融合化の試みを概説する．次に，われわれが最近開発したいくつかの融合アルゴリズムを示し，ある種の悪条件代数方程式に対し，融合アルゴリズムを用いるときわめて高精度の解が得られることを示す．

2. 数値計算と数式処理

数値計算の幅広い分野への普及とともに，計算誤差の困難を克服する手法の開発が重要な課題になってきた．現在のところ，二つの方向が考えられている．第1は四則演算ごとに誤差限界を算定する区間演算の利用であり，第2は誤差のない計算を基本とする算法の利用である．本特集の多くは前者を扱っているが，本稿は後者について述べる．

数式処理は基本的には正確に計算することを旨としており，その立場で全過程を数式処理のみで計算すれば誤差の困難は一切ない．しかし，この立場では，整数や有理数あるいは近似なしの数式を扱うため，計算途中で数や式が非常に大きくなる（中間式膨張）などの非能率性と適用範囲の限定が問題になる．そこで，数値計算と数式処理を融合し，前者の効率性と融通性（適用範囲の広さ）および後者の正確性を生かして，有用かつ効率的な計算を行う試みが古くから提唱されている[1]．これらの多くは「数式処理の環境下での数値計算」を目指しており，数式処理システムへの数値計算の取り込みが中心である．REDUCE で高精度の浮動小数演算を行う[2]，FORTRAN のソースプログラムを出力する[3]，などなどの機能の付加はこの流れに属する．

これとは逆に，数値計算の前処理として数式処理を使う例がさまざまな応用分野で増えている．たとえば，次の多項式の計算を見よう[4]．

例1

$$p_n(x,y,z)=(x^2-1)(y^2-4)(z^2-9)$$

$$p_e=x^2y^2z^2-4x^2z^2-y^2z^2-9x^2y^2+36x^2+9y^2+4z^2-36$$

p_n は p_e の展開形であり，明らかに任意の x,y に対して $z=3$ のとき，$p_n=p_e=0$ となる．表-1 は $x=11111$, $y=22222$, $z=3$ に対する p_n と p_e の計算を FORTRAN（表では F と表す），PASCAL-SC[5]，および数式処理で計算したものである．数式処理以外では p_e の計算は意味をもたないことが分かる（FORTRAN と PASCAL では，整数のオーバフローが起きるため，答がでたらめになるのである）．しかし，数式処理による前処理で p_e を因数分解して p_n 形にしてお

表-1

	p_n	p_e
F 整数	0	2104
F 実数	0.0	5.26392E+5
F 倍精度実数	0.0	5.263920…0D+5
PASCAL-SC	[0, 0]	[−3.288…E+6, 2.711…E+6]
数式処理	0	0

† Validated Computations by Symbolic/Numeric Hybrid Algorithms by Matu-Tarow NODA (Ehime University, Faculty of Engineering), Tateaki SASAKI and Masayuki SUZUKI (The Institute of Physical and Chemical Research).

†† 愛媛大学工学部情報工学科
††† 理化学研究所情報科学研究室

けば FORTRAN でも正しく計算できる（もちろん，今の場合，前処理として因数分解するのはバカげているが）．なお，文献 4) はこの不都合を解消するため，項の計算順序の変更とともに，区間演算を適用していかに正しい結果を求めるかの研究を記述している．

上記のような，数値計算と数式処理の単純な結合形態を少し進めれば，新しいパッケージの作成につながる．パッケージは常微分方程式に対して多く作成され，国内でもいくつか開発が進められた[6]．たとえば，文字処理可能な逐次形言語（たとえば Turbo Pascal）で，FORTRAN コードを生成するパッケージ[7]とか，数値微分を記号微分に置き換えて高い精度の計算を可能にするパッケージ[8),9)]，などである．

数式処理システムも，MACSYMA や REDUCE などの伝統的なものに頼るばかりでなく，数値計算と結合し，ユーザインタフェースの改良を目指した新しい形態のシステムが登場している．その代表例が MAPLE[10] と MATHMATICA[11] である．これらは記述言語として伝統的な LISP でなく，C を採用している．FORTRAN コードの生成は当然として，計算結果のグラフィック表示などに際だった特徴を有している．国内の数式処理システムの開発（事例も研究者数も極端に少ないが）も，やはり数値計算との結合を重視している．OS レベルでの FORTRAN との結合を意図する GAL[12]，パソコン上で FORTRAN や PASCAL-SC との結合を実現している SYNC[13]，その他である．

以上はいずれも数式処理と数値計算の一方が他方を取り込み，利用することであり，この種の融合化は比較的研究も進み，利用範囲も広まりつつある．

一方，数式処理と数値計算をアルゴリズムレベルで融合し，新しい形のアルゴリズムを開発しようという試みが最近なされている．これは狭義には，数式処理で扱われてきた厳密な代数計算を，数値計算で扱う浮動小数を含む計算に拡張しようというものであり，**近似的代数計算**と名付けられている．1988年，重根や近接根をもつ悪条件代数方程式の有効な解法として発表され，より広範な演算へと拡張されつつある[14]．このような融合アルゴリズムを用いると，従来の数値計算では良い解を得られないいくつかの問題に対して，きわめて高精度の解が得られることが分かった．このレベルに至って初めて，数値計算と数式処理の融合は本物といえるであろう．

以上，数値計算と数式処理の融合を 4 つに分類して解説したが，前 3 者はいずれも詳しい説明なしでも大雑把に理解してもらえると思う．説明を要するのはアルゴリズムレベルの融合による精度向上であるが，それは本特集のテーマでもある．したがって，以下では近似的代数算法による精度向上に話を限定する．**3.** では，近似的代数算法のなかで最も分かりやすい**近似的 GCD 算法**と，その一変数悪条件代数方程式への応用について述べる．**4.** では，近似的 GCD 算法の多変数への拡張と，ある種の悪条件連立方程式への適用を説明する．そして，これらの融合算法が有効であることを実際の計算例で示す．最後に，数値数式融合アルゴリズムのさらに広い分野への応用の可能性について言及する．

3. 一変数悪条件代数方程式と近似的 GCD 算法

3.1 代数方程式の悪条件性と数式処理の算法

一変数代数方程式

$$P(x)=a_n x^n+a_{n-1}x^{n-1}+\cdots+a_0=0,\ a_n\neq 0 \tag{3.1}$$

を考える．ここで，n を多項式の次数，a_n を主係数といい，それぞれ $\deg(P)$，$\mathrm{lc}(P)$ と表す．すでに述べたように，通常の数式処理で扱われる多項式は係数 a_i が整数か有理数の場合に限られる．そのため，数式処理では計算途中で係数の桁数が巨大になり速度の低下を招きやすい．したがって，代数方程式の実用的解法は通常，ニュートン法，あるいはその変形アルゴリズムに基づく数値計算に限られる．この場合，係数の変化による根の変動の大きさが問題になる．方程式によっては，わずかの係数の変化が大きな根の変動をもたらす．いま，k 番目の係数の変化 $a_k\to a_k+\Delta a_k$ が i 番目の根 x_i にもたらす変動を Δx_i とする．両者は，s_i を根 x_i の多重度として，

$$\Delta x_i \fallingdotseq -\left\{\frac{s_i!\,x_i^{\,k}\Delta a_k}{(d/dx)^{s_i}P(x_i)}\right\}^{1/s_i}$$

なる関係をもつことが知られている[15]．すなわち，代数方程式 (3.1) の根の多重度が 1 でなくなると根の計算精度が低下する．さらに，重根をもつ場合に限らず，非常に接近した根（近接根）をもつ場合でも Δx_i が大となる．このような場合を**悪条件**と呼ぶ．方程式が悪条件である場合には数値解法はきわめて非能率になったり，不安定になったりする．

代数方程式が重根をもつ場合には，数式処理で無平方分解を行って重根部分を分離し，重根を含まない方

程式に変換できる．ここで，多項式Pの無平方分解とは以下を意味する．

【無平方分解】 多項式Pが多重度2以上の因子を含まないとき，P は無平方 (Square-free) であるといい，Pを無平方多項式の積に分解することを無平方分解という．すなわち，各因子 Q_m を無平方として，

$$P(x)=Q_1(x)Q_2^2(x)\cdots Q_k^k(x)$$

なる $Q_1, Q_2, \cdots, Q_k$ を決定することである．

この算法は記号微分と最大公約多項式 (GCD) 算法のみで容易に実現され，数式処理では代表的な算法である（具体的には**3.3**を見よ）．m 重根部分は Q_m に含まれる．多項式の係数が有理数なら，GCD は次の**ユークリッドの互除法**により容易に計算できる．

アルゴリズム ユークリッドの互除法
入力 有理係数多項式：$P_1, P_2, \deg(P_1)\geqq\deg(P_2)$
出力 P_1, P_2 の GCD：$\gcd(P_1, P_2)$
方法
```
i:=2;
while P_i≠0 do
begin
  P_{i-1}:=Q_iP_i+P_{i+1}
  i:=i+1
end;
return pp(P_{i-1})
```
アルゴリズム中の pp は多項式の原始的部分を取り出す演算である．

上記の各 Q_m は重根をもたないので，数式処理により代数方程式の悪条件性は簡単に解消しそうに思えるが，実はそうではない．まず，数式処理の場合，計算を正確に行うとすると係数が整数あるいは有理数に限られるが，現実問題では浮動小数が頻繁に現れる．そのような浮動小数を有理数化してもよいが，そうすると計算途中で巨大な桁数の有理数を扱わなければならない．そうしてもなお，無平方因子 Q_m が近接根をもつ場合には悪条件性は解消されない．すなわち，従来の数式処理法のままでは，問題は一部しか解決されないのである．

多項式の係数が浮動小数の場合，誤差のために厳密な重根が存在するとは考えにくく，むしろ近接根の扱いが重要になる．そこで，GCD 算法を浮動小数係数の多項式に拡張し，近接根をあたかも重根のように分離しうる**近似的無平方分解**のアルゴリズムを構成すれば，重根のみか近接根を含む代数方程式の悪条件性を解消する有効な算法が作られるであろう．以下では，**近似的 GCD 算法**がどう構成されたかを解説する．

3.2 精度 ε の近似的 GCD

われわれの目的は，微小正数 δ を与えて，**根間距離**がほぼ δ 以下の根を近似的重根として分離することである．ところで，近接根は最大根間距離に相対的な概念であるから，近接根を議論する際は最大根間距離を1程度に規格化するのが妥当である．これは $P(x)$ を以下に述べるように**正則化**すればよい．

【正則多項式】 $\mathrm{lc}(P)=O(1)$ かつ $\max\{|a_k|, \cdots, |a_0|\}=O(1)$ or 0 のとき，$P(x)$ は正則であるという．

同様に，$Q(x)=b_m x^m+\cdots+b_0, b_m\neq 0$, に対し，$\mathrm{lc}(P)=O(1)$，$\mathrm{lc}(Q)=O(1)$，かつ $\max\{|a_{m-1}|, \cdots, |a_0|, |b_{m-1}|, \cdots, |b_0|\}=O(1)$ or 0 のとき，$\{P, Q\}$ は正則であるという．//

ここで，$O(1)$ は通常のランダウの記号とは異なり「だいたい大きさが1である」ことを表す．なお，正則でない多項式Pは適当なスケール変換 ($P\to\xi P, x\to\eta x$) で正則になる．このため，正則な多項式のみを扱っても一般性を失わない．

以下，便宜上，最大係数なる概念を導入しておく．

【最大係数】 $P(x)$ の絶対値最大の係数の絶対値をいい，$\mathrm{mmc}(P)$ と表す．//

【精度 ε の近似的 GCD】 $0<\varepsilon\ll 1$, $\{P_1, P_2\}$ は正則，かつ P_1 と P_2 が以下を満たすとする．

$$P_1(x)=D(x)\tilde{P}_1(x)+O(\varepsilon(x)),$$
$$P_2(x)=D(x)\tilde{P}_2(x)+O(\varepsilon(x)) \qquad (3.2)$$

ただし，$O(\varepsilon(x))$ は最大係数が ε オーダの微小係数多項式である．(3.2)を満たす $D(x)$のなかで次数が最大のものを**精度 ε の近似的 GCD** といい，$\gcd(P_1, P_2; \varepsilon)$ と表す．//

一般に $\gcd(P_1, P_2; \varepsilon)$ は無限個存在し，一意的ではない．しかし，特定の近似的 GCD はユークリッドの互除法を浮動小数にまで拡張して求めることができる．

次に，精度 ε と互いに近接した根の間の距離 δ との関係が必要である．これも解析されており（文献 14) と 16) 参照），ここでは結果のみを述べる．

【定理】 $\{P_1, P_2\}$ は正則で，P_1 と P_2 は(3.2)を満たすとする．P_1 と P_2 に特別な関係がない場合，$\varepsilon=O(\delta)$ となり，$P_2\propto dP_1/dx$ の場合には $\varepsilon=O(\delta^2)$ となる．

最後に，ユークリッドの互除法を用いて GCD を計算するのであるが，浮動小数係数の場合には新たな問題が生じる．係数が有理数の場合，GCD であるか否かは剰余列があるところで0になることから判定でき

るが，近似的 GCD の場合には 0 にならない段階で判定しなければならない．では，剰余列があるところで 0 に近くなれば近似的 GCD かというと，そうとは限らない．浮動小数係数多項式の場合，剰余に任意の数をかけてもよく，定数倍の不定性があるからである．しかも，われわれは剰余列の係数の大きさから，近似的 GCD の精度 ε を知りたい．そのためには，剰余列の定数倍の不定性を除く必要がある．これはこれまで扱われたことがない問題であり，以下のように解決された．

【浮動小数係数多項式剰余列の規格化】 剰余列を $(P_1, P_2, P_3, \cdots)$ とする．P_{i-1} と P_i の商を Q_i とするとき，剰余 P_{i+1} の数係数を以下のように規格化する

$$P_{i-1}=Q_iP_i+\max\{1,\ \mathrm{mmc}(Q_i)\}\times P_{i+1}. \quad // \tag{3.3}$$

実際の計算では，係数がマシンイプシロン（これを ε_M と表す）程度になった場合の扱いも考慮しなければならない．それについては文献 14) あるいは 16) を参照されたい．

上記の規格化から次の性質が導かれる．

（1） P_1 と P_2 の係数に含まれる誤差が $O(\varepsilon_M)$ ならば，剰余列全体に含まれる誤差も $O(\varepsilon_M)$ である．このことは，(3.3)を書き直した式

$$P_{i+1}=(P_{i-1}-Q_iP_i)/\max\{1,\ \mathrm{mmc}(Q_i)\}$$

から分かる．すなわち，P_{i+1} は P_{i-1} と P_i に係数の絶対値が 1 以下の多項式をかけて加えたものだからである．

（2） $\mathrm{lc}(P_i)\ll\mathrm{mmc}(P_i)$ の場合，$\mathrm{mmc}(Q_i)\gg 1$ となることがあり，P_{i-1} と P_i の剰余の係数は非常に大きくなりうる．しかし，(3.3)で規格化された P_{i+1} は，$\mathrm{mmc}(P_{i-1})\lesssim\mathrm{mmc}(P_i)$ を満たしており，剰余列計算に数値的不安定性をもたらさない．

（3） P_1 と P_2 の近接根の近接度の最大値が δ で，P_k はこれらの根を含む近似的 GCD とする．このとき，$\{P_1, P_2\}$ が正則なら，$\mathrm{mmc}(P_k)\leqq O(1)$ で（ほとんどの場合，$\mathrm{mmc}(P_k)\simeq O(1)$ となる），かつ $\mathrm{mmc}(P_{k+1})=O(\delta)$ $(P_2\propto dP_1/dx$ のときは $O(\delta^2))$ となる．//

以上の性質から，多項式剰余列を式(3.3)で計算することにより，P_1 と P_2 がどの程度の近接根をもっているかが容易に判定できる．

例 2【規格化された多項式剰余列】

$$\begin{cases}P_1=(x-0.5)(x-0.502)(x+1)(x-2)(x-1.5)\\ P_2=(x-0.501)(x-0.503)(x-1)(x+2)(x+1.5)\end{cases}$$

$$P_3=-4.998x^4+5.013997x^3+4.7414925x^2$$
$$\quad -6.0174985x+1.509009$$
$$P_4=0.697880794x^3-0.701037162x^2$$
$$\quad +0.178930204x-0.0014439101$$
$$P_5=0.840067492x^2-0.841442765x$$
$$\quad +0.210704053$$
$$P_6=0.00187196957x-0.000938795693$$
$$P_7=-1.39801471\times 10^{-9}$$

P_1 と P_2 は $x=0.5015$ 近くに近接根をもち，その近接度は 0.001 である．$\mathrm{mmc}(P_6)\approx 10^{-3}$ ゆえ，P_5 が精度 $\approx 10^{-3}$ の GCD であることが分かる．実際，P_5 は 2 次で，その根は二つとも $x=0.501$ の近傍にある．

以上の算法をみると，代数的算法に浮動小数をもちこんで算法の適用範囲を拡張する場合，旧来の算法のままではたいてい駄目で，数値解析で行っているような誤差解析が必要であることが分かる．

さて，代数的算法は，数値算法と異なり，正確な計算を基礎としているから，正確な計算でなければ扱いにくい（数値算法にとって）悪条件な問題に対して，有効な解法となりうるのでは，との期待を抱かせる．以下，この期待がはずれていないことをいくつかの例で示そう．

3.3 近似的無平方分解と根間距離の決定

精度 ε の近似的無平方分解を次のように定める．

$$P(x)=Q_1(x)Q_2^2(x)\cdots Q_k^k(x)+O(\varepsilon(x)). \tag{3.4}$$

ここで，$P(x)$ は正則，各近似因子 Q_m は無平方，ε は正の微小量とする．

$$dP(x)/dx=Q_2(x)\cdots Q_k^{k-1}(x)\tilde{P}(x)+O(\varepsilon'(x)). \tag{3.5}$$

ただし $\gcd(P, \tilde{P})=1$，ε' は ε と同程度の微小量である．すなわち，$Q_2\cdots Q_k^{k-1}$ は P と dP/dx の精度 ε の近似的 GCD となる．したがって，近似的 GCD 算法を繰返し使用することにより，通常の無平方分解と同様に，近似的無平方分解が実行できる．

ここで，正確な無平方分解と近似的無平方分解では，最良な算法が異なることを注意しておく．正確な無平方分解の場合，まず $\gcd(P, dP/dx)=Q_2\cdots Q_k^{k-1}$ を計算し，P をこの式で割って $Q_1Q_2\cdots Q_k$ を計算する．以後はこの式を用いれば，P を何度も微分して GCD を計算する算法より効率がよい（これは 1 変数の場合の話であって，多変数の場合にはより効率的な算法がある）．これに対し，近似的算法の場合には，P を何回も微分しつつ，$Q_2\cdots Q_k^{k-1}, Q_3\cdots Q_k^{k-2}, \cdots$ と順に計算するのがよい．なぜなら，後者の算法では常

に $\gcd(\tilde{P}, d\tilde{P}/dx)$ なる形の計算が行われ，近接根の近接度を δ とするとき，この計算は $O(\delta^2)$ の精度で実行できるからである（前者の算法の精度は $O(\delta)$ である）．近似的算法の場合，精度 ε の値が大きいと，正確な計算から予想される答えと異なる答えが得られることがあるので，注意が必要である．

例3【近似的無平方分解 …$\varepsilon=\delta^2$ で剰余列を打ち切る】

$$P(x)=(2x^2-1)(x-29.0/41.0)(x-70.0/99.0)$$
$$=2x^4-2.8039072x^3+0.0002463661x^2$$
$$+1.4143878x-0.50012318$$

近接度 $\delta=0.01$ 以下を重根とみなす場合

$$Q_3=x-0.707164833$$
$$Q_1=x+0.707106719$$

近接度 $\delta=0.0001$ 以下を重根とみなす場合

$$Q_2=x-0.707094337$$
$$Q_1=x^2-0.000199106643x-0.500140792$$

次に，近似的無平方分解を用いた近接根（重根を含む）の分離算法を説明する．前章に述べたごとく，正則多項式 $P(x)$ が近接度 δ, $0<\delta\ll 1$, の因子をもてば，それらの因子を近似的多重因子とみなして，(3.4)のように表現できる．このとき，$\varepsilon=O(\delta^2)$ であることが文献14）に示されている．

さて，$P(x)$ が(3.4)のように近似的無平方分解されたとしよう．この分解における近似的 GCD 算法から精度 ε の大きさが分かり（多項式剰余列の係数の大きさから分かる），近接度 δ は $\delta\approx\sqrt{\varepsilon}$ と計算できる．$P(x)$ が $x=\alpha$ の近傍に近接度 δ 以下の互いに近接した根を m 個もつならば（この状況を多重度 m の近接根ということにする）．$Q_m(x)$ は $x=\alpha$ の付近に根をもつ．しかも $Q_m(x)$ は近接度 δ 以下の根をもたないから，δ の値が極端に小さくならないところで多項式剰余列を打ち切っておけば，$Q_m(x)=0$ は数値的に良条件方程式であり，通常の数値算法で困難なく解ける．

$Q_m(x)=0$ を解き，根 α_0 を計算したとしよう．$P(x)$ は $x=\alpha_0$ の近傍に m 個の近接根をもつから，$P(x)=0$ のかわりに

$$P(\alpha_0)+P^{(1)}(\alpha_0)(x-\alpha_0)/1!+\cdots$$
$$+P^{(m)}(\alpha_0)(x-\alpha_0)^m/m!=0 \qquad (3.6)$$

を解けば，$P(x)$ の根の近似値（一般に m 個）が得られる．ここで $P^{(i)}(\alpha_0)=d^iP(x)/dx^i|_{x=\alpha_0}$ である．(3.6)は正則でないが，

$x=\alpha_0+\delta z$, δ は近接根の近接度，

とおけば，z の多項式はほぼ正則となり，解きやすい．

以上のことを次の例で説明しよう．

$$P(x)=(x-0.99)^2(x-1.02)(x-2)$$

近似的無平方分解で $P(x)\approx(x-1.0)^3(x-2)$ が得られたとしよう．($\alpha_0=1.0$). $P(x)$ を $x=1.0$ において3次までテイラー展開することにより，式(3.6)として

$$z^3+0.03z^2-2.98z-2.0=0$$

を得る．ただし，$x=1.0+\delta z$, $\delta=0.01$, とおいた．この式を再び近似的無平方分解して解くことにより，大雑把な根として

$z\approx-1.0018$（近似的重根），$z\approx1.9825$（単根）

を得る．前者は $x=0.99$ の重根に対応し，後者は $x=1.02$ の単根に対応する．すなわち，$x=1.0$ の近傍の近接根が分離できた（$x=0.99$ の近傍の二つの根については，$P(x)$ をさらに $x=1.0-0.010018$ の近傍で2次までテイラー展開して，より詳しく調べることになる）．

算法をまとめると，近似的無平方分解で，近接根の位置と多重度および近接度を知り，近接根のある場所を局所的に“拡大”して，根を精確に解くのである．

4. 多変数の悪条件代数方程式の解法

3. で示した近似的 GCD 算法は容易に多変数へ拡張できる．本章でその概略を述べ，ある種の悪条件連立代数方程式への応用を示す．

4.1 多変数多項式の近似的 GCD

多項式 F と G は主変数が x，従変数が $y,\cdots,z$ の多変数多項式とする．F の数係数のなかで絶対値最大の係数の絶対値を最大係数といい，$\mathrm{mmc}(F)$ と表す．$\mathrm{mmc}(F)=1$ であるとき，F は規格多項式という．

F と G は規格多項式，ε は微小正数とする．

$$F(x,\cdots,z)=D(x,\cdots,z)\tilde{F}(x,\cdots,z)+\Delta F(x,\cdots,z)$$
$$G(x,\cdots,z)=D(x,\cdots,z)\tilde{G}(x,\cdots,z)+\Delta G(x,\cdots,z) \qquad (4.1)$$

ただし，$\mathrm{mmc}(\Delta F)=O(\varepsilon)$, $\mathrm{mmc}(\Delta G)=O(\varepsilon)$, であるとき，$D$ を精度 ε の近似的因子という．特に D が次数最大であるとき，精度 ε の近似的 GCD といい

$$D=\gcd(F,G;\varepsilon)$$

と表す．//

多変数の場合，主変数に関する係数が多項式であるから，content（係数部の GCD）の計算法など面倒な部分もあるが，一番問題となるのは一変数の場合と同

様，剰余の規格化である．多変数多項式の剰余列に関してはエレガントな部分終結式理論が完成しており，それに基づいて計算すればよさそうに思えるが，そうではない．例を示そう．

例 4【部分終結式算法による多項式剰余列】

$$P_1=F=(y^2+1)x^3+(y^2+1.001)x^2+(2y+1),$$
$$P_2=G=(y^2-1)x^3+(y^2-1.001)x^2+(y-3),$$
$$P_3=(y^2-1)P_1-(y^2+1)P_2$$
$$=(0.002y^2)x^2+(y^3+4y^2-3y+2),$$
$$P_4=[(0.002y^2)^2P_2-Q_3P_3]/(y^2-1)$$
$$=-(0.002y^2)(y^3+4y^2-3y+2)x$$
$$-(0.002y^2)(y^3+4y^2-3.003y+2.002),$$

ただし，

$$Q_3=(0.002y^2)(y^2-1)x+(0.002y^2)(y^2-1.001),$$
$$P_5=[(0.002y^2)^2(y^3+4y^2-3y+2)^2P_3-Q_4P_4]/(0.002y^2)^2$$
$$=y^9+12.002y^8+39.016y^7-1.98001y^6$$
$$-69.04y^5+168.05006y^4-159.02404y^3$$
$$+102.00801y^2-36y+8,$$

ただし

$$Q_4=-(0.002y^2)(y^3+4y^2-3y+2)x$$
$$+(0.002y^2)(y^3+4y^2-3.003y+2.002).$$

上例では，$\gcd(P_1,P_2;10^{-3})=1$ であるが，$\mathrm{mmc}(P_4)\approx O(10^{-3})$, $\mathrm{mmc}(P_5)\approx O(10^{-2})$ となり，剰余列の係数の大きさからは，どれが近似的 GCD か判定できない．

規格化の問題は，1 変数の場合を拡張して，以下のように解決された．

【剰余の規格化】 多変数の場合，F と G より擬剰余 $\tilde{R}$ は

$$\tilde{R}=\mathrm{lc}(G)^{d+1}F-\tilde{Q}G,\quad \deg(\tilde{R})<\deg(G)$$

と計算される．ただし，$\tilde{Q}$ は擬商を表し，$d=\deg(F)-\deg(G)$ である．このとき，多項式剰余列の剰余を以下のように規格化する．

$$R=\tilde{R}/\max\{\mathrm{mmc}(\mathrm{lc}(G)d+1),\ \mathrm{mmc}(\tilde{Q})\}.\ //$$

すなわち，R を F と G の線形結合で表したとき，その係数多項式の最大係数を 1 に規格化するのである．例を示そう．

例 5【規格化された多項式剰余列】

$$P_1=F=(y^2+1)x^3+(y^2+1.001)x^2+(2y+1),$$
$$P_2=G=(y^2-1)x^3+(y^2-1.001)x^2+(y-3),$$
$$P_3=(0.002y^2)x^2+(y^3+4y^2-3y+2),$$
$$\tilde{P}_4=-(0.002y^2)(y^3+4y^2-3y+2)x$$
$$-(0.002y^2)(y^3+4y^2-3.003y+2.002),$$
$$P_4=\tilde{P}_4/\max\{(0.002)^2,0.002\times1.001\}$$
$$=-(0.999y^2)(y^3+4y^2-3y+2)x$$
$$-(0.999y^2)(y^3+4y^2-3.003y+2.002).$$

この例の P_1 と P_2 は例 4 と同じであるが，$\mathrm{mmc}(P_1)=O(1)$, $\mathrm{mmc}(P_2)=O(1)$ となり，$\gcd(P_1,P_2;10^{-3})=1$ であることが分かる．

文献 17) には，上記の規格化を採用するとき，F と G が精度 ε の近似的 GCD として D をもつならば，多項式剰余列において $\deg(P_k)=\deg(D)$ であるとき，$\mathrm{mmc}(P_{k+1})=O(\varepsilon)$ となることが示されている．

以上のように，近似的 GCD の場合には，正確な GCD 計算では全く現れなかった問題を解決する必要があるが，逆にみればそこに新しさと面白さがあるといえよう．

4.2 ある種の悪条件連立代数方程式への応用

筆者らは，多変数多項式の近似的 GCD の応用として，次のような悪条件連立代数方程式を考えた（本稿では簡単のため，2 変数の場合だけを扱う．3 変数の場合には文献 17) を参照されたい）．

$$\begin{cases}F=D\tilde{F}+\Delta F=0,\\ G=D\tilde{G}+\Delta G=0,\end{cases}$$

ただし，$F,G,D,\tilde{F},\tilde{G},\Delta F,\Delta G$ は **4.1** の初めに与えたものと同じである．

この連立方程式が悪条件であることは以下のように分かる．F と G の微小項 ΔF と ΔG を 0 とおくと，上式は二つの系 $\{D=0\}$ と $\{\tilde{F}=0,\tilde{G}=0\}$ に分かれ，第 1 の系から一般に無限個の解が出る．ところが，元の方程式は一般に有限個の解しかもたない．すなわち，$D=0$ の近傍では $F\simeq0$, $G\simeq0$ となるにもかかわらず，$D=0$ の点は真の解から遠い場合が大部分であり，ニュートン法などの数値解法では真の解への収束がきわめて悪くなる．

与えられた連立方程式 $\{F=0,\ G=0\}$ に対し，近似的 GCD 算法で D を計算したとしよう．D が求まれば F と G を D で割る（精度 ε の近似除算）ことにより $\tilde{F},\tilde{G},\Delta F,\Delta G$ を決定できるから，上記の分解が可能になる．そこで，元の方程式のかわりに

$$\begin{cases}F=0,\\ H=\tilde{G}F-\tilde{F}G=\Delta F\tilde{G}-\Delta G\tilde{F}=0,\end{cases}$$

を考える．$H\propto\tilde{G}F-\tilde{F}G$ ゆえ，$F=0$ のとき $H\propto\tilde{F}G$ となり，この連立方程式は

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

と分解できる．すなわち，新しい連立方程式は元の方程式の解とともに $\{F=0,\tilde{F}=0\}$ のみを満足する余分な解までも含む．ところで，H は D には無関係なの

で，ほとんどの場合，近似因子として D あるいはその約数をもたないと期待できる．すなわち，$\{F=0, H=0\}$ においては前記の悪条件性がほとんどの場合に解消されているといえる．

上記の方法の利点は，単に悪条件性の解消にとどまらず，解の近似値を簡単に計算できる点にもある．連立方程式は数値的にはニュートン法（あるいは他の算法）で解くことになるが，その場合問題になるのが初期値の選び方である．しかし，われわれの問題では，以下のようなうまい方法が存在する．$\{F=0, H=0\}$ は悪条件性をもたないだろうから，解の近似値は $F=D\tilde{F}$ と近似することにより得られるであろう．この近似により方程式は

$\{D=0, H=0\}$，$\{\tilde{F}=0, H=0\}$

の二つの連立方程式に分解される．$H=\tilde{G}F-\tilde{F}G$ であるので，後者は

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

と分解される．このうち，すでに示したように，$\{\tilde{F}=0, F=0\}$ は元の方程式 $\{F=0, G=0\}$ の根とは異なる根を与えるので除外する．以上より，ニュートン法の初期値は

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

の二つの連立方程式を粗く解いて決めればよい．第1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2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やすい根の近似値を与える．次に例を示す．

例6【近似的 GCD による悪条件連立方程式の解法】

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

多項式剰余列は

$$P_1=F$$

$$P_2=G$$

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

$$P_4=0.0001xy^4-0.00001xy^3-0.00002249xy^2+0.0000025xy-0.00000063x+0.00011y^5-0.0001375y^3+0.000025y^2+0.0000275y-0.00000063$$

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．

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

$$H=\tilde{G}F-\tilde{F}G$$
$$=0.0001(-1.1x^2y+xy^2-0.1xy+0.025x+0.025).$$

すなわち，元の方程式は

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

$$G=(x^2+y^2-1)(x-y)-0.00001(x+1),$$

と分解される．$\{F=0, G=0\}$ は悪条件であるが連立代数方程式 $\{F=0, H=0\}$ は良条件なので，数値計算が容易になることが分かる．初期値は次の方程式を粗く解いて求める．

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

表-2

No.	真の解	数値解法 結果と反復回数		融合解法 結果と反復回数	
①	$x=$ 0.9990732	0.9990724		0.9990732	
	$y=$ 0.0432840	0.0433031	19	0.0432840	2
②	$x=-0.0262280$	-0.0262224		-0.0262280	
	$y=$ 0.9996512	0.9996513	17	0.9996512	2
③	$x=-1.0000000$	-1.0000000		-1.0000000	
	$y=$ 0.0	$-3.3942\mathrm{E}-5$	17	$3.428\mathrm{E}-22$	1
④	$x=-0.0227383$	-0.0227343		-0.0227384	
	$y=-0.9997464$	-0.9997465	18	-0.9997464	2
⑤	$x=$ 0.6645091	-0.0262224		0.6645091	
	$y=$ 0.7471453	0.9996513	16	0.7471453	3
⑥	$x=-0.7141434$	-1.0000000		-0.7141434	
	$y=-0.6998564$	$-3.3942\mathrm{E}-5$	29	-0.6998564	3
⑦	$x=$ 0.5000350	0.5000350		0.5000350	
	$y=$ 0.5000650	0.5000650	3	0.5000650	2
⑧	$x=-0.5000550$	-0.5000550		-0.5000550	
	$y=-0.5000450$	-0.5000450	2	-0.5000450	2

表-2 に本稿の方法（融合解法）による計算結果とニュートン法による数値解法の結果を比較する．数値解法では初期値を定める良い方法がないので，われわれの方法で得た初期値を用いた．各初期値に対して，収束した根の値と収束するまでの平均の反復回数を示した．表-2 から明らかなように，$D=0$ の近傍に位置する根（①～⑥）に対して，数値解法では収束するまでの反復回数が多く，特に根⑤，⑥を求めることができないなど，数値解法の非能率性と不完全性がみられる．さらに，われわれは初期値として，通常やるように，$[-1, 1]$ の範囲で乱数を100組発生させたものを

用い，ニュートン法の収束を調べた．この場合には，方程式が悪条件のとき，計算は一部の特定の根に向かって収束し，いくつかの根は非常に得られにくいことが判明した．これらのことから，本稿に述べた融合解法の有用性が理解できるであろう．

5. おわりに

本稿で述べた方法は，本特集に解説されている区間解析のように，得られた解の誤差の上下限を明示する形で精度を保証するものではなく，従来とは異なった考え方に立脚して算法を構成することにより解の精度を上げるもので，その意味では本特集でも取りあげられている高速微分法に近い．しかしながら，高速微分法が微分だけを対象としているのに対し，本稿の方法ははるかに広い発展の可能性をもっている．それは近似的な代数関係式を算法として扱うものであり，それゆえわれわれは「近似的代数算法」と名付けたのであるが，本稿に説明されたのはほんの始まりの一部にすぎない．実際，それは単に代数的算法の浮動小数係数数式への拡張にとどまらず，より広い意味をもつことが本稿の説明で分かっていただけるであろう．今後，どのように発展していくか，現時点では予測しにくいが，当面の発展の方向は文献 18) に示されている．

近似的代数算法は日本で生まれたばかりの新しい算法である．本稿により，この算法の面白さが読者に伝わるならば，筆者らの望外の喜びである．

参考文献

1) Ng, E. W.: Symbolic-Numeric Interface: A Review, EUROSAM 1979, pp. 330-345 (1979).
2) Sasaki, T.: An Arbitrary Precision Real Arithmetic Package in REDUCE, EUROSAM 1979. pp. 358-368 (1979).
3) Gates, B. L.: GENTRAN: An Automatic Code Generation Facility for REDUCE ACM SIGSAM Bulletin, Vol. 19, No. 3 (1985).
4) Lohner, R.: Precise Evaluation of Polynomials in Several Variables, Computing Suppl., Vol. 6, pp. 139-148 (1988).
5) Kulisch, U. W. (Ed.): PASCAL-SC: A PASCAL Extention for Scientific Computation, B. G. Teubner, Stuttgart (1987).
6) 三井斌友：数値処理と数式処理の界面，情報処理，Vol. 27, No. 4, pp. 422-430 (1986).
7) Kahaner, D., Stewart, S. and Barnett, D.: "PLOD": Iterative Solver for Ordinary Differential Equation, Applied Math. and Computing, Vol. 31, p. 302 (1989).
8) Rall, L. B.: Automatic Differentiation: Techniques and Applications, Lect. Notes Comp. Sci. No. 120, Springer, New York (1981).
9) 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の大規模非線形方程式系への応用，情報処理学会論文誌，Vol. 26, No. 11, pp. 1411-1420 (1985).
10) Char, B. W., Fee, G. J., Geddes, K. O., Gonnet, G. H. and Monagan, M. B.: A Tutorial Introduction to MAPLE, J. Symbolic Computation, Vol. 2, pp. 179-200 (1986).
11) Wolfram, S.: Mathematica™, Addison-Wesley Pub. Co. (1988).
12) Suzuki, M. and Sasaki, T.: A Hybrid Algebraic-Numeric System ANS and Its Preliminary Implementation, Lect. Notes Comp. Sci., Vol. 378, pp. 163-171 (1989).
13) 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，Vol. 30, No. 4, pp. 419-426 (1989).
14) Sasaki, T. and Noda, M. T.: Approximate Square-free Decomposition and Root-finding of Ill-conditioned Algebraic Equations, Jour. Inf. Proc., Vol. 12, No. 2, pp. 159-168 (1989).
15) Bareiss, E. H.: The Numerical Solution of Polynomial Equation and the Resultant Procedure, in Mathematical Method for Digital Computers (Vol. 2), John Wiley (1967).
16) Sasaki, T. and Sasaki, M.: Analysis of Accuracy Decreasing in Polynomial Remainder Sequence with Floating-Point Number Coefficients, Jour. Inf. Proc., Vol. 12, No. 4, pp. 394-403 (1990).
17) Ochi, M., Noda, M. T. and Sasaki, T.: Approximate GCD of Multivariate Polynomials and Application to Ill-conditioned System of Algebraic Equations, submitted.
18) 佐々木建昭：近似的代数計算，数値解析の基礎理論とその周辺，数理解析研究所講究録 676, pp. 307-319 (1989).

(平成2年4月13日受付)