数式処理 *J.JSSAC (2005)*
Vol. 11, No. 3,4, pp. 189 - 200

招待論文

悪条件問題とハイブリッド計算

野田 松太郎*

愛媛大学 工学部

概　要

An approximate algebraic computation (AAC) becomes one of the most important research area of the algebraic computation. The basis of the AAC is an approximate greatest common divisor (AppGCD) proposed by T.Sasaki and M.T.Noda. It is used for many applications such as hybrid integration, hybrid rational function approximation (HRFA), and others. The AppGCD and its applications work well for obtaining accurate results of ill-conditioned problems. Algorithms and implementation methods of the AppGCD are briefly surveyed and its applications are described. Finally, features of the HRFA and relations of the HRFA and ill-conditioned problems are discussed in details.

1 はじめに

多項式が悪条件（ill-conditioned）であるということは，係数の微小な変化が，その零点に大きな変動をもたらすと定義される．悪条件代数方程式は，多項式がこのような状況を生じる場合で，以下のいずれかは悪条件性の大きな要因になる:

1. いくつかの根の大きさの比が１に近い
2. （多）重根の存在

第 1 の条件は，代数方程式が近接根を有する場合に対応している．今，代数方程式を

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = 0\ , \quad a_n \neq 0$$

とし，m_j を $P_n(x) = 0$ の根 x_j の多重度であるとする．ここで，係数 a_k に僅かな摂動が加わり，$a_k + \Delta a_k$ となったとする．そのときの根 x_j の変動は，$\Delta a_k \ll a_k$ として，

$$x_j = \{-\frac{m_j! x_j^{n-k} \Delta a_k}{(d/dx)^{m_j} P_n(x_j)}\}^{1/m_j}$$

と表されることがわかっている [1]．代数方程式が重根や近接根を含まない良条件のものである場合には，多くの数値計算のアルゴリズムが知られれており，高速かつ高精度で数値解を得るこ

*noda@cs.ehime-u.ac.jp

とが出来る．しかし，悪条件問題に対しては，ほとんどの数値計算法は効果的ではなく，十分な精度で解を得ることが出来なくなる．

一方，多項式 $P_n(x)$ の係数と根が整数か有理数であるなら，重根は数式処理（代数計算）によって容易に分離できる．すなわち，s-多重根を考えると，$P_n(x)$ は，$d^{(s)}P_n(x)/dx^s$ で整除される．したがって，$P_n(x)$ と，その s-階微分多項式との最大公約因子 (GCD : Greatest Common Divisor) が重要な役割を果たすことになる．2 つの多項式 P_1 と P_2 の最大公約多項式 $\mathrm{GCD}(P_1, P_2)$ は，代数的なユークリッド算法で容易に計算できる．この方法を，係数が整数や有理数でなく，浮動小数などの場合（*inexact* という）の多項式計算に拡張する．この場合には，本来の GCD 計算は，「精度 ϵ の近似 GCD(AppGCD)」すなわち $\mathrm{AppGCD}(P_1, P_2; \epsilon)$ に置きなおされる [11]．上で述べた *inexact* 係数の多項式に対するユークリッド算法の拡張を考えるとき，2 つの方法が存在する．第 1 は，すでに AppGCD（精度 ϵ の近似 GCD）として知られているものであり，第 2 は，区間計算を用いるものである [9]．これら 2 つの方法について，簡単に **2** で述べる．AppGCD は，色々の科学計算に応用されている．それらの中，ハイブリッド積分と呼ばれる手法を **3** で概説する．この方法では，数値計算と数式処理を融合した方法で，悪条件被積分関数に対する高精度の積分値を与える．**4** は，ハイブリッド有理関数近似（Hybrid Rational Function Approximation：略して HRFA）についての考察にあてる．関数やデータを古典的な有理関数補間によって近似しようとする．高い近似精度で関数近似を得ようと，補間点を増加させると，一般に有理関数補間に特異点が生じてしまう．この点の検討により，この特異点は，有理関数補間の分子，分母の多項式が近似的な共通因子を含むことが原因であることがわかる．このような，近似的共通因子は，分子と分母の多項式の AppGCD によって取り去られ，結果としての有理関数近似は極めて良好な近似になっていることが示される．

2 近似 GCD 算法

近似 GCD（AppGCD）を求めるための 2 つの方法について述べる．第 1 のものは，精度 ϵ の近似 GCD として知られており，[11] 以後に発表された多くの近似 GCD 算法の提案の基礎となっているものである．これらの算法といくつかの応用については，[6] に詳しく述べられている．第 2 は区間計算に基づくもので，異なった方向へのハイブリッド計算または数値数式融合計算のための補完的研究になっている．

2.1 精度 ϵ の近似 GCD 算法

2 つの多項式 P_1 と P_2 の，精度 ϵ の近似 GCD (AppGCD) は，ユークリッド算法の自然な拡張で， $\mathrm{AppGCD}(P_1, P_2; \epsilon)$ と表される．ここで，2 つの多項式 P_1 と P_2 の係数は浮動小数等の *inexact* なものであるとする．多項式剰余列 (PRS) を得るためのユークリッド算法は

$$P_{i-1} = P_i Q_i + P_{i+1}, \qquad i = 2, \ldots$$

である．ここで，Q_i は商多項式を表す．係数が *inexact* なので，除算時には常に僅かな誤差が係数に混入する．そこで，PRS を得る場合に，ある許容誤差 ϵ 以下の値を切り捨て，0 にみなす演

算（cutoff 演算）を導入する．このようにして求まる PRS は次のように書ける．

$$P_1, P_2, P_3, \ldots, P_k \neq 0, P_{k+1} = 0(\text{ cutoff } \epsilon.$$

代数計算でのユークリッド算法による PRS では，2 つの多項式が共通因子を持つ場合には，厳密に $P_{k+1} = 0$ となり，P_k の原始的部分 (pp : primitive part) が共通因子になる．しかし，係数が *inexact* の場合には，cutoff 演算が必要であることが大きな相違である．このことから，精度 ϵ の近似 GCD を次のように定義する．

$$\text{AppGCD}(P_1, P_2; \epsilon) = \text{pp}(P_k) .$$

この AppGCD は，近接根を有する悪条件代数方程式やその系を解く目的で開発され，結果の詳細は [11, 10] に示されている．

2.2 区間演算に基づく近似 GCD 算法

係数が *inexact* である場合には，誤差を含んだ数を係数とすることも十分に考えられる．このような場合に，区間数を用いることが一般に考えられる．区間数の演算には，区間演算を用いるが，広く普及している矩形型区間数によるものと，円盤型区間数によるもととがある．多項式演算の場合に，前者は，誤差 ϵ を含んだ係数を，その中点 M として，$[M-\epsilon, M+\epsilon]$ なる区間とするものである．矩形型区間数は演算は容易だが，数学的性質の保持等で問題もある．そのため，ここでは円盤型区間数を用いる．円盤型区間数による区間演算（circular interval arithmetic）は，中点 M と，その近傍の誤差半径（$r(M) = \epsilon$）とで表す．したがって，円盤型区間数は $< M, r(M) >$ のように表現される．計算方法は，精度 ϵ の近似 GCD の場合と同じように，ユークリッド算法で PRS を求めていく．精度 ϵ の近似 GCD の場合には，PRS の終了を示す場合に，cutoff 演算によって，微小な値を 0 とみなしたが，区間演算の場合は異なる．実際に，区間演算で除算を行う場合に，0 を含む区間での除算は不可能になる．すなわち，PRS 計算途中で，多項式の係数が

$$| M |<| r(M) |$$

なる条件が成立すると，円盤型区間数は 0 を含むことになり，計算が進行しなくなる．したがって，PRS 計算の終了条件として，上の条件を付する必要がある [9]．後に，この条件はさらに数学的に厳密に検討され，区間の 0 書き換えと計算桁数を増加させての反復計算によるアルゴリズムの安定化理論として確立されている [12].

2.3 近似 GCD の計算例

次の代数方程式を考える．

$$P(x) = x^4 - 10.4x^3 - 70.96x^2 + 29.6x - 3 = 0$$

この方程式は，$x = -5, 15$ に根を持つほか，$x = 0.2$ に重根を持ち，典型的な悪条件問題の 1 つである．この *inexact* 係数の多項式の GCD を，上で述べた 2 つの近似 GCD の方法を用いて求める．

まず，精度 ϵ の近似 GCD による計算を行う．ユークリッド算法による PRS は次のようになる．

$$
\begin{aligned}
P_1 &= x^4 - 10.4x^3 - 70.96x^2 + 29.6x - 3,\\
P_2 &= 4x^3 - 31.2x^2 - 141.92x + 29.6 \quad (= dP_1/dx),\\
P_3 &= -85.78x^2 - 107.76614385x + 24.98461538,\\
P_4 &= -0.46563934x + 0.09312787,\\
P_5 &= 2.77555756 \times 10^{-17}.
\end{aligned}
$$

P_5 の係数の最大値は，P_4 の係数の最大値に比較して 10^{-16} 倍小さくなる．そこで，cutoff 演算により，$P_5 = 0$ (cutoff 10^{-16}) とみなして，0 とする．P_4 の係数を正規化し，原始的部分を取り出すことによって，精度 $\epsilon = 10^{-16}$ の近似 GCD を次のように求めることが出来る．

$$\mathrm{appGCD}(P_1, P_2; 10^{-16}) = x - 0.2$$

他方，円盤型区間演算による PRS 計算の結果は，次のようになる．

$$
\begin{aligned}
P_1 &= <1.0, 2.2 \times 10^{-16}> x^4 + <-10.4, 1.8 \times 10^{-15}> x^3\\
&\quad + <-70.96, 1.4 \times 10^{-14}> x^2 + <29.6, 3.6 \times 10^{-15}> x + <-3.0, 4.4 \times 10^{-16}>,\\
P_2 &= <4.0, 1.8 \times 10^{-15}> x^3 + <-31.2, 8.9 \times 10^{-15}> x^2\\
&\quad + <-141.92, 5.7 \times 10^{-14}> x + <29.6, 7.1 \times 10^{-15}> \quad (= dP_1/dx),\\
P_3 &= <-8.9 \times 10^2, 3.8 \times 10^{-12}> x^2 + <-1.1 \times 10^3, 9.3 \times 10^{-12}> x\\
&\quad + <2.6 \times 10^2, 1.8 \times 10^{-12}>,\\
P_4 &= <-4.7 \times 10^6, 1.1 \times 10^{-7}> x + <9.5 \times 10^5, 2.2 \times 10^{-8}>,\\
P_5 &= <1.3 \times 10^{-6}, 7.8 \times 10^{-4}> \ni 0.
\end{aligned}
$$

P_5 が，その区間内に 0 を含むので，計算続行は不可能となり，PRS 計算は，ここで打ち切らざるを得なくなる．アルゴリズムの安定化理論 [12] を借用すると，0 書き換え操作により，P_5 は 0 になる．したがって，以下を得る．

$$\mathrm{GCD}(P_1, P_2) = \mathrm{GCD}(P(x), dP(x)/dx) = x - <0.2, 4.7 \times 10^{-10}>.$$

いずれの場合も，得られた AppGCD は，$x = 0.2$ 近傍に $P(x)$ の近似的重根が存在することを示している．以上のような近似 GCD 算法の提案以後，多くの近似 GCD 算法が提案され，色々に活用されている．結果として，現在の代数計算・数式処理研究の中心の一つとなっている．

3 近似 GCD とハイブリッド積分

与えられた関数の不定積分を求める計算は，科学計算の中でも最も重要なものの一つである．関数が整数や有理数を係数とする有理関数で与えられると，多くの場合は数式処理による記号積分のアルゴリズムを用いて不定積分を得ることが出来る．しかし，次のような場合には問題が残っている．

- 積分すべき関数が与えられても，閉形式の解が得られない場合
- 積分すべき関数が表や離散点列で与えられた場合
- 積分すべき関数が有理関数であっても，係数が浮動小数等 *inexact* な場合

上のような場合にも，数値積分公式は，定積分値を与えるのみである．ここで，近似 GCD 算法のように，代数的算法に基礎を置き，数値的手法を有効に組み合わせたハイブリッド積分の考え方を示す．以下に詳しく述べるように，ハイブリッド積分のアルゴリズムによると，積分すべき関数や離散点列に対して，ある種の記号的な不定積分を求めることが出来る．不定積分さえ求まれば，その上下限に数値を代入するのみで，高精度に定積分値も求まる．ハイブリッド積分のアルゴリズムの概略は次の通りである．詳細は [7, 8] にも述べられている．

アルゴリズム 1 (ハイブリッド積分)

入力: 浮動小数係数を持つ 1 変数有理関数 $f(x) = P(x)/Q(x)$．ここで，多項式 $P(x)$ と $Q(x)$ は共通因子を持たず，その次数の関係は $\deg(P) < \deg(Q)$

出力: $f(x)$ の近似的不定積分

計算法:

1. $f(x)$ を有理部分と超越部分に分解

$$\int f(x)dx = \frac{s(x)}{t(x)} + \int \frac{p(x)}{q(x)}dx.$$

ここで，AppGCD を用いる．もし，$\mathrm{AppGCD}(Q(x), dQ(x)/dx; \epsilon) = 1$ なら，$\frac{s(x)}{t(x)} = 0$ である．

2. 超越部分 $p(x)/q(x)$ の積分

(a) $q(x)$ の全ての零点を数値的な Durandt-Kerner 法で求める．$q(x)$ の係数は全て実なので，零点は実か共役複素対になる．これらを $m + n = \deg(q)$ と表す

実の零点 : $a_1, a_2, \ldots, a_m$

共役複素零点 : $b_1 \pm ic_1, \ldots, b_n \pm ic_n$

(b) 部分分数への分解

$$fracpq = \sum_{k=1}^{m} \frac{e_k}{x - a_k} + \sum_{k=1}^{n} \frac{2(f_k x - b_k f_k - c_k g_k)}{x^2 - 2b_k x + b_k^2 + c_k^2},$$

ここで，e_k, f_k及び g_k は，次のように留数定理で求める

- $r(x) = p/dq(x)/dx$ とする
- 実の零点に対し，　$e_k = r(a_k)$
- 共役複素零点に対し，

$$f_k = \Re\{r(b_k + ic_k)\} \ \ and \ \ g_k = \Im\{r(b_k + ic_k)\}$$

ここで，$\Re$ と $\Im$ は，各々実部分と複素部分を表す．

(c) 対数積分に関する，良く知られた次の 2 つの公式に代入する

$$\int \frac{p}{q}dx = \sum_{k=1}^{m} c_k \log|x-a_k| + \sum_{k=1}^{n} f_k \log|x^2-2b_kx+b_k^2+c_k^2| -2g_k\tan^{-1}(\frac{x}{c_k}-\frac{b_k}{c_k}).$$

ハイブリッド積分によって得られた定積分値を，数値計算での積分公式として有名な，32 点ガウス積分 (Gauss32)，二重指数積分 (DE)，ロンバーグ積分 (Romb)，適応型ニュートン・コーツ公式 (NC) による結果と比較する．比較のために，次の悪条件の被積分関数を用いる．

1. 積分範囲（積分の上限，加限）のすぐ外側に特異点が存在

$$I_1 = \int_0^1 \frac{dx}{1000x(x-1)-0.001}$$

2. 積分範囲の中央に鋭いピークが存在

$$I_2 = \int_0^1 \frac{dx}{1000(x-0.5)^2+0.001}$$

3. I_1 と I_2 の双方の性質を持つ関数

$$I_3 = \int_0^1 \frac{dx}{x^5-x^4-0.75x^3+x^2-0.25-10^{-6}}$$

結果は表 1 に示した．ハイブリッド積分の結果が，数値計算公式と比較して優れていることはいうまでもない．特に，I_3 のような極めて悪条件な関数の積分を行おうとすると，ハイブリッド積分以外には頼る方法が存在しない．なお，[7, 8] 等の発表時には，MAPLE 等の数式処理システムを用いても，上のような悪条件被積分関数の定積分値を正しく求めることは出来なかった．特に，I_3 に対しては計算不能であった．しかし，現在流通している MAPLE を用いると，正しく定積分値を求めることが出来，結果は表 1 と一致することを付記する．

	ハイブリッド積分	数値積分			
		G32	DE	Romb	NC
I_1	-0.02763097	-0.01622972	-0.02763097	-0.68482819	5.40966656
I_2	3.13759266	0.19994034	24.6736125	2.98225113	3.13759258
I_3	-5195.24497	-580.39410	-24965.007	-5759.10716	230.75544

表 1: ハイブリッド積分と数値積分の定積分値の比較

4 ハイブリッド有理関数近似 (HRFA)

4.1 有理関数補間と病的振る舞い

近似 GCD 算法の意味を活用する応用問題の中で最も適しているものの一つがハイブリッド計算環境下での有理関数近似である．詳しい議論は，[8, 3, 2, 5] 等にあるが，ここではその概略と

以後の議論の参考となるべき計算事例を示す．まず，２つの多項式（分子多項式と分母多項式）の比で定められる古典的な有理関数補間から出発する．

$$\begin{aligned} r_{m,n}(x) &= \frac{p_m(x)}{q_n(x)} = \frac{\sum_{i=0}^{m} a_i x^I}{1+\sum_{i=1}^{n} b_i x^i} \\ &= \frac{a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m}{1 + b_0 x + b_1 x^2 + \cdots + b_n x^n}. \end{aligned}$$

このように定めた有理関数 (1) は，区間 $[\alpha,\beta]$ で，与えられた関数や離散点列を補間し，(m,n) 有理補間と呼ばれる．関数や離散点列から有理補間を求めるには，以下の手順にしたがう．まず，区間 $\alpha = x_0 < x_1 < \cdots < x_{m+n} = \beta$ で，$m+n+1$ 個の関数値 $f(x_k) = f_k, (k = 0, \ldots, m+n)$ を評価する．いずれの場合も，$m+n+1$ 個の (x_k, f_k) の離散点列の組ができる．そこで，次の，$m+n+1$ 元の連立一次方程式

$$\sum_{i=0}^{m} a_i x_k^i - f_k \sum_{I=i}^{n} b_i k_k^i = f_k, \quad k = 0, 1, \ldots, m+n$$

が構成される．これを行列形で書くと，次のようになる．

$$\begin{pmatrix} 1 & x_0 & \cdots & -f_0 x_0 & \cdots & -f_0 x_0^n \\ 1 & x_1 & \cdots & -f_1 x_1 & \cdots & -f_1 x_1^n \\ 1 & & \vdots & & \vdots & \\ 1 & x_m & \cdots & -f_m x_m & \cdots & -f_m x_m^n \\ 1 & x_{m+1} & \cdots & -f_{m+1} x_{m+1} & \cdots & -f_{m+1} x_{m+1}^n \\ 1 & & \vdots & & \vdots & \\ 1 & x_{m+n} & \cdots & -f_{m+n} x_{m+n} & \cdots & -f_{m+n} x_{m+n}^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_m \\ f_{m+1} \\ \vdots \\ f_{m+n} \end{pmatrix}, \qquad (1)$$

有理補間の浮動小数係数 a_i, b_i は，上の連立一次方程式をガウス消去によって求められる．この有理補間の振る舞いとして，Noda 他は以下を見出している [8].

1. 補間すべき関数が連続関数であっても，有理補間の分母の多項式は，区間 $[\alpha,\beta]$ 内に零点も待つ．この零点は補間すべき関数 $f(x)$ にはないものなので，*undesired pole*（極）と名づける．
2. 上の極以外では，有理補間は，高い精度で $f(x)$ を近似する．

さらに，分子の多項式の零点が *undesired pole* の極めて近くに存在することが，Kai 他によって，数値実験を通して示されている [3]．このような，極と零点の組み合わせを *undesired zero and pole* と呼ぶ．このことは，当然ながら，有理補間の分子多項式と分母多項式が近似的共通因子を持つことを示唆している．有理補間の分子・分母多項式の近似的共通因子を求め，それで有理関数補間を除算すると，*undesired zero and pole* も消去され，補間すべき関数の高精度の近似有理関数が生成される．こうして得られた有理関数ををハイブリッド有理関数補間（Hybrid Rational Function Approximation）と呼び，HRFA と略する．HRFA はデータ平滑化その他の実際的応用で多くの応用分野を持つ．有理補間の連立一次方程式が多くの場合に悪条件になること，系自身

は悪条件になっても，有理補間そのものは，最良近似の意味で優れた近似になっていることが，Litvinov によっても示されている [4].

このように，連続な関数を近似しようと有理補間すると，補間区間内に特異点を生じるという病的振る舞いが起こることと，その振る舞いが有理補間の分子・分母多項式の近似的共通因子を考えることによって取り去ることが出来ることがわかる．近似的共通因子を求めるには AppGCD を用いる．このようにして求めた HRFA は高精度の近似を与える．このアルゴリズムが，**アルゴリズム 2** である．

アルゴリズム 2 (有理関数近似 (HRFA))

入力 ：有理補間

$$r_{m,n}(x) = \frac{p_m(x)}{q_n(x)}$$

出力 ：特異点が除去された有理関数近似

$$\tilde{r}(x) = \frac{\tilde{p}(x)}{\tilde{q}(x)}$$

計算法 ：

1. Approximate-GCD$(p_m(x), q_n(x)) = g(x)$

2. $\tilde{r}(x) = \dfrac{p_m(x)/g(x)}{q_n(x)/g(x)}$

次に，具体例を通じて，有理補間の病的振る舞いと，いかに**アルゴリズム 2** が良い結果を与えるかを見る．今，関数 $f(x) = \log(x+2)$ の補間区間 $[-1, 1]$ での有理補間 $r_{4,4}(x)$ を考える．単精度計算を行うと，有理補間は次のようになる．

$$\begin{aligned} r_{4,4}(x) &\simeq \frac{0.6931 + 214.8599x + 318.6295x^2 + 113.5897x^3 + 9.1269x^4}{1 + 309.2559x + 236.7844x^2 + 48.7819x^3 + 2.1131x^4} \\ &\simeq 4.3195\,\frac{(x+8.77124)(x+2.6710)(x+1)(x+0.0032415)}{(x+16.999)(x+3.8532)(x+2.2286)(x+0.0032416)}\,. \end{aligned}$$

分子・分母多項式各々の 4 項目の因子が近似共通因子であることは明らかである．しかし，これらは厳密には一致しないので，分母多項式の零点である $x \simeq -0.0032416$ で補間区間内に極 (*pole*) が出現し，分子多項式の零点である $x \simeq -0.0032415$ で零点 (*zero*) が出現する．すなわち，この $r_{4,4}(x)$ 有理補間は *undesired zero and pole* を持つ．しかし，*undesired zero and pole* の存在する微小区間以外では，有理補間は，関数 $f(x)$ の高精度の近似を与える．*Undesired zero and pole* は，$p_m(x)$ と $q_n(x)$ の近似 GCD$g(x)$ を求め，$r_{4,4}(x)$ を近似 GCD で除算することによって，取り除くことが出来る．今の場合，

$$g(x) \simeq x + 0.0032423543$$

となる．$g(x)$ による $r_{4,4}(x)$ の分子・分母多項式の除算で，次のようにハイブリッド有理関数近似 $R_{HRFA}(x)$ を得ることが出来る．

$$\begin{aligned} R_{HRFA}(x) &\simeq \frac{9.1269605x^3 + 113.56010x^2 + 318.26127x + 213.82796}{2.1131867x^3 + 48.775051x^2 + 236.62620x + 308.48869} \\ &= 4.3190507\,\frac{(x + 8.7712482)(x + 2.6710225)(x + 0.99999856)}{(x + 16.999383)(x + 3.8532500)(x + 2.2286457)}\,. \end{aligned}$$

こうして得られた有理関数 $R_{HRFA}(x)$ は *undesired zero and pole* を含まず，$f(x)$ の良い近似を与える．

4.2 有理補間と悪条件性

有理補間の病的振る舞いに関しては，以下の事実が数値実験から示されている．

1. *Undesired zero and pole* は常に非常に近い場所に対として現われる
2. *undesired zero and pole* の位置は，有理補間 $r_{m,n}(x)$ の次数や計算桁数によってランダムに変動する
3. 非常に高い計算桁数で有理補間を求めると，*undesired zero and pole* は現われない

第 2 の性質は，有理補間を求める連立一次方程式の悪条件性に起因していると思われる．さらに，高精度の有理補間を求めようとして，補間点を多くする（m, n：大）と，連立一次方程式は悪条件性を増すもわかる．また，第 3 の性質は，厳密計算に近い程度の計算桁数での浮動小数計算は，非常に長時間の計算が必要ではあるが，正しい結果を生じることを示している．有理補間で出現する *Undesired zero and pole* と関連する分子・分母多項式の近似的共通因子の意味についての詳細な議論は [5] で行われる．

以下では，補間点を必要以上に増加させたときに，有理補間はどのような結果を生じるかについて簡単に計算例を紹介する．今，

$$f = \frac{1}{x-3}$$

を $x = -2, -1, 0, 1, 2$ の 5 点で補間することを考える．補間すべき関数が有理関数であるので，$m = n = 2$ の有理関数は一種の過剰決定系になる．(1) に対応する連立一次方程式は

$$\begin{pmatrix} 1 & -2 & 4 & -2/5 & 4/5 \\ 1 & -1 & 1 & -1/4 & 1/4 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1/2 & 1/2 \\ 1 & 2 & 4 & 2 & 4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -1/5 \\ -1/4 \\ -1/3 \\ -1/2 \\ -1 \end{pmatrix}$$

となる．数式処理システムを用いて厳密な計算でガウス消去を行うと，次の結果を得る．

$$\begin{pmatrix} 1 & -2 & 4 & -2/5 & 4/5 \\ 0 & 1 & -3 & 3/20 & -11/20 \\ 0 & 0 & 1 & 1/20 & 3/20 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -1/5 \\ -1/20 \\ -1/60 \\ -1/3 \\ 0 \end{pmatrix}$$

このように，ランク落ちする行が出現し，b_2 を決定できなくなる．そこで，$b_2 = t$ と未定定数 t を代入してみる．この結果，有理補間 $r_{2,2}$ は次のようになる．

$$r_{2,2}(x) = \frac{-\frac{1}{3} + tx}{1 - \frac{1}{3}x - 3tx + tx^2} = \frac{1 - 3tx}{(x-3)(1-3tx)} \Rightarrow \frac{g(x)}{(x-3)g(x)} = \frac{1}{x-3}$$

すなわち，$g(x) = 1 - 3tx$ が分子・分母の共通因子として出現し，これを消去すると，補間すべき関数を再現することが出来る．補間点の数 $(m+n+1)$ をさらに増加すると，次の結果を得る．

$m + n + 1 = 7$ **の場合**

補間点 $x = -3/2, -1, -1/2, 0, 1/2, 1, 3/2$

未定定数 $b_2 = t_1, b_3 = t_2$

有理補間
$$r_{3,3}(x) = \frac{g(x)}{(x-3)g(x)},$$
$$g(x) = -1 + 3t_1x + 9t_2x + 3t_2x^2$$

$m + n + 1 = 9$ **の場合**

補間点 $x = -2, -3/2, -1, -1/2, 0, 1/2, 1, 3/2, 2$

未定定数 $b_2 = t_1, b_3 = t_2, b_4 = t_3$

有理補間
$$r_{3,3}(x) = \frac{g(x)}{(x-3)g(x)},$$
$$g(x) = -1 + 3t_1x + 9t_2x + 27t_3x + 3t_2x^2 + 9t_3x^2 + 3t_3x^3$$

$m + n + 1 = 11$ **の場合**

補間点 $x = -3/2, -1, -1/2, 0, 1/2, 1, 3/2$

未定定数 $b_2 = t_1, b_3 = t_2, b_4 = t_3, b_5 = t_4$

有理補間
$$r_{3,3}(x) = \frac{g(x)}{(x-3)g(x)},$$
$$g(x) = -1 + 3t_1x + 9t_2x + 27t_3x + 81t_4x + 3t_2x^2 + 9t_3x^2 + 27t_4x^2 + 3t_3x^3 + 9t_4x^3 + 3t_4x^4$$

いずれの場合も $r_{2,2}$ の場合と同様の結果を得る．厳密計算の場合は，結果の未定係数は本来 0 であるべき値であるとしてもよく，正しい結果を得る．しかし，浮動小数係数で計算すると，未定定数を代入した部分が数値計算の計算誤差により，0 ではない微小値の数値が代入される．$r_{2,2}$ の場合は，次のようになる．

$$r_{2,2}(x) = \frac{p_2(x)}{q_2(x)} = \frac{-0.3333333 - 0.1111111x - 1.0 \times 10^{-8}}{1 - 0.1111111x^2} \Rightarrow \frac{1}{x-3}$$

ここで，分子・分母多項式の近似 GCD は，

$$\mathrm{AppGCD}(p_2(x), q_2(x); 10^{-8}) = \mathrm{pp}(0.1111111x + 0.3333333) = x + 3$$

となる．結果として，分子多項式での $g(x)$ と分母多項式でのそれは，僅かに異なった数値となる．このため，*undesired zero and pole* が出現する．このような場合には，上で述べたハイブリッ

ド有理関数近似のアルゴリズム (HRFA) は良好な結果を与える．以上の簡単な例は，[5] での解析結果と一致する．

5 むすび

以上のように，本論では近似 GCD 算法に始まる近似代数計算 (AAC) あるいは，数値数式融合ハイブリッド計算の概要と応用について述べてきた．このような計算法の基本は，

悪条件問題を良条件にして解く

ことを基本としている．すなわち，数式処理のみによる計算法の応用の限界性を打破し，数値計算のにみよる計算法の計算誤差の危惧という不安を打破することができる．さらに，ハイブリッド有理関数近似にみられるように，悪条件問題に起因する解決すべき理論的問題を多く含んでいる．今後も多くの関連した研究成果が生まれることを期待する．

最後のなったが，ここで述べた研究の大半は，著者が愛媛大学在籍中に研究の進展に協力してくれた工学部の学生諸君の努力によっているものである．あまりに多くの学生諸君に多年にわたってご協力いただいたので個々のお名前を掲載しないが，深く感謝します．同様に，近似 GCD 算法の構築にあたっては数式処理分野への *New Commer* であった著者とご協力いただいた筑波大学佐々木建昭教授に感謝します．また，今回，著者が愛媛大学を退職する機会に素晴らしい研究会の開催と，会誌の発刊を企画・実施された日本数式処理学会の関係者の皆様に感謝します．

参 考 文 献

[1] E.H.Bareiss : The numerical solution of polynomial equations and the resultant procedure,in “Mathematical Method for Digital Computers Vol.2”, John Wiley, 1967

[2] H.Kai and M.T.Noda : Hybrid rational function approximation and data smmoothing, *J.Jap Appl.Math.*, Vol.**3**, pp.323–336, 1993 (in Japanese)

[3] H.Kai and M.T.Noda : Hybrid rational function approximation and its accuracy analysis, *Reliable Computing* **6**, pp.429–438, 2000

[4] G.Litvinov : Approximate construction of rational approximations and the effect of error autocorrection. Applications, in *Russian Journal of Mathematical Physics*, vol.1, No.3, 1994.

[5] 中島裕美, 甲斐博, 野田松太郎：ハイブリッド有理関数近似と悪条件問題, 数式処理 11 巻,3・４号合併号, *to appear*, 2005.

[6] 野田松太郎, 甲斐博 : 数式処理と数値計算 – いかに結合させるか? , 情報処理,39 巻,pp.105-110, 1998.

[7] M.T.Noda, E.Miyahiro : On the Symbolic/Numeric Hybrid Integration, Proc.ISSAC90, ACM, p.304, 1990.

[8] M.T.Noda,E.Miyahiro and H.Kai: Hybrid rational function approximation and its use in the hybrid integration, in “Advances in Computer Methods for Partial Differential Equations VII”,eds.R. Vichnevetsky, D.Knight and G.Richter, IMACS, pp.565–571,1992

[9] M.T.Noda and T.Sasaki, The interval arithmetic for the ill-conditioned polynomial equation, RIMS(Research Institute of Mathematical xxx, Kyoto University) Lecture Note 673, pp.47-61, 1988

[10] M.T.Noda and T.Sasaki, Approximate GCD and its application to ill-conditioned algebraic equations, JCAM, **38**, pp.335-351, 1991

[11] T.Sasaki and M.T.Noda : Approximate Square-free Decomposition and Root-finding of Ill-conditioned Algebraic Equations, *J. Inf. Proces.*, **12**, pp.159–168, 1989

[12] K.Shirayanagi and M.Sweedler, A Theory Stabilizing Algebraic Algorithms, Texh.rep. 95-28, Cornell Univ., pp.1-92, 1995