

近似的 GCD を用いた有理関数近似¹

野田松太郎[†], 宮広栄一^{††}, 甲斐博[†]

[†] 愛媛大学工学部情報工学科 ^{††} 愛媛大学工学部電子工学科

1 はじめに

関数近似の技法は最も代表的な数値計算法の一つであり、多くの教科書でも詳しく議論されている。以後の解析が困難なような複雑な関数、実験結果のようにデータ列が与えられた場合の近似関数形の導出などに対して色々な関数近似の方法がある。これらに共通するのは、有限個の点（以下サンプル点と呼ぶ）とその点での関数値またはデータ値の組を与えて近似関数を求めることである。このうち多項式近似は最も単純で性質の究明も十分に進んでおり応用も広い。しかし、近似される関数やデータが変化が激しかったり、特異点を含んでいたりすると、点の組を増加させても十分な近似が得られない。現在広く用いられているスプライン関数近似他の多くの高精度の近似法もあるが、得られる関数形が煩雑で定性的な振舞いを見たり、より進んだ解析をしたりするには十分ではない。そこで、本論では多項式近似に劣らず単純で数学的解析も進んでおり、かつ激しい変化や特異点を含むデータにも対応し得る有理関数近似 [4, 5] を取り上げる。有理関数近似を得られれば、近似関数の振舞いの解析はもちろん、数値数式融合アルゴリズムであるハイブリッド積分 [3] を用いて高精度な積分結果を求めることも可能となる。

有理関数近似は分子、分母の多項式の比で関数近似を行うものだが、高精度の有理関数近似を得ようとして、サンプル点とデータの組を増大させると分母の多項式の次数が大きくなり、不必要な零点を持つことがある。すなわち、近似有理関数は本来存在しない不必要な極を持つことになる。これは有理関数近似の限界であり広範囲な利用を不可能にしている。多くの有理関数近似の議論では分子、分母の多項式は互いに素で、かつ分母の多項式は近似区間では零点を持たないという大前提のもとで行われている。本論ではこの点についての考察を加え、有理関数近似の利用範囲を拡大することを考える。実験例を通じて、極を生じる分母の多項式の不必要な零点の位置は分子の多項式の零点の位置と近似的に同じであることを示す。すなわち分子、分母の多項式の次数を高次にすると、それらが近似的共通因子を持つことになる。この近似的共通因子を取り去れば低次の有理関数近似が得られる。このようにして求めた有理関数近似の精度は、最初の有理関数近似を高精度にすれば近似的共通因子の近似度だけ低下することが分かる。もともと少ない点で近似したものより有効な近似有理関数を求めることが可能になると予想できる。結局このような有理関数近似を得る上での要点は近似的共通因子をどのように求めるかである。本論では、これを分子、分母の多項式の近似 GCD(最大公約多項式)[7] を計算して求める。すなわち、ハイブリッド計算での有理関数近似を得ることになる。以下、2 で簡単に有理関数近似を概説し、3 ではハイブリッド計算での有理関数近似を得るアルゴリズムに至る過程を実例を通じてみる。アルゴリズムとその拡張については 4 で述べる。また最後に本論の研究を進める上で、今後に行うべきいくつかの課題について言及する。

¹ Rational function approximation by using approximate GCD, by Matu-Tarow Noda, Eiichi Miyahiro and Hiroshi Kai, Faculty of Engineering, Ehime University

2 有理関数近似と問題点

関数 $f(x)$ の範囲 $[a, b]$ での有理関数近似は次のようにして求める。いまサンプル点の列 $x_0 = a, x_1, \dots, x_N = b$ を与え、各点での関数値 $f(x_0), f(x_1), \dots, f(x_N)$ を計算する。結局、(点、関数値) の有限個の組を与えるので、関数値を実験データなどの離散データに置き換えても支障はない。これらの組を正確に通る有理関数

$$r_{m,n} = \frac{p_m}{q_n} \quad (2.1)$$

を決定する。ここで p_m, q_n は各々次数 m, n の多項式であり、 $p_m = a_0 + a_1x + \dots + a_mx^m, q_n = b_0 + b_1x + \dots + b_nx^n$ とする。(1.1) 有理関数を (m, n) 有理関数と呼ぶ (type of (m, n))。便利のために、特に $b_0 = 1$ とおく。これら多項式の係数を決定するには $n + m + 1$ 個の点と関数値または離散データの組を与え、連立 1 次方程式を解く必要がある。したがって、 $N = n + m$ である。また、 (m, n) 有理関数では関数近似の精度を考えると $m = n$ または $m = n \pm 1$ の場合が最適とされている [8]。

有理関数近似に関する議論は古くから行われている。例えば指数関数のように (m, n) 有理関数の収束定理が示されている場合もある。ただし、これらで扱われる有理関数 $r_{m,n}$ は既約である ($[a, b]$ 内では p_m, q_n は共通の零点を持たない) とされている。教科書 ([5] など) では、これについて

$$\frac{x+1}{x^2+3x+2} = \frac{(x+1)}{(x+1)(x+2)} = \frac{1}{x+2} \quad (2.2)$$

としており、 $r = 1/(x+2)$ を結果の有理関数としている。また、有理関数は考える区間 ($[a, b]$) 内で極を持たないことも仮定されている。すなわち、分母の多項式 q_n の n 個の零点はすべて区間外にあるか複素数であるとされている。しかし、現実に現れる関数近似の問題では多項式の係数は一般に浮動小数であり、通常の方法では上のような多項式の因数分解は不可能である。浮動小数点計算の結果求める $r_{m,n}$ が既約である場合はほとんど無いといつてよいくらいである。特に、 n, m を大きくすると分母の多項式 q_n の実の零点が区間内にある可能性も増大し、近似有理関数が極を持ち近似の意味を失う場合が多い。次にこれを例で示す。

例 1 Runge の例

$$f(x) = \frac{1}{1+25x^2} \quad (2.3)$$

- 近似範囲: $[-1, 1]$
- サンプル点の数と分子、分母多項式の次数:
 1. 5 点、 $(m, n) = (2, 2)$
 2. 9 点、 $(m, n) = (4, 4)$
 3. 17 点、 $(m, n) = (8, 8)$
- 得られた近似有理関数:

$$\begin{aligned} r_{2,2} &= \frac{1 + 2.220446 \times 10^{-16}x^2}{1 + 25x^2} \\ r_{4,4} &= \frac{1 - 0.573334x - 1.304265x^2 - 1.850372 \times 10^{-17}x^3 + 1.319579 \times 10^{-15}x^4}{1 - 0.573334x + 23.6957x^2 - 14.33335x^3 - 32.60662x^4} \\ r_{8,8} &= \frac{p_8(x)}{q_8(x)} \end{aligned}$$

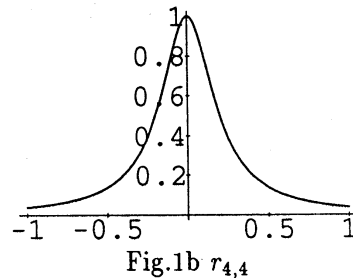
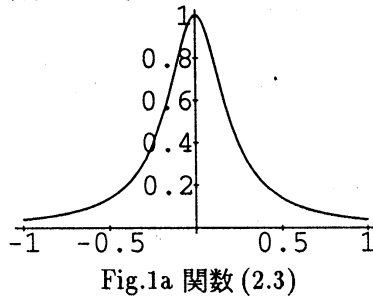
ここで

$$\begin{aligned}
 p_8(x) &= 1 + 0.4189994x - 1.225321x^2 + 0.2264419x^3 - 0.1168712x^4 + 0.3296147x^5 \\
 &\quad - 1.059114x^6 + 8.258448 \times 10^{-16}x^7 + 2.402926 \times 10^{-15}x^8 \\
 q_8(x) &= 1 + 0.4189994x + 23.77468x^2 + 10.70143x^3 - 30.7499x^4 + 5.990661x^5 \\
 &\quad - 3.980895x^6 + 8.240367x^7 - 26.47785x^8
 \end{aligned}$$

5点での近似有理関数 $r_{2,2}$ は元の関数に非常に近い形をしている。しかしサンプル点の数を 9、17 とすると結果は一見良くない。まず、 $r_{4,4}$ を検討する。この有理関数の分母の多項式 q_4 は次のような実の零点を持つ。

$$x \approx -1.122578 \quad \text{and} \quad 0.6829942$$

これらの中、 $x \approx 0.682994$ の零点は近似範囲内にあり、従来から扱われている有理関数近似の前提とは異なっている。当然、 $r_{4,4}$ は極を持つはずである。いま、(2.3) の関数と $r_{4,4}$ を近似範囲内で図示すると各々 Fig.1a, b になる。MathematicaTM で特にズームアップしない Fig.1b では予想される極は出現しない。



これは極を生むべき q_4 の零点が他のものと打ち消し合われていることを意味する。そこで、分子の多項式 p_4 の零点を計算してみる。これはすべて実根で

$$x \approx \pm 3.14387 \times 10^8, -1.122579 \quad \text{and} \quad 0.6829947$$

となる。予想通り q_4 の実根は p_4 のものときわめて近接しており、それらがあたかも両多項式の共通因子のようになっており、係数決定のときの数値計算の誤差と浮動小数点係数とを考慮すると (2.2) の近似計算版であると想像し得る。すなわち、両多項式の近似的 GCD [7] を計算し、 $r_{4,4}$ の分子分母から共通部分を除くと近似的に既約な有理関数近似が得られる。実際に cutoff 値 $\epsilon = 10^{-3}$ (精度 10^{-3} の近似的 GCD) で p_4 と q_4 の近似的 GCD (g) を求めると

$$g = 0.75781 \times 10^{15} - 0.43448 \times 10^{15}x - 0.98839 \times 10^{15}x^2$$

となる。分子、分母の多項式からこれを除くと

$$r_{4,4} \Rightarrow r'_{2,2} = \frac{-1.304264838 - 0.5985697351 \times 10^{-15}x + 0.1319579366 \times 10^{-14}x^2}{-1.304264830 - 0.3820494108 \times 10^{-8}x - 32.60662094x^2}$$

となる。結果を見やすくするため、分母の多項式の定数項を 1 にし、再び多項式中の微小係数項を切り捨てると

$$\tilde{r}_{0,2} = \frac{1}{1 + 25.00000015x^2}$$

となり、(2.3) に与えた $f(x)$ に非常に近い近似関数を得る。 $r_{8,8}$ の場合にも同じことがいえ、 p_8 と q_8 の近似的 GCD を求めると

$$g = 0.41615 \times 10^{15} + 0.17437 \times 10^{15}x - 0.50992 \times 10^{15}x^2 + 0.94235 \times 10^{14}x^3 - 0.48637 \times 10^{14}x^4 \\ + 0.13717 \times 10^{15}x^5 - 0.44076 \times 10^{15}x^6$$

なる 6 次式を得る。近似的除算 $p_8/g, q_8/g$ により近似有理関数は

$$r_{8,8} \Rightarrow r_{2,2}'' = \frac{-1.059114117 + 0.1573677134 \times 10^{-14}x + 0.2402926343 \times 10^{-14}x^2}{-1.059114117 + 0.6331500428 \times 10^{-13}x - 26.47785292x^2}$$

となり、分子の定数項を 1 にし、微小係数項を切り落とすと

$$\tilde{r}_{0,2} = \frac{1}{1 + 25x^2}$$

である。(2.3) と一致している。すなわち、有理関数近似ではサンプル点の数を増大すると一見不必要な極の数が多くなり有理関数近似は不適当と思える。しかし、分子、分母の両多項式が持つ近似的共通因子を計算し、それで分子、分母を割り、近似的規約形を求めると精度のよい近似有理関数となることがわかる。当然、浮動小数係数の多項式間の近似的共通因子を求めるには佐々木、野田により提案された近似的 GCD 算法の使用が考えられる [7]。そこで次に近似的 GCD 算法について簡単にまとめる。

3 近似的 GCD 計算と有理関数近似への応用

近似的 GCD アルゴリズムは悪条件代数方程式

$$P(x) = a_mx^m + a_{m-1}x^{m-1} + \cdots + a_0 = 0, \quad a_m \neq 0. \quad (3.1)$$

の解を得る目的で開発された。ここで多項式の係数は浮動小数を許している。 $P(x)$ を無平方分解、すなわち

$$P(x) = Q_1(x)Q_2^2(x) \cdots Q_l^l(x), \quad (3.2)$$

すると、近似的多重根因子が無平方因子 $Q_i, i = 1, \dots, l$ として取り出すことができる。 $Q_1(x)$ 部分は良条件の単根のみを含み、以下は近接根の性質を用いて全根を求めることができる。ここでは近似的無平方因子を見いだすことが重要になる。これは $P(x)$ と、それを x で微分した多項式 $dP(x)/dx$ の最大公約多項式 (GCD) を求めるとよい。もし、 $P(x)$ の係数が整数か有理数なら、この計算はユークリッド互除法で多項式剰余列 (PRS)

$$(P_1 = P(x), P_2 = dP(x)/dx, \dots, P_{k-1} \neq 0, P_k = 0) \quad (3.3)$$

を求め、 $\text{GCD}(P_1, P_2) = \text{pp}(P_{k-1})$ とすればよい。ここで pp は多項式の原始部分 (primitive part) を表す。しかし、 $P(x)$ の係数が浮動小数になると、(3.3) の PRS での 0 の判定が困難になる。そこで、 $0 < \epsilon \ll 1$ なる微小量 ϵ を導入し、多項式の全ての係数が ϵ 以下になったとき、その多項式を 0 とみなす。これを cutoff ϵ という。したがって、PRS を次のように書き直す。

$$(P_1, P_2, \dots, P_{k-1} \neq 0(\text{cutoff}\epsilon), P_k = 0(\text{cutoff}\epsilon)). \quad (3.4)$$

このとき、ユークリッド互除法を次のように規格化しておく。

$$P_{i-1} = Q_i P_i + \text{Max}(1, \text{mmc}(Q_i)) \times P_{i+1}, \quad i = 2, \dots, k.$$

ここで、 Q_i は多項式 P_{i-1} と P_i の商を表す。cutoff 操作を含む PRS から得られる最大公約多項式を近似的 GCD と呼び、cutoff 値とともに

$$\text{ApxGCD}(P_1, P_2; \epsilon) = \text{pp}(P_{k-1}) \quad (3.4)$$

と書く。したがって、このときの近似的無平方分解は次のようになる。

$$P(x) = Q_1(x)Q_2^2(x) \cdots Q_k^l(x) + O(\epsilon(x)). \quad (3.5)$$

これを用いると 2 の $r_{4,4}$ の近似的共通因子は

$$g = \text{ApxGCD}(p_4, q_4; 10^{-3}) = 0.7578 \dots$$

などと書ける。これを用いて 2 で述べた有理関数近似のアルゴリズムを書くと次のようになる。

アルゴリズム 近似的 GCD を用いた有理関数近似 - I

入力 : 有限個の点、 $x_0, x_1, \dots, x_k$ と対応する関数値または実験データ、 $f(x_0), f(x_1), \dots, f(x_k)$.
近似的 GCD のための cutoff 値 ϵ .

出力 : $(x_0, f(x_0)), (x_1, f(x_1)), \dots, (x_n, f(x_n))$ を近似する有理関数 $r(x) = p(x)/q(x)$ 。

方法 :

1. 入力データを近似する (m, n) 有理関数

$$r^{(0)}(x) = \frac{p^{(0)}(x)}{q^{(0)}(x)} = \frac{\sum_{j=0}^m a_j x^j}{\sum_{j=0}^n b_j x^j}, \quad b_0 = 1$$

を求める。

2. $p^{(0)}(x)$ と $q^{(0)}(x)$ の精度 ϵ の近似的 GCD を求める。

$$g = \text{ApxGCD}(p^{(0)}(x), q^{(0)}(x); \epsilon)$$

3. $p^{(0)}(x)$ と $q^{(0)}(x)$ を $g(x)$ により近似的除算し

$$r^{(1)}(x) = \frac{p^{(1)}(x)}{q^{(1)}(x)} = \frac{\frac{p^{(0)}(x)}{g(x)}}{\frac{q^{(0)}(x)}{g(x)}}$$

とする。

4. 以下の有理関数の正規化を行う。

(a) $r^{(1)}(x)$ を $q^{(1)}(x)$ の定数項で割る。

(b) $r^{(1)}(x)$ の微小係数項を無視する。

5. 正規化した有理関数を $r(x)$ とする。

このアルゴリズムを他の例に用い、さらに改良することを考える。ここで考える例は
例 2 分母に 2 重極を持つ関数の例

$$f(x) = \frac{1}{x \sin x} \quad (3.6)$$

- 近似範囲: $[0, \pi]$
- サンプル点の数と分子、分母多項式の次数:
- 21 点、 $(m, n) = (10, 10)$
ただし、 $x_0 = 0.001, \dots, x_{20} = 3.141$ として 0.157 幅に等間隔にとる。
- 得られた近似有理関数:

$$r^{(0)} = p^{(0)} / q^{(0)}$$

$$p^{(0)} = -1.8476 \times 10^{14} + 5.2531 \times 10^{14}x + 1.5974 \times 10^{14}x^2 - 1.2001 \times 10^{13}x^3 + 6.8986 \times 10^{12}x^4 \\ - 6.9830 \times 10^{11}x^5 + 1.4066 \times 10^{11}x^6 - 1.2594 \times 10^{10}x^7 + 1.5740 \times 10^9x^8 - 9.4088 \times 10^7x^9 \\ + 6.4614 \times 10^6x^{10}$$

$$q^{(0)} = 1 - 1.0210 \times 10^3x - 1.8476 \times 10^{14}x^2 + 5.2531 \times 10^{14}x^3 + 1.9053 \times 10^{14}x^4 \\ - 9.9553 \times 10^{13}x^5 - 2.1264 \times 10^{13}x^6 + 5.6795 \times 10^{12}x^7 + 3.5873 \times 10^{11}x^8 - 1.0047 \times 10^{11}x \\ + 3.4377 \times 10^9x^{10}$$

- 近似的 GCD

$$g = \text{ApxGCD}(p^{(0)}, q^{(0)}; 10^{-3}) = x - 0.3210$$

- 正規化した近似有理関数

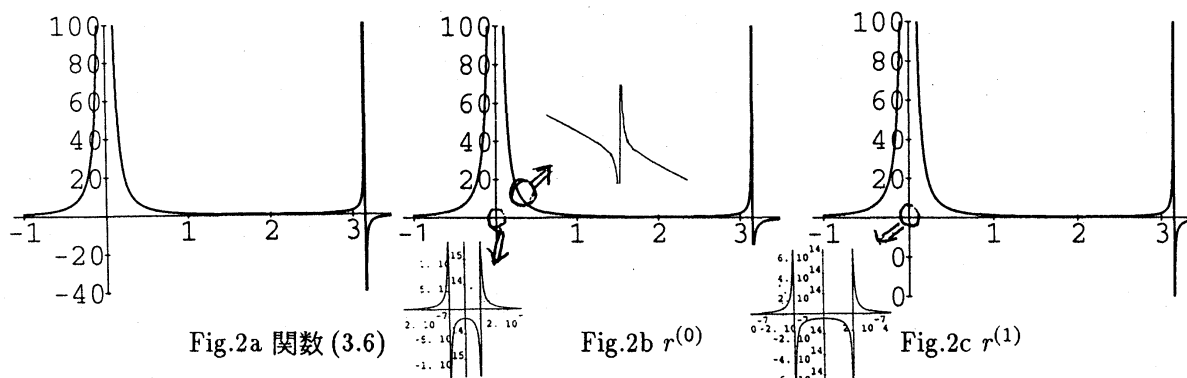
$$r^{(1)} = \frac{p^{(1)}}{q^{(1)}}$$

$$p^{(1)} = -4.681 \times 10^{13} - 1.273 \times 10^{13}x + 8.015 \times 10^{11}x^2 - 5.440 \times 10^{11}x^3 + 5.322 \times 10^{10}x^4 \\ - 1.112 \times 10^{10}x^5 + 9.841 \times 10^8x^6 - 1.256 \times 10^8x^7 + 7.484 \times 10^6x^8 - 5.255 \times 10^5x^9$$

$$q^{(1)} = 1 - 255.59x - 4.681 \times 10^{13}x^2 - 1.273 \times 10^{13}x^3 + 8.604 \times 10^{12}x^4 \\ + 1.578 \times 10^{12}x^5 - 4.705 \times 10^{11}x^6 - 2.658 \times 10^{10}x^7 + 8.083 \times 10^9x^8 - 2.796 \times 10^8x^9$$

- $r(x) = r^{(1)}$ とする。

アルゴリズム近似的 GCD を用いた有理関数近似-I を適用した結果は上の通りである。次に Fig. 2 に最初に与えた (3.6) (Fig.2a)、単純な有理関数近似 $r^{(0)}$ の結果 (Fig.2b) および近似的 GCD で近似的既約化をした $r^{(1)}$ (Fig.2c) を示す。ここでも一見、Fig.2b と Fig.2a は似ているが、 $x = 0$ 付近と $x = 0.3210 \dots$ 近くでは拡大図は大きく異なり単純な近似は定性的に元の関数の近似になっていないことがここからもわかる。後者は例 1 にもあった不必要な極による。近似的既約化をした $r^{(1)}$ では予想通り $x = 0.3210 \dots$ 近くの極は消えている。しかし、例 2 の関数は 1 位の極を $x = \pi$ に持つ他、 $x = 0$ に 2 位の極を持つ。



1位の極はうまく表されているが、 $x=0$ の2位の極は近似誤差により近接した2個の極に分離していることがわかる。これは $q^{(1)}$ の零点を求めれば明かである。それらは

$$x = -5.61, -3.15, -2.43, -0.14615 \cdots \times 10^{-5}, 0.14614 \cdots \times 10^{-5}, 3.14159 \cdots, 6.28, 9.10 \text{ and } 21.5$$

である。本来 $x=0$ にある筈の2位の極が $x = \pm 0.1461 \cdots \times 10^{-5}$ の1位の極2つのなっている。近似有理関数が元の関数と同じ定性的性質を持つためには、 $q^{(1)}$ の近似的無平方分解が必要となる。これは

$$q^{(1)} \rightarrow qq_2^2 qq_1$$

ただし

$$qq_2 = x + 2.8665 \times 10^{-12}$$

$$qq_1 = x^7 - 28.90x^6 + 95.07x^5 + 1.682 \times 10^3 x^4 - 5.644 \times 10^3 x^3 - 3.077 \times 10^4 x^2 + 4.554 \times 10^4 x + 1.674 \times 10^5$$

と達成できる。これをもとに近似有理関数を書き直し

$$r^{(2)} = \frac{p^{(2)}}{q^{(2)}},$$

とすると、 $p^{(2)}, q^{(2)}$ は各々

$$p^{(2)} = 1 + 0.2720x - 1.712 \times 10^{-2}x^2 + 1.162 \times 10^{-2}x^3 - 1.137 \times 10^{-3}x^4 + 2.376 \times 10^{-4}x^5 - 2.102 \times 10^{-5}x^6 + 2.683 \times 10^{-6}x^7 - 1.598 \times 10^{-7}x^8 + 1.122 \times 10^{-8}x^9,$$

$$q^{(2)} = x^2(1 + 0.2720x - 0.1837x^2 - 3.371 \times 10^{-2}x^3 + 1.005 \times 10^{-2}x^4 + 5.678 \times 10^{-4}x^5 - 1.726 \times 10^{-4}x^6 + 5.972 \times 10^{-6}x^7).$$

となる。このときの近似関数の振舞いをFig.2dに示す。高い精度で元の関数を再現していることがわかる。このアルゴリズムを近似的GCDを用いた有理関数近似-IIとする。

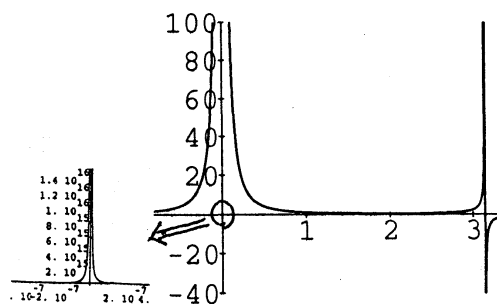


Fig.2d $r(x) = r^{(2)}$

アルゴリズム 近似的GCDを用いた有理関数近似-II

入力 : 近似的 GCD を用いた有理関数近似-I と同じ。

出力 : 近似的 GCD を用いた有理関数近似-I と同じ。

方法 : 1. - 4. : 近似的 GCD を用いた有理関数近似-I と同じ。

5 $q^{(1)}(x)$ の近似的無平方分解をし、結果の有理関数を $r^{(2)}(x)$ とする。

6 正規化-1: $r^{(2)}(x)$ の微小係数項を無視する。

7 正規化-2: 各無平方因子の定数項を 1 にする。

8 結果の有理関数を $r(x)$ とする。

4 近似的 GCD を用いた有理関数近似と誤差

以上で近似的 GCD を用いた有理関数近似の 2 つのアルゴリズムを示した。特に、近似関数の定性的性質を重視するような場合は第 2 のアルゴリズムを用いる必要がある。ただし、元の関数やデータが滑らかで極などがない場合には第 1 のアルゴリズムで終了する方が誤差の面で僅かに得策である（近似的無平方分解にともなうハイブリッド計算の誤差を考えなくてよい）。次にこのアルゴリズムを用いた計算と、その場合の誤差について触れる。いま、近似する関数を $f(x)$ 、結果の近似有理関数を $r(x)$ とすると、近似範囲内の点 x での誤差を

$$E(x) = |f(x) - r(x)|, \quad (4.1)$$

とする。次の例では区間内に 100 個の点を取り、誤差の平均と最大を各々

$$\text{平均: } E_{Ave} = \sum_{i=1}^{100} E(x_i) / 100, \quad (4.2)$$

$$\text{最大: } E_{Max} = \max_{i=1,100} E(x_i), \quad (4.3)$$

とし、関数近似の良さを調べる。用いる例は以下の通りである。

例 3 無理関数の例 ($x \in [0, 1]$)

$$f(x) = \sqrt{1+x} \quad (4.4)$$

例 4 離散データの場合 ($x_i \in [0, 1]$)

100 個の点とデータの組 $(x_i, f(x_i))$, $i = 1, 100$ を与える。

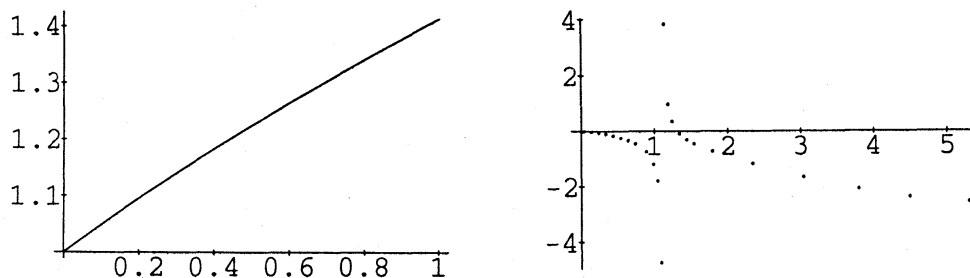


Fig.3 例 3, 例 4 のグラフ

例 3、4 の各々の範囲内でのグラフは上に示した。例 3 に対してまず $N = 11$ 、すなわち (m, n) 有理関数近似としては $(10, 0), (9, 1), \dots, (5, 5), \dots$ の多くの組合せを考える。これらと E_{Ave} 、 E_{Max} を次の Table.1 に示す。ここでの有理関数近似には近似的 GCD はなく、最初の近似 $r^{(0)}$ である。

m/n	error	
	E_{Ave}	E_{Max}
10/0	7.3196×10^{-11}	7.8066×10^{-10}
9/1	1.5129×10^{-12}	2.0558×10^{-11}
8/2	1.2563×10^{-13}	1.8668×10^{-12}
7/3	2.0033×10^{-14}	2.8799×10^{-13}
6/4	4.6762×10^{-15}	2.2870×10^{-14}
5/5	6.1561×10^{-15}	4.0856×10^{-14}
4/6	8.5098×10^{-15}	1.0025×10^{-13}
3/7	4.8163×10^{-14}	7.1465×10^{-13}
2/8	3.8618×10^{-13}	5.5635×10^{-12}
1/9	8.2886×10^{-12}	1.0250×10^{-10}
0/10	1.4266×10^{-9}	1.2113×10^{-8}

これは有利関数近似の特性をよく示しており、近似に用いる点数が同一なら、 (m, n) 有利関数近似では $m = n$ か $m = n \pm 2$ の場合が誤差が最小になる。もちろん、 $m + n$ が奇数の場合には $m = n \pm 1$ が最適になることは明かである。そこで、これらの最適な場合に m 、 n を変化させ、本論で提案したアルゴリズムを用いた結果を Table.2 に示す。

$r^{(0)}$			reduced rational function $r(x)$			
m/n	E_{Ave}	E_{Max}	$\deg(g)$	m/n	E_{Ave}	E_{Max}
10/10	1.8198×10^{-13}	8.1997×10^{-12}	3	7/7	5.9052×10^{-10}	1.4907×10^{-9}
10/9	5.9933×10^{-14}	1.8266×10^{-12}	3	7/6	2.1164×10^{-9}	5.6648×10^{-9}
9/10	1.1434×10^{-14}	2.0761×10^{-13}	3	6/7	1.0142×10^{-10}	2.7263×10^{-10}
9/9	1.0407×10^{-13}	3.1719×10^{-12}	3	6/6	6.1176×10^{-9}	1.3245×10^{-8}
9/8	5.3401×10^{-15}	3.7081×10^{-14}	3	6/5	2.4466×10^{-9}	6.5163×10^{-9}
8/9	6.0162×10^{-15}	2.6867×10^{-14}	3	5/6	3.1309×10^{-9}	8.3396×10^{-9}
8/8	1.4346×10^{-14}	2.0372×10^{-13}	2	6/6	8.5698×10^{-7}	3.1866×10^{-6}
8/7	9.8543×10^{-15}	9.3480×10^{-14}	1	7/6	4.9555×10^{-9}	7.9297×10^{-9}
7/8	1.3530×10^{-14}	1.8973×10^{-13}	1	6/7	2.2462×10^{-9}	3.7790×10^{-9}
7/7	4.1455×10^{-15}	3.4749×10^{-14}	2	5/5	1.2069×10^{-9}	3.1169×10^{-9}
7/6	5.9230×10^{-15}	6.4614×10^{-14}	1	6/5	2.6229×10^{-11}	7.4726×10^{-11}
6/7	6.6724×10^{-15}	1.8074×10^{-13}	1	5/6	1.8494×10^{-11}	5.1699×10^{-11}
6/6	1.6192×10^{-14}	7.1698×10^{-13}	1	5/5	7.1117×10^{-13}	1.8219×10^{-12}
6/5	8.5820×10^{-15}	1.1779×10^{-13}	0	—	—	—
5/6	8.9184×10^{-15}	1.1712×10^{-13}	0	—	—	—
5/5	8.9184×10^{-15}	1.1712×10^{-13}	0	—	—	—

なお、 $\deg(g)$ は近似的 GCD の次数をあらわす。この関数は本来単調で近似が簡単である。したがって最初の有利関数近似も十分過ぎるくらいの正確さを持っている。ただし、この Table からは見れないが、分母の多項式は零点を持っており、定性的にはもとの関数と近似式は大きく異なっている。これに対し、近似的 GCD を用いて極を取り去り、分子分母の次

数を低くした結果は定性的には正しい振舞いをするのはもちろん、工学問題などに要求される精度は保っているといえる。特に、近似的 GCD を用いた有理関数近似は (5, 5) 有理関数あたりに帰着する様子がわかる。このことをはまた最初の近似を (6, 5) 有理関数あるいはそれ以下のものとする近似的 GCD がなく既約なものになっていることから明かである。

次に例 4 についてみる。与えられた 100 個の点とデータの組からサンプル点として 31 個取った場合 (15, 15) 有理関数) と 23 個取った場合 (11, 11) 有理関数) を考える。各々の有理関数近似に対して最初の近似関数、近似的 GCD で次数を減少させた有理関数近似の結果と元のデータの間の誤差をやはり E_{Ave} 、 E_{Max} として Table.3 に示す。

$r^{(0)}$			reduced rational function $r(x)$			
m/n	E_{Ave}	E_{Max}	$\deg(g)$	m/n	E_{Ave}	E_{Max}
15/15	7.6994×10^{-7}	3.8230×10^{-5}	5	10/10	4.5902×10^{-6}	2.1845×10^{-4}
11/11	7.8516×10^{-7}	3.7107×10^{-5}	2	9/9	1.2848×10^{-6}	4.3555×10^{-5}

ここでも実用に耐え得る精度で次数が低く扱い易い有理関数近似が得られることが分かる。

さらにここで提案したアルゴリズムが、ある意味でのデータの平滑化をもたらしていることを再び例を用いて示す。

例 5 (2.3) の関数から得られる $(x_i, f(x_i))$ のデータの組にノイズを入れた。データを Table.4 に示し、ノイズ部分には下線を入れた。

Table.4 (例 5 のデータ)

x_i	$f(x_i)$	x_i	$f(x_i)$	x_i	$f(x_i)$
-1	0.038461538	-0.875	<u>-0.0496509</u>	-0.75	0.066390041
-0.625	0.092888243	-0.5	0.13793103	-0.375	0.22145328
-0.25	<u>-0.390244</u>	-0.125	0.71910112	0.0	1.0
0.125	0.71910112	0.25	0.39024390	0.375	<u>-0.221453</u>
0.5	0.13793103	0.625	0.092888243	0.75	<u>-0.06639</u>
0.875	0.049650892	1.0	0.038461538		

例 6 再び (4.4) の関数から得られるデータの組にノイズを加える。入力データは Table.5 にあり、ここでもノイズ部分は下線で示されている。

Table.5 (例 6 のデータ)

x_i	$f(x_i)$	x_i	$f(x_i)$	x_i	$f(x_i)$
0	1	0.05	1.0246950	0.1	<u>-1.04881</u>
0.15	1.0723805	0.2	<u>0.10954451</u>	0.25	<u>1118.0339</u>
0.3	1.1401754	0.35	<u>6.1618950</u>	0.4	1.1832159
0.45	1.2041594	0.5	<u>12.247448</u>	0.55	1.2449899
0.6	1.2649110	0.65	<u>1.28452×10^{10}</u>	0.7	<u>-6.30384</u>
0.75	1.3228756	0.8	<u>13.416407</u>	0.85	1.3601470
0.9	1.3784048	0.95	<u>139.42400</u>	1.0	1.4142135

このようなノイズを加えたデータの関数近似を得ようとする、通常の関数近似法はしばしばきわめて悪い結果を生じる。例 5, 6 の場合は、各々例 1, 3 にノイズをランダムに加

えたものなので、近似関数の確からしさを見るために次の量を考える。いま、例 5, 6 のデータは下線を施したデータ以外はすべて例 1, 3 に各々等しい曲線をサンプルしたものであるとする。このため、(4.1) の $f(x)$ として、(2.3) および (4.4) を取り、近似範囲 (例 5 では $x \in [-1, 1]$ 、例 6 では $x \in [0, 1]$) の 100 個の点での E の平均と最大を再び E_{Ave} 、 E_{Max} とする。例 5, 6 に近似的 GCD を用いた有理関数近似-I を適用すると次のような結果を得る。

例 5 $n = 8, m = 8$ とする。

- 最初の有理関数近似 ($r_{8,8}^{(0)}$)

$$r_{8,8}^{(0)} = \frac{p_8^{(0)}}{q_8^{(0)}}$$

$$p_8^{(0)} = 8 + 1.2506 \times 10^{15}x + 3.9305 \times 10^{15}x^2 - 1.2704 \times 10^{16}x^3 - 3.1126 \times 10^{16}x^4 \\ + 2.0327 \times 10^{16}x^5 + 4.0655 \times 10^{16}x^6 + 89.164x^7 - 211.83x^8$$

$$q_8^{(0)} = 1 + 1.2506 \times 10^{15}x + 3.9305 \times 10^{15}x^2 + 1.8561 \times 10^{16}x^3 + 6.7137 \times 10^{16}x^4 \\ - 2.9729 \times 10^{17}x^5 - 7.3751 \times 10^{17}x^6 + 5.0819 \times 10^{17}x^7 + 1.0163 \times 10^{18}x^8$$

- 近似的 GCD ($\text{ApxGCD}(p_8^{(0)}, q_8^{(0)}; 0.001)$)

$$g = 5.9038 \times 10^{12}x - 1.8555 \times 10^{13}x^2 + 5.9975 \times 10^{13}x^3 + 1.4694 \times 10^{14}x^4 \\ - 9.5961 \times 10^{13}x^5 - 1.9192 \times 10^{14}x^6$$

- 求める有理関数近似

$$r(x) = \frac{1}{1 + 25x^2} \quad (4.5)$$

例 6 $n = 10, m = 10$ とする。

- 最初の有理関数近似 ($r_{10,10}^{(0)}$)

$$r_{10,10}^{(0)} = \frac{p_{10}^{(0)}}{q_{10}^{(0)}}$$

$$p_{10}^{(0)} = 1 - 2.8449 \times 10^1x + 3.2432 \times 10^2x^2 - 1.9615 \times 10^3x^3 + 6.9219 \times 10^3x^4 \\ - 1.4522 \times 10^4x^5 + 1.7006 \times 10^4x^6 - 7.7438 \times 10^3x^7 - 4.5484 \times 10^3x^8 \\ + 6.7945 \times 10^3x^9 - 2.2447 \times 10^3x^{10}$$

$$q_{10}^{(0)} = 1 - 2.8944 \times 10^1x + 3.3873 \times 10^2x^2 - 2.1314 \times 10^3x^3 + 8.0047 \times 10^3x^4 \\ - 1.8669 \times 10^4x^5 + 2.6974 \times 10^4x^6 - 2.2883 \times 10^4x^7 + 9.5394 \times 10^3x^8 \\ - 5.2876 \times 10^2x^9 - 6.1755 \times 10^2x^{10}$$

- 近似的 GCD ($\text{ApxGCD}(p_{10}^{(0)}, q_{10}^{(0)}; 0.001)$)

$$g = 1.1738 \times 10^{-3} - 3.4196 \times 10^{-2}x + 0.40402x^2 - 2.5775x^3 + 9.8784x^4 \\ - 23.761x^5 + 36.104x^6 - 33.605x^7 + 17.473x^8 - 3.8831x^9$$

- 求める有理関数近似

$$r(x) = \frac{0.99959 + 0.67864x}{1 + 0.1867x} \quad (4.6)$$

以上の結果から E_{Ave} 、 E_{Max} を求めると次のようになる。

Table.6 (例 5,6 に対する有理関数近似)

	最初の有理関数近似		reduced rational function $r(x)$	
	E_{Ave}	E_{Max}	E_{Ave}	E_{Max}
例 5	1.343655	7.000000	1.616354×10^{-14}	5.695444×10^{-14}
例 6	1.369882×10^4	2.3184998×10^5	4.079059×10^{-4}	9.240164×10^{-4}

Table.6 を見れば、本論で提案したハイブリッド計算による有理関数近似の方法がいかにデータの平滑化に強力であるかについて詳しく述べる必要もないであろう。

5 むすび

以上実例を通してハイブリッド計算の立場から有理関数近似を見直し有効なアルゴリズムを提唱した。関数近似の技法は特に数値計算では基本的な技法の一つであり、十分な発展を遂げているように思える。実用的な面からはスプライン関数による近似その他の有力な方法が提案され用いられている。しかし、これらの近似で得られた関数は複雑で以後の解析には適していない。有理関数近似は得られる関数型が単純で近似関数の振舞いを見るには都合がよい。それにもかかわらず従来からは有理関数の既約性を重視するあまり、実用性のついてあまり考慮が払われてこなかった。これに対し、本論で提唱するアルゴリズムでは近似的 GCD 算法 [7] を用いることにより、厳密な有理関数の既約性の概念を近似的既約性に置き換えた。このような置き換えにより、関数近似の適用範囲は飛躍的に拡大する。このような有理関数近似の適用によって、実用的な解析に耐え得る精度で、定性的に正しい複雑な関数あるいは離散データ列の有理関数近似を得ることができる。有理関数近似を得られれば、以後の様々な解析が簡単に行える。たとえば、有理関数に対するハイブリッド積分 [2, 3] を用いた記号積分または高精度の定積分値を得ること、特異点付近の状況の解析、関数の極限操作などは容易に行える。また、近似的 GCD 算法の適用はある種のフィルターの効果を近似有理関数にもたらしることがわかる。厳密な本来の関数近似では有限個の点とその点での関数値またはデータ値を与えるので、これらの点では近似誤差はない。しかし、反面データに含まれるノイズの影響をどのように排除するべきかは大きな問題となる。しかし、例でも示したように本論で提案したハイブリッド計算での有理関数近似ではノイズの影響は自動的に吸収され、よい精度の近似を与えていることがわかる。

しかしここで示したアルゴリズムは発展途上のものであり、今後研究、整備しなければならない課題は非常に多くある。すぐに行えそうなものだけでも

- 最初の有理関数を高精度な Padé 近似、Tschebysceff 関数近似、連分数近似などにする
- 精密な誤差評価と最適近似との関係
- すでに完成されている古典的な有理関数近似の多くの研究成果 [4, 5] と本論での有理関数近似との関係

などがある。また、有理関数近似の計算途中で明かとなった他の問題点として、多項式の係数が非常に大きくなった場合、近似的 GCD アルゴリズムで十分に吸収しきれなくなる場合がある。これらに対しては近似的 GCD アルゴリズムの改良も必要と成ろう。

なお、本研究は部分的に科学研究費補助金（一般研究 (c), 課題番号 03808003）の補助によって行った。

参考文献

- [1] F.S. Acton, *Numerical Methods That Work*, Harper and Row, Pub., New York(1970), pp.279-315.
- [2] 宮広栄一、野田松太郎, ハイブリッド積分アルゴリズムとその応用について, 数理解析研講究録「数式処理とその数学研究への応用」, 1992 (掲載予定).
- [3] M.T. Noda, and E. Miyahiro, A Hybrid Approach for the Integration of a Rational Function, JCAM, March, 1992 (to appear)
- [4] J.R. Rice, *The Approximation of Functions II*, Addison-Wesley Pub. Co. (1969), pp.76-122.
- [5] T.J. Rivlin, *An Introduction to the Approximation of Functions*, Blaisdell Pub. Co. (1969), pp.120-141.
- [6] E.B. Staff, On the Degree of Best Rational Approximation to the Exponential Function, J. Approx. Theory, 9(1973), pp.97-101
- [7] T. Sasaki and M.T. Noda, Approximate Square-free Decomposition and Root-finding Ill-conditioned Algebraic Equations, J. Inf. Proc, 12(1989), 159-168.
- [8] J.L. Walsh, Approximation by Rational Functions: Open Problems, J. Approx. Theory, 3(1970), pp.236-242.