数理解析研究所講究録
第2054巻 2017年 25-30

安定化理論による Padé 近似計算について
Computation of Padé approximant by the theory of stabilizing algebraic algorithms

三宅 宏季
HIROKI MIYAKE
愛媛大学大学院理工学研究科
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING, EHIME UNIVERSITY *

甲斐 博
HIROSHI KAI
愛媛大学大学院理工学研究科
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING, EHIME UNIVERSITY †

Abstract

Padé approximation is a rational approximation constructed from the coefficients of a power series of a given function. The Padé approximation can be obtained by half GCD algorithm with complexity $M(n)\log n$, where $M(n)$ is polynomial multiplication cost, but the algorithm is unstable due to roundoff error if the input function has floating-point coefficients. In this paper, we show a stabilized half GCD algorithm to compute Padé approximation using the theory of stabilizing algebraic algorithms. We will show that the results have no Froissart doublets.

1 序論

Padé 近似は与えられた関数を有理関数の形で近似する手法であり，ある展開点における関数のべき級数展開の係数から計算することができる．Padé 近似の係数の決定は線形方程式や拡張ユークリッド互除法を解くことで計算できるが，Padé 近似が normal でない場合（Padé 近似が与えた次数よりも小さな次数の有理関数になる場合）に，浮動小数点計算を用いると，Froissart doublets と呼ばれる分母の零とそれに近接する分子の零が生じることが知られている [2].

我々は拡張ユークリッド互除法を用いた Padé 近似の計算アルゴリズムに対し安定化理論を用いて常に正しい次数の解を得る手法を提案した [5]. 本研究では half GCD を用いた Padé 近似の計算アルゴリズムに対して同様に安定化理論 [1] を導入し，正しく Padé 近似の次数判定ができることを示す．

*miyake@hpc.cs.ehime-u.ac.jp
†kai@hpc.cs.ehime-u.ac.jp

2 Padé 近似の計算法

関数 $f(x)$ を与えたとき，ある展開点におけるテーラー級数展開を $f(x)=\sum_{i=0}^{m+n}c_ix^i+O(x^{m+n+1})$ とする．有理関数

$$r_{m,n}(x)=r_{m,n}[f](x)=\frac{p_m(x)}{q_n(x)}=\frac{a_0+a_1x+a_2x^2+\cdots+a_mx^m}{1+b_1x+b_2x^2+\cdots+b_nx^n}$$

について，

$$q_n(x)f(x)-p_m(x)=O(x^{m+n+1})$$

となるような有理関数 $r_{m,n}$ を $f(x)$ に対する (m,n) Padé 近似と呼ぶ．Padé 近似を線形方程式を解いて求める場合，Padé 近似が normal でなければ，丸め誤差などにより極が発生しその極を打ち消すために分子にも近接根が生じることが知られている．その極と零のペアは Froissart doublets と呼ばれる．Froissart doublets を発生させずに Padé 近似を得る方法として，特異値分解により特異ベクトルを計算することで Padé 近似を導出する方法 [2] が Ibryaeva らにより提案されている．我々は Ibryaeva とは異なるアプローチとして安定化理論を用いる方法を検討する．

Padé 近似を得る他の方法として Geddes らによって提案された拡張 GCD を用いた方法 [3] がある．2 つの多項式 $a(x),b(x)$ に対する剰余列を $r_j(x)$ とすると拡張ユークリッド互除法を用いて，$s_j(x)a(x)+t_j(x)b(x)=r_j(x)$ を満たす多項式 $s_j(x),t_j(x),r_j(x)$ を計算することができる．但し，$deg(a)>deg(b)$, $r_0(x)=a(x), r_1(x)=b(x)$ と考え，$r_{j+1}=\mathrm{rem}(r_{j-1},r_j)$ とする．ここで，

$$\begin{aligned}a(x) &= x^{m+n+1}\\ b(x) &= b_0+b_1x+b_2x^2+\cdots+b_{m+n}x^{m+n}\end{aligned}$$

とすると，多項式 $t_j(x),r_j(x)$ は以下を満たす．

$$t_j(x)b(x)\equiv r_j(x) \bmod a(x)$$

よって，

$$\frac{r_j(x)}{t_j(x)}-b(x)=O(x^{m+n+1})$$

となり，有理関数 r_j/t_j はべき級数 $b(x)$ の Padé 近似の条件を満たす．したがって，$\deg(r_j)\le m, \deg(t_j)\le n$ となるような j を選択すれば，有理関数 r_j/t_j はべき級数 $b(x)$ の (m,n) Padé 近似となる．すでに我々は安定化理論を用いて拡張ユークリッド互除法に基づく Padé 近似を導出するアルゴリズムを検討しているが，本論では Geddes らによって示されている half GCD を用いた手法について安定化理論で安定化させることを検討する．half GCD を用いた Padé のアルゴリズムは以下の通りである．

アルゴリズム 1 (half GCD による Padé 近似 (PDC)[3])
入力: べき級数 $s(x)$，分子の次数 m，分母の次数 n
出力: 有理関数 t

1. $N:=m+n+1$
2. $u_0:=x^N, u_1:=s \bmod x^N, W:=\begin{bmatrix}1\\0\end{bmatrix}, V:=\begin{bmatrix}0\\1\end{bmatrix}$

3. $\deg(u_1) \le m$ のとき，$t := u_1$ とし終了

4. $\frac{N}{2} \le m \le N$ のとき，8. へ進む

5. $m \le \deg u_1 < \frac{N}{2}$ のとき，以下の処理を行う．

   (a) $q := \mathrm{quo}(u_0, u_1), r := \mathrm{rem}(u_0, u_1)$.

   (b) $u_0 := u_1, u_1 := r, [W, V] := \begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix} [W, V]$

   (c) $N := \deg u_0$

6. $m \le \deg u_1 < \frac{N}{2}$ でないとき，以下の処理を行う．

   (a) $[U_1, W_1, V_1] := \mathrm{PRSDC1}\left(u_0, u_1, \frac{N}{2}\right)$

   (b) $[u_0, u_1] := U_1^T, [W, V] := [W_1, V_1]\,[W, V]$

   (c) $N := \deg u_0$

7. 4. へ戻る

8. $[U_1, W_1, V_1] := \mathrm{PRSDC1}\,(u_0, u_1, m)$

9. $[W, V] := [W_1, V_1]\,[W, V]$

10. $\deg U_1[0] \le m$ かつ $\deg V[0] \le n$ であるとき，$t := (U_1[0]/V[0])$ とし終了

11. $t := (U_1[1]/V[1])$ とし終了

アルゴリズム 1 のステップ 6.(a), 8. で用いられている PRSDC1 関数は half GCD のサブルーチンに当たる関数であり，以下の通りである．

アルゴリズム 2 (Polynomial Remainder Sequence Divide and Conquer 1 (PRSDC1) [3])
入力: 多項式 $u_0(x), u_1(x)$, 次数 r
出力: 行列 T

1. $n := \deg u_0$

2. $\deg u_1 < r$ または $n = 0$ のとき，$T := \begin{bmatrix} u_0 & 1 & 0 \\ u_1 & 0 & 1 \end{bmatrix}$ として終了．

3. $m := \lceil r \rceil$

4. $u_0 \to b_0 x^m + c_0, u_1 \to b_1 x^m + c_1$

5. $[U_1, W_1, V_1] := \mathrm{PRSDC1}\left(b_0, b_1, \frac{1}{2}\deg b_0\right)$

6. $\begin{bmatrix} d \\ e \end{bmatrix} := U_1 x^m + [W_1, V_1] \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$

7. $\deg e < r$ であるとき，$T := \begin{bmatrix} \begin{bmatrix} d \\ e \end{bmatrix} & W_1 & V_1 \end{bmatrix}$ として終了

8. $q := \mathrm{quo}(d, e), f := \mathrm{rem}(d, e)$

9. $k := 2m - \deg e$

10. $e \to g_0 x^k + h_0, f \to g_1 x^k + h_1$

11. $[U_2, W_2, V_2] := \mathrm{PRSDC1}\left(g_0, g_1, \frac{1}{2}\deg g_0\right)$

12. $T := \left[U_2 x^k + [W_2, V_2]\begin{bmatrix} h_0 \\ h_1 \end{bmatrix}, [W_2, V_2]\begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix}[W_1, V_1]\right]$ として終了

もともと拡張 Euclid 互除法を用いた手法の計算量は $O(N^2)$ であるが，half GCD を用いたアルゴリズム 1 の計算量は $O(M(N)\log N)$ であることが知られている．ここで $M(N)$ は 2 つの N 次多項式の乗算の計算量である．以下ではアルゴリズム 1 のことを PDC と書く．

3 Padé 近似に対する安定化理論の適用

PDC の浮動小数点演算における問題は多項式の次数判定が正しく行えないことである．例えば，$x = 0$ における関数 $f(x) = \frac{x-2.01}{(x+0.1)(x+2.01)}$ の $(2,3)$ Padé 近似を計算する．$(1,2)$ 有理関数の Padé 近似を求めるので $(1,2)$ 有理関数で近似されるはずであるが，表 1 に示すように近似精度をいくらあげても $(2,3)$ 有理関数で近似される．また計算機環境は CPU: Core i7-4770 3.50GHz, メモリ: 8GB，OS: Windows 8.1 Pro 64 bit を用い，数式処理システムは Maple2015 を用いた．

表 1: 浮動小数点演算による PDC の計算結果

計算精度	PDC による結果の次数	計算精度	PDC による結果の次数
100	$(2,3)$	600	$(2,3)$
200	$(2,3)$	700	$(2,3)$
300	$(2,3)$	800	$(2,3)$
400	$(2,3)$	900	$(2,3)$
500	$(2,3)$	1000	$(2,3)$

本研究では，関数 $f(x)$ が正確に与えられていると仮定し，PDC に対して安定化理論を適用する．安定化理論では区間演算を行い精度を上げながら計算を行う．Maple で区間演算を行うために intpakX v1.0[6] を使用する．また，近似が収束したかどうかを判定するために，[5] でも用いた Padé 近似が満たすべき性質である次の Duality を用いる．

定理 1 (Duality)
$f(0) \neq 0$ である関数 $f(x)$ に対して，新たな関数 $g(x) = 1/f(x)$ を定義する．Padé 近似 $r_{n,m}[f](x), r_{m,n}[g](x)$ が存在するならば次式が成り立つ．

$$r_{m,n}[g](x) = \frac{1}{r_{n,m}[f](x)}$$

この性質を利用して，$r_{m,n}[g](x)$ と $1/r_{n,m}[f](x)$ をそれぞれ計算し，次数や係数を比較することで，真の解に収束したかどうかを判定する．具体的には以下の手順で PDC を計算する．

1. $g = 1/f$
2. f の (m,n) Padé 近似 $r_{m,n}[f]$ と g の (n,m) Padé 近似 $r_{n,m}[g]$ を以下のようにそれぞれ求める.
   (a) 計算精度を設定する.
   (b) べき級数の係数を区間数に置き換える.
   (c) PDC に従い計算を行う. また, 計算が行われる度に区間数のゼロ書き換えを行う
3. $r_{m,n}[f]$ と $r_{n,m}[g]$ の次数を比較することで正しく近似が得られたかを確認する. 正しい近似が得られるまで, 精度を上げてステップ 1,2 の計算を行う.

4 数値実験

安定化理論を用いたPDCが正しく動作しているかどうかを確認するために以下の例に対して実験を行った. 有理関数 $f(x) = \frac{x-2.01}{(x+0.1)(x+2.01)}$ の $(10,10)$ Padé 近似を計算する. この例でも明らかに $(1,2)$ 次の有理関数として出力されるべきである. 表 2 に示すように精度をあげていくと Padé 近似の次数が安定して, 入力の有理関数の次数と一致するという結果が得られた.

表 2: 安定化理論を用いた PDC の計算結果

計算精度	$\deg(r)$	$\deg(s)$	$\mathrm{zr}(r,s)$
20	$(3,1)$	$(2,1)$	$\neq 0$
40	$(8,1)$	$(2,1)$	$\neq 0$
60	$(6,8)$	$(2,1)$	$\neq 0$
80	$(8,10)$	$(2,1)$	$\neq 0$
100	$(1,2)$	$(2,1)$	$= 0$

表2では, f の (m,n) Padé 近似を r, g の (n,m) Padé 近似を s と表している. また, Duality が満たされているかどうかを判定するために, $r = p/q, s = u/v$, としたとき $r - 1/s = (pu-qv)/uq$ が0になるかどうかを計算する. ここで, 分子の多項式 $pu - qv$ を求め, 得られた多項式の係数にゼロ書き換えを行う関数を $\mathrm{zr}(r,s)$ と定義する. $\mathrm{zr}(r,s)$ が0になるかどうかで Duality を満たしているかどうかが判定できる.

表2より, 安定化理論を用いたPDCでは, 計算精度 $20 \sim 80$ 桁の時 s の次数は正しい次数となっているが, r の次数が異なることから正しい近似が得られていないと判定される. 計算精度を 100 桁に上げると r, s の次数が一致し, かつ, $\mathrm{zr}(r,s) = 0$ となるため, 計算精度 100 桁で結果を返す. 結果として正しい次数の近似が得られる.

次に, 同じ関数に対して次数を $(i,i), i = 10,20,30,40,50$ として安定化理論を用いたPDCを適用した場合の計算時間を表3に示す. 浮動小数点演算の精度を段階的に 10 から開始し, 正しく近似が得られるまで精度を 10 ずつ上げる. 比較のため, [5] で示した安定化理論による拡張ユークリッド互除法を用いたアルゴリズム (以下ではPEAと書く) の計算時間も示す.

この例の場合でも安定化理論を用いたPDCにより正しい近似を得ることができた. PEA, PDC の収束精度はほぼ同じであるが, 計算時間の比較としてはPEAの方が高速であるという結果が得

表 3: 計算時間と収束精度

	PEA		PDC	
次数	収束精度	計算時間 (s)	収束精度	計算時間 (s)
(10,10)	90	0.687	100	2.891
(20,20)	180	4.031	190	14.610
(30,30)	270	13.719	270	38.516
(40,40)	350	28.500	360	90.297
(50,50)	440	56.406	450	182.094

られた．本研究における PDC の実装では，多項式乗算アルゴリズムとして古典的な多項式乗算アルゴリズムを用いているので $M(N) = O(N^2)$ になる．従って，PDC の計算量は $O(N^2 \log N)$ となる．高速な多項式乗算アルゴリズムを用いた時にどのように結果が変わってくるかについての考察は今後の課題である．

5 結論

本研究では，近似する関数が与えられた場合，Padé 近似の関数形を求める問題について検討した．本研究では half GCD を用いた Padé 近似のアルゴリズム 1 に対して，安定化理論を導入することにより安定的に解が得られることを確認した．今後の課題としては，以下のようなことが挙げられる．

- FFT などより高速な多項式乗算アルゴリズムの適用
- べき級数のみが与えられた場合 (誤差が含まれている場合) の妥当な処理の検討

参 考 文 献

[1] K.Shirayanagi and M. Sweedler, A Theory of Stabilizing Algebraic Algorithms, Technical Report 95-28, Mathematical Sciences Institute, Cornell University, pp.1-92, 1995.

[2] O.L. Ibryaeva and V.M. Adukov, An algorithm for computing a Padé approximant with minimal degree denominator, Journal of Computational and Applied Mathematics, Vol.237 Issue 1, pp. 529-541, 2013.

[3] S. R. Czapor and K.O. Geddes, A Comparison of Algorithm for the Symbolic Computation of Padé Approximants, Proceedings EUROSAM '84, pp.248-259, 1984.

[4] George A. Baker and Jr. Peter Graves-Morris, Padé approximants 2nd edition, Cambridge, 1996.

[5] 三宅宏季, 甲斐博, 浮動小数点係数の Pade 近似計算について, 数理解析研究所講究録 1955, 27-32, 2015.

[6] intpakX, `http://www2.math.uni-wuppertal.de/~xsc/software/intpakX/`