

PADÉ 展開の係数の計算

川端 親雄 (岡山大学・理)

一松 信 (京大・数理研)

1. 問題

Padé 展開とは、整級数

$$(1) \quad f(x) = \sum_{k=0}^{\infty} a_k x^k$$

で表わされる函数に対して、有理函数近似

$$(2) \quad P_m^n(x) = \frac{P(x)}{Q(x)} = \frac{\alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n}{1 + \beta_1 x + \cdots + \beta_m x^m}$$

を作り、

$$f(x)Q(x) - P(x)$$

の Taylor 展開のはじめの方 (じつは x^{n+m} まで) がすべて消えるようにすることである。その係数は

$$(3) \quad \begin{cases} \sum_{k=0}^m a_{n-k+j} \beta_k = 0 & (j=1, \dots, m) \\ \alpha_i = \sum_{k=0}^i a_{i-k} \beta_k & (i=0, \dots, n) \end{cases}$$

で求められる。ただし $\beta_0 = 1$ とする。

$f(x)$ が極をもつとしても、有理函数近似にはさしつかえ

ないが、代数的分岐点をもち

$$f(x) = A(x-x_c)^\alpha (1 + O(x-x_c))$$

の形のときには、対数微分

$$(4) \quad g(x) = \frac{d}{dx} \log f(x) = \frac{\alpha}{x-x_c} + O(1)$$

の Padé' 展開を求めるのがおつうである。

物理学の問題への応用上、(1) から (4) で作られる $g(x)$

の Taylor 展開

$$(5) \quad g(x) = \sum_{k=0}^{\infty} c_k x^k$$

の係数を組織的に求めることが必要になる。

対数微分には定数因子は問題にならないから、 $a_0 = 1$ と

してよい。

$$(6) \quad \log f(x) = \sum_{k=1}^{\infty} b_k x^k$$

とすれば、

$$(7) \quad c_k = (k+1) b_{k+1}$$

となる。(6) の係数を計算すると、 $|x|$ が十分小さく絶対収束だから順序をかえてよい；形式的整級数としてもよい)

$$\begin{aligned} \log f(x) &= \log \left(1 + \sum_{k=1}^{\infty} a_k x^k \right) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \left(\sum_{k=1}^{\infty} a_k x^k \right)^n \\ &= \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \sum_{n=k_1+\dots+k_N} \frac{n!}{k_1! \dots k_N!} a_1^{k_1} \dots a_N^{k_N} x^{k_1+2k_2+\dots+Nk_N} \end{aligned}$$

となるから、1つをよく C に直すと、

$$(8) \quad k b_k = C_{k-1}$$

$$= \sum k \frac{(-1)^{k_1 + \dots + k_N - 1}}{k_1 + \dots + k_N} \cdot \frac{(k_1 + \dots + k_N)!}{k_1! \dots k_N!} a_1^{k_1} \dots a_N^{k_N}$$

$$(9) \quad \sum k \text{ は } k = k_1 + 2k_2 + 3k_3 + \dots + Nk_N \quad (k_i > 0)$$

という分割全体にわたる、

という形で表わされる。 b_k の係数は分母に $k_1 + \dots + k_N$ がつくが、 k 倍した C_{k-1} の係数はすべて整数になることが証明できる。

われわれの仕事は、 C_k の具体的な式を計算したことである。その点、数値解析というよりもむしろ 数式処理 まがいの話である。ただし、計算機にのせるためのアルゴリズム的な苦心があったので報告したい。

2. 計算の手法

算法の手順はつぎのとおりである。

(i) 整数 k を (9) のように分割し、この分割をもれなく順次作らだす。

(ii) 各分割に応じて (8) の係数を (できれば整数形で) 求める。

あと印刷上の問題があるが、これは純粹にプログラミング技法にすぎない。結果は各分割ごとに改行して、下記のよう
な形に印刷した:

$$C_0 =$$

$$1 * (A_1 ** 1)$$

$$c_0 = a_1$$

$$C_1 =$$

$$2 * (A_2 ** 1)$$

$$- 1 * (A_1 ** 2)$$

$$c_1 = 2a_2 - a_1^2$$

$$C_2 =$$

$$3 * (A_3 ** 1)$$

$$- 3 * (A_1 ** 1) * (A_2 ** 1)$$

$$1 * (A_1 ** 3)$$

$$c_2 = 3a_3 - 3a_1a_2 + a_1^3$$

だが10くらいまでなら、手でも何とかできるが、 C_{20} とも
なると、792項になり、手計算ではたいへん間違えるので、
計算機でやったわけである。

3. 分割の生成

(i) については、一松 加、情報処理 7, 2 (1966) のプロ
グラムのページに発表したもののうち、第2の算法を採用し
た。その算法は下記のとおりである。

0° K 但(以上)の配列 I を用意する。—— K は上の n に

あたり, c_{k-1} を計算する. $I(J)$ の内容が e_j に相当する.

1° 個数の制御変数 N を K から 1 まで 1 ずつ下げる.

2° 当初 $I(2) = \dots = I(N-1) = 0$, $I(N) = 1$,

$I(1) = K - N$ とセット. ただし $N = 1$ のときには,

$I(1) = K$ とセットする.

3° $K = N \times I(N)$ になったらその段完了 ($N := N - 1$ とし, $N = 1$ なら完了).

4° $IS = K - N \times I(N) < N$, $I(IS) = 1$ なら, その段完了. (して新分割)

5° $I(1) > 1$ なら, $I(1)$ を 2 へとし, $I(2)$ を 1 へや.

6° $I(1) \leq 1$ なら, $I(2), \dots, I(J)$ をみてゆき,

$I(2) = \dots = I(J-1) = 0$, $I(J) > 0$ をさがす.

7° $I(1) = 1$ または $I(J) > 1$ なら, $I(J+1)$ を 1 へやし

$I(J) = 0$ とし残りを $I(1)$ に入れる. (じつせいには

$$I(1) := (I(J) - 1) * J + I(1) - 1$$

となって, 新分割をうる.

8° 6°で, 7°でないときには, さらに $I(J+1) = \dots =$

$I(J'-1) = 0$, $I(J') > 0$ がある J' をさがし,

$I(J')$ と $I(J)$ を 0, $I(J'+1)$ を 1 へやし, 残りを

$I(1)$ に入れる. (じつせいには

$$I(1) := (I(J') - 1) * J' + J - 1$$

となって新分割をうる。8° は、変数の名をうまくつけかえ、 J' を J とし、7° の $I(1)$ と 8° の J を別の名前に保存しておけば、共通の式にできる。

4. 係数の計算

(ii) については、 $k \leq 10$ 程度ならば、直接

$$\frac{k(k_1 + \dots + k_N - 1)!}{k_1! \dots k_N!}$$

の分母と分子を整数形で計算できる。分割が (9) の形であるため、この係数は、 $k!$ よりもはるかに小さく、 $k \leq 21$ で最大 10^5 程度である。しかし研究所の TOSBAC-3400 では、整数変数は 2 語とっているにもかかわらず 1 語長 (24 ビット) しか使えないので、 $k=21$ (C_{20}) までおめようとするとき、このままではあふれてしまう。そこで $k_1 + \dots + k_N \geq 11$ になる組については、つぎのようを算法をとった。なお符号はもちろん (-1) のべきを作ったりせず、 $k_1 + \dots + k_N$ が偶数のときだけ - をつけるようにした。

0° 因数を表わす配列 I を用意する ((i) の I とは別物である)。

1° $I(1), \dots, I(k_1 + \dots + k_N - 1)$ および $I(k)$ を 1, 他を 0 とセットする。(じつは $I(1)$ は不要)。

2° 各 k_i について、 $I(2), \dots, I(k_i)$ を 1 ずつつづ

す。 $k_i = 1$ なら、この操作をとはす。

3° $I(2)$ から始めて $I(J) < 0$ だったら、

$I(2J)$, $I(3J)$, --- としうて、 $I(MJ) > 0$ である所をさがす。それがあれば、 $I(MJ)$ を1つへらし、

$I(M)$, $I(J)$ を1つずつ小やす。—— じつは $I(MJ)$ と $-I(J)$ との小さい方を N とし、 $I(MJ)$ を N へらし、 $I(M)$, $I(J)$ を N 小やしてもよい。

4° $I(2)$ まではいつでも $I(MJ) > 0$ である所がなければ、 J を素因数分解する。じつさいには完全に素因数にわけなくても、 $2, 3, \dots, M$ とおがして、 $M|J$, $J/M > 1$ である所をみつけ、 $I(M)$ と $I(J/M)$ をそれぞれ $-I(J)$ 小やし、 $I(J) = 0$ とする。 $M = J/M$ なら、 $-2I(J)$ 小やすことになるが、これは小やす操作を逐次実行すればよい。そして M と J/M との小さい方を J として 3° にもどる。

5° 4° でこのような M がなければ誤である。—— このための誤り表示は加えたが、幸いこれは一度も動作しなかった。

6° 以上の 3°, 4° をくりかえし、ついにすべての I が ≥ 0 になつたら、整理をやめ

$$2^{I(2)} \cdot 3^{I(3)} \cdots N^{I(N)} \cdots$$

を計算する。これも $I(J) = 0$ はとばし、また ** で書く

と、 $e^{I(N) \log N}$ の形に変換すれば誤差が入るおそれがあるので、1からはじめ、じつさい N を $I(N)$ 回かけるようにプログラムを書いた。

代案 1° 2, 3, --- をすべて素因数分解し、2, 3, 5, --- の個数を求めて掛ける方法。—— かえって手間がかかる。じつさい 4° の操作が不要なのは、 $n \leq 21$ の範囲では、 $I(4)$, $I(6)$ について、ごく少数のものに対してだけであった。

2° 浮動小数点で計算して、あとで整数に直す。—— 誤差の入るのをおそれて、試験だけでやめたが、これがかえって賢明だったかもしれない。

3° 全部上記の因数の形で求める。—— 時間がかかるので、 $n_1 + \dots + n_N \leq 10$ については、直接分子 $n(n_1 + \dots + n_N - 1)!$ を分母 $n_i!$ で順次割ってゆく方法をとった。なお陪乗は 10! まででよいので、毎回計算せず、はじめに 10! まで計算した表を作って利用した。

4° $I(j)$ の表の整理のところで、 $I(4)$, $I(6)$, $I(8)$, --- ; $I(6)$, $I(9)$, $I(12)$, --- ; と 4° のような因数分解の操作を、はじめにやってしまう手がある。—— 時間は少し余分にかかるそうだが、プログラムはるのほうで短くてすみそうである。

$n \leq 21$ (C_{20} まで) 上の方法で、約 10 分を要した。

5. Cumulant 展開

$$C(\xi) = \sum_{n=0}^{\infty} \frac{1}{n!} \mu_n \xi^n, \quad \mu_0 = 1$$

に対して,

$$\log C(\xi) = \sum_{n=1}^{\infty} \frac{1}{n!} \lambda_n \xi^n$$

とした係数 λ_n を n 次の Cumulant とよび, λ_n を μ_n の多項式で表現した形を Cumulant 展開 という. そのはじめのほうをあげると,

$$\lambda_1 = \mu_1$$

$$\lambda_2 = \mu_2 - \mu_1^2$$

$$\lambda_3 = \mu_3 - 3\mu_1\mu_2 + 2\mu_1^3$$

である. $\lambda_k = \sum \beta_{k_1 \dots k_n} \mu_1^{k_1} \dots \mu_n^{k_n}$ とするとき, $\sum \beta_{k_1 \dots k_n}$ は, $k_1 = 1$ のとき 1, $k_1 \geq 2$ のとき 0 である. なぜなら $\mu_1 = 1$ のとき, $C(\xi) = e^\xi$, $\log C(\xi) = \xi$ だからである.

物理学の問題においては, $-\infty$ から $+\infty$ までの積分が 1 になる函数 $f(x)$ を重みとした平均を $\langle \rangle$ で表わし,

$$C(\xi) = \langle e^{\xi x} \rangle = \int_{-\infty}^{\infty} e^{\xi y} f(y) dy$$

とするのが普通である. このとき $\mu_n = \langle x^n \rangle$ となる. これ

に対して, $\lambda_n = \langle x^n \rangle_c$ と書かれることが多い. この関係を
とまめて (形式的に)

$$\log \langle e^{\beta x} \rangle = \langle e^{\beta x} - 1 \rangle_c$$

と書く.

λ_n の具体的な式は, 本文の a_n と b_n との関係を用いて,
同じように表現することができる.

[参] Cumulant 展開とその応用については, 次の論文に
詳しい.

R. Kubo (久保亮五), Generalized Cumulant Expansion Method, J. Phys. Soc. Japan, 17, No. 7 (1962, July)
p. 1100 — 1120.

6. 多変数の Cumulant 展開

前節の x を $\sum_{j=1}^N x_j X_j$ におきかえれば, 多変数の
Cumulant 展開をうる. 変数の個数と次数またはそれ以上に
とれば, もっとも一般の形がえられる. それはつきのような
形になる (上記の Kubo の論文に証明がある).

$$\langle X_1 \cdots X_m \rangle_c = \sum (-1)^{|k|-1} (|k|-1)! \\ \times \underbrace{\langle X_{i_1} \cdots X_{i_N} \rangle \langle X_{i_{N+1}} \cdots X_{i_N'} \rangle \cdots \langle X_{i_1} \rangle \langle X_{i_m} \rangle}_{\text{下記参照}}$$

Σ はつぎの範囲にわたる.

1.° $m = 1 \cdot k_1 + 2 \cdot k_2 + \dots + m \cdot k_m$ と分割する. ここで

$|k| = k_1 + k_2 + \dots + k_m$ を意味する.

2.° 各分割について, $X_1, \dots, X_m$ そうきのように分配

する. m 個の要素の組 k_m 組, $(m-1)$ 個の要素の組

k_{m-1} 組, $\dots$, 1 個の要素の組 k_1 組. ($k_j = 0$ 不

るば, そこはとばす) 各集合ごとにはまとめて $\langle \rangle$ に

入れる. それを順次かけあわせる.

3.° 2.° のような組分けを可能な限りすべて作る. ただし,

$k_j \geq 2$ のとき, k_j 組の組全体の入れかえは同一とする.

(たとえば $\langle 1, 2 \rangle \langle 3, 4 \rangle$ と $\langle 3, 4 \rangle \langle 1, 2 \rangle$ とは同じとして

1 回しか作らない). またかって, このような組は合計

$$\frac{m!}{k_1! 2!^{k_2} k_2! 3!^{k_3} k_3! \dots m!^{k_m} k_m!} \quad (j!^{k_j} k_j! \text{ は } k_j = 0 \text{ なら } 1 \text{ に等しい})$$

個できる.

1 変数の Cumulant 展開は $X_1 = \dots = X_m = X$, つまりこれだけ個の項を同一にまとめればえられるから, 表を作るとすれば, 多変数の形で求めたほうが賢明であろう.

前記分割_{生成}のプログラムに 2.° 3.° の操作を加えて, 多変数の Cumulant 展開の式を計算機で求めることを, ひきつづき実行する予定である.

付録 討論と意見

1. 占部教授より, 係数 c_k の計算は $f(x) g(x) = f'(x)$ したがって

$$(\sum a_n x^n) (\sum c_n x^n) = \sum n a_n x^{n-1}$$

からえられる漸化式を利用した方がよいのではないかと
の御指摘があった。

2. 3節のアルゴリズムでは, 分割の順序がきれいでない.
辞書式順序にすべきかもしれない. またこの項の新しい
FORMACでは, べき乗は2行として X^9 というように
印刷しているので, 印刷結果をきれいにするよう,工夫
すべきかもしれない.

3. $c_k = \sum \gamma_{k_1, \dots, k_N} a_1^{k_1} \dots a_N^{k_N}$ とすると,
 $\sum \gamma_{k_1, \dots, k_N} = 1$ となる. 否ぜなら, すべての $a_n = 1$ の
とき, $f(x) = (1-x)^{-1}$, $g(x) = (1-x)^{-1}$ で, $c_k = 1$
となるからである. したがって k を固定し, えられた
係数を全部加えて1になるか否かを判定するのは, 一
つの検査として役立つ (後にこの検査を加えた).

Cumulant の方でも, 加えて 0 ($m \geq 2$ について) という
検査は入れる予定である.