お話：数値解析 第6回
収束の加速法(中編)

長田直樹

1　はじめに

前回は、円に内接する正多角形の周長から円周率を計算する過程で、エイトケン Δ^2 法とリチャードソン補外が発見されたことを話した。今回は、円周率を表す級数のうち最初に発見されたライプニッツ級数を巡る加速法の話をする。

2　ライプニッツ級数

逆正接関数 $\tan^{-1} x$ の級数展開

$$\tan^{-1} x = x - \frac{x^3}{3} + \frac{x^5}{5} - + \cdots, \quad (-1 \le x \le 1) \tag{1}$$

は J. グレゴリが 1671 年 J. コリンズ宛の手紙に書いたため、**グレゴリ級数**と呼ばれている。$-+\cdots$ は符号が交代しながら続くという記号である。

(1) において、$x=1$ とおくと、交代級数

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - + \cdots \tag{2}$$

が得られる。グレゴリは (2) については一度も言及していない。一方、1673 年に G. ライプニッツが、グレゴリと独立に級数 (1) と (2) を発見した。そこで、(2) は**ライプニッツ級数**と呼ばれている。二人の発見の詳細は [8] にある。

補題 1 ライプニッツ級数は漸近公式

$$\sum_{i=1}^{\nu} \frac{(-1)^{i-1}}{2i-1} = \frac{\pi}{4} + (-1)^{\nu-1}\left(\frac{1}{4\nu} - \frac{1}{16\nu^3} + \frac{5}{64\nu^5} - \frac{61}{256\nu^7} + O(\frac{1}{\nu^9})\right) \tag{3}$$

を満たす。

証明 [6] を見よ。□

ライプニッツ級数の計算は容易であるが、収束は極めて遅い。補題 1 より、$s_\nu = \sum_{i=1}^{\nu}(-1)^{i-1}/(2i-1)$ の誤差は、ν が大きいとき $(-1)^{\nu-1}/4\nu$ である。4 倍して円周率を 10 桁得るにはおよそ 100 億 ($=10^{10}$) 項までの和が必要で、実用にはならない。

そこで、ライプニッツ級数あるいはグレゴリ級数を用いて円周率を計算する際、3 つの方法がとられた。1 つ目は (1) の $|x|$ を小さく取ることにより収束を速めることである。1699 年に A. シャープは

$$\frac{\pi}{6} = \tan^{-1}\frac{1}{\sqrt{3}}$$

を用いて円周率を 72 桁計算した。2 つ目は $\tan\theta$ の加法定理から得られる公式

$$\tan^{-1}\frac{x+y}{1-xy} = \tan^{-1} x + \tan^{-1} y$$

を用いて収束の速い級数の和 (差) への変形である。1706 年に J. マーチンは

$$\pi = 16\tan^{-1}\frac{1}{5} - 4\tan^{-1}\frac{1}{239}$$

を用いて 100 桁計算した。3 つ目はライプニッツ級数あるいはグレゴリ級数を加速することである。これが今回の主題である。

3　マーダヴァの円周率

級数 (1)(2) は、グレゴリやライプニッツより二百数十年前に南インドの数理天文学者マーダヴァ(1380-1420) により得られていた。これらは 1550 年頃シャンカラ (マーダヴァ学派の学者) によってサンスクリット (梵語) で書かれた『クリヤークラマカリー』に記録されている。[13] に日本語訳と詳細な解説がある。

マーダヴァは直径 d の円周を近似する公式を 2 つ与えた。

$$C_1(\nu) = \sum_{i=1}^{\nu}(-1)^{i-1}\frac{4d}{2i-1} + (-1)^{\nu} d\frac{1}{\nu}$$

$$C_2(\nu) = \sum_{i=1}^{\nu}(-1)^{i-1}\frac{4d}{2i-1} + (-1)^{\nu}4d\frac{\nu}{4\nu^2+1}$$

彼は ν を大きくすると、「非常に精密になるだろう」[13, p.14] と収束についても言及している。

$d=1$ とおくとライプニッツ級数 (の 4 倍) に補正項を付けた

$$\pi \fallingdotseq \sum_{i=1}^{\nu}(-1)^{i-1}\frac{4}{2i-1} + \frac{(-1)^{\nu}}{\nu}$$

などが得られる。シャンカラはさらに

$$C_3(\nu) = \sum_{i=1}^{\nu}(-1)^{i-1}\frac{4d}{2i-1} + (-1)^{\nu}4d\frac{\nu^2+1}{4\nu^3+5\nu}$$

を与えている。

以下では簡単のため $d=1$ としておく。マーダヴァの公式 $C_1(\nu)$ は (3) の $1/\nu$ までの漸近展開なので誤差は $1/\nu^3$ のオーダー、$C_2(\nu)$ は

$$\frac{4\nu}{4\nu^2+1} - \frac{1}{\nu} + \frac{1}{4\nu^3} = \frac{1}{4\nu^3(4\nu^2+1)} = O(\frac{1}{\nu^5})$$

より $1/\nu^5$ のオーダーである。また、シャンカラの公式 $C_3(\nu)$ は

$$\begin{aligned} & \frac{4\nu^2+4}{4\nu^3+5\nu} - \frac{1}{\nu} + \frac{1}{4\nu^3} - \frac{5}{16\nu^5} \\ = & \frac{-25}{16\nu^5(4\nu^2+5)} = O(\frac{1}{\nu^7}) \end{aligned}$$

より $1/\nu^7$ のオーダーである。30 項までの値は表 1 のようになる。イタリックの部分は円周率の真値と異なる数字である。

表 1: ライプニッツ級数とその加速

ν	ライプニッツ級数	マーダヴァ $C_1(\nu)$	マーダヴァ $C_2(\nu)$
1	*4.00000000*	3.*00000000*	3.*20000000*
2	*2.66666667*	3.1*6666667*	3.1*3725490*
3	3.*46666667*	3.1*3333333*	3.14*234234*
4	3.*33968254*	3.14*523810*	3.141*39194*
5	3.*33968254*	3.1*3968254*	3.141*66274*
10	3.*04183962*	3.141*83962*	3.14159*024*
20	3.*09162381*	3.141*62381*	3.14159*258*
30	3.1*0826857*	3.141*60190*	3.1415926*4*

ν	シャンカラ $C_3(\nu)$
1	3.1*1111111111111*
2	3.14*285714285714*
3	3.141*46341463415*
4	3.141*61490683230*
5	3.1415*8730158730*
10	3.141592*70534916*
20	3.14159265*401986*
30	3.141592653*61527*

30 項まで計算した際、ライプニッツ級数 (2) では 2.0 桁しか合わないが、マーダヴァの公式 $C_1(\nu)$ では 5.5 桁、$C_2(\nu)$ では 8.5 桁、シャンカラの公式 $C_3(\nu)$ では 11.1 桁一致する。(一致する桁数は、$-\log_{10}|$(近似値 − 真値)/真値 $|$ により与えられる。) ライプニッツ級数の部分和のなす数列 $\{s_\nu\}$ を $\{C_1(\nu)\}, \{C_2(\nu)\}, \{C_3(\nu)\}$ 等に移す変換は加速法である。恐らく、マーダヴァの $\{s_\nu\} \to \{C_1(\nu)\}$ は世界で最初の加速法である。

マーダヴァは $C_1(\nu)$ と「同じ原理によって、望みの正弦からその弧を求めることもできる。」[13, p.31] として、グレゴリ級数について述べている。

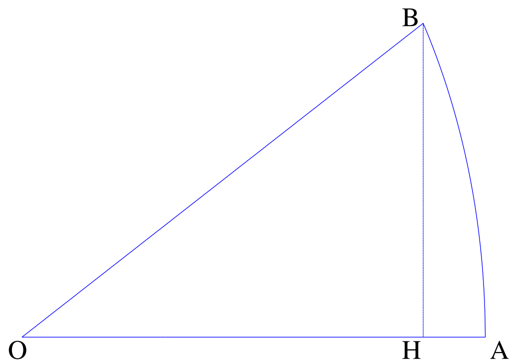

図 1: マーダヴァの弧長の計算

図 1(OA が半径、$\widehat{\mathrm{AB}}$ が弧、BH が正弦、OH が余弦) において、

$$\begin{aligned} \widehat{\mathrm{AB}} &= \mathrm{OA}\left(\frac{\mathrm{BH}}{\mathrm{OH}} - \frac{1}{3}\frac{\mathrm{BH}^3}{\mathrm{OH}^3} + \frac{1}{5}\frac{\mathrm{BH}^5}{\mathrm{OH}^5} - \cdots \right. \\ & \left. + (-1)^{\nu-1}\frac{\mathrm{BH}^{2\nu-1}}{(2\nu-1)\mathrm{OH}^{2\nu-1}} + \cdots \right) \end{aligned}$$

が弧を求める方法である。さらに、OH < BH のときは、「どんなに繰り返しても、果は終結しないだろう」と収束条件を注意している [13, p.31]。

$\theta = \widehat{\mathrm{AB}}/\mathrm{OA}$ とおくと $\tan\theta = \mathrm{BH}/\mathrm{OH}$ だから

$$\theta = \tan\theta - \frac{1}{3}\tan^3\theta + \frac{1}{5}\tan^5\theta - + \cdots$$

と書き直せる。さらに、$x = \tan\theta$ とおくとグレゴリ級数 ($\tan^{-1} x$ の展開式) になる。

4 オイラー変換

数列 $\{s_\nu\}$ に対し、高階差分演算子を

$$\Delta^0 s_\nu = s_\nu$$
$$\Delta^m s_\nu = \Delta^{m-1} s_{\nu+1} - \Delta^{m-1} s_\nu, m = 1, 2, \ldots$$

により定義する。

L. オイラーは 1755 年に出版した微分学教程 [3, p.232] において、べき級数

$$\sum_{i=1}^{\infty} (-1)^{i-1} a_i x^i = a_1 x - a_2 x^2 + a_3 x^3 - + \cdots \quad (4)$$

を $x/(1+x)$ に関するべき級数

$$\begin{aligned} & \sum_{i=1}^{\infty} (-1)^{i-1} \frac{x^i}{(1+x)^i} \Delta^{i-1} a_1 \\ = \; & \frac{x}{1+x} a_1 - \frac{x^2}{(1+x)^2} \Delta a_1 + \frac{x^3}{(1+x)^3} \Delta^2 a_1 \\ & - + \cdots \end{aligned} \quad (5)$$

に変換する方法を発表した。(5) を**一般オイラー変換**という。$x = 1$ とおくと、

$$\begin{aligned} & \sum_{i=1}^{\infty} \frac{(-1)^{i-1} \Delta^{i-1} a_1}{2^i} \\ = \; & \frac{1}{2} a_1 - \frac{1}{2^2}(a_2 - a_1) + \frac{1}{2^3}(a_3 - 2a_2 + a_1) \\ & - + \cdots \end{aligned} \quad (6)$$

が得られる。交代級数

$$\sum_{i=1}^{\infty} (-1)^{i-1} a_i = a_1 - a_2 + a_3 - + \cdots \quad (7)$$

から (6) への変換 [3, p.233] を**オイラー変換**という。

ライプニッツ級数にオイラー変換を適用すると

$$\begin{aligned} \frac{\pi}{4} &= \frac{1}{2} - \frac{1}{2^2}\left(\frac{1}{3} - 1\right) + \frac{1}{2^3}\left(\frac{1}{5} - \frac{2}{3} + 1\right) \\ & \quad - \frac{1}{2^4}\left(\frac{1}{7} - \frac{3}{5} + \frac{3}{3} - 1\right) + - \cdots \\ &= \frac{1}{2}\left(1 + \frac{1}{3} + \frac{1 \cdot 2}{3 \cdot 5} + \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \cdots\right) \end{aligned} \quad (8)$$

となる。

べき級数が奇数べきのみ

$$\sum_{i=1}^{\infty} (-1)^{i-1} a_i x^{2i-1} = a_1 x - a_2 x^3 + a_3 x^5 - + \cdots \quad (9)$$

のときは、$t = x^2$ のべき級数と考え一般オイラー変換を適用すると

$$\frac{x}{1+x^2} \sum_{i=1}^{\infty} (-1)^{i-1} \left(\frac{x^2}{1+x^2}\right)^{i-1} \Delta^{i-1} a_1 \quad (10)$$

となる。

グレゴリ級数に一般オイラー変換 (10) を適用して得られる級数は、

$$\begin{aligned} & \tan^{-1} x \\ = \; & \frac{x}{1+x^2}\left(1 + \frac{2}{3}\frac{x^2}{1+x^2} + \frac{2 \cdot 4}{3 \cdot 5}\left(\frac{x^2}{1+x^2}\right)^2 \right. \\ & \left. + \frac{2 \cdot 4 \cdot 6}{3 \cdot 5 \cdot 7}\left(\frac{x^2}{1+x^2}\right)^3 + \cdots \right) \end{aligned} \quad (11)$$

である。ライプニッツ級数にオイラー変換を適用した (8) は (11) において $x = 1$ とおいた級数である。

数列 $\{a_\nu\}$ が正で単調減少であるとは $a_1 > a_2 > \cdots > 0$ のときであるが、差分演算子を用いると

$$(-1)^m \Delta^m a_\nu > 0, m = 0, 1, \ \forall \nu \in \mathbb{N} \quad (12)$$

と表せる。(12) がすべての非負整数 m について成り立つとき、数列 $\{a_\nu\}$ は**完全単調**であるという。交代級数 (7) の各項の絶対値のなす数列 $\{a_\nu\}$ が完全単調で 0 に収束するとき、交代級数はオイラー変換により加速される。

定理 1 (クノップ [4, p.263])
交代級数 (7) が、

$$(-1)^m \Delta^m a_\nu > 0, \quad m = 0, 1, 2, \ldots \forall \nu \in \mathbb{N}$$
$$\lim_{\nu \to \infty} a_\nu = 0$$

を満たすとき、(6) は公比 1/2 の等比級数と同じかそれより速く収束する。

証明 [5, pp.53-54] を見よ。□

$a_1 - a_2 + \cdots + (-1)^{\mu-1}a_\mu$ は通常の方法で加え、$(-1)^\mu(a_{\mu+1} - a_{\mu+2} + \cdots + (-1)^{\nu-\mu-1}a_\nu)$ にオイラー変換を適用する変換を**遅延オイラー変換**という。

$$D_{\nu,\mu} = \sum_{i=1}^{\mu}(-1)^{i-1}a_i + (-1)^\mu \sum_{j=1}^{\nu-\mu} \frac{(-1)^{j-1}\Delta^{j-1}a_{\mu+1}}{2^j} \tag{13}$$

定義から $D_{\nu,0}$ はオイラー変換である。

オイラー変換は、A. ファン・ワインハールデンによって「エレガントで巧妙な」[7] アルゴリズムが与えられ、ALGOL 60(Pascal や C 言語に大きな影響を与えたプログラミング言語) の定義をまとめた報告書 [1] に、手続き宣言の例として収録された。C 言語に書き直したプログラムを [7] で見ることができる。ワインハールデンのアイデアは次の定理である。

定理 2 $\nu = 1, 2, \ldots$ に対し $T_k^{(\nu)}$ を

$$\begin{aligned} T_0^{(\nu)} &= s_\nu \\ T_k^{(\nu-k)} &= \frac{1}{2}(T_{k-1}^{(\nu-k)} + T_{k-1}^{(\nu-k+1)}) \\ & \quad k = 1, 2, \ldots, \nu \end{aligned} \tag{14}$$

により定義する。このとき、

$$T_k^{(\nu)} = D_{\nu+k,\nu}$$

である。特に、$T_\nu^{(0)}$ はオイラー変換である。

証明 k に関する数学的帰納法による。$k = 1$ のときは、

$$\begin{aligned} T_1^{(\nu)} &= \frac{1}{2}(s_\nu + s_{\nu+1}) \\ &= s_\nu + \frac{1}{2}(-1)^\nu a_{\nu+1} = D_{\nu+1,\nu} \end{aligned}$$

$k > 1$ のとき

$$T_k^{(\nu)} = D_{\nu+k,\nu}$$

の成立を仮定する。

$$\begin{aligned} & T_{k+1}^{(\nu)} = \frac{1}{2}\left(T_k^{(\nu)} + T_k^{(\nu+1)}\right) \\ = \; & s_\nu + (-1)^\nu \left(\sum_{j=1}^{k} \frac{(-1)^{j-1}}{2^{j+1}}\Delta^{j-1}a_{\nu+1} \right. \\ & \left. + \frac{1}{2}a_{\nu+1} - \sum_{j=1}^{k}\frac{(-1)^{j-1}}{2^{j+1}}\Delta^{j-1}a_{\nu+2} \right) \\ = \; & s_\nu + (-1)^\nu \left(\frac{1}{2}a_{\nu+1} + \sum_{j=1}^{k}\frac{(-1)^j}{2^{j+1}}\Delta^j a_{\nu+1} \right) \\ = \; & D_{\nu+k+1,\nu} \end{aligned}$$

となり $k + 1$ のとき成立する。□

定理 2 の (14) は

$$T_k^{(\nu-k)} = T_{k-1}^{(\nu-k+1)} - \frac{1}{2}(T_{k-1}^{(\nu-k+1)} - T_{k-1}^{(\nu-k)})$$

と書き直せるので、遅延オイラー変換はリチャードソン補外の特別な場合である。

オイラー変換と遅延オイラー変換をライプニッツ級数の 4 倍 $s_\nu = 4\sum_{i=1}^{\nu}(-1)^{i-1}/(2i-1)$ に適用した結果は表 2 である。$\nu \leq 19$ のときは $D_{\nu,5}$ のほうが $D_{\nu,10}$ よりよいが、$\nu = 20$ で追いつく。最適な μ は求めたい精度により変わってくる。

表 2: ライプニッツ級数にオイラー変換を適用

	オイラー変換	遅延オイラー変換	
ν	$D_{\nu,0}$	$D_{\nu,5}$	$D_{\nu,10}$
6	3.*0590007215*	3.1*578643579*	*2.9760461760*
7	3.1*009067322*	3.14*38783439*	3.*2837384837*
8	3.1*215045371*	3.14*20135420*	3.*0170718171*
9	3.1*316571806*	3.141*6844593*	3.*2523659347*
10	3.1*366719197*	3.141*6151788*	3.*0418396189*
11	3.13*91528966*	3.1415*986834*	3.1*370777142*
12	3.14*03819100*	3.14159*43803*	3.141*2185009*
		中略	
18	3.1415*743468*	3.14159265*58*	3.1415926*452*
19	3.1415*835376*	3.14159265*44*	3.14159265*16*
20	3.1415*881130*	3.141592653*9*	3.141592653*1*

5 e 変換

数列 $\{s_\nu\}$ は

$$s_\nu = s + c\lambda^\nu, \quad 0 < |\lambda| < 1 \tag{15}$$

を満たすものとする。ここで、s は未知の極限値、$c(\neq 0)$ は未知の定数である。このとき、λ が既知ならリチャードソン補外、未知ならエイトケン Δ^2 法により正確な極限値 s が得られることを前回話した。(15) が等式ではなく漸近式 $s_\nu = s + c\lambda^\nu + o(\lambda^\nu)$ の場合は、リチャードソン補外および Δ^2 法により加速できる。

(15) で $|\lambda| > 1$ のとき $\{s_\nu\}$ は発散するが、リチャードソン補外あるいはエイトケン Δ^2 法により正確な s が得られる。このような s は**反極限**といわれる。

(15) を一般化し、数列 $\{s_\nu\}$ が任意の $\nu \in \mathbb{N}$ に対し、

$$s_\nu = s + c_1\lambda_1^\nu + \cdots + c_k\lambda_k^\nu, \quad \lambda_j \neq 1 \tag{16}$$

を満たしている場合を考える。ここで、$s, c_1, \ldots, c_k$ は未知の定数である。$\lambda_1, \ldots, \lambda_k$ がすべて既知の場合は、リチャードソン補外で正確な極限値を得ることができる。では、これらが未知の場合はどうだろうか。$1 > |\lambda_1| > \cdots > |\lambda_k| > 0$ のときは、前回の定理 1 からエイトケン Δ^2 法により加速できる。しかしながら、エイトケン Δ^2 法を繰り返し適用したとしても、正確な極限値は得られない。D. シャンクス [9] は (16) に対し正確な極限値を与える e 変換を提案した。

(16) を $\nu, \nu+1, \ldots, \nu+k$ について連立させると、

$$\left\{\begin{array}{lllll} s_\nu & = s & +c_1\lambda_1^\nu & \cdots & +c_k\lambda_k^\nu \\ s_{\nu+1} & = s & +c_1\lambda_1^{\nu+1} & \cdots & +c_k\lambda_k^{\nu+1} \\ & & & \cdots & \\ s_{\nu+k} & = s & +c_1\lambda_1^{\nu+k} & \cdots & +c_k\lambda_k^{\nu+2k} \end{array}\right. \tag{17}$$

となる。いま、$\lambda_1, \ldots, \lambda_k$ を零点に持つ k 次多項式を

$$\phi(t) = a_0 + a_1 t + \cdots + a_k t^k$$

とおく。$\lambda_1, \ldots, \lambda_k$ は未知なので、係数 $a_0, \ldots, a_k$ は未知である。

(17) の最初の式に a_0 を掛け、2 番目の式に a_1 を掛け、...、$k+1$ 番目の式に a_k を掛け加えると

$$\begin{aligned} & a_0 s_\nu + a_1 s_{\nu+1} + \cdots + a_k s_{\nu+k} \\ = & \left(\sum_{i=0}^k a_i\right) s + c_1\lambda_1^\nu \phi(\lambda_1) \cdots + c_k\lambda_k^\nu \phi(\lambda_k) \\ = & \left(\sum_{i=0}^k a_i\right) s \end{aligned} \tag{18}$$

が得られる。ここで、$\sum_{i=0}^k a_i = \phi(1) \neq 0$ であるので、

$$s = \frac{a_0 s_\nu + a_1 s_{\nu+1} + \cdots + a_k s_{\nu+k}}{a_0 + a_1 + \cdots + a_k} \tag{19}$$

である。(19) は、s が連続する $k+1$ 項 $s_\nu, \ldots, s_{\nu+k}$ の重み付き平均となることを示している。

(18) より、

$$a_0(s_\nu - s) + \cdots + a_k(s_{\nu+k} - s) = 0$$

が任意の ν について成立するので、

$$\left\{\begin{array}{llll} a_0(s_\nu - s)+ & \cdots & +a_k(s_{\nu+k} - s) & = 0 \\ & \cdots & & \\ a_0(s_{\nu+k} - s)+ & \cdots & +a_k(s_{\nu+2k} - s) & = 0 \end{array}\right. \tag{20}$$

が成立する。(20) を $a_0, \ldots, a_k$ に関する連立 1 次方程式と考えると、s は

$$\begin{vmatrix} s_\nu - s & s_{\nu+1} - s & \cdots & s_{\nu+k} - s \\ s_{\nu+1} - s & s_{\nu+2} - s & \cdots & s_{\nu+k+1} - s \\ & & \cdots & \\ s_{\nu+k} - s & s_{\nu+k+1} - s & \cdots & s_{\nu+2k} - s \end{vmatrix} = 0$$

を満たす。$j = k, \ldots, 1$ に対し、$j+1$ 行から j 行を減じ、第 1 行に関し和に分けると

$$s\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix} = \begin{vmatrix} s_\nu & s_{\nu+1} & \cdots & s_{\nu+k} \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix}$$

がいえるので、s は 2 つの行列式の比で表すことができる。

$$s = \frac{\begin{vmatrix} s_\nu & s_{\nu+1} & \cdots & s_{\nu+k} \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix}} \tag{21}$$

数列 $\{s_\nu\}$ に対し (21) の右辺の数列

$$e_k(s_\nu) = \frac{\begin{vmatrix} s_\nu & s_{\nu+1} & \cdots & s_{\nu+k} \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta s_\nu & \Delta s_{\nu+1} & \cdots & \Delta s_{\nu+k} \\ & & \cdots & \\ \Delta s_{\nu+k-1} & \Delta s_{\nu+k} & \cdots & \Delta s_{\nu+2k-1} \end{vmatrix}}$$

を対応させる変換を**シャンクス e 変換**あるいはシャンクス変換という。便宜上 $e_0(s_\nu) = s_\nu$ とおく。$e_k(s_\nu)$ と同値な式は、R.J. シュミット [10] が連立 1 次方程式をガウス・ザイデル法で解く過程で提案したが、数列の加速として扱ったのは D. シャンクス [9] であるため、彼の名前がついている。

e_1 は

$$\begin{aligned} e_1(s_\nu) &= \frac{s_\nu \Delta s_{\nu+1} - s_{\nu+1}\Delta s_\nu}{\Delta s_{\nu+1} - \Delta s_\nu} \\ &= s_\nu - \frac{(\Delta s_\nu)^2}{\Delta^2 s_\nu} \end{aligned} \qquad (22)$$

より、エイトケン Δ^2 法である。

$k = 1, 2, 3$ のとき e_k は容易に計算できるが、k が大がきくなると行列式の計算は手間がかかる。そのため、シュミットの結果は発表から 15 年間注目されなかったのかもしれない。

6 ϵ 算法

P. ウィン [11] は、シャンクス変換 $e_k(s_\nu)$ を再帰的に計算する **ϵ 算法**を提案した。

定理 3 $\nu = 1, 2, \cdots$ に対し、

$$\begin{aligned} &\epsilon_{-1}^{(\nu)} = 0, \\ &\epsilon_{2k}^{(\nu)} = e_k(s_\nu),\ k = 0, 1, \ldots \\ &\epsilon_{2k+1}^{(\nu)} = \frac{1}{e_k(\Delta s_\nu)},\ k = 0, 1, \ldots \end{aligned}$$

とおいたとき、漸化式

$$\epsilon_{l+1}^{(\nu)} = \epsilon_{l-1}^{(\nu+1)} + \frac{1}{\epsilon_l^{(\nu+1)} - \epsilon_l^{(\nu)}}, \quad l = 0, 1, \ldots, \quad (23)$$

が成立する。ただし、分母は 0 にならないものとする。

証明 行列式に関する準備が必要なので次回に行う。
□

数列 $\{s_\nu\}$ に ϵ 算法を適用したとき、$\epsilon_l^{(\nu)}$ は、以下の表の左と上から順に計算する。

$$\begin{array}{cccccccc}
 & s_1 = \epsilon_0^{(1)} & & & & & & \\
 & & \searrow & & & & & \\
\epsilon_{-1}^{(2)} = 0 & \longrightarrow & & \epsilon_1^{(1)} & & & & \\
 & & \nearrow & & \searrow & & & \\
 & s_2 = \epsilon_0^{(2)} & & \longrightarrow & & \epsilon_2^{(1)} & & \\
 & & \searrow & & \nearrow & & \searrow & \\
\epsilon_{-1}^{(3)} = 0 & \longrightarrow & & \epsilon_1^{(2)} & & \longrightarrow & & \epsilon_3^{(1)} \\
 & & \nearrow & & \searrow & & \nearrow & \\
 & s_3 = \epsilon_0^{(3)} & & \longrightarrow & & \epsilon_2^{(2)} & & \\
 & & \searrow & & \nearrow & & & \\
\epsilon_{-1}^{(4)} = 0 & \longrightarrow & & \epsilon_1^{(3)} & & & & \\
 & & \nearrow & & & & & \\
 & s_4 = \epsilon_0^{(4)} & & & & & &
\end{array}$$

図式化すると次のようになる。

$$\begin{array}{ccc} & N & \\ W & & E \\ & S & \end{array}$$

とおいたとき、

$$E = W + \frac{1}{S - N}$$

このような算法を**菱形算法**という。ϵ 算法以後、多くの菱形算法が研究された。

菱形算法など 2 次元以上の漸化式で表されるアルゴリズムを理解する最良の方法は、具体例を手で計算してみることである。おおよそ理解できたら何らかのプログラミング言語でプログラムを作る。その際、手計算の結果を参照してデバッギングを行えば良い。C 言語による ϵ 算法のプログラムを

`http://www.cis.twcu.ac.jp/~osada/rikei/`

に載せておく。

例 1 ライプニッツ級数の 4 倍に ϵ 算法を適用する。s_ν は有理数で与えられるので、有理数のまま計算する。

$$\begin{aligned} &s_1 = 4,\ s_2 = \frac{8}{3},\ s_3 = \frac{52}{15},\ s_4 = \frac{304}{105},\ s_5 = \frac{1052}{315} \\ &\epsilon_1^{(1)} = -\frac{3}{4},\ \epsilon_1^{(2)} = \frac{5}{4},\ \epsilon_1^{(3)} = -\frac{7}{4},\ \epsilon_1^{(4)} = \frac{9}{4}, \\ &\epsilon_2^{(1)} = \epsilon_0^{(2)} + \frac{1}{\epsilon_1^{(2)} - \epsilon_1^{(1)}} = \frac{8}{3} + \frac{1}{\dfrac{5}{4} + \dfrac{3}{4}} = \frac{19}{6} \end{aligned}$$

$$\epsilon_2^{(2)} = \frac{52}{15} + \cfrac{1}{-\cfrac{7}{4} - \cfrac{5}{4}} = \frac{47}{15},\ \epsilon_2^{(3)} = \frac{1321}{420}$$

$$\epsilon_3^{(1)} = \frac{5}{4} + \cfrac{1}{\cfrac{47}{15} - \cfrac{19}{6}} = -\frac{115}{4},\ \epsilon_3^{(2)} = \frac{329}{4}$$

$$\epsilon_4^{(1)} = \frac{47}{15} + \cfrac{1}{\cfrac{329}{4} + \cfrac{115}{4}} = \frac{1744}{555}$$

6.1 ϵ 算法の漸近的性質

ウィン [12] は特別な漸近展開を持つ数列に ϵ 算法を適用した際の $\epsilon_{2k}^{(n)}$ の漸近評価を与えた。

定理 4 数列 $\{s_\nu\}$ は $\nu \to \infty$ のとき

$$s_\nu \sim s + \sum_{j=1}^{\infty} c_j \lambda_j^\nu, \quad 1 > |\lambda_1| > |\lambda_2| > \ldots > 0$$

と漸近展開されるものとする。ここで、$c_1, c_2, \ldots, \lambda_1, \lambda_2, \ldots$ は未知の定数で、$1 > |\lambda_1| > |\lambda_2| > \ldots > 0$ である。数列 $\{s_\nu\}$ に ϵ 算法を適用すると、k を固定し、$\nu \to \infty$ としたとき、

$$\epsilon_{2k}^{(\nu)} \sim s + \frac{c_{k+1}(\lambda_{k+1} - \lambda_1)^2 \ldots (\lambda_{k+1} - \lambda_k)^2 \lambda_{k+1}^\nu}{(1 - \lambda_1)^2 \ldots (1 - \lambda_k)^2}$$

証明 略 □

定理 4 の $k = 1$ の場合は、前回の定理 1 である。

定理 5 ϵ 算法を

$$s_\nu \sim s + (-1)^{\nu-1} \sum_{j=1}^{\infty} c_j (\nu + b)^{-j}, \quad c_1 \neq 0$$

を満たす数列 $\{s_\nu\}$ に適用する。k を固定し、$\nu \to \infty$ とすると、

$$\epsilon_{2k}^{(\nu)} \sim s + \frac{(-1)^{\nu-1}(k!)^2 c_1}{4^k (\nu + b)^{2k+1}}$$

証明 略 □

ライプニッツ級数 (2) は補題 1 より定理 5 の条件を満たしているので ϵ 算法で加速できる。ライプニッツ級数の 4 倍に ϵ 算法を適用した結果は表 3 のようになる。$\nu = 1, 2, 3, 4, 5$ については、例 1 と一致している。k を固定すると $\epsilon_{2k}^{(\nu-2k)}$ (縦の並び) は円周率に収束していることが分かる。表には含めてないが、$\nu = 20, k = 8$ のとき $\epsilon_{16}^{(4)} = 3.141592653589789$ の誤差は -4.0×10^{-15} である。

表 3: ライプニッツ級数に ϵ 算法を適用

ν	$\epsilon_0^{(\nu)}$	$\epsilon_2^{(\nu-2)}$	$\epsilon_4^{(\nu-4)}$
1	4.0000000000		
2	2.6666666667		
3	3.4666666667	3.1666666667	
4	2.8952380952	3.1333333333	
5	3.3396825397	3.1452380952	3.1423423423
10	3.0418396189	3.1412548236	3.1415854357
20	3.0916238067	3.1415563303	3.1415925228
ν	$\epsilon_6^{(\nu-6)}$	$\epsilon_8^{(\nu-8)}$	$\epsilon_{10}^{(\nu-10)}$
10	3.1415920729	3.1415925053	
20	3.1415926523	3.1415926536	3.1415926536

K. クノップの、「ライプニッツ級数は美しい」が、「数値目的には役に立たない」[4, p.214,252] を D. シャンクスは引用し [9, p.5]、e 変換を用いれば 10 項までで 8 桁円周率が得られることを強調している。($\epsilon_8^{(2)}$ は円周率と 7.3 桁一致している。)

その後、1961 年にシャンクスは、W. レンチと共同で円周率を 10 万桁計算した。彼らはそれまで使われていたマーチンの公式

$$\pi = 16 \tan^{-1} \frac{1}{5} - 4 \tan^{-1} \frac{1}{239}$$

ではなく、C. ステルマーの公式

$$\pi = 24 \tan^{-1} \frac{1}{8} + 8 \tan^{-1} \frac{1}{57} + 4 \tan^{-1} \frac{1}{239}$$

に基づき、$\tan^{-1} 1/8$ などの値はグレゴリ級数を 2 項ずつまとめた

$$\tan^{-1} x = \sum_{i=0}^{\infty} \frac{(4i+3) - (4i+1)x^2}{16i^2 + 16i + 3} x^{4i+1}$$

で求め、8 時間 43 分で計算した [2, pp.326-329]。1 万桁 100 分というのがそれまでの記録であったので 20 倍の速度である。e 変換は必要なかった。

参考文献

[1] J.W. Backus et al., Report on the algorithmic language ALGOL 60, Numer. Math. 2(1960), 106-136. (報告書の改訂版は以下で見ることができる。)
http://www.masswerk.at/algol60/report.htm

[2] L. Berggren, J. Borwein and P. Borwein, Pi: a source book, Springer, 1997

[3] L. Euler, Institutiones calculi differentialis cum eius usu in analysi finitorum ac doctrina serierum(微分学教程), 1755
`http://www.math.dartmouth.edu/~euler/pages/E212.html`

[4] K. Knopp, Theory and Application of Infinite Series, Dover, 1990 (ドイツ語版の初版は 1921 年、英語版の初版は 1928 年)

[5] 森口繁一、計算数学夜話、日本評論社、1978

[6] 長田直樹、交代級数の漸近展開と加速法、情報処理学会論文誌、28(1987), 431–436

[7] H. Press et al., Numerical Recipes in C, 2nd ed., Cambridge University Press, 1992 (初版の日本語訳が丹慶勝市他訳で技術評論社から出版されている。)

[8] R. Roy, The discovery of the series Formula for π by Leibniz, Gregory and Nilakantha, Mathematics Magazine 63(1990), 291-306 ([2, pp.92-107] に収録)

[9] D. Shanks, Non-linear transformations of divergent and slowly convergent sequences, J. Math. and Phys. 34(1955), 1–42.

[10] R.J. Schmidt, On the numerical solution of linear simultaneous equations by an iterative method, Phil. Mag. 7, 32(1941), 369–383.

[11] P. Wynn, On a Device for computing the $e_m(S_n)$ transformations, MTAC 10(1956), 91–96.

[12] P. Wynn, On the convergence and stability of the epsilon algorithm, J. SIAM Numer. Anal(1966), 91–122.

[13] 楠葉隆徳・林隆夫・矢野道雄、インド数学研究、恒星社厚生閣、1997

(おさだ なおき/東京女子大学)