

振動的関数の積分法

慶大理工学部 高橋秀俊
(Hidetosi Takahasi)

解析関数の数値積分に関しては各種の有力な方法が開発されていて、ほとんど研究し尽された観があるが、そのいずれの方法もうまく行かないのが、収束の遅い振動的な関数の無限積分（またはそのような積分に帰着させられるような積分）である。その中で最も多くあらわれる問題は、 $f(x)$ を $x \rightarrow \infty$ でゆっくり単調に 0 に漸近する関数としたとき、

$$I = \int_a^{\infty} f(x) \sin(x+\alpha) dx \quad (1)$$

のような形をした積分であるが、またベッセル関数を含む無限積分のように (1) の形にならないものもある。しかし、多くの場合、被積分関数は（必要なら適当な変数変換をすることにより）漸近的には (1) の形になる

この種の積分には、複素積分を応用して積分路を実軸の外に動かすことにより、より収束の速い関数または単調な関数

の積分に変換することができ、そのような方法は積分値を閉じた形で得る目的には常用されているが、数値積分の場合には、被積分関数を複素平面上で計算するのは一般に面倒であり、値がわからない場合も多いので、そのような方法は実際的とは言えない。

そこでここでは、実は最も初等的と見える方法、すなわち (1) の積分を $\sin(x+\alpha)$ の半周期に等しい π の幅の区間に切り分けて、区間毎に積分し、それを加え合わせるという方法が、この種の問題の解決には最も有効であることを例示することにする。

すなわち、積分を、区間の積分 I_k の和として

$$I_k = \int_{a+k\pi}^{a+(k+1)\pi} f(x) \sin(x+\alpha) dx \quad (2)$$

$$I = \sum_{k=0}^{\infty} I_k \quad (3)$$

として計算するのである。ここで無限和 (3) はその項の符号が交互に正、負と交代する交項級数 (alternating series) となるのが普通であるが、その収束は一般に遅い。しかし交項級数の総和に関しては各種の加速法が知られており、それらを (3) に適用すれば、比較的少数の項から無限和のよい近似値を得ることが出来る。

区分点の選択 積分の区間の切り方として、(2)では単純に下限 a から始めて、機械的に π ずつに切って行った。しかし、もし適切な理由があれば、任意の大きさを一様にずらせてもかまわないことは明らかである。その場合、 $x+\alpha=0$ となる点が、区間の中央になるように選ぶのが有利であると思われる。そうすれば、被積分関数の正の部分と負の部分とが近似的に消し合い、しかも x の大きいところの方がよく消し合うので、 I_k の絶対値が小さくなり、しかも $x \rightarrow \infty$ での収束も速くなるからである。その場合の式は

$$I_k = \int_{(k+\frac{1}{2})\pi-\alpha}^{(k+\frac{3}{2})\pi-\alpha} f(x) \sin(x+\alpha) dx \quad (4)$$

となる。こうした場合、余った半端な区間についての積分を別に計算して加える必要があることは、言うまでもない。

区間積分の計算法 有限積分 (2) の計算には、よく知られた各種の方法 — Simpson 法、高次の Newton-Cotes の式、Gauss の式、Romberg の方法、~~等~~ 等、何を使ってもよからう。ただ、 k の小さいところと大きいところとでは被積分関数の“たちの良さ”が変わるから、Romberg 法、適応的 Simpson 法などのような、適応性をもつ方法の方が有利かもしれない。

◎ Euler 変換とその本質について.

つぎに (3) の総和の計算については, Euler 変換とか, Aitken の δ^2 算法がよく知られているが, われわれは前者を用いることにした. その際はからずも "Euler 変換とはそもそも何であるか" ということを考え直してみる機会をもったので, それについて若干述べたい.

まず, Euler 変換の教科書的な定義を記すと, 交項級数

$$S = \sum_{k=0}^{\infty} (-1)^k a_k \quad (5)$$

があったとき, a_k の前方逐次差分 $\Delta^r a_k$ を

$$\begin{aligned} \Delta^0 a_k &= a_k \\ \Delta^r a_k &= \Delta^{r-1} a_{k+1} - \Delta^{r-1} a_k \end{aligned}$$

と定義すると,

$$S = \sum_{r=0}^{\infty} (-1)^r 2^{-r-1} \Delta^r a_0 \quad (6)$$

ということである. そして $\{a_k\}$ が単調減少の数列である場合, 多くの場合 (6) の方がもとの級数 (5) より速く収束するので, (5) $\rightarrow$ (6) の変換が Euler 変換と呼ばれて, 総和の加速法として使われるわけである.

また多くの場合, (6) を使うよりは, はじめの数項はそのままとし, a_n から後の部分に Euler 変換を適用して

$$S = \sum_{k=0}^{n-1} (-1)^k a_k + \sum_{r=0}^{\infty} (-1)^{n+r} 2^{-(r+1)} \Delta^r a_n \quad (7)$$

とした方が有利である. これを遅延 Euler 変換と呼ぼう.

さて, 以下では, (5) が交項級数であることは一応忘れて,

$$S = \sum_{k=0}^{\infty} A_k \quad (A_k = (-1)^k a_k) \quad (8)$$

と書くことにする. すると a_k の逐次差分 $\Delta^r a_k$ は

$$(-1)^{k+r} 2^{-r} \Delta^r a_k = A_k^{(r)}$$

と書くと, 関係

$$A_k^{(r)} = \frac{1}{2} (A_k^{(r-1)} + A_{k+1}^{(r-1)}) \quad (9)$$

$$(A_k^{(0)} = A_k)$$

によってまることがわかる. ということは, 平均演算子 $\mathcal{M}$ を

$$\mathcal{M} A_k = \frac{1}{2} (A_k + A_{k+1})$$

によって定義するなら,

$$A_k^{(r)} = \mathcal{M}^r A_k \quad (10)$$

と書けるということである. したがって 遅延 Euler 変換 の式

(7) は, よりすっきりした形

$$S = \sum_{k=0}^{n-1} A_k + \frac{1}{2} \sum_{r=0}^{\infty} \mathcal{M}^r A_n \quad (11)$$

に書きかえられる.

ここで (11) を有限で打ち切った値として

$$S_n^{(m)} = \sum_{k=0}^{n-1} A_k + \frac{1}{2} \sum_{r=0}^{m-1} A_n^{(r)} \quad (12)$$

を定義する. すると $S_n^{(m)}$ に対し, 差分関係

$$S_n^{(m+1)} - S_n^{(m)} = \frac{1}{2} A_n^{(m)} \quad (13)$$

が成り立つことは明らかである, ところで, もう一つ, $S_n^{(m)}$ の n に関する差分関係

$$S_{n+1}^{(m)} - S_n^{(m)} = A_n^{(m)} \quad (14)$$

が成立することを示そう. それには m に対する数学的帰納法を用いる.

まず $m=0$ に対しては, (14) は

$$S_{n+1} - S_n = A_n$$

を意味するので, これは明らかである. つぎに, (14) が肩荷が $0, 1, \dots, m$ までは正しいとする. すると

$$\begin{aligned} S_{n+1}^{(m+1)} - S_n^{(m+1)} &= S_{n+1}^{(m)} - S_n^{(m)} + \frac{1}{2} A_{n+1}^{(m)} - \frac{1}{2} A_n^{(m)} \\ &= A_n^{(m)} + \frac{1}{2} A_{n+1}^{(m)} - \frac{1}{2} A_n^{(m)} \\ &= \frac{1}{2} (A_n^{(m)} + A_{n+1}^{(m)}) = A_n^{(m+1)} \end{aligned}$$

で, たしかに $m+1$ に対しても成り立つことが示された.

ここで更に, (13) と (14) から $A_n^{(m)}$ を消去すると, 関係

$$S_n^{(m+1)} = \frac{1}{2} (S_n^{(m)} + S_{n+1}^{(m)}) = \mathcal{M} S_n^{(m)} \quad (15)$$

が得られる. この関係は $A_n^{(m)}$ に対する関係 (9) と全く同じ形であり, したがって

$$S_n^{(m)} = \mathcal{M}^m S_n \quad (16)$$

と書いてもよい.

さて, $S_n^{(m)}$ に対する (16) 式は Euler 変換に対する新しい解釈を与える. すなわち, 級数の収束を (遅延) Euler

変換によって加速するということは、原級数の部分和 $S_1, S_2, \dots, S_n, \dots$ という、真値 S をはさんで交互に上下する波状の数列を、隣同志の平均をとるという平滑化の操作によって、速く S に収束させることにほかならない。

④ 適応型アルゴリズム

ところでわれわれの目的は、要求された精度で S を近似する $S_n^{(m)}$ をなるべく速く計算することである。ここで、 n, m の値、つまり $S_n^{(m)}$ のつくる配列上での、求める $S_n^{(m)}$ のあり場所があらかじめわかっているならば、これを計算するには、まず部分和を S_{n+m} まで計算し、それから (16) に従って、 $\frac{m(m+1)}{2}$ 回の平均操作をすればよい。そのとき、図 1 の点線で囲んだ領域が計算される。

実際には n, m はわかっていないから、配列の中から探し出さねばならない。そこでいわゆる適応型の算法が必要になる。すなわち、配列の上端 $S_1^{(1)}$ から出発し、配列上をできるだけ S に近づくような途をたどりながら進む。そのため、1 歩ごとに毎回進行方向として、 n を増す方向（下方）と m を増す方向（右下方）のいずれか、 $S_n^{(m)}$ の値が真値 S により近くなる方を選ぶことになる。

ここで実際には真値 S はわかっていない。しかも二つの値は通常は真値をはさむ形になっているので、真値がわから

なければ、いずれがより真値に近いかも決定できない。そこで、真値と比較するかわりに、隣同士の差の絶対値 $|S_n^{(m+1)} - S_{n+1}^{(m)}|$ と $|S_{n+1}^{(m)} - S_{n+2}^{(m-1)}|$ を比較することにした。そして、前者の方が小さければ右下(つまり m を増す方向)へ、後者の方が小さければ下方(つまり n を増す方向)へ進むことにする。

この方式は、つぎのように考えても妥当である。 $S_n^{(m)}$ を求める実際のプログラムは2重のループ構造になるが、その場合、内側のループでは p を一定にして r を1から m まで増しながら $S_{p-r}^{(r)}$ を計算し、外側のループで p を増して行く、という形をとるのが具合がいい。すると $S_{p-r}^{(r)}$ は真値 S の上と下の値を交互にとりながら振動的に変る。そして、その振幅ははじめ大きく、だんだん幅がせまくなって極小を通過し、再びひろがる。そして前述の判定は、その最小幅のところを求めたことにほかならない。そこで、その幅が許容誤差と同程度になったなら、その両側の値の平均値をとって、 S の最良近似値として採用するのが合理的であろう。

例題

つぎの表1は Rabinowitz が提出した積分の例題⁽⁴⁾を変数変換して得られる無限積分

$$I = \int_1^{\infty} \frac{\sin x}{x^2} dx \quad (17)$$

に以上の方法を適用したときの収束の様子を示したものである。表には p 項までの部分和を用いたときの最適平滑次数 m , (平滑前の) 部分和, 最良近似値 $(S_{p-m+1}^{(m-1)} + S_{p-m}^{(m)})/2$ 等を p の各値に対して示してある。これを見ると, q 項まで採れば, 小数点以下 6 桁の精度が得られることがわかる。なお, 各区間での積分には 8 点の Gauss-Lobatto 公式を用いた。

なお, 振動的関数の積分に関しては文献(2)が上と全く同じ方法で扱っていることを, 後になって知ったが, あまり知られていないので再録した。

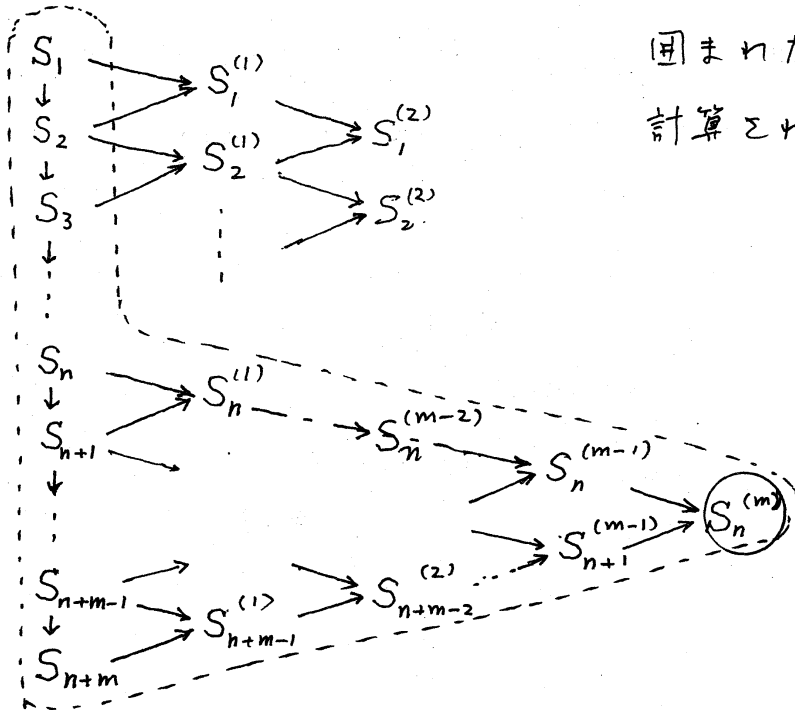
文 献

- 1) 戸田, 小野 日本物理学会誌 Vol.37, No 8 (1982)p. 655.
- 2) I.M. Longman Proc. Cambridge Phil. Soc. **52**, 764 (1956)

表 1. (17) 式の積分の最良近似値

p	m	A_{n-1}	S_p	$\frac{1}{2}(S_{p-m+1}^{(m-1)} + S_{p-m}^{(m)})$
1	1	.545449	.545449	
2	2	-.0543585	.491091	.50468
3	2	.0190652	.510156	.50539
4	3	-.0095725	.500583	.504183
5	3	5.72757E-03	.506311	.503928
6	4	-3.80369E-03	.502507	.504048
7	5	2.70692E-03	.505214	.504066
8	5	-2.02357E-03	.503191	.50407
9	6	1.56944E-03	.50476	.504068
10	7	-1.25249E-03	.503508	.504068
11	7	1.02259E-03	.50453	.504068

図 1. 累次平均の配列



$S_n^{(m)}$ の計算には点線で
囲まれた部分が最小限
計算される。

表 2. $S_n^{(m)}$ 配列の例. $S_n = \sum_{k=0}^{n-1} \frac{4}{2k+1} (-1)^k$

0.00000					
	2.00000				
4.00000		2.66667			
	3.33333		2.93333		
2.66667		3.20000		3.04762	
	3.06667		3.16191		3.09841
3.46667		3.12381		3.14921	3.12150
	3.18095		3.13651		3.14459
2.89524		3.14921		3.13997	3.14281
	3.11746		3.14343		3.14104
3.33968		3.13766		3.14210	3.14139
	3.15787		3.14077		3.14175
2.97605		3.14388		3.14139	3.14164
	3.12989		3.14201		3.14154
3.28374		3.14015		3.14169	3.14158
	3.15041		3.14136		3.14162
3.01707		3.14256		3.14155	3.14160
	3.13472		3.14174		3.14158
3.25237		3.14091		3.14162	3.14159
	3.14710		3.14150		3.14160
3.04184		3.14209		3.14158	
	3.13708		3.14166		
3.23232		3.14122			
	3.14536				
3.05840					

図 2. $S_{p-m}^{(m)}$ の変化 ($p=8$)

(表 2 の配列で, $m+n=8$ の線上をプロットしたもの)

