ハイパフォーマンス
コンピューティング 47－7
（1993．6．18）

自動微分法による無限級数の加速法

平山　弘

神奈川工科大学
機械システム工学科

ある種の漸近級数は、通常の関数の級数展開と同じような方法で、四則演算、代入、比較演算および関数計算等を定義することができる。この演算をＣ＋＋言語を利用して記述すると、通常の四則演算、関数、条件式などを含む関数を漸近級数に容易に展開することができる。これを利用すると、関数を任意の次数まで数値的に高精度展開可能である。この展開式を利用すれば、収束の遅い無限級数を数式処理を利用することなく加速することができる。この計算法の有効性を示すために、数値例を示した。

Acceleration Methods for Infinite Series by Automatic Differentiation

Hiroshi Hirayama

Department of Mechanical Systems Engng.,
Kanagawa Institute of Technology
1030 Shimo-ogino, Atsugi-si, Kanagawa-ken 243-02, Japan

The four rule operations and functions of a asymptotic series can be defined like ordinary power series. The functions which consist of four rule operations, pre-defined functions and conditional statements can be expanded in a asymptotic series easily by C++ programing language. Using this methods, The functions can be expanded in asymptotic series of arbitraly order on high precison. This asymptotic series gives effective acceleration methods of various slowly convergent series. This scheme is illustrated by numerial examples.

1．はじめに

自動微分法と呼ばれる方法[1][2]は、通常の数値計算プログラム上で、解析的手法を用いて、微分係数を求める方法である。この方法は主に多変数関数の低次の微分係数を計算するために用いられる方法である。この計算を行うには、プログラムをあるルールに従って変換する必要がある。この変換は大きいプログラムでは大変な作業となるため、自動的に変換するプリ・プロセッサが提供されることが望まれている。

C++言語[3]には、オペレター・オーバーロード(operator overload)機能と呼ばれる機能がある。この機能は、通常よく使われる演算子（+,-,*,/,=など）を、被演算型が異なる場合、別の意味を与えることができる機能である。この機能を使えば、分数や行列などの計算が、通常の数値のように扱うことができる。

自動微分法は、本質的に多項式の四則演算、関数演算であるから、多項式の四則演算、関数計算などを定義してあれば、C++のプログラムの宣言部分を変更するだけで、容易に微分係数を求めることができる。

本論文では、第一に、有限項で打ち切ったpoincare型の漸近級数間に四則演算、関数計算などの演算を、多変数の多項式と同じように定義することができることを示す。第二に、この定義に従って、任意の関数計算プログラムを計算すると、計算結果としてその関数の漸近級数を得ることができる[4]ことを示す。第三に、この漸近級数を長田[5][6]の無限級数の加速法に適用すると、数値微分や数式処理を含まないため、高速で高精度の数値的加速方法が得られることを示す。

これらの計算には、NEC社製PC-PC9801RA21を使用した。C++コンパイラーはBorland社製のBorland C++ 3.0を使用した。数値計算は拡張精度（10進で約19桁）で行なった。

2．漸近展開

この節では、関数を漸近展開するための基本的な考え方を説明し、その計算方法や計算公式について述べる。

2．1 漸近展開級数の四則演算

漸近級数の四則演算については、岡田[4]に述べられている。この公式は、Rall[1]で述べられているべき級数の四則演算とほぼ同じである。ここでは、(2.1)に示すようなPoincare型と呼ばれる漸近級数を扱う。この型は、長田[5][6][8]やSmith and Ford[7]らによってよく研究された型である。もし必要があれば、(2.1)の型の漸近展開をさらに一般化を扱うことも出来る。

二つの漸近展開級数f(x)、g(x)およびh(x)が、以下のように与えられているものとする。

$$f(x)=x^{p}\left(f_0+\frac{f_1}{x}+\frac{f_2}{x^2}+\dots\right) \qquad (2.1)$$

$$g(x)=x^{q}\left(g_0+\frac{g_1}{x}+\frac{g_2}{x^2}+\dots\right) \qquad (2.2)$$

$$h(x)=x^{r}\left(h_0+\frac{h_1}{x}+\frac{h_2}{x^2}+\dots\right) \qquad (2.3)$$

もし、

$$h(x)=f(x)\pm g(x) \qquad (2.4)$$

であるとき、f,gおよびhの係数には、(2.5)のような関係になる。

$$h_i=f_{r+i-p}\pm g_{r+i-q} \qquad (2.5)$$

計算は、添え字が負でない範囲で計算する。rはpとqの小さい方の値となる。f(x)とg(x)の指数部分の差が整数でない場合には、この演算は定義されない。

h(x)がf(x)とg(x)の積の場合、すなわち

$$h(x)=f(x)\,g(x) \qquad (2.6)$$

のとき、

$$r=p+q \qquad (2.7)$$

$$h_n = \sum_{k=0}^{n} f_k g_{n-k} \qquad (2.8)$$

となる。

h(x)がg(x)とf(x)の商で与えられるとき、すなわち、

$$h(x) = \frac{g(x)}{f(x)} \qquad (2.9)$$

であるとき、f(x)の係数とh(x)の係数には、以下のような関係が成り立つ。

$$r = q - p \qquad (2.10)$$

$$h_0 = \frac{g_0}{f_0} \qquad (2.11)$$

n≧1とき

$$h_n = \frac{1}{f_0}\left(g_n - \sum_{k=0}^{n-1} h_k f_{n-k}\right) \qquad (2.12)$$

である。

(2.1)で表される漸近級数の乗除算は、加減算と異なり、計算上の制限は少ない。

(2.12)の式は、(2.9)において、両辺にf(x)を掛け、(2.1)、(2.2)および(2.3)を代入し、展開する。両辺の同じ次数の係数が等しいことを利用して得られる。

2.2 漸近級数の関数

ここでは漸近展開の指数関数、対数関数、三角関数などの計算方法を示す。この方法は後に漸近級数の単純な例になっている。

(1) 指数関数 e^x

$h = e^f$ とおくと、

$$\frac{dh}{dx} = \frac{df}{dx} h \qquad (2.13)$$

である。(2.13)に(2.1)、(2.3)を代入して、両辺の係数を比較することによって、つぎのような関係が得られる。この式が成り立つためには、pは整数でなければならない。

$$r = 0 \qquad (2.14)$$

h_0はp=0のとき

$$h_0 = e^{f_0} \qquad (2.15a)$$

p≠0のとき

$$h_0 = 1 \qquad (2.15b)$$

となる。一般にh_nは　n≧1とき

$$h_n = \frac{1}{n} \sum_{k=1}^{n} k h_k f_{n-k} \qquad (2.16)$$

このように、漸近展開式の指数関数の計算には、指数関数の計算は、１回しか入らないので、漸近展開の指数関数の計算量は、通常の四則演算とそれほど大きな違いにはならない。

(2) 対数関数 $\log x$

$h = \log f$ とおくと、

$$\frac{dh}{dx} = \frac{1}{f} \frac{df}{dx} \qquad (2.17)$$

である。(2.16)の両辺にｆを掛けた式に(2.1)、(2.3)を代入して、両辺の係数を比較することによって、つぎのような関係が得られる。この式においても、以下の関係式が成り立つ為には、p=0でなければならない。

$$r = 0 \qquad (2.18)$$

$$h_0 = \log f_0 \qquad (2.19)$$

n≧1とき

$$h_n = \frac{1}{n f_0}\left(n f_n - \sum_{k=1}^{n} k f_k h_{n-k}\right) \qquad (2.20)$$

このように、漸近展開式の対数関数の計算には、指数関数の計算と同様に関数計算は、１回しか入らないので、四則演算とそれほど大きな違いにはならない。

(3) 三角関数 $\sin x$ と $\cos x$

三角関数は、次の三角関数のTaylor展開式に代入することによって計算を行う。

$$\sin x = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)!} \qquad (2.21)$$

$$\cos x = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!} \qquad (2.22)$$

これは、多項式のRall[1]の計算方法と異なる点である。これは、

$$\cos(\sqrt{x})$$

のような計算をできるようにしたためである。

(2.21), (2.22)にxをf(x)と見なして、(2.1)を代入して、左辺の漸近展開を計算する。

（4）べき乗f^a （aは定数）

$h = f^a$とおくと、

$$\frac{dh}{dx} = \frac{a}{f} h \frac{df}{dx} \qquad (2.23)$$

である。(2.23)の両辺にfを掛けてから、(2.1), (2.3)を代入して、両辺の係数を比較することによって、つぎのような関係が得られる。

$$r = ap \qquad (2.24)$$

$$h_0 = f_0{}^a \qquad (2.25)$$

n≧1とき

$$h_n = -\frac{1}{nf_0}\left(a\sum_{k=1}^{n} k f_k h_{n-k} - \sum_{k=1}^{n-1} k h_k f_{n-k}\right) \qquad (2.26)$$

a=0.5とすれば、平方根関数となる。平方根の関数は、この関数の特別な場合である。

2.4 漸近展開の数値例

（1）易しい例として、次のような関数を漸近展開する。

$$\frac{1+x^2+x^4}{x^2(1+x^2)} \qquad (2.27)$$

xの漸近展開は、(2.1)の式で考えると、p=1、f_0=1、それ以外の係数はゼロとなる。この式を、(2.27)の式に代入し、計算すると(2.27)式の漸近展開式が得られる。この式の計算とほぼ同じプログラムで漸近展開式が得られる。漸近展開式の型の名前をpoincareとすると、(2.27)の式は

```
poincare f227( poincare x )
{
    poincare t1, t2 ;
    t1 = x*x ;
    t2 = t1 * t1 ;
    return (1+t1+t2)/(t1*(1+t2)) ;
}
```

のプログラムで計算出来る。この関数を呼び出す関数でxはどのような漸近展開式であるかを指定する必要がある。この式を17項まで計算すると以下のようになる。

```
y = x^(-2)*(1+x^(-2)-1*x^(-6)
   +x^(-10)-1*x^(-14))                    (2.28)
```

ここで使われている記号'^'はべき乗を意味する。この式に現れていない項は係数がゼロの項である。このように漸近展開を簡単に計算することが出来る。

（2）次にやや複雑な式を漸近展開する。

$$\frac{(-1)^{x-1}}{x}\binom{-1/2}{x-1} = \frac{\Gamma(x-1/2)}{\sqrt{\pi}\, x\Gamma(x)} \qquad (2.29)$$

(2.29)の式を漸近展開するために、長田[3]は、特別の公式を利用している。ここでは、計算機の演算能力を利用して、よく見かけるガンマ関数の漸近展開式[9]を使うことができる。この公式は、

$$\log(\Gamma(x)) = \left(x-\frac{1}{2}\right)\log x - x$$

$$+\frac{1}{2}\log(2\pi) + \sum_{m=1}^{\infty} \frac{B_{2m}}{2m(2m-1)z^{2m-1}}$$

(2.30)

である。この中で使われているB_{2m}はBernoulli数である。この公式は、このまま計算すると、(2.1)の型の漸近級数にならないため、破綻をきたすことになる。このため、

$$\frac{\Gamma(x+b)}{\Gamma(x+a)} \quad (2.31)$$

を計算するプログラムを作成した。このプログラムの計算結果は、(2.1)で表されるpoincare型の漸近展開となる。この式を使って、第17項まで漸近展開を計算すると、(2.29)の式は(2.32)式のようになる。

```
y = x^(-1.5)*(0.56419+0.211571*x^(-1)
   +0.110193*x^(-2)+0.057851*x^(-3)
   +0.028564*x^(-4)+0.013423*x^(-5)
   +0.006829*x^(-6)+0.004076*x^(-7)
   +0.00195*x^(-8)+3.185393e-05*x^(-9)
   +0.000138*x^(-10)+0.002224*x^(-11)
   +0.000836*x^(-12)-0.006797*x^(-13)
   -0.002482*x^(-14)+0.032045*x^(-15)
   +0.011815*x^(-16))                    (2.32)
```

この計算では小数点以下6桁までの数値を表示しているが、計算精度は10進数で約19桁で行なっている。

このように、漸近展開を容易に計算することができる。この計算を通常の自動微分法と同様にプリ・プロセッサで変換して計算するためには、漸近展開専用のプリ・プロセッサ準備するか、漸近展開にも使えるように、さらに一般的なプリ・プロセッサに作り直さなければならないと思われる。

この計算は、以下の級数の加速の数値例でも使われている。

3．漸近展開を利用した無限級数の加速

この節では、長田[5][6]によって導かれた公式を利用して、交代級数および対数収束級数の和を計算する方法を示す。長田でも取り上げられているSmith and Ford[7]のテスト問題の計算例を示す。

3.1 無限級数の加速公式

（1）交代級数の加速公式

長田[5]によると、交代級数の一般項a_nが

$$a_n = (-1)^{n-1} f(n) \quad (3.1)$$

と表されるとする。また、f(n)が(2.1)のように展開出来るものとする。すなわち、

$$f(n) = n^p \left(a_0 + \frac{a_1}{n} + \frac{a_2}{n^2} + \ldots\right) \quad (3.2)$$

と漸近展開できるものとする。このとき、数列c_jを次のように定義する。

$$c_j = \frac{1}{2}a_j + \sum_{k=1}^{\infty} \frac{B_{2k}(2^{2k}-1)}{(2k)!} a_{j+1-2k}(p-j-1-2k) \ldots (p-j+1)$$

(3.3)

このc_jを使うと、(3.1)の交代級数の和Sは次のように表される。

$$S = S_n + (-1)^n n^p \sum_{j=0}^{\infty} \frac{c_j}{n^j} \quad (3.4)$$

ここで、S_nは(3.1)の級数の第n項までの和である。この公式を使うと、収束の遅い交代級数を効率的に計算できる。

（2）対数収束級数の加速公式

長田[6]によると、対数収束級数の一般項a_nが

$$a_n = f(n) \quad (3.5)$$

と表されれ、f(n)が(3.2)のように展開できるものとする。このとき、数列c_jを次の定義する。

$$c_0 = \frac{a_0}{1+p}$$

$$c_1 = \frac{a_1}{p} + \frac{a_0}{2}$$

$$c_j = \frac{a_j}{p+1-j} + \frac{1}{2} a_{j-1} + \sum_{k=1}^{[j/2]} \frac{B_{2k}}{(2k)!}(p-j+2k)\ldots(p-j+2)\, a_{j-2k} \tag{3.6}$$

ここで、[j/2]はjが偶数のときはj/2、奇数のとき[(j-1)/2]を表す。

この公式を使うと、収束の遅い対数収束級数を効率的に計算できる。

3.2 級数の数値例

（1）交代級数

Smith and Ford[7]の問題を漸近展開を用いて計算を行なった。計算した式を表1に、その計算結果を付録の表2に示す。

最初のn項を定義通り計算し、漸近展開の第10項まで利用して計算したものである。

表1　交代級数のテスト問題

	$a_n = (-1)^{n-1} f(n)$
SF-1	$\frac{1}{n}$
SF-2	$\frac{1}{2n-1}$
SF-3	$\frac{1}{\sqrt{n}}$
SF-4	$\binom{-1/2}{n-1}\frac{(-1)^{n-1}}{n}$
SF-5	$\binom{-1/2}{n-1}^2$

この結果は、長田[5]の結果と全く同じである。異なる点は、数値微分を用いていないので、数値微分の精度の悪さが計算結果に現れない点である。

（2）対数収束級数

交代収束級数と同様にSmith and Ford[7]の問題を漸近展開を用いて計算を行なった。計算した式を表3に、その計算結果を付録の表4に示す。

最初のn項を定義通り計算し、漸近展開の第10項まで利用して計算したものである。

表3　交代級数のテスト問題

	$a_n = f(n)$
SF-6	$\frac{1}{n^2}$
SF-7	$\frac{1+n^2+n^4}{n^2(1+n^4)}$
SF-8	$\frac{2n-1}{n(n+1)(n+2)}$
SF-9	$\sin\frac{1}{n}\cos\frac{1}{\sqrt{n}}$
SF-10	$\frac{1}{n}+\log\frac{n-1}{n}$
SF-11	$\log\frac{n+1}{n}\log\frac{n+2}{n+1}$
SF-12	$(n+e^{1/n})^{-\sqrt{2}}$
SF-13	$\binom{-1/2}{n-1}\frac{(-1)^{n-1}}{4n-1}$

交代級数の結果と同じく、長田[6]の結果と全く同じである。長田は漸近展開を数式処理で行なっている点がこの計算法と異なる点である。

計算結果は、有効桁数の小数点以下第1、2位で異なるがあるが、ほぼ完全に一致していると判断できる。

この計算方法は、通常の加速法と異なり、交代級数の加速の方が、対数収束級数より計算精度が

悪い。これは、単に例が悪いのか、計算の本質であるのか興味ある問題である。

このような点から、漸近級数を利用した方法は、対数収束級数の加速に有力な方法であるといえる。

４．結論

Ｃ＋＋を使って、poincare型の漸近級数展開を行なうことができることを示した。この漸近展開を利用すると、交代級数や対数収束級数の加速を効率的の行なうことができる。この方法は、交代級数より、対数収束級数の加速を得意とする方法である。これは従来の加速方法とは反対の性質である。

参考文献

[1] Rall,L. B. : "Automatic Differentiation-Technique and Applications,Lecture Notes in Computer Science", Vol.120, Springer-Verlag, Berlin-Heidelberg-New York(1981)

[2] 伊理、久保田：高速自動微分、数理科学、No.285 pp.41-48　(1987)

[3] Ellis M. A. and Stroustrup B. : "The Annotated C++ Reference Manual", Addison-Wesley,(1990)

[4] 岡田良知：級数概論、岩波書店、第１３章、(1969)

[5] 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、Vol.28, No.5 pp.431-436(1987)

[6] 長田直樹：対数収束級数の漸近展開と加速法 情報処理学会論文誌、Vol.29, No.3 pp.259-261(1988)

[7] Smith, D. A. and Ford, W. F. : Acceleration of Linear and Logarithmic Convergence, SIAM J.Numer. Anal., Vol.16, No.2, pp.223-240(1979)

[8] Naoki Osada: A Convergence Acceleration Method for Some Logarithmically Convergent Sequences, SIAM J.Numer. Anal., Vol.27, No.1, pp.178-189(1990)

[9] Abramowitz M. and Stegun A. : Handbook of Mathematical Functions, Dover,(1972)

付録

表１ 交代級数の計算の有効桁数

n	SF-1	SF-2	SF-3	SF-4	SF-5
2	4.91	6.12	4.91	5.52	5.58
3	6.62	6.12	6.73	6.51	6.33
4	6.62	7.65	6.73	7.11	7.29
5	8.00	7.65	8.18	7.99	7.83
6	8.00	8.91	8.18	8.40	8.71
7	9.15	8.91	9.38	9.23	9.07
8	9.15	9.96	9.38	9.49	9.89
9	10.12	9.96	10.39	10.29	10.11
10	10.12	10.85	10.39	10.40	10.90

表２　対数収束級数の計算の有効桁数

n	SF-6	SF-7	SF-8	SF-9	SF-10	SF-11	SF-12	SF-13
2	6.48	4.33	2.79	5.34	6.08	3.51	4.03	5.83
3	6.48	5.53	3.48	6.47	8.41	4.30	4.84	7.80
4	8.63	7.62	4.18	8.06	8.41	5.09	5.68	8.19
5	8.63	7.62	4.88	8.62	10.39	5.86	7.39	10.06
6	10.49	8.34	5.57	10.22	10.39	6.63	7.70	10.22
7	10.49	9.33	6.27	10.49	12.13	7.39	8.73	11.94
8	12.13	11.21	6.97	12.05	12.13	8.15	9.70	12.01
9	12.13	11.21	7.67	12.14	13.69	8.90	10.40	13.57
10	13.61	12.25	8.37	13.68	13.69	9.64	11.47	13.60