Taylor 展開を利用した置換積分による無限区間 振動型関数の数値積分法

平山 弘^{1,a)} 小宮 聖司^{1,b)}

概要：Taylor 級数の四則演算および関数計算は、Fortran 90 や C++ 言語を使えば容易に定義できる。これを使えば、次のような無限区間の振動関数型積分

$$\int_0^{\infty} f(x)g(h(x))dx = \int_0^a f(x)g(h(x))dx + \int_a^{\infty} f(x)g(h(x))dx$$

(ここで $f(x)$ は穏やかな減少関数で、 $g(x)$ は $\sin x$ or $\cos x$ で、 $h(x)$ は単調増加関数である。) 右辺第 2 項の積分を漸近展開を置換積分および部分積分法によって容易に求めることができる。この展開式を評価することによって、これらの積分を効率的に計算できる。

Numerical integration of oscillatory functions over Infinite interval by Integration by substitution using Taylor series

HIROSHI HIRAYAMA^{1,a)} KOMIYA SEIJI^{1,b)}

Abstract: Arithmetic operations and functions of Taylor series can be defined easily by FORTRAN 90 and C++ program language. Using this, it is shown that the asymptotic expansion of the following integral for oscillatory functions over the infinite interval

$$\int_0^{\infty} f(x)g(h(x))dx = \int_0^a f(x)g(h(x))dx + \int_a^{\infty} f(x)g(h(x))dx$$

(where $f(x)$ is slowly decaying function, $g(x)$ is $\sin x$ or $\cos x$, $h(x)$ is monotonically increasing function.) The asymptotic expansion of the integral of the second term on the right side can be easily obtained by the substitution integration and the partial integration method. Evaluating this expansion gives an effective numerical integration method for this kind of integrals.

Keywords: Taylor series, integral for oscillatory functions over the infinite interval, High Precision number

1. はじめに

Taylor 級数の係数を浮動小数点数の配列で表し、有限項で打ち切った Taylor 級数を Taylor 級数と呼ぶことにする。オペレータ・オーバーロード機能を持つ C++ 言語、Fortran

等を使うとこれらの Taylor 級数間の四則演算、Taylor 級数の関数演算を容易に定義することができる。このプログラムを使うと、プログラムの形で与えられた任意の関数を Taylor 級数に展開することができる。常微分方程式の初期値問題の解も任意次数の Taylor 級数に展開 [4] することができる。1 変数関数の逆関数は常微分方程式で定義できるから、逆関数も任意次数の Taylor 級数に展開できる。Taylor 級数を使えば、部分積分法を使い被積分関数を変形して数値積分を計算する方法は、無限区間振動型数値積分 [4] の計算に有効であることが知られている。

¹ 神奈川工科大学創造工学部自動車システム開発工学科
Department of Vehicle System Engineering, Faculty of Creative Engineering, Kanagawa Institute of Technology, Shimo-Ogino 1030, Atsugi, Kanagawa, 243-0292, Japan

a) hirayama@kanagawa-it.ac.jp

b) kom@eng.kanagawa-it.ac.jp

$$I = \int_0^{\infty} f(x)g(x)dx \quad (1)$$

ここで、 $f(x)$ は減少関数で $f^{(n)}(x)$ は $O(x^{\alpha-n})$ ($\alpha < 0$)、 $g(x)$ は $\sin x$ または $\cos x$ とする。

本論文では、Taylor 展開を使えば置換積分を容易に行うことができ、多くの被積分関数を数値積分が容易な関数形に変換することができる。ここでは、Taylor 展開法を次のような無限区間振動型数値積分に適用し計算する方法を提案する。

$$I = \int_0^{\infty} f(x)g(h(x))dx \quad (2)$$

ここで、 $h(x)$ は単調増加とする。この式に置換積分を使うと、すでに知られた計算法を使って、これらの無限区間振動型数値積分が計算できる。

このような計算が可能な理由の一つに、多くの関数の Taylor 級数の係数は、数学的に厳密な計算法を使うため、定数項を除いて基本的な演算や関数計算では打ち切り誤差以外は入らないためである。定数項はほぼ丸め誤差程度の誤差で計算できるため、Taylor 級数の係数は、高い精度で計算できる。

二重指数型数値積分法などのように、これまでの数値積分における変数変換 (置換積分) は独立変数の変換に限られていたが、Taylor 展開法を使えば、解析的方法では不可能な置換積分型の変換も可能になるなど、かなり一般的な変換も行え、多くの数値積分に有効である。

本文で扱う Taylor 展開式は、厳密な Taylor 展開式の計算と比較すると丸め誤差等は生じるが、計算の融通性が増し、高速計算が可能になる。ここでの数値計算は、主に Visual Studio 2017 の C++ 言語を使用した。

2. Taylor 級数の計算

この節では、関数を Taylor 級数に展開する方法を簡単に説明する。詳しくは Rall[9] および 平山等 [3] を参照せよ。

展開位置が同じ Taylor 級数間の演算は、一般性を失うことなく、原点で展開された Taylor 級数であると仮定することができる。Taylor 級数の展開位置は、平行移動により任意の位置に移動できるため、原点で展開された Taylor 級数のみを考慮すれば十分である。 n 次の Taylor 級数を次のように定義する。

$$f(x) = f_0 + f_1x + f_2x^2 + f_3x^3 + f_4x^4 \cdots + f_nx^n \quad (3)$$

$$g(x) = g_0 + g_1x + g_2x^2 + g_3x^3 + g_4x^4 \cdots + g_nx^n \quad (4)$$

$$h(x) = h_0 + h_1x + h_2x^2 + h_3x^3 + h_4x^4 \cdots + h_nx^n \quad (5)$$

2.1 Taylor 級数の四則演算

n 次の Taylor 級数の四則演算プログラムは容易に作成できる。以下の公式は、原点で展開された級数だけでなく、任意の点で展開された Taylor 級数でも有効である。

(1) 加減算

もし $h(x) = f(x) \pm g(x)$ ならば f, g の係数から h の係数は $h_i = f_i \pm g_i$ ($i = 0, \dots, n$) と計算できる。

(2) 乗算

もし $h(x) = f(x)g(x)$ ならば f, g の係数から h の係数は $h_j = \sum_{k=0}^j f_k g_{j-k}$ ($j = 0, \dots, n$) と計算できる。

(3) 除算

もし $h(x) = \frac{f(x)}{g(x)}$ ならば f, g, h の係数の関係式は $g_0 \neq 0$ ならば、次のようになる。

$$h_0 = \frac{f_0}{g_0}, \quad h_j = \frac{1}{g_0} \left(f_j - \sum_{k=0}^{j-1} h_k g_{j-k} \right) \quad (j \geq 1)$$

もし $f_i = g_i = 0$ ($i = 0, \dots, j$) ならば、分子と分母を x^{j+1} で割り、その式を上記の計算式で除算を行うことができる。

2.2 Taylor 級数の数学関数計算

多くの基本関数は、単純な微分方程式を満たす。この微分方程式を利用することにより、Taylor 級数の関数を容易に計算できる。

(1) 指数関数

もし $h(x) = e^{f(x)}$ ならば、次の微分方程式を満たす。

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

(3) と (5) をこの微分方程式に代入し、同じ次数の係数を比較すると次の関係式が得られる。

$$h_0 = e^{f_0}, \quad h_j = \frac{1}{j} \sum_{k=1}^j k h_{j-k} f_k \quad (j \geq 1)$$

(2) べき乗関数

もし $h(x) = f(x)^\alpha$ (α は定数) ならば、次の微分方程式を満たす。

$$f(x) \frac{dh(x)}{dx} = \alpha \frac{df(x)}{dx} h(x)$$

(3) と (5) をこの微分方程式に代入し、同じ次数の係数を比較すると次の関係式が得られる。

$$h_0 = f_0^\alpha, \quad h_j = \frac{1}{j f_0} \sum_{k=1}^j (k(\alpha + 1) - j) f_k h_{j-k} \quad (j \geq 1)$$

$\alpha = \frac{1}{2}$ とすると平方根の計算になる。他の数学関数についても、同様な微分方程式が得られる。得られた微分方程式から、Taylor 級数の数学関数を計算する公式が得られる。

これまでの式から分かるように、 n 次の Taylor 級数が与えられれば、それらの Taylor 級数の加減乗算、指数対数関数、三角関数等が計算できる。その計算の結果である Taylor 級数の n 次までの係数は、丸め誤差は生じるが通常の数値計算程度の正確さで計算することができる。

2.3 Taylor 級数の計算例

関数を Taylor 級数展開する簡単なプログラム例を以下に示す。関数 $f(x)$ を $x = a$ で Taylor 展開するには、次の公式によって行う。

$$f(x) = f(a + (x - a)) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n + \dots \quad (6)$$

この式は関数 $f(x)$ の x に $a + (x - a)$ を代入すると、 $f(x)$ の Taylor 級数が得られることを意味する。 $(x - a)$ を t と置いて、 x に $a + t$ を代入して、 t の Taylor 級数を計算する。その後 t を $(x - a)$ と置き換えると (6) の Taylor 級数が得られる。

次の関数を $x = 0$ で Taylor 展開する。

$$f(x) = \frac{x}{e^x - 1} \quad (7)$$

C++言語用のプログラムは以下のようになる。

```
1 : #include "taylor_template.h"
2 : typedef taylor_template<double> taylor ;
3 : int main()
4 : {
5 :     taylor x, y ;
6 :     x = taylor( 0.0, 0.0, 1.0);
7 :     y = x/(exp(x)-1) ;
8 :     cout << y << endl ;
9 : }
```

ここで、コンストラクター $\text{taylor}(a, b, c)$ は $b + c(x - a)$ を定義する。この計算では、除算演算で分子と分母の Taylor 級数の定数項がゼロになるので、分子と分母を x で割ってから除算を行っている。(7) の $f(x)$ を $x = 0$ で Taylor 展開し 10 次まで表示すると以下ようになる。係数の絶対値が 10^{-10} 以下のものはゼロとして、消去すると以下のようになる。

$$1 - 0.5x + 0.0833333x^2 - 0.00138889x^4 + 3.30688e-05x^6 - 8.2672e-07x^8 + 2.08768e-08x^{10}$$

3. 逆関数の Taylor 級数展開

ここでは、積分の変数変換に必要な逆関数の Taylor 展開法について述べる。関数の逆関数の Taylor 展開法については昔からいろいろな方法が知られている。Taylor 級数に Taylor 級数を代入して $f(f^{-1}(x)) = x$ を満たすように Taylor 級数を決める方法や Lagrange Inversion Formula[1] を使う方法など多数知られている。ここでは常微分方程式の解として計算する。 $y = f(x)$ の逆関数は、 $x = f(y)$ と書けるので、両辺を x で微分することによって、次の逆関数の微分方程式を得る。

$$\frac{dy}{dx} = \frac{1}{f'(y)} \quad (8)$$

この微分方程式を Picard の逐次計算方法 [4] によって、関

数 $f(x)$ の逆関数の Taylor 展開式を計算することがきる。

例として関数 $f(x) = e^{-x} - 2x - 3$ の逆関数 $y = f^{-1}(x)$ を求める。(8) から、逆関数は、次の微分方程式を満たす。

$$\frac{dy}{dx} = -\frac{1}{e^{-y} + 2} \quad \text{初期条件: } y(-2) = 0 \quad (9)$$

初期条件は、 $x = f(y)$ に $y = 0$ を代入して決めている。この微分方程式の Taylor 級数解を 6 次まで計算すると、

$$y = -0.333333(x + 2) + 0.0185185(x + 2)^2 - 7.61389 \times 10^{-20}(x + 2)^3 - 0.000114312(x + 2)^4 + 5.08053 \times 10^{-6}(x + 2)^5 + 1.12901 \times 10^{-6}(x + 2)^6$$

が得られる。この展開式を利用すれば、 $f(x) = 0$ の解 (零点) を計算することができる。上の展開式に、 $x = 0$ を代入すると、 $y = -0.594203952757019$ となり、約 6 桁の精度で $f(x) = 0$ の解が得られる。このように逆関数の Taylor 展開式は容易に計算できる。プログラムでは、逆関数を計算する関数 $\text{invers}(f)$ を準備している。

$$f = f_0 + f_1(x - a) + f_2(x - a)^2 + f_3(x - a)^3 + \dots \quad (10)$$

としたとき、逆関数 $g = \text{invers}(f)$ は、次のような形で得られる。

$$g = a + g_1(x - f_0) + g_2(x - f_0)^2 + g_3(x - f_0)^3 + \dots \quad (11)$$

他の Taylor 級数の計算と異なり、この様に一般に Taylor 級数の展開位置が変わる。

4. 三角関数を含む無限区間の振動型数値積分

次のように振動関数として三角関数を含む積分を考える。

$$I = \int_0^\infty f(x) \sin x dx \quad (12)$$

このような無限区間の振動型数値積分の計算には、Longman[6] のように、三角関数の隣り合う零点区間で数値積分を行い、無限交代級数に変換し、Euler 変換などの補外法によって、積分値を求める方法、Ooura and Mori[7] による発見的な方法および平山 [4] の Taylor 展開法等が発表されている。ここでは Taylor 展開法を利用する方法について述べる。

積分区間を $x = a$ で二分割すると、次のように二つの積分の和の形になる。

$$I = \int_0^\infty f(x) \sin x dx = \int_0^a f(x) \sin x dx + \int_a^\infty f(x) \sin x dx \quad (13)$$

ここで、 $f(x)$ は減少関数で $f^{(n)}(x)$ は $O(x^{\alpha-n})$ ($\alpha < 0$) と仮定する。右辺の最初の積分の値は、通常の数値積分法を使って計算できる。第 2 項の積分は、部分積分法を M 回繰り返すと、次のような展開式が得られる。

$$\int_a^\infty f(x) \sin x dx = f(a) \cos a - f'(a) \sin a - f''(a) \cos a + f'''(a) \sin a + \dots + f^{(M-1)}(a) \sin(a + \frac{M\pi}{2}) + \int_a^\infty f^{(M)}(x) \sin(x + \frac{M\pi}{2}) dx \quad (14)$$

関数 $f(x)$ の導関数 $f^{(n)}(x)$ は、仮定により、 x が大きいとき、 $O(x^{\alpha-n})$ となるので、(14) の右辺の積分項の値は、 $O(a^{\alpha-M+1})$ となる。この積分項は、第 M 項まで計算したときの打ち切り誤差となるので、この漸近展開式の誤差は、 $O(a^{\alpha-M+1})$ となる。この積分は、 a を十分大きな値にすれば、いくらでも小さな値にすることができるので、この方法によって、(13) の積分を任意の精度で計算できる。全体の誤差は、(13) の右辺第 1 項の数値積分の誤差と (14) の打ち切り誤差の和となる。

この計算では関数 $f(x)$ の高階導関数の値が必要である。この値を計算するには、 $f(x)$ を Taylor 級数展開し、その係数を階乗倍すれば得られる。差分法による計算と違い、高階導関数の値も高精度で計算できる。式 (13)、(14) は、正弦関数 ($\sin x$) だけでなく余弦関数 ($\cos x$) に対しても同様な公式が成り立つので、 $I = \int_0^\infty f(x) \cos x dx$ の積分値も同様に計算できる。

5. 置換積分による数値積分

次のよう無限区間の振動型積分を考える。

$$I = \int_0^\infty f(x)g(h(x))dx \quad (15)$$

ここで、 $f(x)$ は減少関数で $f^{(n)}(x)$ は $O(x^{\alpha-n})$ ($\alpha < 0$)、 $g(x)$ は $\sin x$ または $\cos x$ 、 $h(x)$ は増加関数とする。

この積分を $x = a$ を境界にして、二つの部分に分ける。

$$I = \int_0^a f(x)g(h(x))dx + \int_a^\infty f(x)g(h(x))dx \quad (16)$$

(16) の右辺第 2 項の積分 I_2 で、 $t = h(x)$ と置くと、置換積分によって次の形になる。

$$I_2 = \int_b^\infty s(t)g(t)dt \quad (17)$$

ここで

$$b = h(a), \quad s(t) = f(h^{-1}(t)) \frac{d}{dx} h^{-1}(t) \quad (18)$$

(16) の右辺第 1 項 I_1 は、有限区間の通常の数値積分法によって計算することができる。(16) の右辺第 2 項 I_2 は、振動型関数 $g(t)$ が $\sin t$ や $\cos t$ の三角関数であるから (14) の公式から容易に計算できる。 $h(t)$ の逆関数やその微分が解析的に計算できるならば、これまでの計算方法で計算できる。

$h(t)$ の逆関数やその微分が解析的に求められない場合でも、これらの数値積分は Longman や Ooura and Mori[7] の方法で計算することは可能であるが、この場合、多くの

標本点について、 $h^{-1}(x)$ や $\frac{d}{dx} h(x)$ を計算する必要があるため、逆関数はニュートン法などを使って計算する必要があり、微分係数は精度よく計算するのは困難である。このため、かなりの計算時間を必要とし、効率的な計算とはならないように思われる。

6. 数値例

6.1 積分値が知られている積分

この計算法が有効であることを示すために、次の積分を計算する。

$$I_1 = \int_0^\infty \sin x^2 dx = \frac{1}{2} \sqrt{\frac{\pi}{2}} \quad (19)$$

この式は、 $x^2 = t$ と置くことによって、置換積分を行うと、次のようになる。

$$I_1 = \frac{1}{2} \int_0^\infty \frac{\sin t}{\sqrt{t}} dt = \frac{1}{2} \sqrt{\frac{\pi}{2}} \quad (20)$$

このように (12) の形式になる。4 節の計算法で計算できるが、ここでは (15) の型の積分として計算する。

$$I_1 = \int_0^a \sin x^2 dx + \int_a^\infty \sin x^2 dx \quad (21)$$

$f(x) = 1$ 、 $h(x) = x^2$ として計算する。 $a = 7$ として、 $h(x)$ を 20 次まで Taylor 展開する。倍精度で計算する場合、経験的に $h(a) \approx 50$ となるように a を選び、Taylor 級数は 20 次まで計算する。

$$h(x) = 49 + 14(x - 7) + (x - 7)^2$$

この関数の逆関数を計算する。4 次まで示すと次のようになる。

$$h^{-1}(x) = 7 + 0.0714286(x - 49) - 0.000364431(x - 49)^2 + 3.71869 \times 10^{-6}(x - 49)^3 - 4.74323 \times 10^{-8}(x - 49)^4$$

したがって、

$$s(x) = 0.0714286 - 0.000728863(x - 49) + 1.11561 \times 10^{-5}(x - 49)^2 - 1.89729 \times 10^{-7}(x - 49)^3 + 3.38802 \times 10^{-9}(x - 49)^4$$

(21) の右辺第 1 項の積分は、二重指数関数型数値積分法で計算すると標本点数 197 で 0.605886931627828 となった。右辺第 2 項の積分は、13 次までの項を使って、0.02077013702992306 と得られた。これを加えると、積分 I_1 は 0.6266570686577513 となる。厳密解 0.626657068657750125 との差が 1.22×10^{-15} であった。この方法によって、容易に計算できることがわかる。

6.2 SIAM の問題

SIAM の問題 [2] とは、次の積分を 100 桁の精度で求め

る問題である。

$$I = \int_1^{\infty} \frac{\cos(u \log u)}{u} du = \int_0^{\infty} \cos(xe^x) dx \quad (22)$$

上の右辺の2個の積分は、同じ積分である。右辺第1項の積分で $u = e^x$ と置くと右辺第2項の積分が得られる。

この問題を Gautschi[2] は Longman[6] の方法で計算している。この計算のためには、(22)の右辺第1項の積分を $x = u \log u$ とおいて、次のように変形する。

$$I = \int_0^{\infty} \frac{\cos x}{x + u(x)} dx \quad (23)$$

ここで、 $u(x)$ は x を与えた時、 $u \log u = x$ を満たす u を意味する。これを次のような無限級数に変換する。

$$I = \int_0^{\pi/2} \frac{\cos x}{x + u(x)} dx + \sum_{k=1}^{\infty} (-1)^k \int_{-\pi/2}^{\pi/2} \frac{\cos t dt}{t + k\pi + u(t + k\pi)} \quad (24)$$

$u(x)$ の計算をニュートン法等を使って、多数計算しなければならない。この論文では、関数 $u(x)$ の近似関数を求め、その関数値を初期値としてニュートン法で計算する方法を提案している。論文のかなりの部分は $u(x)$ の計算法に費やされており、積分の計算法と言うより、 $u(x)$ の計算法となっている。 $u(x)$ を計算し、各積分を計算する。これは交代級数であるから、Euler 法などで加速して計算する。

(23)の積分は、原理的には大浦の連続 Euler 変換 [8] で計算できるが、 $u(x)$ の計算は毎回、Newton 法を適用し、非線形方程式を解かなければならない。このためあまり現実的な計算法とは言えない。

ここでは、次のように (22) の右辺第2項を計算する。 $x = a$ で分割すると次の式になる。

$$I = \int_0^{\infty} \cos(xe^x) dx = \int_0^a \cos(xe^x) dx + \int_a^{\infty} \cos(xe^x) dx \quad (25)$$

ここで、上の式の第2項の積分を $xe^x = t$ と置換する。

(25)の右辺第2項の積分は、Taylor 展開を利用した方法で計算できる。このためには、(25)の $h(u) = ue^u$ を Taylor 展開し、その逆関数を計算する。逆関数の Taylor 展開は、3節で与えたように簡単にでき、その微分も容易に計算できる。(25)の右辺の第1項の積分は、式変形しないでそのまま通常の数値積分法を利用して計算する。ここでは、この積分を二重指数型数値積分法を使って計算を行った。 $a = 3$ として、 $h(u)$ を $u = a$ で Taylor 展開すると、6次まで表示すると以下のようになる。

$$h(u) = 60.2566 + 80.3421(u - 3) + 50.2138(u - 3)^2 + 20.0855(u - 3)^3 + 5.85828(u - 3)^4 + 1.33904(u - 3)^5 + 0.251069(u - 3)^6$$

これから、逆関数 $s(u) = h^{-1}(u)$ を求め、微分すると以下のようなになる。

$$s'(u) = 0.0124468 - 0.000193653(u - b) + 3.07319 \times 10^{-6}(u - b)^2 - 4.91892 \times 10^{-8}(u - b)^3 + 7.91224 \times 10^{-10}(u - b)^4 - 1.27691 \times 10^{-11}(u - b)^5 + 2.06563 \times 10^{-13}(u - b)^6$$

ここで、 $b = 3e^3 = 60.2566$ である。この結果を (14) の公式に代入して、(25)の右辺第2項の積分を求める。第1項は二重指数型数値積分法で計算した結果を利用した。どちらも要求計算精度を 1.0×10^{-12} とした。第1項の積分は標本点数が 395 で 0.3168555029084669 に収束した。第2項の積分値は 20 次の Taylor 展開式を使い、漸近展開式を計算した。漸近展開式は 11 項までの計算で要求精度を満たした。これらの積分の値を加えると 0.32336743167777826 となった。この結果は、以下の高精度計算と比較すると最後の2桁を除いた小数点以下15桁まで一致する。ほぼ倍精度計算の限界まで一致する。単調増加で関数であればこの方法が使えるので、かなり一般的な計算法と言える。

6.3 SIAM の問題の 100 桁精度の計算

この計算法が、高精度計算でも有効であることを示すために、この積分の本来の問題である 100 桁までの高精度計算を行う。計算には MPPACK[5] を改良した多倍長精度計算ライブラリーを使用し、 10^8 進数で 20 桁 (10 進数で 160 桁) で行った。ここでは、 $a = 6$ として、(25)の第1項を要求精度 1.0×10^{-120} として二重指数型数値積分公式を使って数値積分を行った。このときの標本点数は 21857 であった。この数値積分の時間は 5.403 秒であった。Taylor 展開式の次数は 60 次とした。Taylor 展開を行い逆関数を計算し、それを微分するには、3.157 秒の計算時間を要した。第2項の積分の漸近展開式は 58 次の項で要求精度を満たした。この方法で計算した積分値を小数点以下 100 桁まで表示すると以下のようになる。

```
0.3233674316 7777876139 9370087952 1704466510
4662572546 9661681036 4434317903 3721067289
4431930370 4641024513
```

確認のため、 $a = 5$ として計算した。結果は 110 桁まで完全に一致した。この計算値は SIAM[10] の計算結果とも一致している。このように 100 桁程度の数値も Taylor 展開を使った変数変換法で容易に求められる。

この時の計算時間は、Intel i7-8700K 3.7GHz で 8.63 秒であった。C++言語として Visual Studio 2017 の C++言語を使用した。

Gautschi は Sun Ultra 5 workstation (with 360 MHz UltraSPARC-III processor) を使用して 64 桁の計算に約 88 分を要した。本計算法は Gautschi 計算法の約 1000 倍以上

高速に計算することができた。コンピュータが違うので正確には比較できないが、本計算法はかなり効率的であると思われる。

6.4 複雑な問題の計算例

いろいろな計算例を以下に示す。計算は倍精度と4倍精度 [11][12] で計算した。計算結果の数値は、倍精度と4倍精度が一致した部分を示した。

(1) 指数関数・対数関数を含む例

$$\begin{aligned} I &= \int_0^{\infty} \cos(\log(1+x)e^x) dx \\ &= 0.357913519007124 \dots \\ I &= \int_0^{\infty} \sin(\log(1+x)e^x) dx \\ &= 0.57849700088881 \dots \end{aligned}$$

倍精度計算および4倍精度計算共に $a = 4$ として計算した。 $h(4) = 87.8723$ である。倍精度の計算では20次のTaylor級数を使って計算した。4倍精度の計算では30次のTaylor級数を使って計算した。

(2) 対数関数と減少関数を含む例

$$\begin{aligned} I &= \int_0^{\infty} \frac{x}{1+x^2} \cos(x \log(1+x)) dx \\ &= 0.213637933436 \dots \\ I &= \int_0^{\infty} \frac{x}{1+x^2} \sin(x \log(1+x)) dx \\ &= 0.44204438473179 \dots \end{aligned}$$

倍精度計算では $a = 15$ として計算した。 $h(15) = 41.5888$ である。20次のTaylor級数を使って計算した。4倍精度計算では $a = 30$ として計算した。 $h(30) = 103.0196$ である。30次のTaylor級数を使って計算した。

(3) 代数関数を含む例

$$\begin{aligned} I &= \int_0^{\infty} \sqrt{\frac{x^2+9x+20}{x+1}} \cos\left(\frac{x^4+2x^2+5}{x^2+4}\right) dx \\ &= -1.1043321171895 \dots \\ I &= \int_0^{\infty} \sqrt{\frac{x^2+9x+20}{x+1}} \sin\left(\frac{x^4+2x^2+5}{x^2+4}\right) dx \\ &= 5.0657504197927 \dots \end{aligned}$$

倍精度計算では $a = 7$ として計算した。 $h(7) = 47.2453$ である。20次のTaylor級数を使って計算した。4倍精度計算では $a = 10$ として計算した。 $h(7) = 98.125$ である。30次のTaylor級数を使って計算した。

7. 終わりに

Taylor 展開式を利用すれば、変数変換が解析的にできない積分問題でも、置換積分ができ、都合のよい形に変形できる場合がある。このように変形された積分は、部分積分法などが適用可能で数値的な扱いが困難な問題も容易に計

算することができる。特にここで扱った無限区間振動型積分では、計算した Taylor 級数を無駄なく利用する計算法で非常に効率的である。

本論文では、無限区間振動型関数を扱ったが、それに限らず変数変換が有効な問題は多数あると考えられる。それらに対して本手法は有効な計算法を提供するものと思われる

参考文献

- [1] Bruijn, N.D., Asymptotic Methods in Analysis, Dover, New York, 1981
- [2] Gautschi W., The numerical evaluation of a challenging integral, Numerical Algorithms, **49** (2008), 187–194.
- [3] 平山、小宮、佐藤, Taylor 級数法による常微分方程式の解法, 日本応用数学会論文誌, **12** (2002), 1–8.
- [4] 平山弘, 部分積分法による半無限区間振動型積分の数値計算法, 日本応用数学会論文誌, **7** (1997), 131–138.
- [5] 平山弘, C++ 言語による高精度計算パッケージの開発, 日本応用数学会論文誌, **5** (1995), 307–318
- [6] Longman, I.M., Note on a method for computing infinite integrals of oscillatory functions. Proc. Cambridge Philos. Soc. **52**, 764.768 (1956)
- [7] Ooura T. and Mori M., The double exponential formula for oscillatory functions over half infinite interval, J. Comput. Appl. Math., **38** (1991), 353–360.
- [8] Ooura T., A continuous Euler transformation and its application to the Fourier transform of a slowly decaying function, J. Comput. Appl. Math., **130** (2001), 259–270.
- [9] Rall, L. B., Automatic Differentiation-Technique and Applications, Lecture Notes in Computer Science, Vol.120, Springer-Verlag, Berlin-Heidelberg-New York (1981)
- [10] The SIAM 100-Digit Challenge, <http://www-m3.ma.tum.de/m3old/bornemann/challengebook>
- [11] Yozo Hida, Xiaoye S. Li, David H. Bailey, Library for Double-Double and Quad-Double Arithmetic, Proc. 15th Symposium on Computer Arithmetic, (2007), 155–162
- [12] Yozo Hida, Xiaoye S. Li, David H. Bailey, Algorithms for Quad-Double Precision Floating Point Arithmetic, Lawrence Berkeley National Laboratory, (2000)