

多項式関数による変数変換型数値積分

平山 弘^{1,a)} 小宮 聖司^{1,b)}

概要：Euler-Maclaurin の総和公式とは、積分とその積分を台形公式を使って計算した値との誤差評価と見なせる。この誤差評価式には積分区間両端点における被積分関数の微分係数が含まれている。もし積分区間の両端点で被積分関数の関数値や微分係数がゼロにすることができれば積分は台形公式によって精度よく計算できると期待できる。この性質を使った数値積分公式としては、DE 公式や IMT 公式などがある。これらの公式では、超越関数を使った変数変換を使っている。

本文では、多項式を使って変数変換を行い数値積分が行うことを提案し、その有効性を調べた。また、この方法を使って Kahaner の問題を計算し、その性能やその性質を調べた。

キーワード：変数変換型数値積分公式、多項式、Euler-Maclaurin の総和公式

Numerical integration method by variable transformation by polynomial functions

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

Abstract:

Euler-Maclaurin's formula can be regarded as an error evaluation between the integral and the value calculated using the trapezoidal rule. This error evaluation formula includes the differential coefficient of the integrand at both ends of the integration interval. If the function value or derivative of the integrand can be zero at both ends of the integration interval It can be expected that the integral can be calculated accurately by the trapezoidal rule. Numerical integration formulas that use this property include the DE formula and the IMT formula. These formulas use change of variables using transcendental functions.

In this paper, we proposed that variable transformation is performed using polynomials and numerical integration is performed, and its effectiveness was investigated. We also used this method to calculate Kahaner's problems and investigate these performance and properties.

Keywords: Numerical Taylor series, series expansion method for differential equations, Euler-Maclaurin's formula

1. はじめに

数値積分や級数和の計算でよく使われる公式にオイラー・マクローリンの総和公式 (Euler-Maclaurin formula) がある。この公式はいろいろな文献 [1][5][6] 等で紹介されている。ここでは主に長田 [4] を参考にした。関数 $f(x)$ は区間

$[a, b]$ で連続で微分可能とする。積分区間 $[a, b]$ を n 等分して、台形公式を適用した計算値と厳密な積分値との差を微分係数を含む m 項で近似すると、次の式が成り立つ。

$$h \left\{ \frac{1}{2} f(a) + \sum_{k=1}^{n-1} f(a + kh) + \frac{1}{2} f(b) \right\} - \int_a^b f(x) dx (1) \\ = \sum_{k=1}^m \frac{B_{2k}}{(2k)!} h^{2k} \left\{ f^{(2k-1)}(b) - f^{(2k-1)}(a) \right\} + R_m$$

ここで、 B_n は Bernoulli 数 (Bernoulli Number)、 $h = \frac{b-a}{n}$ であり、

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

$$|R_m| \leq \frac{h^{2m+2} |B_{2m+2}|}{(2m+2)!} \int_a^b |f^{(2m+2)}(t)| dt \quad (2)$$

である。

Bernoulli 数 B_n は次のように定義されている。

$$\frac{x}{e^x - 1} = \sum_{k=0}^{\infty} B_k \frac{x^k}{k!} \quad (3)$$

最初の数値を書き出すと $B_0 = 1$ 、 $B_1 = -\frac{1}{6}$ 、 $B_2 = \frac{1}{6}$ 、 $B_3 = 0$ 、 $B_4 = -\frac{1}{30}$ 、 $B_5 = 0$ となる。 B_n は有理数で $n \geq 3$ が奇数ならば $B_n = 0$ となる。

上の公式 (1) から積分区間の両端点で微分係数が等しいすなわち $f^{(2k-1)}(a) = f^{(2k-1)}(b)$ ならば台形公式で高精度で積分が計算される可能性があることがわかる。このような例として、周期関数の一周に渡る積分はこのような例となる。また積分区間の両端点で $f^{(2k-1)}(a) = f^{(2k-1)}(b) = 0$ となるようにすると周期関数の一周に渡る積分のように高精度で計算できる可能性がある。このような変換として、次の IMT(伊理, 森口, 高澤) 公式がある。以下で紹介する。積分は有限区間の積分は区間 $[0, 1]$ の積分に簡単に変換できる。次の積分の変数変換を考える。

$$I = \int_a^b f(x) dx \quad (4)$$

$x = a + (b-a)s$ と変数変換すると、次のように積分区間を $[0, 1]$ にすることが出来る。

$$I = (b-a) \int_0^1 f(a + (b-a)s) ds \quad (5)$$

したがって、一般性を失うことなく、積分を区間 $[0, 1]$ の積分だけを考える。

$$I = \int_0^1 f(x) dx \quad (6)$$

この積分に対して、次のような変換を行う。 $g(s) = e^{-\frac{1}{s} - \frac{1}{1-s}}$ とすると $g(s)$ は積分区間の両端点で関数値および微分係数はゼロとなる関数なので、これを使って、次の $h(t)$ 関数を定義する。この方法が IMT 公式である。

$$h(t) = \frac{1}{Q} \int_0^t g(s) ds, \quad Q = \int_0^1 g(s) ds \quad (7)$$

$x = h(s)$ と変数変換すると

$$I = \frac{1}{Q} \int_0^1 f(h(s)) g(s) ds \quad (8)$$

積分区間を N 等分し、台形公式を利用して計算すると

$$I_h = h \sum_{k=1}^{N-1} f(h(hk)) g(hk) ds, \quad h = \frac{1}{N} \quad (9)$$

積分の端点では、関数 $f(x)$ の値は計算されないため積分区間の端点が特異点でも計算できる特徴を持つ。計算には、

$g(x)$ と $h(x)$ の値が必要である。 $g(x)$ は指数関数を含み計算が大変であるが、 $h(x)$ は指数関数を含む $g(x)$ の積分値が必要で計算がさらに大変である。このような困難さがあるためか、あまり使われていない。

2. 多項式による変換

ここでは次のような多項式による変数変換を行う。

$$I = \int_0^1 f(x) dx \quad (10)$$

を、次の関数

$$g(s) = (s(1-s))^n \quad (11)$$

を使って変数変換を行う。関数 $g(s)$ は積分区間の両端点 $[0, 1]$ でゼロで n 階微分係数までゼロである。

$$h(t) = \frac{1}{Q} \int_0^t (s(1-s))^n ds, \quad Q = \int_0^1 (t(1-t))^n dt \quad (12)$$

$x = h(t)$ と変数変換すると (10) の積分は

$$I = \frac{1}{Q} \int_0^1 f(h(t)) g(t) dt = \int_0^1 f(h(t)) (t(1-t))^n dt \quad (13)$$

IMT の公式と同様に

$$I_h = h \sum_{k=1}^{n-1} f(h(hk)) g(hk) ds, \quad h = \frac{1}{n} \quad (14)$$

Q の値は n が与えられれば、被積分関数は多項式式なので容易に計算できる。

$$Q_n = \int_0^1 x^n (1-x)^n dx$$

とすると、ベータ関数を使って次のように書ける。

$$Q = \frac{1}{Q_n} = \frac{1}{B(n+1, n+1)} = \frac{(2n+1)!}{(n!)^2}$$

色々な数値に対する Q の値を表 1 に示す。

表 1 $\frac{1}{Q_n}$ の値

n	積分値 ($1/Q_n$)
1	6
2	30
3	140
4	630
5	2772
10	3879876
15	4808643120
20	5651707681620
25	6446940928325352
30	7214139475456546864

多項式は Taylor 級数として計算することができる。Taylor 級数で多項式に関連する部分を以下で説明する。Taylor

級数を多項式と見なせば容易に推察できる。展開位置は常に原点と考えればよい。

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。

$$f(x) = f_0 + f_1x + f_2x^2 + f_3x^3 + f_4x^4 \cdots + f_nx^n \quad (15)$$

$$g(x) = g_0 + g_1x + g_2x^2 + g_3x^3 + g_4x^4 \cdots + g_nx^n \quad (16)$$

$$h(x) = h_0 + h_1x + h_2x^2 + h_3x^3 + h_4x^4 \cdots + h_nx^n \quad (17)$$

2.1 Taylor 級数の四則演算

n 次の多項式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 $h(x) = f(x) \pm g(x)$

$h(x) = f(x) \pm g(x)$ となるので、この式に (15) の $f(x)$ 、(16) の $g(x)$ および (17) の $h(x)$ を代入し、同じ次数の係数を等しいとすると次の関係式が得られる。

$$h_j = f_j \pm g_j (j = 0, \dots, n)$$

(2) 乗算 $h(x) = f(x)g(x)$

$h(x) = f(x)g(x)$ となるので、この式に (15) の $f(x)$ 、(16) の $g(x)$ および (17) の $h(x)$ を代入し、同じ次数の係数を等しいとすると次の関係式が得られる。

$$h_j = \sum_{k=0}^j f_k g_{j-k} (j = 0, \dots, n)$$

(3) べき乗関数 $h(x) = f(x)^\alpha$ (α は定数)

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。

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

(15) と (17) 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。

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

$\alpha = \frac{1}{2}$ とすると平方根を計算できる。

(4) 微分 $h(x) = \frac{df(x)}{dx}$

$h(x) = \frac{df(x)}{dx}$ のとき、次のような関係式が得られる。

$$h_n = 0, \quad h_j = (j+1)f_{j+1} \quad (j = 0, \dots, n)$$

このように、最高次数の係数 h_n は、0 とする。

(5) 積分 $h(x) = \int_0^x f(t)dt$

$h(x) = \int_0^x f(t)dt$ のとき次のような関係式が得られる。

$$h_0 = 0, \quad h_j = \frac{1}{j} f_{j-1} \quad (j = 1, \dots, n)$$

定数項は、積分定数なので、任意で良いが、ここで作成したプログラムでは、0 とする。

2.2 多項式変数変換関数の計算

$h(x)$ の計算には Taylor 展開用のプログラム [2] を使って計算した。多項式は Taylor 級数の四則演算を使えば容易に計算できる。当然ながら Taylor 展開式の積分ルーチンを使えば容易に積分出来る。Taylor 展開ルーチンでは、約 64 桁の浮動小数点数を使って計算した。上の表は整数になることが知られているので丸め処理を行って計算した。数式処理を使うことも考えたが、通常の数値積分の連携が悪いのでデバッグ中は大いに使用したが最終的にはすべて C++ 言語のプログラムになるようにした。

このような計算法は昔から知られていると思われるが途中で煩わしい計算が入るために研究がなされていないのではないかと推測している。

代数方程式の式が少し複雑だと、各次数の係数を計算するのが面倒なので各係数の値を計算する必要のない DKA 法を使っていると聞いている。これも同じ理由だと考えている。

積分値は次のように計算できる。

$$I_h = \frac{h}{Q} \sum_{k=1}^{N-1} f(h(hk))g(hk)ds, \quad h = \frac{1}{N} \quad (18)$$

2.3 プログラム例

ここでは $g(s) = x^n(1-x)^m$ としたとき $h(s)$ および Q の計算を行うプログラムである。1 行から始まる setcof 関数は $g(s)$ を積分し、 $h(s)$ を求める。 $(1-x)^m$ を 10 行目で展開計算する 11 行から始まるループでは、得られた展開式には x^n が乗算されているものとして積分する。15 行目で得られた展開式に 1 を代入して Q を求める。0 を代入するとゼロであるから 1 を代入すれば Q は求まる。その逆数を aa に入れる。20 行目で展開式を cof に代入する。22 行目以降で $h(x)$ を計算する。26 行目から始まるループでホーナー法で展開式を計算する。30 行目で $g(s)$ の x^n を考慮して値を計算する。

```
1: void setcof( const int n, const int m )
2: {
3:     taylor x, fd, f, fd1, fr, fl, ft ;
4:     double one=1, zero=0, half=0.5 ;
5:     double Q, s, t, h ;
6: //
7:     n0 = n ;
8:     n1 = m ;
```

```

9:      x = taylor( zero, zero, one ) ;
10:     fd = gr( x ) ;
11:     for( int i=0 ; i<=n1 ; i++ )
12:     {
13:         fd[i] = fd[i]/(i+n0+1) ;
14:     }
15:     Q = fd(one) ;
16:     s = 1.0 / Q;
17:     fd *= s ;
18:     np = fd.cur_deg ;
19:     aa = s ;
20:     cof = fd ;
21: }
22: double h( const double x )
23: {
24:     double ret ;
25:     ret = 0.0 ;
26:     for( int i=n1 ; i>=0 ; i-- )
27:     {
28:         ret = ret * x + cof[i] ;
29:     }
30:     ret *= pow(x,n0+1) ;
31:     return ret ;
32: }

```

このように簡単にプログラムを作成できる。ここで宣言されていない変数はグローバル変数として宣言されている。関数 g の $(1-x)^m$ は次のようにテンプレートで宣言されている。展開するとき、引数として Taylor 展開式、数値計算の時引数は double 型を使用するためである。1 回記述することによって展開式の計算と、積分の数値計算に利用される。

```

33: template< typename T>
34: T g( const T& x )
35: {
36:     T r = pow(T(1.0)-x,n1 ) ;
37:     return r ;
38: }

```

3. 数値実験

問題として Kahaner のテスト問題を計算した。Kahaner のテスト問題とは次のような問題である。2 番の問題は不連続関数なので省略した。

多項式で変換する場合には、次数は自由に選ぶことができる。その問題に合わせた次数を選ぶことができる。今回は (11) 式の n を 10 以下の多項式を利用しても計算したがあまり良い結果は得られなかった。 n を 10 から 18 の多項式を使った場合大体同じくらいの効率でかなり良いと思われた。表 2 に $n = 16$ の PL 場合と DE 法の結果を示した。

要求精度は 10^{-9} とした。DE 法 [7] のプログラムとして大浦 [3] 氏のプログラムを使用した。

DE 法では、関数が十分小さくなったら計算を止めるようになっているが多項式法は倍々と標本点が増えるようになっているため関数値を見ていないため不要な関数計算を行うことになり、標本点数が増える傾向にある。そのため計算精度が上がる傾向がある。

4. 終わりに

Taylor 展開ルーチンを使うことによって、多項式による変数変換を行う数値積分公式を自動的に作成することが出来た。

32 次の多項式を使った場合、DE 法と同じくらいの効率で計算することが出来た。

この方法を利用すれば、実行時に適切な多項式を選ぶことによって効率的な計算が可能になると思える。

今回は積分ルーチンの作成であったが、複雑な 1 変数の多項式の零点を計算するのに各次数の係数を計算することによって、通常の代数方程式の解法ルーチンを容易に使えるようにしたいと考えている。

現在一応多項式を使った変数変換型数値積分法のプログラムが作成できたがどのような多項式を選べば効率あるかを調べる必要がある。

積分区間内にピークがある場合どのような多項式を選べばよいか調べる必要がある。

参考文献

- [1] Davis P.J., Rabinwitz P.(森 正武訳), 計算機による数値積分法, 日本コンピュータ協会, (1981)
- [2] 平山, 舘野, 浅野, 川口, Taylor 級数演算ライブラリの使用法, 東北大学情報シナジーセンター大規模科学計算システム広報 SENAC, 40(2007) 29–68
- [3] 大浦拓哉, Ooura's Mathematical Software Packages, <http://www.kurims.kyotou>.
- [4] 長田直樹, お話: 数値解析第 3 回 オイラー・マクローリンの公式, <http://www.cis.twcu.ac.jp/osada/rikei/rikei2008-7.pdf>
- [5] 篠原能材, 数値解析の基礎, 7 章, 日新出版, (1978)
- [6] 杉原, 室田, 数値計算法の数理, 12 章, 岩波書店, (2007)
- [7] Takahasi, H. and Mori, M., Double exponential formula for numerical integration, *Publ. RIMS, Kyoto Univ.*,9(1974),121–141

表 2 多項式変換による数値積分の結果

No.	Integral	N	PL	N	DE
1	$\int_0^1 e^x dx$	62	2.0e-15	33	3.6e-13
3	$\int_0^1 \sqrt{x} dx$	62	6.7e-16	33	9.7e-14
4	$\int_{-1}^1 0.92 \cosh x - \cos x dx$	62	1.3e-15	67	1.8e-13
5	$\int_{-1}^1 \frac{1}{x^4+x^2+0.9} dx$	126	7.8e-15	65	9.7e-13
6	$\int_0^1 x \sqrt{x} dx$	62	6.1e-16	33	9.7e-14
7	$\int_0^1 \frac{1}{\sqrt{x}} dx$	30	1.6e-07	39	1.8e-11
8	$\int_0^1 \frac{1}{x^4+1} dx$	62	2.3e-10	65	1.1e-12
9	$\int_0^1 \frac{2}{2+\sin(31.4159x)} dx$	1022	7.0e-14	519	2.1e-12
10	$\int_0^1 \frac{1}{1+x} dx$	62	1.4e-15	33	1.5e-13
11	$\int_0^1 \frac{1}{e^x+1} dx$	62	5.0e-16	33	7.4e-14
12	$\int_0^1 \frac{x}{e^x-1} dx$	62	8.9e-16	33	1.5e-13
13	$\int_{0.1}^1 \frac{\sin(314.159x)}{3.14159x} dx$	1022	2.0e-16	451	9.5e-13
14	$\int_0^{10} \sqrt{50} \exp(-50 \times 3.14159x^2) dx$	254	1.9e-10	279	9.1e-14
15	$\int_0^{10} 25 \exp(-25x) dx$	126	4.9e-10	141	9.3e-14
16	$\int_0^{10} \frac{50}{3.14159(2500x^2+1)} dx$	254	1.0e-12	141	2.9e-14
17	$\int_{0/01}^1 \left(\frac{\sin(50 \times x)}{50 \times 3.14159x} \right)^2 \times 50 dx$	1022	2.9e-14	555	4.6e-14
18	$\int_0^\pi \cos(\cos x + 3 \sin x + 2 \cos 2x + 3 \sin 2x + 3 \cos 3x) dx$	254	2.2e-16	131	7.7e-13
19	$\int_0^1 \log(x) dx$	62	1.5e-14	33	3.1e-12
20	$\int_{-1}^1 \frac{1}{x^2+1.005} dx$	126	1.3e-15	65	1.4e-12
22	$\int_0^1 \frac{1}{\cosh^2(10(x-0.2))} + \frac{1}{\cosh^4(100(x-0.4))} + \frac{1}{\cosh^6(1000(x-0.6))} dx$	2046	1.1e-03	2009	7.7e-04