

# 数値積分 (3)

桂田 祐史

2016年5月18日, 2016年7月20日

まだまだ粗い。

5月18日の予定

1. 周期関数の一周積分に台形公式が有効な理由の説明。Euler-Maclaurin 展開。
2. 最初に Mathematica で変数変換をやってみる方が良いかも (§4)。DE 公式のアイデアの説明。数値例の紹介。
3. DE 公式の誤差解析についてのお話 — これは出来なかった。この文書からもカットした (5/25 の講義ノートの付録に回した)。

## 1 周期関数の周期積分に対する台形公式の誤差, 実関数論的な説明

周期関数の周期積分に対しては台形公式がベスト

この事実は古くから知られているが、通常、次の Euler-Maclaurin 展開 (実は有名な公式) によって説明される。左辺は  $I = \int_a^b f(x) dx$ , 右辺の第1項は複合台形則  $T_n$  であるから、誤差が右辺の第2項と第3項の和であるが、 $f$  が周期  $b-a$  であれば、右辺第2項の  $\sum$  の項はすべて0になり、誤差は  $O(h^{2m+1})$  であることが分かる。

**命題 1.1 (Euler-Maclaurin 展開, Euler (1736), MacLaurin (1742), Jacobi (1834), )**  
 $f: [a, b] \rightarrow \mathbb{R}$  が  $C^{2m}$  級であれば、

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

$$R_m = \frac{h^{2m+1}}{(2m)!} \int_0^1 B_{2m}(t) \left( \sum_{k=0}^{n-1} f^{(2m)}(a + kh + th) \right) dt.$$

ただし  $B_m, B_m(t)$  は、それぞれ次式で定義される **Bernoulli 数**、**Bernoulli 多項式** である:

$$\frac{s}{e^s - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} s^n, \quad \frac{se^{ts}}{e^s - 1} = \sum_{n=0}^{\infty} \frac{B_n(t)}{n!} s^n \quad (|s| < 2\pi).$$

(Bernoulli 数については、荒川・伊吹山・金子 [1] が詳しい。「複素関数」の講義ノート [2] の §9.5 にも少し解説しておいた。最初の数項を書いておく (それで数値積分の話には十分のはず)。

$$B_0 = 1, \quad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \quad B_3 = 0, \quad B_4 = -\frac{1}{30}, \quad B_5 = 0, \quad B_6 = \frac{1}{42}, \quad \dots$$

Mathematica には、ベルヌーイ数、ベルヌーイ多項式を計算する関数  $BernoulliB[n]$ ,  $BernoulliB[n, x]$  が用意されている。) )

**余談 1.1 (歴史メモ)** ハイラー・ヴァンナー [3] によると、Euler と Maclaurin は、 $\sum_{k=1}^n \frac{1}{k}$  や、 $\sum_{k=2}^n \log k =$

$\log n!$ ,  $\sum_{k=1}^n k^r$ ,  $\sum_{k=1}^n \frac{1}{k^r}$  などを計算するために

$$\sum_{i=1}^n f(i) = \int_0^n f(x) dx + \frac{1}{2} (f(n) - f(0)) + \frac{1}{12} (f'(n) - f'(0)) - \frac{1}{720} (f'''(n) - f'''(0)) - \frac{1}{30240} (f^{(5)}(n) - f^{(5)}(0)) + \dots$$

という公式を独立に導いた (例えば、 $r$  を自然数として、 $f(x) = x^r$  としてみると冪乗和  $\sum_{k=1}^n k^r$  の公式が得られる。公式の右辺が有限和になることに注意せよ。)

$$B_0 = 1, \quad \sum_{i=0}^{k-1} \binom{k}{i} B_i = 0 \quad (k = 2, 3, \dots)$$

で定義される Bernoulli 数<sup>1</sup>を用いると

$$\sum_{i=1}^n f(i) = \int_0^n f(x) dx + \frac{1}{2} (f(n) - f(0)) + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} (f^{(2k-1)}(n) - f^{(2k-1)}(0))$$

と書ける。この式は実は一般に成り立つ式ではない。例えば  $f(x) = \cos 2\pi x$  に対して、左辺は  $1 + 1 + \dots$ , 右辺は  $0 + 0 + \dots$  となり、左辺は収束すらしない。有限項で打ち切って、剰余項を与える必要がある。その結果

$$\sum_{i=1}^n f(i) = \int_0^n f(x) dx + \frac{1}{2} (f(n) - f(0)) + \sum_{j=2}^k \frac{B_j}{j!} (f^{(j-1)}(n) - f^{(j-1)}(0)) + \tilde{R}_k,$$

$$\tilde{R}_k = \frac{(-1)^{k-1}}{k!} \int_0^n \tilde{B}_k(x) f^{(k)}(x) dx.$$

が得られた (Jacobi, 1834 年)。ここで  $\tilde{B}_k(x)$  は、 $[0, 1]$  区間での  $B_k(x)$  を周期 1 の周期関数となるように  $\mathbb{R}$  全体に拡張したものを表す。その後、Wirtinger により発見された部分積分を繰り返す証明が簡明であるとか。 ■

## (参考) IMT 公式

(ここは 2016 年度の授業ではカットする。)

IMT 公式は、周期関数の 1 周期区間の積分は台形公式で高精度に近似出来るという事実に注目して考案された (伊理・森口・高澤 [4], [5])。)

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

---

<sup>1</sup>  $\frac{u}{e^u - 1} = \sum_{n=0}^{\infty} \frac{B_n}{n!} u^n$  で定義されると言っても同じである。

を計算するために、次式で定義される  $\varphi$  を用いて、変数変換  $x = \varphi(t)$  を行い、台形公式を用いる。

$$\varphi(x) = \frac{1}{Q} \int_0^x \exp\left(-\frac{1}{t} - \frac{1}{1-t}\right) dt,$$

$$Q := \int_0^1 \exp\left(-\frac{1}{t} - \frac{1}{1-t}\right) dt$$

= 0.00702 98584 06609 65623 92412 70530 35395 60761 55399 47535  
72487 96129 73882 86445 45869 54635 93462 85080 43897 70339 32

(一松 [6] では、小数点以下 34 桁までの数値が記してあって、25 桁以降が 70530 3401 となっているが、32 桁目から食い違っている。)

すなわち、次の  $I_n$  を近似値に採用する。

$$I_n = \frac{1}{n} \sum_{k=1}^{n-1} w_k^{(n)} f\left(x_k^{(n)}\right), \quad x_k^{(n)} := \varphi\left(\frac{k}{n}\right), \quad w_k^{(n)} := \varphi'\left(\frac{k}{n}\right).$$

適当な  $n$  に対して、数列  $\{x_k^{(n)}\}$ ,  $\{w_k^{(n)}\}$  を用意する必要がある。(今でこそ、Mathematica を使えば容易に計算できるが、そういうものがなければ、結構面倒である。)

## 2 $\mathbb{R}$ 上の広義積分に対する台形公式の誤差, 数値実験例の紹介

$\mathbb{R}$  上の広義積分

$$I = \int_{-\infty}^{\infty} f(x) dx$$

で、 $f$  が解析関数 ( $\mathbb{C}$  における実軸の近傍で正則な関数に拡張できる) の場合、 $I$  の近似値として

$$I_h := h \sum_{n=-\infty}^{\infty} f(nh)$$

を採用した場合の精度が非常に (神秘的に感じられるくらい) 良い。特に  $|x| \rightarrow \infty$  での減衰が速い場合、 $I_h$  を有限和で打ち切った

$$I_{h,N} = h \sum_{n=-N}^N f(nh)$$

が少ない計算量で高精度の値を与える。 $I_h$ ,  $I_{h,N}$  も **台形公式** と呼ばれる。

この事実の定式化と証明は一応用意したが (<http://nalab.mind.meiji.ac.jp/~mk/complex2/tmh.pdf> の付録 B)、ここでは数値例を紹介するにとどめて、次節ではそれを利用したアルゴリズム (DE 公式) の話をしよう。

**例 2.1 (遠方で速く減衰する関数の  $\mathbb{R}$  全体での積分)** (これは先週の資料にも載せたもののコピー) 有名な

$$I = \int_{-\infty}^{\infty} e^{-x^2} dx \quad (= \sqrt{\pi})$$

を、適当な  $h > 0$ ,  $N \in \mathbb{N}$  を選んで、

$$I_N := h \sum_{j=-N}^N f(jh)$$

で近似してみる。無限区間であるからこれまでとは異なるが、これも台形公式と呼ばれる。

$h = 1, N = 6; h = 1/2, N = 12; h = 1/4, N = 24$  として実行した ( $N$  は  $-6 \leq x \leq 6$  の範囲までと考えて定めた。  $|x| > 6$  のとき、  $0 < f(x) \leq 2.4 \times 10^{-16}$  であり、もう  $hf(jh)$  を加えても値が変わらない)。

```
$ cc -o example5 example5.c
$ ./example5
N      h              I              I-T
6      1      1.772453850905516  -1.833539e-04
12     0.5    1.772453850905516  -2.220446e-16
24     0.25   1.772453850905516  -4.440892e-16
```

$h = 1/2$  という粗い分割で ( $2N + 1 = 25$  点での値を用いて)、ほぼ使用している処理系の最高精度に到達している。 ■

### 3 二重指数関数型数値積分公式

高橋秀俊, 森正武の研究として名高い**二重指数関数型数値積分公式** (double exponential formula, DE 公式) を解説する。

授業では、先に絵的に理解してもらった方が良くかもしれないので、話す順番は工夫すること (先に 4 節の内容をやった)。

#### 3.1 基本的なアイデア

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

に

$$a = \lim_{t \rightarrow -\infty} \varphi(t), \quad b = \lim_{t \rightarrow \infty} \varphi(t)$$

を満足する滑らかな単調増加関数  $\varphi: \mathbb{R} \rightarrow (a, b)$  を用いて変数変換

$$x = \varphi(t)$$

を施すと

$$I = \int_{-\infty}^{\infty} f(\varphi(t))\varphi'(t) dt.$$

十分小さな  $h > 0$  を取り、次の式で  $I$  を近似することを考える。

$$I_h = h \sum_{n=-\infty}^{\infty} f(\varphi(nh))\varphi'(nh).$$

実際には、無限和の計算は実行出来ないなので、適当な  $N \in \mathbb{N}$  を選び、

$$I_{h,N} = h \sum_{n=-N}^N f(\varphi(nh))\varphi'(nh).$$

と打ち切ったものを計算する。  $I_h, I_{h,N}$  も台形公式と呼ばれる。

注  $I_{h,N}$  は、一応  $\sum_n f(x_n)w_n$  (標本点における  $f$  の値に重みをかけたものの総和) という形をしている。

$\varphi$  をどう選択するのが良いだろうか? 高橋・森 ([7], [8]) の結論は、(有界区間における積分  $\int_a^b f(x) dx$  の場合には)

$$(1) \quad \varphi'(t) \doteq \exp(-C \exp |t|) \quad (|t| \rightarrow \infty)$$

という評価が成り立つようにするのが最適である、というものであった。これが「二重指数関数型数値積分公式」の名前の由来である (変数変換そのものも「二重指数関数型変数変換」と呼ばれる)。

念のため: (1) はもちろん

$$\lim_{t \rightarrow \infty} \varphi'(t) = 0 \quad (\text{収束が非常に速い})$$

を意味するが、それから  $\varphi(t)$  が  $t \rightarrow -\infty, \infty$  のとき  $\varphi(a), \varphi(b)$  に急速に近づくことにもなる。

以下は話が細くなるのでお話。

(1) がどう導かれたか

話を簡単にするために  $h$  は固定しておくことにする。

$$I - I_{h,N} = \Delta I_h + E_{h,N}, \quad \Delta I_h := I - I_h, \quad E_{h,N} := I_h - I_{h,N}$$

と分解して考える。

$E_{h,N} = I_h - I_{h,N}$  (「項の打ち切り誤差」) を小さくしたいが、そのためには  $|t| \rightarrow \infty$  とするとき、 $f(\varphi(t))\varphi'(t)$  は速く 0 に減衰することが望まれる。ところがあまり急激に減衰すると、刻み幅  $h$  が相対的に大きくなり、逆に精度が落ちると考えられる。

離散化誤差  $\Delta I_h = I - I_h$  と  $E_{h,N}$  がほぼ等しくなるところで無限和を切ると仮定して解析を行うと、(1) が導かれる、ということである。その後、杉原正顯氏によって、この最適性は定理の形でより厳密に述べられるようになった ([9], [10])。

## 3.2 具体的な積分公式

### 3.2.1 有限区間上の積分

有界区間  $(a, b)$  上の積分は、1 次関数による変数変換  $x = a + \frac{b-a}{2}(u+1)$  により、 $(-1, 1)$  上の積分に変換できるので、

$$I = \int_{-1}^1 f(x) dx$$

を考えれば十分である ( $(a, b)$  の場合に具体的にどうするか、「レポート課題 part 1 について」<sup>2</sup> に書いておいた)。

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f(a + (b-a) * (u+1)/2) du.$$

これに対しては

$$(2) \quad \varphi_1(t) := \tanh\left(\frac{\pi}{2} \sinh t\right) \quad (t \in \mathbb{R})$$

<sup>2</sup><http://nalab.mind.meiji.ac.jp/~mk/complex2/report1-hint.pdf>

とにおいて、変数変換  $x = \varphi_1(t)$  を施す。

$$\varphi_1'(t) = \frac{\pi}{2} \cdot \frac{\cosh t}{\cosh^2(\pi/2 \sinh t)}$$

であり、

$$\lim_{t \rightarrow \pm\infty} \varphi_1(t) = \pm 1 \quad (\text{複号同順}), \quad \lim_{t \rightarrow \pm\infty} \varphi_1'(t) = 0.$$

念のため公式を書いておくと

$$I_h = \frac{\pi}{2} h \sum_{n=-\infty}^{\infty} f\left(\tanh\left(\frac{\pi}{2} \sinh(nh)\right)\right) \frac{\cosh nh}{\cosh^2(\pi/2 \sinh(nh))}.$$

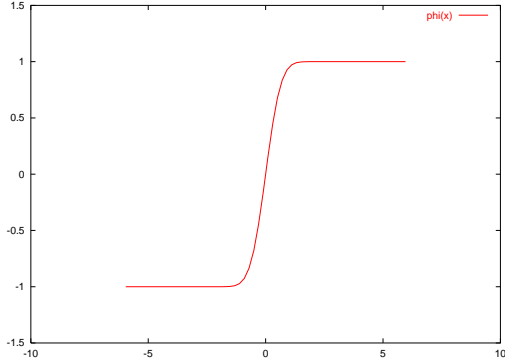


図 1:  $\varphi(t) = \tanh\left(\frac{\pi}{2} \sinh t\right)$

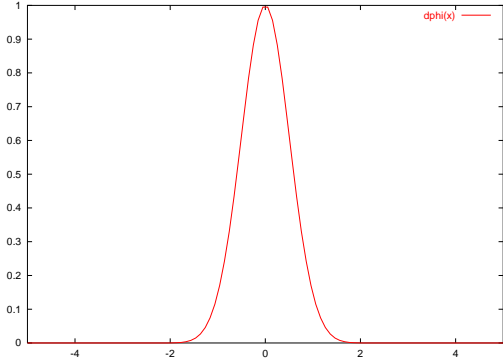


図 2:  $\varphi'(t) = \frac{\pi}{2} \cdot \frac{\cosh t}{\cosh^2(\pi/2 \sinh t)}$

### 3.2.2 $\mathbb{R}$ 上の減衰の緩い関数の積分

$$I = \int_{-\infty}^{\infty} f(x) dx$$

において、 $x \rightarrow \infty$  のとき  $f$  の減衰が緩い、例えば

$$\exists r > 1 \quad \text{s.t.} \quad f(x) \sim \frac{C}{|x|^r}$$

のような代数的な減衰の場合は、直接台形則を適用するのではなく、

$$(3) \quad \varphi_2(t) = \sinh\left(\frac{\pi}{2} \sinh t\right)$$

とにおいて、変数変換  $x = \varphi_2(t)$  を施すことで、効率の良い公式が得られる。

$$\lim_{t \rightarrow \pm\infty} \varphi_2(t) = \pm\infty \quad (\text{複号同順}).$$

この場合は、 $t \rightarrow \pm\infty$  のとき  $\varphi_2'(t)$  は減衰しないが、 $f(\varphi_2(t))\varphi_2'(t)$  は二重指数関数的に減衰する。  
念のため公式を書いておくと

$$I_h = \frac{\pi h}{2} \sum_{n=-\infty}^{\infty} f\left(\sinh\left(\frac{\pi}{2} \sinh nh\right)\right) \cosh nh \cosh\left(\frac{\pi}{2} \sinh nh\right).$$

( $\mathbb{R}$  上のそこそこ減衰の速い関数の場合はどうするか？と言っても、 $f(x) = e^{-|x|}$  は解析的ではないし、 $f(x) = e^{-x^2}$  は十分速いので、あまり考える必要はないか？)

### 3.2.3 半無限区間上の減衰の緩い関数の積分

(これは 2016/5/18 の授業では説明しなかった関数。)

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

において  $f$  の減衰が代数的な場合は、

$$(4) \quad \varphi_3(t) := \exp(\pi \sinh t) \quad (t \in \mathbb{R})$$

とにおいて、変数変換  $x = \varphi_3(t)$  を施す。

$$\varphi_3'(t) = \pi \exp(\pi \sinh t) \cosh t$$

であり、

$$\lim_{t \rightarrow -\infty} \varphi_3(t) = 0, \quad \lim_{t \rightarrow \infty} \varphi_3(t) = \infty, \quad \lim_{t \rightarrow -\infty} \varphi_3'(t) = 0.$$

$t \rightarrow \infty$  のとき  $\varphi_3'(t)$  は減衰しないが、 $f(\varphi_3(t))\varphi_3'(t)$  は二重指数関数的に減衰する。

念のため公式を書いておくと

$$I_h = \pi h \sum_{n=-\infty}^{\infty} f(\exp(\pi \sinh nh)) \exp(\pi \sinh nh) \cosh nh.$$

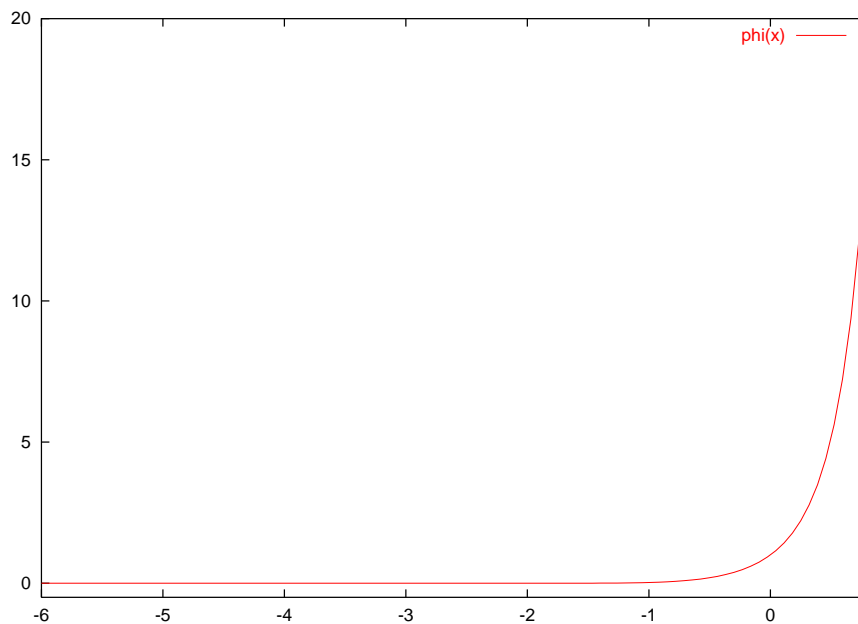


図 3:  $\varphi(t) = \exp(\pi \sinh t)$

### 3.2.4 半無限区間上の一重指数関数的な減衰をする関数の積分

例えば

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

において、

$$f(x) \sim f_1(x)e^{-x} \quad (x \rightarrow \infty), \quad (f_1(x) \text{ は減衰が代数的か、あるいは単に有界で減衰しない})$$

のような場合、

$$(5) \quad \varphi_4(t) = \exp(t - \exp(-t))$$

とにおいて、変数変換  $x = \varphi_4(t)$  を施してから台形公式を適用する。

$$\lim_{t \rightarrow -\infty} \varphi_4(t) = 0, \quad \lim_{t \rightarrow \infty} \varphi_4(t) = \infty, \quad \lim_{t \rightarrow -\infty} \varphi_4'(t) = 0.$$

$t \rightarrow \pm\infty$  のとき  $\varphi_4'(t)$  は減衰しないが、 $f(\varphi_4(t))\varphi_4'(t)$  は二重指数関数的に減衰する。  
公式は

$$I_h = h \sum_{n=-\infty}^{\infty} f(\exp(nh - \exp(-nh))) (1 + \exp(-nh)) \exp(nh - \exp(-nh)).$$

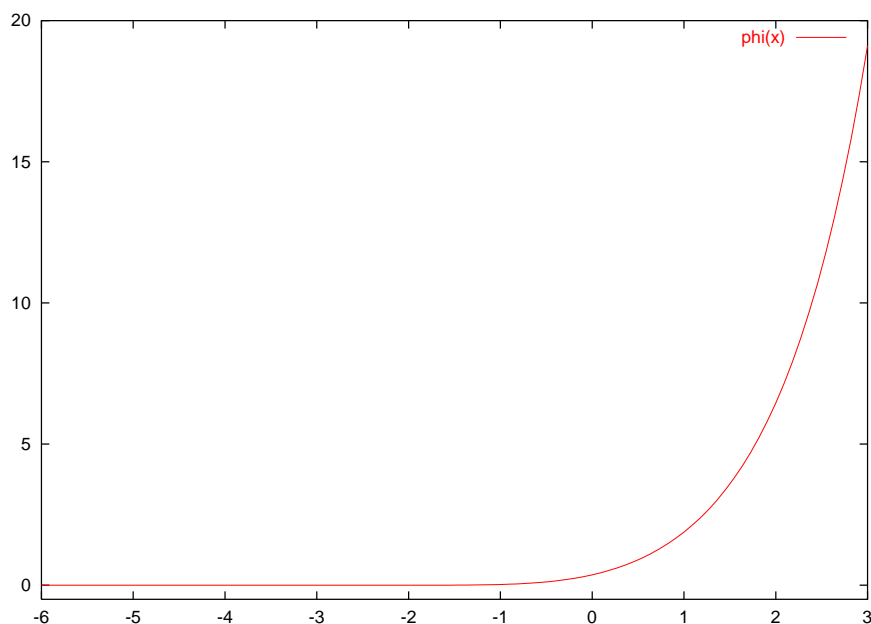


図 4:  $\varphi(t) = \exp(t - \exp(-t))$

### 3.3 試してみよう

例 3.6 (端点の特異性に強い)

$$I = \int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}, \quad I = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx = \pi.$$

プログラム `example6.c`<sup>3</sup> を実行してみよう。

<sup>3</sup><http://nalab.mind.meiji.ac.jp/~mk/complex2/example6.c>



```

$ cc -o example6 example6.c
$ ./example6
test1 (sqrt(1-x^2) の積分)
h=1.000000, I_h=      1.7125198292703636, I_h-I=1.417235e-01
h=0.500000, I_h=      1.5709101233831166, I_h-I=1.137966e-04
h=0.250000, I_h=      1.5707963267997540, I_h-I=4.857448e-12
h=0.125000, I_h=      1.5707963267948970, I_h-I=4.440892e-16
h=0.062500, I_h=      1.5707963267948968, I_h-I=2.220446e-16
h=0.031250, I_h=      1.5707963267948972, I_h-I=6.661338e-16
h=0.015625, I_h=      1.5707963267948974, I_h-I=8.881784e-16
h=0.007812, I_h=      1.5707963267948941, I_h-I=-2.442491e-15
h=0.003906, I_h=      1.5707963267948979, I_h-I=1.332268e-15
h=0.001953, I_h=      1.5707963267948972, I_h-I=6.661338e-16
test2 (1/sqrt(1-x^2) の積分)
h=1.000000, I_h=      3.1435079763395439, I_h-I=1.915323e-03
h=0.500000, I_h=      3.1415926717394895, I_h-I=1.814970e-08
h=0.250000, I_h=      3.1415926194518016, I_h-I=-3.413799e-08
h=0.125000, I_h=      3.1415926318228000, I_h-I=-2.176699e-08
h=0.062500, I_h=      3.1415926343278699, I_h-I=-1.926192e-08
h=0.031250, I_h=      3.1415926326210668, I_h-I=-2.096873e-08
h=0.015625, I_h=      3.1415926323669527, I_h-I=-2.122284e-08
h=0.007812, I_h=      3.1415926327540080, I_h-I=-2.083579e-08
h=0.003906, I_h=      3.1415926312582507, I_h-I=-2.233154e-08
h=0.001953, I_h=      3.1415926319069589, I_h-I=-2.168283e-08

```

$I$  については、 $h = 1/8$ ,  $N = 24$  で誤差が  $10^{-16}$  程度になっている。 $J$  については、 $h = 1/2$ ,  $N = 6$  で誤差が  $10^{-8}$  程度になっている (それ以上細かくしても改善されない)。シンプソン則の場合と比較してみると。 $N$  は試行錯誤で決めたが、 $h$  や  $N$  を自動的に選択するプログラムも作成可能である (詳しいことは省略)。■

なお、 $1/\sqrt{1-x^2}$  の数値積分の精度が  $10^{-8}$  程度までしか上がらない理由は、 $\pm 1$  に近い  $x$  に対して  $\sqrt{1-x^2}$  を計算する際に、桁落ちが生じて精度が落ちるからである。きちんと桁落ちの対策をすると  $10^{-16}$  近くまで精度をあげることが出来る (2016 年度はレポート課題にするので、桁落ち対策プログラムの公開は遅らせる)。

### 例 3.7 (減衰の遅い関数)

$$I = \int_{-\infty}^{\infty} \frac{dx}{1+x^2} = \pi.$$

$x \rightarrow \pm\infty$  のとき、被積分関数の減衰はあまり速くないため、単純に台形公式を適用すると、なかなか精度の良い近似値が得られない (その数値例も用意すべきであったか...)。  $x = \varphi_2(t)$  を用いた DE 公式を用いると次のようになる。 ( $|t| \leq 4$  まで計算している。)

```

$ cc -o example7 example7.c
$ ./example7
1 / (1 + x^2) の積分)
h=1.000000, I_h=      3.1435079789309333, I_h-I=1.915325e-03
h=0.500000, I_h=      3.1415926733057047, I_h-I=1.971591e-08
h=0.250000, I_h=      3.1415926535897944, I_h-I=1.332268e-15
h=0.125000, I_h=      3.1415926535897931, I_h-I=0.000000e+00
h=0.062500, I_h=      3.1415926535897931, I_h-I=0.000000e+00
h=0.031250, I_h=      3.1415926535897936, I_h-I=4.440892e-16
h=0.015625, I_h=      3.1415926535897927, I_h-I=-4.440892e-16
h=0.007812, I_h=      3.1415926535897913, I_h-I=-1.776357e-15
h=0.003906, I_h=      3.1415926535897913, I_h-I=-1.776357e-15
h=0.001953, I_h=      3.1415926535897976, I_h-I=4.440892e-15

```

$h = 1/4$ ,  $N = 16$  で誤差  $10^{-15}$  になっている。 ■

### 3.4 基本的な性質

- (名前の由来) 要するに変数変換後に得られる被積分関数  $f(\varphi(t))\varphi'(t)$  ( $t \in \mathbb{R}$ ) が二重指数関数的に減衰する:

$$|f(\varphi(t))\varphi'(t)| \leq C \exp(-C' \exp|t|) \quad (|t| \rightarrow \infty).$$

- 端点における特異性に強い。例えば次のような積分でも大丈夫。

$$I = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx$$

- (誤差の性質)

$$\Delta I_h \sim \exp\left(-\frac{C}{h}\right).$$

これから

$$\Delta I_{h/2} \sim \exp\left(-\frac{2C}{h}\right) \sim (\Delta I_h)^2.$$

つまり刻み幅を半分にすると、結果の有効桁数が 2 倍になる。

- Simpson 則などと比べて桁違いに高性能
  - 同じ手間で精度が何桁も良い
  - 同じ精度を得るのに手間が桁違いに少ない
- Gauss 型公式 (IMT 公式, DE 公式発見以前は究極の公式だった) と比べても
  - やはり桁違いに高性能 (Simpson 則との比較と同様)
  - 自動積分が出来るのは有利 (これが出来ないのは Gauss 型公式の弱点)
  - 分点や重みが計算しやすい (Gauss 型の場合は手間がかかり注意が必要 — 面倒)

一方、以下のことは注意すべきである。

- 低次の多項式に対しても誤差が 0 とはならない (**固有誤差**)。
- アンダーフロー、オーバーフローが起りやすく、プログラムを書くときに注意が必要である。

DE 公式のプログラミング上の注意については、森 [11] が詳しい (と言われるけれど、これは入手しにくいので、誰かエッセンスを抽出して新しく解説を書くべきだよね)。

## 4 Mathematica で試してみよう

### 4.0 おまけ

次の  $\varphi_0$  も使うことがある (説明を準備する? でも一重指数関数的にしか減衰しないような気がするが)。

$$\varphi_0(t) = \tanh(t).$$

$$\varphi_0(\mathbb{R}) = (-1, 1).$$

次の  $\psi$  はぱつと見は  $\varphi_0$  と似たグラフになるが、役に立たない。

$$\psi(t) = \frac{2}{\pi} \arctan(t).$$

$$\psi(\mathbb{R}) = (-1, 1).$$

## 4.1 $\int_{-1}^1 f(x) dx$ 用の変数変換

$$I = \int_{-1}^1 f(x) dx$$

を計算するため。

$$\varphi_1(t) := \tanh\left(\frac{\pi}{2} \sinh(t)\right).$$

$$\varphi_1(\mathbb{R}) = (-1, 1).$$

$F(t) := f(\varphi_1(t))\varphi_1'(t)$  のグラフは釣鐘形をしている (図 5)。

```

phi1[t_] := Tanh[Pi/2 Sinh[t]]
Plot[phi1[t], {t, -3, 3}]
Plot[phi1'[t], {t, -3, 3}]

f[x_] := 1/(x+2)
Plot[f[x], {x, -1, 1}]
Plot[phi1'[t], {t, -3, 3}]
Plot[F[t], {t, -5, 5}, PlotRange -> All]

f[x_] := 1/Sqrt[1-x^2]
Plot[f[x], {x, -1, 1}]
F[t_] := f[phi1[t]] phi1'[t]
g1=Plot[F[t], {t, -3, 3}, PlotRange -> All]

g2=LogPlot[F[t], {t, -3, 3}]

```

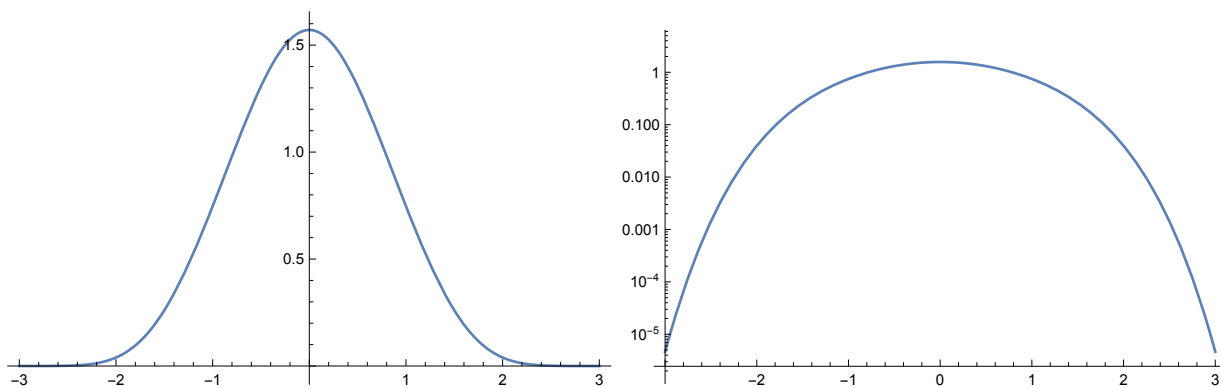


図 5:  $F(t) = f(\varphi_1(t))\varphi_1'(t)$  のグラフ (普通の日盛り、縦軸対数日盛り)

## 4.2 $\int_{-\infty}^{\infty} f(x) dx$ (ゆっくり減衰) 用の変数変換

$$I = \int_{-\infty}^{\infty} f(x) dx$$

を計算するため。ここで  $f(x)$  の  $x \rightarrow \pm\infty$  での減衰はゆっくりとする (例えば  $f(x) = \frac{1}{1+x^2}$ )。

$$\varphi_2(t) := \sinh\left(\frac{\pi}{2} \sinh(t)\right).$$

$$\varphi_2(\mathbb{R}) = \mathbb{R}.$$

```

phi2[t_] := Sinh[Pi/2 Sinh[t]]
Plot[phi2[t], {t, -2, 2}]
Plot[phi2'[t], {t, -2, 2}]

f[x_] := 1/(1+x^2)
Plot[f[x], {x, -5, 5}]
F[t_] := f[phi2[t]]phi2'[t]
Plot[F[t], {t, -5, 5}, PlotRange -> All]

LogPlot[F[t], {t, -5, 5}, PlotRange -> All]

```

やはり  $F(t) := f(\varphi_2(t))\varphi_2'(t)$  のグラフは釣鐘形になる。 $F$  が急激に減衰することは、やや分かりにくい。導関数  $\varphi_2'$  は、 $t \rightarrow \pm\infty$  のとき急激に減衰する関数でなく、その反対に急激に増大する。この点は  $\varphi_1$  とは大きく異なる。それにもかかわらず、 $f(\varphi_2(t))\varphi_2'(t)$  は急激に減衰する関数になる。これは少し考えると納得できるが(ぜひ考えてみよう)、ここでは論より証拠、目で見よう(図6)。(  $t = \pm 5$  では、 $10^{52.39}$  に  $10^{-100.64}$  をかけて  $10^{-48.25}$  弱とか、出入りが実に激しい…実に豪快な技ですね。それにしても良くこういうことを思いつくものだ。)

```

gr=LogPlot[{phi2'[t],f[phi2[t]],F[t]},{t,-5,5},PlotRange->All,AspectRatio->1]
Export["phi2.eps", gr]

```

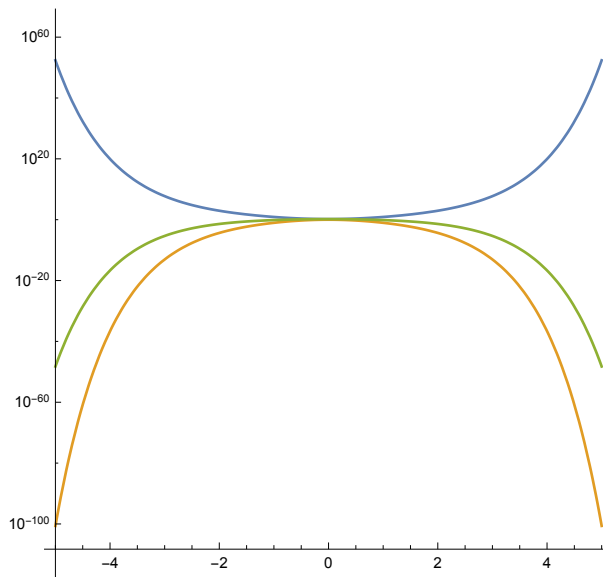


図6: 上から順に  $\varphi_2'$ ,  $f(\varphi_2(t))\varphi_2'(t)$ ,  $f(\varphi_2(t))$  のグラフ (縦軸対数目盛)

### 4.3 $\int_0^\infty f(x) dx$ 用の変数変換

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

を計算するため。 $x \rightarrow +\infty$  のときの  $f(x)$  の減衰はそこそこ速い(一重指数関数的)とする。

2016/5/18 の授業では、この関数に  $\varphi_3$  という名前をつけたが、 $\varphi_4$  という名前に変更することにした。

この場合は、

(6) 
$$\varphi_4(t) := \exp(t - \exp(-t))$$

という変数変換を用いる。これは  $\mathbb{R}$  を  $(0, \infty)$  に写す。

$$\varphi_4(\mathbb{R}) = (0, \infty).$$

結論を…  $F(t) := f(\varphi_4(t))\varphi_4'(t)$  は釣鐘形。

```
phi4[t_] := Exp[t-Exp[-t]]
Plot[phi4[t], {t, -5, 5}]
Plot[phi4'[t], {t, -5, 5}]

f[x_] := x^3 Exp[-x]
Plot[f[x], {x, 0, 10}]
F[t_] := f[phi4[t]]phi4'[t]
g1=Plot[F[t], {t, -5, 5}, PlotRange -> All]

g2 = LogPlot[F[t], {t, -4.5, 7}, PlotRange -> All]
```

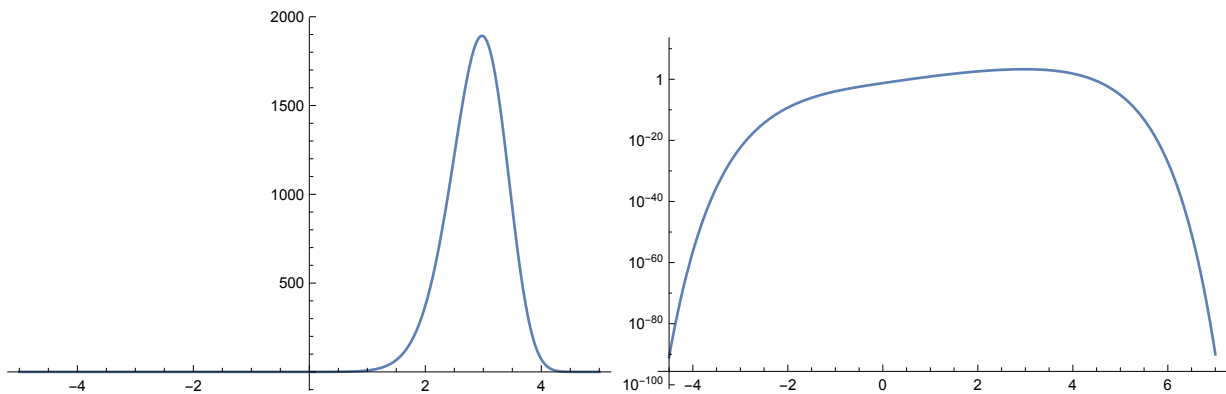


図 7:  $F(t) = f(\varphi_4(t))\varphi_4'(t)$  のグラフ (普通の日盛り、縦軸対数日盛り)

グラフを見ると、この変数変換を使って数値積分するときは、 $I_{h,N}$  でなく、

$$I_{h,N_1,N_2} = h \sum_{n=-N_1}^{N_2} f(\varphi_4(nh))\varphi_4'(nh)$$

を使うべきということも良く分かる。

## A 浮動小数点数メモ & DE 公式のプログラミングのヒント

### A.1 浮動小数点数の精度、値の範囲

要点

Mac の C 言語処理系の double は、10 進法に換算して 16 桁弱の精度、その精度を保って計算できる値の範囲は絶対値で  $2.2 \times 10^{-308} \sim 1.79 \times 10^{308}$

現在、C 言語等でプログラミングする場合、実数は浮動小数点数で表現される。パソコン、ワークステーションでは、IEEE 754 規格に準拠した浮動小数点数が使える。

double 型のデータでは、10 進法に換算して、16 桁弱の精度を持つ (2 進法で 53 桁なので、いわゆる計算機イプシロンは  $2^{-52} = 2.22045 \dots \times 10^{-16}$  である)。正の正規化数のうち、最小の数は

$2^{-1022} \doteq 2.2250738585 \times 10^{-308}$ , 最大の数は  $2^{1024}(1 - 2^{-53}) \doteq 1.797693134862315 \times 10^{308}$  であり、正の非正規化数のうち最小の数は  $2^{-1074} \doteq 4.940656 \times 10^{-324}$  である ([12])。

## A.2 DE 公式を使う場合の注意

DE 公式では、指数関数や二重指数関数が出て来るので、絶対値が極度に大きい数、極度に小さい数が登場する。

$\exp(x)$  は  $-708.396419 < x < 709.782712 \dots$  位までは、オーバーフローもアンダーフローもしない。

いわゆる IBM 互換の大型電子計算機では、 $10^{75}$  位までしか表現できない (詳しいことは知らない)。 $-173 < x < 172$  位まではオーバーフローもアンダーフローもしない。

$\varphi_1(t) = \tanh\left(\frac{\pi}{2} \sinh t\right)$  が、 $\varphi_1'(t) = 10^{-75}$  となるのは、 $t = 4.733$  位。これはかなり小さい。 $\varphi_1'(t) = 10^{-308}$  となるのは、 $t = 6.12163$  位。

$x$	$\varphi_1^{-1}(x)$ ( $\varphi_1(t) = x$ となる $t$ )
$2^{-53} = 2.22045 \dots \times 10^{-16}$	3.24973264407207547743154112423
$10^{-75}$	4.73348753384913205340507233810
$2.225 \times 10^{-308}$	6.12163124319116296579106744297

表 1: 注意すべき限界値

$|nh| \leq 3$  では十分な精度が得られない可能性がある。一方  $|nh| = 10$  のようなのは大きすぎて明らかな無駄。IBM 互換機ならば  $|nh| \simeq 4.5$  程度, IEEE 754 規格を用いたコンピュータでは  $|nh| \simeq 6$  程度が目安か。

$t = 3.154$  のとき、 $1 - \varphi_1(t) = 2.22 \times 10^{-16}$  なので、それを超えると、 $\varphi_1(t)$  の計算値は 1 になるおそれがある。 $f(x) = \frac{1}{\sqrt{1-x^2}}$  のように、 $x = \pm 1$  で定義されていない関数の積分を計算する場合は対策が必要である。

$t$  が 3 を超えて  $|f(\varphi_1(t))\varphi_1'(t)|$  が目標とする精度よりも小さくなっていけば良いが、そうでない場合は、何か対策を施さないと十分な精度が得られない可能性が高い。

以上は  $t > 0$  の場合を考えたが、 $t < 0$  でも同様である。 $t = -3.145$  のとき、 $\varphi_1(t) - (-1) = 2.22 \times 10^{-16}$  なので、 $t$  がそれより小さいと、 $\varphi_1(t)$  の計算値が  $-1$  になるおそれがある。 $t$  が  $-3$  より小さくなったとき、 $|f(\varphi_1(t))\varphi_1'(t)|$  が目標とする精度よりも小さくなっていけば良いが、そうでない場合は、対策が必要である。

一度、どういう数値の和を取っているのか、個々の数値を表示して検討してみることを勧める。

```

N = ceil(3.25 / h);
s = 0.0;
for (n = - N; n <= N; n++) {
    t = n * h;
    f1 = f(phi(t));
    f2 = dphi(t);
    term = f1 * f2;
    s += term;
    printf("%f %e %e %e %20g\n", t, f1, f2, term, s);
}

```

## 参考文献

- [1] 荒川恒男, 伊吹山知義, 金子昌信: ベルヌーイ数とゼータ関数, 牧野書店 (2001).
- [2] 桂田祐史: 応用複素関数, 現象数理解析学科での講義科目「応用複素関数」の講義ノート. <http://nalab.mind.meiji.ac.jp/~mk/complex2/ouyou.pdf> (2015~).
- [3] E. ハイラー, G. ヴァンナー: 解析教程 上, 下, シュプリンガーフェアラー東京 (1997), “Analysis by its History” の邦訳. Wanner は元は「ワナー」と綴っていた.
- [4] 伊理正夫, 森口繁一, 高澤嘉光: ある数値積分公式について, 京都大学数理解析研究所講究録, Vol. 91, pp. 82–119 (1970), <http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/0091-03.pdf>.
- [5] Iri, M., Moriguti, S. and Takasawa, Y.: On a certain quadrature formula, *J. Comput. Appl. Math*, Vol. 17, pp. 3–20 (1987).
- [6] ひとつまっしん 一松信: 留数解析 — 留数による定積分と級数の計算, 共立出版 (1979), 第5章は高橋-森理論の解説.
- [7] 高橋秀俊, 森正武: 変数変換によって得られる積分公式 (2), 数理解析研究所講究録, Vol. 172, pp. 88–104 (1973), <http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/0172-06.pdf>.
- [8] Takahashi, H. and Mori, M.: Double exponential formulas for numerical integration, *Publ. RIMS Kyoto Univ.*, Vol. 9, pp. 721–741 (1974).
- [9] 杉原正顕: DE 変換公式の最適性について, 京都大学数理解析研究所講究録, Vol. 585, pp. 150–175 (1986), <http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/0585-09.pdf>.
- [10] Sugihara, M.: Optimality of the double exponential formula – functional analysis approach –, *Numerische Mathematik*, Vol. 75, pp. 379–395 (1997).
- [11] 森正武: FORTRAN 77 数値計算プログラミング, 岩波書店 (1986, 1987).
- [12] 桂田祐史: IEEE754 倍精度浮動小数点数のフォーマット, [http://nalab.mind.meiji.ac.jp/~mk/labo/text/ieee\\_format.pdf](http://nalab.mind.meiji.ac.jp/~mk/labo/text/ieee_format.pdf) (2002).