

積分区間の分割による数値積分の高速化

平山 弘^{1,a)}

概要: 悪条件の積分の積分区間 $[a,b]$ を 2 または 3 以上に分割し、高次の数値積分公式をこれらの積分区間に適用すると、少ない標本点数で良い結果が得られる場合がある。

本論文では、複素平面上の経路積分から、数値積分の収束速度定義し、積分区間の分割方法によって、効率的に高速に数値積分を計算する方法を論じる。

キーワード: 数値積分, 積分区間分割, 複素平面上の経路積分

Effective Numerical Integration Method by Division of Integral Interval

HIROSHI HIRAYAMA^{1,a)}

Abstract: When an integral interval $[a,b]$ of the integral with the bad condition is divided into two or more intervals and higher order numerical integral formula is applied every area and the whole one, good results are sometimes obtained with few sample points.

In this paper, we show that you can get the condition to define convergence speed of numerical integration and partition method from numerical integration method of contour integral on complex plane.

Keywords: Numerical Integration, Division of Integral Interval, Contour Integration on Complex Plane

1. はじめに

次のような有限区間 $[a,b]$ に渡る関数 $f(x)$ の積分を考える。

$$I = \int_a^b f(x) dx \quad (1)$$

ここでは、関数 $f(x)$ は積分区間の近くに特異点を持つような条件の悪い積分とする。式 (1) で表される積分は、Cauchy の積分表示を使うと、周期関数の 1 周期に渡る積分に変換できる。周期関数の 1 周期に渡る積分は、台形公式を使って効率的に計算できることが知られているので、この変換によって効率的に数値積分できると期待できる。積分区間内で特異性を持つ場合、被積分関数は、 $f(x) = w(x)g(x)$ のように表現することができる。

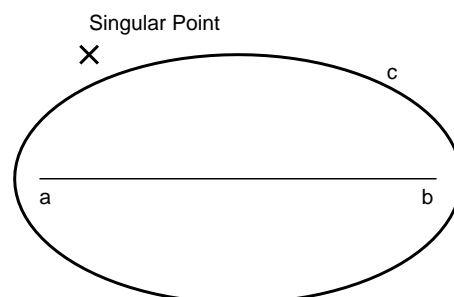


図 1 周回積分の積分路

ここで、 $w(x)$ は特異性を持つ部分、 $g(x)$ は積分区間内で正則な部分である。 $w(x)$ と $g(x)$ の選び方は一意性はないが、以下の議論ではどのようにとっても同じ結果となる。Cauchy の積分表示を利用すると、次のように周回積分に

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

変換することができる。

$$\begin{aligned}
 I &= \int_a^b f(x)dx \\
 &= \int_a^b w(x) \frac{1}{2\pi i} \int_c \frac{g(z)}{z-x} dz dx \\
 &= \frac{1}{2\pi i} \int_c g(z) \left[\int_a^b \frac{w(x)}{z-x} dx \right] dz \\
 &= \frac{1}{2\pi i} \int_c g(z) H(z) dz
 \end{aligned} \tag{2}$$

積分路 c は、図 1 のように積分区間を囲み、積分区間以外に特異点を含まないような閉積分路である。括弧 $[]$ の部分 $H(z)$ は、関数 $w(x)$ の Hilbert 変換と呼ばれる。たとえば、 $w(x) = 1$ の場合 $\int_a^b \frac{w(x)}{z-x} dx = \log \frac{z-a}{z-b}$ であるから、次のようになる。

$$\int_a^b f(x)dx = \frac{1}{2\pi i} \int_c f(x) \log \frac{z-a}{z-b} dz$$

$w(x) = 1$ のとき以外の有限区間の Hilbert 変換例を表 1 に示す。

表 1 Finite Hilbert Transformation

区間	関数	Hilbert 変換
$[a, b]$	1	$\log \left(\frac{z-a}{z-b} \right)$
$[0, 1]$	$\frac{1}{\sqrt{x}}$	$\frac{1}{\sqrt{z}} \log \left(\frac{\sqrt{z}+1}{\sqrt{z}-1} \right)$
$[0, 1]$	$x^a(1-x)^b$	$\frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)} \frac{1}{z} F \left(a+1, 1; a+b+2; \frac{1}{z} \right)$
$[0, 1]$	$\log x$	$-\sum_{k=1}^{\infty} \frac{1}{k^2 z^k}$

ここで、 $\Gamma(x)$ はガンマ関数、 $F(a, b; c; x)$ は Gauss の超幾何関数である。

議論を単純化するために、次のような変数変換を行う。 $z = w + \frac{b+a}{2}$ と置くと、積分は w の積分になり積分区間は $[-d, d]$ ($d = \frac{b-a}{2}$) となる。さらに、ジューコフスキー (Joukowski) 変換

$$w = \frac{1}{2} \left(u + \frac{d^2}{u} \right)$$

を行うと、元の積分区間は、原点を中心とする半径 d の円になる。このような変換を行えば、特異点がどこにあっても、常に積分路を原点を中心とする円にとることができる [3][8][9]。

本論文では、このように変形された積分は、高精度で計算でき、どの程度の速さで収束するか計算できる。収束の速さを高める方法として、積分区間を分割する方法を提案する。この手法を使えば、これまで多くの自動積分法で 2~3 桁程度しか計算出来なかった Kahaner[5] の 21 番目の問題も容易に 12 桁以上の高精度で計算可能である。

2. 積分の収束率

式 (2) のように変形され、平行移動や Joukowski 変換がなされた被積分関数を $F(z)$ とする。この関数は、次のようにローラン (Laurent) 展開できるものとする

$$F(z) = \sum_{k=0}^{\infty} a_k z^k + \sum_{k=1}^{\infty} b_k z^{-k}$$

この関数を次のように複素平面上で 1 周期に渡る積分を計算することを考える。

$$\begin{aligned}
 I &= \frac{1}{2\pi i} \int_c F(z) dz \\
 &= \frac{1}{2\pi i} \int_c \sum_{k=0}^{\infty} a_k z^k dz + \frac{1}{2\pi i} \int_c \sum_{k=1}^{\infty} b_k z^{-k} dz
 \end{aligned} \tag{3}$$

(3) の第 1 項の積分 J は、原点を中心とした半径 r の円を積分路 c にとり、 N 等分に分割し台形公式を使って計算する。その値を J_N とすると

$$J_N = \frac{1}{N} \sum_{k=0}^{\infty} a_k r^{k+1} \left(\sum_{j=0}^{N-1} e^{i(k+1)hj} \right) \tag{4}$$

ここで、 $h = 2\pi/N$ である。この式の括弧 $()$ の部分 S は、等比級数なので、容易に計算出来る。 $k+1$ が N の倍数になると、すべての項が 1 となり和は N となる。 $k+1$ が N の倍数でない場合、等比級数として和を計算すると分子が零になることから 0 となる。すなわち、次のようになる。

$$S = \frac{1 - e^{i(k+1)hN}}{1 - e^{i(k+1)h}} = \begin{cases} N & \text{mod}(k+1, N) = 0 \\ 0 & \text{mod}(k+1, N) \neq 0 \end{cases}$$

したがって、次の式が得られる。

$$J_N = \sum_{k=0}^{\infty} a_{kN-1} r^{kN}$$

ここで、

$$a_m = \frac{1}{2\pi i} \int_{c_1} \frac{F(z)}{z^m} dz$$

であるから、

$$\begin{aligned}
 J_N &= \frac{1}{2\pi i} \int_{c_1} F(z) \sum_{k=1}^{\infty} \left(\frac{r}{z} \right)^{kN} dz \\
 &= \frac{1}{2\pi i} \int_{c_1} \frac{F(z) \left(\frac{r}{z} \right)^N}{1 - \left(\frac{r}{z} \right)^N} dz
 \end{aligned} \tag{5}$$

この積分値は正則関数 $F(z)$ の積分だから、コーシー (Cauchy) の積分定理により零である。したがって、(5) の J_N が (3) の 1 項から生じる誤差となる。

(5) の積分を評価するために、積分路 c_1 として原点を中心とする半径 R_1 の円周上の積分を考える。半径 R_1 と半径 r は自由に選べるから、 $r < R_1$ とし、半径 r の円周上の $F(z)$ の絶対値の最大値を M_1 とすると次のように評価できる。

$$|J_N| \leq \frac{\left(\frac{r}{R_1}\right)^N}{1 - \left(\frac{r}{R_1}\right)^N} M_1 < M_1 \left(\frac{r}{R_1}\right)^N$$

R_1 が大きければ、この誤差は、小さく出来る。 R_1 は、原点に最も近い特異点までの距離にとると最大のになる。

同様に式 (3) の第 2 項の積分から $k=1$ の項を除いた積分

$$K_N = \frac{1}{2\pi i} \int_c \sum_{k=2}^{\infty} b_k z^{-k} dz$$

も評価できる。 $k=1$ の項の積分値は、 b_1 となるので、その項を除いて評価すると誤差評価になる。半径 R_2 上での関数 $F(z)$ の最大値を M_2 とすると

$$|K_N| < M_2 \left(\frac{R_2}{r}\right)^N$$

R_2 が小さくすれば、この誤差も、小さく出来る。 R_2 は、原点からの積分区間までの距離 e になった時最小のになる。したがって、誤差 Er は、次のようになる。

$$Er = |J_N| + |K_N| = M_1(r/R_1)^N + M_2(R_2/r)^N$$

誤差 Er が最小になるのは、相加相乗平均の式を適用すると、第 1 項と第 2 項の式が一致した場合である。これから、

$$r = \sqrt[2N]{\frac{M_2}{M_1}} \sqrt{R_1 R_2}$$

のとき、誤差が最小になる。 N が十分大きいとき

$$r = \sqrt{R_1 R_2}$$

となる。このとき、誤差 Er は次のようになる。

$$Er = (M_1 + M_2) \left(\frac{R_2}{R_1}\right)^{\frac{N}{2}}$$

通常は Joukowski 変換で半径 1 の円に変換するので、 $R_2 = 1 (e = 1)$ となる。 R_1 はこのように変換したときも原点から最も近い特異点までの距離となる。したがって、誤差 Er は、

$$Er = (M_1 + M_2) e^{-\left(\frac{1}{2} \log R_1\right) N} = O(e^{-qN})$$

となる。この $q = \frac{1}{2} \log R_1$ を収束率と定義する。収束率 q が大きいほど、速く収束する。 q は特異点が積分区間に近いほど小さくなる。積分区間が広がるほど、相対的に特異点は原点に近づくため、小さくなり、収束が悪くなる。式 (1) のように与えられた積分の収束率 q は、C 言語風で書くと、次のような手順で計算出来る。与えられた被積分関数 $f(x)$ の特異点を複素数 z とし、積分区間を $[a, b]$ とする。

```
double q( complex z, double a, double b )
{
    complex u, v ;
    double w ;
```

```
    u=2/(b-a)*(z-(a+b)/2) ;
    v=sqrt(u*u-1) ;
    w=max(abs(u+v),abs(u-v)) ;
    return log(w) ;
```

}

q が求める収束率である。この計算を行う関数を $q(z, a, b)$ とする。

複数の特異点がある場合、その特異点は m 個あり、それを z_i ($i = 1, \dots, m$) とする。この場合、積分区間 $[a, b]$ の収束率 q_m は、特異点毎に収束率を計算し、その最小値が収束率になるので、次の式で与えられる。

$$q_m = \min(q(z_1, a, b), q(z_2, a, b), \dots, q(z_m, a, b)) \quad (6)$$

3. 積分区間の分割

積分区間を分割する前の収束率を q 、標本点数を N とする。積分区間を 2 分割したとき、それぞれの区間積分の収束率を q_1 および q_2 とし、それぞれの区間の標本点数を N_1 、 N_2 とする。

$$q_1 N_1 + q_2 N_2 > qN \quad (7)$$

であり、かつ

$$N_1 + N_2 < N \quad (8)$$

が成り立つならば、分割数が少なく、精度の良い結果が得られると期待できる。3 以上の分割も同様に、求められる。

上の式からわかるように、積分公式の誤差が $O(e^{-qN})$ である公式が対象となる。すなわち、ガウスの数値積分やここで扱った周回積分法などが対象となる。この収束率が分割によって非常に効率的になる場合は、多くの数値積分公式でも有効である場合が多い。実際この分割を、決定するには、 $q_1 = q_2$ と仮定して、区間分割点の位置を求め、(7) と (8) の条件を満たすかどうかを判定する。この計算は、一般に連立非線形方程式となる。

4. 数値例

積分区間の分割の方法は、複素平面上の周回積分だけでなく、ガウス型数値積分でも成り立つ。ガウス・ルジャンドル型数値積分公式では誤差は、標本点数の指数数的に減少し、また、扱う問題では積分区間の端点を含めて特異点がないので、積分区間分割に対してほぼ同様な性質を持っている。

ここでは、ガウス・ルジャンドル型数値積分公式を利用して積分区間の分割の有効性を述べる。また、二重指数型数値積分法の有効性についても述べる。

4.1 簡単な数値例

次のような積分を考える。

$$I_1 = \int_{-1}^1 \frac{dx}{x^2 + \frac{1}{100}} = 20 \tan^{-1} 10 = 29.422553 \dots \quad (9)$$

この被積分関数を図2に示す。x=0で鋭いピークを持つ関数である。よく知られているように、このような積分は、[-1, 0]と[0, 1]と分割すると効率的に計算出来る。実際収束率を計算すると区間[0, 1]の積分の収束率 $q(0.1i, 0, 1)$ は、0.4543となり、区間[-1, 1]の収束率 $q(0.1i, -1, 1)$ は、0.0998となった。収束率が約4.5倍で2倍以上であるから、積分区間を2分割すれば、効率的に計算可能と思われる。これを計算した結果を表2に示した。

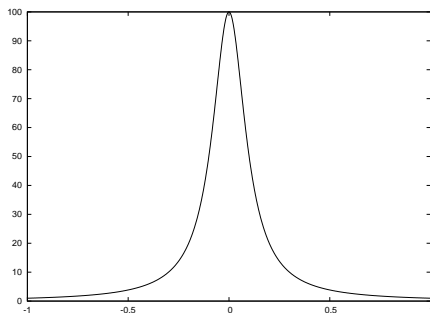


図2 $\frac{1}{x^2 + \frac{1}{100}}$ のグラフ

表2 Integration of I_1 by 2 division

全標本 点数	非分割計算	非分割 相対誤差
10	22.560912728363004	2.33e-01
20	28.392588485307613	3.50e-02
40	29.403241313016732	6.56e-04
80	29.422546916974156	2.23e-07
全標本 点数	2分割計算	2分割 相対誤差
10	30.227737074337142	2.74e-02
20	29.420645049898184	6.49e-05
40	29.422554238301103	2.56e-08
80	29.422553486074690	1.21e-16

3分割する場合、区間[-1, 1]を[-1, c1], [c1, c2], [c2, 1]と3分割し、3つの区間で収束率を同じになるように、c1と2を選ぶ。すなわち、次の連立方程式を解く。

$$q(0.1i, -1, c1) = q(0.1i, c1, c2) = q(0.1i, c2, 1)$$

この方程式を解くと $c1 = -0.117428745$ となる。対称性から $c2 = 0.11742874$ となる。3つの区間で収束率は0.77244で、区間[-1, 1]の収束率の約7.7倍となり3倍以上であるから、分割して計算するのは意味があると思われる。実際計算した結果を表3に示した。

ガウスの数値積分公式を使って計算したものが非分割計算で、標本点数の半分の標本点数のガウスの数値積分公式を使って、2分割した部分の積分値を計算し加算したものが2分割計算である。たとえば、全標本点数40の場合、

表3 Integration of I_1 by 3 division

全標本点数 N	3分割誤差	非分割誤差
15	2.65e-04	7.90e-01
30	3.12e-08	4.82e-03
60	3.63e-14	1.21e-05
90	2.41e-16	3.03e-08

区間[-1, 1]の積分を40点のガウスの数値積分で計算したものが非分割計算で、20点のガウスの数値積分法を区間[-1, 0]と[0, 1]に適用して計算したのが2分割計算である。誤差は相対誤差である。ガウス型数値積分公式の分点と重みはDavis[1]にしたがって、多倍長[4]で計算した。

3分割する場合、区間[-1, 1]を[-1, c1], [c1, c2], [c2, 1]と3分割し、3つの区間で収束率を同じになるように、c1と2を選ぶ。この条件でc1を選ぶと $c1 = 0.117428745$ となる。対称性から $c2 = 0.11742874$ となる。3つの区間で集束率は0.77244で、区間[-1, 1]の収束率の約7.7倍となり3倍以上であるから、分割して計算するのは意味があると思われる。積分区間を2分割した場合の収束率は0.4543であるから約1.7倍である。標本点数は1.5になるから、3分割の計算は、2分割の計算と比較するとわずかに良い程度である。実際計算した結果を表3に示した。

この問題を二重指数型数値積分法[10]で要求精度 $1.0e-10$ で計算すると、分割なしの時標本点数551、誤差 $4.45e-11$ となり、原点で2分割すると、標本点数の合計が150、誤差が $1.11e-12$ であった。この問題に対しては、2分割は二重指数型数値積分法にも有効であることがわかる。

上の例題と同じように、積分区間を3分割し、要求精度 $1.0e-10$ で計算すると、標本点数の合計が223で、誤差が $2.30e-12$ となった。標本点数が増加し、わずかではあるが誤差が増加している。この問題に対して、3分割は2分割より効果がないことがわかる。

二重指数型数値積分公式などの自動数値積分法は、積分の区間の端点およびその付近にある特異性は考慮されているが、積分区間の中間点付近にある特異点は考慮されていない。中間点付近にある特異点は、その中間点で2分割することによって、積分区間の端点付近の特異性となるため、効率的に計算できる。

4.2 非対称な数値積分

積分のピークが積分区間の中央にある場合、中点で2分割するのは、ほぼ自明であるように思える。ここでは、ピークが積分区間の端点にある場合も分割によって、効率的に計算できることを示す。

次の積分を考える。

$$I_2 = \int_0^1 \frac{dx}{x + \frac{1}{100}} = \log 101 = 4.61512051 \dots \quad (10)$$

を計算する。このような積分は、積分区間を分割して計算することは通常ないが、以下のように分割することによ

て、効率的に計算できることがわかる。この積分を2分割して積分する。この場合、分割点 $c = 0.0904987475$ となる。この分割点を使って分割計算すると表4のようになる。分割された区間の積分の収束率 0.6531548292、全区間の収束率は 0.1996681578 となり約 3.2 倍で2倍以上なので、少し効率的に計算できると推定できる。

表4 Integration of I_2 by 2 division

全本点数 N	2分割誤差	非分割誤差
10	1.90e-03	1.72e-02
20	2.88e-06	3.55e-04
40	6.25e-12	1.25e-07
80	1.92e-16	1.50e-14

この積分を3分割して積分する。この場合、分割点 $c_1 = 0.0365700877$, $c_2 = 0.2068773911$ となる。これを計算すると表5のようになる。分割された区間の積分の収束率 1.0035723982、全区間の収束率は 0.1996681578 となり約 5.0 倍で3倍以上なので、効率的に計算できると推定できる。

表5 Integration of I_2 by 3 division

全本点数 N	3分割誤差	非分割誤差
15	6.20e-05	4.05e-02
30	2.80e-09	6.70e-06
60	0.00e-12	4.29e-11
90	1.92e-16	5.77e-16

この積分は積分区間の端点近くに特異点がある。多くの数値積分公式は、積分区間の端点における特異性があっても効率的に計算できるので、このような分割は通常不要である。

ここで使ったガウス型数値積分公式は、積分区間の端点における特異性はあまり考慮していない公式なので分割計算は有効である。

この問題を二重指数型数値積分法で計算すると、分割なしの時標本点数 75、誤差 $7.03e-13$ となり、 $x=0.0904987475$ で2分割すると、標本点数の合計が 150、誤差が $1.39e-13$ であった。この問題に対しては、2分割は二重指数型数値積分法には有効でないことがわかる。 $x=0.0365700877$ と $x=0.2068773911$ で積分区間を3分割し、要求精度 $1.0e-10$ で計算すると、標本点数の合計が 111 で、誤差が $1.72e-13$ となった。この問題に対しては、2分割でも3分割しても効果がないことがわかる。

4.3 特異点が積分区間に近接した積分

Kahaner[5] の21番目の数値積分問題を考える。この問題は多くの自動数値積分法のテスト問題で、これまでの自動数値積分法では5桁以上の精度で計算出来なかった問題である。この積分は、

$$I_3 = \int_0^1 f_1(x) + f_2(x) + f_3(x) dx \quad (11)$$

ここで、

$$f_1(x) = \frac{1}{(\cosh 10(x-0.2))^2}$$

$$f_2(x) = \frac{1}{(\cosh 100(x-0.4))^4}$$

$$f_3(x) = \frac{1}{(\cosh 1000(x-0.6))^6}$$

である。この積分は解析的に積分計算出来て

$$I_3 = 0.210802735500549277375643 \dots$$

が得られる。これを関数のグラフを図3に示す。 $x = 0.4$ と $x = 0.6$ に鋭いピークを持つことから、この近辺に特異点を持つと推定できる。

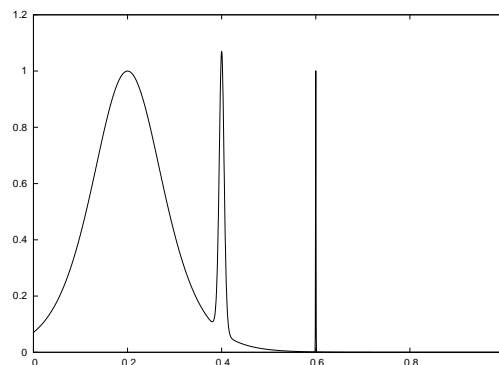


図3 Kahaner の 21 番目の問題のグラフ

$x = 0.4$ 付近では、 $z = 0.4 \pm \frac{\pi}{100}i$, $x = 0.6$ 付近では、 $z = 0.6 \pm \frac{\pi}{1000}i$ の特異点を持っている。特に $x=0.6$ 近くの特異点の位置から収束率を計算すると 0.0064126997 が得られる。数値例1では収束率が約 0.1 でも分割が有効だったので、この収束率では当然分割すべきである。したがって積分は、区間 $[0, 0.6]$ と $[0.6, 1]$ の二つに分割して計算する。

これまでのように収束率が分割された区間で同じになるように分割すると分割点は 0.599358 となるが、このように特異点が積分路に近い場合は、特異点に最も近い積分路上の点で分割することは、非常に良い近似になる。

積分区間 $[0, 0.6]$ の間にも積分区間に近い特異点がある。これを使って収束率を計算すると 0.110761 となる。ここでも分割することになると、積分は区間 $[0, 0.4]$, $[0.4, 0.6]$, $[0.6, 1]$ と分割する。このように積分区間を分割し、収束率を計算すると 0.388969 となり、0.4 は十分精度の良い分割点の近似になっていることがわかる。

このように分割してから二重指数型積分公式を使って計算すると、分割なしの場合、標本点数 2275 で、誤差 $8.35e-04$ で約 3 桁の精度が得られるが、全く無力である。上のように積分区間を3分割すると、総標本点数 617 で、誤差は $2.68e-14$ で約 13 桁の精度で計算できる。

同じように3分割された積分に96点のガウスの数値積分公式を適用すると、分点数は 288 で、誤差が $1.351e-07$ で約 6 桁の精度が得られた。3個の分点 (0.399900207288915,

0.596203536399132, 0.607822949897239) で4分割し、それぞれの積分に96点のガウスの数値積分公式を適用すると、分点数384で誤差が $-8.32e-17$ となり、高精度の結果が得られる。この問題は、積分区間に近い特異点はあるが、積分端点の近くに特異点がないので積分区間分割によって、高精度で計算できることがわかる。

この計算結果は、日比野等 [2] の計算 (AQE11D) 結果と比較すると、この問題に関しては非常に高精度の結果であることがわかる。日比野等では、要求精度 $1.0e-9$ の時、標本点数147、誤差 $1.1e-03$ である。約2~3桁程度の結果である。

日比野等で比較対象になっている AQNN9[7] や DQXG2 のプログラムでも同様である。

4.4 積分区間とその端点付近に特異点がある積分

宮広等 [6] によると、次の積分

$$I_4 = \int_0^1 \frac{-dx}{x^5 - x^4 - 0.75x^3 + x^2 - 0.25x - 10^{-6}} \quad (12)$$

$$= 5195.2449734453507030173478765019311 \dots$$

は、図4のように、積分区間に近いところに特異点があり、この図ではわからないが、積分区間端点の近くに特異点がある問題である。この積分に多くの数値積分法 (シンプソン、ロンバーグ、ガウス型、二重指数型、適応型ニュートン・コーツ) をそのまま適用すると精度の良い結果が得られない。

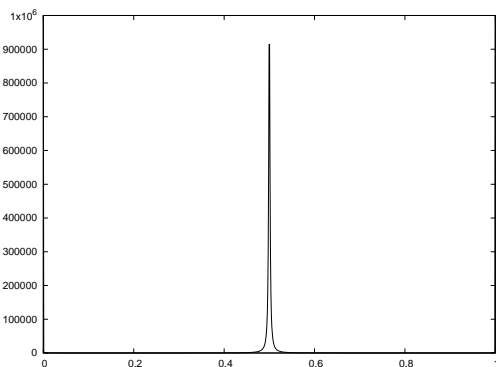


図4 宮広等の問題のグラフ

I_4 の特異点を求めると、次の5点を得られる。

$$x_1 = -3.99993600166395 \times 10^{-6}$$

$$x_2 = -9.9999778061344 \times 10^{-1}$$

$$x_3 = 5.0000754308532 \times 10^{-1} + 1.6343937579178 \times 10^{-3}i$$

$$x_4 = 5.0000754308532 \times 10^{-1} - 1.6343937579178 \times 10^{-3}i$$

$$x_5 = 1.00000200834555$$

x_3 と x_4 は、積分区間内のピークの原因の特異点である。0.5で積分区間をすべきであることは容易に推定できる。このように分割すると収束率は、 $3.27e-3$ から $8.08e-2$ と約

24倍効率的になる。このように分割しても、 x_1 と x_5 の積分区間端点付近の特異性は除去できない。このように分割すれば、積分区間に特異点がないので、二重指数型数値積分法が効率的に計算出来ると期待できる。この積分を二重指数型数値積分法で計算すると、分割なしの時、標本点数2393で、相対誤差 $1.1e-2$ であったが、2分割することによって、標本点数314で、相対誤差 $4.2e-13$ であった。一見難しそうな積分も2分割することによって、簡単に計算できることがわかる。 i_4 の数値も、2分割で高精度計算したものである。標本点数1730、要求精度 $1.0e-50$ で高精度計算すると誤差が $1.3e-44$ であった。

積分 I_2 のように積分区間の端点付近に特異点がある場合、 I_2 のように、端点に近くない場合は、区間分割法によって効率的に求めることができるが、積分 I_4 のように非常に近くにある場合、非常に多くの分割が必要となるため、区間分割法の実用性はあまりないように思われる。

5. まとめ

積分区間の近くに被積分関数の特異点があるような条件の悪い積分を計算する場合積分区間を分割する本方法が非常に有効である場合があることを示した。特異点と積分区間が決まれば積分値の計算効率を示す収束率を定義した。これは、ガウス型積分公式の収束率であるが、この収束率が大きくなるように積分区間を分割すれば、多くの数値積分公式でも有効である。この積分区間分割法では、これまでの数値積分法では実際上不可能な数値積分も容易に計算が可能である場合があることを示した。ここでは、そのような例として Kahner の 21 番の問題および宮広の問題を挙げた。

本計算では、積分路付近の特異点の位置が必要である。このためには、被積分関数の微分係数の零点を求める必要がある。この部分を拡充することがこれからの目標となる。

参考文献

- [1] Davis P.J., Rabinowitz, (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] 日比野, 長谷川, 二宮, 細田, 佐藤, 二宮法と FLR 法の結合による新しい適応型積分法, 情報処理学会論文誌, 44(2003)2419-2427
- [3] 平山 弘, 周回積分変換法による数値積分法, 第 44 回数値解析シンポジウム, (2015)21-24
- [4] 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, 5(1995)123-134
- [5] Kahaner, D.K., Comparison of numerical quadrature formulas, Mathematical Software, Rice, J.R.(Ed.), Academic Press, (1971)229-259
- [6] 宮広, 野田, 新しい有理関数近似によるハイブリッド積分の拡張について, 応用数学会論文誌, 2(1992)193-206
- [7] 二宮市三, 適応型ニュートン・コーツ積分法の改良, 情報処理学会論文, 21(1980)505-513
- [8] 緒方, 平山, 数値積分に対する超関数法, 応用数学会論文誌, 26(2016)33-43

- [9] Ogata H., Hirayama H., Numerical integration based on hyperfunction theory, JCAM, 327(2017) 243–259
- [10] Takahasi, H. and Mori, M., Double exponential formula for numerical integration, Publ. RIMS, Kyoto Univ., 9(1974) 121–141