積分区間分割による数値積分の効率化

4B−3

平山　弘

神奈川工科大学

1．はじめに

数値積分法の研究[1]は、古くから行なわれており、多数の研究者によって、非常に多くの研究成果を得らている。積分区間の近くに特異点がある場合、数値積分の収束性が悪くなることは良く知られた事実である。両端点における特異性をどの様に除去し、いかに効率良く計算するかは数値積分法の主目的である。このことは、多くの数値積分公式では積分区間の両端点付近で多数の標本点を使うことからも容易に想像出来る。積分区間の両端点付近では、一方向からしか関数の情報を得ることが出来ないため、端点付近では多数の標本点をとり、その関数の正確な情報を得て積分の値を計算している。これらの計算法では、特異点が端点付近だけにある場合には大変有効であるが、積分区間の中央部にある場合には、あまり有効でない。例えば、次のような積分を考えて見る。

$$\int_{-10}^{10} \frac{1}{1+100x^2}dx \qquad (1.1)$$

この積分は、被積分関数自体は積分区間内で通常の意味では、特異性をもたない。しかし、この積分を数値積分を行なうと非常に精度が悪いことが分かる。標本点40個を使うガウス型数値積分法で計算しても数値は1桁も合わない。この現象はガウスの数値積分法に限らず、多くの数値積分法に対して、同様な現象が起きる。この種の積分にガウス型数値積分法や二重指数関数型数値積分法を直接適用する事は、効率的な計算法とは言えない。

このような積分の場合、積分区間の分割が非常に有効である。積分(1.1)の積分区間[-10,10]を[-10,0]と[0,10]の二つの区間に分割し、それぞれにガウス型数値積分法とか二重指数関数型数値積分法を適用する方法である。この方法に従って、(1.1)の積分を数値積分を行なうと、分割された二つの区間で使う標本点の総数が、直接数値積分法を行なった場合の標本点数より少なくてもかなり良い結果得ることが出来る。

本論文では、第一に、上の例の様に積分区間の端点以外にある、数値積分の精度を悪くする要因を調べ、数値積分の値の求め易さを示す数値を提案する。長い積分区間、被積分関数の積分路に近くにある特異点の存在が数値積分を求め難くする原因であることを示す。長い積分区間、積分路の近くにある特異点は独立な原因ではなく、積分区間の長さに対して、どの程度特異点が近くにあるかが問題である。大まかに言い方をすれば、数値積分の値の求め易さは、

$$\frac{\text{積分区間と特異点までの距離}}{\text{積分区間の長さ}} \qquad (1.2)$$

と表現出来る。

第二に、数値積分を求め易くするための方法を上の結果から論じる。積分区間の短縮化、すなわち、積分区間の分割について論じる。

2．数値積分の求め易さ

ここでは、次のような積分を考える。

$$I=\int_a^b F(x)\,dx \qquad (2.1)$$

この積分は、次のような変数変換(2.2)を行うことによって、区間[-1,1]の積分に変換できる。

$$x=\frac{a+b}{2}+\frac{b-a}{2}t \qquad (2.2)$$

(2.1)の積分を(2.2)の変数変換を行い、変数 t を x と置き換えると、一般に次のような積分になる。

$$I=\int_{-1}^{1} f(x)\,dx \qquad (2.3)$$

(2.3)の積分にガウス型数値積分公式を適用すると誤差[1]は、標本点の数が大きいとき、漸近的に

$$\text{誤差}=c\rho^{-2n-1} \qquad (2.4)$$

となる。ここで、c は定数、

$$\rho=\min_i |z_i+\sqrt{z_i^2-1}| \qquad (2.5)$$

i はすべての特異点に対して行う。最小になるのは、おおまかな言い方をすれば、積分区間に最も近い特異点の場合である。(2.4)の評価式は、ルジャンドル・ガウス型やチェビシェフ・ガウス型積分公式などでもまったく同じ形になる。このように考えると、(2.5)の式の ρ は積分の計算し易い程度を表す数字と考えられる。ここでは、積分の計算し易さとして、ρ の対数を考える。すなわち、計算し易さAを次のように定義する。

$$A=\log \rho \qquad (2.6)$$

この数値をここでは収束率と呼ぶことにする。ρ は幾何学的には、焦点(1,0)、(-1,0)を持ち特異点を通る楕円の長半径と短半径の和である。この意味を考えると、収束率を大きくするには、特異点から積分区間に垂線を下ろ

Improvement of Numerical Integration by Division of Integral Interval

Hiroshi HIRAYAMA

Kanagawa Institute of Technology

した点で分割すれば良いことがわかる。また、容易にわかるように、二分割、三分割をした場合、Aは二倍、三倍以上の大きさになれば、その分割は計算効率を上げるために有効であることがわかる。

3．収束率の限界

2．で収束率を定義したが、この数が大きいときは、確かに計算を非常に効率よく計算出来る場合が多いが、たとえ収束率が大きくても精度を得るのが困難な問題もある。たとえば、

$$I=\int_0^1 x+x^{1000}dx \qquad (3.1)$$

のような積分である。この積分は、収束率は非常に大きいので、非常に効率よく計算出来ると期待できるが、被積分関数をみれば容易にわかる様に標本点を１０００個程度の標本点を取らなければ、この積分の値は正確には求められない。　収束率は、この様な問題の場合を考えると、計算のしやすさと言うより、高次の積分公式の適用しやすさと言えば、より正確な言い方となる。

また、(2.1)の式から分かるように、(2.1)式のconstが大きい時、確かに収束率が大きければ、計算は効果的に出来るが、通常の計算出来る範囲では、誤差はなかなか小さくならない場合もある。このような時も、収束率が示す数値はあまり意味がないことになる。

4．数値例

上の理論を確認するために、ガウス型数値積分法、二重指数関数型積分法で積分を実行した。

例4．1

$$I=\int_{-10}^{10}\frac{dx}{1+x^2} \qquad (4.1)$$

を計算してみよう。積分区間を分割しない場合と２分割した場合、標本点Nに対する誤差を表４．１のようになる。２分割した場合の標本点の数は、それぞれの分割数の合計である。上の例から分かるようにこの問題では二分割した方が効率よく計算できることがわかる。この問題の収束率は、分割しない時は

A=0.099834

分割した時の収束率は

A=0.45431

となり、かなり積分しやすさが向上する。定義からもわかるように、分割によって収束率が２倍以上にならない場合は、分割しない方が能率的に計算できることになる。上の問題の場合、積分効率は、約４．５になっているので、分割する意味がある場合に対応する。ａはガウス積分、ｂは二重指数関数積分で計算した結果である。２分割の標本点は２つの区間で使われた標本点の合計である。二重指数関数型数値積分の標本点数は、２分割の場合、最大で３個程度の標本点数が異なっている場合がある。

表４．１－ａ
ガウス型数値積分法

N	分割なし	２分割
4	1.8111E+00	
8	9.7046E-01	2.7020E-01
16	2.2436E-01	9.8108E-03
32	9.5270E-03	7.1372E-06
64	1.6029E-05	1.0250E-12
128	4.5207E-11	1.4743E-13

表４．１－ｂ
二重指数関数型数値積分法

N	分割なし	２分割
17	5.1466E-00	
33	1.5979E-00	5.8051E-02
61	2.6936E-01	2.6443E-04
115	1.0636E-02	5.0261E-09
223	1.7943E-05	8.8818E-16
435	5.1243E-11	
855	2.6645E-15	

5．結論

積分区間の両端点の特異点の問題が解決した数値積分のアルゴリズムでは、つぎに問題になるのが積分区間の近くにある特異点の問題である。

本論文では、この特異点の問題の難しさを示す指標（収束率）を提案した。この収束率は、数値積分の実行しやすさを示す指標として十分に役立つだけでなく、積分を計算しやすい積分に変換するのに役立つ。とくに、積分区間の分割や特異点除去などの指針として、かなり有効である。

また、逆に特異点が多数ある場合、どの特異点が積分の計算をやりにくくしているかを容易に見いだすことができ、それに対処する方法を与える。

6．参考文献

1) P.J.Davis,P.Rabinowitz：
Methods of Numerical Integration,
ACADEMIC PRESS(1975)
2) 森　正武：数値解析と複素関数論，
築摩書房(1975)
3) 平山　弘：数値積分の計算効率向上方法，
情報処理学会数値解析研究会30-6(1987)