

佳作論文

適応型ニュートン・コーツ積分法の改良†

二宮 市三††

ニュートン・コーツ則に基づく適応型自動積分法に、次の三つの改良を行った。

- (1) 誤差推定の強化
- (2) 収束判定の緩和
- (3) 異常点の検出処理能力の付与

これらの改良により、積分法の信頼性、経済性および融通性が著しく向上した。

改良案をとり入れた、適応型9点積分ルーチン DAQN9 の内容を説明し、著名積分ルーチンとの性能比較の結果を示した。

1. ま え が き

被積分関数 $f(x)$ 、積分区間 $[a, b]$ 、許容誤差 ϵ_0 を与えられたとき、積分区間の分割と分点の決定を自動的に行って、

$$\left| \int_a^b f(x) dx - S \right| < \epsilon_0$$

を満足する近似値 S を計算する算法を、自動積分法という。自動積分法には、大域型と適応型の二つの型がある。

大域型に属するものとしては、ロンバーグ法が典型的であるが、最近ではクレンショー・カーチス法^{1),2)}や、高橋・森³⁾の二重指数関数型積分法が、より有効な方法として有名である。これらの積分法は、被積分関数の性質を大域的にとらえ、その局所的な変化に無関係に、あらかじめ定められた方式に従って、分点を積分区間全体にわたって一様に増大——多くの場合倍増——していく。

これに対して適応型としては、シンプソン則を始めとするニュートン・コーツ則に基づくものが主流をなしている^{4),5)}が、それらの共通の目標は、関数の性質を局所的にとらえ、その変化の緩急に分点の密度を適応させることによって、計算効率の向上を図ることである。

本論文において、著者は適応型積分法の長所を進展させ、その融通性を飛躍的に増進するために、三つの改良案を提出する。

(1) 誤差推定の強化

ある区間に、ニュートン・コーツ $n+1$ 点則を適用

してえられる近似値 S の誤差を推定するために、従来から用いられている方法は、その区間を左右に二等分し、それぞれの半区間に同じ積分則を施してえられる近似値 S' と S の差を考慮することである。そのためには、もとの区間の n 個の $2n$ 等分点での関数値を新たに計算することが必要である。

これに対して、新提案によれば、同じ誤差推定の目的を達成するには、二個の関数値を追加するだけで十分であることが示される。

(2) 収束判定の緩和

適応型積分法では、各小部分区間ごとに別々に収束判定を行わなければならない。そのときに使用する誤差規準は、積分全体に対して与えられた誤差規準 ϵ_0 に応じて、どのように定められるべきであろうか。 ϵ_0 を部分区間の長さ按比例して分配する今までの方法は、自然ではあるが保守的に過ぎるきらいがある。これに対して、部分区間で誤差規準を積極的に緩和する一方法を提案する。

(3) 異常点の検出処理能力の付与

科学技術計算に実際に現れる積分は、必ずしも解析関数の積分だけではない。ときには不連続点や特異点を含む関数の積分が要求されることがあり、自動積分法がその処理能力をそなえていることは、非常に望ましいことである。大域型では、二重指数関数型積分法³⁾が、区間の端にある特異点を処理するのに有効であることが知られている。一方適応型では de Boor⁶⁾が、ロンバーグ法による適応型ルーチン CADRE に、そのような能力を与えた。著者は、ニュートン・コーツ則による適応型積分法においても、同様なことが可能であることを明らかにする。

著者⁷⁾は以上の三つの改良案をとり入れた、ニュートン・コーツ 5, 7, 9 点則による適応型ルーチン

† Improvement of Adaptive Newton-Cotes Quadrature Methods by ICHIZO NINOMIYA (Department of Information Science, Faculty of Engineering, Nagoya University).

†† 名古屋大学工学部情報工学科

AQNN5D*, AQNN7D*, AQNN9D* を作成し、すでに名古屋大学大型計算機センターのプログラムライブラリに登録している。また、最近 AQNN9D に相対誤差規準の入力、推定誤差の出力などの機能を追加した、より実用的なサブルーチン DAQN9* を書き、富士通の科学サブルーチンライブラリ SSL II[®] に登録している。ここでは、DAQN9 の仕様と算法手順のあらましを説明する。さらに、標準テスト問題として有名な Kahaner⁵⁾ の 21 問題について行った DAQN9 の性能調査の結果を示し、二、三の著名サブルーチンと比較対照する。

2. 誤差推定の強化

ニュートン・コーツ則は、積分区間の等分点を用いた補間型の積分則で、一度計算した関数値を蓄えておいて再利用することが要求される適応型積分法には最も適したものである。特に、区間を偶数等分し、両端を含めて奇数個の分点を用いるものは、分点の数が1だけ多い偶数分点のものとの誤差の次数が同一であるという点で、有利である。ここでは実際の見地から、5点則、7点則および9点則について考えよう。

$2n+1$ 点則に対して、区間の $2n$ 等分点の外に、両端の $4n$ 等分点を加え、図1のように番号をつけよう。積分の厳密値を Q 、誤差を e 、区間の半幅を h 、分点 i における関数値を f_i と記すことにすると、ニュートン・コーツ $2n+1$ 点則 ($n=2, 3, 4$) と本章の後半で導出するその誤差評価式は、次のように表わされる。ただし $f^{(6)}, f^{(8)}, f^{(10)}$ はそれぞれ区間内の適当な点での6次、8次および10次微係数の値である。

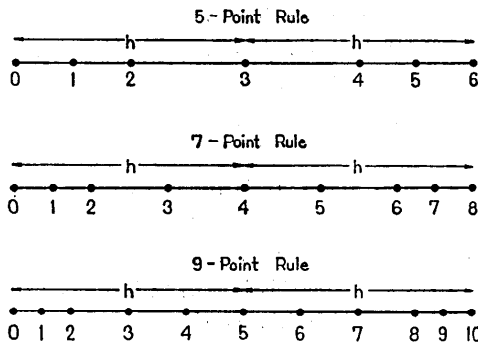


図1 分点の配置

Fig. 1 Arrangement of Sample Points.

* これらは倍精度版である。単精度版はそれぞれ AQNN5S, AQNN7S, AQNN9S および AQN9 である。

(1) 5 点 則

$$Q = \frac{h}{45} \{7(f_0 + f_6) + 32(f_2 + f_4) + 12f_3\} - e, \quad (2.1)$$

$$e = \frac{h^5 f^{(5)}}{15120} \approx \frac{32h}{6615} \{15(f_0 + f_6) - 64(f_1 + f_5) + 84(f_2 + f_4) - 70f_3\}. \quad (2.2)$$

(2) 7 点 則

$$Q = \frac{h}{420} \{41(f_0 + f_8) + 216(f_2 + f_6) + 27(f_3 + f_5) + 272f_4\} - e, \quad (2.3)$$

$$e = \frac{h^7 f^{(7)}}{1020600} \approx \frac{4h}{9625} \{105(f_0 + f_8) - 512(f_1 + f_7) + 770(f_2 + f_6) - 825(f_3 + f_5) + 924f_4\}. \quad (2.4)$$

(3) 9 点 則

$$Q = \frac{h}{14175} \{989(f_0 + f_{10}) + 5888(f_2 + f_8) - 928(f_3 + f_7) + 10496(f_4 + f_6) - 4540f_5\} - e, \quad (2.5)$$

$$e = \frac{37h^{11} f^{(11)}}{30656102400} \approx \frac{4736h}{468242775} \{3003(f_0 + f_{10}) - 16384(f_1 + f_9) + 27720(f_2 + f_8) - 38220(f_3 + f_7) + 56056(f_4 + f_6) - 64350f_5\}. \quad (2.6)$$

誤差 e を推定するための従来の方法は次のとおりである。左端が x で、半幅 h の区間に、ニュートン・コーツ $2n+1$ 点則を適用してえられる近似値を、 $S(x, h)$ としよう。上の諸式により $S(x, h)$ は、

$$S(x, h) = Q + e = Q + ch^{2n+3}$$

の形である。ここに係数 c は定数ではないが、仮りにこれを定数とみなすことにする。区間を二等分し、それぞれの半区間に $2n+1$ 点則を施せば、

$$S(x, h/2) = Q_1 + c(h/2)^{2n+3},$$

$$S(x+h, h/2) = Q_2 + c(h/2)^{2n+3}$$

となる。ただし Q_1, Q_2 はそれぞれ左および右の半区間の積分値である。

$$Q = Q_1 + Q_2$$

であるから、

$$e = ch^{2n+3} \approx \frac{2^{2n+2}}{2^{2n+2}-1} \{S(x, h) - (S(x, h/2) + S(x+h, h/2))\} \quad (2.7)$$

であることがわかる。この方法によって、 $2n+1$ 点則の誤差 e を推定するには、 $2n$ 個の $4n$ 等分点での関数値を新たに計算しなければならない。

さて、(2.7) 式の導出の経過をみれば、この推定法

は誤差の理論式の内容の内の、 e が h^{2n+3} に比例することだけしか利用していないことがわかる。実は e は $h^{2n+3}f^{(2n+2)}$ に比例しているのであるから、 $f^{(2n+2)}$ を直接推定すればよいのである。そのためには $2n+3$ 個の点での関数値があればよい。すなわち、新たに計算すべき関数値は二つで十分である。

この二つの関数値を計算する分点は、関数値の再利用と対称性から、左右対称にとられた、 $4n$ 等分点の対でなければならない。しかしながら、この点対を端に近くとるか、中央に近くとるかの選択がまだ残っている。

これに関しては、端の方に密度が高い、チェビシェフ多項式やルジャンドル多項式の零点を用いた補間が種々の好適な性質を持つことや、後出の区間の端の特異点の処理における有利さを考え合せて、最も端に近い二点を使うこととした。

さて、 $2n+1$ 個の $2n$ 等分点と、二個の $4n$ 等分点での関数値の一次結合としての、 $2n+2$ 次数値微分汎関数を D としよう。 D は明らかに対称であるから、未知の係数は $n+2$ 個である。この係数を定めるには、 D を区間 $[-h, h]$ で、関数 x^{2k} ($k=0, 1, \dots, n+1$)に適用した結果、

$$\begin{cases} D(x^{2k})=0, & k=0, 1, \dots, n \\ D(x^{2n+2})=(2n+2)! \end{cases}$$

を、 $n+2$ 個の係数に関する連立一次方程式として解けばよい。(2.2), (2.4), (2.6)式はこのようにしてえられたものである。実際には、上の連立一次方程式を多倍長演算を用いて精密に解き、解の近似値から容易に看取できる循環構造を利用して、係数を有理数で表わした。

新しい誤差推定法によってもたらされる関数計算の節減を考えよう。いま、旧来の誤差推定法を用いる、積分法Aと、誤差推定法を新方式に改めた以外はすべてAと同一の積分法Bとを、同じ問題に応用した場合を想像しよう。さらに、積分区間のすべての場所で、二つの方法が同じ深さの分割で収束するものと仮定する。そうすれば、積分区間は同じ配置、同じ大きさそして同じ個数 N の小区間に分割され、Aの場合には各小区間に両端を含めて $4n+1$ 個の分点が、Bの場合には $2n+3$ 個の分点が存在することになる。したがって、分点の総数はそれぞれ $4nN+1$, $(2n+2)N+1$ となり、BのAに対する節減率は

$$\frac{(2n+2)N+1}{4nN+1} \approx \frac{n+1}{2n} \quad (2.8)$$

で与えられる。その値は、5点則、7点則、9点則に対してそれぞれ $3/4$, $2/3$, $5/8$ となり、点数の増加とともに増大する。AとBとが同一の収束過程をたどるといふ、上の仮定は厳密には正しくないが、正常な関数の場合には事実に近いと考えられるので、上にえられた節減率は、かなり信頼できる値であるとみてよい。

3. 収束判定の緩和

大域的積分法では、常に積分全体に対する近似値が用意され、これについて収束判定が行われる。一方、適応型の場合には、その成り立ちから、各々の部分区間において別々に収束判定が行われる。このとき、部分区間ごとに用いられる誤差規準は、積分全体に対して与えられた誤差規準 ϵ_0 に対応して、どのように定められるべきであろうか。

自明な解答は比例配分法である。積分区間 $I_0: [a, b]$ の半幅を $h_0=(b-a)/2$ 、部分区間 I_i の半幅を h_i とすると、

$$\epsilon_i = \epsilon_0 h_i / h_0 \quad (3.1)$$

を I_i での誤差規準とするわけである。 I_i での積分値を Q_i 、積分法による近似値を S_i とし、各部分区間で ϵ_i を規準とする収束判定が受け入れられたものとする、全体の誤差 E は、

$$\begin{aligned} |E| &= \left| \sum_{i=1}^N (S_i - Q_i) \right| \leq \sum_{i=1}^N |S_i - Q_i| \leq \sum_{i=1}^N \epsilon_i \\ &= \epsilon_0 \sum_{i=1}^N h_i / h_0 = \epsilon_0 \end{aligned} \quad (3.2)$$

となり、確かに要請、

$$|E| \leq \epsilon_0 \quad (3.3)$$

を満足している。しかしながら、一般的には、(3.2)の二つの不等式は、したがって、不等式(3.3)は、その両辺の差が大きい、鈍い不等式になることが多い。特に S_i が積分則から直接計算される値から、誤差の推定値を差し引いた値である場合には、その傾向が強い。要するに、比例配分法は単純で自然ではあるが、極めて保守的であって積極性に乏しい。

これに対してKahaner⁹⁾は預金法(Banking method)を考えた。各々の部分区間で誤差規準 ϵ_i : (3.1)と誤差の推定値 e_i を考え、 $\epsilon_i - |e_i| > 0$ ならば余剰分を銀行に預け、 $\epsilon_i - |e_i| < 0$ ならば、不足分を銀行から引き出そうというわけである。この方法の実際の効果について、著者は詳かではないが、誤差の推定値というものが、そのような厳しい加法性に耐えられるとは信じ難い。

著者は次のように考える。影響の小さい、短い区間に対しては誤差規準を積極的に緩和して、収束判定を早めに受け入れるようにしても、全体としての精度は結構満足なものであることが多いのではないかと。そこで、(3.1)に代って、

$$\varepsilon_i = \varepsilon_0(h_i/h_0) \log_2(h_0/h_i) \quad (3.4)$$

ととることを提案する。緩和因子として、 $\log_2(h_0/h_i)$ を選ぶ理由は、第一にこれが非常に慎重な緩和法の一つを与えるからであり、第二にそのために要する計算が微小であるからである。ところが、この緩和法を適用した結果についての、解析的な、あるいは統計的な理論を、今のところ持ち合わせていない。したがって、その成否はもっぱら実験結果によって判断する以外にはない。著者の行って来た大量の実験に関していえば、否定的な材料はなく、少なくとも振動型の関数の場合のように、区間全体にわたって細分化が進行するときには、関数計算回数を節減するのに著効があることが認められる。

4. 異常点の検出処理能力の付与

局所的に多項式で良く近似できる関数を積分する場合を考えよう。区間の細分が進むと、誤差 e の理論式に現れる微係数は、次第に定数と考えてもよい状態になる。この段階に達したとき、区間をさらに二分すれば、 e は $1/2^{2n+3}$ になるが、一方緩和因子を省略すれば、誤差規準 ε は $1/2$ になるので、 e/ε は結局 $1/2^{2n+2}$ に減少する。 $2n+1$ 点法の収束の速さを表わす e/ε のこの性質は $\varepsilon = e/h$ によって、より端的に示される。

ところが、区間内に不連続点や特異点が存在する場合にはどうであろうか。まず、無限大を避けるために、対数特異点や位数が負の代数特異点での関数値は、例えば0のような有限値で置き換えられているものとする。このような異常点を含む区間では、奇蹟的な偶然を除いては、収束条件が成立することはありえない。したがって、強制的に途中で停止させないかぎり、二分化が際限なく進行し、その内部または境界に同一の異常点を含み、長さが次第に半減する区間の列が発生する。今、異常点が、適当な正の整数 m に対して、区間の一つの 2^m 等分点に位置している場合だけを考えると、ある段階からは異常点は区間の端に来るに違いない。簡単のために、異常点は原点であって、しかも区間の左端にあるとしても一般性を失わない。

さて、このような区間列 $\{I_i; i=1, 2, \dots\}$ における ε の値の振舞を観察してみよう。区間 I_i での ε の値

を ε_i と書くことにする。 ε は線形汎関数であり、正常な関数に対してはその値は上に述べたように急速に減少するから、区間の長さが短くなるにつれて、 ε_i はついに異常点の性質を純粋に示すようになるであろう。

4.1 異常点の検出

(1) 不連続点

原点に δ だけの跳躍がある関数 f を考えよう。

$$\begin{cases} f(0) = \alpha + \delta \\ f(x) = \alpha, \quad x > 0 \end{cases} \quad (4.1)$$

とすると、明らかに $\varepsilon(\alpha) = 0$ であるから、直ちに

$$\varepsilon_i = c_0 \delta \quad (4.2)$$

となる。ただし、 c_0 は ε での区間の端の関数値の係数である。したがって、数列 $\{\varepsilon_i, i=1, 2, \dots\}$ は定数となる。

(2) 対数特異点

原点の近傍で

$$\begin{cases} f(0) = \delta \\ f(x) = \alpha \log x, \quad x > 0 \end{cases} \quad (4.3)$$

と仮定しよう。 ε_i は適当な $x_j, j=1, 2, \dots, 2n+2$ によって、

$$\varepsilon_i = \sum_{j=1}^{2n+2} c_j f(2^{-j} x_j) \quad (4.4)$$

の形に書けるから、この式に(4.3)を代入し、

$$\varepsilon(1) = \sum_{j=0}^{2n+2} c_j = 0$$

であることを利用すれば、

$$\varepsilon_i = c_0 \delta + \alpha \sum_{j=1}^{2n+2} c_j \log x_j + i \alpha c_0 \log 2 \quad (4.5)$$

となる。すなわち、数列 $\{\varepsilon_i, i=1, 2, \dots\}$ は公差が $\alpha c_0 \log 2$ の等差数列となる。

(3) 代数特異点

原点の付近で、

$$\begin{cases} f(0) = \delta \\ f(x) = \alpha x^p, \quad x > 0 \end{cases} \quad (4.6)$$

と仮定しよう。(4.4)より直ちに

$$\varepsilon_i = c_0 \delta + \alpha 2^{-ip} \sum_{j=1}^{2n+2} c_j x_j^p \quad (4.7)$$

となる。これは $\delta=0$ でなければ、このままでは等比級数ではない。しかし ε_i の差分

$$\Delta \varepsilon_i = \varepsilon_{i+1} - \varepsilon_i$$

をとれば、 $c_0 \delta$ が消えて

$$\Delta \varepsilon_i = \alpha 2^{-ip} (2^{-p} - 1) \sum_{j=1}^{2n+2} c_j x_j^p \quad (4.8)$$

となる。したがって、数列 $\{\Delta \varepsilon_i, i=1, 2, \dots\}$ は公比

2^{-p} の等比数列となる。

以上から、異常点の検出は次のように行うことができる。異常点を共通の端点とし、長さが順次半減する区間列についての δ の値の列 $\{\delta_i, i=1, 2, \dots\}$ を観察する。

(a) 数列が定数 c に収束すれば、端点に

$$\delta = c/c_0 \quad (4.9)$$

だけの跳躍をもつ不連続点が存在する。

(b) 数列が公差 d の等差数列に収束すれば、端点に、主要部の係数が

$$\alpha = d/(c_0 \log 2) \quad (4.10)$$

の対数特異点がある。

(c) δ_i の差分の数列 $\{\Delta\delta_i, i=1, 2, \dots\}$ が公比 r の等比数列に収束すれば、端点には位数が

$$p = -\log_2 r \quad (4.11)$$

の代数特異点がある。

4.2 異常点の処理

異常点の存在が検出されたとき、これを端を含む小区間での積分の近似値は、半ば解析的に次のように計算される。簡単のため、ここでも異常点は原点で区間の左端にあると仮定する。

(1) 不連続点

不連続点での関数値から、検出された跳躍量 δ を差し引いて補正したものを、積分則に代入すればよい。

(2) 対数特異点

特異点 $x=0$ の付近で、被積分関数が

$$f(x) = \alpha \log x + \beta + \gamma x \quad (4.12)$$

と表わされるものとしよう。区間の半幅を h とすると、問題の積分は

$$\int_0^{2h} f(x) dx = 2h \{ \alpha (\log 2h - 1) + \beta + \gamma h \} + O(h^3)$$

となるが、これは区間の midpoint の値

$$f(h) = \alpha \log h + \beta + \gamma h$$

を用いて

$$\int_0^{2h} f(x) dx = 2h \{ f(h) + \alpha (\log 2 - 1) \} + O(h^3) \quad (4.13)$$

として計算することができる。

(3) 代数特異点

特異点 $x=0$ の近傍で、

$$f(x) = \alpha x^p + \beta x^{p+1} + \gamma \quad (4.14)$$

と表わされるものとしよう。ここで位数 $p > -1$ は既知である。小区間 $[0, 2h]$ での積分は

$$\int_0^{2h} f(x) dx = 2h \left\{ \frac{\alpha \cdot (2h)^{p+1}}{p+1} + \frac{\beta \cdot (2h)^{p+2}}{p+2} + \gamma \right\} + O(h^3) \quad (4.15)$$

となる。これを計算するのに、三つの関数値 $f(0)$, $f(h)$, $f(2h)$ を利用しよう。この中で $f(0)$ について一つの問題がある。 $f(0)$ としては、特異点の主要部 αx^p を 0 と置き換えてえられる値、すなわち

$$f(0) = \gamma$$

とすることに決めておけば簡単で都合がよいが、実際問題として、特異点の主要部だけを取り除いた値を計算することは必ずしも容易なことではない。そこで、被積分関数についての予備知識をなるべく仮定しないという自動積分法の建前にしたがって、特異点での関数値に任意の偏りを許すことにする。いいかえれば、特異点と不連続点の共存を許すということである。このようにして、

$$\begin{cases} f(0) = \gamma + \delta \\ f(h) = \alpha h^p + \beta h^{p+1} + \gamma \\ f(2h) = \alpha \cdot (2h)^p + \beta (2h)^{p+1} + \gamma \end{cases} \quad (4.16)$$

となる。偏り δ は、(4.7), (4.8) から

$$\delta = \frac{(2^{-p}-1)\delta_i - \Delta\delta_i}{(2^{-p}-1)c_0} \quad (4.17)$$

として与えられる。(4.16) と (4.17) を使って、(4.15) の積分を計算することは極めて容易なことである。

以上のようにして、異常点を処理することができるが、これには、異常点が積分区間の 2^m 等分点の一つであるという、仮定があることを忘れてはならない。したがって、必要ならば変数変換や積分区間の分割などによって、異常点を区間の端や中央のようなわかりやすい点に移すことが望ましい。また、異常点の処理を精度よく行うためには、その近傍での関数の精度を、桁落ちによる損失から保護することが肝要である。そのためには、なるべくならば異常点を原点にもって来ることが最も賢明である。

5. 適応型自動積分ルーチン DAQN 9

本論文の三つの改良案を組み入れた、ニュートン・コーツ 9 点則に基づく適応型自動積分ルーチン DAQN 9 の仕様と算法のあらましを説明する。詳細は文献 8) を参照のこと。

5.1 機能

被積分関数 $f(x)$, 積分区間 $[a, b]$, 絶対誤差規準 ε_a , 相対誤差規準 ε_r が与えられたとき、

$$\left| S - \int_a^b f(x) dx \right| \leq \max(\varepsilon_a, \varepsilon_r \cdot \left| \int_a^b f(x) dx \right|)$$

を満足する近似値 S を計算する。

5.2 使用法

CALL DAQN 9 (A, B, FUN, AEPS, REPS, NMIN, NMAX, S, ERR, N, ICON)

A 入力. 積分区間の下限 a .
 B 入力. 積分区間の上限 b .
 FUN 入力. 被積分関数 $f(x)$ を計算する
 関数副プログラム.

AEPS 入力. 絶対誤差規準 ϵ_a .

REPS 入力. 相対誤差規準 ϵ_r .

NMIN 入力. 関数計算回数下限.

NMAX 入力. 関数計算回数上限.

S 出力. 積分の近似値.

ERR 出力. S の絶対誤差の推定値.

N 出力. 関数計算回数.

ICON 出力. 計算過程における異常の有無
 を使用者に知らせるコンディション
 コード.

プログラムは FORTRAN-7000 の文法を忠実に守り、互換性を十分意識して書かれている。また、必要な記憶容量も 3,000 語程度であるので、中程度以上のほとんどの計算機で利用可能である。

5.3 計算手順

手順 1.....初期設定

積分区間の左端 x , 半幅 h , 積分の近似値 S の初期値をそれぞれ

$$x=0, \quad h=h_0=(b-a)/2, \quad S=0$$

と定める。

その他の諸変数の初期値と、種々の判定定数の値を定める。

積分区間 $[a, b]$ の、両端を含む 8 等分点と、最も端に近い二つの 16 等分点での、11 個の関数値を計算する。

手順 2.....二等分割と情報の蓄え

現在の部分区間 (左端 x , 半幅 h) を二等分割し、左半分と右半分の 8 等分点の中で、まだ計算されていない 6 個の点での関数値を計算する。

右半分の区間の積分を、9 点則で計算し、8 等分点での関数値、半幅 h などとともにスタックに蓄える。この積分値は、相対誤差規準 ϵ_r に乗せられる全体の積分の近似値 S' の一部分として用いられると同時に、後に右半分が処理されるときにも再利用される。

左半分を処理するために、手順 3へ進む。

手順 3.....収束判定

現在の部分区間の積分値がまだ計算されていない場合 (手順 2 から来た場合) には、これを 9 点則で計算する。

両端の二つの 16 等分点での関数値を計算し、これらと 8 等分点での関数値とを用いた評価式 (2.6) によって打切誤差 e を推定する。

e に対して収束判定を行う。すなわち、

$$|e| \leq \max(\epsilon_a, \epsilon_r |S'|) \cdot (h/h_0) \log_2(h_0/h)$$

が成立するかどうかをしらべる。ただし、 S' は最新の情報を用いて、その都度計算される。

収束判定が合格ならば手順 7へ、不合格ならば手順 4へ進む。

手順 4.....異常点の検査

半幅 h が、 $a, b, \epsilon_a, \epsilon_r$ から定められる許容最大値よりも大きければ、直ちに手順 2へ行く。

半幅 h が、 $a, b, \epsilon_a, \epsilon_r$ から定められる許容最小値よりも小であれば、手順 5へ進む。

現在の部分区間の端になんらかの異常点が存在するかどうかをしらべ、存在が認められれば手順 6へ、さもなければ手順 2へ行く。

手順 5.....異常点の再検査

検出規準を弱めて異常点の存在の再検査を行い、合格ならば手順 6へ進む。

不合格ならば、これ以上の処理を断念し、収束判定不成立のまま現在の区間の近似値を受け入れて、手順 7へ進む。

手順 6.....異常点の処理

検出された異常点の種類と特性値によって決められる解析的な計算によって、現在の区間の積分値を定め、手順 7へ向う。

手順 7.....積分値の積算と情報の取り出し

現在の区間の積分値。すなわち、9 点則による近似値から推定誤差 e を差し引いたもの、または異常点に対する特別の処置による近似値を、現在までの S の値に積算する。

最後に蓄えておいた情報をスタックから取り出して手順 2へもどる。もしもスタックが空であれば、処理を終了する。

6. 数値例

DAQN 9 の性能を調査し、これを既成の著名ルーチンと比較する目的で、積分ルーチンの標準テスト問題として定評のある、Kahaner の 21 問題について数値実験を行った。使用計算機は名古屋大学大型計算

表 1 Kahaner のテスト問題集
Table 1 Kahaner's Test Problems.

NO.	LL	UL	INTEGRAL	INTEGRAND
(1)	0.0	1.0000000	1.7182818285D+00	EXP (X)
(2)	0.0	1.0000000	7.0000000000D-01	AINT (AMIN 1 (X/0.3,1))
(3)	0.0	1.0000000	6.6666666667D-01	SQRT (X)
(4)	-1.00	1.0000000	4.7942822669D-01	0.92*COSH (X)-COS (X)
(5)	-1.00	1.0000000	1.5822329637D+00	1/(X**4+X**2+0.9)
(6)	0.0	1.0000000	4.0000000000D-01	X*SQRT (X)
(7)	0.0	1.0000000	2.0000000000D+00	1/SQRT (X)
(8)	0.0	1.0000000	8.6697298734D-01	1/(X**4+1)
(9)	0.0	1.0000000	1.1547006690D+00	2/(2+SIN (31.4159*X))
(10)	0.0	1.0000000	6.9314718056D-01	1/(1+X)
(11)	0.0	1.0000000	3.7988549304D-01	1/(EXP (X)+1)
(12)	0.0	1.0000000	7.7750463411D-01	X/(EXP (X)-1)
(13)	0.10	1.0000000	9.0986452566D-03	SIN (314.159*X)/(3.14159*X)
(14)	0.0	10.0000000	5.0000021117D-01	SQRT (50)*EXP (-50*3.14159*X*X)
(15)	0.0	10.0000000	1.0000000000D+00	25*EXP (-25*X)
(16)	0.0	10.0000000	4.9936380287D-01	50/3.14159/(2500*X**2+1)
(17)	0.01	1.0000000	1.1213956963D-01	(SIN (50*3.14159*X)/(50*3.14159*X))*2*50
(18)	0.0	3.1415927	8.3867632338D-01	COS (COS (X)+3*SIN (X)+2*COS (2*X)+3*SIN (2*X)+3*COS (3*X))
(19)	0.0	1.0000000	-1.0000000000D+00	ALOG (X)
(20)	-1.00	1.0000000	1.5643964441D+00	1/(X**2+1.005)
(21)	0.0	1.0000000	2.1080273550D-01	1/COSH (10*(X-0.2))*2+1/COSH (100*(X-0.4))*4 +1/COSH (1000*(X-0.6))*6

表 2 適応型積分ルーチンの性能比較
Table 2 Comparison of Performance of Adaptive Quadrature Routines.

NO.	EXACT	ERROR REQUIREMENT									
		CADRE	QNC 7		DAQN 9		QUAD		QABS		
		ERROR	N	ERROR	N	ERROR	N	ERROR	N	ERROR	N
(1)	1.71828182846D+00	1.4E-08	9	3.6E-14	25	5.2E-18	21	1.4E-14	37	2.2E-13	13
(2)	7.0000000000D-01	2.9E-04	53	1.0E-04	121	2.2E-06	141	6.9E-05	163	5.7E-05	141
(3)	6.66666666667D-01	9.1E-08	17	6.8E-05	49	1.4E-04	31	1.0E-04	55	3.0E-06	77
(4)	4.79428226689D-01	3.1E-08	17	1.1E-12	25	3.5E-17	21	2.1E-14	37	1.4E-11	13
(5)	1.58223296373D+00	9.3E-08	33	4.1E-08	25	2.7E-08	21	7.8E-10	37	7.9E-07	13
(6)	4.0000000000D-01	5.4E-05	9	2.7E-06	25	1.6E-06	21	9.1E-07	37	5.1E-07	13
(7)	2.0000000000D+00	1.6E-04	33	4.8E-04	241	1.7E-10	91	3.8E-04	361	1.0E+03*	133
(8)	8.66972987340D-01	7.8E-07	9	1.9E-10	25	3.3E-11	21	6.0E-13	37	2.3E-09	13
(9)	1.15470066904D+00	1.7E-07	183	8.9E-10	97	1.0E-05	81	5.8E-10	145	4.8E-07	149
(10)	6.93147180560D-01	7.2E-07	9	4.2E-11	25	3.9E-13	21	3.9E-14	37	5.0E-10	13
(11)	3.79885493042D-01	2.0E-06	5	1.8E-14	25	3.3E-18	21	1.8E-15	37	2.6E-13	13
(12)	7.77504634112D-01	4.0E-08	9	2.1E-14	25	2.2E-18	21	2.1E-14	37	1.1E-14	13
(13)	9.09864525657D-03	1.2E-07	1028	1.2E-01*	49	6.2E-07	321	1.1E-08	865	2.8E-08	573
(14)	5.00000211166D-01	1.3E-06	62	2.4E-08	97	8.1E-10	71	1.1E-09	127	1.8E-09	85
(15)	1.0000000000D+00	1.0E-06	88	1.3E-07	85	1.1E-06	61	1.9E-07	109	7.9E-09	85
(16)	4.99363802871D-01	5.4E-06	81	2.5E-09	121	5.2E-07	91	9.1E-09	163	9.3E-08	109
(17)	1.12139569627D-01	3.6E-04	512	1.1E-03*	165	7.4E-04	101	4.1E-05	307	5.0E-04	149
(18)	8.38676323381D-01	1.4E-07	107	3.8E-07	85	5.4E-06	51	7.6E-05	73	2.9E-07	77
(19)	-1.0000000000D+00	4.1E-06	137	3.2E-06	217	4.6E-11	91	4.1E-06	307	2.6E-05	181
(20)	1.56439644407D+00	7.0E-07	17	2.5E-08	25	1.1E-08	21	6.1E-07	37	6.6E-07	13
(21)	2.10802735501D-01	1.1E-03*	108	1.1E-03*	97	1.1E-03*	61	1.1E-03*	127	1.1E-03*	77
		95%	120	86%	79	95%	66	95%	149	90%	93

機センターの FACOM 230-75, 使用言語は FORT RANH-OPT 2, 精度は 10 進 18 桁相当の倍精度である。

比較の対象として選んだ適応型積分ルーチンは, de Boor のロンバーグ法による CADRE⁹⁾, Kahaner の 7 点則による QNC 7⁸⁾ と, 10 点則による QUAD⁹⁾ および O'Hara-Smith のクレンショールカーチス法によ

る QABS^{5),10)} である。CADRE の結果は CDC 6500, その他のルーチンの結果は CDC 6600 によるもので, すべて 10 進相当 14 桁の精度で行われた実験結果である。

Kahaner の問題集は表 1 にまとめられている。表の各行には, 問題番号, 積分区間の下限, 上限, 積分の値および被積分関数を, この順序に示した。

表 3 適応型積分ルーチンの性能比較
Table 3 Comparison of Performance of Adaptive Quadrature Routines.

NO	EXACT	ERROR REQUIREMENT 1.0E-06									
		CADRE		QNC 7		DAQN 9		QUAD		QABS	
		ERROR	N	ERROR	N	ERROR	N	ERROR	N	ERROR	N
(1)	1.71828182846D+00	2.8E-10	17	3.6E-14	25	5.2E-18	21	1.4E-14	37	2.2E-13	13
(2)	7.00000000000D-01	4.8E-08	119	1.0E-07	241	3.3E-08	201	7.2E-09	361	5.6E-08	261
(3)	6.66666666667D-01	3.2E-08	33	6.0E-09	157	2.5E-12	111	8.8E-09	217	5.9E-09	145
(4)	4.79428226689D-01	4.8E-10	33	1.1E-12	25	3.5E-17	21	2.1E-14	37	8.0E-14	25
(5)	1.582323296373D+00	1.0E-08	49	3.6E-11	49	1.1E-11	41	7.8E-10	37	4.0E-12	49
(6)	4.00000000000D-01	2.7E-07	65	1.5E-08	61	5.1E-08	41	2.9E-08	73	5.0E-10	65
(7)	2.00000000000D+00	8.9E-08	129	4.8E-04*	241	1.7E-10	111	3.8E-04*	361	1.0E+03*	89
(8)	8.66972987340D-01	1.7E-08	17	1.9E-10	25	3.3E-11	21	6.0E-13	37	1.1E-10	25
(9)	1.15470066904D+00	3.3E-09	409	9.3E-09	289	9.3E-09	221	7.9E-10	397	1.1E-10	313
(10)	6.93147180560D-01	1.4E-08	17	4.2E-11	25	3.9E-13	21	3.9E-14	37	5.0E-10	13
(11)	3.79885493042D-01	2.5E-09	9	1.8E-14	25	3.3E-18	21	1.8E-15	37	2.6E-13	13
(12)	7.77504634112D-01	1.6E-10	9	2.1E-14	25	2.2E-18	21	2.1E-14	37	1.1E-14	13
(13)	9.09864525657D-03	4.2E-10	1449	9.9E-12	1525	1.1E-10	641	1.0E-11	1639	3.8E-11	1449
(14)	5.0000211166D-01	4.6E-08	89	8.1E-10	133	3.4E-10	91	8.0E-11	163	8.6E-11	109
(15)	1.00000000000D+00	4.9E-09	140	4.6E-10	133	5.5E-10	81	9.5E-11	145	3.3E-11	133
(16)	4.99363802871D-01	1.0E-07	145	6.8E-10	181	1.6E-09	101	1.2E-10	181	2.3E-10	145
(17)	1.12139569627D-01	1.4E-09	1237	1.1E-03*	385	2.8E-09	491	3.2E-11	1009	9.1E-11	829
(18)	8.38676323381D-01	4.3E-09	177	7.9E-10	181	1.7E-09	111	5.7E-12	199	1.3E-12	205
(19)	-1.00000000000D+00	1.3E-08	233	8.5E-07	241	4.6E-11	111	5.7E-07	361	1.0E+03*	105
(20)	1.56439644407D+00	4.8E-09	33	3.0E-13	48	1.1E-08	21	6.1E-10	37	5.0E-14	49
(21)	2.10802735501D-01	1.1E-03*	189	1.1E-03*	205	1.1E-03*	111	1.1E-03*	253	1.1E-03*	197
		95%	219	86%	201	95%	124	90%	269	86%	202

表 4 適応型積分ルーチンの性能比較
Table 4 Comparison of Performance of Adaptive Quadrature Routines.

NO	EXACT	ERROR REQUIREMENT 1.0E-09									
		CADRE		QNC 7		DAQN 9		QUAD		QABS	
		ERROR	N	ERROR	N	ERROR	N	ERROR	N	ERROR	N
(1)	1.71828182846D+00	6.0E-11	17	3.6E-14	25	5.2E-18	21	1.4E-14	37	2.8E-14	25
(2)	7.00000000000D-01	2.8E-10	173	1.0E-07*	241	3.2E-11	301	7.2E-09*	361	5.5E-11	381
(3)	6.66666666667D-01	1.2E-10	129	6.0E-12	289	2.5E-12	161	2.9E-12	361	4.4E-12	289
(4)	4.79428226689D-01	1.3E-13	33	1.1E-12	25	3.5E-17	21	2.1E-14	37	2.0E-14	85
(5)	1.582323296373D+00	4.2E-12	129	1.5E-12	97	1.8E-11	61	1.5E-12	73	1.6E-12	181
(6)	4.00000000000D-01	2.7E-10	529	4.2E-12	133	8.9E-12	91	5.0E-12	163	5.1E-13	137
(7)	2.00000000000D+00	1.3E-10	625	4.8E-04*	585	1.7E-13	311	3.8E-04*	685	1.0E+03*	89
(8)	8.66972987340D-01	2.1E-10	65	1.2E-12	73	5.0E-15	41	0.0	73	7.1E-15	97
(9)	1.15470066904D+00	8.4E-12	785	4.1E-12	697	1.9E-12	441	3.3E-13	757	1.7E-13	893
(10)	6.93147180560D-01	3.6E-12	33	1.7E-12	37	3.9E-13	21	3.9E-14	37	1.6E-13	49
(11)	3.79885493042D-01	1.4E-12	17	1.8E-14	25	3.3E-18	21	1.8E-15	37	0.0	25
(12)	7.77504634112D-01	2.2E-12	17	2.1E-14	25	2.2E-18	21	2.1E-14	37	1.1E-14	13
(13)	9.09864525657D-03	3.8E-13	3505	5.6E-13	3073	6.3E-14	1271	5.6E-13	2773	5.7E-13	3197
(14)	5.0000211166D-01	1.2E-12	202	1.8E-13	241	9.4E-14	141	7.5E-14	253	7.1E-14	245
(15)	1.00000000000D+00	8.6E-12	215	5.0E-13	241	2.3E-13	131	1.5E-13	217	5.0E-14	281
(16)	4.99363802871D-01	1.3E-11	337	2.9E-12	397	4.8E-12	211	3.3E-12	343	9.4E-13	397
(17)	1.12139569627D-01	2.9E-12	2329	3.6E-04*	1345	1.2E-12	1031	3.3E-12	1999	3.3E-12	2025
(18)	8.38676323381D-01	1.5E-12	417	8.6E-13	409	1.1E-12	201	1.0E-12	343	1.1E-12	589
(19)	-1.00000000000D+00	4.5E-09*	369	8.5E-07*	421	2.9E-12	201	5.7E-07*	415	1.0E+03*	85
(20)	1.56439644407D+00	1.1E-09*	129	2.8E-14	97	2.3E-13	61	3.6E-14	73	5.7E-14	145
(21)	2.10802735501D-01	1.1E-10	661	9.8E-11	709	1.1E-03*	221	1.0E-10	685	1.0E-10	633
		90%	510	81%	437	95%	237	86%	465	90%	470

性能比較の結果は、絶対誤差規準の三つの値 10^{-3} , 10^{-6} , 10^{-9} の各々に対して、それぞれ表 2, 表 3, 表 4 に収められている。これらの表では、ERROR の見出しの下に、計算値の絶対誤差の絶対値、N の見出しの下に関数計算回数を与えられている。また、ERROR

の数値の右につけられた星印は、その誤差が規準を越えていることを意味している。さらに、最下行には、21 問題全体に対する成功率と、平均関数計算回数がつけ加えられている。

さて、これらの表を観察すれば、次のような事実が

認められるであろう。

(1) 全般的に、関数計算回数の少なさにおいて、DAQN 9 が断然すぐれている。これは主に、誤差推定の強化に起因するものと考えられる。

(2) 振動型の問題 (9, 13, 17, 18) に対して計算回数が一様に大となっていて、適応型積分法の振動型問題に対する弱点を如実に表わしている。この点については、DAQN 9 も例外ではないが、他のルーチンに対する優位はこの場合さらに著しい。これは収束判定の緩和の効果であると思われる。

(3) 異常点をもつ問題 (2, 3, 7, 19) については、当然のことながら、DAQN 9 と CADRE とが有利である。これがそのまま成功率の高さに反映している。

(4) ピーク型の問題 (14, 15, 16, 21) は適応型積分法の最も得意とするところである。しかし、三つの、針のように鋭いピークをもつ難問 21 だけは例外である。その中でも DAQN 9 が最悪である。これは、DAQN 9 では分点が少ないので、 $x=0.6$ にある最も鋭いピークが分点の網から完全にもれてしまうためである。

以上の観察から、少なくとも Kahaner の問題集に関して、そして恐らくは実際の計算に現れる多くの問題において、DAQN 9 の優秀性が結論づけられるものと考ええる。

7. あとがき

三つの新考案の導入により、信頼性、経済性および融通性に富む強力な適応型積分法がえられた。

今後の課題としては、振動型問題への対策、収束判定緩和法に関する理論の形成そしてコーシーの主値積分の処理などが挙げられる。

参考文献

1) Clenshaw, C. W. and Curtis, A. R.: A Method for Numerical Integration on an Automatic

Computer, Numer. Math., Bd. 2, pp. 197-205 (1960).

2) 鳥居連生, 長谷川武光, 二宮市三: 等差数列的に標本数を増す補間的自動積分法, 情報処理, Vol. 19, No. 3, pp. 248-255 (1978).

3) Takahashi, H. and Mori, M.: Quadrature Formulas obtained by Variable Transformation, Numer. Math., Bd. 21, pp. 206-219 (1973).

4) McKeeman, W. M.: Algorithm 145, Adaptive Numerical Integration by Simpson's Rule, Commun. ACM, Vol. 5, No. 12, p. 604 (1962).

5) Kahaner, D. K.: Comparison of Numerical Quadrature Formulas II, in Mathematical Software (Rice, J. R., ed.), pp. 229-259, Academic Press, New York (1971).

6) de Boor, C.: CADRE: An Algorithm for Numerical Quadrature, in Mathematical Software (Rice, J. R., ed.), pp. 417-449, Academic Press, New York (1971).

7) 名古屋大学大型計算機センター: ライブラリプログラム利用の手引(改訂版), pp. 80-83 (1978).

8) 富士通: FACOM FORTRAN SSLII 使用手引書, 99SF-0050-4, pp. 308-311 (1979).

9) Forsythe, G. E., Malcom, M. A. and Moler, C. B.: Computer Methods for Mathematical Computation, p. 107, Prentice-Hall, Englewood Cliffs, N. J. (1977).

10) O'Hara, H. and Smith, T. J.: The Evaluation of Definite Integrals by Interval Subdivision, Comput. J., Vol. 12, pp. 179-182 (1969).

11) Lyness, J. N.: Notes on the Adaptive Simpson Quadrature Routine, JACM, Vol. 10, No. 3, pp. 483-495 (1969).

12) Lyness, J. N. and Kaganove, J. J.: Comments on the Nature of Automatic Quadrature Routines, ACM Trans. Math. Software, Vol. 2, No. 1, pp. 65-81 (1976).

13) Davis, P. J. and Rabinowitz, P.: Methods of Numerical Integration, Academic Press, New York (1975).

(昭和54年8月31日受付)