Improvements of Adaptive Newton-Cotes Quadrature Methods

ICHIZO NINOMIYA*

The following three new ideas are introduced to adaptive Newton-Cotes quadrature methods.

(1) Refinement of error estimation.

(2) Relaxation of convergence criterion.

(3) Treatment of extraordinary points.

By virtue of these improvements, reliability, efficiency and versatility of the methods are largely enhanced.

The adaptive Newton-Cotes 9-point quadrature subroutine DAQN9 incorporating these improvements is described and its performances in comparison with well-known subroutines are shown.

1. Introduction

An algorithm which computes an approximation S, satisfying the condition

$$\left|\int_a^b f(x)\,dx - S\right| < \varepsilon_0,$$

for a given integral $\int_a^b f(x)dx$ and an error tolerance ε_0, automatically subdividing the integral region and determining sample points, is called an automatic quadrature. There exist two types of algorithms, global and adaptive.

A classic example of the global method is Romberg integration. However it is a well-known fact that the Clenshaw-Curtis method [1, 2] and the double exponential formulas of Takahashi and Mori [3] are more efficient. These methods rely on some global properties of integrands such as analyticity or at least smoothness or periodicity, and distribute and increase sample points uniformly in the region according to a certain prescribed manner.

Adaptive methods, on the other hand, have the common objective of adapting the density of sample points to the local behavior of the integrand thereby promoting computational efficiency. There are various members of this type of method [4, 5]. However the principal ones are those based on Newton-Cotes quadrature rules.

In this paper, the following three improvements to adaptive Newton-Cotes quadrarures for the purpose of extending their merits and largely enhancing their versatility are proposed.

(1) Refinement of Error Estimation

Suppose that Newton-Cotes $n+1$-point rule has been applied to an integral on an interval and an approximation S has been obtained. The usual way to estimate the truncation error is to bisect the interval, apply the same rule to both of the subintervals, obtain another approximation S' and examine the difference $S'-S$. Obviously the above procedure requires new function values at n sample points.

It will be shown that, by virtue of the refinement of the error estimation, only two additional samples are needed.

(2) Relaxation of Convergence Criterion

In an adaptive quadrature, a convergence test should be done separately for every subinterval. The problem here is how to assign a local convergence criterion ε to each subinterval corresponding to the overall criterion ε_0. The conventional way is to distribute ε_0 over each subinterval in proportion to its length. This is straight forward and exact but it is also too pessimistic and conservative. Instead, a progressive strategy of relaxing local criterion is proposed in this paper.

(3) Treatment of Extraordinary Points

The integrals encountered in practical scientific and technological calculations do not always involve analytic functions. Sometimes integrals with discontinuous or singular integrands are required. Therefore, it is highly desirable for an automatic quadrature to have the capability of processing such integrals.

Among global methods, double exponential function type methods are known to be competent for treating singularities situated at the ends of the integration region. On the other hand, de Boor [6] gave such an ability to his famous adaptive quadrature subroutine CADRE which was constructed on the basis of the Romberg integration method. This paper shows that the same is possible for adaptive Newton-Cotes quadratures.

The subroutines AQNN5D, AQNN7D and AQNN9D implementing adaptive Newton-Cotes 5, 7 and 9 point quadrature respectively were developed for the program library of Nagoya University computation Center [7]. Recently, the subroutine DAQN9, added facilities for the input of relative error criterion and the output of an estimated error to AQNN9D. It is registered as a

*Dept. of Information Science, Nagoya University, Nagoya, 464, Japan.

member of the scientific subroutine library SSL-II of the Fujitsu Company. Later in this paper the specifications (user interface) and outline of the DAQN9 algorithm will be explained. Furthermore, the results of performance tests of DAQN9 conducted for Kahaner's 21 test problems will be shown and contrasted with those of several well-known subroutines.

2. Refinement of Error Estimation

Newton-Cotes rules are interpolatory quadrature formulas which make use of equidistant mesh points of the interval. These formulas are the most suitable for adaptive automatic quadrature, because function values, once computed, are stored and reused afterwards. The rules using odd numbers of sample points are especially favorable in that their orders of accuracy are the same as those using one more sample point. Here we confine our attention to 5, 7 and 9-point rules from a practical viewpoint.

Now, let us consider an interval and a mesh of points dividing it into $2n$ equal parts. A mid point is added to each of the outermost subintervals and the points are numbered as shown in Fig. 1.

Let Q be the exact value of an integral over the interval, e be the truncation error, h be the half width of the interval and f_i be the function values at the point i, $i=1, 2, \cdots, 2n+3$.

Then, Newton-Cotes $2n+1$-point rules and corresponding error estimators, which are derived below, are given as follows, where $f^{(6)}, f^{(8)}$ and $f^{(10)}$ denote derivatives of the order 6, 8 and 10 respectively at some appropriate points of the interval.

(1) 5-Point Rule

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

$$e=\frac{hf^{(6)}}{15120}\approx\frac{32h}{6615}\{15(f_0+f_6)-64(f_1+f_5)+84(f_2+f_4)-70f_3\}. \qquad (2.2)$$

(2) 7-Point Rule

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

$$e=\frac{h^9 f^{(8)}}{1020600}\approx\frac{4h}{9625}\{105(f_0+f_8)-512(f_1+f_7)+770(f_2+f_6)-825(f_3+f_5)+924f_4\}. \qquad (2.4)$$

(3) 9-Point Rule

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

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

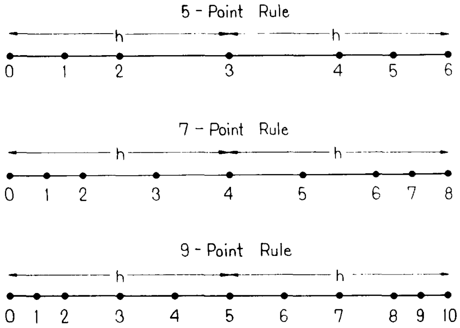

Fig. 1 Arrangement of sample points.

The usual way to estimate the error e is as follows. Let $S(x, h)$ denote the approximation obtained by $2n+1$-point rule for an integral on the interval with lower limit x and half width h. S is given in the form of

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

from the above formulas. In order to estimate e, we bisect the interval and apply the $2n+1$-point rule to each of the half intervals. Evidently, the results are given by

$$S(x, h/2)=Q_1+c_1(h/2)^{2n+3},$$

and

$$S(x+h, h/2)=Q_2+c_2(h/2)^{2n+3},$$

where Q_1 and Q_2 denote the integrals over the two halves. Making use of the relation

$$Q=Q_1+Q_2,$$

and equating c_1 and c_2 with c, we obtain

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

It will be observed here that $2n$ additional sample must be supplied to estimate the truncation error of the $2n+1$-point rule by this method.

Now a little examination of the above derivation of (2.7) reveals that it only utilizes the proportionality of e to h^{2n+3} out of the totality of the error theory. As a matter of fact, e is proportional to $h^{2n+3}f^{(2n+2)}$. Therefore, e can be estimated by estimating $f^{(2n+2)}$ directly using $2n+3$ sample values. In other words, two more function values are sufficient.

It is almost evident from the principle of bisection and symmetry that the additional sample points should be a symmetric pair of mid points of 2 subintervals. But there is still an indeterminacy concerning the choice of the pair. Taking account of the excellent properties of interpolation based on zeros of Chebyshev or Legendre polynomials which have high densities near the ends of interval and the advantages in treating end singularities, the pair of the outermost mid points is selected.

Now, let D denote the numerical differentiation functional for the $2n+2-nd$ derivative which is a linear combination of function values at the above mentioned $2n+3$ points. Since D is symmetric, the number of unknown coefficients is $n+2$. We apply the functional D to the special functions x^{2k}, $k=0, 1, \cdots, n+1$, on the interval $[-h, h]$ and obtain.

$$D(x^{2k})=0, \quad k=0, 1, \cdots, n,$$
$$D(x^{2n+2})=(2n+2)!$$

Solving these $n+2$ equations simultaneously, we can determine the $n+2$ coefficients. The formulas (2.2), (2.4) and (2.6) were obtained in this manner. In fact, the equations may be solved very accurately by means of multiple precision arithmetic. The exact solutions may be reconstructed in the form of rational numbers from the easily recognizable cyclic structures of the numerical solutions.

Let us consider the saving of function evaluations brought about by the adoption of the new error estimate. Suppose that an algorithm A which uses the old error estimate, and another one B which is the same as A except that it uses the new error estimate, are applied to the same integral. Assume further that the both algorithms terminate with the same level of subdivision everywhere. Then, the integral region will be partitioned into the same number, say N, of subintervals each of which contains $4n+1$ and $2n+3$ sample points including the end points in cases A and B respectively. Accordingly, the total numbers of sample points are $4nN+1$ and $(2n+2)N+1$ respectively. Thus, the rate of saving is given by:

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

This formula takes on the values of 3/4, 2/3 and 5/8 correspoinding to the 5, 7 and 9-point rules. Although the assumption that A and B pursue the same course of convergence is not strictly true, it is probably true on the average and the above values of the saving rate are considered to be rather realistic.

3. Relaxation of Convergence Criterion

In a global quadrature, approximations are always provided and tested for convergence for the integral as a whole. In the case of an adaptive quadrature, on the contrary, the convergence test must be performed locally and separately for each subinterval. The problem here is how to determine the local error criterion ε according to the overall criterion ε_0. The obvious solution is to use a proportional allocation. In this scheme, the local error criterion ε_i for the i-th subinterval is determined by

$$\varepsilon_i=\varepsilon_0 h_i/h_0, \tag{3.1}$$

where h_0 and h_i are the half widths of the whole interval and the i-th subinterval respectively. Suppose that every subinterval passes the convergence test for the criterion ε_i, then overall error E satisfies:

$$|E|=\left|\sum_{i=1}^{N} e_i\right| \leqq \sum_{i=1}^{N} |e_i| \leqq \sum_{i=1}^{N} \varepsilon_i=(\varepsilon_0/h_0) \sum_{i=1}^{N} h_i=\varepsilon_0, \tag{3.2}$$

and consequently the requirement:

$$|E| \leqq \varepsilon_0. \tag{3.3}$$

It is not rare that the inequalities appearing in (3.2) and hence the inequality (3.3) turn out to be slack ones with large discrepancies. When the integrand is oscillatory or when local approximations are refined by subtraction of estimated error, the tendency will be particularly pronounced. In short, proportional allocation is extremely conservative and devoid of positivity.

Kahaner has conceived the Banking Method [9]. Everytime the estimated error is compared with error criterion the surplus is deposited in the bank and any deficit is made up by with drawing from the bank. The idea itself is attractive but its practical utility seems dubious. In practical applications, estimated error is not able to withstand such a rigorous additivity.

Thus, in so far as the infallibility to meet the error requirement (3.3) is adhered to, there seems to be no good method other than proportional allocation. Observe, however, that no algorithm is ever infallible. Indeed, for every algorithm, however ingeneous and sophisticated, we can easily construct a problem which, when handled by it, yields an erroneous answer [12].

In view of the above fact,

$$\varepsilon_i=\varepsilon_0(h_i/h_0) \log_2 (h_0/h_i) \tag{3.4}$$

is proposed in place of (3.1) in order to positively relax error criterion for small subintervals. The reason the relaxation factor $\log_2 (h_0/h_i)$ is used is that in the first place it leads to a cautious strategy and in the second place its computational cost is negligible. Unfortunately, no adequate analytical or statistical theory concerning the application of the new method has been conceived as yet. Consequently, the merit must be judged exclusively by experimentation. As far as the numerous experiments worked out so far, there seems to be no negative evidence. In the case of oscillatory integrals, it proves to be efficient in preventing fruitless subdivisions at early stages thereby saving function evaluations.

4. Treatment of Extraordinary Points

Consider an integral of a function well approximated locally by a polynomial of a moderate degree. As the process of bisection goes on, we eventually arrive at a stage where the derivative appearing in the error formula can be regarded as a constant. From then on, further bisection divides estimated error e by 2^{2n+3} while it halves error criterion ε. (Here we neglect the relaxation factor). Thus, the quantity e/ε will be reduced to $1/2^{2n+2}$ times the last value. This property of e/ε indicating the rapid convergence of the $2n+1$-point rule is displayed in a more straight forward manner by the normalized error $\tilde{e}=e/h$.

Suppose now that the integration region contains discontinuities and/or singularities. At the outset, it is assumed that function values at singularities where they are undefined originally, are replaced by an arbitrary finite value, say, 0. In such a situation, the condition for convergence will not generally be met. Consequently, the bisection process goes on without limit and a series of subintervals containing the same extraordinary point in the interior or at end points will be generated. Assume here that the point in question is located at one of the points which divide the region into 2^m equal parts, where m is a positive integer, then, from a certain stage on, the point is the common extremity of the series of subintervals. Without loss of generality, we may assume further that the extremity is the left one and is the origin.

Now let us examine the behavior of the values $\tilde{e}_i$ of normalized error in such a sequence of subintervals $\{I_i, i=1, 2, \cdots\}$. Since $\tilde{e}$, when considered as a functional, is linear and decreases rapidly for normal functions, $\tilde{e}_i$ will eventually manifest genuine properties of the extraordinary point.

4.1 Detection of an Extraordinary Point

(1) Discontinuity

Consider the case where the integrand f has a discontinuity δ at the origin and is constant otherwise.

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

Then, since $\tilde{e}(\alpha)=0$, we immediately obtain

$$\tilde{e}_i=c_0\delta, \tag{4.2}$$

where c_0 is the coefficient of the function values at the ends of interval in the error estimator. Thus, the sequence $\{\tilde{e}_i, i=1, 2, \cdots\}$ becomes a constant sequence.

(2) Logarithmic Singularity

Assume that

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

in the vicinity of the origin. Clearly, $\tilde{e}_i$ can be written as

$$\tilde{e}_i=\sum_{j=0}^{2n+2} c_i f(2^{-i}x_j), \tag{4.4}$$

for appropriately chosen x_j, $j=0, 1, \cdots, 2n+2$, where $x_0=0$. Inserting (4.3) into (4.4) and using the property

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

we obtain

$$\tilde{e}_i=c_0\delta+\alpha\sum_{j=1}^{2n+2} c_j \log x_j+i\alpha c_0 \log 2. \tag{4.5}$$

Thus, the sequence $\{\tilde{e}_i, i=1, 2, \cdots\}$ becomes an arithmetical progression with the constant increment $\alpha c_0 \log 2$.

(3) Algebraic Singularity

Assume that

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

in the vicinity of the origin. Inserting (4.6) into (4.4) we have

$$\tilde{e}_i=c_0\delta+\alpha 2^{-ip}\sum_{j=1}^{2n+2} c_j x_j^p. \tag{4.7}$$

Define the first difference $\Delta\tilde{e}_i$ by

$$\Delta\tilde{e}_i=\tilde{e}_{i+1}-\tilde{e}_i,$$

then, from (4.7), we obtain

$$\Delta\tilde{e}_i=\alpha 2^{-ip}(2^{-p}-1)\sum_{j=1}^{2n+2} c_j x_j^p. \tag{4.8}$$

Thus, the sequence $\{\Delta\tilde{e}_i, i=1, 2, \cdots\}$ becomes a geometrical progression with the constant ratio 2^{-p}.

Now we can summarize the detection of extraordinary points as follows.

Examine the behavior the sequence $\{\tilde{e}_i, i=1, 2, \cdots\}$ of normalized errors $\tilde{e}=e/h$ for the sequence of subintervals each of which is a half of its predecessor.

(a) If the sequence tends to a constant c, the common extremity is a discontinuity with discrepancy δ.

$$\delta=c/c_0 \tag{4.9}$$

(b) If the sequence converges to an arithmetical progression with the increment d, the common extremity is a logarithmic singularity whose coefficient is α.

$$\alpha=d/(c_0 \log 2) \tag{4.10}$$

(c) If the sequence of differences $\{\Delta\tilde{e}_i, i=1, 2, \cdots\}$ converges to a geometrical progression with the ratio r, the common extremity is an algebraic singularity with the order p.

$$p=-\log_2 r \tag{4.11}$$

4.2 Evaluation of Integrals Involving Extraordinary Points

When the existence of an extraordinary point is recognized, the integral over the subinterval containing it in one of the extremities can be evaluated analytically as follows. Here, again, we assume for the sake of simplicity that the point is at the left end and is the origin $x=0$.

(1) Discontinuity

It suffices to correct the value $f(0)$ by subtracting the discrepancy δ and use it in the quadrature rule.

(2) Logarithmic Singularity

Suppose that the integrand is expressed as

$$f(x)=\alpha \log x+\beta+\gamma x, \tag{4.12}$$

in the subinterval $[0, 2h]$. The integral under consideration is given analytically by

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

which can be readily computed as

$$\int_0^{2h} f(x)\,dx = 2h\{f(h)+\alpha(\log 2-1)\}. \tag{4.13}$$

(3) Algebraic Singularity

Suppose that the integrand is expressed as

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

in the subinterval $[0, 2h]$, where the order $p>-1$ is assumed to be known. The integral over $[0, 2h]$ is given analytically by

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

In order to compute the right hand side of (4.15), we may use three values $f(0)$, $f(h)$ and $f(2h)$. Among them, $f(0)$ provides a problem. It would be simple and convenient to request that $f(0)$ should be the value obtained by replacing the principal part αx^p with 0, i.e., $f(0)=\gamma$. In practice, however, it is not always an easy task to determine the principal part of an algebraic singularity.

Here we follow the philosophy of automatic quadrature to presuppose as little preliminary knowledge concerning the integrand as possible and we decide to allow an arbitrary deviation for the function value at singularity.

Thus, letting δ denote the deviation, we have

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

It is a simple matter to compute (4.15) by means of (4.16), since δ is given by

$$\delta=\frac{(2^{-p}-1)\tilde{e}_i-\Delta\tilde{e}_i}{(2^{-p}-1)c_0}, \tag{4.17}$$

from (4.7) and (4.8).

It must be emphasized here that the above explanation of the treatment of improper integrals assumes that the extraordinary point is located at one of the mesh points dividing the integral region into 2^m equal subintervals. Therefore, if necessary, variable transformations or subdivision of the integral region must be performed in order to transfer the extraordinary points to such mesh points, e.g., the end point or the mid point. Furthermore, it is important to protect the function values from a loss of singnificance near the singularities in order to ensure the precision in the treatment of improper integrals. In this respect, it is advisable to move the singularity to the origin $x=0$.

5. Adaptive Automatic Quadrature Subroutine DAQN9

In this chapter we describe the specifications and outline of the algorithm of the adaptive automatic quadrature subroutine DAQN9 based on Newton-Cotes 9-point rule. See ref. [8] for details.

5.1 Purpose

Given the integrand $f(x)$, the interval of integration $[a, b]$, absolute error criterion ε_a and relative error criterion ε_r, DAQN9 computes an approximation S which hopefully satisfies the following condition.

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

5.2 Usage

A FORTRAN statement that may be used to initiate DAQN9 follows: CALL DAQN9 (A, B, FUN, AEPS, REPS, NMIN, NMAX, S, ERR, N, ICON) where:

A···Input.	Lower limit of integration region.
B···Input.	Upper limit of integration region.
FUN···Input.	Function subprogram for computing the integrand $f(x)$.
AEPS···Input.	Absolute error criterion ε_a.
REPS···Input.	Relative error criterion ε_r.
NMIN···Input.	Lower bound for the number of function evaluations.
NMAX···Input.	Upper bound for the number of function evaluations.
S···Output.	Approximation for the integral.
ERR···Output.	Magnitude of estimated absolute error of S.
N···Output.	Number of function evaluations.
ICON···Output.	Condition code for the quality of result.

The program is coded in FORTRAN strictly in conformity to level 7000 grammer, and ample caution has been exercised to establish portability. The necessary memory space is about 10 KB, it can be run on any middle or large scale computer.

5.3 Outline of the Algorithm

Step 1···Initialization

Set the lower limit x, half width h and running approximation S as $x=a$, $h=h_0=(b-a)/2$, $S=0$.

Initialize various variables and determine several test criteria.

Evaluate the integrand at 9 equally spaced nodes and 2 mid points of the outermost subintervals of the interval $[a, b]$ (see Fig. 1).

Step 2···Bisection and Stacking of Data

Bisect the current subinterval which is specified by x and h.

Evaluate the integrand at 6 points which, when added to 11 already sampled points, constitute the complete set of 17 equidistant mesh points.

Compute the integral over the right half by the 9-point rule and store the result in the stack together with relevant function values, half width h and so on.

The integral is used as a part of the approximation S' which is multiplied by the relative error criterion ε_r for the purpose of the convergence test. It is also reused later when the right half is treated.

Proceed to Step 3 to treat the left half.

Step 3···Convergence Test

If the integral over the current subinterval is not available (when we come from Step 2), evaluate it by the 9 point rule. Evaluate the integrand at the outermost mid points and estimate the truncation error e by means of the formula labeled (2.6). Test e for convergence, i.e., examine the inequality:

$$|e| \leqq \max(\varepsilon_a, \varepsilon_\gamma |S'|)\cdot(h/h_0) \log_2 (h_0/h).$$

Here S' is recomputed everytime using the most recent data.

If the inequality is satisfied, subtract e from the integral and proceed to Step 7. Otherwise proceed to Step 4.

Step 4···Examination for an Extraordinary Point

Return to Step 2 if h is larger than a certain allowable limit which is determined from input data.

Proceed to Step 5 when h is smaller than a certain allowable limit which is determined from input data.

Examine the existence of an extraordinary point at one of the ends of the current subinterval.

Go to Step 6, if the existence is recognized, and return to Step 2 otherwise.

Step 5···Reexamination for an Extraordinary Point

Reexamine the existence of an extraordinary point with attenuated criterion for detection.

Proceed to Step 6 if it succeeds. Otherwise, give up further treatments, accept the current subinterval without convergence and proceed to Step 7.

Step 6···Analytic Treatment of an Extraordinary Point

Perform an analytic calculation according to the category and characteristic value of the extraordinary point for the integral over the current subinterval and proceed to Step 7.

Step 7···Updating of the Running Approximation and Retrieval of Data

Add the value of the integral over the current subinterval to the running approximation S.

Terminate the algorithm if the stack is empty. Otherwise, take the necessary data from the stack and return to Step 2.

6. Numerical Examples

Numerical experiments have been conducted on Kahaner's 21 test problems to investigate the performance of DAQN9 and compare it with those of well-known established subroutines. The details of the experiments are as follows.

The Computer···FACOM 230–75 of Nagoya University Computation Center

Language ·····FORTRANH-OPT2

Precision ·······61 bits≈18D

Error requirements···10^{-3}, 10^{-6}, 10^{-9} (absolute error)

The adaptive quadrature subroutines selected for comparison are de Boor's CADRE [6], Kahaner's QNC7 [5] and QUAD [5], and O'Hara-Smith's QABS [5, 10]. The experiment data for these subroutines are reproduced from refs. [5] and [6].

Kahaner's test problems are collected in Table 1. From left to right, the table lists problem numbers, lower limits, upper limits, values of integrals and integrands. The results of comparisons are divided into three tables, Tables 2, 3 and 4 corresponding to the error requirements. In each of the tables, magnitudes of absolute errors and numbers of function evaluations are given under the titles ERROR and N respectively. Asterisks attached to errors indicate the failure to meet the error requirement. In the bottom lines, percentages of success for all the problems and average sample numbers are added.

A little examination of the tables reveals the following facts.

(1) As a whole, DAQN9 decisively excels on the number of function evaluations. This is considered to be mainly due to the refinement of error estimation.

(2) The weakness of adaptive quadratures for oscillatory integrals is manifested by the general increase of the number of sample points for problems 9, 13, 17 and 18. DAQN9 is not free from this weakness either but its superiority in sample point numbers is more pronounced in these cases. This is possibly the effect of the relaxation of convergence criterion.

(3) As it should be, DAQN9 and CADRE are advantageous for the integrals with extraordinary points, numbers 2, 3, 7 and 19, and this is reflected in the high percentage of successes.

(4) Adaptive quadratures are good at problems with peaks, numbers 14, 15 and 16. But the extremely difficult problem 21 with three sharp peaks is exceptional. DAQN9 along with others, behaves the worst for this problem. Since DAQN9 uses fewer sample points, the sharpest peak situated at $x=0.6$ escapes completely from the net of sample points.

From the above observations, the excellence of DAQN9 for Kahaner's problems and probably for many practical problems can be safely concluded.

7. Conclusion

A new, reliable, economical and versatile adaptive quadrature has been constructed with the introduction of three improvements to adaptive Newton-Cotes quadrature methods.

Out of the tasks remaining for the future, only the following are mentioned here: strategies to cope with oscillatory integrals, the formation of an adequate theory for the distribution of error tolerance and the treatment of Cauchy's principal value integrals.

Table 1 The 21 test problems of Kahaner.

NO	LL	UL	INTEGRAL	INTEGRAND
(1)	0.0	1.0000000	1.7182818285D+00	EXP(X)
(2)	0.0	1.0000000	7.0000000000D-01	AINT(AMIN1(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*X+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

Table 2 Comparison of performance of adaptive quadrature routines for 21 problems of Kahaner.

ERROR REQUIREMENT 1.0E-03

		CADRE		QNC7		DAQN9		QUAD		QABS	
NO	EXACT	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)	7.00000000000D-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)	4.00000000000D-01	5.4E-05	9	2.7E-06	25	1.6E-06	21	9.1E-07	37	5.1E-07	13
(7)	2.00000000000D+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	6.9E-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)	1.00000000000D+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)	-1.00000000000D+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

Table 3 Comparison of performance of adaptive quadrature routines for 21 problems of Kahaner.

ERROR REQUIREMENT 1.0E-06

		CADRE		QNC7		DAQN9		QUAD		QABS	
NO	EXACT	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)	1.58223296373D+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)	5.00002111166D-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

Table 4 Comparison of performance of adaptive quadrature routines for 21 problems of Kahaner.

ERROR REQUIREMENT 1.0E-09

		CADRE		QNC7		DAQN9		QUAD		QABS	
NO	EXACT	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)	1.58223296373D+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)	5.00002111166D-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

References

1. CLENSHAW, C. W. and CURTIS, A. R. A Method for Numerical Integration on an Automatic Computer, *Numer. Math.*, Bd. 2 (1960) 197–205.
2. TORII, T. HASEGAWA, T. and NINOMIYA, I. An Automatic Integrati^n of Interpolatory Type Increasing the Sample Points with Arithmetic Progression, *Jyoho shori*, **19**, 3 (1978) 248–255. (In Japanese).
3. TAKAHASHI, H. and MORI, M. Quadrature Formulas Obtained by Variable Transformation, *Numer. Math.*, Bd. 21 (1973) 206–219.
4. MCKEEMAN, W. M. Algorithm 145, Adaptive Numerical Integration by Simpson's Rule, *Commun. ACM*, **5**, 12 (1962) 604.
5. KAHANER, D. K. Comparison of Numerical Quadrature Formulas II, in Matheamtical Software (Rice, J. R., ed.), 229–259, Academic Press, New York (1971).
6. DE BOOR, C. CADRE: An Algorithm for Numerical Quadrature, in Mathematical Software (Rice, J. R., ed.), 417–449, Academic Press, New York (1971).
7. Nagoya University Computation Center, User's Guide for Program Library, Rev. ed., (1978) 80–83. (In Japanese).
8. Fujitsu Co. FACOM FORTRAN SSL II User's Guide (Scientific Subroutine Library) 99SP-0050-E4, (1980).
9. FORSYTHE, G. E., MALCOM, M. A. and MOLER, C. B. Computer Methods for Mathematical Computation, 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.*, 12 (1969) 179–182.
11. LYNESS, J. N. Notes on the Adaptive Simpson Quadrature Routine, *JACM*, **10**, 3 (1969) 483–495.
12. LYNESS, J. N. and KAGANOVE, J. J. Comments on the Nature of Automatic Quadrature Routines, *ACM Trans. Math. Software*, **2**, 1 (1976) 65–81.
13. DAVIS, P. J. and RABINOWITZ, P. Methods of Numerical Integration, Academic Press, New York (1975).