Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

INTERPOLATORY INTEGRATION FORMULAS FOR OPTIMAL COMPOSITION

P.Favati, G. Lotti, F. Romani

Nota interna B04-15

Maggio 1987

Interpolatory Integration Formulas for Optimal Composition

by Paola Favati, Grazia Lotti and Francesco Romani

Abstract. A set of symmetric, closed, interpolatory integration formulas on the interval [-1,1] is investigated. These formulas, called recursive monotone, have the property that higher order or compound rules can be applied without wasting previous computed functional values. An exhaustive search shows the existence of 27 families of recursive monotone formulas with positive weights and increasing degree of precision, stemming from the simple trapezoidal rule. The numerical behaviour of the formulas is experimented.

1. Introduction. A common practice in numerical integration is to estimate the error of an integration formula by using a more accurate one. A generalization of this procedure leads to define a family $I^{(1)}, \ldots, I^{(n)}$ of integration formulas with increasing precision to be used in sequence. The application of the formula $I^{(k)}$ does not waste the functional values previously computed for $I^{(k-1)}$ only if the nodes of $I^{(k-1)}$ are a subset of the nodes of $I^{(k)}$. E.g. this property holds for Gauss-Kronrod and Clenshaw-Curtis rules but not for Gauss rules.

Another tool to improve the error bounds (especially useful when the integrand function has low continuity degree) is the use of compound rules.

In this case the functional values are wasted unless the nodes of the formulas have special geometric properties. This happens for Newton-Cotes rules and Romberg schemata but not for Gauss-Kronrod, Clenshaw-Curtis and Gauss rules.

In this work we study families $I^{(1)}, \dots, I^{(n)}$ of integration formulas with the following properties:

a) the nodes of $I^{(k-1)}$ are a subset of the nodes of $I^{(k)}$;

b) composition does not imply wasting of functional values;

c) the degree of precision of $I^{(k)}$ is greater enough than the one of $I^{(k-1)}$;

d) the distance between nodes is shorter near the boundary of the interval;

e) all the formulas stem from the simple trapezoidal formula (i.e. $I^{(1)}$ is the 2-point Newton-Cotes formula);

f) all the formulas in the family are numerically stable, i.e. all the weights are positive.

In section 2 the families of integration formulas (called *recursive monotone*) with properties (a)-(e) are studied, a compact notation to describe the position of the nodes is presented and it is shown how to generate algorithmically all these families.

In section 3 the algebraic properties of recursive monotone formulas are studied and an exhaustive search shows that only 27 families exist which satisfy property (f). Moreover theoretical estimates for the truncation error are given.

In section 4 numerical experiments allow estimating the behaviour of the new formulas.

2. Geometric properties of the nodes. Since we are dealing with interpolatory formulas, symmetric in the interval $[-1,1]$, a formula is completely determined once given the set of nodes in the interval $[0,1]$. Henceforth formulas will be represented by the set of their nodes in $[0,1]$.

Let $Q = \{ x_i \mid 0 \le x_1 < \dots < x_m \le 1 \}$ represent the formula under consideration. We define

$$L(Q) = \{ x \in Q \mid x < 1/2 \},$$

$$R(Q) = \{ x \in Q, x \ge 1/2 \};$$

moreover if $g : \mathbf{R} \to \mathbf{R}$ is a real function, then we define

$$g(Q) = \{ g(x) \mid x \in Q \}.$$

$\mathbf{Z}^+$ denotes the set of nonnegative integers.

Definition 2.1. Given a formula Q the *father* of Q is defined as

(2.1) $\quad F(Q) = 2\,R(Q) - 1;$

moreover the following notation is used

$$F^0(Q) = Q;$$

$$F^k(Q) = F(\,F^{k-1}(Q)\,), \ \ k \in \mathbf{N}.$$

Let $\ell(x) = 2x-1$ and let $\ell^1(x) = \ell(x)$; $\ell^k(x) = \ell\,(\ell^{k-1}(x)\,)$.

Definition 2.2. A formula Q is called *recursive* if $Q = \{1\}$ or the following properties hold:

(2.2) $\quad 1 - 2\,L(Q) \subseteq F(Q)$;

(2.3) $\quad F(Q)$ is recursive;

(2.4) $\quad F(Q) \subseteq Q.$

Let Q be recursive and consider the class of formulas $F^k(Q)$, $k \in \mathbf{N}$, then property (2.4) guarantees that the nodes of $F^k(Q)$ are re-usable for $F^{k-1}(Q)$. Moreover, let $n \times Z$ denote the compound formula which arises when the interval $[-1,1]$ is subdivided into n parts and the Z formula is applied to each subinterval. Then property (2.2) and Definition 2.1 guarantee that the compound formulas $2\times Q$ and $2\times F(Q)$ make use of all the nodes of Q. More generally if Q is recursive the $2^k \times F^i(Q)$, $i= 0,1,\ldots, k$, formulas make use of all the nodes of Q. Note that for any recursive formula $x_m = 1$.

Definition 2.3. Let

$$\Delta_i = x_{i+1} - x_i, \quad i=1,2,\ldots, m-1.$$

Then a formula Q is called *monotone* if $\Delta_{i-1} \geq \Delta_i$ for all the Δ_j which are defined.

Example 2.1. The integration rule $Q = \{0, 1/2, 3/4, 1\}$ is recursive and monotone. Indeed from Definition 2.1 it follows that

$$F(Q) = \{0,1/2, 1\},$$

$$F^2(Q) = \{0, 1\},$$

$$F^k(Q) = \{1\}, \quad k \geq 3,$$

and properties (2.2), (2.3), (2.4) can be easily verified. Note that $F(Q)$, $F^2(Q)$ and $F^3(Q)$ are the Newton-Cotes formulas on $[-1,1]$ with 5, 3, and 2 points, respectively.

In the following some theorems will be given in order to characterize the recursive and monotone formulas.

THEOREM 2.1. *Let* Q *be a recursive formula. Let* $x\in Q$, *let* $k\in \mathbf{N}$ *be such that* $1-2^{-k+1} \leq x < 1- 2^{-k}$, *then the point symmetric to* x *with respect to* $1-2^{-k}$ *belongs to* Q.

Proof. Case (a) : $x \in L(Q)$, i.e. $k=1$. From property 2.2 and Definition 2.1 it follows that $1 - x \in Q$.

Case (b): $x \in R(Q)$, i.e. $k>1$. From the definition of $F^{k-1}(Q)$ it follows that

$\ell^{k-1}(x) \in L(F^{k-1}(Q))$ and, as in case (a), we have $y = 1-\ell^{k-1}(x) \in F^{k-1}(Q)$. Moreover the point $z\in R(Q)$ such that $\ell^{k-1}(z)=y$ is symmetric to x with respect to $1-2^{-k}$. □

THEOREM 2.2. *Let* Q *be a recursive monotone formula*, $Q \neq \{1\}$. *Let* $x=\max\{x_i\in Q \mid x_i\neq 1\}$, *and let* $h\in Z^+$ *such that* $1-2^{-h+1}< x \leq 1-2^{-h}$. *Then*

(i) all the points $y_j = 1-2^{-j}$, $j = 0,1,\ldots, h$ are in Q,

(ii) $x_i = 1-\alpha_i\, 2^{-h}$, $\alpha_i \in Z^+$,

(iii) $\Delta_i \leq 2\,\Delta_{i+1}$.

(iv) $\Delta_i = 2^{-\beta_i}$, $\beta_i \in Z^+$.

Proof. (i). We can see that x has to be equal to $1-2^{-h}$. If $x=0$ then $h=0$ and this fact is trivial, otherwise, if $x\neq 1-2^{-h}$ then from Theorem 2.1 a point $z\in Q$ exists, for which $z\neq 1$, $z>x$. Moreover, from Definition 2.1 and property (2.4), all the points $y_j = 1-2^{-j}$, $j = 0, 1,\ldots, h$, result to belong to Q, and $y_0 = 0 = x_1$.

(ii) We prove by induction the following assertion:

$$A(k):\qquad Q \cap (1-2^{-h+k}, 1) \subseteq \{x = 1-i2^{-h}, i \in Z^+\}.$$

For $k=0$, from the definition of h, $Q \cap (1-2^{-h}, 1) = \emptyset$ and A(0) is true. Assume A(k) then, from Theorem 2.1, A(k) implies A(k+1). Hence A(h) is true and the property (ii) follows.

(iii) Suppose $\Delta_i > 2\Delta_{i+1}$. From (i) it follows that an integer k exists such that $1-2^{-k} \leq x_i < x_{i+1} \leq 1-2^{-k-1}$. Then, because the formula is monotone, the distance between two consecutive nodes of Q in the set $[1-2^{-k-1}, 1]$ is strictly less than $\Delta_i/2$. From Definition 2.1 and property (2.4) we have

$$Q \cap [1-2^{-k}, 1] \supseteq F(Q) \cap [1-2^{-k}, 1] = \ell(Q \cap [1-2^{-k-1}, 1])$$

hence the distance between two consecutive nodes of Q in the set $[1-2^{-k}, 1]$ is strictly less than Δ_i and this leads to deduce $\Delta_i < \Delta_i$ which is a contradiction.

(iv) It follows directly from (i), (ii) and (iii) since, if $\Delta_i > \Delta_{i+1}$ then $\Delta_i = 2\Delta_{i+1}$. □

Theorem 2.1 allows introducing a compact notation to describe recursive monotone formulas. Indeed, the set of differences between consecutive nodes $\{\Delta_i\}$ completely characterizes a monotone formula Q. Moreover if Q is recursive monotone, $Q \neq \{1\}$, the lengths of the intervals Δ_i are powers of 1/2 and the formula can be determined by counting the number of intervals of length 1, 1/2, 1/4 , ... 2^{-h} , respectively.

COROLLARY 2.1. *Any recursive monotone formula* $Q \neq \{1\}$, *can be represented by an array* $(\alpha_0, \alpha_1, \dots, \alpha_h)$ *where* $\alpha_i \in Z^+$ *denotes the number of intervals of length* 2^{-i}, *and* $\sum_{i=0}^{h} \alpha_i 2^{-i} = 1$. □

COROLLARY 2.2. *The nodes of a recursive monotone formula* $(\alpha_0, \alpha_1, \dots, \alpha_h)$ *are machine numbers on a binary arithmetic with at least h bits in the mantissa.* □

Note that with our notation monotone non-recursive formulas can be denoted as well.

Example 2.2. The array (0,1,2) represents the recursive monotone integration rule Q = {0, 1/2, 3/4, 1}. The array (0,0,1,2,8) represents a monotone integration rule which is not recursive.

THEOREM 2.3. *Let Q be a recursive monotone formula*, $Q \neq \{1\}$, $Q \neq \{0, 1\}$.

Let $L(Q) = \{ x_i \mid i=1,2, \ldots, k\}$. *Then* $\Delta_k \leq \Delta_1 \leq 2\, \Delta_k$.

Proof. Since the formula is monotone we have $\Delta_{k+1} \leq \Delta_k \leq \Delta_1$. Moreover, from Definition 2.1 and property (2.4) it follows $\Delta_1 \leq 2\, \Delta_{k+1}$, then $\Delta_1 \leq 2\, \Delta_k$. □

From Theorems 2.2 and 2.3 it follows that in a recursive monotone formula the distances between consecutive points belonging to $L(Q)$ can assume only two different values.

Definition 2.4. Given a recursive monotone formula Q the set $\mathbb{S}(Q)$ of the "sons" of Q is defined as

$$\mathbb{S}(Q) = \{Z \mid F(Z)=Q \text{ , } Z \text{ is recursive monotone}\}.$$

THEOREM 2.4. *Let* Q *be a recursive monotone formula. Let* $Z = Q \cup (Q+1)/2$, *then* (i) $Z \in \mathbb{S}(Q)$,

(ii) $Y \in \mathbb{S}(Q)$ implies $Z \subseteq Y$.

Proof. (i) from the definition of Z it follows that $F(Z) = Q$, then Z is a recursive monotone formula since $1 - 2L(Z) = 1 - 2L(Q) \subseteq F(Q) \subseteq Q$.

(ii) from $Q = \mathbb{L}(R(Y))$ and $Q = \mathbb{L}(R(Z))$ we have $R(Y) = R(Z)$, moreover $L(Z)=L(Q)$ and $Q \subseteq Y$ imply that $L(Z) \subseteq L(Y)$. □

Theorem 2.3 and 2.4 allow us to generate algorithmically all the families of recursive monotone formulas starting from $Q = \{1\}$. Indeed the set $\mathbb{S}(Q)$ consists of Z and all the formulas obtained from Z by halving the longest intervals in $L(Z)$ according to Theorem 2.3 and Definition 2.3.

Example 2.3. Let $Q=(0,0,3,1,2)$, (i.e. $Q=\{0,\ 1/4,\ 1/2,\ 3/4,\ 7/8,\ 15/16,\ 1\}$). Then $\mathbb{S}(Q) = \{Z, Y_1, Y_2\}$ with $Z = (0,0,2,3,1,2)$, $Y_1 = (0,0,1,5,1,2)$ and $Y_2 = (0,0,0,7,1,2)$.

3. Error estimates. The set of the weights of an interpolatory formula Q can be obtained by solving an appropriate linear system [1, pp.76]. It is well known that formulas having both positive and negative weights are less favorable for what concerns the roundoff error in the numerical computation of the integral. If all the weights are positive the roundoff error can grow, at worst, as the number of nodes [1, pp.276]. This fact suggests the following definition

Definition 3.1. A recursive monotone formula Q is called *stable* if $Q=\{1\}$ or both the following properties hold:

(3.1) all the weights of Q are positive;

(3.2) F(Q) is stable.

All the recursive monotone stable formulas stemming from $Q=\{1\}$ (henceforth denoted as RMS formulas) have been generated by exhaustive search. They form a finite tree with 74 elements and 27 leaves. Table I lists all these formulas. RMS formulas are identified by a symbolic name given by the number of the nodes of the formula in [-1,1] followed by a letter when two formulas with the same number of nodes exist. The tree of the symbolic names is plotted in Fig.1.

Definition 3.2. Let $f(x) \in C^m[-1,1]$ and let $I_n(f)$ be the linear functional

$I_n(f) = \sum_{i=1}^{n} w_i f(x_i)$, the *truncation error* of I_n is the linear functional

$$E_n(f) = \int_{-1}^{1} f(x)\,dx - I_n(f).$$

It is well known from the theory that if n is odd and I_n is a symmetric interpolatory formula then $E_n(p) = 0$ for all the polynomials p(x) of degree

Table I

name	formula	name	formula
2	Newton-Cotes 2-point rule	69B	(0,0,0,0,6,15,8,3,2)
3	(1)	71A	(0,0,0,0,4,20,6,3,2)
5	(0,2)	71B	(0,0,0,0,5,17,8,3,2)
7	(0,1,2)	79	(0,0,0,0,3,20,10,3,1,2)
9	(0,1,1,2)	81	(0,0,0,0,2,22,10,3,1,2)
11	(0,0,3,2)	83	(0,0,0,0,2,22,9,5,1,2)
13	(0,0,3,1,2)	85A	(0,0,0,0,1,24,9,5,1,2)
15	(0,0,2,3,2)	85B	(0,0,0,0,2,21,11,5,1,2)
17	(0,0,2,3,1,2)	87	(0,0,0,0,1,23,11,5,1,2)
19	(0,0,1,5,1,2)	89	(0,0,0,0,0,25,11,5,1,2)
21	(0,0,0,7,1,2)	91	(0,0,0,0,0,24,13,5,1,2)
23	(0,0,0,6,3,2)	93	(0,0,0,0,0,23,15,5,1,2)
25	(0,0,0,6,3,1,2)	97	(0,0,0,0,0,23,14,6,3,2)
27	(0,0,0,5,5,1,2)	99	(0,0,0,0,0,22,16,6,3,2)
29	(0,0,0,4,7,1,2)	111	(0,0,0,0,0,19,20,10,3,1,2)
31	(0,0,0,4,6,3,2)	113	(0,0,0,0,0,18,22,10,3,1,2)
33	(0,0,0,3,8,3,2)	115	(0,0,0,0,0,18,22,9,5,1,2)
35	(0,0,0,2,10,3,2)	117A	(0,0,0,0,0,17,24,9,5,1,2)
37	(0,0,0,2,10,3,1,2)	117B	(0,0,0,0,0,18,21,11,5,1,2)
39A	(0,0,0,1,12,3,1,2)	119	(0,0,0,0,0,17,23,11,5,1,2)
39B	(0,0,0,2,9,5,1,2)	121	(0,0,0,0,0,16,25,11,5,1,2)
41	(0,0,0,1,11,5,1,2)	123	(0,0,0,0,0,15,27,11,5,1,2)
43	(0,0,0,0,13,5,1,2)	125	(0,0,0,0,0,15,26,13,5,1,2)
45	(0,0,0,0,12,7,1,2)	127	(0,0,0,0,0,14,28,13,5,1,2)
47	(0,0,0,0,12,6,3,2)	129	(0,0,0,0,0,13,30,13,5,1,2)
49	(0,0,0,0,11,8,3,2)	137	(0,0,0,0,0,12,31,14,6,3,2)
53	(0,0,0,0,10,10,3,1,2)	139	(0,0,0,0,0,11,33,14,6,3,2)
55A	(0,0,0,0,9,12,3,1,2)	141	(0,0,0,0,0,10,35,14,6,3,2)
55B	(0,0,0,0,10,9,5,1,2)	143	(0,0,0,0,0,9,37,14,6,3,2)
57	(0,0,0,0,9,11,5,1,2)	175	(0,0,0,0,0,2,46,22,9,5,1,2)
59	(0,0,0,0,8,13,5,1,2)	177	(0,0,0,0,0,1,48,22,9,5,1,2)
61	(0,0,0,0,7,15,5,1,2)	179	(0,0,0,0,0,1,48,21,11,5,1,2)
63	(0,0,0,0,7,14,7,1,2)	181A	(0,0,0,0,0,0,50,21,11,5,1,2)
65A	(0,0,0,0,5,19,5,1,2)	181B	(0,0,0,0,0,1,47,23,11,5,1,2)
65B	(0,0,0,0,7,14,6,3,2)	185	(0,0,0,0,0,1,45,27,11,5,1,2)
67	(0,0,0,0,6,16,6,3,2)	201	(0,0,0,0,0,0,44,31,14,6,3,2)
69A	(0,0,0,0,5,18,6,3,2)	249	(0,0,0,0,0,0,30,54,21,11,5,1,2)

less than n+1. Assume m=n and $f^{(n+1)}(x)$ piecewise continuous, in this case, from the Peano's Theorem [1, pp. 285-292] the truncation error can be bounded as follows

$$|E_n(f)| \le \frac{c_{n+1}}{(n+1)!} \max_{-1 \le x \le 1} |f^{(n+1)}(x)|,$$

where

$$c_{n+1} = (n+1) \int_{-1}^{1} \left| \frac{(1-t)^{n+1}}{n+1} - \sum_{x_i > t} w_i (x_i - t)^n \right| dt.$$

This error estimate is the best possible one in the sense that there exists a function f(x) for which the equality sign holds. The computation of the constants c_{n+1} which depend only on the integration formula is a hard task from the numerical point of view.

The weights of RMS formulas have been computed with extended precision arithmetic (28 hexadecimal digits mantissa) and the resulting accuracy has been checked, achieving at least 25 exact decimal digits for all formulas except the last one (249 points) for which only 19 decimal digits resulted to be exact. The computation of the constants c_{n+1} can be carried out only for the formulas with no more than 47 points. In fig.2 the values of $-\log_{10} c_{n+1}$ for RMS formulas, Clenshaw-Curtis and Newton-Cotes formulas, are plotted versus n (n<49).

The behaviour of constants c_{n+1} suggests an alternative approach to select among recursive monotone formulas. Let Q be a recursive monotone formula with n nodes and let c(Q) denote the associate constant c_{n+1}, we define the class of formulas showing the fastest descent of the constants c_{n+1}.

FIG. 1

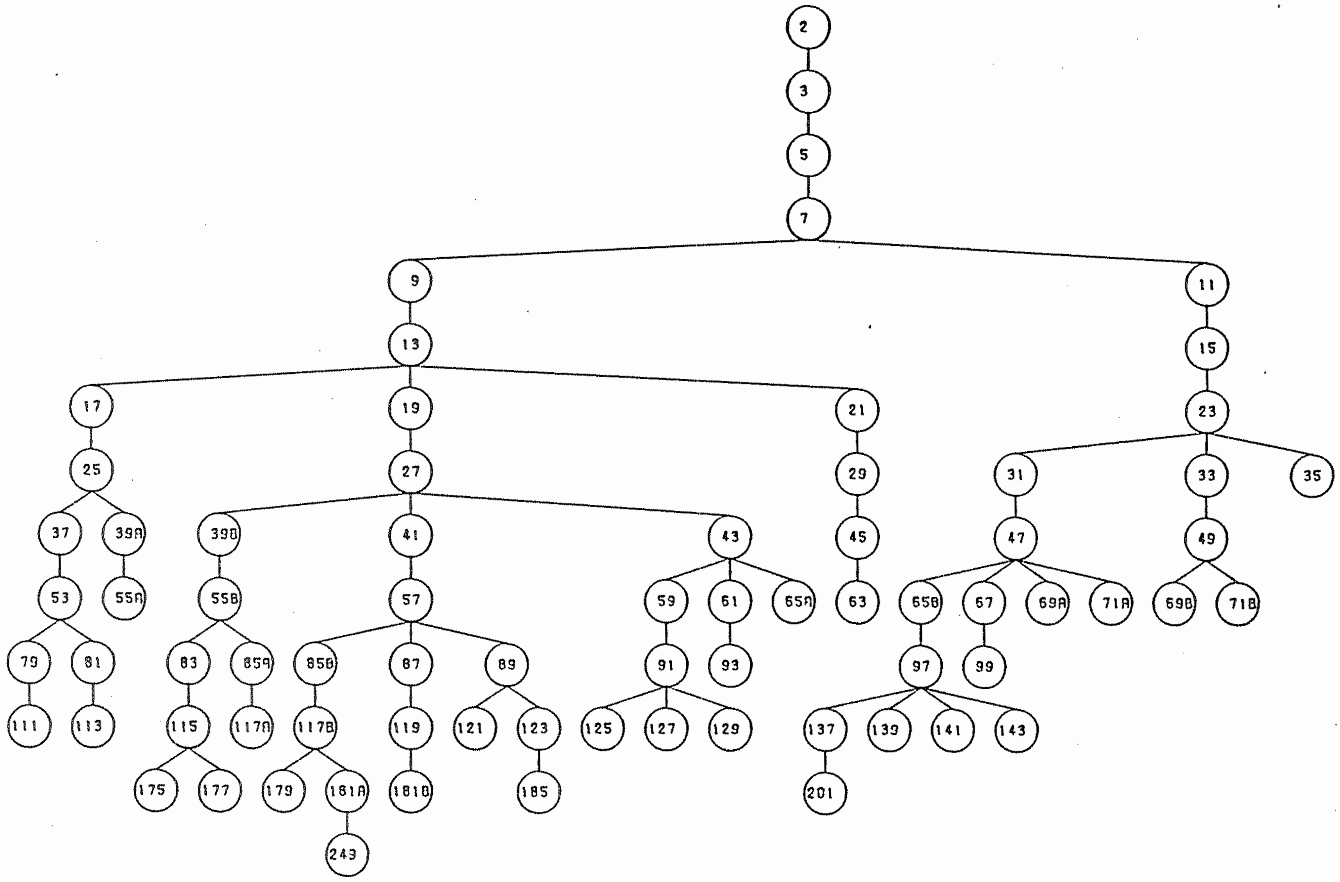

Fig. 1. Tree of RMS formulas stemming from simple trapezoid rule.

FIG.2

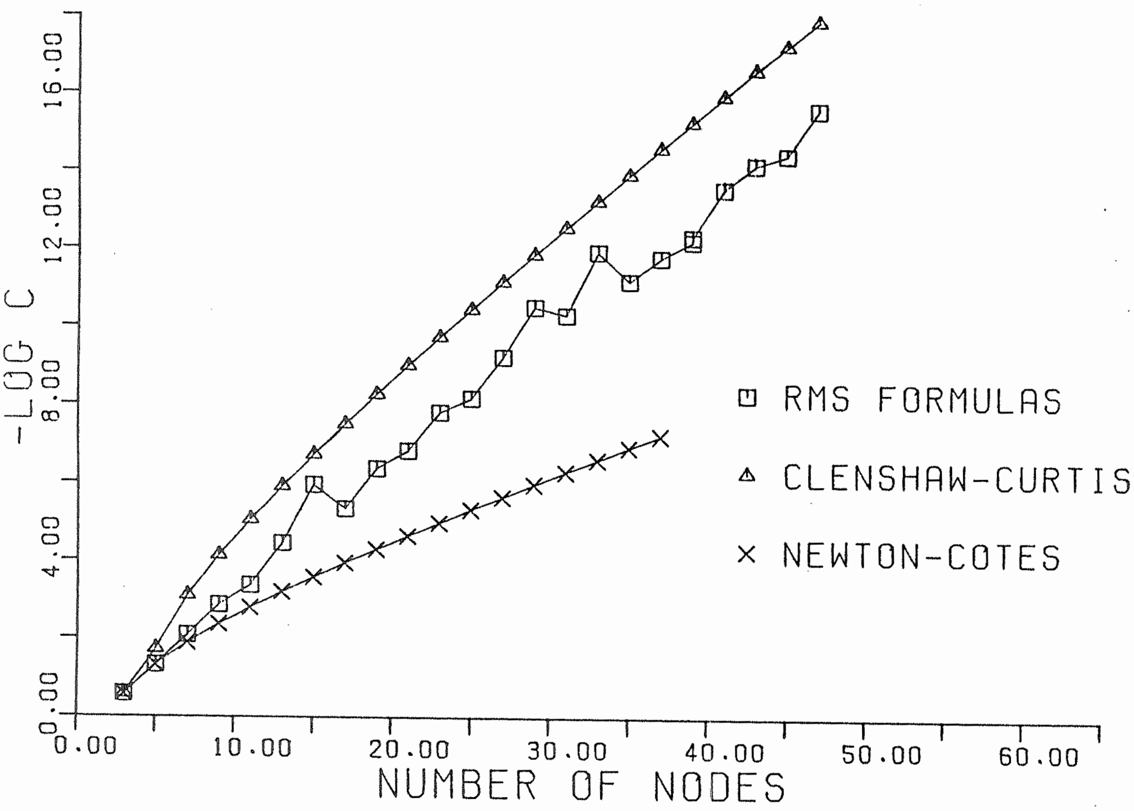

Fig. 2. Values of $-\log_{10} c_{n+1}$ for RMS, Clenshaw-Curtis and Newton-Cotes formulas, plotted versus n.

Definition 3.3. A recursive monotone formula Q is called *accurate* if Q={1} or both the following properties hold:

(3.3) $c(Q) = \min\{c(Z) \mid Z$ recursive monotone having no more points than $Q\}$

(3.4) $F(Q)$ is accurate.

All the recursive monotone accurate formulas stemming from Q={1} having no more than 47 nodes have been generated by exhaustive search. They form a tree with 16 elements and 4 leaves. Table II lists all these formulas, it is interesting to note that all accurate formulas have positive weights and are stable. Table III presents the statistic of formulas having no more than 43 nodes.

4. Numerical experiments. All the RMS formulas were tested with six problems:

$$\int_0^1 f_k(x)\, dx\ ,\quad k=1,2, \dots 6,$$

were f_1, f_2, f_3 are smooth functions, namely

$$f_1 = 1/(1+25x^2)\ , \qquad f_2 = \exp(5x)\ , \qquad f_3 = (1/251)\, x^{250},$$

and f_4, f_5, f_6 are non-smooth ones:

$$f_4 = x^{1/2}, \qquad f_5 = x^{-1/2}\, dx\ , \qquad f_6 = x \ln x\ .$$

(Functions 1,2,4,5,6 were used for numerical experiments in [3]). For each formula Q_n and for each function f_k the absolute and relative errors are computed using system IBM /370 double precision arithmetic (u = machine precision = $16^{-13} \approx 2.22\ 10^{-16}$)

Table II

name	formula	name	formula
2	Newton-Cotes 2-point rule	19	(0,0,1,5,1,2)
3	(1)	21	(0,0,0,7,1,2)
5	(0,2)	23	(0,0,0,6,3,2)
7	(0,1,2)	27	(0,0,0,5,5,1,2)
9	(0,1,1,2)	29	(0,0,0,4,7,1,2)
11	(0,0,3,2)	33	(0,0,0,3,8,3,2)
13	(0,0,3,1,2)	41	(0,0,0,1,11,5,1,2)
15	(0,0,2,3,2)	43	(0,0,0,0,13,5,1,2)

Table III

Statistics of formulas having no more than 43 nodes

RM (Recursive Monotone formulas)	606
RMP (Recursive Monotone formulas having positive weights)	29
RMS (Recursive Monotone stable formulas)	23
RMA (Recursive Monotone accurate formulas)	16

Note: $RMA \subset RMS \subset RMP \subset RM$

$$E_{abs}(Q_n,f_k) = \int_0^1 f_k(x)\,dx - I_{computed}, \qquad E_{rel}(Q_n,f_k) = E_{abs}(Q_n,f_k) \,/\, (\int_0^1 f_k(x)\,dx).$$

We define the quantities

$$\eta_{n,k} = \mathrm{Max}\,(\,u, \mathrm{Min}\,(1, |E_{rel}(Q_n,f_k)|\,)\,)$$

and compute the mean logarithmic error:

$$E(n) = (1/6) \sum_{k=1}^{6} -\log_{10} \eta_{n,k}.$$

E(n) is plotted in Fig.3 for all RMS formulas and some Clenshaw Curtis formulas.

In order to test how RMS formulas can be suitable for automatic adaptive quadrature, the family of RMS formulas with 2, 3, 5, 7, 9, 13, 19, 27, 41, 57, 85(b), 117(b), 181(a), 249 nodes, respectively, was used into a preliminary version of an automatic quadrature routine called QIC (**Q**uadrature **I**nterpolatory and **C**ompound). QIC uses essentialy the same strategy of QAGS [8], the main difference consists in trying to use as long as possible high precision local quadrature formulas before splitting subintervals. To avoid wasting of functional evaluations an hashing technique is used as suggested in [4]. QIC, CADRE [2] and QAGS were tested for the same function $1/(1+c+x)$ used in [6] to compare SQUANK [5], SQUAGE [6], CADRE and AIND [7]. Fig.4.a shows the logarithm of the number N of functional evaluations for various values of c, Fig 4.b shows log N for c=1.01 and various values of the absolute tolerance *eps*. By comparing the results with those of [6], it is self evident that for this example QIC outperforms all the other quadrature routines.

FIG. 3

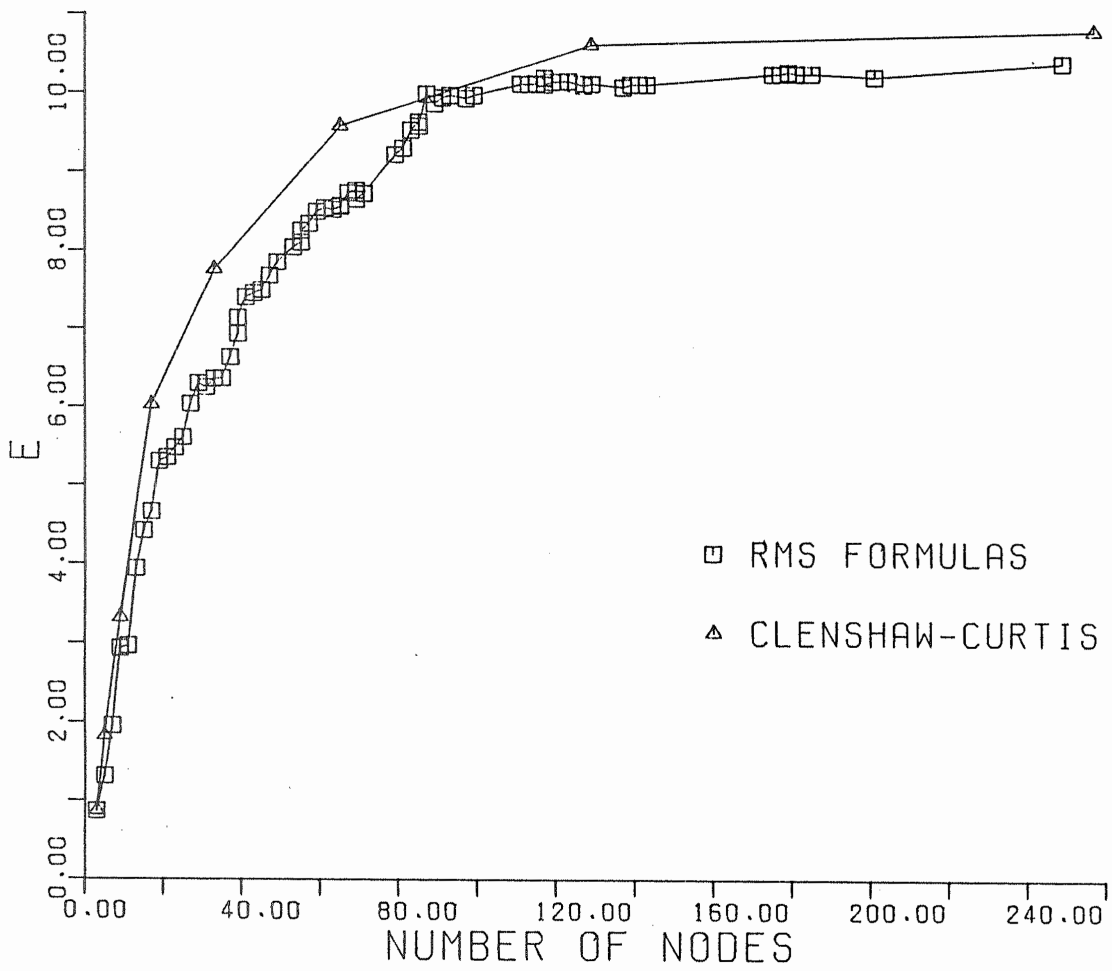

Fig. 3. Values of $E(n)$ for RMS and Clenshaw-Curtis formulas, plotted versus n.

FIG. 4

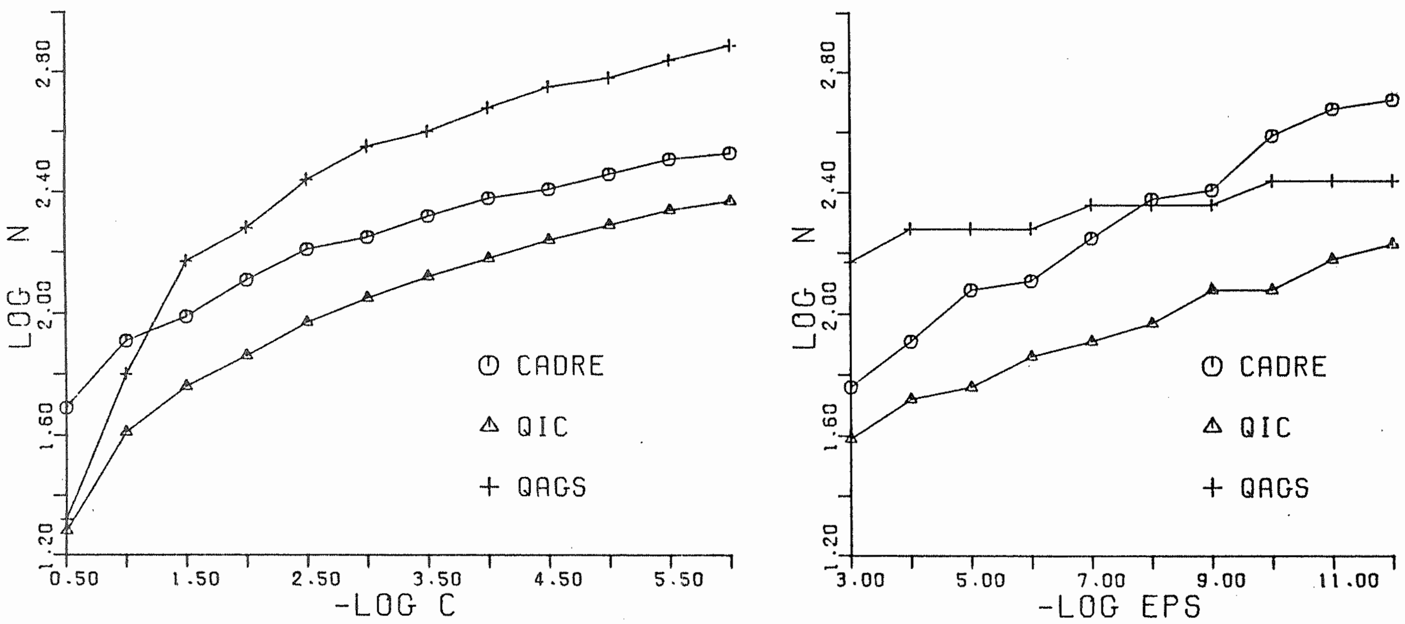

Fig. 4. (a) values of $\log_{10} N$ for QIC, CADRE and QAGS, plotted versus $-\log_{10} c$, for *eps* = 10^{-6}; (b) values of $\log_{10} N$ for QIC, CADRE and QAGS, plotted versus $-\log_{10}$ *eps*, for c =0.01.

Istituto di Elaborazione dell'Informazione - CNR,
Via S.Maria 46, PISA, ITALY.

Dipartimento di Informatica,
Universita' di Pisa,
Corso Italia 40, PISA, ITALY.

Istituto di Elaborazione dell'Informazione - CNR,
Via S.Maria 46, PISA, ITALY.

1. P. J. Davis, P. Rabinowitz, *Methods of Numerical Integration*. Academic Press, New York, 1984.
2. C. DeBoor, "CADRE: an Algorithm for Numerical Quadrature," in *Mathematical Software*, (J. Rice, ed.), Academic Press, New York, 1971, pp. 417-449.
3. D. Kahaner, J. Stoer, "Extrapolated Adaptive Quadrature," *SIAM J. Sci. Stat. Comput.*, v. 4, 1983, pp. 31-44.
4. D. K. Kahaner, M. B. Wells, "An experimental Algorithm for N-Dimensional Adaptive Quadrature," *ACM Trans. Math. Software*, v. 5, 1979, pp. 86-96.
5. J. N. Lyness, " SQUANK (Simpson Quadrature Used Adaptively-Noise Killed)," Algorithm 379, *Comm. ACM*, v. 13, 1970, pp. 260-263.
6. M. A. Malcom, R. B. Simpson, "Local Versus Global Strategies for Adaptive Quadrature," *ACM Trans. Math. Software*, v. 1, 1975, pp. 129-146.
7. R. Piessens, "An Algorithm for Automatic Integration," *Angewandte Informatik*, 1973, pp. 399-401.
8. R. Piessens et al., *QUADPACK a Subroutine Package for Automatic Integration*, Springer-Verlag, Berlin, 1983.