Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione

Asymptotic behaviour of automatic quadrature

G. Di Marco, P. Favati, G.Lotti
e F. Romani

Nota Interna IEI-B4-35
settembre 1992

Via S. Maria, 46 - 56126 Pisa (Italy)
Phone : +39.(0)50.553159
Fax : +39.(0)50.554342
Telex : 590305 IEICNR I

Asymptotic Behaviour of Automatic Quadrature

GASPARE DI MARCO, PAOLA FAVATI[1], GRAZIA LOTTI[2]
AND FRANCESCO ROMANI[3]

[1]*IEI-CNR Via S. Maria 46, 56100 Pisa, Italy*
[2]*Dipartimento di Matematica, University of Trento, 38050 Povo (TN), Italy.*
[3]*Dipartimento di Informatica, University of Pisa, Corso Italia 40, 56100 Pisa, Italy*

Running Head: Automatic Quadrature

Preferred address for reply:

Prof. Dr. Francesco Romani
Dipartimento di Informatica,
University of Pisa,
Corso Italia 40,
56100 Pisa, Italy

E-mail address: romani.di.unipi.it

ABSTRACT

The complexity of automatic quadrature programs is investigated under the hypothesis of exactness or asymptotical consistence of local error estimates. The complexity measure used is the number N of functional evaluations in real exact arithmetic versus the number E of exact decimal digits in the result. The methods of integration analysed are m-panel rules, Clenshaw-Curtis quadrature, Romberg method, global adaptive quadrature, and double adaptive quadrature. For m-panel and global adaptive quadrature, based on a local rule of degree r-1 the constants hidden by the "Oh" notation are determined in terms of the r-th derivative of the integrand and the numerical properties of the chosen local rule. The complexity of global adaptive quadrature results to be of order $\Theta(10^{E/r})$, regardless of the regularity of the integrand. The double adaptive quadrature achieves O(E) complexity for regular integrands and $O(E^2)$ for singular ones.

1. INTRODUCTION

The problem

We study the asymptotic complexity of computing the exact real representation of the definite integral

$$(1.1) \qquad I = \int_a^b f(x)\, dx,$$

by using an automatic integration routine and infinite precision arithmetic.

The complexity measure investigated is the number N of functional evaluations as a function of the number of exact decimal digits in the result. More precisely, once given an algorithm that computes (1.1) with any accuracy, the k-th step of the integration process produces a value $I^{(k)}$ that approximates the integral with an absolute error $e^{(k)} = I - I^{(k)}$. Let $E^{(k)} = - \mathrm{Log}_{10} | e^{(k)} |$ and let $N^{(k)}$ be the overall number of functional values computed from step 1 to step k. The pairs $(E^{(k)}, N^{(k)})$, k=1,2,... can be viewed as instances of a function N(E), whose asymptotical behaviour is investigated here. From the point of view of the numerical analyst it is more natural to measure the error $e^{(k)}$ as a function of the number of evaluations $N^{(k)}$; both the measures can be easily converted one into the other, and, in Section 6, the complexity results proved in this work, are summarised in both the notations.

Moreover it is worth noting that, to estimate the computational load of the algorithm, a more detailed analysis would be required, which takes into account the cost

of the arithmetic operations, the complexity of the integrand and the overhead due to the algorithm itself.

In Traub *et al.* (1988) the information-based complexity of the problem of integrating functions belonging to a given class is studied and optimal algorithms, with respect to a complexity measure that is consistent with our one, are presented. However, our approach is quite different since we study the computational complexity of several adaptive and non adaptive practical algorithms on a large selection of integrand classes.

Section 2 is devoted to preliminaries: the main notations, the basic assumptions and the classes of integrands taken into consideration are introduced. In Section 3 and 4 the complexities of non-adaptive quadrature and of classical adaptive quadrature are investigated. In Section 5 a double-adaptive quadrature algorithm, which achieves outstanding performance, is presented. In Section 6 the complexity bounds are summarised. Some auxiliary results are given in the Appendix.

2. PRELIMINARIES

Notation

Given two real functions $f(x)$, $g(x)$ and $x_0 \in \mathbb{R} \cup \{+\infty\}$

$f(x) = O(g(x))$	means that	$\limsup_{x \to x_0} \lvert f(x)/g(x) \rvert = c,\ c \in \mathbb{R};$
$f(x) = o(g(x))$	means that	$\lim_{x \to x_0} \lvert f(x)/g(x) \rvert = 0;$
$f(x) = \Omega(g(x))$	means that	$g(x) = O(f(x));$
$f(x) = \Theta(g(x))$	means that	$f(x) = O(g(x))$ and $f(x) = \Omega(g(x));$
$f(x) \sim g(x)$	means that	$\lim_{x \to x_0} \lvert f(x)/g(x) \rvert = 1;$
$f(x) \lesssim c\, g(x)$	means that	$\limsup_{x \to x_0} \lvert f(x)/g(x) \rvert \le c,\ c \in \mathbb{R}.$

In the following the point x_0 will be implicitely defined by the context.

Classes of integrands

A detailed analysis of quadrature algorithms can be made only under precise assumptions on the nature of the integrand. Clearly, if the integration interval is split into a finite number of subintervals and the same algorithm is applied to each of them, the order of the overall complexity is that attained on the worst subinterval. This

consideration allows reducing the case of integrands with a finite number of singularities to the case of integrands with only one singularity in the integration interval. It is remarkable that many automatic integration algorithms (namely global adaptive ones, see Malcolm and Simpson (1975), Piessens (1973), Piessens *et al.* (1983)) perform such a splitting during the integration process.

In the following we study the asymptotic complexity for three classes of integrands:

1) analytical functions in the integration interval.

2) bounded functions with only one singularity in the integration interval. These functions are assumed to be analytical in the other points of the integration interval. If the singularity is located in a point y of the form $a + (b-a)\, p\, 2^{-k}$ with p and k integer, and a global algorithm based on bisections is used, the singularity will be located at the endpoint of two adjacent subintervals during the integration process (this kind of points will be called *reachable*). Integration of functions with singularities in reachable points is automatically reduced to the case of two integrations with one endpoint singularity, and sometimes the singularity is removed.

3) unbounded functions with only one singularity at one extreme of the integration interval. Also these functions are assumed to be analytical in the other points of the integration interval, moreover a (reasonable) assumption is made on the rate of growth of the absolute value of the function near the singularity. These functions can be integrated by an automatic integration routine, by ignoring the singularity (Davis and Rabinowitz, 1984, page 180). The singularity must be confined in one extreme of the integration interval (or equivalently in a reachable point), otherwise it would be impossible to bound the values of function on the interval and, consequently, the errors in the integration process.

More precisely let us define $\Gamma(a,b,\delta)$ as the region of the complex plane obtained as the closed union of two semicircles of radius δ and centres at a and b respectively and the rectangle of vertices $(a,\pm\delta)$, $(b,\pm\delta)$; $\partial\Gamma$ denotes the boundary of Γ.

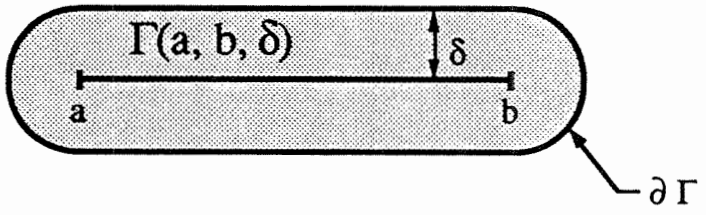

Fig. 2.1. The region $\Gamma(a,b,\delta)$.

We define the following classes

Class $\mathcal{A}$[a,b,δ] (Analytical): Analytical functions in [a,b], continuable analytically to be single valued and regular in $\Gamma(a,b,\delta)$, $\delta>0$.

Examples:

F0: e^x in $[-1,\pi/2]$, (analytical on the complex plane);
F1: $\sin(x)$ in $[-1,\pi]$, (analytical on the complex plane);
F2: $(1+x^2)^{-1}$ in $[-1,\pi]$, (analytical on $\Gamma(-1,\pi,0.99)$);
F3: $x \cos 100\, x^2$ in $[0,1]$, (oscillating, analytical on the complex plane);
F4: $(10^{-4}+(x-1)^2)^{-1}$ in $[0,\pi]$, (peaked, analytical on $\Gamma(0,\pi,0.0099)$)

Class $\mathcal{B}$[a,b,y,s,δ] (Singular-bounded). A function f in $\mathcal{B}$[a,b,y,s,δ] has the following properties:

1) f is bounded in [a,b];
2) f is bounded and continuable analytically both in $\Gamma(a,y,\delta_1) \cap \{Re(z)<y\}$ and in $\Gamma(y,b,\delta_2) \cap \{Re(z)>y\}$, with $0 < \delta = \min[\delta_1,\delta_2]$, $y \in [a,b]$;
3) if s=-1, a step singularity in y is present;
   if $s\geq 0$, f is in C^s[a,b] with one singularity on the (s+1)-th derivative in y.

An interesting special case is that of piecewise polynomials, i.e. functions of the type

$$f(x) = \begin{cases} p(x), & x \leq y \\ q(x), & x > y \end{cases}$$

to be integrated in [a,b], $a\leq y\leq b$, where p(x) and q(x) are polynomials. In this case, if the rule used is enough precise, the only source of error is the singularity.

Examples:

F5: $x^{3/2}$ in [0,1], (singularity on the second derivative, F5 $\in \mathcal{B}$[0,1,0,1,δ], $\delta>0$).

F6: $\begin{cases} (x-1)^2 & x\leq 1 \\ -(x-1)^2 & x>1 \end{cases}$ in $[0,\pi]$; (F6 $\in \mathcal{B}$[0,π,1,0,δ], $\delta>0$).

F7: $\begin{cases} \sin 10\, x, & x \leq 1, \\ 2\, e^x, & x > 1, \end{cases}$ in $[0,\pi]$; (F7 $\in \mathcal{B}$[0,π,1,-1,δ], $\delta>0$).

F8: $\begin{cases} -e^x, & x \leq 0, \\ e^x, & x > 0, \end{cases}$ in $[-1,\pi/2]$; (F8 $\in \mathcal{B}$[-1,π/2,0,-1,δ], $\delta>0$).

Class $\mathcal{U}[a,b,\alpha,\delta]$ (Unbounded): Analytical functions in (a,b], continuable analytically in $\Gamma(a,b,\delta)\cap\{Re(z)>a\}$, such that

$$\limsup_{x\to a^+} |f(x)| = \infty,$$

with the additional property that, for any $0 < \varepsilon < 1-\alpha$,

$$|f(z)| = O(|z-a|^{-\alpha-\varepsilon}), \qquad 0<\alpha<1, \qquad z\in\Gamma(a,b,\delta)\cap\{Re(z)>a\}.$$

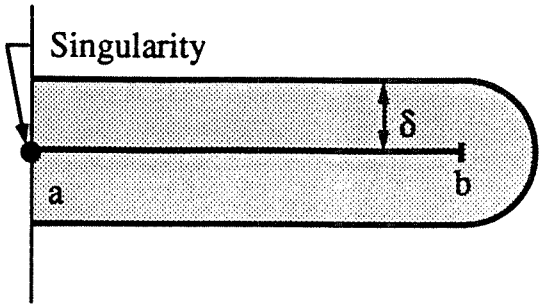

Fig. 2.2. The region $\Gamma(a,b,\delta)\cap\{Re(z)>a\}$.

Examples:

F9: $x^{-1/2}\,\mathrm{Log}\,x$, in $[0,\pi]$, $(F9 \in \mathcal{U}[0,\pi,1/2,\delta],\ \delta>0)$.

F10: $x^{-1/4}(3/4\,\mathrm{Cos\,Log}\,x - \mathrm{Sin\,Log}\,x)$ in $[0,1]$, $(F10 \in \mathcal{U}[0,1,1/4,\delta],\ \delta>0)$.

The assumptions of our ideal model

The analysis of the asymptotical complexity of numerical quadrature is, in general, an hard task without some simplifying assumptions. The critical problems that need to be simplified are the roundoff error management and truncation error estimates, e.g. see Kahaner and Stoer (1983). We make the following two main assumptions.

Assumption 2.1. We assume to use exact real arithmetic, or equivalently, a multiple precision arithmetic with a variable number of digits sufficient to neglect roundoff errors. ❑

Assumption 2.2. The local error bound e_{est} used in the programs is exact, i.e.

$$e_{est} = e_{true}.$$ ❑

These assumptions force us to study ideal programs and the results are not directly applicable to practical environments, but this approach has the following advantages:

1) the analysis of automatic quadrature algorithms with exact arithmetic and exact error bounds allows investigating the problems related to the integration strategy (choice of the rules, type of algorithm, etc.,) without the influence of the error estimate techniques.

2) all the questions concerning fails, successes and quit can be discarded.

Moreover the performance of local error estimate techniques is commonly investigated by comparing the performances of the same quadrature algorithm with several local error estimate routines, getting a relative rating of the techniques. The performance of algorithms with exact error bounds, as studied here, is in some sense the optimal case, and this allows an absolute rating of practical strategies.

In the Appendix, it will be proved that for enough regular functions Assumption 2.2 is asymptotically true; namely, for interpolatory rules of degree of precision r-1, applied on a subinterval of length 2λ to $C^{r'+2}$, $r'>r$, functions, there exists a simple error estimate such that:

$$e_{est} = e_{true} + O(\lambda^{r'+1}),$$

where

$$e_{true} = C\ \lambda^{r+1} + O(\lambda^{r+3}), \qquad C \text{ constant independent of } \lambda.$$

This is in some sense an extension of the concept of asymptotical consistence of local error estimates given in Kahaner and Stoer (1983).

Number of evaluations

In the context of automatic quadrature programs, the composition arises often by bisections: a particular sequence of subintervals is obtained by repeated bisection of intervals. The total number of evaluations depends on the geometrical properties of the used rules. There exist rules that allow reusing the previously computed functional values: the most common examples are Newton-Cotes rules, but they can be used only with small degree. RMS rules (Favati *et al.*, 1991a) have the same geometrical property and can be applied with a high number of points. For our purposes it is sufficient to observe that in any case the number of functional evaluations N is proportional to the number of nodes n of the used rule and the number m of subintervals:

$$N = v_1\, m + v_0,$$

e.g., for Newton-Cotes rules $v_0 = 1$, $v_1 = n-1$, for Gauss-Legendre and Gauss-Kronrod rules $v_0 = -n$, $v_1 = 2n$.

Algorithm presentation

The quadrature algorithms are presented in the form of Pascal-like programs, the following procedure heading is used

```
procedure Quadrature(a, b, epquad: real; var  abserr, result: real);
```

where

on input	a, b	are the extremes of the integration interval;
	epquad	is the absolute error tolerance specified by the user.
on output	abserr	is the absolute error estimate returned by the program
	result	is the integral value estimated by the program

The quadrature programs use a local routine

```
procedure rule(a, b: real; k: integer; var  abserr, result: real);
```

that computes:

result	an approximation of the integral obtained by applying the k-th element of a family of rules Q_1,..., Q_k, ... of increasing accuracy;
abserr	the absolute value of the exact error.

The algorithms are presented in a very schematic form, without any roundoff checking or exceptional conditions handling, according to Assumptions 2.1 and 2.2.

Graphical presentations of numerical experiments

To improve the clarity of the presentation, the results of numerical experiments will be shown by means of two types of graphics. The plot of the complexity function allows comparing the overall behaviour of different algorithms or of the same algorithm on different integrands (Fig 2.3).

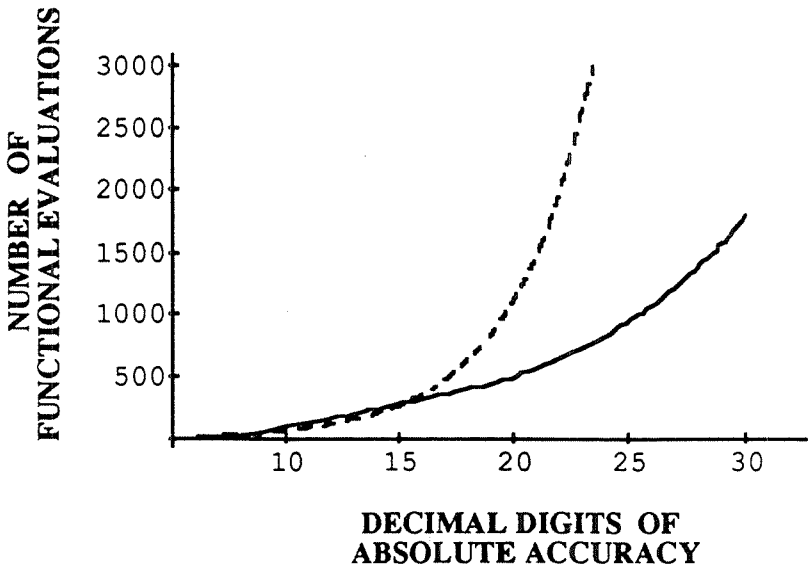

Fig. 2.3. Plot of the complexity of two quadrature routines.

Another type of graph allows a remarkable insight into the structure of the computation of a program based on the composition of elementary rules. Two quantities are plotted in decimal logarithmic scale versus the integration interval: the number of exact digits in any subinterval (black points) and the density of evaluation in the subintervals (grey points). Moreover the number of decimal digits of total accuracy, the total number of functional evaluations and the number of subintervals are printed at the top of the graph.

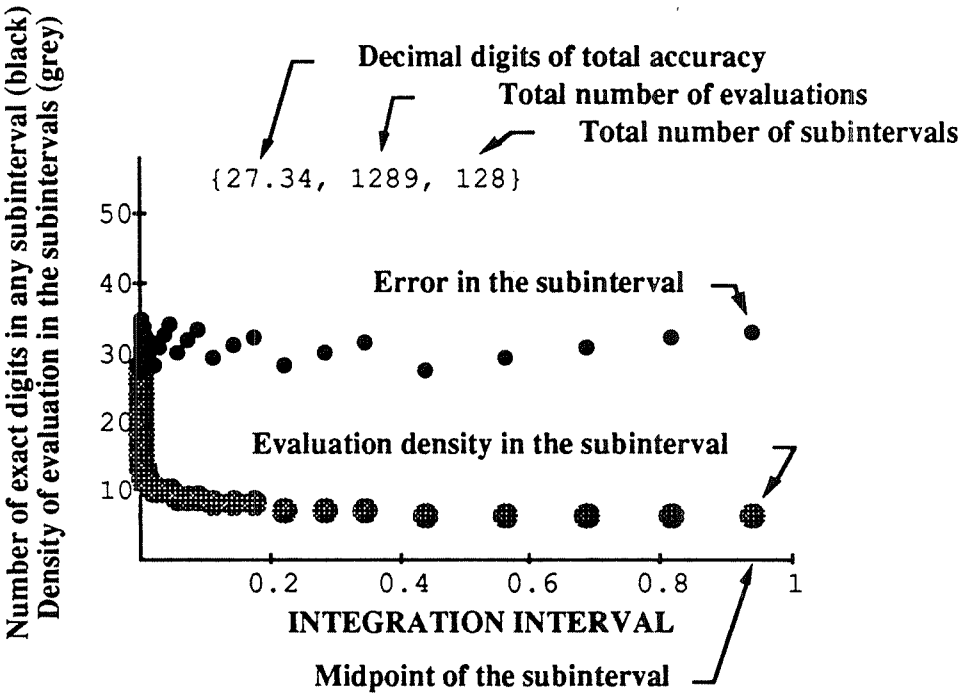

Fig. 2.4. Plot of the status of a quadrature routine.

All the computations were carried out with the help of *Mathematica*™ on an Apple® Macintosh™ and a workstation Sun™, using 70 decimal digits arithmetic.

Clenshaw-Curtis quadrature

Algorithm 3.1 below outlines a non-adaptive automatic quadrature routine based on a sequence of different rules of increasing precision. Probably, the best choice is the family of Clenshaw-Curtis formulas (Clenshaw and Curtis, 1960) which can be computed, with any number of nodes, via the fast Fourier transform algorithm without precomputing weights and nodes, (Gentleman, 1972).

Algorithm 3.1:

```
procedure CCQuadrature(a, b, epquad: real; var abserr, result: real);
     var k: integer;
{    procedure CCrule(a,b: real; k: integer; var abserr, result: real);          }
{              implements Clenshaw-Curtis Quadrature with 2^k+1 points           }
begin
     k:=1;
     CCrule(a, b, k, abserr, result);
     while abserr > epquad do begin
          k:=k+1;
          CCrule(a, b, k, abserr, result);
     end;
end;
```

❑

The number of evaluations after k steps is

$$N = 2^k + 1.$$

This algorithm converges very well for regular functions. For any function f in $\mathcal{A}[a,b,\tau(b-a)]$, with some $\tau>0$, using relation (A.6) of the Appendix we get the bound:

$$|I - I^{(k)}| \leq C\,\rho^{-(N+2)}, \qquad \rho > 1 + 2\tau,$$

where C is independent of N. Then

$$E = -\mathrm{Log}_{10}|I - I^{(k)}| \geq (N+2)\ \mathrm{Log}_{10}\rho - \mathrm{Log}_{10} C,$$

whence

(3.1) $$N = O(E).$$

Non adaptive composition of a fixed rule

Algorithm 3.2 below outlines a generic non-adaptive automatic quadrature routine based on panel rules. This schema will be called PNAQ in the following.

Algorithm 3.2:

```
procedure PNAQuadrature(a, b, epquad: real; var:  abserr, result: real);
     var i,n: integer;
          x,y,z,h:real;
{         k: integer        is a global variable denoting the rule used }
begin
     n:=1;
     repeat
          x:=a;
          h:=(b-a)/n;
          result := 0;
          for i:=1 to n do begin
               rule(x, x+h, k, z, y);
               result:=result+y;
               x:=x+h;
          end;
          abserr:=    a global estimate of the total error
          n:=2*n
     until abserr <  epquad
end;
```

❑

Let us discuss the complexity of this method for a $C^s[a,b]$, $s \geq 1$, function. Let r-1 be the degree of precision of the rule Q, $q=\min[r,s]$; let v_0 and v_1 be the evaluation coefficients for the rule Q and let 2^k be the number of subintervals. The number of evaluations is

$$N = 2^k v_1 + v_0. \tag{3.2}$$

Applying relation (A.4) of the Appendix to each subinterval of length $h = 2^{-k}(b-a)$ we get:

$$|I - I^{(k)}| \leq (b-a)^{q+1} 2^{-q(k+1)-1} c_q \max_{x \in [a,b]} \frac{|f^{(q)}(x)|}{q!}, \tag{3.3}$$

that is

$$E = -\operatorname{Log}_{10} |I - I^{(k)}| \geq k\, q \operatorname{Log}_{10} 2 + C_1, \qquad C_1 \text{ independent of } k. \tag{3.4}$$

Combining (3.2) and (3.4) we get

$$N = O(10^{E/q}).$$

The same result can be trivially derived for functions in the classes $\mathcal{A}[a,b,\delta]$ or $\mathcal{B}[a,b,y,s,\delta]$, $s \geq 1$, with $q=r$ or $q=\min[r,s]$, respectively.

For $C^{r+2}[a,b]$ functions a more accurate result can be derived by applying Theorem A.1 to each subinterval:

$$I - I^{(k)} = \left(\frac{h}{2}\right)^{r+1} \frac{\beta_r}{r!} \sum_{i=1}^{2^k} f^{(r)}(a + (i - \frac{1}{2})\, h) + O(h^{r+2}).$$

Since

$$\sum_{i=1}^{2^k} f^{(r)}(a + (i - \frac{1}{2})\, h) = h^{-1} \int_a^b f^{(r)}(x)\, dx + O(h)$$

we have

$$(3.5) \qquad I - I^{(k)} = 2^{-kr} \left(\frac{b-a}{2}\right)^r \frac{\beta_r}{2\,r!} [f^{(r-1)}(b) - f^{(r-1)}(a)] + O(2^{-k(r+2)}).$$

It is remarkable that (3.5) can be derived from Theorem 2.9 of Piessens *et al.* (1983), page 45, using a simple relation between the coefficient β_r and the error of integrating the r-th Bernoulli polynomial by using rule Q_r.

By using (3.5), if $\int_a^b f^{(r)}(x)\, dx \neq 0$, we get:

$$10^{-E} \sim 2^{-kr} \left| \left(\frac{b-a}{2}\right)^r \frac{\beta_r}{2\,r!} [f^{(r-1)}(b) - f^{(r-1)}(a)] \right|$$

and

$$(3.6) \qquad N \sim v_1 \left| \frac{b-a}{2} \right| \left| \frac{\beta_r}{2\,r!} [f^{(r-1)}(b) - f^{(r-1)}(a)] \right|^{\frac{1}{r}} 10^{E/r}.$$

The behaviour of PNAQ for the analytical function (F1: sin(x)) can be exemplified with the help of the following graph.

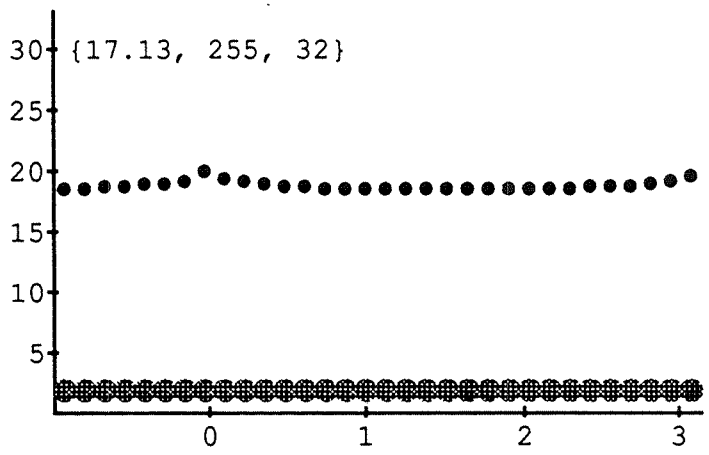

Fig. 3.1. Sine function, Clenshaw-Curtis 7-point, 32 subintervals.

It is remarkable that, if the function presents some difficulties, the non adaptive programs are not able to change strategy in order to improve the error behaviour near the difficulty. The following example shows the behaviour of PNAQ, after a

subdivision into 64 subintervals, for the function F4 that presents a large peak at the point x=1 and complex poles at z = 1 ± i /100.

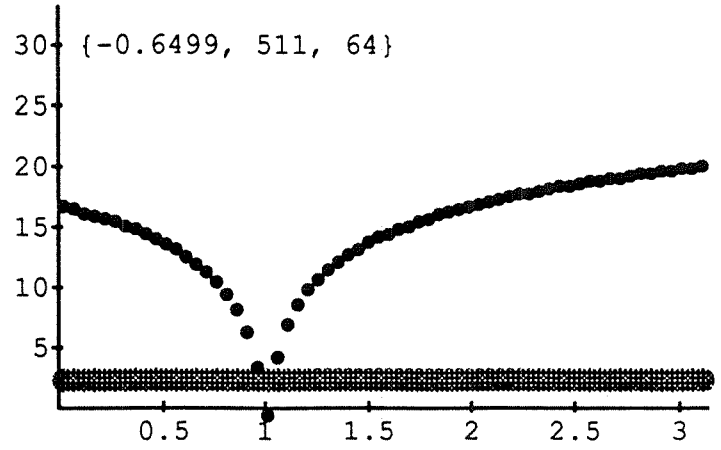

Fig. 3.2. Peak function, Clenshaw-Curtis 7-point, 64 subintervals.

Romberg extrapolation

Romberg quadrature is a well-known non-adaptive quadrature algorithm based on the application of Richardson's extrapolation to Euler-MacLaurin sum formula. The algorithm is presented here in the classical form, as described in Davis and Rabinowitz, (1984), (page 435). Assume a=0, b=1, and let f be a $C^{2m+1}[0,1]$ function. Let

$$T_0^{(0)} = \frac{1}{2}\,[f(0) + f(1)],$$

$$T_0^{(k)} = \frac{1}{2n}\,[f(0) + 2\sum_{i=1}^{n-1} f(\frac{i}{n}) + f(1)], \qquad n=2^k,$$

denote the trapezoidal sums. The extrapolated values $T_m^{(k)}$ are defined by

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}\,, \quad k>0,$$

and they form the so called T-table:

$$\begin{array}{cccc} T_0^{(0)} & & & \\ T_0^{(1)} & T_1^{(0)} & & \\ T_0^{(2)} & T_1^{(1)} & T_2^{(0)} & \\ T_0^{(3)} & T_1^{(2)} & T_2^{(1)} & T_3^{(0)} \\ \ldots & & & \end{array}$$

Every term in the table is an approximation of the integral, the terms in the i-th row are computed using 2^i+1 functional values and the terms on the j-th column have degree of precision 2j+1. Indeed $T_m^{(k)}$ is essentially the integral approximated by a symmetrical interpolatory rule of degree 2m+1 computed on $2^{(k+m)}$ points. For any function f analytical in $\mathcal{A}[0,1,\delta]$ with some $\delta>0$, the error bound for the terms in the diagonal of the T-table is given by

$$|I - T_m^{(0)}| \leq C_1 \, | B_{2m+2} | \, 2^{-m(m+1)} \, \delta^{-2m},$$

(see Davis and Rabinowitz,1984, page 436).

where C_1 is independent of m and the constants B_{2j} are the Bernoulli numbers: $B_2=1/6$, $B_4=-1/30$, $B_6=1/42$, ... Taking the negative logarithm to base 10 we get

$$E^{(m)} \geq C_2 + m(m+1) \, \mathrm{Log}_{10} 2 + 2m \, \mathrm{Log}_{10} \delta - \mathrm{Log}_{10} | B_{2m+2} |.$$

With elementary calculus it is possible to prove that there exist constants m_0 and C_3 for which

$$E^{(m)} \geq C_3 \, m^2, \qquad \text{for } m>m_0,$$

that is

$$C_4 \sqrt{E^{(m)}} \geq m, \qquad \text{for } m \geq m_0.$$

Since $N = 2^m$ we get $N = O(10^{C\sqrt{E}})$.

Euler-MacLaurin formula only applies to smooth integrands. It is possible to devise error expansions that work also for algebraic and logarithmic endpoints singularities (Lyness and Ninham, 1967). In these cases the ε-algorithm (Wynn, 1956, Piessens *et al.*, 1983) can be applied obtaining an asymptotic complexity similar to that of Romberg extrapolation also for some function in class $\mathcal{U}$.

4. GLOBAL ADAPTIVE QUADRATURE SCHEMATA

Global adaptive composition of a fixed rule

The first schema of automatic adaptive quadrature we consider (called GQ in the following) is essentially the algorithm used in QAG (Piessens *et al.*, 1983), and QXG (Favati *et al.*, 1991b) and consists of repeatedly dividing the worst subinterval until the total error estimate is better than the user requested tolerance. The essential information on the subintervals is stored in a priority queue, ordered according the error estimate,

and accessed by the routines put_interval and get_interval. We assume that the routine

```
procedure rule(a, b :real; k: integer; var abserr, result: real)
```

implements a symmetric interpolatory rule Q with degree of precision r-1, and evaluation coefficients v_0 and v_1.

Algorithm 4.1:

```
procedure GQuadrature (a,b,epquad: real; var  abserr,result: real);
     var  res,est,res1,est1,res2,est2,middle: real;
{          k: integer        is a global variable denoting the rule used }
begin
     rule(a,b,k,abserr,result);              { integrate on the first interval }
     put_interval(a,b,abserr,result);        { put the interval in the queue }
     while abserr > epquad do begin
          get_interval(a,b,est,res);         { get the interval with the largest error }
          middle:=(a+b)/2                    { perform bisection }
          rule(a,middle,k,est1,res1);
          rule(middle,b,k,est2,res2);
          put_interval(a,middle,est1,res1);
          put_interval(middle,b,est2,res2);
          result:=result-res+res1+res2;          { update estimates }
          abserr:=abserr-est+est1+est2;
     end;
end;
```

❑

The algorithm is based on the estimate abserr of the quantity $\sum_{i=1}^{m} |e^{(i)}|$ which, in general, is greater than or equal to $|I-I_{computed}|$. The asymptotic complexity is then studied by investigating the relationship between the number of functional evaluations N and the quantity $E = -\text{Log}_{10}$ abserr in the infinite integration process (carried with exact real arithmetic) produced by a zero error requirement.

Note that, if f is a polynomial of degree less than r, only one interval suffices to exactly compute the integral, and the complexity is O(1), otherwise

i) any subinterval with non-zero error will be halved in the integration process;
ii) subintervals with exactly zero error do not further influence the integration process;
iii) the number m of subintervals increases of one unity at each bisection.

Analysis of GQ for C^{r+2} *functions*

Let us discuss the complexity of GQ for $C^{r+2}[a,b]$ functions. The following analysis is applicable to functions in the classes $\mathcal{A}[a,b,\delta]$ or $\mathcal{B}[a,b,y,s,\delta]$, with $s \geq r+2$.

We denote with $\mathcal{R}(m)$ the set of the m subintervals in the queue at the m-th step of the integration process. Let $R = [x-\lambda,x+\lambda] \in \mathcal{R}(m)$ be a subinterval of length 2λ with

a non zero associated error e, let $R_1, R_2 \in \mathcal{R}(m+1)$ be the subintervals obtained by halving R and let e_1, e_2 be the errors of the integrals of f on R_1 and R_2, respectively. In virtue of Theorem A.1, since $|f^{(r+2)}(x)|$ is bounded on [a,b], we have, apart from numerical coincidences

$$(4.1) \qquad e \sim \lambda^{r+1} f^{(r)}(x) \frac{\beta_r}{r!},$$

then from Theorem A.5 it results

$$(4.2) \qquad e_1 = 2^{-(r+1)} e + O(\lambda^{r+2}), \qquad e_2 = 2^{-(r+1)} e + O(\lambda^{r+2}).$$

The above relations mean that, typically, when a sufficiently small subinterval is halved, two subintervals are generated for which the error is reduced by a factor close to $2^{-(r+1)}$.

Let the integration process be forwarded enough to have m_0 subintervals sufficiently small to neglect the $O(\lambda^{r+2})$ terms in (4.2). Let $e^{(i)}$, i=1,2,...,m_0 be the errors associated to the subintervals and let P be the better subinterval (with non zero error) in $\mathcal{R}(m_0)$, i.e.

$$|e_P| = \min_{\substack{1 \le i \le m_0 \\ e^{(i)} \neq 0}} |e^{(i)}|.$$

Let m_1 be the number of subintervals when P is halved. All the other subintervals in $\mathcal{R}(m_1)$ have been generated by bisections of subintervals having an error with absolute value greater than $|e_P|$. Then, applying Theorem A.5, for any $R \in \mathcal{R}(m_1)$ of length λ we have

$$(4.3) \qquad 2^{-(r+1)} |e_P| (1 + O(\lambda)) \le |e_R| \le |e_P|,$$

Let $$e_{MIN} = \min_{\substack{1 \le i \le m \\ e^{(i)} \neq 0}} |e^{(i)}|, \qquad e_{MAX} = \max_{1 \le i \le m} |e^{(i)}|,$$

then, for $m = m_1$ relation (4.3) implies

$$e_{MAX} \le 2^{r+1} e_{MIN} (1 + O(\lambda)).$$

Observe that the above relation holds at any successive step of the integration process. The proof follows by induction on the number m of subintervals starting from m_1 and using relation (4.2). This allows us to write

$$(4.4) \qquad e_{MAX} \lesssim 2^{r+1} e_{MIN}.$$

In the graphical representation, this means that most black points lie in a strip large (r+1) $Log_{10}\,2$ decimal digits, (see Fig. 4.1, below).

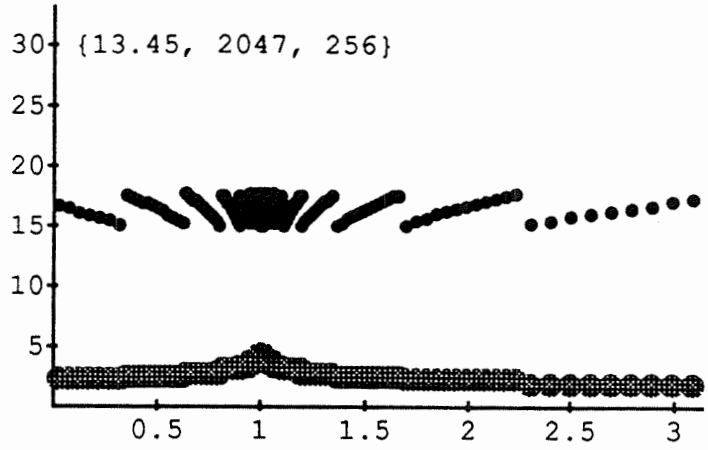

Fig. 4.1. Peak function, Clenshaw-Curtis 7-point, 256 subintervals.

Let consider a point u belonging to the integration interval; let $2\lambda(u)$ and e(u) denote the length and the associated error of the subinterval containing the point u at some step of the integration process. Relation (4.1) can be written as

$$e(u) \sim \lambda(u)^{r+1}\, f^{(r)}(y(u))\, \frac{\beta_r}{r!},$$

where y(u) denotes the midpoint of the subinterval containing u.

Since $y(u) \to u$ as $\max_{u\in[a,b]} \lambda(u) \to 0$ then we can write

$$e(u) \sim \lambda(u)^{r+1}\, f^{(r)}(u)\, \frac{\beta_r}{r!}, \qquad \text{as } \max_{u\in[a,b]} \lambda(u) \to 0,$$

and therefore

$$\lambda(u) \sim \left(|e(u)| \,/\, |f^{(r)}(u)|\, \frac{|\beta_r|}{r!} \right)^{\frac{1}{r+1}}.$$

The density of subintervals is $(2\lambda(u))^{-1}$ and the number m of subintervals in the region T=[a,b] is

$$(4.5) \qquad m = \frac{1}{2}\int_T \lambda(u)^{-1}\, du \sim \frac{1}{2}\int_T \left(|f^{(r)}(u)|\, \frac{|\beta_r|}{r!} \,/ |e(u)| \right)^{\frac{1}{r+1}} du \le C_T\, e_{MIN}^{-\frac{1}{r+1}},$$

where

$$(4.6) \qquad C_T = \frac{1}{2}\left(\frac{|\beta_r|}{r!}\right)^{\frac{1}{r+1}} \int_T |f^{(r)}(u)|^{\frac{1}{r+1}}\, du$$

is a constant depending on Q, f and T.

Since

$$10^{-E} = \text{abserr} = \sum_{i=1}^{m} |e^{(i)}| = \frac{1}{2} \int_T |e(u)|\, \lambda(u)^{-1}\, du,$$

we have

$$(4.7) \quad 10^{-E} \sim \frac{1}{2} \int_T |e(u)|^{\frac{r}{r+1}} \left(|f^{(r)}(u)| \frac{|\beta_r|}{r!}\right)^{\frac{1}{r+1}} du \leq C_T\, e_{MAX}^{\frac{r}{r+1}},$$

Combining (4.5) and (4.7) and using $N = v_1 m + v_0$ we get:

$$(4.8) \qquad v_1 C_T^{\frac{r+1}{r}} 10^{\frac{E}{r}} \lesssim N \lesssim \left(\frac{e_{MAX}}{e_{MIN}}\right)^{\frac{1}{r+1}} v_1 C_T^{\frac{r+1}{r}} 10^{\frac{E}{r}} \lesssim 2 v_1 C_T^{\frac{r+1}{r}} 10^{\frac{E}{r}}.$$

The proof of the first asymptotic inequality follows from Lemma A2 of the Appendix, the second is trivial, the third follows from (4.4). Finally, (4.8) can be rewritten as:

$$(4.8.a) \qquad \frac{v_1}{2} \left(\frac{|\beta_r|}{2r!}\right)^{\frac{1}{r}} \left[\int_T |f^{(r)}(u)|^{\frac{1}{r+1}} du\right]^{\frac{r+1}{r}} 10^{\frac{E}{r}} \lesssim N$$

$$(4.8.b) \qquad N \lesssim v_1 \left(\frac{|\beta_r|}{2r!}\right)^{\frac{1}{r}} \left[\int_T |f^{(r)}(u)|^{\frac{1}{r+1}} du\right]^{\frac{r+1}{r}} 10^{\frac{E}{r}}$$

i.e.

$$N = \Theta(10^{\frac{E}{r}}).$$

Note that in the special case $f(x) = x^r$, the bounds given by (4.8.a) and that of (3.6) for PNAQ are both equal to

$$v_1 \left|\frac{b-a}{2}\right|^{\frac{r+1}{r}} |\beta_r|^{\frac{1}{r}} 10^{\frac{E}{r}}.$$

The complexity of GQ for the function x^8, integrated in [0,1] with Clenshaw-Curtis 7-point rule is presented in Fig 4.2 compared with the theoretical bounds of (4.8). We note that, since the r-th derivative is a constant, the lower bound is periodically reached when all the intervals are of the same length and the algorithm coincides with PNAQ.

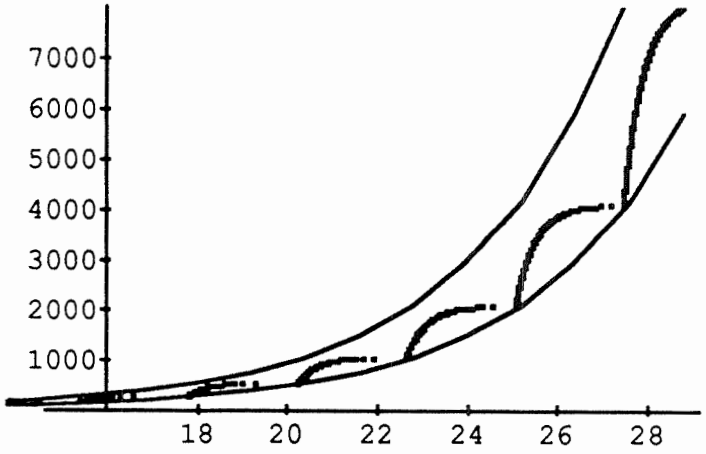

Fig. 4.2. N versus E, compared with the theoretical bounds, program GQ, integrand x^8, Clenshaw-Curtis 7-point rule.

The complexity of a less regular function like exp(20 x), integrated in [0,1] is less oscillating (Fig 4.3).

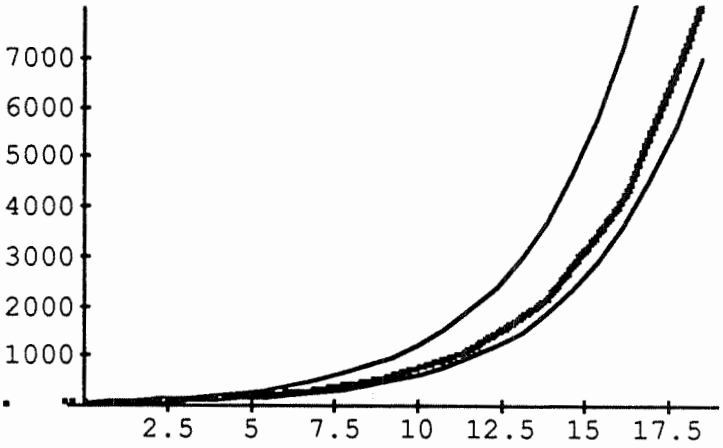

Fig. 4.3. N versus E, compared with the theoretical bounds, program GQ, integrand exp(20 x), Clenshaw-Curtis 7-point rule.

Analysis of GQ for bounded C^s *functions, (s<r+2)*

Let us investigate the complexity of the global adaptive algorithm for a bounded function in the class $\mathcal{B}[a,b,y,s,\delta]$ with $-1 \leq s < r+2$. We divide the subintervals present in the queue, at any time of the integration process, into three classes:

1) the "red" subinterval which contains the singularity,
2) the "white" subintervals,
3) the "black" ones.

The definition of the colour of a subinterval is recursive:

i) at the start of the integration process there is only the red subinterval: [a,b],

ii) the red subinterval is bisected into one red subinterval (containing the singularity) and a white one,
iii) a white subinterval is bisected into two black ones,
iv) a black subinterval is bisected into two black ones.

The meaning of white and black subintervals is clear in the following graph taken after 127 bisections for the step function F7:

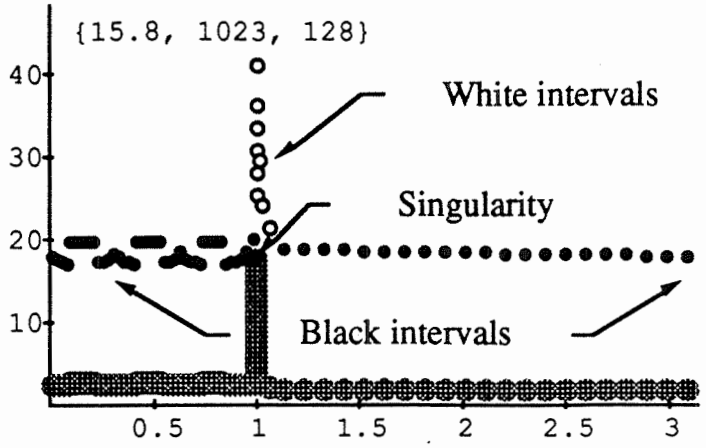

Fig. 4.4. Step function, Clenshaw-Curtis 7-point, 128 subintervals.

We denote with S(i) the red subinterval after i bisections. In the first phase of the quadrature process the error $e_{S(i)}$ on the subinterval S(i) dominates and the red subinterval is divided more and more. When the error on S(i) is less than the error on some white subinterval the latter is divided and black subintervals start to be produced. After this phase the integration on black subintervals proceeds as in the previous case, since their processing is not affected by the presence of the singularity.

Lemma 4.1. The number i of bisections performed to get the red subinterval S(i) is bounded as follows

$$i \leq - \mathrm{Log}_2 e_{MAX} + \mathrm{Log}_2 D.$$

Proof. From Theorem A.2 of the Appendix we have $| e_{S(i)} | \leq B\ (b-a)\ 2^{-i-1}$, i.e.

$$i \leq - \mathrm{Log}_2 |e_{S(i-1)}| + \mathrm{Log}_2 B(b-a)$$

Since $| e_{S(i-1)} | \geq e_{MAX}$ the thesis is proved with D = B(b-a). ❑

Let T be the subset of the integration interval covered by black subintervals and let m_B, m_W be the numbers of black and white subintervals, respectively. Since black subintervals are obtained by halving subintervals that do not contain the singularity, on the set T, relation (4.5) holds for m_B. Moreover there are

$$m_W \leq - \mathrm{Log}_2 e_{MAX} + \mathrm{Log}_2 D$$

white subintervals. Since (4.4) holds on [a,b] - {y} and

$$(4.9) \qquad C_T\, e_{MAX}^{-\frac{1}{r+1}} \lesssim m = m_B + m_w + 1 \lesssim C_T\, e_{MIN}^{-\frac{1}{r+1}} + O(-\operatorname{Log} e_{MAX}),$$

where

$$T \to [a,b] - \{y\} \quad \text{for} \quad \max \lambda(x) \to 0,$$

then the logarithmic term in (4.9) is dominated and the complexity is the same of the regular case (4.8).

It is possible to verify experimentally this fact for the two functions

$$\text{F0: } e^x \qquad \text{and} \qquad \text{F8: } \begin{cases} -e^x, & x \le 0 \\ e^x, & x > 0 \end{cases} \qquad \text{integrated on } [-1,\pi/2],$$

which present the same value of the constant $C_{[-1,\pi/2]}$. Both the functions have been integrated with GQ (using Gauss-Legendre 5 point Rule) until 42 decimal digits of accuracy have been obtained. As a matter of fact, after an initial transitory phase in which the smooth function is integrated much better, both the complexity plots remain into the theoretical limits of (4.8).

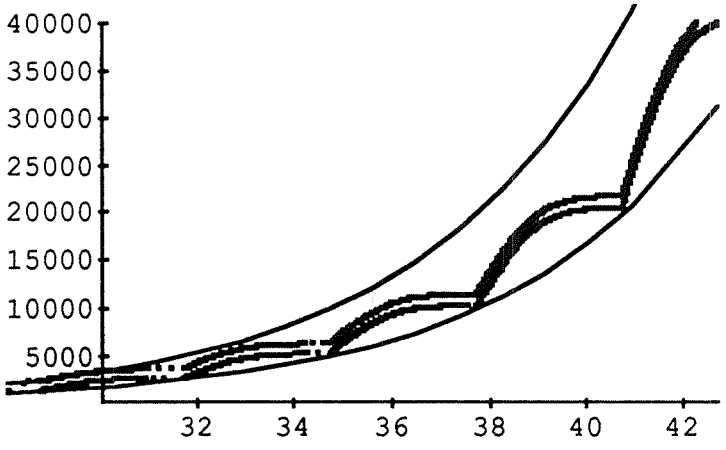

Fig. 4.5. N versus E, (30<E<42), compared with the theoretical bounds, program GQ, Gauss-Legendre 5-point rule, functions F0 and F8.

In the case of piecewise polynomials of degree less than r, there are no black subintervals since the error on white subintervals is zero. In this case the only source of error is the singularity. It is easy to prove that the complexity is linear, i.e. $N=\Theta(E)$. It is remarkable that the linear complexity is attained by GQ for any singular function in the transition phase before black subintervals start to be generated.

Let us investigate now the complexity of the global adaptive algorithm for unbounded function restricting the analysis to a $\mathcal{U}[0,1,\alpha,\delta]$ function f with the additional property

(4.10) $\quad | f^{(r)}(x) | = \Omega(x^{-\alpha+\varepsilon-r}), \quad x>0$, for any $\varepsilon > 0$.

We divide the subintervals present in the queue, at any time of the integration process, into four classes: red, white, grey and black subintervals. The definition of the colour of a subinterval is recursive, once fixed a constant $\varepsilon>0$ with the additional conditions $\varepsilon < \frac{1-\alpha}{3r+4}$.

i) at the start of the integration process there is only the red subinterval: [0,1],
ii) the red subinterval is bisected into one red subinterval (that containing the singularity) and a white one,
iii) a white subinterval is bisected into two grey ones,
iv) a grey subinterval $[x-\lambda, x+\lambda]$, with $\lambda \geq x^{1+3\varepsilon}$ is bisected into two grey ones,
v) a grey subinterval $[x-\lambda, x+\lambda]$, with $\lambda < x^{1+3\varepsilon}$ is bisected into two black ones,
vi) a black subinterval is bisected into two black ones.

Lemma 4.2. The number i of bisections performed to get the red subinterval S(i) is bounded as follows

$$i \leq \frac{-\operatorname{Log}_2 e_{MAX} + \operatorname{Log}_2 B}{1-\alpha-\varepsilon}.$$

Proof. The proof is analogous to that of Lemma 4.1, using Theorem A.3 instead of Theorem A.2. ❑

Theorem 4.1. The numbers m_W and m_G of white and grey subintervals, respectively, are bounded as follows:

$$m_W = O(-\operatorname{Log} e_{MAX})$$

$$m_G = O(e_{MAX}^{-\frac{3\varepsilon}{1-\alpha-\varepsilon}}) = o(e_{MAX}^{-\frac{1}{r+1}}).$$

Proof. The first part of the thesis is trivial. To prove the second part we note that the j-th white subinterval $[2^{-j},2^{-j+1}]$ cannot generate more than $2^{3j\varepsilon-1}$ grey subintervals of semi-length $\lambda>2^{-j(1+3\varepsilon)}$. Then

$$m_G \leq \sum_{j=0}^{i} 2^{3j\varepsilon-1} = 2^{-1}(2^{3(i+1)\varepsilon}-1)/(2^{3\varepsilon}-1) = O(e_{MAX}^{-\frac{3\varepsilon}{1-\alpha-\varepsilon}}). \qquad \square$$

The analysis of the asymptotic complexity can be carried out with a technique similar to the previous one. Let T be the subset of the integration interval covered by black subintervals, relation (4.5) is true for m_B and relation (4.7) holds for the error on T. Applying Theorem A.6 to the black subintervals with the fixed positive constant $\varepsilon < \frac{1-\alpha}{3r+4}$, we can derive relation (4.4) on T. Since

$$m = m_B + m_w + m_G + 1,$$

we have

$$C_T\, e_{MAX}^{-\frac{1}{r+1}} \lesssim \quad m \quad \lesssim C_T\, 2\, e_{MAX}^{-\frac{1}{r+1}} + O(-\operatorname{Log} e_{MAX}) + o(e_{MAX}^{-\frac{1}{r+1}}).$$

Note that, since from Lemma A.1

$$|f^{(r)}(x)|^{\frac{1}{r+1}} = O(x^{-\frac{\alpha+\varepsilon+r}{1+r}}),$$

then $C_{[0,1]}$, defined in (4.6), has a finite value and, for $T \rightarrow (0,1]$, relation (4.8) can be derived again.

This result has been experimentally verified for the singular unbounded function F9: $x^{-1/2} \operatorname{Log} x$, integrated in $[0,\pi]$.

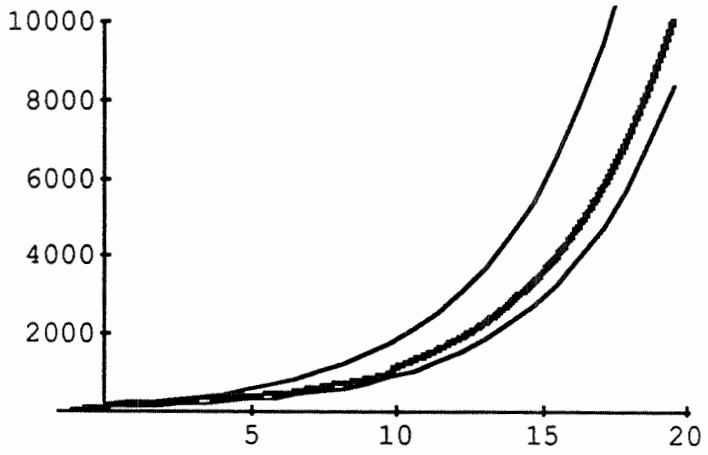

Fig. 4.6. N versus E, compared with the theoretical bounds, program GQ, integrand F9, Gauss-Legendre 5-point rule.

It is interesting to verify experimentally the behaviour of a singular unbounded function not satisfying property (4.10), (F10, integrated in [0,1]).

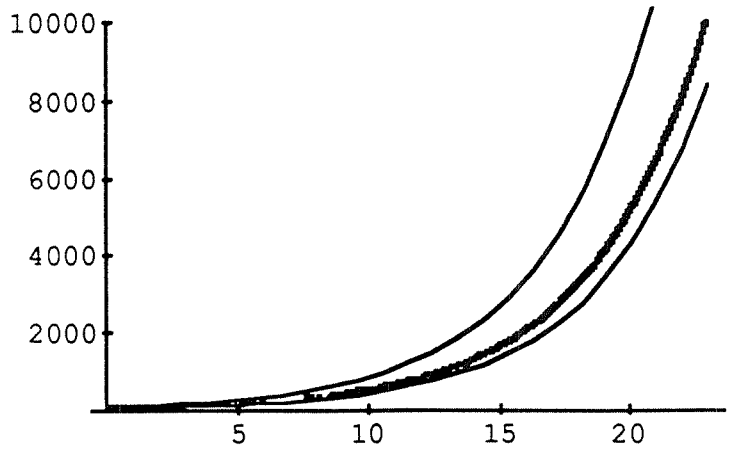

Fig. 4.7. N versus E, compared with the theoretical bounds, program GQ, integrand F10, Gauss-Legendre 5-point rule.

Also in this case the theoretical bounds bracket the complexity; it can be conjectured that hypothesis (4.10) is not needed for (4.8) to hold.

Analysis of GQ for general functions

The main property of global quadrature algorithms is that the integration process is carried out independently on the various subintervals while the global queue management keeps balanced the computational load. Now we have the theoretical background to study the behaviour of the total complexity of the integration of a function with various regularity properties in different parts of the integration interval. As a matter of fact it is trivial to prove the following theorem.

Theorem 4.2. Consider an integrable function f and a set of intervals T_1, T_2, ... T_k, assume that on each T_j the following relations hold:

$$m_j \sim \frac{1}{2} \int_{T_j} \left(|f^{(r)}(u)| \frac{|\beta_r|}{r!} / |e(u)| \right)^{\frac{1}{r+1}} du,$$

$$10^{-E_j} \sim \frac{1}{2} \int_{T_j} |e(u)|^{\frac{r}{r+1}} \left(|f^{(r)}(u)| \frac{|\beta_r|}{r!} \right)^{\frac{1}{r+1}} du$$

Then let $T = T_1 \cup T_2 \cup ... \cup T_k$, the total complexity on T behaves as in (4.8) with

$$C_T = \sum_{j=1}^{k} C_{T_j} = \frac{1}{2} \left(\frac{|\beta_r|}{r!} \right)^{\frac{1}{r+1}} \int_{T_1 \cup T_2 \cup ... \cup T_k} |f^{(r)}(u)|^{\frac{1}{r+1}} du . \qquad \square$$

Global adaptive + linear extrapolation

In Kahaner and Stoer (1983) an asymptotical analysis of an extrapolated adaptive quadrature is presented. The basic algorithm (called GAQ) is essentially the same of GQ and uses the Trapezoidal rule. For this rule r=2 and using the above results it is easy to state that the complexity of GAQ is $N= O(10^{E/2})$, i.e. $10^{-E} = O(N^{-2})$ this result is consistent with the bound given in Kahaner and Stoer (1983), and was already known in literature (De Boor and Rice, 1979). The extrapolated algorithm GAX performs Romberg extrapolation inside a global adaptive schema with a limited number k of columns in the T-table. Kahaner and Stoer prove that GAX, for $C^{2n+2}[a,b]$ functions, with n>k, attains the bound $10^{-E} = O(N^{-2k-2})$ which in our notation becomes $N=O(10^{E/(2k+2)})$.

In Kahaner and Stoer (1983) none analysis is done of the complexity of GAX with an unlimited number of columns in the T-table when applied to C^{∞} functions. Since for uniformly smooth functions (like sin x) no gain is achieved using adaptivity, it is reasonable to conjecture that the overall performance cannot be better than the value $N=O(10^{C\sqrt{E}})$ of the non-adaptive Romberg extrapolation.

5. DOUBLE ADAPTIVE QUADRATURE SCHEMATA

Overall description of the algorithm

From the results of the previous sections it turns out that the best performances are obtained for class $\mathcal{A}$ functions by the Clenshaw-Curtis Quadrature (N=O(E)) and for class $\mathcal{B}$ and $\mathcal{U}$ functions by the GQ algorithm ($N=\Theta(10^{E/r})$). In this section we investigate a quadrature schema that is applicable to functions of any class, attaining better complexity bounds than GQ.

The basic idea (see also Oliver, 1972) consists in combining, in a general global adaptive schema, the two main strategies for improving the approximation of the integral, i.e. the interval subdivision and the application of more accurate formulas. In other words the active subintervals are ordered into a queue according to the error estimate, and at any step the subinterval with the worst error estimate can be bisected or processed with a higher degree formula (in the following we will use formulas in the Clenshaw-Curtis family). The choice between the two alternatives can be determined by the presence of difficulties in the subinterval. Obviously, if the location of the singularities is not *a priori* known, an empirical test (which can sometimes fail) has to be adopted.

First, we discuss two theoretical algorithms that know the location of the singularity. In these cases the strategy of integration consists in halving the worst subinterval if it contains the singularity and processing it with a higher order formula

otherwise. Finally we present a general algorithm that can be applied to any function (including functions with several singularities).

For class $\mathcal{A}$ functions, since (apart from transitory numerical difficulties) no singularities can be detected, the integration process is reduced to the application of the Clenshaw-Curtis non adaptive algorithm of Section 3 on a finite number of subintervals and the overall complexity is O(E).

It is remarkable that the resulting complexity (namely O(E) for $\mathcal{A}$ functions, $O(E^2)$ for $\mathcal{B}$ and $\mathcal{U}$ functions) is consistent with the theoretical bounds of Traub *et al.* (1988, pag. 118).

Analysis for integrands of class $\mathcal{U}$

Let us investigate now the complexity of the double adaptive algorithm for $\mathcal{U}[0,1,\alpha,1/4]$ functions. Since the singularity is in 0, only the leftmost subinterval is bisected; at any step there is one "red" subinterval containing the singularity, which is halved more and more, generating "black" subintervals that never will be halved.

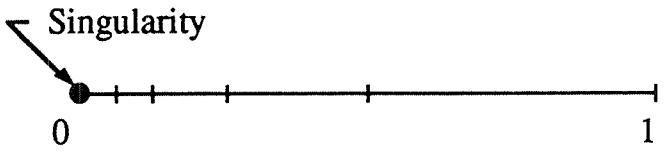

Fig. 5.1. Subinterval structure.

Let $S^{(i)} = [0,2^{-i}]$ be the red subinterval, on which a Clenshaw-Curtis formula with m_0 nodes is applied. A bound to i as a function of the absolute value ω of the error of the worst subinterval, is given in Lemma 4.2. Let $R^{(j)} = [2^{-j},2^{-j+1}]$ be a black subinterval and $e^{(j)}(\mu)$ be the associated error when the Clenshaw-Curtis rule with $\mu \geq m_0$ nodes is applied on $R^{(j)}$. Let m_j be the number of nodes of the formula actually used on $R^{(j)}$ and assume the integration process to be forwarded enough to guarantee that

$$m_h = \max_{0 \leq j \leq i} m_j > m_0.$$

From Theorem A.4, with $\tau=1/2$, $\rho = 1+\sqrt{2} > 2$, we get

$$| e^{(j)}(m_j) | \leq F \; 2^{-j-1} \max_{z \in \partial\Gamma(2^{-j},2^{-j+1},2^{-j-1})} |f(z)| \; \rho^{-m_j-2}, \quad j=1,2,\ldots,i,$$

moreover for any $0 < \varepsilon < 1-\alpha$, there exists a constant Q for which

$$| e^{(j)}(m_j) | \leq F\,Q\, 2^{-j-1-(-j-1)(\alpha+\varepsilon)} \rho^{-m_j-2} < F\,Q\, 2^{-(1-\alpha-\varepsilon)(j+1)-m_j-2}, \quad j=1,2,\ldots,i.$$

When the red subinterval is divided we have

$$| e^{(j)}(m_j) | \leq \omega, \qquad j=1,2,\ldots,i;$$

moreover, since

$$F\,Q\,2^{-(1-\alpha-\varepsilon)(h+1)-m_h/2-5/2} > |\,e^{(h)}(m_h/2+1/2)\,| \geq \omega,$$

i.e.

$$-\,\mathrm{Log}_2\,(F\,Q) + (1-\alpha-\varepsilon)(h+1) + m_h/2 + 5/2 < \mathrm{Log}_2\,1/\omega,$$

then

$$-\,\mathrm{Log}_2\,(F\,Q) + m_h/2 + 5/2 < \mathrm{Log}_2\,1/\omega.$$

Therefore, there exist constants $C_1,\ldots,C_5$ for which, for ω sufficiently small,

$$m_h < C_1\,\mathrm{Log}_2\,1/\omega + C_2;$$

$$N \leq i\,m_h + (i+1)\,m_0 < C_3\,(\mathrm{Log}_2\,1/\omega)^2$$

$$10^{-E} \leq (i+1)\,\omega \leq C_4\,\omega\,\mathrm{Log}_2\,1/\omega$$

$$E \geq \mathrm{Log}_{10}\,1/\omega - \mathrm{Log}_{10}\,\mathrm{Log}_2\,(1/\omega) - \mathrm{Log}_{10}\,C_4 \geq C_5\,\mathrm{Log}_2\,1/\omega$$

from which, finally,

$$N(E) = O(E^2).$$

Analysis for class $\mathcal{B}$ integrands

Let us investigate now the complexity of the double adaptive algorithm for a $\mathcal{B}$[0,1,y,s,1/4], function. In this subsection we assume that the abscissa of the singularity is a real number whose binary digits can be computed by the quadrature algorithm.

There is one red subinterval containing the singularity, which is halved more and more, each time generating one red and one white subinterval. Any white subinterval is divided into two black subintervals if neither one is adjacent to the singularity, otherwise it is split into one black and one white subinterval. On red and white subintervals a Clenshaw-Curtis formula with m_0 node is applied. The black subintervals are processed with higher order formulas and they are never halved.

The structure of subintervals can be exemplified with the help of a binary tree. The root is a level 0 and denotes the interval [0,1], the 2^i nodes at level i denote the possible subintervals of length 2^{-i}. Red nodes are the subintervals containing the singularity, the white ones are the two subintervals surrounding the red one.

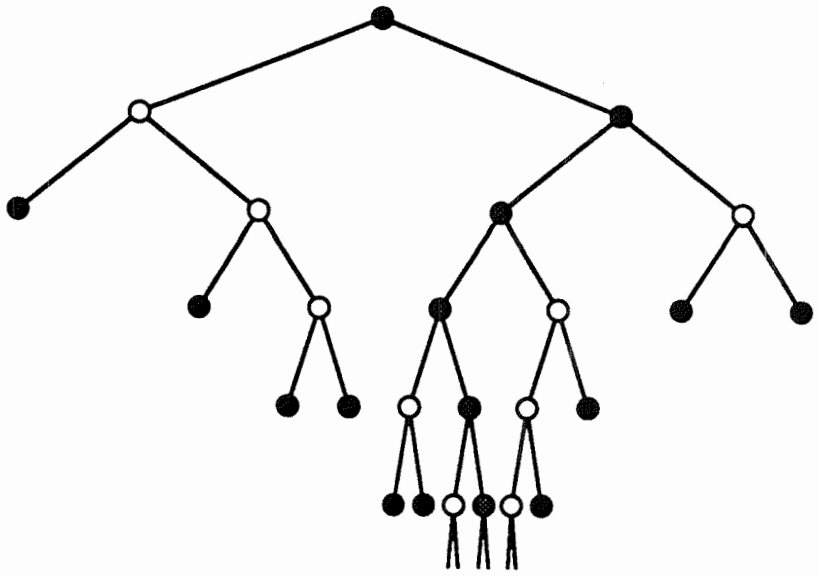

Fig 5.2. Tree associated with the subinterval structure

Assuming that the red subinterval is not larger than any other one, it is easy to see that if the red subinterval has length 2^{-i} (i>1) there are 2 white subintervals and 3i-5 black ones. Let $S^{(i)}$ be the red subinterval, a bound to i as a function of the absolute value ω of the error of the worst subinterval is given in Lemma 4.1. For what concerns black subintervals, we point out that the distance from the singularity of any black subinterval of length λ is greater than or equal to λ. Then, if Clenshaw-Curtis rule with m_j nodes is used on the j-th black subinterval, Theorem A.4, with $\rho>2$, can be applied

$$| e^{(j)}(m_j) | \leq \frac{1}{2} M F 2^{-m_j-2} < C_1 2^{-m_j},$$

where

$$M = \max \left\{ \max_{z \in \partial\Gamma(0,y,1/4) \cap \{Re(z)<y\}} |f(z)| , \max_{z \in \partial\Gamma(y,1,1/4) \cap \{Re(z)>y\}} |f(z)| \right\}.$$

The analysis is similar to the previous one. When the red subinterval is divided we have

$$| e^{(j)}(m_j) | \leq \omega, \qquad j=1,2,...3i-5.$$

Let $m_h = \max_{0 \leq j \leq 3i-5} m_j > m_0$, then

$$C_1 2^{-m_h/2-1/2} > | e^{(h)}(m_h/2+1/2) | \geq \omega.$$

Then there exist constants $C_2, ..., C_6$ for which, for ω sufficiently small,

$$m_h < C_3 \, \mathrm{Log}_2 \, 1/\omega + C_2;$$

$$N \leq (3\,i-5)\, m_h + 3\, i\, m_0 < C_4 \, (\mathrm{Log}_2 \, 1/\omega)^2$$

$$10^{-E} \leq (3i-2)\, \omega \leq C_5 \, \omega \, \mathrm{Log}_2 \, 1/\omega$$

$$E \geq \text{Log}_{10}\, 1/\omega - \text{Log}_{10}\, \text{Log}_2\, (1/\omega) - \text{Log}_{10}\, C_5 \geq C_6\, \text{Log}_2\, 1/\omega$$

from which, finally,

$$N(E) = O(E^2).$$

DAQ Algorithm

In the following we describe the main lines of a general algorithm, implementing the double adaptive strategy, which may also work with several singularities located anywhere in the interval. The program is based on the consideration that an empirical test on fast convergence can detect the presence both of singularities and of integration difficulties that prevent the estimated error to behave how predicted by the asymptotical theory.

By using the theory of the error for Clenshaw-Curtis quadrature for functions of class $\mathcal{A}$, and assuming that the bounds in relation (A.6) of the Appendix are asymptotically sharp we get

$$\begin{aligned} |e| &= |I - I^{(m)}| \quad \sim C_1\, \rho^{-m-2}, \\ |e_1| &= |I - I^{(2m-1)}| \quad \sim C_1\, \rho^{-2m-1}, \end{aligned}$$

$$\left|\frac{e^\beta}{e_1}\right| \sim C_1^{(\beta-1)}\, \rho^{(1-2\beta)}\, \rho^{(2-\beta)m} = C_2\, \rho^{(2-\beta)m}, \quad \beta \leq 2.$$

The last relation shows that, given a $\beta<2$, (e.g. $\beta=1.5$) the relation $\lim_{m\to\infty} \left|\frac{e_1}{e^\beta}\right| = 0$ is a necessary condition for the asymptotical validity of (A.6). These considerations suggest a family of empirical tests of the type:

$$|e_1| < 0.01\, |e|^{1.5},$$

which are asymptotically consistent with the regularity of the integrand.

The Algorithm 5.1 shows how the double adaptive quadrature can be implemented. The basic strategy is to apply bisection in presence of integration difficulties and to forward the Clenshaw-Curtis rule if the convergence test has been passed. The function test(est,est1) returns true if $est1 > 0.01\, est^{1.5}$ and is used to stop the use of more accurate rules and to force the application of bisection.

Obviously the test may fail but the asymptotical validity of the test ensures that the resulting overload does not affect the asymptotical complexity. It is remarkable that, for a class $\mathcal{B}$ integrand the white subintervals are bisected only if the singularity is enough close to affect the convergence of Clenshaw-Curtis rules, then the complexity analysis made before is conservative.

Algorithm 5.1

```
procedure DAQuadrature (a, b, epquad: real; var abserr, result: real);
    var res,est,res1,est1,res2,est2,middle: real;
        k: integer;

{       k0: integer; is a global variable denoting the starting point for the local rule    }
{   procedure CCrule(a,b: real; k: integer; var abserr, result: real);                      }
{           implements Clenshaw-Curtis Quadrature with 2^k+1 points                         }
{   function test(est,est1 : real):boolean;                                                 }
{           test the convergence after doubling the points                                  }
{   put_interval and get_interval handle the prority queue, the actual value of k is stored,}
{           if a singularity has been detected a value of -1 is stored instead.           }

begin
    CCrule(a,b,k0,abserr,result);          { integrate on the first interval }
    put_interval(a,b,k0, abserr, result);  { put the interval in the queue }
    while abserr > epquad do begin
        get_interval(a,b,k,est,res);       { get the interval with the largest error }
        if k>=0 then

{   no singularity nor an integration difficulty was detected for the interval [a,b] }

        begin
            k:=k+1;
            CCrule(a,b,k, est1, res1);
            result:=result-res+res1;              { update estimates }
            abserr:=abserr-est+est1;
            if test(est,est1) then k:=-1;         {test for convergence}
            put_interval(a,b,k,abserr,result)
        end else

{   a singularity or an integration difficulty was detected for the interval [a,b]}

        begin
            middle:=(a+b)/2                       { perform bisection }
            CCrule(a,middle,k0,est1,res1);
            put_interval(a,middle,k0, est1, res1);
            CCrule(middle,b,k0,est2,res2);
            put_interval(middle,b,k0,est2,res2);
            result:=result-res+res1+res2;         { update estimates }
            abserr:=abserr-est+est1+est2
        end
    end;
end;
```

❑

Figures 5.5, 5.6 and 5.7 show the behaviour of GQ and DAQ compared for the function F5 ($x^{3/2}$ integrated in [0,1]).

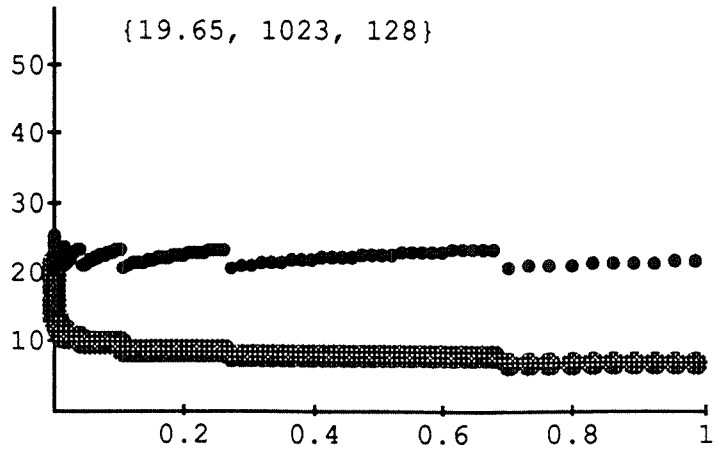

Fig. 5.5. Function F5, Clenshaw-Curtis 7-point, 128 subintervals, program GQ.

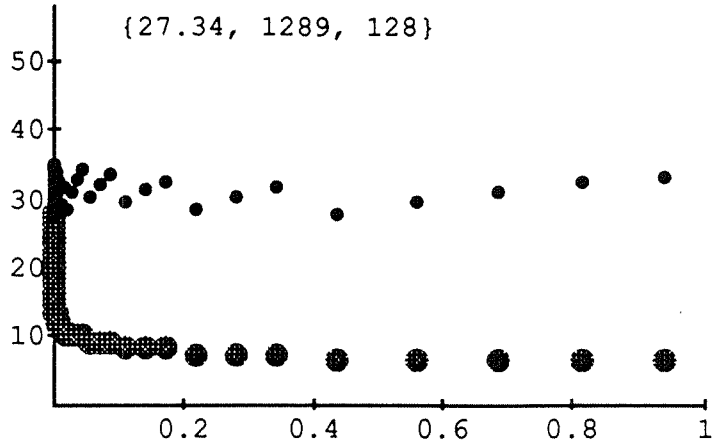

Fig. 5.6. Function F5, 128 subintervals, program DAQ.

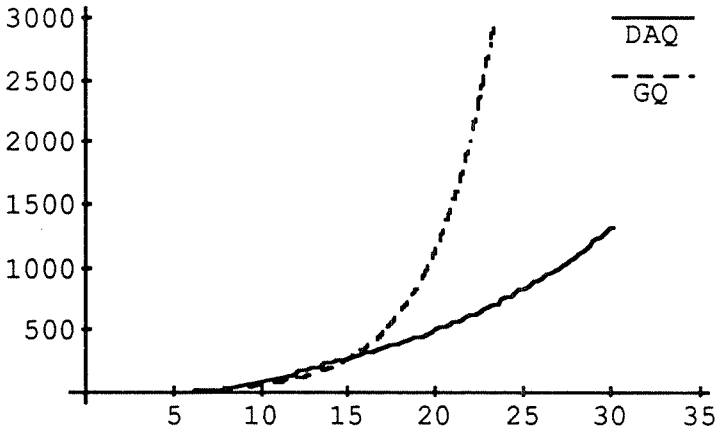

Fig. 5.7. Function F5, GQ-Clenshaw-Curtis 7-point, DAQ complexities.

6. CONCLUSION

The complexity results can be summarised in the following table:

Class → / Algorithm ↓	$\mathcal{A}$	Piecewise polynomials	$\mathcal{B}$	$\mathcal{U}$
Clenshaw-Curtis (CCQ)	E	—	—	—
Panel of degree r-1	$10^{E/r}$	$10^{E/\min[r,s]}$	$10^{E/\min[r,s]}$	—
Romberg	$10^{C\sqrt{E}}$	—	—	—
Global adaptive (GQ) (degree r-1)	$10^{E/r}$	E	$10^{E/r}$	$10^{E/r}$ (*)
GQ + k-column extrapolation	$10^{E/(2k+2)}$	E	—	—
Double adaptive (DAQ)	E	E	E^2	E^2

Table I. Magnitude order of the number of evaluations N vs. the number of exact decimal digits in the result

(*) This result has been proved under the additional hypothesis (4.10).

The same results can be stated expressing the error in terms of the number of evaluations:

Class → / Algorithm ↓	$\mathcal{A}$	Piecewise polynomials	$\mathcal{B}$	$\mathcal{U}$
Clenshaw-Curtis (CCQ)	C^{-N}, $C>1$	—	—	—
Panel of degree r-1	N^{-r}	$N^{-\min[r,s]}$	$N^{-\min[r,s]}$	—
Romberg	$N^{-C \log N}$, $C>0$	—	—	—
Global adaptive (GQ) (degree r-1)	N^{-r}	C^{-N}, $C>1$	N^{-r}	N^{-r} (*)
GQ + k-column extrapolation	$N^{-(2k+2)}$	C^{-N}, $C>1$	—	—
Double adaptive (DAQ)	C^{-N}, $C>1$	C^{-N}, $C>1$	$C^{-\sqrt{N}}$, $C>1$	$C^{-\sqrt{N}}$, $C>1$

Table II. Magnitude order of the error vs. the number of evaluations N.

Finally, for any class of integrands, the best algorithms are presented.

Class	Complexity	Algorithms
Polynomials	O(1)	CCQ, DAQ
Analytical in [a,b]	O(E)	CCQ, DAQ
$C^{\infty}([a,b] - \{y_1,...,y_k\})$	$O(E^2)$	DAQ

Table III. Best automatic quadrature algorithms for various classes of functions

(*) This result has been proved under the additional hypothesis (4.10).

APPENDIX

Error of a single rule, regular functions

The following important property of the derivatives of class $\mathcal{A}[a,b,\delta]$ functions is known (Davis and Rabinowitz, 1984, page 301):

$$(A.1) \qquad \max_{x\in[a,b]} \frac{|f^{(j)}(x)|}{j!} \le \delta^{-j} \max_{z\in\partial\Gamma(a,b,\delta)} |f(z)|, \qquad j \ge 1.$$

Consider an interpolatory formula Q with degree of precision r-1 and a function $f\in C^s[-1,1]$. Let $I_{computed}$ be the result of the approximation of the integral

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

obtained using Q. The following equalities can be stated for the truncation error:

$$(A.2) \qquad I - I_{computed} = \int_{-1}^{1} f^{(j)}(t)\, K_{j-1}(t)\, dt, \qquad 0 < j \le \min[r,s],$$

where $K_j(t)$ is the so called Peano Kernel defined as in Davis and Rabinowitz, (1984), page 286.

To apply the quadrature formula to the integration on the interval [a,b] of a function $f\in C^s[a,b]$ under a linear transformation, we compute the integral as follows

$$\int_{a}^{b} f(x)\, dx = \frac{b-a}{2} \int_{-1}^{1} f\Big(\frac{(b-a)}{2} t + \frac{(a+b)}{2}\Big)\, dt.$$

The equality (A.2) becomes

$$(A.3) \qquad I - I_{computed} = \Big(\frac{b-a}{2}\Big)^{j+1} \int_{-1}^{1} f^{(j)}\Big(\frac{(b-a)}{2} t + \frac{(a+b)}{2}\Big)\, K_{j-1}(t)\, dt, \quad 1<j\le \min[r,s],$$

hence

$$(A.4) \qquad |I - I_{computed}| \le \Big(\frac{b-a}{2}\Big)^{j+1} c_j \max_{x\in[a,b]} \frac{|f^{(j)}(x)|}{j!}, \quad 1<j\le \min[r,s],$$

where we denote by c_j the quantities:

$$c_j = j! \int_{-1}^{1} | K_{j-1}(t) | \, dt, \qquad 1 < j \leq r.$$

For $\mathcal{A}[a,b,\delta]$ functions, combining (A.1) and (A.4) we get

$$(A.5) \qquad |I - I_{computed}| \leq \left(\frac{b-a}{2}\right)^{r+1} c_r \, \delta^{-r} \max_{x \in \partial\Gamma(a,b,\delta)} | f(z) |.$$

A more precise representation of the error can be obtained by applying the results of Favati *et al.* (1992) which involve constants β_r and γ_{r+2}, related to the Peano coefficients, depending on Q only.

Theorem A.1. Let $\lambda>0$ and let f be a $C^{r+2}[x-\lambda,x+\lambda]$ function. Then the error of the integral of f on $[x-\lambda,x+\lambda]$ computed by the symmetrical rule Q can be expressed as

$$I - I_{computed} = \lambda^{r+1} f^{(r)}(x) \frac{\beta_r}{r!} + \lambda^{r+3} \Delta,$$

with

$$|\Delta| \leq \frac{|\gamma_{r+2}|}{(r+2)!} \max_{u \in [x-\lambda,x+\lambda]} | f^{(r+2)}(u) |,$$

where β_r and γ_{r+2} are constants depending on Q only.

Proof. The thesis is a direct consequence of Theorem 3.3 in Favati *et al.* (1992) ❑

Theorem A.1 allows proving the asymptotical validity of Assumption 2.2. Consider two symmetrical rules Q and Q' of degrees of precision r-1 and r'-1, r'>r. Let $I_{Q'}$ and I_Q be the corresponding computed values of the integral. For $C^{r'+2}[x-\lambda,x+\lambda]$ functions we have

$$I - I_Q = \lambda^{r+1} \frac{\beta_r}{r!} f^{(r)}(x) + O(\lambda^{r+3}),$$

and analogously

$$I - I_{Q'} = \lambda^{r'+1} \frac{\beta'_{r'}}{r'!} f^{(r')}(x) + O(\lambda^{r'+3}).$$

Since r' > r

$$I_{Q'} - I_Q = I - I_Q - \lambda^{r'+1} \frac{\beta_{r'}}{r'!} f^{(r')}(x) + O(\lambda^{r'+3}) = I - I_Q + O(\lambda^{r'+1}).$$

This proves the asymptotic correctness of the trivial error estimate $I_{Q'} - I_Q$.

Error of a single rule, singular functions

The following theorem provides an upper bound to the error that holds for all bounded integrands.

Theorem A.2. Let f be a bounded function in [a,b]. Suppose to integrate f by an interpolatory rule with positive weights in an interval of length $[x-\lambda,x+\lambda] \subseteq [a,b]$. Then

$$|I - I_{computed}| \leq B\,\lambda.$$

where B is a constant depending on f , a, b, and independent on λ.

Proof. Let

$$f_{max} = \max_{u \in [a,b]} f(u), \qquad f_{min} = \min_{u \in [a,b]} f(u),$$

Let $\{x_i\}$ be the nodes and $\{\omega_i\}$, $\omega_i > 0$, be the weights of the rule under consideration, respectively, we have

$$2\lambda\, f_{min} \leq \int_{x-\lambda}^{x+\lambda} f(u)du = I \leq 2\lambda\, f_{max},$$

$$2\lambda\, f_{min} \leq \sum_{i=1}^{n} \omega_i f(x_i) = I_{computed} \leq 2\lambda\, f_{max},$$

and

$$|I - I_{computed}| \leq 2\lambda\,(f_{max} - f_{min}).$$

The thesis follows with

$$B = 2\, f_{max} - 2\, f_{min}\,.$$ ❑

Theorem A.3. Let f be of class $\mathcal{U}[a,b,\alpha,\delta]$. Let $0 < \lambda \leq (b-a)/2$, and suppose to integrate f in $[a,a+2\lambda]$ with an interpolatory rule Q with positive weights. Then

$$|I - I_{computed}| \leq B\,\lambda^{(1-\alpha-\varepsilon)}.$$

where $0<\varepsilon<1-\alpha$, and B is a constant depending on ε, Q, f, a, b, and independent of λ.

Proof. For any $\varepsilon>0$ there exists a constant Q for which the absolute value of the exact integral can be bounded by

$$\left| \int_a^{a+2\lambda} f(x)dx \right| \le \int_a^{a+2\lambda} |f(x)|dx \le \int_a^{a+2\lambda} \frac{Q}{(x-a)^{\alpha+\varepsilon}} dx = \frac{Q}{1-\alpha-\varepsilon} (2\lambda)^{1-\alpha-\varepsilon}.$$

On the other hand, let x_1 be the smallest node greater than -1 of the formula (given in [-1,1]), and let $W = 1+x_1$ then

$$|I_{computed}| = \left| \lambda \sum_{i=1}^{n} \omega_i f(\lambda x_i + a + \lambda) \right| \le 2 \frac{Q}{W^{\alpha+\varepsilon}} \lambda^{1-\alpha-\varepsilon},$$

and

$$|I - I_{computed}| \le |I| + |I_{computed}| \le 2 Q \left(\frac{2^{-\alpha-\varepsilon}}{1-\alpha-\varepsilon} + \frac{1}{W^{\alpha+\varepsilon}} \right) \lambda^{1-\alpha-\varepsilon}.$$

and the thesis follows with

$$B = 2 Q \left(\frac{2^{-\alpha-\varepsilon}}{1-\alpha-\varepsilon} + \frac{1}{W^{\alpha+\varepsilon}} \right).$$ ❑

Asymptotical behaviour of Clenshaw-Curtis formula

Let $I^{(m)}$ be the result of the integration of a function f on [-1,1] by a Clenshaw-Curtis rule with m nodes. Chawla (1968) and Riess and Johnson (1972) prove that if f is analytical on [-1,1] and continuable analytically in the closed elliptic disk $\mathcal{E}_\rho$ in the complex plane bounded by the ellipse with foci in $z=\pm1$ and sum of the semiaxes $\rho >1$ then,

$$|I-I^{(m)}| \le \rho^{-m-2} \left[\frac{2}{(m-3)(m-1)} - \frac{2}{(m+3)(m+1)} + \rho^{-2} + O(\rho^{-m}) \right] \max_{z \in \partial \mathcal{E}_\rho} |f(z)|.$$

This result leads to the more general Theorem.

Theorem A.4. Let f be in the class $\mathcal{A}[a,b,\tau(b-a)]$, $\tau>0$, and let $I^{(m)}$ be the result of the integration by a Clenshaw-Curtis rule with m nodes. Then there exist constants F and m_0 for which

$$|I - I^{(m)}| \le F \frac{b-a}{2} \max_{z \in \partial\Gamma(a,b,(b-a)\tau)} |f(z)| \ \rho^{-m-2}, \tag{A.6}$$

where

$$m \ge m_0, \qquad \rho = 2\tau + (1+4\tau^2)^{1/2} > 1+ 2\tau.$$

Proof. We compute the integral as follows

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} f\Big(\frac{(b-a)}{2}t + \frac{(a+b)}{2}\Big)\,dt.$$

The integrand is continuable analytically in $\Gamma(-1,1,2\tau)$. The largest ellipse $\mathcal{E}_\rho$ contained in $\Gamma(-1,1,2\tau)$ is that with $\rho = 2\tau + (1+4\tau^2)^{1/2}$. Then there exist constants F and m_0 for which the thesis follows. ❑

Error behaviour after bisections

Theorem A.5. Let $R = [x-\lambda,x+\lambda]$, $R_1 = [x-\lambda,x]$, $R_2 = [x,x+\lambda]$. Let f be a $C^{r+2}[x-\lambda,x+\lambda]$ function, let e, e_1, e_2, be the errors of the integrals of f on R, R_1 and R_2, respectively, computed by the symmetrical rule Q of degree r-1. Then

$$e_1 = 2^{-(r+1)}\,e + O(\lambda^{r+2}),$$

$$e_2 = 2^{-(r+1)}\,e + O(\lambda^{r+2}).$$

Proof. Applying Theorem A.1 to R, R_1,and R_2 the associated errors are

$$e = \lambda^{r+1}\frac{\beta_r}{r!}\,f^{(r)}(x) + \lambda^{r+3}\Delta_0,$$

$$e_1 = 2^{-(r+1)}\lambda^{r+1}\frac{\beta_r}{r!}\,f^{(r)}(x-\lambda/2) + 2^{-(r+3)}\lambda^{r+3}\Delta_1,$$

$$e_2 = 2^{-(r+1)}\lambda^{r+1}\frac{\beta_r}{r!}\,f^{(r)}(x+\lambda/2) + 2^{-(r+3)}\lambda^{r+3}\Delta_2,$$

$$|\Delta_0|, |\Delta_1|, |\Delta_2| \le \frac{|\gamma_{r+2}|}{(r+2)!}\max_{u\in R}\big|f^{(r+2)}(u)\big|.$$

Using Taylor formula we can write

$$f^{(r)}(x \pm \lambda/2) = f^{(r)}(x) \pm \frac{\lambda}{2}f^{(r+1)}(x) + \frac{\lambda^2}{8}\Gamma, \qquad |\Gamma| \le \max_{u\in[a,b]}|f^{(r+2)}(u)|.$$

Hence

$$e_1 = 2^{-(r+1)}e - 2^{-(r+2)}\lambda^{r+2}\frac{\beta_r}{r!}\,f^{(r+1)}(x) + 2^{-(r+3)}\lambda^{r+3}\Big(\frac{\beta_r}{r!}\frac{\Gamma_1}{2} + \Delta_1 - 4\Delta_0\Big) =$$

$$= 2^{-(r+1)}\,e + O(\lambda^{r+2}),$$

$$e_2 = 2^{-(r+1)}e + 2^{-(r+2)}\lambda^{r+2}\frac{\beta_r}{r!}f^{(r+1)}(x) + 2^{-(r+3)}\lambda^{r+3}\left(\frac{\beta_r}{r!}\frac{\Gamma_2}{2} + \Delta_2 - 4\Delta_0\right) =$$

$$= 2^{-(r+1)}e + O(\lambda^{r+2}), \qquad |\Gamma_1|, |\Gamma_2| \le \max_{u\in[a,b]} |f^{(r+2)}(u)|,$$

from which the thesis follows. ❑

Lemma A.1. Let f be a $\mathcal{U}[0,1,\alpha,\delta]$ function, let $R = [x-\lambda, x+\lambda]$, $\lambda < x/2 < \delta$ we have

$$\max_{u\in R} |f^{(j)}(u)| = O(x^{-\alpha-\varepsilon-j}).$$

Proof. By using relation A.1 on $\Gamma(x-\lambda, x+\lambda, x/2)$, we get

$$\max_{u\in R} \frac{|f^{(j)}(u)|}{j!} \le (x/2)^{-j} \max_{z\in\partial\Gamma(x-\lambda,x+\lambda,x/2)} |f(z)| = O(x^{-\alpha-j-\varepsilon}).$$ ❑

Theorem A.6. Let f be a $\mathcal{U}[0,1,\alpha,\delta]$ function with the additional property

$$|f^{(r)}(x)| = \Omega(x^{-\alpha+\varepsilon-r}), \quad x>0, \quad \text{for any} \quad \varepsilon > 0.$$

Let $R=[x-\lambda,x+\lambda]$, $R_1=[x-\lambda,x]$, $R_2=[x,x+\lambda]$, $0<x<1$. Let $\lambda = o(x^{1+2\varepsilon})$ for some positive constant ε. Let e, e_1, e_2, be the errors of the integrals of f on R, R_1 and R_2, respectively, computed by the symmetrical rule Q of degree r-1. Then

$$e_1 = 2^{-(r+1)}e + o(e),$$

$$e_2 = 2^{-(r+1)}e + o(e).$$

Proof. Applying Theorem A.1 to R, R_1,and R_2 and using lemma A.1 the associated errors can be written

$$e = \lambda^{r+1}\frac{\beta_r}{r!}f^{(r)}(x) + O(\lambda^{r+3}x^{-\alpha-\varepsilon-r-2}),$$

$$e_1 = 2^{-(r+1)}e + O(\lambda^{r+2}x^{-\alpha-\varepsilon-r-1}) + O(\lambda^{r+3}x^{-\alpha-\varepsilon-r-2}),$$

$$e_2 = 2^{-(r+1)}e + O(\lambda^{r+2}x^{-\alpha-\varepsilon-r-1}) + O(\lambda^{r+3}x^{-\alpha-\varepsilon-r-2}).$$

From

$$\lambda^{r+1}|f^{(r)}(x)| = \Omega(\lambda^{r+1}x^{-\alpha-r+\varepsilon}).$$

since

$$\lim_{\lambda\to 0} (\lambda^{r+3} x^{-\alpha-\varepsilon-r-2}) / (\lambda^{r+1} x^{-\alpha-r+\varepsilon}) = \lim_{\lambda\to 0} (\lambda/x^{1+\varepsilon})^2 = 0,$$

then

$$|e| \sim \lambda^{r+1} \frac{|\beta_r|}{r!} |f^{(r)}(x)| = \Omega(\lambda^{r+1} x^{-\alpha-r+\varepsilon}).$$

Analogously

$$\lim_{\lambda\to 0} (\lambda^{r+2} x^{-\alpha-\varepsilon-r-1}) / (\lambda^{r+1} x^{-\alpha-r+\varepsilon}) = \lim_{\lambda\to 0} \lambda/x^{1+2\varepsilon} = 0$$

and the thesis follows. ❑

Proof of a simple inequality

Lemma A.2 Let $\varphi(x)$, $\eta(x)$ be non negative functions, and let $r>1$, then

$$\text{(A.7)} \qquad \left(\int_a^b \varphi(x)/\eta(x)\, dx\right) \left(\int_a^b \varphi(x)\, \eta(x)^r dx\right)^{\frac{1}{r}} \geq \left(\int_a^b \varphi(x)\, dx\right)^{\frac{r+1}{r}},$$

where the equality holds if and only if $\eta(x)$ is a constant.

Proof. Let us write, for two non negative functions f(x) and g(x), the Holder inequality:

$$\left(\int_a^b f(x)^p dx\right)^{\frac{1}{p}} \left(\int_a^b g(x)^q dx\right)^{\frac{1}{q}} \geq \int_a^b f(x)\, g(x)\, dx, \qquad p,q>1, \quad \frac{1}{p}+\frac{1}{q}=1$$

and the equality holds if and only if $f(x)^{p-1}/g(x)$ is a constant.

Raising both members of (A.7) to $\frac{r}{r+1}$ we get

$$\left(\int_a^b \varphi(x)/\eta(x)\, dx\right)^{\frac{r}{r+1}} \left(\int_a^b \varphi(x)\, \eta(x)^r dx\right)^{\frac{1}{r+1}} \geq \int_a^b \varphi(x)\, dx.$$

Let $p=\frac{r+1}{r}$ and $q=r+1$ and let $f(x)=[\varphi(x)/\eta(x)]^{\frac{r}{r+1}}$ and $g(x)=[\varphi(x)\, \eta(x)^r]^{\frac{r}{r+1}}$ then $f(x)\, g(x)=\varphi(x)$ and $f(x)^{p-1}/g(x)=\eta(x)^{-1}$ and the thesis follows. ❑

REFERENCES

CHAWLA, M.M. (1968), Error estimates for Clenshaw-Curtis Quadrature. *Math. Comp.* **22**, 651-656.

CLENSHAW, C.W. AND A.R. CURTIS (1960), A method for Numerical Integration on an Automatic Computer. *Numer. Math.* **2**, 197-205.

DAVIS, P.J. AND P. RABINOWITZ (1984), "Methods of Numerical Integration" Academic Press, New York.

DEBOOR, C. AND J.R. RICE (1979), An Adaptive Algorithm for Multivariate Approximation Giving Optimal Convergence Rates. *J. Approx. Theory*, **25**, 337-359.

FAVATI, P., G. LOTTI AND F. ROMANI (1991a), Interpolatory Integration Formulas for Optimal Composition. *ACM Trans. Math. Software* **17**, 207-217.

FAVATI, P., G. LOTTI AND F. ROMANI (1991b), Improving QUADPACK Automatic Integration Routines. *ACM Trans. Math. Software* **17**, 218-232.

FAVATI, P., G. LOTTI AND F. ROMANI (1992), Asymptotic Expansion of Error in Interpolatory Quadrature, (to appear in *Comput. Math. Appl.*).

GENTLEMAN, G.M. (1972), Implementing the Clenshaw-Curtis Quadrature *Comm. ACM* **15**, 337-360.

KAHANER, D. AND J. STOER (1983), Extrapolated Adaptive Quadrature. *Siam. J. Sci. Stat. Comput.* **4**, 31-44.

LYNESS, J.N. AND B.W. NINHAM (1967), Numerical Quadrature and Asymptotic Expansions. *Math. Comp.* **21**, 162-178.

MALCOLM, M.A. AND R.B. SIMPSON (1975), Local versus Global Strategies for Adaptive Quadrature. *ACM Trans. Math. Software* **1**, 129-146.

OLIVER, J. (1972), A Doubly-adaptive Clenshaw-Curtis Quadrature Method, *Comput. J.* **15**, 141-147.

PIESSENS, R. (1973), An Algorithm for Automatic Integration, *Angewandte Informatik* **9**, 399-401.

PIESSENS, R., E. DE DONCKER-KAPENGA, C. ÜBERHUBER AND D.K. KAHANER (1983), "QUADPACK: A Subroutine Package for Automatic Integration". Springer, Berlin.

RIESS R.D. AND L.W. JOHNSON (1972). Error estimates for Clenshaw-Curtis Quadrature. *Numer. Math.* **18** , 345-353.

TRAUB J.F., WASILKOWSKY AND H. WOZNIAKOWSKY (1988). "Information-based Complexity" Academic Press, New York.

WYNN, P. (1956), On a Device for Computing the $e_m(S_n)$ Transformation. *Math. Comp.* **10**, 91-96.