Numerical integration of peak functions

Erich Novak and Ingo Roschmann

April 1996
Modified: May 23, 1996

Summary. We study the logarithmic error of numerical methods for the integration of monotone or unimodal positive functions. We compare adaptive and nonadaptive methods in the worst case setting. It turns out that adaption significantly helps for the class of unimodal functions, but does not help for the class of monotone functions. We do not assume any smoothness properties of the integrands and obtain numerical methods that are reliable even for discontinuous integrands. Numerical examples show that our method is very competitive in the case of nonsmooth and/or peak functions.

1. Examples

Let us begin our discussion with some examples. Assume that we want to compute

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

for a function with an extreme peak, such as

$$f(x) = \frac{1}{10^{-k} + x^2}, \qquad a = -1, \quad b = 1, \tag{1.1}$$

with $k = 10$. Using symbolic calculation, such as Maple, one obtains

$$I(f) \approx 314157 \approx 2 \cdot \int_{-10^{-5}}^{10^{-5}} f(x)\,dx.$$

Assume now that f is not known by its formula but only via a black box that outputs $f(x)$ on input x. In this case we have to apply a numerical scheme and it is apparent that existing software will have difficulties to find $I(f)$ since half of the integral is in an interval of relative length 1/100000.

In this paper we present an algorithm (called PEAK) that will estimate $I(f)$ for any positive unimodal function f on the basis of n function values. The method uses Fibonacci-search to approximate the maximum of the function and then computes an approximation of the integral. For given a and b (with $a < b$), the input of PEAK is a triple (f, n, k) where f is a positive unimodal function, n is the total number of knots that can be used and k is the number of knots used for the approximation of the maximum. The output of PEAK is denoted by $I_n^k(f)$.

We tried the special example (1.1) with $n = 100$ function values. The best values of k are between 27 and 40. The results lie between

$$I_{100}^k(f) = 314010 \qquad \text{and} \qquad I_{100}^k(f) = 315086.$$

This is not bad, given the extreme properties of the function. We even obtain meaningful (but not accurate, of course) results for $n = 30$. For such a small number of knots, most of them should be used for the approximation of the maximum. For k between 25 and 28 we obtain values between

$$I_{30}^k(f) = 317876 \qquad \text{and} \qquad I_{30}^k(f) = 321930.$$

Another class of integrands is given by characteristic functions

$$f : [0, 1] \to \mathbf{R}, \qquad f = 1_{[c,\, d]}. \tag{1.2}$$

Consider a nonadaptive quadrature formula Q_n with n ordered knots $x_1, \ldots, x_n$. Then there exists a subinterval $[x_i,\, x_k]$ such that

$$(x_{k+1} - x_{i-1}) - (x_k - x_i) \geq \frac{2}{n+1},$$

where $x_0 = 0$ and $x_{n+1} = 1$. Consider the two functions $1_{[x_i,\, x_k]}$ and $1_{[x_{i-1}+\varepsilon,\, x_{k+1}-\varepsilon]}$ which agree on all knots if $\varepsilon > 0$ is sufficiently small. It follows that the maximal error of Q_n (on the set of functions of the form (1.2)) is at least

$$\frac{1}{n+1}. \tag{1.3}$$

This lower bound is optimal since the midpoint rule has an error $1/(n+1)$. The adaptive method PEAK is exponentially faster on functions (1.2) with an error bounded by α^n for an $\alpha < 1$. We tried PEAK with functions of the type $f = 1_{[c,\, c+1/2]}$ with different $c \in [0,\, 1/2]$ and 30 knots. We always got $0.499 \leq I_{30}^{15}(f) \leq 0.500$. The algorithm PEAK is defined in Section 6 which you may want to read now. There you also find a discussion of the choice of k, see Remark 4.

2. Introduction

We derive almost optimal methods for the numerical integration of positive unimodal functions with respect to the logarithmic error. We compare adaptive methods with nonadaptive methods and also the absolute error criterion with the logarithmic error criterion.

Our results can be compared with two other possibilities for the integration of peak functions discussed in the literature:

(1) One can use adaptive methods that are based on an heuristic strategy for choosing the knots, based on some error estimator. Guaranteed error bounds are usually not obtained for such methods.

(2) Adaptive methods are used that are based on upper bounds of $\sup_{x\in J} |f^{(k)}(x)|$, where $k \in \mathbf{N}$ and J is a given subinterval of $[a, b]$. If such upper bounds are available, then this approach leads to guaranteed error bounds.

We do not assume that bounds for the derivatives can be computed. In fact, we do not even assume smoothness properties of the integrand. We prove error bounds using the a priori information that f is monotone or unimodal, respectively.

In many applications we want to calculate an integral

$$I(f) = \int_\Omega f(x)\,dx,$$

where $\Omega = \mathbf{R}^d$ and the following is known about f:

(1) The function f is nonnegative, $f \geq 0$.

(2) The function f has a single maximum x^* and, furthermore, the sets $\{x \mid f(x) \geq c\}$ are convex for any positive c.

We study numerical methods I_n of the form

$$I_n(f) = \varphi(f(x_1), \ldots, f(x_n)),$$

i.e., I_n only uses n function values to compute an approximation of $I(f)$. It is not possible to prove any error bound for any quadrature formula I_n if we do not know more about f. We must know something about the location of large values of f and therefore assume that (possibly after some rescaling) the integral can be replaced by

$$I(f) = \int_\Omega f(x)\,dx,$$

where $\Omega = [0, 1]^d$. In this paper we only study the case $d = 1$ and assume that $f : [0, 1] \to \mathbf{R}_0^+$ is increasing in $[0, x^*]$ and decreasing in $[x^*, 1]$.

Further assumptions on f are not required in the final algorithm but are necessary to prove error bounds and the optimality of the algorithm. Therefore, for the theoretical considerations, we restrict our class of functions further and assume that $f \leq 1$. In some cases we know the unique maximum x^*. In this case one can treat the intervals $[0,\ x^*]$ and $[x^*,\ 1]$ separately and, after some simple rescaling, we obtain the class

$$F_0^m = \{f : [0, 1] \to [0, 1] \mid f(0) = 0,\ f(1) = 1,\ f \text{ is nondecreasing}\}.$$

For the case where the maximum x^* is unknown, we consider the class

$$F_0^u = \{f : [0, 1] \to [0, 1] \mid f \text{ is increasing in } [0, x^*] \text{ for an } x^* \text{ and decreasing in } [x^*, 1]\}.$$

Functions from F_0^u are called unimodal. The classes F_0^m and F_0^u are suitable classes if we want to study the absolute error of numerical methods. For the logarithmic error, however, we additionally need a lower bound on the integral $I(f)$ and therefore we define the classes

$$F_\delta^m := F_0^m \cap \{f \mid I(f) \geq \delta\}$$

and

$$F_\delta^u := F_0^u \cap \{f \mid I(f) \geq \delta\}.$$

The parameter δ is between 0 and 1 and is usually rather small.

In some cases we may only know that $f : [0,1] \to [0,1]$ is nondecreasing in an interval $[0, x^*]$ and nonincreasing in $[x^*, 1]$, for some unknown $x^* \in [0,1]$. This information is much weaker than $f \in F_0^u$ because it is more difficult now to find x^*. Assume, for example, that we know $f(1/4) = f(3/4) = 10^{-4}$. If $f \in F_0^u$ then $f(x) \in [0,\, 10^{-4}]$ for all $x \in [0,\, 1/4]$ and $x \in [3/4,\, 1]$ and $x^* \in [1/4,\, 3/4]$. This is important information on f. If, however, we only know that f is unimodal in the weaker sense then we do not know anything about the position of x^* and the integral of f in the interval $[0, 1/4]$ may be large. We stress that PEAK works fine for all positive unimodal functions with the stronger monotonicity condition. It also works in some other cases, such as for functions of the form (1.2), but we do not analyze this in detail.

To compute the integral $I(f)$ approximately we first compute some function values

$$N(f) = (f(x_1), \ldots, f(x_n))$$

of f. If the knots x_k are fixed in advance we speak of nonadaptive information. We also consider adaptive methods, where $x_k = \psi_k(f(x_1), \ldots, f(x_{k-1}))$ depends on what is already known. We consider all methods of the form

$$I_n(f) = \varphi(f(x_1), \ldots, f(x_n)) = \varphi(N(f)),$$

where $\varphi : \mathbf{R}^n \to \mathbf{R}$ and the knots are chosen adaptively or nonadaptively. If we want to stress that I_n is nonadaptive then we write I_n^{non}. Much is known about the power of adaption, see Traub et al. (1988) and the recent survey Novak (1996).

The usual error criterion in theoretical studies of numerical analysis is the absolute error, defined by $|I(f) - I_n(f)|$. In the worst case setting we study the maximal error

$$\Delta_{\mathrm{abs}}(I_n) = \sup_{f \in F} |I(f) - I_n(f)|,$$

where F is a function class, such as F_δ^m or F_δ^u.

We now motivate the logarithmic error. Assume, for example, that after computing $N(f) = y \in \mathbf{R}^n$ we know that either $I(f) \in [10^{-6}, 10^{-5}]$ or $I(f) \in [1/2 + 10^{-6}, 1/2 + 10^{-5}]$. In both cases we can estimate $I(f)$ up to an error of $4.5 \cdot 10^{-6}$, so both cases are similar if we consider the absolute error. However, we also could say that in the first case the largest possible value of $I(f)$ is ten times larger than the smallest value, while this factor is much smaller in the second example. Such a logarithmic scaling is more adequate in many applications and therefore we want to study the logarithmic error, defined by

$$\Delta_{\log}(I_n) = \sup_{f \in F} |\log I(f) - \log I_n(f)|.$$

We use the natural logarithm in this paper, but of course this is not important.

Instead of the logarithmic error also the relative error

$$\Delta_{\mathrm{rel}}(I_n) = \sup_{f\in F} \frac{|I(f) - I_n(f)|}{|I(f)|}$$

can be used. The logarithmic error and the relative error are similar, if both are small.

Assume that

$$l(y) = \inf\{I(f) \mid f \in F,\ N(f) = y\} > 0$$

and

$$u(y) = \sup\{I(f) \mid f \in F,\ N(f) = y\}.$$

With respect to the logarithmic error, the optimal estimate of $I(f)$ is the geometric mean

$$\varphi_{\log}(y) = \sqrt{l(y)u(y)} \tag{2.1}$$

with a local logarithmic error

$$\frac{\log u(y) - \log l(y)}{2}.$$

With the same information the optimal estimate $\varphi_{\mathrm{rel}}(y) = x$ for the relative error is given by $\frac{u-x}{u} = \frac{x-l}{l}$, i.e., by the harmonic mean

$$\varphi_{\mathrm{rel}}(y) = \frac{2l(y)u(y)}{l(y)+u(y)} \tag{2.2}$$

and the local error is

$$\frac{u(y) - l(y)}{u(y) + l(y)} < 1.$$

Observe that the optimal estimate is quite different if $\gamma = \frac{u}{l}$ is large. In this case we still have $\varphi_{\log}(y) = \sqrt{l(y)u(y)}$ while $\varphi_{\mathrm{rel}}(y) \approx 2l(y)$ is almost independent of $u(y)$. In both cases, however, the error is determined by the quotient $\gamma = \frac{u}{l}$, i.e., good information tries to minimize this ratio. This means that optimal information N for the logarithmic error is also optimal for the relative error and the optimal methods $\varphi_{\log} \circ N$ and $\varphi_{\mathrm{rel}} \circ N$ only differ with respect to the φ. We have

$$\Delta_{\mathrm{rel}}(\varphi_{\mathrm{rel}} \circ N) = \sup_{y\in N(F)} \frac{\gamma(y)-1}{\gamma(y)+1} \le \frac{1}{2} \sup_{y\in N(F)} \log \gamma(y) = \Delta_{\log}(\varphi_{\log} \circ N)$$

for any N and always $\Delta_{\mathrm{rel}}(\varphi_{\mathrm{rel}} \circ N) \le 1$. The relative error is bounded by the logarithmic error and, in particular, our upper bounds for $\Delta_{\log}$ also apply to Δ_{rel}. The logarithmic error seems to be more natural to us and so we prefer this notion.

It is well known that the definitions of Δ_{rel} or $\Delta_{\log}$ do not lead to interesting worst case results if the class F is convex and balanced, because algorithms with small relative errors only exist in trivial cases. See Traub et al. (1988), in particular Section

4.6, for this and related results. The relative error leads to very interesting results, also for a convex and symmetric F, if the probabilistic setting is used. See Traub et al. (1988), Section 8.6.

For the classes F_δ^m and F_δ^u with $\delta > 0$ the study of relative or logarithmic errors makes sense also in the worst case setting. We will study the class F_δ^m in Section 4, the main results are the following:

a) Adaptive methods are not better than nonadaptive methods,

$$\inf_{I_n} \Delta_{\log}(I_n) = \inf_{I_n^{\mathrm{non}}} \Delta_{\log}(I_n^{\mathrm{non}}).$$

b) We characterize the optimal (nonadaptive) knots. For small δ, equidistant knots are far from beeing optimal. Equidistant knots lead to an error $\Delta_{\log}(I_n) \approx 1/(2n\delta)$, while optimal methods lead to an error

$$\Delta_{\log}(I_n^*) \approx \frac{\log \frac{1}{\delta}}{2n}.$$

The class F_δ^u of unimodal functions is studied in Section 5, the results are quite different:

a) In the class of nonadaptive methods, equidistant knots are almost optimal with an error $\Delta_{\log}(I_n) \approx 1/(\delta n)$.

b) Adaptive methods are much better and lead to an error of the order

$$\Delta_{\log}(I_n^*) \asymp \frac{\log \frac{1}{\delta}}{n}.$$

See Remark 1 and Remark 3 for some numerical values. The algorithm PEAK is described in Section 6. We begin with some results for the absolute error.

3. The absolute error

The class F_0^m was already studied by Kiefer (1957), see also Novak (1992). It is easy to see that the same results are true for all F_δ^m with $0 \le \delta \le 1/4$. In the case of a large δ the difference between the absolute and the relative error is small and therefore we are more interested in the case of small δ. Hence, for simplicity, in this Section we only consider the case $\delta \le 1/4$.

Proposition 1. Consider the class F_δ^m with $0 \le \delta \le 1/4$. Then we have

$$\inf_{I_n} \Delta_{\mathrm{abs}}(I_n) = \inf_{I_n^{\mathrm{non}}} \Delta_{\mathrm{abs}}(I_n^{\mathrm{non}}) = \frac{1}{2n+2}$$

and the (nonadaptive) method

$$I_n^*(f) = \frac{1}{2n+2} + \frac{1}{n+1} \sum_{i=1}^{n} f\left(\frac{i}{n+1}\right)$$

is optimal.

Sketch of proof: For the lower bound it is enough to study the information for the function $f(x) = x$, i.e., we consider

$$N(f) = (f(x_1), \ldots, f(x_n)) = (x_1, \ldots, x_n) =: y$$

with the knots x_i that are taken for f. Then we have

$$u(y) - l(y) \geq \frac{2}{2n+2}$$

for all $n \in \mathbf{N}$ and the lower bound follows. It is easy to prove the well known upper bound for I_n^*.

□

For the classes F_δ^u we prove that adaption may help insignificantly. We obtain $\inf_{I_n} \Delta_{\mathrm{abs}}(I_n) \approx \inf_{I_n^{\mathrm{non}}} \Delta_{\mathrm{abs}}(I_n^{\mathrm{non}}) \approx n^{-1}$ and hence the error of the optimal method is about twice as large as for the class F_δ^m.

Proposition 2. Consider the class F_δ^u with $0 \leq \delta \leq 1/4$. Then we have

$$\frac{1}{n+2} \leq \inf_{I_n} \Delta_{\mathrm{abs}}(I_n) \leq \inf_{I_n^{\mathrm{non}}} \Delta_{\mathrm{abs}}(I_n^{\mathrm{non}}) \leq \frac{1}{n+1}$$

and the (nonadaptive) method

$$I_n^*(f) = \frac{1}{n+1} \sum_{i=1}^{n} f\left(\frac{i}{n+1}\right)$$

is almost optimal with $\Delta_{\mathrm{abs}}(I_n^*) = 1/(n+1)$.

Proof: First we prove the upper bound for the special method I_n^*. Assume that we know $N(f) = (f(x_1), \ldots, f(x_n)) = (y_1, \ldots, y_n) = y$, where $x_i = i/(n+1)$ and $f \in F_\delta^u$. Then there is a unique $k \in \{1, \ldots, n\}$ such that

$$0 < y_1 < y_2 < \ldots < y_{k-1} < y_k$$

and

$$y_k \geq y_{k+1} > y_{k+2} > \ldots > y_n > 0.$$

We know that the maximum x^* is in the interval $[x_{k-1}, x_{k+1}]$. Here we put $x_0 = 0$ and $x_{n+1} = 1$ for simplicity. A lower bound of the integral is given by

$$I(f) \geq \sum_{i=1}^{k-1} y_i \cdot (x_{i+1} - x_i) + \sum_{i=k}^{n-1} y_{i+1} \cdot (x_{i+1} - x_i) = \frac{1}{n+1} \left(\sum_{i=1}^{n} y_i - y_k \right).$$

Similarly, an upper bound for the integral is given by

$$I(f) \leq \frac{1}{n+1}\left(\sum_{i=1}^{n} y_i + 1\right).$$

Therefore it is clear that $I_n^*(f)$ is a good estimate of $I(f)$ with

$$|I(f) - I_n^*(f)| \leq \frac{1}{n+1}$$

for $f \in F_\delta^u$. We already know from the discussion in Section 1, see (1.3), that this upper bound is optimal for any nonadaptive method, at least for small δ.

Now we assume that I_n is any adaptive method. We apply I_n to the function $f^* \in F_\delta^u$, defined by $f^*(x) = f^*(1-x) = 2x$ for all $0 \leq x \leq 1/2$ and obtain (for this function) some knots $x_1 < x_2 < \ldots < x_k \leq 0.5 < x_{k+1} \ldots < x_n$. We know that $y_i = f^*(x_i) = 2x_i$ for $i \leq k$ and $y_i = f^*(x_i) = 2(1-x_i)$ for $i > k$. Let $u = \sup\{\int_0^1 g\,dx \mid g \in F_\delta^u,\ N(g) = N(f^*)\}$ and $l = \inf\{\int_0^1 g\,dx \mid g \in F_\delta^u,\ N(g) = N(f^*)\}$. If $l > \delta$ then

$$u - l = \sum_{i=0}^{n} 2(x_{i+1} - x_i)^2 + (x_{k+1} - x_k) \cdot (1 - 2(x_{k+1} - x_k) - \min(y_{k+1}, y_k)).$$

For which knots is $u - l$ as small as possible? We can assume that $x_{i+1} - x_i = \alpha$ for $i = 0, \ldots, k-1$ and $x_{i+1} - x_i = \gamma$ for $i = k+1, \ldots, n$. We also put $x_{k+1} - x_k = \beta$ and obtain

$$u - l = 2k\alpha^2 + 2(n-k)\gamma^2 + \beta \cdot (1 - \min(y_{k+1}, y_k)).$$

Using $\beta = 1 - k\alpha - (n-k)\gamma$ we obtain

$$u - l = 2k\alpha^2 + 2(n-k)\gamma^2 + \beta \cdot \max(1 - 2(n-k)\gamma, 1 - 2k\alpha) \geq 2k\alpha^2 + 2(n-k)\gamma^2 + \beta^2.$$

Some trivial calculations show that the right side of the last display takes its smallest value if $\alpha = \gamma = 1/(n+2)$ and $\beta = 2/(n+2)$. Hence we obtain

$$u - l \geq \frac{2}{n+2}$$

and therefore $\Delta_{\mathrm{abs}}(I_n) \geq 1/(n+2)$.

$\square$

By the results on the absolute error we directly get some results for the logarithmic error. First we assume that F_δ is any class of functions such that

$$0 < \delta \leq \int_0^1 f(x)\, dx \leq 1$$

for all $f \in F_\delta$. There are simple relations between $\inf \Delta_{\mathrm{abs}}(I_n)$ and $\inf \Delta_{\log}(I_n)$. Observe that it is enough to consider optimal methods $\varphi_{\log} \circ N$ and $\varphi_{\mathrm{abs}} \circ N$ for both error criteria. The method $\varphi_{\log}$ is defined by (2.1) and it is clear that

$$\varphi_{\mathrm{abs}}(y) = \frac{l(y) + u(y)}{2}.$$

For any N and $y \in \mathbf{R}^n$ we have $|u(y) - l(y)| \leq 2 \cdot \inf_\varphi \Delta_{\mathrm{abs}}(\varphi \circ N)$ and

$$|\log l(y) - \log u(y)| \leq |\log \delta - \log(\delta + (u(y) - l(y)))|.$$

Hence we obtain

$$\Delta_{\log}(\varphi_{\log} \circ N) \leq \frac{1}{2} \,|\log \delta - \log(\delta + 2\, \Delta_{\mathrm{abs}}(\varphi_{\mathrm{abs}} \circ N))|$$

and therefore

$$\Delta_{\log}(\varphi_{\log} \circ N) \leq \frac{\Delta_{\mathrm{abs}}(\varphi_{\mathrm{abs}} \circ N)}{\delta}. \tag{3.1}$$

Similarly we obtain

$$\Delta_{\log}(\varphi_{\log} \circ N) \geq \frac{1}{2} \,|\log 1 - \log(1 - 2\, \Delta_{\mathrm{abs}}(\varphi_{\mathrm{abs}} \circ N))|$$

and therefore

$$\Delta_{\log}(\varphi_{\log} \circ N) \geq \Delta_{\mathrm{abs}}(\varphi_{\mathrm{abs}} \circ N). \tag{3.2}$$

As a consequence of (3.1) and (3.2) we obtain the following estimates from Proposition 1 and 2.

Proposition 3. Consider the class F_δ^m with $0 \leq \delta \leq 1/4$. Then we have

$$\frac{1}{2n+2} \leq \inf_{I_n} \Delta_{\log}(I_n) \leq \inf_{I_n^{\mathrm{non}}} \Delta_{\log}(I_n^{\mathrm{non}}) \leq \frac{1}{\delta\,(2n+2)}.$$

Consider the class F_δ^u with $0 \leq \delta \leq 1/4$. Then we have

$$\frac{1}{n+2} \leq \inf_{I_n} \Delta_{\log}(I_n) \leq \inf_{I_n^{\mathrm{non}}} \Delta_{\log}(I_n^{\mathrm{non}}) \leq \frac{1}{\delta\,(n+1)}.$$

In the interesting case of very small (positive) δ, these estimates are rather poor. In the following we construct new methods and prove error bounds for these methods and so improve the upper bounds of Proposition 3 significantly. In particular we will see that equidistant knots are far from being optimal, with respect to the logarithmic error.

4. Monotone functions

Let $0 < \delta < 1$ and let $N : F_\delta^m \to \mathbf{R}^n$ be a nonadaptive information,

$$N : f \mapsto (f(x_1), \ldots, f(x_n)).$$

We assume that the x_i are ordered and also put $x_0 = 0$ and $x_{n+1} = 1$, hence

$$0 = x_0 \le x_1 < x_2 < \ldots < x_n \le x_{n+1} = 1.$$

The optimal algorithm $\varphi_{\log}$ using N is given by

$$\varphi_{\log}(N(f)) = \sqrt{u(N(f)) \cdot l(N(f))},$$

where

$$u(N(f)) = \sum_{i=0}^{n} (x_{i+1} - x_i) f(x_{i+1})$$

and

$$l(N(f)) = \max\left(\delta, \sum_{i=0}^{n} (x_{i+1} - x_i) f(x_i) \right).$$

By our general discussion above we also know that

$$\Delta_{\log}(\varphi_{\log} \circ N) = \sup_{y \in N(F_\delta^m)} \frac{1}{2} \, |\log u(y) - \log l(y)|.$$

The following result shows that the supremum over all $y \in N(F_\delta^m)$ can be replaced by a maximum over a finite set. We skip the proof, as well as the proof of other results. We hope that the main ideas are still clear and refer the reader to Roschmann (1995) for the technical details.

Proposition 4. For any nonadaptive information $N : F_\delta^m \to \mathbf{R}^n$ the error of the optimal method $\varphi_{\log} \circ N$ is given by

$$\Delta_{\log}(\varphi_{\log} \circ N) = \max_y \frac{1}{2} \, |\log(u(y)) - \log(l(y))|.$$

Here the maximum runs over all $y \in \mathbf{R}^n$ of the form

$$y = (0, 0, \ldots, 0, \tilde{y}, \ldots, \tilde{y}, 1, \ldots, 1),$$

with

$$l(y) = \delta = \sum_{i=0}^{n} (x_{i+1} - x_i) y_i.$$

□

Using Proposition 4 it is possible to characterize the optimal nonadaptive knots which yield the optimal nonadaptive method I_n^* for a given n and $\delta \in]0,1[$. Again we skip the details. It turns out that the optimal knots are "equidistant in a logarithmic sense". Only in the special case $\delta = \frac{n}{2n+2}$ we obtain equidistant points.

Theorem 1. Assume that $\delta \neq \frac{n}{2n+2}$. Then the optimal nonadaptive knots $x_1, \ldots, x_n$ are given by

$$x_{i+1} - x_i = \alpha\beta^{n-i},$$

i.e., $1 - x_n = \alpha$ and $x_n - x_{n-1} = \alpha\beta$ and so on, or

$$x_i = 1 - \alpha \cdot \frac{\beta^{n-i+1} - 1}{\beta - 1}, \qquad i = 1, \ldots, n.$$

The parameter $\beta \in \mathbf{R}^+$ is defined by

$$\delta = \frac{n\beta^{n+1} - (n+1)\beta^n + 1}{(\beta^{n+1} - 1)^2} \tag{3.3}$$

with $\alpha = (\beta - 1)/(\beta^{n+1} - 1)$. In the case $\delta = \frac{n}{2n+2}$ the optimal knots are equidistant, i.e.,

$$x_i = \frac{i}{n+1}.$$

The error of the optimal nonadaptive method is

$$\Delta_{\log}(I_n^*) = \frac{1}{2} \log \frac{(n+1)\beta^{n+2} - (n+2)\beta^{n+1} + 1}{n\beta^{n+1} - (n+1)\beta^n + 1} = \frac{1}{2} \log \left(\frac{\alpha}{\delta} + \beta \right) \tag{3.4}$$

if $\delta \neq \frac{n}{2n+2}$ and

$$\Delta_{\log}(I_n^*) = \frac{1}{2} \log \frac{n+2}{n}$$

if $\delta = \frac{n}{2n+2}$.

□

Remark 1. It follows from Theorem 1 that, if $\delta > 0$ is small, optimal knots are much better than equidistant knots. Consider the case $\delta = 10^{-6}$. For $n = 10$ the optimal parameter β, given by (3.3), is $\beta = 4.21125$. The error of the optimal method is $\Delta_{\log}(I_n^*) = 0.768$. For $n = 100$ the respective values are $\beta = 1.1771$ and $\Delta_{\log}(I_n^*) = 0.087$.

Now we compare these values with the optimal method I_n that uses equidistant knots $x_i = i/(n+1)$ for $i = 1, \ldots, n$. The error of I_n is

$$\Delta_{\log}(I_n) = \frac{1}{2} \log \frac{\delta(n+1) + 1}{\delta(n+1)}.$$

For $\delta = 10^{-6}$ we obtain, for example,

$$\Delta_{\log}(I_{10}) = 5.71, \quad \Delta_{\log}(I_{100}) = 4.60, \quad \Delta_{\log}(I_{10^5}) = 1.20, \quad \Delta_{\log}(I_{10^6}) = 0.35.$$

With equidistant knots we need more than 250000 function values to obtain the same error as with 10 optimal knots.

Adaptive methods. The local error $\frac{1}{2}\,|\log u(y) - \log l(y)|$ of the optimal nonadaptive method I_n^* is maximal for information y of the form that was described in Proposition 4. For the proof of lower bounds for adaptive methods, however, it is important that the maximal local error is also attained by a different $y \in \mathbf{R}^n$. Let f_0 be the smallest function in F_δ^m such that

$$f_0(x_i) = 1 - x_{n-i+1},$$

where the x_i are the optimal knots from Theorem 1. The integral of this function is given by

$$\int_0^1 f_0(x)\,dx = \frac{n\beta^{n+1} - (n+1)\beta^n + 1}{(\beta^{n+1}-1)^2} = \delta$$

and it turns out that the error of the optimal method I_n^* can also be written as

$$\Delta_{\log}(I_n^*) = \frac{1}{2}\,|\log(u(N(f_0)) - \log(l(N(f_0))|.$$

Now consider any adaptive method I_n for the particular function

$$k(x) = \frac{cx}{1-x+c} \qquad \text{with} \qquad c = \frac{1}{\beta^{n+1}-1}$$

and observe that

$$k(x_i) = f_0(x_i) = 1 - x_{n-i+1}$$

for the knots x_i of the optimal nonadaptive information. One can show that for any n knots $u(N(k))/l(N(k))$ is minimal for the special knots of I_n^*. It follows that I_n^* is optimal even in the larger class of all adaptive methods.

□

The optimal β is given by (3.3) and the respective error of the optimal method I_n^* is given by (3.4). If particular values of n and δ are given, then it is easy to compute β and $\Delta_{\log}(I_n^*)$ numerically. We can apply Newton's method to (3.3), i.e., we compute the zero $\beta \in \mathbf{R}^+$ of the function

$$h(\beta) = \frac{n\beta^{n+1} - (n+1)\beta^n + 1}{(\beta^{n+1}-1)^2} - \delta.$$

The function h is monotone decreasing and continuous on $\mathbf{R}^+$ if we also put

$$h(1) := \lim_{\beta\to 1} h(\beta) = \frac{n}{2n+2} - \delta.$$

We want to prove more explicit error bounds, in particular for fixed δ and $n \to \infty$. We put $\beta = 1 + \frac{c}{n}$ and consider the function

$$H_n(c) = \frac{n(1+\frac{c}{n})^{n+1} - (n+1)(1+\frac{c}{n})^n + 1}{((1+\frac{c}{n})^{n+1} - 1)^2}.$$

We obtain

$$\lim_{n\to\infty} H_n(c) = \frac{c \cdot e^c - e^c + 1}{(e^c - 1)^2} =: H_\infty(c).$$

Also the sequence

$$n \cdot (H_n(c) - H_\infty(c))$$

converges and therefore is bounded, i.e.,

$$H_n(c) = \frac{c \cdot e^c - e^c + 1}{(e^c - 1)^2} + O(n^{-1}).$$

We are looking for $c = c(n, \delta)$ such that $H_n(c) = \delta$. Then the optimal $\beta = \beta(n, \delta)$ is given by $\beta(n, \delta) = 1 + \frac{c(n,\delta)}{n}$. The function H_∞ is decreasing on $\mathbf{R}$ with

$$\lim_{c\to-\infty} H_\infty(c) = 1, \quad \lim_{c\to 0} H_\infty(c) = \frac{1}{2}, \quad \text{and} \quad \lim_{c\to\infty} H_\infty(c) = 0.$$

Hence for $0 < \delta < 1$ we obtain a unique $c(\infty, \delta) \in \mathbf{R}^+$ such that

$$H_\infty(c(\infty, \delta)) = \delta$$

and we also obtain

$$\beta(n, \delta) = 1 + \frac{c(\infty, \delta)}{n} + O(n^{-2}).$$

For large n it follows from (3.4) that the error of the optimal method is asymptotically given by

$$\Delta_{\log}(I_n^*) \approx \frac{1}{2} \cdot \frac{(n+1)\beta^{n+2} - (2n+2)\beta^{n+1} + (n+1)\beta^n}{n\beta^{n+1} - (n+1)\beta^n + 1} = \frac{\beta^n (n+1)(\beta - 1)^2}{2(n\beta^{n+1} - (n+1)\beta^n + 1)}.$$

We again replace β by $1 + c/n$ and define

$$F_n(c) = \frac{(1 + \frac{c}{n})^n \cdot (n+1) \cdot \frac{c^2}{n^2}}{2(n(1 + \frac{c}{n})^{n+1} - (n+1)(1 + \frac{c}{n})^n + 1)}.$$

We obtain

$$\lim_{n\to\infty} n \cdot F_n(c(n, \delta)) = \frac{c^2 e^c}{2(ce^c - e^c + 1)}.$$

So we get for each $0 < \delta < 1$ the following result.

Theorem 2.

$$\lim_{n\to\infty} n \cdot \Delta_{\log}(I_n^*) = \frac{c^2 e^c}{2(ce^c - e^c + 1)},$$

where c is given by

$$\frac{ce^c - e^c + 1}{(e^c - 1)^2} = \delta.$$

Remark 2. If δ is very small then $c \approx -\log\delta$ and hence $\beta \approx 1 - \frac{\log\delta}{n}$ and $\Delta_{\log}(I_n^*) \approx -\frac{\log\delta}{2n}$. This proves that the error of the optimal method weakly depends on δ and the lower bound in Proposition 3 is almost optimal.

An adaptive method for F_δ^m. We already know that adaption does not help in the worst case setting. It is true, however, that adaption helps on the average concerning the absolute error, see Novak (1992), and we also expect that an adaptive "greedy" algorithm is better on the average with respect to the logarithmic error. We do not prove error bounds here but want to suggest an adaptive method I_n which was usually better than I_n^* in our numerical experiments. This adaptive method I_n is also similar to one part of our algorithm PEAK.

We describe I_n without further comments and assume that δ and $f \in F_\delta^m$ and n are given. We take the first knot $x_1 = 1 - \sqrt{\delta}$. After m steps ($m \in \mathbf{N}$) we have ordered knots $x_1, \ldots, x_m$. We also put $x_0 = 0$ and $x_{m+1} = 1$.
First define

$$u_m(x,f) = \sum_{i=0}^{m} (x_{i+1} - x_i)\, f(x_{i+1}) \quad \text{and} \quad l_m(x,f) = \max(\delta, \sum_{i=0}^{m} (x_{i+1} - x_i)\, f(x_i)).$$

Furthermore put

$$\lambda = \frac{u_m(x,f)}{l_m(x,f) + u_m(x,f)}$$

and

$$x_k^* = \lambda x_{k+1} + (1-\lambda) x_k \in [x_k,\, x_{k+1}], \quad k = 0, \ldots, m.$$

Now the new knot $x^* = x_k^*$ is from $\{x_0^*, \ldots, x_m^*\}$ such that

$$\frac{u_m(x,f) - (x_k^* - x_k) \cdot (f(x_{k+1}) - f(x_k))}{l_m(x,f)}$$

is minimal. Finally, renumber the $(m+1)$ knots $x_1, \ldots, x_m$ and x^* to get the ordered knots $0 = x_0 < \ldots < x_{m+2} = 1$.

5. Unimodal functions

Now we discuss results for the class F_δ^u. It is intuitively clear that equidistant knots are almost optimal in the class of nonadaptive methods. Let I_n^{eq} be the optimal method $\varphi_{\log} \circ N$ based on equidistant knots. One can prove that

$$\Delta_{\log}(I_n^{\mathrm{eq}}) = \frac{1}{2} \log \frac{\delta(n+1)+2}{\delta(n+1)} \approx \frac{1}{\delta(n+1)}.$$

For every nonadaptive method I_n and odd n one can prove the lower bound

$$\Delta_{\log}(I_n) \geq \frac{1}{2} \log \frac{\delta(n-1)+2}{\delta(n+1)} \approx \frac{1-\delta}{\delta(n+1)}.$$

The main ideas for the proof of these results are the following. First we define the slightly larger class $\widetilde{F}_\delta^u$. Here we only require that f is nondecreasing in $[0,\, x^*]$ and nonincreasing in $[x^*,\, 1]$. The different monotonicity conditions are important for adaptive methods but do not lead to different results in the case of nonadaptive methods. Now one can prove a result which is analogous to Proposition 4.

Lemma. For any nonadaptive information $N : \widetilde{F}_\delta^u \to \mathbf{R}^n$ the error of the optimal method $\varphi_{\log} \circ N$ is given by

$$\Delta_{\log}(\varphi_{\log} \circ N) = \max_y \frac{1}{2} \left| \log(u(y)) - \log(l(y)) \right|.$$

Here the maximum runs over all $y \in \mathbf{R}^n$ of the form

$$y = (0, 0, \dots, 0, \tilde{y}, \tilde{y}, \dots, \tilde{y}, 1, \dots, 1, 0, 0, \dots, 0),$$

or

$$y = (0, 0, \dots, 0, 1, 1, \dots, 1, \tilde{y}, \tilde{y}, \dots, \tilde{y}, 0, 0, \dots, 0),$$

with $l(y) = \delta$.

$\square$

Adaptive methods are much better. We define an almost optimal adaptive method $\widetilde{I}_n$ for the class F_δ^u and $n = 3m$, where $m \geq 2$.
1. Step: We use m knots to compute the maximum x^* by the Fibonacci search method of Kiefer (1953). We obtain an interval $[x_{\text{left}},\, x_{\text{right}}]$ such that $f(x_{\text{left}})$ and $f(x_{\text{right}})$ are known and $x^* \in [x_{\text{left}},\, x_{\text{right}}]$. We do not use the other $(m-2)$ function values in the following. In our implementation of the method we always have

$$c_m := x_{\text{right}} - x_{\text{left}} = \frac{2}{\text{fib}(m+1)},$$

where $\text{fib}(k)$ denotes the kth Fibonacci number, $\text{fib}(1) = 1$, $\text{fib}(2) = 2$, $\text{fib}(k+2) = \text{fib}(k) + \text{fib}(k+1)$.
2. Step: We define m knots in the left interval $[0,\, x_{\text{left}}]$ and use, more or less, the optimal nonadaptive method for monotone functions, discussed in Section 4. More exactly, we define $x_{m+1}, \dots, x_{2m}$ by

$$x_{i+1} - x_i = \alpha_l \beta^{2m-i}, \qquad i = m, \dots, 2m,$$

where $x_0 = 0$ and (only here!) $x_{2m+1} = x_{\text{left}}$. The parameter β is defined by

$$\delta = \frac{m\beta^{m+1} - (m+1)\beta^m + 1}{(\beta^{m+1} - 1)^2}$$

and

$$\alpha_l = x_{\text{left}} \cdot \frac{\beta - 1}{\beta^{m+1} - 1}.$$

3. Step: We define m knots in $[x_{\text{right}},\, 1]$ analogously.

In this way we obtain the knots of a method and, with (2.1), the method $\widetilde{I}_n$ itself. One can prove the following error bound.

Theorem 3. For the class F_δ^u and the method $\widetilde{I}_n$

$$\Delta_{\log}(\widetilde{I}_n) \le \frac{1}{2}\log\left(2\beta - 1 + \frac{\alpha(1-c_m)+c_m}{\delta}\right),$$

where $\alpha = \frac{\beta-1}{\beta^{m+1}-1}$ and $c_m = \frac{2}{\mathrm{fib}(m+1)}$.

Remark 3. For large $n = 3m$ the upper bound tends to

$$\frac{1}{2}\log(2\beta - 1 + \alpha/\delta).$$

This tends to $\beta - 1 + \frac{\alpha}{2\delta}$ for $n \to \infty$ and $\delta \to 0$. Compare this with the optimal error bound

$$\Delta_{\log}(I_m^*) = \frac{1}{2}\log(\alpha/\delta + \beta) \approx \frac{1}{2}(\beta - 1) + \frac{\alpha}{2\delta} \qquad \text{for } F_\delta^m$$

of Section 4. This means that the error of $\widetilde{I}_n$ is (for $n = 3m$) only slightly larger than the error of the optimal method I_m^* for the class F_δ^m. For large n and small δ the error of $\widetilde{I}_n$ is bounded by $-\frac{3\log\delta}{n}$.

We give some numerical values. Assume that $\delta = 10^{-6}$ and $n = 90$. We take 30 values for the Fibonacci search, $m = 30$. The optimal value β is given by $\beta = 1.690$ and we obtain the error bound $\Delta_{\log}(\widetilde{I}_{90}) \le 0.61$. Compare this with nonadaptive methods. We have the lower bound

$$\Delta_{\log}(I_n) \ge \frac{1}{2}\log\frac{\delta(n-1)+2}{\delta(n+1)},$$

and it follows that more than 800000 function values are necessary to beat $\widetilde{I}_{90}$.

6. The algorithm PEAK

The algorithm PEAK works well for all positive unimodal functions, without the knowledge of an upper bound for $\|f\|_\infty$ or a lower bound δ for the integral of f. It is usually favorably compared to the method of Section 5, independently of δ. It is easier, however, to prove error bounds for the method of Section 5.

To define $I_n^k(f)$ for a positive unimodal function and $2 \le k \le n$ we proceed as follows.

1. Step: We use k knots to compute the maximum x^* by the Fibonacci search method. We obtain an interval $[x_{\mathrm{left}}, x_{\mathrm{right}}]$ with length c_k such that $f(x_{\mathrm{left}})$ and $f(x_{\mathrm{right}})$ are known and $x^* \in [x_{\mathrm{left}}, x_{\mathrm{right}}]$. Again, $c_k = 2/\mathrm{fib}(k+1)$, where $\mathrm{fib}(k)$ is kth Fibonacci number.

2. Step: Assume now that m function values were already computed, $k \le m < n$. We renumber the knots such that

$$0 = x_0 \le x_1 < \ldots < x_m \le x_{m+1} = 1$$

and define k by

$$0 = y_0 < y_1 < y_2 < \ldots < y_{k-1} < y_k$$

and

$$y_k \geq y_{k+1} > y_{k+2} > \ldots > y_m > y_{m+1} = 0,$$

where $y_i = f(x_i)$ for $i = 1, \ldots, m$. We know that the maximum x^* is in the interval $[x_{k-1}, x_{k+1}]$. We define the numbers $u_m(x, f)$ and $l_m(x, f)$ by

$$u_m(x,f) = \sum_{i=1}^{k} (x_i - x_{i-1}) \cdot y_i + \sum_{i=k+1}^{m+1} (x_i - x_{i-1}) \cdot y_{i-1}$$

and

$$l_m(x,f) = \sum_{i=1}^{k} (x_i - x_{i-1}) \cdot y_{i-1} + \sum_{i=k+1}^{m+1} (x_i - x_{i-1}) \cdot y_i.$$

Observe that $l_m(x, f)$ is the smallest possible value of the integral, while $u_m(x, f)$ is the largest possible value of the integral only for unimodal functions with $\|f\|_\infty \leq f(x_k)$. Since we do not know an upper bound on f the actual value of the integral of f might be larger than $u_m(x, f)$. Now we define

$$\lambda = \frac{u_m(x,f)}{l_m(x,f) + u_m(x,f)}$$

and $x_l^* \in [x_l,\, x_{l+1}]$, for $l = 0, \ldots, m$ by

$$x_l^* = \lambda x_{l+1} + (1-\lambda) x_l, \qquad l = 0, \ldots, k-2,$$

and

$$x_l^* = \lambda x_l + (1-\lambda) x_{l+1}, \qquad l = k+1, \ldots, m,$$

and

$$x_{k-1}^* = (x_k + x_{k-1})/2, \qquad x_k^* = (x_{k+1} + x_k)/2.$$

Now we compute for each $l = 0, \ldots, m$ a number p_l and take the new knot $x^* = x_l^*$ such that p_l is minimal. The number p_l is defined by

$$p_l = \frac{u_m(x,f) - (x_l^* - x_l) \cdot (f(x_{l+1}) - f(x_l))}{l_m(x,f)}, \qquad l = 0, \ldots, k-1,$$

and

$$p_l = \frac{u_m(x,f) - (x_{l+1} - x_l^*) \cdot (f(x_l) - f(x_{l+1}))}{l_m(x,f)}, \qquad l = k, \ldots, m.$$

3. Step: After we have defined the ordered knots $x_1, \ldots, x_n$ in this way we define the algorithm by

$$\varphi(N(f)) = \sqrt{u_n(x,f) \cdot l_n(x,f)}.$$

Remark 4. We should say one word about the choice of k, the number of knots for the Fibonacci search. If n is small, say $n \leq 30$, then most of the knots should be used for the Fibonacci search. For $n = 30$, a good choice is $k = 25$.

For larger n it is not necessary to increase k very much since the Fibonacci numbers increase rapidly. If $n = 100$ or $n = 200$ then k between 30 and 35 seems to be reasonable.

Acknowledgments: We thank Henryk Woźniakowski for valuable comments.

References

[1] J. Kiefer (1953): Sequential minimax search for a maximum. Proc. of the AMS **4**, 502–505.

[2] J. Kiefer (1957): Optimum sequential search and approximation methods under minimum regularity assumptions. J. Soc. Indust. Appl. Math. **5**, 105–136.

[3] E. Novak (1992): Quadrature formulas for monotone functions. Proc. Amer. Math. Soc. **115**, 59–68.

[4] E. Novak (1996): On the power of adaption. Preprint.

[5] I. Roschmann (1995): Über den logarithmischen Fehler von Quadraturformeln. Diplomarbeit, Universität Erlangen.

[6] J. F. Traub, G. W. Wasilkowski, H. Woźniakowski (1988): Information-Based Complexity. Academic Press.

Erich Novak and Ingo Roschmann
Mathematisches Institut
Universität Erlangen
Bismarckstr. 1 1/2
91054 Erlangen
Germany
novak@mi.uni-erlangen.de