Reliable Computing 2 (3) (1996), pp. 241–253

# Two adaptive Gauss-Legendre type algorithms for the verified computation of definite integrals

Walter Krämer and Stefan Wedner

We propose a two algorithms for computation of (sharp) enclosures of definite interevals: a *local adaptive algorithm* (LAA) and a *global adaptive algorithm* (GAA). Both algorithms are based on Gauss-Legendre quadrature. Error terms are bounded using automatic differentiation in combination with interval evaluations.

Several numerical examples are presented; these examples include comparison with an adaptive interval Romberg scheme.

# Два адаптивных алгоритма типа Гаусса-Лежандра для верифицируемого вычисления определенных интегралов

В. Кремер, С. Веднер

Предлагаются два алгоритма для вычисления (тесных) включений определенных интегралов: локальный адаптивный алгоритм и глобальный адаптивный алгоритм. Оба алгоритма основаны на квадратуре Гаусса-Лежандра. Члены, характеризующие погрешность, находятся с помощью автоматического дифференцирования в сочетании с интервальными оценками.

Представлено несколько численных примеров, которые включают сравнение с адаптивной интервальной схемой Ромберга.

## 1. Notations and basic facts

By $IR$ we denote the set of all closed real intervals. The symbol $\diamondsuit$ means the interval rounding and $\diamondsuit_+, \diamondsuit_-, \diamondsuit_*, \diamondsuit_/ : IR \times IR \longrightarrow IR$ mean interval operations. By $[a]$ we indicate an interval enclosure of the real quantity $a$.

Let $x \in [a,b]$, $n \in \mathbb{N}$ and $x_{k,n}$, $k = 1,\ldots,n$ be given. Then we denote by $\omega_n(x)$ the polynomial

$$\omega_n(x) := \prod_{k=1}^{n}(x - x_{k,n}) \tag{1}$$

associated with the abscissas $x_{k,n}$, $k = 1,\ldots,n$. The *Lagrange basis* $\{l_{k,n}\}_{k=1}^{n}$ with respect to the nodes $x_{k,n}$ is given by

$$l_{k,n}(x) = \frac{\omega_n(x)}{(x - x_{k,n})\omega_n'(x_{k,n})}, \quad k = 1,\ldots,n. \tag{2}$$

We denote the linear space of all polynomials whose degree does not exceed $n \in \mathbb{N}$ by

$$\Pi_n := \{p \mid p \text{ polynomial, degree}(p) \leq n\}$$

and the space of all polynomials

$$\Pi := \bigcup_{k=0}^{\infty} \Pi_k.$$

**Theorem 1.** *Let $V \supseteq \Pi$ be a function space over $[a,b] \subseteq \mathbb{R}$ and $I : V \to \mathbb{R}$ an isotone[1] linear functional. Then for any fixed $n \in \mathbb{N}$, the following holds (see [16]):*

*a) There exists exactly one operator $Q^{(n)} : V \to \mathbb{R}$ of the form*

$$Q^{(n)}(f) = \sum_{k=1}^{n} w_{k,n} f(x_{k,n}) \tag{3}$$

*with $x_{k,n} \in [a,b]$, $w_{k,n} \in \mathbb{R}$ and*

$$Q^{(n)}(f) = I(f) \quad \text{for all polynomials } f \in \Pi_{2n-1}.$$

*b) It is not possible to find numbers $x_{k,n}$ and $w_{k,n}$, $k = 1, \ldots, n$ such that (3) holds for all polynomials $p \in \Pi_{2n}$.*

*c) The $n$ nodes $x_{k,n}$, $k = 1, \ldots, n$ are all different, and they are the roots of the polynomial $\omega_n(x)$ as defined by (1). This polynomial is orthogonal to all polynomials $p \in \Pi_{n-1}$ with respect to the scalar product $\langle f, g \rangle := I(f \cdot g)$.*

*d) The (positive) weights $w_{k,n}$, $k = 1, \ldots, n$ are uniquely defined by*

$$w_{k,n} := I(l_{k,n}), \quad k = 1, \ldots, n$$

*here $l_{k,n}(x)$ denotes the Lagrange basis defined by (2).*

Applying Theorem 1 to the functional

$$I(f) := \int_a^b f(x) w(x)\, dx \quad (< \infty)$$

with weight function $w(x) \geq 0$, $w$ measurable on $[a,b]$, $0 < \int_a^b w(x)\, dx < \infty$, $\int_a^b x^k w(x)\, dx < \infty$ for $k = 1, 2, \ldots$ leads to the well known Gaussian quadrature formula

$$\int_a^b f(x) w(x)\, dx = \sum_{k=1}^{n} w_{k,n} f(x_{k,n}) + R^{(n)}(f) =: Q^{(n)}(f) + R^{(n)}(f)$$

which is exact for polynomials $\in \Pi_{2n-1}$. Their abscissas as well as the weights are chosen to maximize the order of the integration method.

**Theorem 2** (see [13]). *For $f \in C^{2n}[a,b]$, the remainder term $R^{(n)}(f)$ can be written in the form*

$$\begin{aligned} R^n(f) &= \int_a^b f(x) w(x)\, dx - Q^{(n)}(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \langle \omega_n(x), \omega_n(x) \rangle \\ &= \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b (\omega_n(x))^2 w(x)\, dx \end{aligned}$$

[1] $f \in V$, $f(x) \geq 0$ for $x \in [a,b]$ and $f \not\equiv 0 \Rightarrow I(f) > 0$.

*for some* $\xi \in (a,b)$.

This representation can be derived using well known facts about related Hermitian interpolation problems (see [13]) or from Peano kernel considerations (see [2]).

Let us consider some special weights. The weight function $w(x) := \frac{1}{\sqrt{1-x^2}}$ leads to the Gauss-Chebyshev integration formula over the interval $[-1,1]$:

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx \approx Q^{(n)}(f) = \frac{\pi}{n}\sum_{k=1}^{n} f\left(\cos\left(\frac{2k-1}{2n}\pi\right)\right).$$

In this case, the abscissas are given by the roots of the $n$-th Chebyshev polynomial, i.e.,

$$x_{k,n} = \cos\left(\frac{2k-1}{2n}\pi\right), \quad k = 1,2,\ldots,n$$

and the weights can be shown to be $w_{k,n} = \pi/n$, $k = 1,\ldots,n$. For functions $f \in C^{2n}[-1,1]$, we have

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx = Q^{(n)}(f) + \frac{\pi f^{(2n)}(\xi)}{(2n)!\,2^{2n-1}}, \quad \xi \in (-1,1).$$

As a second example let us consider $\int_{-1}^{1} f(x)\,dx$ (weight function $w(x) \equiv 1$). If we assume an arrangement $x_{1,n} < x_{2,n} < x_{3,n} < \cdots < x_{n,n}$ of the nodes from left to right, then:

$$\left.\begin{aligned} x_{i,n} &= -x_{n-i+1,n} \\ w_{i,n} &= w_{n-i+1,n} \end{aligned}\right\} \quad i = 1,\ldots,n, \tag{4}$$

$$w_{1,n} < w_{2,n} < \cdots < w_{n/2,n}, \qquad \text{for } n \text{ even},$$

$$w_{1,n} < w_{2,n} < \cdots < w_{(n+1)/2,n}, \qquad \text{for } n \text{ odd}.$$

For $f \in C^{2n}(-1,1)$ the remainder term is given by

$$R^{(n)}(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi), \quad \xi \in (-1,1). \tag{5}$$

If we choose an arbitrary interval $[a,b]$ as interval of integration, we find for $f \in C^{2n}[a,b]$ that

$$\int_a^b f(x)\,dx = Q^{(n)}_{[a,b]}(f) + \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi), \quad \xi \in (a,b).$$

Here we denote the approximation by $Q^{(n)}_{[a,b]}(f)$ because, in general, the abscissas and the corresponding weights depend on the bounds of the interval of integration. In the case of the Gauss-Legendre quadrature this relationship is quite simple.

Our goal is to compute verified *enclosures* of definite integrals. To achieve this goal, we must know enclosures of the nodes as well as of the corresponding weights. We will not discuss the generation of such enclosures here. Various methods for doing this are described in detail in [14, 15]. We will only point out that the combination of the method described by Sack and Donovan [12] with that of Givens' is well suited for the computation of such enclosures. This method is based on the computation of modified moments to form a co-diagonal matrix, whose eigenvalues are the Gaussian abscissas (see also [3]). The calculations have to be done using a multi-precision interval arithmetic [6]. As an example of the outcome of such a procedure, we list enclosures for the nodes and weights for the Gauss-Legendre quadrature for $n = 20$ in Table 1.

Relations (4) can be used to find the missing values for $k = 11,\ldots,20$.

As a second example the enclosures for the nodes and weights for the approximation $Q^{(8)}(f) \approx \int_0^1 f(x)\ln(1/x)\,dx$ are listed in Table 2.

| Nodes and weights $n = 20$ | | |
|---|---|---|
| $i$ | $x_{i,20}$ | $w_{i,20}$ |
| 1 | $-9.931285991850949^{2}_{3}$E$-01$ | $1.761400713915211^{9}_{8}$E$-02$ |
| 2 | $-9.63971927277913^{79}_{80}$E$-01$ | $4.060142980038694^{2}_{1}$E$-02$ |
| 3 | $-9.122344282513259^{0}_{1}$E$-01$ | $6.267204833410906^{4}_{3}$E$-02$ |
| 4 | $-8.391169718222188^{2}_{3}$E$-01$ | $8.327674157670474^{9}_{8}$E$-02$ |
| 5 | $-7.46331906460150^{79}_{80}$E$-01$ | $1.019301198172404^{4}_{3}$E$-01$ |
| 6 | $-6.360536807265150^{2}_{3}$E$-01$ | $1.181945319615184^{2}_{1}$E$-01$ |
| 7 | $-5.10867001950827^{09}_{10}$E$-01$ | $1.316886384491766^{3}_{2}$E$-01$ |
| 8 | $-3.737060887154195^{6}_{7}$E$-01$ | $1.420961093183820^{8}_{5}$E$-01$ |
| 9 | $-2.277858511416450^{7}_{8}$E$-01$ | $1.491729864726037^{5}_{4}$E$-01$ |
| 10 | $-7.652652113349733^{3}_{4}$E$-02$ | $1.527533871307258^{6}_{5}$E$-01$ |

Table 1. Nodes and weights $w \equiv 1$

| Nodes and weights $n = 20$ | | |
|---|---|---|
| $i$ | nodes $x_{i,8}$ | weights $w_{i,8}$ |
| 1 | $1.33202441608924^{7}_{6}$E$-2$ | $1.64416604728002^{9}_{8}$E$-1$ |
| 2 | $7.975042901389494^{4}_{3}$E$-2$ | $2.37525610023306^{1}_{0}$E$-1$ |
| 3 | $1.97871029326188^{1}_{0}$E$-1$ | $2.26841984431919^{2}_{1}$E$-1$ |
| 4 | $3.54153994351909^{5}_{4}$E$-1$ | $1.75754079006070^{3}_{2}$E$-1$ |
| 5 | $5.29458575234917^{3}_{2}$E$-1$ | $1.12924030246759^{1}_{0}$E$-1$ |
| 6 | $7.01814529939^{1000}_{0999}$E$-1$ | $5.78722107177820^{8}_{7}$E$-2$ |
| 7 | $8.49379320441106^{7}_{6}$E$-1$ | $2.09790737421329^{8}_{7}$E$-2$ |
| 8 | $9.53326450056359^{8}_{7}$E$-1$ | $3.68640710402762^{20}_{19}$E$-3$ |

Table 2. Nodes and weights $w = \ln(1/x)$

# 2. Verified quadrature

Starting with the formula

$$I(f) = Q(f) + R(f) = \sum_{k=1}^{n} w_k f(x_k) + ch^{2n+1} \frac{f^{(2n)}(\xi)}{(2n)!}$$

$\xi \in [a, b]$, $h = b - a$ and $c \in \mathbb{R}$, we are now looking for an interval enclosure $[I(f)]$ of $I(f)$. Let $F, F^{(2n)} : \mathbb{IR} \longrightarrow \mathbb{IR}$ be interval functions with

$$\begin{array}{lllll} \mathbb{IR} & \ni & F([x_i]) & \supseteq & \{f(x) \mid x \in [x_i]\}, \\ \mathbb{IR} & \ni & F^{(2n)}([a,b]) & \supseteq & \{f^{(2n)}(x)/(2n)! \mid x \in [a,b]\}. \end{array}$$

With $[x_i] \ni x_i$, $[w_i] \ni w_i$, $[c] \ni c$ we find that

$$Q(f) \in [Q(f)] := \diamondsuit \sum_{k=1}^{n} [w_k] \circledast F([x_k]).$$

A corresponding enclosure $[R(f)]$ of the remainder term $R(f)$ can be computed by

$$R(f) \in [R(f)] := [c] \circledast \diamondsuit \operatorname{diam}[a,b]^{2n+1} \circledast F^{(2n)}([a,b])$$

yielding

$$I(f) = Q(f) + R(f) \in [I(f)] := [Q(f)] \diamondplus [R(f)].$$

The enclosure $F^{(2n)}([a,b])$ of the $2n$-th Taylor coefficient can be computed using interval arithmetic in combination with automatic differentiation. Program code for the process of automatic differentiation is given, for example, in [11].

# 3. Two adaptive algorithms

It is well known that $\lim_{n\to\infty} Q^{(n)}(f) = I(f)$. However, increasing the order $n$ of the Gaussian quadrature rule requires the computation of a new set of nodes and weights. Also, very high derivates in the error term may lead to numerical difficulties. For the Gauss-Legendre quadrature ($w(x) \equiv 1$) the accuracy of the approximation can often be increased using a composite rule.

Starting with the representation

$$\int_a^b f(x)\,w(x)\,dx = Q^{(n)}(f) + R^{(n)}(f) \tag{6}$$

and applying a linear transformation $x = x(t) = r \cdot t + s$ with

$$r := \frac{\tilde{b} - \tilde{a}}{b - a}, \qquad s := \frac{b\tilde{a} - a\tilde{b}}{b - a}$$

which maps the interval $t \in [a,b]$ onto the interval $x \in [\tilde{a},\tilde{b}]$, we find

$$\int_{\tilde{a}}^{\tilde{b}} f(x)\,w(x)\,dx = r \int_a^b f(r \cdot x + s)\,w(r \cdot x + s)\,dx. \tag{7}$$

In general, the transformation leads to a modified weight function. This means that, in general, integrals with the same weight function but different ranges of integration cannot be handled using the same Gaussian quadrature rule. For example,

$$\int_0^1 \frac{e^x}{\sqrt{1-x^2}}\,dx = \int_{-1}^1 \frac{e^{x/2+1/2}}{\sqrt{4-(1+x)^2}}\,dx.$$

Here it is obviously not possible to apply the Gauss-Chebyshev-quadrature to the integral on the right.

From now on we will restrict our discussion to the case $w(x) \equiv 1$, i.e., to the Gauss-Legendre formula. If the quadrature rule (6) is applied to the transformed integral (7), a new quadrature rule is created:

$$Q^{(n)}_{[\tilde{a},\tilde{b}]}(f) = \sum_{\nu=1}^{n} \tilde{w}_{\nu,n} f(\tilde{x}_{\nu,n}) \tag{8}$$

with

$$\begin{aligned} \tilde{x}_{\nu,n} &= \frac{\tilde{b} - \tilde{a}}{b - a} x_{\nu,n} + \frac{b\tilde{a} - a\tilde{b}}{b - a}, \\ \tilde{w}_{\nu,n} &= \frac{\tilde{b} - \tilde{a}}{b - a} w_{\nu,n}. \end{aligned}$$

For the subdivision $Z : a = y_0 < y_1 < \cdots < y_m = b$, the composite Gauss-Legendre formula is found, if quadrature rule (8) is applied to all subintervals $[y_i, y_{i+1}]$, $i = 0, \ldots, m-1$:

$$Q_Z^{(n)}(f) = \sum_{i=0}^{m-1} \sum_{k=1}^{n} w_{k,n} \frac{y_{i+1} - y_i}{2} f\left(\frac{y_{i+1} - y_i}{2} x_{k,n} + \frac{y_{i+1} + y_i}{2}\right).$$

The corresponding remainder term is given by

$$R_Z^{(n)}(f) = \sum_{i=0}^{m-1} \frac{(y_{i+1} - y_i)^{2n+1}}{2n+1} \frac{(n!)^4}{((2n)!)^2} \frac{f^{(2n)}(\xi_i)}{(2n)!}, \quad \xi_i \in (y_i, y_{i+1}).$$

The local error is of the order $2n+1$, whereas the global error is of the order $2n$ (see [10]).

Let us now discuss how enclosures of $Q_Z^{(n)}$ and $R_Z^{(n)}$ can be computed automatically. The breakpoints $y_i$, $i = 0, \ldots, m$ are assumed to be floating-point numbers. Then we find

$$\begin{aligned}
[Q_Z^{(n)}(f)] &= \Diamond \sum_{i=0}^{m-1} \sum_{k=1}^{n} [\tilde{w}_{k,n}^{(i)}] \, F([\tilde{x}_{k,n}^{(i)}]) \ni Q_Z^{(n)}(f), \\
\text{with } [\tilde{x}_{k,n}^{(i)}] &= (y_{i+1} \diamondminus y_i) \diamondslash 2 \circledast [x_{k,n}] \oplus (y_{i+1} \oplus y_i) \diamondslash 2, \\
\text{and } [\tilde{w}_{k,n}^{(i)}] &= (y_{i+1} \diamondminus y_i) \diamondslash 2 \circledast [w_{k,n}].
\end{aligned}$$

The remainder term can be enclosed by

$$R_Z^{(n)}(f) \in [R_Z^{(n)}(f)] = [c_n] \circledast \Diamond \sum_{i=0}^{m-1} (y_{i+1} \diamondminus y_i)^{2n+1} \circledast F^{(2n)}([y_i, y_{i+1}])$$

where $[c_n]$ is an enclosure of

$$c_n = \frac{(n!)^4}{(2n+1)((2n)!)^2}.$$

First, the enclosures $[\tilde{x}_{k,n}^{(i)}]$ and $[\tilde{w}_{k,n}^{(i)}]$ are computed. Using these quantities, the enclosure $[Q_Z^{(n)}]$ is the result of an interval scalar product. To reduce overestimations, this scalar product should be computed in an optimal way, as it is possible in the so called XSC languages [5, 8].

The quality of the enclosure $[I(f)]$ depends on:

1) the order of the basic Gauss-Legendre formula;

2) the choice of the subdivision $Z$, especially of its "fineness" $|Z| = \max |y_{i+1} - y_i|$;

3) the smoothness of $f$, or more precisely on the behaviour of the $2n$-th derivative of $f$ on $[a, b]$.

An optimal choice of the subdivision $Z$ will usually reduce the effort required for the computation of $Q^{(n)}(f)$ substantially. It is usually more economical to use subintervals whose length is determined by the local behaviour of the integrand. Using automatic differentiation gives the required information about the derivative appearing in the error term. The error depends on both the size of $f^{(2n)}$ on $[y_i, y_{i+1}]$ and the size of $|y_{i+1} - y_i|$ of the actual subinterval. The contribution to the overall error from each subinterval should be approximately equal.

Let us now discuss two different adaptive strategies.

**Local Adaptive Algorithm LAA:** First we describe the local adaptive algorithm (LAA). The main idea of this method is as follows: To find a subdivision $Z$ of $[a, b]$ with an overall error $R_Z(f) < \varepsilon$, we look for two subdivisions $Z_1$ of $J_1 := [a, \frac{a+b}{2}]$ and $Z_2$ of $J_2 := [\frac{a+b}{2}, b]$ with error requirement $R_{Z_1}(f) < \frac{\varepsilon}{2}$ on $J_1$ and $R_{Z_2}(f) < \frac{\varepsilon}{2}$ on $J_2$. If for $J_1$ and/or $J_2$ the error requirement is not satisfied, $J_1$ and/or $J_2$ have to be subdivided again but now with a prescribed accuracy of $\frac{\varepsilon}{4}$ for the integration on each subinterval. The process is repeated until the required accuracy is reached (success) or the diameter of the actual subinterval is smaller than a prescribed quantity $d_{\min}$ (to avoid infinite loops). On all subintervals we use Gauss-Legendre quadrature of the same order $n$. (Of course it would be possible to choose different formulae for different subintervals.) Therefore we suppress the superscript $n$ in the description. In Pascal–XSC the process can be realized most easily using recursion. The foregoing is summarized below as Algorithm LAA:

```
INTEGRAL(u, o, ε)

  1. [E] := REMAINDER(u, o);
     {REMAINDER computes an enclosure of R_[u,o](f)}

  2. IF diam([E]) > ε AND (o − u) > d_min THEN
       ε := ε * 0.5
       INTEGRAL(u, mid(u, o), ε)
       INTEGRAL(mid(u, o), o, ε)
     ELSE
       [Q] := [Q] + GAUSS(u, o)
       {GAUSS computes an enclosure of Q_[u,o](f)}
       [R] := [R] + [E]

-------------------------------------------------------------
Initialize global interval variables [Q] = [R] = 0
Then the call INTEGRAL(a, b, ε) results in:
        b
       ∫ f(x) dx ∈ [I] := [Q] + [R]
        a
```

Figure 1. Local Adaptive Algorithm LAA

The quantities $u$, $o$ denote the boundaries of the actual subinterval and $\varepsilon$ the actual error requirement.

**Global Adaptive Algorithm GAA:** The global adaptive strategy determines a suitable subdivision $Z$ of the interval of integration $[a, b]$ using only information coming from interval evaluations of error terms over subintervals. The enclosure $[Q_Z^{(n)}(f)]$ of the approximation $Q_Z^{(n)}$ is computed in a final step when the subdivision $Z$ (which is equivalent to an appropriate list of subintervals with different sizes) is known. To avoid infinite loops during the generation of $Z$ (for example due to overestimations of interval calculations) we restrict the maximal number of subintervals to $k_{\max}$. Notice that our algorithm also computes an enclosure of $I(f)$ in the case that the number of subintervals is $k_{\max}$. But in this case the error requirement may be unsatisfied for the corresponding subdivision. This only means that the prescribed

error bound (supplied by the user) cannot be guaranteed. Nevertheless, it holds that $I(f) \in [Q_Z^{(n)}(f)] + [R_Z^{(n)}(f)]$.

Again we use Gauss-Legendre quadrature of the same order $n$ on all subintervals and suppress the superscript $n$ in the following description of Algorithm GAA:

1. Initialization:
   $M := \{[a, b]\}$
   $R_{\inf} := \inf[R_{[a,b]}(f)]$
   $R_{\sup} := \sup[R_{[a,b]}(f)]$
   $[Q] := 0$
2. Choose subinterval $J$ from $M$ with $\operatorname{diam}[R_J(f)]$ being a maximum
3. Subdivide $J$ in $k$ subintervals $J_1, \ldots, J_k$
4. Remove $J$ from $M$
   $R_{\inf} := R_{\inf} - \inf[R_J(f)]$ {Update remainder $R$}
   $R_{\sup} := R_{\sup} - \sup[R_J(f)]$
5. Insert $J_1, \ldots, J_k$ in $M$
   **for** $i$:=1 **to** $k$ **do**
   $R_{\inf} := R_{\inf} + \inf[R_{J_i}(f)]$
   $R_{\sup} := R_{\sup} + \sup[R_{J_i}(f)]$
6. **if** $|M| < k_{\max}$ **and** $(R_{\sup} - R_{\inf}) > \varepsilon$
   **then goto** Step 2
   **else for** $J \in M$ **do** $[Q] := [Q] \diamondsuit [Q_J(f)]$
7. **Output** $[I] := [Q] \diamondsuit [R_{\inf}, R_{\sup}] \ni \int_a^b f(x)\,dx$

Figure 2. Global Adaptive Algorithm GAA

# 4. Numerical examples

In this section we compare the two adaptive strategies discussed above. In our comparison we also include an interval method which is based on the local adaptive strategie using Romberg integration (see [4, 9]).

First, we consider the integral $I(f) = \int_0^4 f(x)\,dx \approx -0.151963942232931$ with

$$f(x) = \frac{1}{a^2 + (3x-1)^2} - \frac{1}{a^2 + (3x-4)^2} + \frac{1}{a^2 + (3x-7)^2} - \frac{1}{a^2 + (3x-10)^2}$$

and $a = 0.1$ (see Figure 3). Table 3 shows the number of function evaluations $\#f$ (to compute the quadrature sum) for the global adaptive algorithm[2] (GAA), for the local adaptive

[2]For our numerical examples we choose $k = 2$.

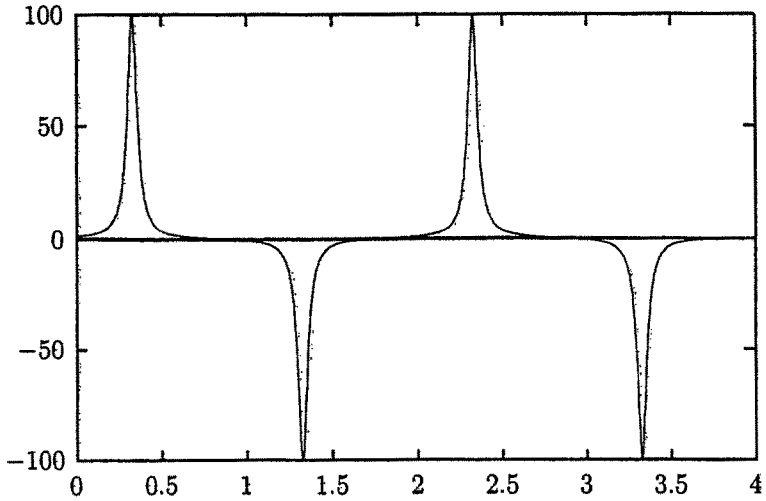


Figure 3. $f(x) = \frac{1}{a^2+(3x-1)^2} - \frac{1}{a^2+(3x-4)^2} + \frac{1}{a^2+(3x-7)^2} - \frac{1}{a^2+(3x-10)^2}$

| | GAA | | | LAA | | | IROMBERG | | |
|---|---|---|---|---|---|---|---|---|---|
| $\varepsilon$ | diam($[I]$) | $\#f$ | t | diam($[I]$) | $\#f$ | t | diam($[I]$) | $\#f$ | t |
| 1E−01 | 1.94E−02 | 416 | 11 | 1.34E−02 | 448 | 12 | 3.19E−02 | 1828 | 17 |
| 1E−02 | 9.34E−03 | 472 | 12 | 9.60E−05 | 568 | 15 | 3.65E−03 | 1384 | 16 |
| 1E−04 | 9.60E−05 | 568 | 15 | 6.76E−06 | 696 | 18 | 9.36E−06 | 2080 | 20 |
| 1E−06 | 9.48E−07 | 704 | 18 | 4.95E−08 | 736 | 19 | 3.64E−08 | 4160 | 30 |
| 1E−08 | 8.23E−09 | 800 | 21 | 1.06E−09 | 960 | 24 | 2.53E−09 | 5120 | 37 |
| 1E−10 | 9.02E−11 | 1032 | 27 | 1.64E−11 | 1176 | 31 | 3.51E−11 | 5888 | 45 |
| 1E−12 | 1.26E−12 | 1304 | 34 | 6.00E−13 | 1432 | 37 | 2.22E−13 | 8640 | 60 |

Table 3. Results for $I(f)$

algorithm (LAA), and for the interval Romberg scheme (IROMBERG). The column denoted by $\varepsilon$ shows the value of the required (absolute) accuracy. The quantity $t$ denotes the time needed and diam($[I]$) is the diameter of the enclosure $[I(f)]$. The basic Gauss-Legendre formula uses eight abscissas.

As a second example we consider the integrand $g(x) = 2xe^{x^2}\sin(e^{x^2})$ (see Figure 4).

For this integrand the value of the integral $I(g) = \int_0^2 g(x)\,dx$ is given by $I(g) = [-\cos(e^{x^2})]_0^2 \approx 0.910964039266$. Table 4 shows a comparison of GAA, LAA, and IROMBERG.

In our last example we compute enclosures of the leading Fourier coefficients of the function

$$f_r(x) = \frac{1 - r\cos x}{1 - 2r\cos x + r^2}, \quad 0 < r < 1.$$

The Fourier expansion of the even function $f_r$ is given by

$$f_r(x) = \alpha_0 + 2\sum_{\nu=1}^{\infty}\Big(\alpha_\nu\cos(\nu x) + \beta_\nu\sin(\nu x)\Big)$$

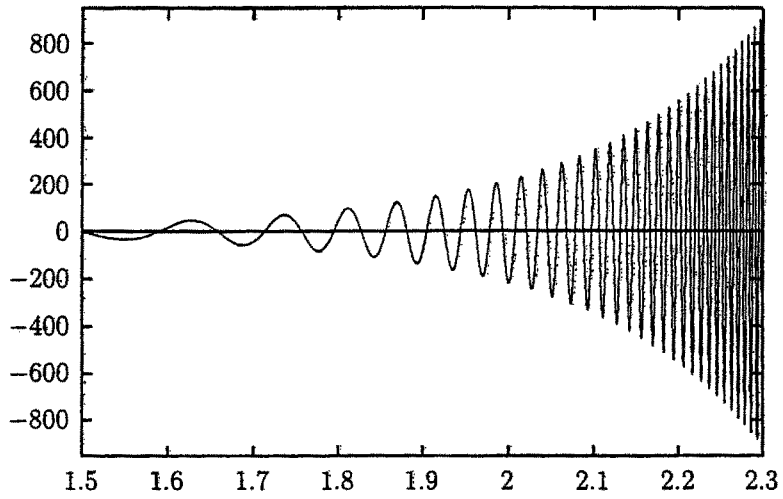


Figure 4. $g(x) = 2xe^{x^2}\sin(e^{x^2})$

| | GAA | | | LAA | | | IROMBERG | | |
|---|---|---|---|---|---|---|---|---|---|
| $\varepsilon$ | diam($[I]$) | $\#f$ | t | diam($[I]$) | $\#f$ | t | diam($[I]$) | $\#f$ | t |
| 1E−01 | 5.00E−02 | 72 | 4 | 4.29E−02 | 80 | 5 | 8.86E−03 | 352 | 16 |
| 1E−02 | 7.56E−03 | 80 | 5 | 4.19E−04 | 88 | 6 | 4.38E−03 | 416 | 18 |
| 1E−04 | 2.82E−06 | 112 | 7 | 2.82E−06 | 112 | 7 | 1.95E−05 | 640 | 27 |
| 1E−06 | 6.87E−07 | 120 | 7 | 1.45E−08 | 152 | 9 | 9.94E−08 | 800 | 34 |
| 1E−08 | 7.88E−09 | 160 | 10 | 1.77E−10 | 192 | 13 | 9.98E−10 | 928 | 40 |
| 1E−10 | 6.78E−11 | 200 | 13 | 1.18E−11 | 224 | 14 | 2.63E−11 | 1376 | 58 |
| 1E−12 | 2.42E−12 | 256 | 16 | 1.49E−12 | 312 | 20 | 5.12E−13 | 1728 | 73 |

Table 4. Results for $I(g)$

with the Fourier coefficients $\beta_\nu = 0$, $\nu = 1, 2, \ldots$ and with

$$\alpha_\nu = \frac{1}{2\pi}\int_0^{2\pi} f_r(x)\cos(\nu x)\,dx, \quad \nu = 0, 1, 2, \ldots \tag{9}$$

We want to use these integral formulae (9) to compute enclosures of $\alpha_\nu$. Because it is not possible to represent the upper bound $2\pi$ of the range of integration as a floating-point number we have to split the interval of integration:

$$\begin{aligned}\int_0^{2\pi} f_r(x)\cos(\nu x)\,dx &= 2\int_0^{\pi} f_r(x)\cos(\nu x)\,dx \\ &= 2\left(\int_0^{\underline{\pi}} f_r(x)\cos(\nu x)\,dx + \int_{\underline{\pi}}^{\pi} f_r(x)\cos(\nu x)\,dx\right).\end{aligned}$$

Here $[\underline{\pi}, \overline{\pi}] \ni \pi$ denotes the best possible enclosure of the transcendental number $\pi$ by an interval whose bounds $\underline{\pi}$ and $\overline{\pi}$ are two adjacent floating-point numbers. The first integral

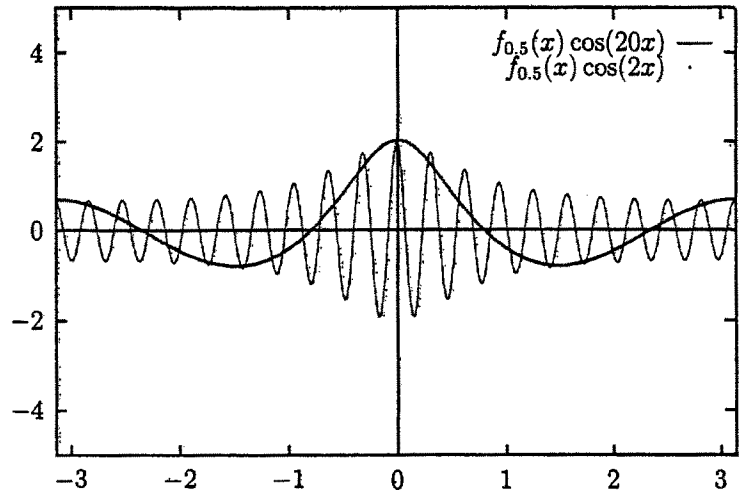


Figure 5. $f_{0.5}(x)\cos(2x)$ and $f_{0.5}(x)\cos(20x)$

on the right hand side can be computed easily by the methods described in this paper. An enclosure for the second integral can be found using the mean value theorem for integration:

$$\int_{\underline{\pi}}^{\pi} f_r(x)\cos(\nu x)\,dx = f_r(\xi)(\pi - \underline{\pi})\cos(\nu\xi), \quad \xi \in [\underline{\pi}, \pi]$$

i.e.,

$$\int_{\underline{\pi}}^{\pi} f_r(x)\cos(\nu x)\,dx \in f_r([\underline{\pi}, \overline{\pi}]) \diamondsuit [0, \triangle(\overline{\pi} - \underline{\pi})] \diamondsuit \cos(\nu[\underline{\pi}, \overline{\pi}]).$$

(The symbol $\triangle$ denotes upwardly directed rounding.) Table 5 lists, for the different methods, the amount of work which has to be done to compute enclosures of $\alpha_0, \alpha_1, \ldots, \alpha_{20}$ for the function $f_{0.5}(x)$ with an absolute error requirement $\varepsilon_\nu := 2^{-\nu} \cdot 10^{-10}$ for $a_\nu$, $\nu = 0, \ldots, 20$. The choice of $\varepsilon_\nu$ takes into account that the exact values $\alpha_\nu$ are given by

$$\alpha_\nu = \begin{cases} 1 & \text{for } \nu = 0 \\ 0.5\,r^\nu & \text{for } \nu \neq 0 \end{cases} \quad \left(\text{in our example } r = \frac{1}{2}\right).$$

So the relative error requirement for all integral enclosures is approximately $10^{-10}$.

# 5. Conclusion

The comparison shows that in all three examples the adaptive Gauss-Legendre algorithms need significantly fewer function evaluations than the adaptive algorithm based on Romberg integration. The global adaptive algorithm is somewhat faster than the local adaptive algorithm and up to five times faster than the interval Romberg algorithm. All three algorithms deliver enclosures of the exact value of the definite integrals under consideration. The width of such an enclosure shows immediately the quality of the computed result.

| | GAA | | | LAA | | | IROMBERG | | |
|---|---|---|---|---|---|---|---|---|---|
| $\nu$ | diam($[\alpha_\nu]$) | $\#f$ | t | diam($[\alpha_\nu]$) | $\#f$ | t | diam($[\alpha_\nu]$) | $\#f$ | t |
| 0 | 2.48E−11 | 56 | 4 | 4.69E−12 | 64 | 5 | 5.71E−13 | 385 | 19 |
| 1 | 9.59E−12 | 72 | 7 | 9.97E−13 | 72 | 8 | 1.11E−12 | 385 | 26 |
| 2 | 3.38E−12 | 80 | 8 | 8.30E−14 | 80 | 8 | 2.49E−12 | 385 | 26 |
| 3 | 2.83E−12 | 88 | 9 | 2.76E−13 | 80 | 8 | 6.06E−13 | 385 | 26 |
| 4 | 3.08E−15 | 96 | 10 | 1.80E−14 | 88 | 9 | 3.03E−13 | 417 | 28 |
| 5 | 5.01E−15 | 96 | 10 | 3.52E−14 | 88 | 9 | 1.22E−13 | 513 | 35 |
| 6 | 7.99E−15 | 96 | 10 | 6.34E−15 | 96 | 10 | 4.74E−14 | 577 | 39 |
| 7 | 1.60E−14 | 96 | 10 | 1.17E−14 | 96 | 10 | 3.12E−14 | 641 | 43 |
| 8 | 3.54E−14 | 96 | 10 | 2.44E−14 | 96 | 10 | 5.89E−15 | 705 | 46 |
| 9 | 5.54E−14 | 104 | 11 | 1.64E−14 | 112 | 12 | 1.15E−14 | 705 | 47 |
| 10 | 3.04E−14 | 120 | 12 | 5.96E−15 | 144 | 15 | 1.20E−14 | 705 | 48 |
| 15 | 6.81E−15 | 184 | 19 | 6.24E−15 | 208 | 22 | 8.87E−15 | 1153 | 78 |
| 20 | 7.18E−15 | 256 | 27 | 7.18E−15 | 256 | 28 | 1.05E−14 | 2049 | 136 |
| $\varepsilon = 0.5^\nu * 1\text{E} - 10$ | | | | | | | | | |

Table 5. Fourier coefficients $\alpha_\nu$

The algorithms can be modified to allow a relative error requirement $\varepsilon_{\text{rel}}$ by the user. In such a case, we first compute a coarse enclosure $[\underline{z}, \overline{z}]$ of $I(f)$. If this enclosure does not contain zero, we find an appropriate absolute error requirement $\varepsilon_{\text{abs}}$ by

$$\varepsilon_{\text{abs}} := \min\{|\underline{z}|, |\overline{z}|\} \cdot \varepsilon_{\text{rel}}.$$

Now LAA or GAA is called with the absolute error bound $\varepsilon_{\text{abs}}$. If the algorithm finishes with a final enclosure $[I(f)]$ with $\operatorname{diam}[I(F)] < \varepsilon_{\text{abs}}$, then it holds that for all $q \in [I(f)]$

$$\left|\frac{\int_a^b f(x)\,dx - q}{\int_a^b f(x)\,dx}\right| \leq \left|\frac{\operatorname{diam}[I(f)]}{\int_a^b f(x)\,dx}\right| \leq \frac{\varepsilon_{\text{abs}}}{\min\{|\underline{z}|, |\overline{z}|\}} = \varepsilon_{\text{rel}}.$$

The error terms which have been used to get the coarse enclosure $[\underline{z}, \overline{z}]$ and the actual subdivision associated with the computation of $[\underline{z}, \overline{z}]$ can be used as the starting point for the computation of $[I(f)]$. However, in general, it is not possible to use the function evaluations done so far.

The authors would like to thank the referees for their helpful comments.

# References

[1] Adams, E. and Kulisch, U. (eds) *Scientific computing with automatic result verification*. Academic Press, San Diego, 1993.

[2] Brass, H. *Quadraturverfahren*. Vandenhoeck & Ruprecht in Göttingen, 1977.

[3] Gautschi, W. *On the construction of Gaussian quadrature rules from modified moments*. Math. Comp. **24** (1970), pp. 245–260.

[4] Kelch, R. *Numerical quadrature by extrapolation with automatic result verification.* In: [1], pp. 143–185.

[5] Klatte, R., Kulisch, U., Neaga, M., Ratz, D., and Ullrich, Ch. *PASCAL–XSC—language reference with examples.* Springer-Verlag, 1992.

[6] Krämer, W. *Multiple-precision computations with result verification.* In: [1], pp. 325–356.

[7] Krämer, W. and Wedner, S. *Computing narrow inclusions for Cauchy principal value integrals.* In: Alefeld, G. and Frommer, A. (eds) "Scientific Computing and Validated Numerics", Akademie Verlag, Berlin, 1996, pp. 45–51.

[8] Kulisch, U. W. and Miranker, W. L. (eds) *Computer arithmetic in theory in practice.* Academic Press, New York, 1981.

[9] Lohner, R. *Habilitationsschrift.* Universität Karlsruhe, 1994.

[10] Niederdrenk, K. and Yserentant, H. *Funktionen einer reellen Veränderlichen.* Rechnerorientierte Ingenieurmathematik, Vieweg, 1987.

[11] Rall, L. B. *Differentiation and generating of Taylor coefficients in PASCAL–SC.* In: [8].

[12] Sack, R. A. and Donovan, A. F. *An algorithm for Gaussian quadrature given modified moments.* Numer. Math. **18** (1972), pp. 465–478.

[13] Stoer, J. and Bulirsch, R. *Introduction to numerical analysis.* Springer-Verlag, 1980.

[14] Storck, U. *Verified calculation of the nodes and weights for Gaussian quadrature formulas.* Interval Computation 4 (1993), pp. 114–124.

[15] Wedner, S. *Numerische Quadratur mit automatischer Ergebnisverifikation.* Diplomarbeit, Universität Karlsruhe, 1994.

[16] Weissinger, J. *Numerische Mathematik auf Personal-Computern Teil 1.* B.I.–Wissenschaftsverlag, 1984.

Received: December 20, 1995
Revised version: February 19, 1996

W. KRÄMER
Institut für Wissenschaftliches Rechnen
und Mathematische Modellbildung (IWRMM)
E-mail: ae36@iamk4508.uni-karlsruhe.de

S. WEDNER
Institut für Angewandte Mathematik
Universität Karlsruhe
76128 Karlsruhe
Germany