Mathematical and Computer Modelling 53 (2011) 1133–1139

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Explicit forms of weighted quadrature rules with geometric nodes

Mohammad Masjed-Jamei [a,c,*], Gradimir V. Milovanović [b], M.A. Jafari [a]

[a] Department of Mathematics, K. N. Toosi University of Technology, P.O. Box 16315–1618, Tehran, Iran

[b] Faculty of Computer Sciences, Megatrend University, Bulevar Umetnosti 29, 11070 Novi Beograd, Serbia

[c] School of Mathematics, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395–5746, Tehran, Iran

ARTICLE INFO

Article history:
Received 9 February 2010
Received in revised form 21 November 2010
Accepted 22 November 2010

Keywords:
Geometric nodes
Weighted quadrature rules
q-binomial theorem
Newton interpolation
Node polynomials

ABSTRACT

A weighted quadrature rule of interpolatory type is represented as

$$\int_a^b f(x)w(x)\,\mathrm{d}x = \sum_{k=0}^{n} w_k f(x_k) + R_{n+1}[f],$$

where $w(x)$ is a weight function, $\{x_k\}_{k=0}^n$ are integration nodes, $\{w_k\}_{k=0}^n$ are the corresponding weight coefficients, and $R_{n+1}[f]$ denotes the error term. During the past decades, various kinds of formulae of the above type have been developed. In this paper, we introduce a type of interpolatory quadrature, whose nodes are geometrically distributed as $x_k = aq^k$, $k = 0, 1, \ldots, n$, and obtain the explicit expressions of the coefficients $\{w_k\}_{k=0}^n$ using the q-binomial theorem. We give an error analysis for the introduced formula and finally we illustrate its application with a few numerical examples.

1. Introduction

Let us start our discussion with a general $(n + 1)$-point weighted quadrature formula of the form

$$\int_a^b f(x)w(x)\mathrm{d}x = \sum_{k=0}^{n} w_k f(x_k) + R_{n+1}[f], \tag{1.1}$$

where $w(x)$ is a positive weight function on $[a, b]$, $\{x_k\}_{k=0}^n$ and $\{w_k\}_{k=0}^n$ are nodes and weight coefficients, respectively, and $R_{n+1}[f]$ is the corresponding error. Let $\mathcal{P}_d$ be the set of algebraic polynomials of degree at most d. The quadrature formula (1.1) has degree of exactness d if for every $p \in \mathcal{P}_d$ we have $R_{n+1}(p) = 0$. In addition, if $R_{n+1}(p) \neq 0$ for some $p \in \mathcal{P}_{d+1}$, formula (1.1) has precise degree of exactness d.

The convergence order of quadrature formula (1.1) depends on the smoothness of the function f, as well as on its degree of exactness. It is well known that for given $n + 1$ mutually different nodes $\{x_k\}_{k=0}^n$ we can always achieve a degree of exactness $d = n$ by interpolating at these nodes and integrating the interpolation polynomial instead of f. Namely, taking the *node polynomial*

$$\Psi_{n+1}(x) = \prod_{k=0}^{n}(x - x_k), \tag{1.2}$$

* Corresponding author at: Department of Mathematics, K. N. Toosi University of Technology, P.O. Box 16315–1618, Tehran, Iran.
E-mail addresses: mmjamei@kntu.ac.ir, mmjamei@yahoo.com, mmjamei@aut.ac.ir (M. Masjed-Jamei), gvm@megatrend.edu.rs (G.V. Milovanović), m_jafari@dena.kntu.ac.ir (M.A. Jafari).

0895-7177/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.11.076

by integrating the Lagrange interpolation formula

$$f(x) = \sum_{k=0}^{n} \ell_k(x) f(x_k) + r_{n+1}(f; x),$$

where

$$\ell_k(x) = \frac{\Psi_{n+1}(x)}{\Psi'_{n+1}(x_k)(x - x_k)}, \quad k = 0, 1, \ldots, n,$$

we obtain (1.1), with

$$w_k = \frac{1}{\Psi'_{n+1}(x_k)} \int_a^b \frac{\Psi_{n+1}(x) w(x)}{x - x_k} \mathrm{d}x, \quad k = 0, 1, \ldots, n, \tag{1.3}$$

and $R_{n+1}(f) = \int_a^b r_{n+1}(f; x) w(x) \mathrm{d}x$. Notice that for each $f \in \mathcal{P}_n$, we have $r_{n+1}(f; x) = 0$, and therefore $R_{n+1}(f) = 0$. Quadrature formulae obtained in this way are known as *interpolatory*. It is well known that any interpolatory quadrature (1.1), with nonnegative coefficients (1.3) is convergent for all continuous functions on $[a, b]$.

Usually, the simplest interpolatory quadrature formula (1.1), with given nodes $x_k \in [a, b]$, is called the *weighted Newton–Cotes formula*. The classical Newton–Cotes formula is for $w(x) = 1$ and the equidistant nodes $x_k = a + kh$, $k = 0, 1, \ldots, n$, where $h = (b - a)/n$ is the step size. On the other side, the interpolatory formulae of the maximal degree of exactness are known as *Gaussian quadrature formulae*. Their construction is closely connected with orthogonal polynomials. For details on quadrature formulae see, for example, [1, pp. 152–185] and [2, pp. 319–361]. In addition, we mention that several other types of standard and nonstandard quadratures have been recently developed (cf. [3–6]).

In this paper we consider a kind of interpolatory quadratures with geometric distribution of nodes. The paper is organized as follows. In Section 2, we consider the geometric nodes $\{x_k = aq^k\}_{k=0}^n$ in the $(n + 1)$-point quadrature formula (1.1) and by a new computational technique we explicitly find the general form for the coefficients $\{w_k\}_{k=0}^n$, together with the error term $R_{n+1}[f]$. In Section 3, we give an error analysis for such quadrature formulae and obtain an explicit form of the error bounds corresponding to the weight function $w(x) = 1$. Finally, in the last section, we examine the weighted quadrature rule with geometric nodes in solving some definite integrals.

2. Weighted quadrature rules with geometric nodes

To establish weighted quadrature rules with geometric nodes, instead of the Lagrange interpolation we consider the Newton interpolation formula [1, pp. 96–1001] at the nodes $0 < a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ as follows

$$f(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \cdots + b_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) + r_{n+1}(f; x), \tag{2.1}$$

where $b_0 = f[x_0]$, $b_1 = f[x_0, x_1]$, $b_2 = f[x_0, x_1, x_2], \ldots, b_n = f[x_0, x_1, \ldots, x_n]$ respectively denote divided differences, and $r_{n+1}(f; x)$ is the corresponding error

$$r_{n+1}(f; x) = f[x_0, x_1, \ldots, x_n, x] \Psi_{n+1}(x),$$

where the node polynomial $\Psi_{n+1}(x)$ is defined in (1.2). Furthermore, if $f \in C^{n+1}[a, b]$ this error can be expressed in the form

$$r_{n+1}(f; x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \Psi_{n+1}(x), \quad \xi(x) \in (a, b).$$

Precisely, the point $\xi(x)$ is strictly between the smallest and largest of the points $x_0, x_1, \ldots, x_n, x$.

Using Newton interpolation formula, we can directly obtain the coefficients $\{w_i\}_{i=0}^n$ in the weighted quadrature rules with the geometric nodes. For this purpose, we first assume in (2.1) that $\{x_i\}_{i=0}^n = \{aq^i\}_{i=0}^n$ for $q = \sqrt[n]{b/a}$. According to the Cauchy q-binomial theorem [7] we have

$$(x; q)_n = \prod_{i=1}^{n} (1 - xq^{i-1}) = \sum_{k=0}^{n} (-1)^k \begin{bmatrix} n \\ k \end{bmatrix}_q q^{\binom{k}{2}} x^k, \tag{2.2}$$

where

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{(q; q)_n}{(q; q)_k (q; q)_{n-k}}, \qquad \binom{k}{2} = \frac{k(k-1)}{2}, \quad (q; q)_m = \prod_{i=1}^{m} (1 - q^i).$$

If $x = t/a$ and $q \to 1/q$, then (2.2) reduces to

$$\left(\frac{t}{a}; \frac{1}{q}\right)_n = \prod_{i=1}^{n} \left(1 - \frac{t}{aq^{i-1}}\right) = \sum_{k=0}^{n} (-1)^k \begin{bmatrix} n \\ k \end{bmatrix}_{1/q} q^{-\binom{k}{2}} a^{-k} t^k,$$

which is equivalent to

$$\prod_{i=1}^{n}\left(t-aq^{i-1}\right)=(-a)^n q^{\frac{n(n-1)}{2}}\sum_{k=0}^{n}(-1)^k\begin{bmatrix}n\\k\end{bmatrix}_{1/q} q^{-\frac{k(k-1)}{2}}a^{-k}t^k. \tag{2.3}$$

On the other hand, since

$$\begin{bmatrix}n\\k\end{bmatrix}_{1/q}=q^{-k(n-k)}\begin{bmatrix}n\\k\end{bmatrix}_q,$$

the polynomials (2.3) are eventually transformed to

$$G_n(x)=G_n(x;a,q)=\prod_{i=1}^{n}\left(x-aq^{i-1}\right)=\sum_{k=0}^{n}(-a)^{n-k}\begin{bmatrix}n\\k\end{bmatrix}_q q^{\binom{n-k}{2}}x^k,$$

where $n=1,2,\ldots,$ and $G_0(x)=1$.

By using (2.3) one can easily find out that

$$\begin{aligned}A_m(a,q)&=\int_a^b G_m(x;a,q)w(x)\mathrm{d}x\\&=\sum_{k=0}^{m}(-a)^{m-k}\begin{bmatrix}m\\k\end{bmatrix}_q q^{\binom{m-k}{2}}\left(\int_a^b x^k w(x)\mathrm{d}x\right),\end{aligned} \tag{2.4}$$

where $\int_a^b x^k w(x)\mathrm{d}x=\mu_k$ denotes the moment of order k with respect to the weight function $w(x)$. Hence, (2.4) becomes

$$A_m(a,q)=\sum_{k=0}^{m}(-a)^{m-k}\begin{bmatrix}m\\k\end{bmatrix}_q q^{\binom{m-k}{2}}\mu_k.$$

Now, let us replace $\{x_i\}=\{aq^i\}_{i=0}^n$ into Newton interpolation formula (2.1) to get

$$f(x)=b_0+b_1G_1(x)+\cdots+b_nG_n(x)+f[x_0,x_1,\ldots,x_n,x]G_{n+1}(x), \tag{2.5}$$

where $b_i=f[a,aq,\ldots,aq^i],\ i=0,1,\ldots,n$. Multiplying by $w(x)$ and then integrating from both sides of (2.5) on $[a,b]$, we obtain

$$\int_a^b w(x)f(x)\mathrm{d}x=\sum_{i=0}^{n}A_i(a,q)f[a,aq,\ldots,aq^i]+\int_a^b f[x_0,x_1,\ldots,x_n,x]G_{n+1}(x)w(x)\mathrm{d}x. \tag{2.6}$$

This formula can still be simplified if one uses the general identity

$$f[x_0,x_1,\ldots,x_n]=\sum_{k=0}^{n}\frac{f(x_k)}{\Psi'_{n+1}(x_k)},$$

in which $\Psi'_{n+1}(x_k)$ is the derivative of the node polynomial $\Psi_{n+1}(x)$ at $x=x_k$. Therefore, summation on the right hand side of (2.6) is simplified as

$$\begin{aligned}\sum_{i=0}^{n}A_i(a,q)f[a,aq,\ldots,aq^i]&=\sum_{i=0}^{n}A_i(a,q)\left(\sum_{k=1}^{i+1}\frac{f(aq^{k-1})}{G'_{i+1}(aq^{k-1})}\right)\\&=\sum_{i=0}^{n}f(aq^i)\left(\sum_{k=i}^{n}\frac{A_k(a,q)}{G'_{k+1}(aq^i)}\right).\end{aligned}$$

Consequently, to compute the coefficients of the quadrature rule

$$\int_a^b w(x)f(x)\mathrm{d}x=\sum_{i=0}^{n}w_i(a,q)f(aq^i)+R_{n+1}[f], \tag{2.7}$$

where

$$R_{n+1}[f]=\int_a^b f[x_0,x_1,\ldots,x_n,x]G_{n+1}(x)w(x)\mathrm{d}x,$$

we finally obtain

$$w_i(a,q)=\sum_{k=i}^{n}\frac{A_k(a,q)}{G'_{k+1}(aq^i)},\quad i=0,1,\ldots,n,$$

where

$$A_m(a,q)=\sum_{k=0}^{m}(-a)^{m-k}\begin{bmatrix} m \\ k \end{bmatrix}_q q^{\binom{m-k}{2}}\mu_k \quad \left(\mu_k=\int_a^b x^k w(x)\mathrm{d}x\right) \tag{2.8}$$

and

$$G_m(x;a,q)=\sum_{k=0}^{m}(-a)^{m-k}\begin{bmatrix} m \\ k \end{bmatrix}_q q^{\binom{m-k}{2}}x^k. \tag{2.9}$$

In this way we have just proved the following statement:

Theorem 2.1. *The coefficients of the interpolatory quadrature formula (2.7) with geometric distributed nodes* $\{x_k\}=\{aq^k\}_{k=0}^n$ *for* $q=\sqrt[n]{b/a}$ *can be expressed in the form*

$$w_i(a,q)=\sum_{k=i}^{n}\frac{A_k(a,q)}{G'_{k+1}(aq^i)}, \tag{2.10}$$

where $A_k(a,q)$ *and* $G_m(x;a,q)$ *are determined by (2.8) and (2.9), respectively.*

For functions $f\in C^{n+1}[a,b]$*, the remainder* $R_{n+1}[f]$ *has the form*

$$R_{n+1}[f]=\frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi(x))G_{n+1}(x;a,q)w(x)\mathrm{d}x,$$

where $\xi(x)\in(a,b)$*. Moreover, if* $|f^{(n+1)}(x)|\le M_{n+1}$ *for each* $x\in(a,b)$*, then*

$$|R_{n+1}[f]|\le\frac{M_{n+1}}{(n+1)!}\int_a^b|G_{n+1}(x;a,q)|w(x)\mathrm{d}x.$$

Using the following program, written by Maple 12, one can easily generate the coefficients $w_i(a,q)$ for any given i and a,b. However, note that in this program we took $N=5$, $w(x)=1$ and $[a,b]=[1,2]$, which can be changed by the user easily.

```
restart:
with(combinat):
Digits:= 10:
N:= 5:
a:= 1.:
b:= 2.:
w:= 1:
q:= (b/a)**(1/N):
for i from 0 to N do
X[i]:= a*q**(i);
od:
for i from 0 to N do
G[i+1]:= product((x-X[k]),k=0..i);
od:
for i from 0 to N do
G[i+1]:= diff(G[i+1],x);
od:
for i from 0 to N do
G[i+1]:= unapply(G[i+1],x);
od:
for i from 0 to N do
A[i]:= add((-a)**(i-k)*q**(numbcomb(i-k,2))*((product((1-q**j),j=1..i))/
((product((1-q**j),j=1..k))*(product((1-q**j),j=1..i-k))))*(int(w*x**k,x=a..b)),
k=0..i);
od:
for i from 0 to N do
W[i]:= add(A[k]/(G[k+1](X[i])),k=i..N);
od;
```

Remark 2.2. The coefficients $\{w_i(a,q)\}_{i=0}^n$ given by (2.10) can be also obtained from the following system of linear equations

$$\int_a^b f(x)w(x)\mathrm{d}x = \sum_{i=0}^n w_i(a,q)f(aq^i), \tag{2.11}$$

for $f(x) = x^j$, $j = 0, 1, \ldots, n$. Thus, replacing $f(x) = x^j$ in (2.11) we obtain

$$\sum_{i=0}^n w_i(a,q)(aq^i)^j = \sum_{i=0}^n w_i x_i^j = \int_a^b x^j w(x\,\mathrm{d}x = \mu_j), \quad j = 0, 1, \ldots, n,$$

which is equivalent to the following Vandermonde matrix system of equations

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_n \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_0^n & x_1^n & x_2^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}.$$

The system has the unique solution because the Vandermonde matrix for the mutually different nodes $x_k = aq^k$, $k = 0, 1, \ldots, n$, where $q = \sqrt[n]{b/a} > 0$, is non-singular. However, such a matrix is ill-conditioned because its condition number increases significantly with n, and therefore this method for determining coefficients $\{w_i(a,q)\}_{i=0}^n$ is not applicable for n sufficiently large.

3. Error analysis

By referring to the shape of quadrature (2.7) one can find an error bound for it in the class of functions from $C^{n+1}[a,b]$. First, according to Theorem 2.1, we have

$$|R_{n+1}[f]| \le \frac{1}{(n+1)!} \max_{x\in[a,b]} |f^{n+1}(x)| \int_a^b |G_{n+1}(x;a,q)|w(x)\mathrm{d}x. \tag{3.1}$$

But, since $G_{n+1}(x)$ is decomposable, the integral of the right hand side of (3.1), i.e.,

$$F_n(q;a,w) = \int_a^b |G_{n+1}(x)|w(x)\mathrm{d}x,$$

can directly be computed. For this purpose, two odd and even cases should be considered for n.

If in the first case n is odd (say $n = 2m+1$) then the mentioned integral is computed as

$$\begin{aligned} F_{2m+1}(q;a,w) &= \int_a^b |(x-a)(x-aq)\cdots(x-aq^{2m+1})|w(x)\mathrm{d}x \\ &= -\int_a^{aq} G_{2m+2}(x)w(x)\mathrm{d}x + \int_{aq}^{aq^2} G_{2m+2}(x)w(x)\mathrm{d}x - \cdots - \int_{aq^{2m}}^{aq^{2m+1}} G_{2m+2}(x)w(x)\mathrm{d}x \\ &= \sum_{k=0}^{2m}(-1)^{k+1}\int_{aq^k}^{aq^{k+1}} G_{2m+2}(x)w(x)\mathrm{d}x \\ &= \sum_{k=0}^{2m}(-1)^{k+1}\int_{aq^k}^{aq^{k+1}} \left(\sum_{j=0}^{2m+2}(-a)^{2m+2-j}\begin{bmatrix}2m+2\\ j\end{bmatrix}_q q^{\binom{2m+2-j}{2}}x^j\right)w(x)\mathrm{d}x \\ &= \sum_{k=0}^{2m}(-1)^{k+1}\left(\sum_{j=0}^{2m+2}(-a)^{2m+2-j}\begin{bmatrix}2m+2\\ j\end{bmatrix}_q q^{\binom{2m+2-j}{2}}\int_{aq^k}^{aq^{k+1}} x^j w(x)\mathrm{d}x\right). \end{aligned}$$

Similarly for $n = 2m$ we have

$$\begin{aligned} F_{2m}(q;a,w) &= \int_a^b |(x-a)(x-aq)\cdots(x-aq^{2m})|w(x)\mathrm{d}x \\ &= \int_a^{aq} G_{2m+1}(x)w(x)\mathrm{d}x - \int_{aq}^{aq^2} G_{2m+1}(x)w(x)\mathrm{d}x + \cdots - \int_{aq^{2m-1}}^{aq^{2m}} G_{2m+1}(x)w(x)\mathrm{d}x \\ &= \sum_{k=0}^{2m-1}(-1)^k\int_{aq^k}^{aq^{k+1}} G_{2m+1}(x)w(x)\mathrm{d}x \end{aligned}$$

$$= \sum_{k=0}^{2m-1}(-1)^k \int_{aq^k}^{aq^{k+1}} \left(\sum_{j=0}^{2m+1} (-a)^{2m+1-j} \begin{bmatrix} 2m+1 \\ j \end{bmatrix}_q q^{\binom{2m+1-j}{2}} x^j \right) w(x)\mathrm{d}x$$
$$= \sum_{k=0}^{2m-1}(-1)^k \left(\sum_{j=0}^{2m+1} (-a)^{2m+1-j} \begin{bmatrix} 2m+1 \\ j \end{bmatrix}_q q^{\binom{2m+1-j}{2}} \int_{aq^k}^{aq^{k+1}} x^j w(x)\mathrm{d}x \right).$$

Combining these two latter equalities, we finally obtain

$$\begin{aligned} F_n(q;\, a,\, w) &= \int_a^b |G_{n+1}(x)| w(x)\mathrm{d}x \\ &= \sum_{k=0}^{n-1}(-1)^{n+k} \left(\sum_{j=0}^{n+1} (-a)^{n+1-j} \begin{bmatrix} n+1 \\ j \end{bmatrix}_q q^{\binom{n+1-j}{2}} \int_{aq^k}^{aq^{k+1}} x^j w(x)\mathrm{d}x \right) \\ &= (-1)^n \sum_{j=0}^{n+1} (-a)^{n+1-j} \begin{bmatrix} n+1 \\ j \end{bmatrix}_q q^{\binom{n+1-j}{2}} \left(\sum_{k=0}^{n-1} (-1)^k \int_{aq^k}^{aq^{k+1}} x^j w(x)\mathrm{d}x \right). \end{aligned}$$

In a special case for $w(x) = 1$ we get the following explicit formula

$$F_n(q;\, a,\, 1) = a^{n+2} \sum_{j=0}^{n+1} (-1)^{j+1} \begin{bmatrix} n+1 \\ j \end{bmatrix}_q q^{\binom{n+1-j}{2}} \frac{(-1)^{n+1} q^{(j+1)n} + 1}{j+1} \cdot \frac{q^{j+1}-1}{q^{j+1}+1},$$

so that

$$|R_{n+1}[f]| = \left| \int_a^b f(x)\mathrm{d}x - \sum_{k=0}^{n} w_k(a, q) f(aq^k) \right| \le \frac{M_{n+1}}{(n+1)!} F_n(q;\, a,\, 1),$$

where $|f^{(n+1)}(x)| \le M_{n+1}$.

4. Applications

In order to illustrate the application of our quadrature formula with geometric distribution of nodes, we consider two examples.

Example 4.1. In this example we consider application of the (non-weighted) quadrature formula (2.7) to $\int_a^b f(x)\mathrm{d}x$ for a few standard functions. We take nodes as $x_k = aq^k$, $k = 0, 1, \ldots, n$, with $q = \sqrt[n]{b/a}$ and $n = 5, 10, 15$, and 20. The absolute errors of quadrature approximations are displayed in Table 4.1.

Evidently, the fastest convergence is for analytic (holomorphic) functions (exponential and trigonometric functions), and the slowest convergence is for the function $1/x$, because of the influence of its singularity at the origin.

Example 4.2. We apply now our quadrature (2.7) to numerical calculation of the integral

$$I = \int_1^3 \mathrm{e}^{-x} \log x \,\mathrm{d}x = 0.15163886817562858131\ldots.$$

Since the both functions e^{-x} and $\log x$ are nonnegative on the interval $[1, 3]$, we illustrate three possible applications:

Case 1. $f(x) = \mathrm{e}^{-x} \log x$, $w(x) = 1$;

Case 2. $f(x) = \log x$, $w(x) = \mathrm{e}^{-x}$;

Case 3. $f(x) = \mathrm{e}^{-x}$, $w(x) = \log x$.

Table 4.1
Absolute errors of geometric (non-weighted) quadrature rules for $n = 5(5)20$.

(a, b)	$f(x)$	$n = 5$	$n = 10$	$n = 15$	$n = 20$
$(1, 2)$	$\sqrt{x}$	8.62×10^{-7}	2.20×10^{-10}	1.62×10^{-13}	1.68×10^{-16}
$(1, 2)$	$\sqrt[3]{x}$	9.28×10^{-7}	2.66×10^{-10}	2.10×10^{-13}	2.28×10^{-16}
$(3, 5)$	e^x	2.98×10^{-3}	5.04×10^{-9}	1.87×10^{-15}	1.75×10^{-22}
$(\pi/3, \pi)$	$\sin x$	1.22×10^{-4}	2.99×10^{-10}	5.41×10^{-16}	5.14×10^{-23}
$(\pi/3, \pi)$	$\cos x$	6.83×10^{-5}	6.32×10^{-10}	2.39×10^{-16}	1.22×10^{-22}
$(1, 3)$	$\log x$	2.97×10^{-4}	2.06×10^{-6}	3.14×10^{-8}	6.58×10^{-10}
$(1, 3)$	$1/x$	1.02×10^{-3}	1.32×10^{-5}	2.97×10^{-7}	8.21×10^{-9}

Table 4.2
Absolute errors e_n of quadrature sums I_n, $n = 5(5)20$, in three different cases.

(a, b)	$f(x)$	$w(x)$	$n = 5$	$n = 10$	$n = 15$	$n = 20$
(1, 3)	$e^{-x} \log x$	1	4.69×10^{-4}	2.50×10^{-6}	3.55×10^{-8}	7.19×10^{-10}
(1, 3)	$\log x$	e^{-x}	2.13×10^{-5}	1.08×10^{-7}	1.67×10^{-8}	3.44×10^{-11}
(1, 3)	e^{-x}	$\log x$	1.50×10^{-5}	6.40×10^{-11}	4.47×10^{-17}	8.07×10^{-24}
(1, 5)	e^{-x}	$\log x$	2.38×10^{-3}	5.42×10^{-7}	1.96×10^{-11}	1.81×10^{-16}

Applying the corresponding quadrature for $n = 5$, 10, 15, and 20, we obtain results I_n with absolute errors $e_n = |I - I_n|$, which are presented in Table 4.2.

As we can see the best results are obtained in the third case. It is quite clear because in that case we have an integration of the exponential function, which is an entire function (holomorphic over the whole complex plane). Even taking double interval of integration, i.e.,

$$I' = \int_1^5 e^{-x} \log x \, dx = 0.20739133145194522248\ldots,$$

the convergence is fast (last row in Table 4.2). In other cases the integrand has a logarithmic behavior and therefore the convergence is significantly slower.

Acknowledgements

The research of the first author was partially supported by a grant from IPM, No. 89330021. The second author was partially supported by the Serbian Ministry of Science and Technological Developments (Project: Approximation of integral and differential operators and applications, grant number #174015).

References

[1] W. Gautschi, Numerical Analysis: An Introduction, Birkhauser, Boston, 1997.
[2] G. Mastroianni, G.V. Milovanović, Interpolation Processs: Basic Theory and Applications, in: Springer Monographs in Mathematics, Springer-Verlag, Berlin, Heidelberg, 2008.
[3] M. Masjed-Jamei, A new type of weighted quadrature rules and its relation with orthogonal polynomials, Appl. Math. Comput. 188 (2007) 154–165.
[4] M. Masjed-Jamei, On constructing new interpolation formulas using linear operators and an operator type of quadrature rules, J. Comput. Appl. Math. 216 (2008) 307–318.
[5] G.V. Milovanović, A.S. Cvetković, Nonstandard Gaussian quadrature formulae based on operator values, Adv. Comput. Math. 32 (2010) 431–486.
[6] G.V. Milovanović, A.S. Cvetković, Gaussian quadrature rules using function derivatives, IMA J. Numer. Anal., in press (doi:10.1093/imanum/drp036).
[7] R. Koekoek, R.F. Swarttouw, The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue, Report No. 98-17, Technical Universiteit Delft, Faculty of Technical Mathematics and Informatics, Delft, 1998. Web site: http://aw.twi.tudelft.nl/koekoek/askey/.