Electronic Transactions on Numerical Analysis.
Volume 45, pp. 476–498, 2016.

ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

WEIGHTED HERMITE QUADRATURE RULES*

MOHAMMAD MASJED-JAMEI† AND GRADIMIR V. MILOVANOVIĆ‡

Abstract. In this paper, a new representation of Hermite osculatory interpolation is presented in order to construct weighted Hermite quadrature rules. Then, explicit forms of several special cases of the established quadrature are obtained such as weighted Hermite quadrature rules with arithmetic and geometric nodes as well as standard Gauss-Christoffel quadrature rules and Gaussian quadratures rules using only function derivatives. Some numerical examples are also given for the above mentioned cases.

Key words. weighted Hermite quadrature rule, Hermite interpolation, Gaussian quadrature, divided differences, distribution of nodes

AMS subject classifications. 65D05, 65D30, 41A55, 65D32

1. Introduction.

Due to the well-known Weierstrass theorems on the uniform approximation of continuous functions on a finite interval by algebraic and trigonometric polynomials, polynomial functions are very popular in "Approximation Theory" and one of the basic subjects of this theory is "interpolation of functions by polynomials". While the classical theory of interpolation deals mainly with equidistant nodes, for which there exist many results on divergent interpolation processes (see, e.g., the books by Davis [2] and Szabados and Vértesi [23]), in recent years new results on convergent interpolation processes in the uniform and other integral norms have appeared. Such processes are treated with basic tools such as orthogonal polynomials, moduli of smoothness, K-functionals, etc.; cf. Mastroianni and Milovanović [15]. The most common interpolation processes are related to Lagrange interpolation polynomials, and, in general, to Hermite interpolation polynomials. However, there are also special interpolation problems— so-called Birkhoff interpolation; cf. the books by Lorentz, Jetter, and Riemenschneider [11] and Shi [21].

Lagrange interpolation is a well-known classical method for the approximation of continuous functions $f:[a,b]\to\mathbb{R}$ at n distinct nodes $a\le x_1^{(n)}<x_2^{(n)}<\cdots<x_n^{(n)}\le b$ such that

$$L_n(f;x_\nu^{(n)})=f(x_\nu^{(n)}),\qquad \nu=1,2,\dots,n, \tag{1.1}$$

where the Lagrange interpolation polynomial of degree at most $n-1$ can be expressed in the form [15, pp. 39–40]

$$L_n(f;x)=\sum_{\nu=1}^{n} f(x_\nu^{(n)})\ell_\nu^{(n)}(x)\qquad \text{for}\quad \ell_\nu^{(n)}(x)=\frac{\omega_n(x)}{(x-x_\nu^{(n)})\omega_n'(x_\nu^{(n)})},$$

with the node polynomial $\omega_n(x)=(x-x_1^{(n)})(x-x_2^{(n)})\cdots(x-x_n^{(n)})$. In $\mathcal{P}_{n-1}$, the space of all polynomials of degree at most $n-1$, $L_n(f;x)$ is unique among all polynomials solving the interpolation problem (1.1).

A simple Hermite interpolation of a function $f:[a,b]\to\mathbb{R}$ at n nodes $\left\{x_\nu^{(n)}\right\}_{\nu=1}^{n}$ in $[a,b]$, which satisfies the conditions

$$H_{2n-1}(f;x_\nu^{(n)})=f(x_\nu^{(n)}),\qquad H_{2n-1}'(f;x_\nu^{(n)})=f'(x_\nu^{(n)}),\qquad \nu=1,\dots,n, \tag{1.2}$$

*Received May 31, 2016. Accepted November 11, 2016. Published online on December 9, 2016. Recommended by F. Marcellan. The work of the second author was supported in part by the Serbian Ministry of Education, Science and Technological Development.

†K. N. Toosi University of Technology, P.O. Box 16315–1618, Teheran, Iran (mmjamei@kntu.ac.ir).

‡Serbian Academy of Sciences and Arts, 11000 Belgrade, Serbia & State University of Novi Pazar, 36300 Novi Pazar, Serbia (gvm@mi.sanu.ac.rs).

ETNA
Kent State University
http://etna.math.kent.edu

can be established by (cf. Davis [2, p. 28 & pp. 36–37])

$$H_{2n-1}(f;x) = \sum_{\nu=1}^{n} f(x_\nu^{(n)}) \left(1 - \frac{\omega_n''(x_\nu^{(n)})}{\omega_n'(x_\nu^{(n)})}(x - x_\nu^{(n)})\right) \left(\ell_\nu^{(n)}(x)\right)^2 + \sum_{\nu=1}^{n} f'(x_\nu^{(n)})(x - x_\nu^{(n)}) \left(\ell_\nu^{(n)}(x)\right)^2. \tag{1.3}$$

It is known that the two polynomials

$$H_{\nu,0}(x) = \left(1 - \frac{\omega_n''(x_\nu^{(n)})}{\omega_n'(x_\nu^{(n)})}(x - x_\nu^{(n)})\right) \left(\ell_\nu^{(n)}(x)\right)^2 \quad \text{and} \qquad H_{\nu,1}(x) = (x - x_\nu^{(n)}) \left(\ell_\nu^{(n)}(x)\right)^2, \tag{1.4}$$

of degree $2n-1$ and the functionals

$$L_\nu(f) = f(x_\nu^{(n)}) \quad \text{and} \quad M_\nu(f) = f'(x_\nu^{(n)}), \qquad \nu = 1, 2, \ldots, n,$$

are biorthonormal (cf. Davis [2, p. 37]), and therefore (1.3) is unique as an element in $\mathcal{P}_{2n-1}$ solving the so-called osculatory interpolation problem (1.2). There are several references related to polynomial Hermite interpolation in one and several variables, including barycentric implementations; cf. [3, 4, 8, 9, 19, 18, 20, 23]. Recently, a new fast implementation of the second barycentric formula for higher-order Hermite-Fejér interpolation at Gauss-Jacobi or Jacobi-Gauss-Lobatto point systems has been presented in [25].

This paper is organized as follows. In the next section, we obtain a new representation for the Hermite interpolation polynomial (1.3) in order to construct weighted Hermite quadrature rules having several special cases. Indeed, by using this representation, we introduce a class of weighted interpolatory quadrature formulas in Section 3. Special cases of such formulas are, respectively, considered in Section 4 with equidistant nodes and in Section 5 with geometric nodes. Standard Gauss-Christoffel quadrature rules are briefly described in Section 6 and Gaussian quadrature rules using only function derivatives are finally considered in Section 7. Several numerical examples with software implementation are given in the sequel.

2. A new representation for Hermite interpolation. In this section, we obtain an alternative form of the Hermite interpolation polynomial (1.3). In order to shorten the notation, for the nodes, we will omit the superscript n and write simply x_ν instead of $x_\nu^{(n)}$.

Let us start with n distinct points $\{x_\nu\}_{\nu=1}^n$ in $[a,b]$ and introduce the node polynomials as

$$\Phi_k(x) = (x - x_k)\Phi_{k-1}(x) = \prod_{\nu=1}^{k}(x - x_\nu), \tag{2.1}$$

with $\Phi_0(x) = 1$ and $\Phi_n(x) = \omega(x) = (x - x_1)(x - x_2)\cdots(x - x_n)$.

We see that the polynomials (1.4) can be first represented as

$$H_{\nu,0}(x) = \left(1 - \frac{\Phi_n''(x_\nu)}{\Phi_n'(x_\nu)}(x - x_\nu)\right) \left(\frac{\Phi_n(x)}{(x - x_\nu)\Phi_n'(x_\nu)}\right)^2 \quad \text{and} \qquad H_{\nu,1}(x) = \frac{1}{x - x_\nu} \left(\frac{\Phi_n(x)}{\Phi_n'(x_\nu)}\right)^2.$$

ETNA
Kent State University
http://etna.math.kent.edu

Hence, their substitution into (1.3) gives

$$H_{2n-1}(f;x)=\sum_{\nu=1}^{n}\left(U_{n,\nu}\left(\frac{\Phi_n(x)}{x-x_\nu}\right)^2+V_{n,\nu}\left(\frac{\Phi_n(x)^2}{x-x_\nu}\right)\right), \tag{2.2}$$

where

$$U_{n,\nu}=\frac{1}{\Phi_n'(x_\nu)^2}f(x_\nu)\quad\text{and}\quad V_{n,\nu}=-\frac{\Phi_n''(x_\nu)}{\Phi_n'(x_\nu)^3}f(x_\nu)+\frac{1}{\Phi_n'(x_\nu)^2}f'(x_\nu). \tag{2.3}$$

On the other hand, by denoting the distribution of nodes as

$$z_{2\nu-1}=z_{2\nu}=x_\nu,\qquad \nu=1,2,\dots,n,$$

we can refer to divided differences and write $H_{2n-1}(f;x)$ as

$$\begin{aligned}H_{2n-1}(f;x)=b_0+b_1(x-x_1)+b_2(x-x_1)^2+b_3(x-x_1)^2(x-x_2)+\cdots\\ +b_{2n-2}(x-x_1)^2(x-x_2)^2\cdots(x-x_{n-1})^2\\ +b_{2n-1}(x-x_1)^2(x-x_2)^2\cdots(x-x_{n-1})^2(x-x_n),\end{aligned}$$

which is equivalent to

$$H_{2n-1}(f;x)=\sum_{k=1}^{n}\left(b_{2k-2}\Phi_{k-1}(x)^2+b_{2k-1}\Phi_{k-1}(x)\Phi_k(x)\right), \tag{2.4}$$

in which b_{2k-2} and b_{2k-1}, for $k=1,2,\dots,n$, denote divided differences, i.e.,

$$b_{2k-2}=f[x_1,x_1,x_2,x_2,\dots,x_{k-1},x_{k-1},x_k]$$

and

$$b_{2k-1}=f[x_1,x_1,x_2,x_2,\dots,x_{k-1},x_{k-1},x_k,x_k],$$

respectively. Note that due to (2.1), the polynomial (2.4) can be rearranged as

$$H_{2n-1}(f;x)=\sum_{k=1}^{n}\left(b_{2k-2}+b_{2k-1}(x-x_k)\right)\Phi_{k-1}(x)^2. \tag{2.5}$$

Considering $H_{2n-1}(f;x)$ as a polynomial in x, the leading coefficient is evidently b_{2n-1} and the following one, i.e., the coefficient of x^{2n-2} is

$$b_{2n-2}-\left(2(x_1+\cdots+x_{n-1})+x_n\right)b_{2n-1}$$

because

$$\begin{aligned}(x-x_n)\Phi_{n-1}(x)^2&=(x-x_n)\left(x^{n-1}-(x_1+\ldots+x_{n-1})x^{n-2}+\cdots\right)^2\\ &=x^{2n-1}-\left(2(x_1+\cdots+x_{n-1})+x_n\right)x^{2n-2}+\cdots.\end{aligned}$$

Since

$$\begin{aligned}\frac{\Phi_n(x)}{x-x_\nu}&=x^{n-1}-\left(\sum_{k=1}^{n}x_k-x_\nu\right)x^{n-2}+\cdots,\\ \left(\frac{\Phi_n(x)}{x-x_\nu}\right)^2&=x^{2n-2}-2\left(\sum_{k=1}^{n}x_k-x_\nu\right)x^{2n-3}+\cdots,\\ \frac{\Phi_n(x)^2}{x-x_\nu}&=x^{2n-1}-\left(2\sum_{k=1}^{n}x_k-x_\nu\right)x^{2n-2}+\cdots,\end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

from (2.2) we obtain

$$H_{2n-1}(f;x) = \left(\sum_{\nu=1}^{n} V_{n,\nu}\right) x^{2n-1} + \left(\sum_{\nu=1}^{n} \left(U_{n,\nu} - \left(2\sum_{k=1}^{n} x_k - x_\nu\right) V_{n,\nu}\right)\right) x^{2n-2} + \cdots .$$

Thus, comparing it with (2.5) yields

$$\begin{aligned} b_{2n-1} &= \sum_{\nu=1}^{n} V_{n,\nu}, \\ b_{2n-2} &= \left(2\sum_{k=1}^{n} x_k - x_n\right) b_{2n-1} + \sum_{\nu=1}^{n} \left(U_{n,\nu} - \left(2\sum_{k=1}^{n} x_k - x_\nu\right) V_{n,\nu}\right). \end{aligned}$$

In other words

$$b_{2n-1} = \sum_{\nu=1}^{n} V_{n,\nu} \qquad \text{and} \qquad b_{2n-2} = \sum_{\nu=1}^{n} \left(U_{n,\nu} + (x_\nu - x_n) V_{n,\nu}\right), \tag{2.6}$$

where $U_{n,\nu}$ and $V_{n,\nu}$ are given in (2.3). This leads to the following result:

LEMMA 2.1. *The identities*

$$b_{2k-2} = f[x_1, x_1, x_2, x_2, \ldots, x_{k-1}, x_{k-1}, x_k] = \sum_{\nu=1}^{k} \left(p_{k,\nu} f(x_\nu) + q_{k,\nu} f'(x_\nu)\right)$$

and

$$b_{2k-1} = f[x_1, x_1, x_2, x_2, \ldots, x_{k-1}, x_{k-1}, x_k, x_k] = \sum_{\nu=1}^{k} \left(r_{k,\nu} f(x_\nu) + s_{k,\nu} f'(x_\nu)\right)$$

hold true, where

$$\begin{aligned} p_{k,\nu} &= \frac{\Phi_k'(x_\nu) + (x_k - x_\nu)\Phi_k''(x_\nu)}{\Phi_k'(x_\nu)^3}, & q_{k,\nu} &= -\frac{x_k - x_\nu}{\Phi_k'(x_\nu)^2}, \\ r_{k,\nu} &= -\frac{\Phi_k''(x_\nu)}{\Phi_k'(x_\nu)^3}, & s_{k,\nu} &= \frac{1}{\Phi_k'(x_\nu)^2}, \end{aligned} \tag{2.7}$$

and $\Phi_k(x) = (x - x_1)\cdots(x - x_k)$.

Proof. Using (2.3), from (2.6) we can directly get the result for $k = n$. Also it is clear that the result holds for each $k \le n$. ∎

In the sequel, we make use of the following definition

$$h_{1,1} = 0, \quad h_{k,\nu} = \frac{\Phi_k''(x_\nu)}{\Phi_k'(x_\nu)} = 2\sum_{\substack{i=1 \\ i\neq\nu}}^{k} \frac{1}{x_\nu - x_i}, \qquad 2 \le \nu \le k \le n. \tag{2.8}$$

REMARK 2.2. By using (2.8) and the coefficients $p_{k,\nu}$, $q_{k,\nu}$, $r_{k,\nu}$, and $s_{k,\nu}$ in (2.7), we can define a 2×2 matrix as

$$Q_{k,\nu} = \begin{bmatrix} p_{k,\nu} & q_{k,\nu} \\ r_{k,\nu} & s_{k,\nu} \end{bmatrix} = \frac{P_{k,\nu}}{\Phi_k'(x_\nu)^2}, \qquad 1 \le \nu \le k,$$

where

$$P_{k,\nu} = \begin{bmatrix} 1 + (x_k - x_\nu)h_{k,\nu} & x_\nu - x_k \\ -h_{k,\nu} & 1 \end{bmatrix} \quad \text{and} \quad \Phi_k'(x_\nu) = \prod_{\substack{i=1 \\ i\neq\nu}}^{k} (x_\nu - x_i), \qquad 1 \le \nu \le k.$$

ETNA
Kent State University
http://etna.math.kent.edu

For example, the matrices $P_{k,\nu}$, for $k \le 4$, are given by

$$P_{1,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad P_{2,1} = \begin{bmatrix} -1 & x_1 - x_2 \\ -\frac{2}{x_1 - x_2} & 1 \end{bmatrix}, \qquad P_{2,2} = \begin{bmatrix} 1 & 0 \\ \frac{2}{x_1 - x_2} & 1 \end{bmatrix},$$

$$P_{3,1} = \begin{bmatrix} \frac{-3x_1 + x_2 + 2x_3}{x_1 - x_2} & x_1 - x_3 \\ -2\left(\frac{1}{x_1 - x_3} + \frac{1}{x_1 - x_2}\right) & 1 \end{bmatrix},$$

$$P_{3,2} = \begin{bmatrix} -\frac{x_1 - 3x_2 + 2x_3}{x_1 - x_2} & x_2 - x_3 \\ -2\left(\frac{1}{x_2 - x_3} + \frac{1}{x_2 - x_1}\right) & 1 \end{bmatrix},$$

$$P_{3,3} = \begin{bmatrix} 1 & 0 \\ -2\left(\frac{1}{x_3 - x_2} + \frac{1}{x_3 - x_1}\right) x_3 & 1 \end{bmatrix},$$

$$P_{4,1} = \begin{bmatrix} 2\left(\frac{1}{x_2 - x_3} + \frac{1}{x_1 - x_4} + \frac{1}{x_1 - x_2}\right)(x_4 - x_1) + 1 & x_1 - x_4 \\ -2\left(\frac{1}{x_1 - x_3} + \frac{1}{x_1 - x_4} + \frac{1}{x_1 - x_2}\right) & 1 \end{bmatrix},$$

$$P_{4,2} = \begin{bmatrix} 2\left(\frac{1}{x_2 - x_3} + \frac{1}{x_2 - x_4} + \frac{1}{x_2 - x_1}\right)(x_4 - x_2) + 1 & x_2 - x_4 \\ -2\left(\frac{1}{x_2 - x_3} + \frac{1}{x_2 - x_4} + \frac{1}{x_2 - x_1}\right) & 1 \end{bmatrix},$$

$$P_{4,3} = \begin{bmatrix} 2\left(\frac{1}{x_3 - x_2} + \frac{1}{x_3 - x_4} + \frac{1}{x_3 - x_1}\right)(x_4 - x_3) + 1 & x_3 - x_4 \\ -2\left(\frac{1}{x_3 - x_2} + \frac{1}{x_3 - x_4} + \frac{1}{x_3 - x_1}\right) & 1 \end{bmatrix},$$

$$P_{4,4} = \begin{bmatrix} 1 & 0 \\ -2\left(\frac{1}{x_4 - x_2} + \frac{1}{x_4 - x_3} + \frac{1}{x_4 - x_1}\right) & 1 \end{bmatrix}, \qquad \text{etc.}$$

Introducing the two-dimensional vectors $\mathbf{f}_\nu = [f(x_\nu)\ f'(x_\nu)]^T$, for $\nu = 1, \dots, n$, and $\mathbf{b}_k = [b_{2k-2}\ b_{2k-1}]^T$, for $k = 1, \dots, n$, the result of Lemma 2.1 can be interpreted as follows:

$$\mathbf{b}_k = \sum_{\nu=1}^{k} Q_{k,\nu}\, \mathbf{f}_\nu, \qquad k = 1, \dots, n, \tag{2.9}$$

which in block-matrix form reads as

$$\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} Q_{1,1} & & & \\ Q_{2,1} & Q_{2,2} & & \\ \vdots & & \ddots & \\ Q_{n,1} & Q_{n,2} & & Q_{n,n} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}.$$

Now, an alternative form of the Hermite interpolation polynomial is stated in the following theorem.

THEOREM 2.3. *The unique Hermite interpolation polynomial satisfying* (1.2) *can be represented in terms of the node polynomial* (2.1) *as*

$$H_{2n-1}(f; x) = \sum_{\nu=1}^{n} f(x_\nu) H_{\nu,0}(x) + \sum_{\nu=1}^{n} f'(x_\nu) H_{\nu,1}(x), \tag{2.10}$$

where

$$H_{\nu,0}(x) = \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi'_k(x_\nu)}\right)^2 \left(1 - (x - 2x_k + x_\nu)\, h_{k,\nu}\right), \tag{2.11}$$

ETNA
Kent State University
http://etna.math.kent.edu

$$H_{\nu,1}(x) = \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 (x - 2x_k + x_\nu), \tag{2.12}$$

and $h_{k,\nu}$ is defined by (2.8). Moreover, for functions $f \in C^{2n}[a,b]$, the error term in

$$f(x) = H_{2n-1}(f;x) + r_n(x) \tag{2.13}$$

can be expressed in the form

$$r_n(x) = \frac{f^{(2n)}(\xi_x)}{(2n)!}\, \Phi_n(x)^2, \qquad a < \xi_x < b. \tag{2.14}$$

Proof. According to (2.4) and Lemma 2.1, the following representation holds for the Hermite polynomials (1.3):

$$H_{2n-1}(x;f) = \sum_{k=1}^{n} \Psi_k(x;f)\Phi_{k-1}(x)^2,$$

in which $\Psi_k(x;f)$ is a linear function in x, given by

$$\begin{aligned} \Psi_k(x;f) &= b_{2k-1}x + (b_{2k-2} - b_{2k-1}x_k) = b_{2k-1}(x - x_k) + b_{2k-2} \\ &= [1 \quad x - x_k]\, [b_{2k-2} \quad b_{2k-1}]^T. \end{aligned}$$

But using (2.9) and Remark 2.2, the above relation can be expressed in the form

$$\begin{aligned} \Psi_k(x;f) &= [1 \quad x - x_k] \sum_{\nu=1}^{k} Q_{k,\nu}\, \mathbf{f}_\nu = \sum_{\nu=1}^{k} \frac{1}{\Phi_k'(x_\nu)^2}\, [1 \quad x - x_k]\, P_{k,\nu}\, \mathbf{f}_\nu \\ &= \sum_{\nu=1}^{k} \frac{1}{\Phi_k'(x_\nu)^2} \Big(\big(1 - (x - 2x_k + x_\nu)h_{k,\nu}\big) f(x_\nu) + (x - 2x_k + x_\nu) f'(x_\nu) \Big), \end{aligned}$$

where $h_{k,\nu}$ is defined by (2.8). Thus, we have

$$\begin{aligned} &H_{2n-1}(x;f) \\ &= \sum_{k=1}^{n} \sum_{\nu=1}^{k} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 \Big(\big(1 - (x - 2x_k + x_\nu)h_{k,\nu}\big) f(x_\nu) + (x - 2x_k + x_\nu) f'(x_\nu) \Big) \\ &= \sum_{\nu=1}^{n} \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 \Big(\big(1 - (x - 2x_k + x_\nu)h_{k,\nu}\big) f(x_\nu) + (x - 2x_k + x_\nu) f'(x_\nu) \Big), \end{aligned}$$

which yields the same expression as (2.10), where $H_{\nu,0}(x)$ and $H_{\nu,1}(x)$ are given by (2.11) and (2.12), respectively. For functions $f \in C^{2n}[a,b]$, it is clear that (2.13) and (2.14) hold; cf. Davis [2, p. 67]. ∎

REMARK 2.4. The total number of arithmetic operations in (2.10) is slightly bigger than in (1.3), but in practical applications when we wish to add an additional node x_{n+1}, the representation (2.10) has a computational advantage with respect to the classical form (1.3). Namely, adding another point $(x_{n+1}; f(x_{n+1}), f'(x_{n+1}))$ requires to only calculate the term

$$\begin{aligned} \Psi_{n+1}(x;f) &= [1 \quad x - x_{n+1}]\, [b_{2n} \quad b_{2n+1}]^T \\ &= \sum_{\nu=1}^{n+1} \frac{1}{\Phi_{n+1}'(x_\nu)^2} \Big(\big(1 - (x - 2x_{n+1} + x_\nu)h_{n+1,\nu}\big) f(x_\nu) + (x - 2x_{n+1} + x_\nu) f'(x_\nu) \Big), \end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

as well as $\Phi_n(x) = (x - x_n)\Phi_{n-1}(x)$. This means that

$$H_{2n+1}(x; f) = H_{2n-1}(x; f) + \Psi_{n+1}(x; f)\Phi_n(x)^2.$$

3. Weighted Hermite quadrature rules. In this section, we introduce a class of weighted interpolatory quadrature formulae on $[a, b]$ as

$$I(w; f) = \int_a^b w(x)f(x)\mathrm{d}x = \sum_{\nu=1}^{n} (A_\nu f(x_\nu) + B_\nu f'(x_\nu)) + R_n(f), \tag{3.1}$$

where $w : (a, b) \to \mathbb{R}^+$ is a given weight function for which all moments

$$\mu_k = \int_a^b w(x)\, x^k \mathrm{d}x, \qquad k = 0, 1, \ldots, \tag{3.2}$$

exist and $\mu_0 > 0$. Indeed, we obtain *explicit* forms of the weight coefficients $\{A_\nu, B_\nu\}_{\nu=1}^n$.

For this purpose, integrating (2.13) with respect to the weight function $w(x)$ and using (2.10) yield

$$I(w; f) = \sum_{\nu=1}^{n} \left(\int_a^b w(x) H_{\nu,0}(x)\mathrm{d}x \right) f(x_\nu) + \sum_{\nu=1}^{n} \left(\int_a^b w(x) H_{\nu,1}(x)\mathrm{d}x \right) f'(x_\nu) + R_n(f),$$

where $R_n(f) = \int_a^b w(x) r_n(x)\mathrm{d}x$.

Now, using (2.11) and (2.12), we obtain

$$\begin{aligned} A_\nu &= \int_a^b w(x)H_{\nu,0}(x)\mathrm{d}x = \int_a^b w(x) \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 (1 - (x - 2x_k + x_\nu)h_{k,\nu})\, \mathrm{d}x \\ &= \sum_{k=\nu}^{n} \frac{1}{\Phi_k'(x_\nu)^2} \int_a^b w(x)\, (1 - (x - 2x_k + x_\nu)h_{k,\nu})\, \Phi_{k-1}(x)^2 \mathrm{d}x \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} B_\nu &= \int_a^b w(x)H_{\nu,1}(x)\mathrm{d}x = \int_a^b w(x) \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 (x - 2x_k + x_\nu)\mathrm{d}x \\ &= \sum_{k=\nu}^{n} \frac{1}{\Phi_k'(x_\nu)^2} \int_a^b w(x)(x - 2x_k + x_\nu)\Phi_{k-1}(x)^2 \mathrm{d}x, \end{aligned} \tag{3.4}$$

where $h_{k,\nu}$ is defined by (2.8).

The relations (3.3) and (3.4) show that only the two following values have to be computed to complete formula (3.1):

$$r_k = \int_a^b w(x)x\, \Phi_{k-1}(x)^2 \mathrm{d}x \qquad \text{and} \qquad s_k = \int_a^b w(x)\, \Phi_{k-1}(x)^2 \mathrm{d}x.$$

These quantities can be directly obtained from the moments (3.2).

THEOREM 3.1. *Let $c_j^{(k)}$, $j = 0, 1, \ldots, k$, be the coefficients of the node polynomial $\Phi_k(x)$, i.e.,*

$$\Phi_k(x) = \prod_{j=1}^{k} (x - x_j) = \sum_{j=0}^{k} c_j^{(k)} x^j, \qquad k = 1, \ldots, n,$$

ETNA
Kent State University
http://etna.math.kent.edu

in which $c_k^{(k)} = 1$, *for* $k = 1, \ldots, n$, *and* $\Phi_0(x) = 1$. *Also let*

$$\lambda_m^{(k)} = \sum_{j=0}^{m} c_j^{(k)} c_{m-j}^{(k)}, \qquad m = 0, 1, \ldots, 2k. \tag{3.5}$$

Then the weighting coefficients A_ν *and* B_ν, *for* $\nu = 1, \ldots, n$, *in* (3.1) *can be computed as*

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \big(-h_{k,\nu}\mu_{m+1} + (1 + (2x_k - x_\nu)h_{k,\nu})\,\mu_m\big) \tag{3.6}$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \big(\mu_{m+1} - (2x_k - x_\nu)\mu_m\big), \tag{3.7}$$

where $h_{k,\nu}$ *is defined by* (2.8).

Proof. Since

$$\Phi_k(x)^2 = \left(\sum_{j=0}^{k} c_j^{(k)} x^j\right)\left(\sum_{i=0}^{k} c_i^{(k)} x^i\right) = \sum_{m=0}^{2k}\left(\sum_{j=0}^{m} c_j^{(k)} c_{m-j}^{(k)}\right) x^m = \sum_{m=0}^{2k} \lambda_m^{(k)} x^m,$$

by using the moments (3.2), we obtain

$$r_k = \int_a^b w(x)x\,\Phi_{k-1}(x)^2 \mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \int_a^b w(x)x^{m+1}\mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)}\mu_{m+1} \tag{3.8}$$

and

$$s_k = \int_a^b w(x)\Phi_{k-1}(x)^2 \mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \int_a^b w(x)x^{m}\mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)}\mu_{m}. \tag{3.9}$$

Hence, the weight coefficients A_ν and B_ν are respectively simplified as

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \big(-h_{k,\nu}r_k + (1 + (2x_k - x_\nu)h_{k,\nu})\,s_k\big)$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \big(r_k - (2x_k - x_\nu)s_k\big),$$

which yield (3.6) and (3.7). ∎

REMARK 3.2. By referring to (3.5) and (3.9), note that we have

$$\int_a^b w(x)\Phi_n(x)^2 \mathrm{d}x = s_{n+1} = \sum_{m=0}^{2n} \lambda_m^{(n)}\mu_m.$$

Therefore, if $f \in C^{2n}[a, b]$, then according to (2.14), the remainder term $R_n(f)$ of the quadrature rule (3.1) can be expressed in the form

$$R_n(f) = \frac{1}{(2n)!}\int_a^b w(x) f^{(2n)}(\xi_x)\Phi_n(x)^2 \mathrm{d}x = \frac{f^{(2n)}(\xi)}{(2n)!} \sum_{m=0}^{2n} \lambda_m^{(n)}\mu_m,$$

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

ETNA
Kent State University
http://etna.math.kent.edu

4. Weighted Hermite-Newton quadrature rules with equidistant nodes. Recently in [14], we derived closed expressions for the Cotes numbers in the weighted Newton-Cotes quadrature formulae in terms of moments and Stirling numbers of the first kind for the following three types of equidistant nodes:

$$\begin{aligned}
&1^\circ \quad x_\nu = a + (\nu-1)h \quad && \text{with } h = \tfrac{b-a}{n-1},\\
&2^\circ \quad x_\nu = a + \nu h && \text{with } h = \tfrac{b-a}{n+1},\\
&3^\circ \quad x_\nu = a + \left(\nu - \tfrac{1}{2}\right) h && \text{with } h = \tfrac{b-a}{n}.
\end{aligned}$$

In this section, we present the quadratures of the form (3.1) corresponding to these three types of equidistant nodes. For this purpose, we first recall the Stirling numbers of the first kind $s(k,j)$, which are defined as the coefficients in the expansion

$$(t)_k = t(t-1)\cdots(t-k+1) = \sum_{j=0}^{k} s(k,j)t^j,$$

where $(t)_0 = 1$ and $s(0,0) = 1$. In general, the recurrence relation

$$s(k+1,j) = s(k,j-1) - k\,s(k,j), \qquad 1 \le j < k,$$

holds with the initial conditions $s(k,0) = 0$ and $s(1,1) = 1$.

Case 1°. First we have

$$\Phi_k(x) = \prod_{j=1}^{k}(x - x_j) = \prod_{j=1}^{k}(x - a - (j-1)h).$$

Since

$$\Phi_k(a+th) = h^k t(t-1)\cdots(t-k+1) = h^k \sum_{j=0}^{k} s(k,j)t^j,$$

we obtain

$$\Phi_k(x) = \sum_{j=0}^{k} h^{k-j} s(k,j)(x-a)^j.$$

This means that $c_j^{(k)} = h^{k-j}s(k,j)$, for $j = 0,1,\ldots,k$, and x should be replaced by $x-a$ in the corresponding expressions. Also we have

$$\begin{aligned}
&\Phi_k'(x_\nu) = \Phi_k'(a+(\nu-1)h) = h^{k-1}\sum_{j=1}^{k} j\, s(k,j)(\nu-1)^{j-1},\\
&2x_k - x_\nu = a + (2k-\nu-1)h,
\end{aligned} \tag{4.1}$$

and

$$\Phi_k(x)^2 = \sum_{m=0}^{2k} \lambda_m^{(k)}(x-a)^m, \qquad \lambda_m^{(k)} = h^{2k-m}\sum_{j=0}^{m} s(k,j)s(k,m-j). \tag{4.2}$$

ETNA
Kent State University
http://etna.math.kent.edu

Finally, we obtain for s_k and r_k

$$s_k = \int_a^b w(x)\Phi_{k-1}(x)^2 \mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \int_a^b w(x)(x-a)^m \mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \mu_m^{(1)}$$

and

$$r_k = \int_a^b w(x)(x-a+a)\Phi_{k-1}(x)^2 \mathrm{d}x = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(1)} + a\mu_m^{(1)}\right),$$

where $\mu_m^{(1)} = \int_a^b w(x)(x-a)^m \mathrm{d}x$.

COROLLARY 4.1. *The coefficients A_ν and B_ν, for $\nu = 1, \ldots, n$, in the weighted Hermite-Newton quadrature rule of the form (3.1) with equidistant nodes $x_\nu = a + (\nu-1)h$ and $h = (b-a)/(n-1)$ are given by*

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\big(1 + h_{k,\nu}(2k-\nu-1)h\big)\mu_m^{(1)} - h_{k,\nu}\mu_{m+1}^{(1)}\right)$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(1)} - (2k-\nu-1)h\mu_m^{(1)}\right),$$

where $\mu_m^{(1)} = \int_a^b w(x)(x-a)^m \mathrm{d}x$,

$$h_{1,1} = 0, \qquad h_{k,\nu} = \frac{2}{h} \sum_{\substack{i=1 \\ i\neq\nu}}^{k} \frac{1}{\nu - i}, \quad 2 \le \nu \le k \le n, \tag{4.3}$$

and $\Phi_k'(x_\nu)$ and $\lambda_m^{(k)}$ can be directly computed from (4.1) and (4.2), respectively.

Case 2°. First we have

$$\Phi_k(x) = \prod_{j=1}^{k}(x - x_j) = \prod_{j=1}^{k}(x - a - jh).$$

Since

$$\Phi_k(a + (t+1)h) = h^k t(t-1)\cdots(t-k+1) = h^k \sum_{j=0}^{k} s(k,j)t^j,$$

we obtain $c_j^{(k)} = h^{k-j}s(k,j)$, for $j = 0, 1, \ldots, k$, and x should be replaced by $x - a - h$ in the corresponding expressions. The expression for $\Phi_k'(x_\nu)$ is the same as in (4.1) and we have

$$\begin{aligned} &2x_k - x_\nu = a + (2k-\nu)h, \\ &\Phi_k(x)^2 = \sum_{m=0}^{2k} \lambda_m^{(k)}(x-a-h)^m, \qquad \lambda_m^{(k)} = h^{2k-m} \sum_{j=0}^{m} s(k,j)s(k,m-j). \end{aligned} \tag{4.4}$$

ETNA
Kent State University
http://etna.math.kent.edu

In this case, the integrals s_k and r_k are computed as

$$s_k = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \mu_m^{(2)},$$
$$r_k = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \mu_{m+1}^{(2)} + (a+h)s_k = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(2)} + (a+h)\mu_m^{(2)} \right),$$

where $\mu_m^{(2)} = \int_a^b w(x)(x-a-h)^m \mathrm{d}x$.

COROLLARY 4.2. *The coefficients A_ν and B_ν, for $\nu = 1, \ldots, n$, in the weighted Hermite-Newton quadrature rule of the form (3.1) with equidistant nodes $x_\nu = a + \nu h$ and $h = (b-a)/(n+1)$ are given by*

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\big(1 + h_{k,\nu}(2k-\nu-1)h\big)\mu_m^{(2)} - h_{k,\nu}\mu_{m+1}^{(2)} \right)$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(2)} - (2k-\nu-1)h\mu_m^{(2)} \right),$$

where $\mu_m^{(2)} = \int_a^b w(x)(x-a-h)^m \mathrm{d}x$ and $h_{k,\nu}$, $\Phi_k'(x_\nu)$, and $\lambda_m^{(k)}$ are given by (4.3), (4.1), and (4.4), respectively.

Case 3°. Since $x_\nu = a + \left(\nu - \frac{1}{2}\right) h$, $h = \frac{b-a}{n}$, and $c_j^{(k)} = h^{k-j} s(k,j)$, for $j = 0, 1, \ldots, k$, we have

$$\Phi_k(x)^2 = \sum_{m=0}^{2k} \lambda_m^{(k)} \left(x - a - \frac{h}{2} \right)^m, \qquad \lambda_m^{(k)} = h^{2k-m} \sum_{j=0}^{m} s(k,j)s(k,m-j) \tag{4.5}$$

as well as

$$s_k = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \mu_m^{(3)}, \qquad r_k = \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(3)} + \left(a + \frac{h}{2} \right) \mu_m^{(3)} \right),$$

where $\mu_m^{(3)} = \int_a^b w(x)(x-a-h/2)^m \mathrm{d}x$. In addition, $2x_k - x_\nu = a + (2k-\nu-1/2)h$, and $\Phi_k'(x_\nu)$ is the same as in (4.1).

COROLLARY 4.3. *The coefficients A_ν and B_ν, for $\nu = 1, \ldots, n$, in the weighted Hermite-Newton quadrature rule of the form (3.1) with equidistant nodes $x_\nu = a + \left(\nu - \frac{1}{2}\right) h$ and $h = (b-a)/n$ are given by*

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\big(1 + h_{k,\nu}(2k-\nu-1)h\big)\mu_m^{(3)} - h_{k,\nu}\mu_{m+1}^{(3)} \right)$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1}^{(3)} - (2k-\nu-1)h\mu_m^{(3)} \right),$$

where $\mu_m^{(3)} = \int_a^b w(x)(x-a-h/2)^m \mathrm{d}x$ and $h_{k,\nu}$, $\Phi_k'(x_\nu)$, and $\lambda_m^{(k)}$ are given by (4.3), (4.1), and (4.5), respectively.

As an illustration, let us present the program code in MATHEMATICA for the nodes and weights of Corollary 4.1:

ETNA
Kent State University
http://etna.math.kent.edu


```
Nn1[n_,a_,b_,w_]:=Module[{h=(b-a)/(n-1),mu,x,k,m,nu,AF,La,
    hk,nodes,vekA,vekB,Aw,Bw},
 mu=Table[Integrate[(x-a)^m w[x],{x,a,b}],{m,0,2n-1}];
 AF=Table[h^(k-1)If[k==1,1,Sum[j StirlingS1[k,j]
    If[nu==1&&j==1,1,(nu-1)^(j-1)],{j,1,k}]],
    {nu,1,n},{k,nu,n}]//Simplify;
 La[m_,k_]:=h^(2k-m)Sum[StirlingS1[k,j]
    StirlingS1[k,m-j],{j,0,m}]//Simplify;
 hk[nu_,k_]:=2/h If[k==1,0,Sum[If[i==nu,0,1/(nu-i)],{i,1,k}]];
 vekA=Table[Sum[La[m,k-1]((1+hk[nu,k](2k-nu-1)h)mu[[m+1]]
    -hk[nu,k]mu[[m+2]]),{m,0,2k-2}],{nu,1,n},{k,nu,n}]
    //Simplify;
 vekB=Table[Sum[La[m,k-1](mu[[m+2]]-(2k-nu-1)h mu[[m+1]]),
    {m,0,2k-2}],{nu,1,n},{k,nu,n}]//Simplify;
 nodes=Table[a+(k-1)h,{k,1,n}];
 Aw=Table[(1/AF[[nu]]^2).vekA[[nu]],{nu,1,n}]//Simplify;
 Bw=Table[(1/AF[[nu]]^2).vekB[[nu]],{nu,1,n}]//Simplify;
 Return[{nodes,Aw,Bw}];]
```

Consider the following weight functions

```
 w1[x_]:= 1;
 w2[x_]:= x^2;
 w3[x_]:= Abs[x];
 w4[x_]:= Exp[x];
 w5[x_]:= Cos[Pi x/2];
 w6[x_]:= x^(-1/2)Log[1/x];
 w7[x_]:= Cos[100 Pi x];
```

where the last one is not a standard (nonnegative) weight function.

Note that in the case when a symbolic integration of the moments $\mu_m^{(1)}$ is not possible, a numerical calculation must be included in the above subprogram. Now, using the given code, we can obtain the following results for some selected number of nodes, intervals, and weights:

```
In[1]:= {nodes, Aw, Bw} = Nn1[6, -1, 1, w1]
```

$$\text{Out[1]:=}\ \left\{\left\{-1,-\frac{1}{5},\frac{1}{5},\frac{3}{5},1\right\},\left\{\frac{63\,817}{456\,192},\frac{5125}{16\,896},\frac{15\,875}{28\,512},\frac{5125}{16\,896},\frac{63\,817}{456\,192}\right\},\right.$$
$$\left.\left\{\frac{493}{88\,704},-\frac{12\,575}{266\,112},-\frac{2725}{66\,528},\frac{2725}{66\,528},\frac{12575}{266\,112},-\frac{493}{88\,704}\right\}\right\}$$

```
In[2]:= {nodes, Aw, Bw} = Nn1[5, -1, 1, w3]
```

$$\text{Out[2]:=}\ \left\{\left\{-1,-\frac{1}{2},0,\frac{1}{2},1\right\},\ \left\{\frac{22}{135},\frac{32}{135},\frac{1}{5},\frac{32}{135},\frac{22}{135}\right\},\right.$$
$$\left.\left\{\frac{1}{120},\ -\frac{2}{45},0,\frac{2}{45},-\frac{1}{120}\right\}\right\}$$

ETNA
Kent State University
http://etna.math.kent.edu


```
In[3]:= {nodes, Aw, Bw} = Nn1[5, 0, 1, w6]
```

$$\text{Out[3]:=}\ \left\{\left\{0,\frac{1}{4},\frac{1}{2},\frac{3}{4},1\right\},\left\{\frac{221\,236\,741\,818\,208}{87\,932\,340\,851\,355},\frac{383\,158\,808\,363\,008}{816\,514\,593\,619\,725},\right.\right.$$
$$\left.\frac{343\,823\,093\,248}{586\,396\,035\,225},\frac{447\,163\,627\,421\,696}{1\,143\,120\,431\,067\,615},\frac{212\,846\,917\,630\,988}{5\,715\,602\,155\,338\,075}\right\}$$
$$\left\{\frac{11\,420\,487\,293\,024}{381\,040\,143\,689\,205},-\frac{56\,651\,333\,531\,648}{381\,040\,143\,689\,205},-\frac{29\,241\,181\,511\,936}{211\,688\,968\,716\,225},\right.$$
$$\left.\left.-\frac{4\,242\,817\,583\,104}{112\,070\,630\,496\,825},-\frac{3\,744\,788\,102\,368}{1\,905\,200\,718\,446\,025}\right\}\right\}$$

EXAMPLE 4.4. For the numerical evaluation of the integral

$$I(f)=\int_0^1 \frac{1}{\sqrt{x}}\log\frac{1}{x}\sin\frac{\pi x}{2}\mathrm{d}x = 0.64795292437351204146489364 5\ldots,$$

if we apply the quadrature rule (3.1) with respect to the weight function $w(x)=w_6(x)=x^{-1/2}\log(1/x)$ on $(0,1)$, then the obtained quadrature sums

$$Q_n(f)=\sum_{\nu=1}^{n}\left(A_\nu f(x_\nu)+B_\nu f'(x_\nu)\right), \qquad n=2,4,6,8,10,$$

as well as the corresponding relative errors

$$\mathrm{err}_n(f)=\left|\frac{Q_n(f)-I(f)}{I(f)}\right|, \qquad n=2,4,6,8,10,$$

for $f(x)=\sin(\pi x/2)$ are displayed in the Tables 4.1 and 4.2, respectively. Note that the digits in the error that are underlined and the numbers in parenthesis indicate the decimal exponents. As we can observe, the convergence of the quadrature sums is very fast.

TABLE 4.1
Quadrature sums $Q_n(f)$ for $f(x)=\sin(\pi x/2)$ and the weight function w_6.

n	$Q_n(f)$
2	0.64043984167600953459352044293
4	0.64795281970675867596236728562
6	0.64795292437311740461895795031
8	0.64795292437351204091373653691
10	0.64795292437351204146489330692

TABLE 4.2
Relative errors $\mathrm{err}_n(f)$ in the quadrature sums $Q_n(f)$ for $f(x)=\sin(\pi x/2)$ and the weight function w_6.

n	2	4	6	8	10
$\mathrm{err}_n(f)$	$1.16(-2)$	$1.62(-7)$	$6.09(-13)$	$8.51(-19)$	$5.22(-25)$

EXAMPLE 4.5. Now consider integrals with respect to the oscillatory weight function $w(x)=w_7(x)=\cos(100\pi x)$ on $[-1,1]$. Let $f(x)=\mathrm{e}^{x-2x^2}$. The graph of the corresponding integrand is displayed in Figure 4.1, and the exact value is equal to

$$\int_{-1}^{1} w(x)f(x)\mathrm{d}x=-\frac{\sqrt{2\pi}}{4}\mathrm{e}^{\frac{1}{8}-1250\pi^2}\Re\left(\mathrm{erf}\left(\frac{5(1+20i\pi)}{2\sqrt{2}}\right)+\mathrm{erf}\left(\frac{3+100i\pi}{2\sqrt{2}}\right)\right),$$
$$=-0.00001370444406834843908676486 5782\ldots,$$

ETNA
Kent State University
http://etna.math.kent.edu

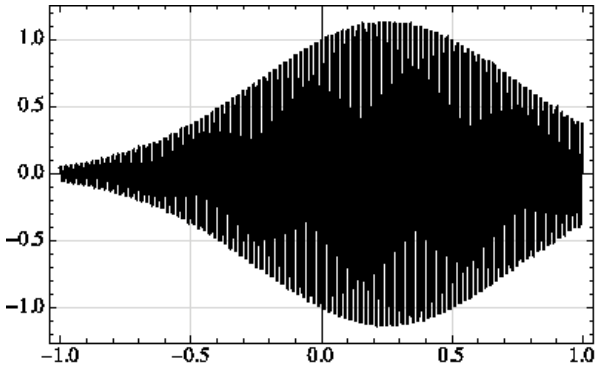

FIG. 4.1. *Integrand* $x \mapsto \mathrm{e}^{x-2x^2}\cos(100\pi x)$ *on* $[-1,1]$.

TABLE 4.3
Relative errors $\mathrm{err}_n(f)$ *and* $\overline{\mathrm{err}}_{2n}(f)$ *in the quadrature sums* $Q_n(f)$ *and* $\overline{Q}_{2n}(f)$*, respectively, for the integrand* $f(x)=\mathrm{e}^{x-2x^2}$ *and the oscillatory weight function* $w_7(x)=\cos(100\pi x)$.

n	4	6	8	10	12	14
$\mathrm{err}_n(f)$	9.57(−5)	2.45(−6)	8.61(−7)	3.53(−8)	6.67(−10)	7.45(−12)
$\overline{\mathrm{err}}_{2n}(f)$	8.73(−3)	8.42(−4)	3.39(−5)	5.70(−7)	5.68(−9)	3.79(−11)

where $\mathrm{erf}(z)$ is the error function defined by

$$\mathrm{erf}(z)=\frac{2}{\sqrt{\pi}}\int_0^z \mathrm{e}^{-t^2}\mathrm{d}t.$$

By applying the quadrature rule (3.1) with respect to the weight function $\cos(100\pi x)$ on $[-1,1]$ for the above integral for $n=4,6,8,10,12,14$, we obtain quadrature sums with relative errors given in Table 4.3. In order to compare these results with the ones obtained by the weighted Newton-Cotes formula at $2n$-points (see [13, Theorem 2.1]),

$$\int_{-1}^{1} w(x)f(x)\mathrm{d}x \approx \overline{Q}_{2n}(f)=\sum_{\nu=1}^{2n} W_k f(x_k), \tag{4.6}$$

we also consider the corresponding relative errors

$$\overline{\mathrm{err}}_{2n}(f)=\left|\frac{\overline{Q}_{2n}(f)-I(f)}{I(f)}\right|,$$

which are also given in Table 4.3. Note that the complexity of the quadrature formulas (3.1) and (4.6) is the same, but the quadrature sums $Q_n(f)$ show slightly faster convergence.

5. Weighted Hermite quadrature rules with geometric nodes. In this case, we assume that the nodes have a geometric distribution as $\{x_\nu\}_{\nu=1}^n = \{aq^{\nu-1}\}_{\nu=1}^n$, where $q=\sqrt[n-1]{b/a}$. By using the q-binomial theorem, we have recently obtained explicit expressions of the coefficients $\{A_\nu\}_{\nu=1}^n$ in the quadrature formula [13]

$$\int_a^b f(x)w(x)\mathrm{d}x=\sum_{\nu=1}^{n} A_\nu f(x_\nu)+R_n(f).$$

ETNA
Kent State University
http://etna.math.kent.edu

We now present the quadratures of the form (3.1) for the geometric nodes. First, according to Cauchy's q-binomial theorem [10], we have

$$(x;q)_k = \prod_{i=1}^{k}(1-xq^{i-1}) = \sum_{j=0}^{k}(-1)^j \begin{bmatrix} k \\ j \end{bmatrix}_q q^{\binom{j}{2}} x^j, \tag{5.1}$$

in which

$$\begin{bmatrix} k \\ j \end{bmatrix}_q = \frac{(q;q)_k}{(q;q)_j (q;q)_{k-j}}, \qquad \binom{j}{2} = \frac{j(j-1)}{2}, \qquad (q;q)_k = \prod_{i=1}^{k}(1-q^i).$$

If we set in (5.1) $x = t/a$ and $q \to 1/q$, then

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

which yields

$$\prod_{i=1}^{k}\left(t-aq^{i-1}\right) = (-a)^k q^{\frac{k(k-1)}{2}} \sum_{j=0}^{k}(-1)^j \begin{bmatrix} k \\ j \end{bmatrix}_{1/q} q^{-\frac{j(j-1)}{2}} a^{-j} t^j. \tag{5.2}$$

On the other hand, since

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

the polynomials (5.2) finally change to

$$\Phi_k(x) = \prod_{i=1}^{k}\left(x-aq^{i-1}\right) = \sum_{j=0}^{k}(-a)^{k-j} \begin{bmatrix} k \\ j \end{bmatrix}_q q^{\binom{k-j}{2}} x^j, \qquad k = 1,\dots,n, \tag{5.3}$$

where $\Phi_0(x) = 1$. Therefore, we have

$$c_j^{(k)} = (-a)^{k-j} \begin{bmatrix} k \\ j \end{bmatrix}_q q^{\binom{k-j}{2}}. \tag{5.4}$$

By using (5.3), the following items can be computed straightforwardly:

$$\begin{aligned} &\Phi_k'(x_\nu) = \Phi_k'(aq^{\nu-1}) = a^{k-1} q^{\frac{k^2-k-2\nu+1}{2}} \sum_{j=1}^{k}(-1)^{k-j} j \begin{bmatrix} k \\ j \end{bmatrix}_q q^{\frac{j^2+(2\nu-1-2k)j}{2}}, \\ &2x_k - x_\nu = a\left(2q^{k-1} - q^{\nu-1}\right), \end{aligned} \tag{5.5}$$

as well as $\Phi_k(x)^2 = \sum_{m=0}^{2k} \lambda_m^{(k)} x^m$, where, according to (3.5) and (5.4),

$$\begin{aligned} \lambda_m^{(k)} &= \sum_{j=0}^{m}(-a)^{k-j} \begin{bmatrix} k \\ j \end{bmatrix}_q q^{\binom{k-j}{2}} (-a)^{k-m+j} \begin{bmatrix} k \\ m-j \end{bmatrix}_q q^{\binom{k-m+j}{2}} \\ &= (-a)^{2k-m} q^{k^2-k(m+1)+\frac{1}{2}m(m+1)} \sum_{j=0}^{m} \begin{bmatrix} k \\ j \end{bmatrix}_q \begin{bmatrix} k \\ m-j \end{bmatrix}_q q^{j(j-m)}. \end{aligned} \tag{5.6}$$

ETNA
Kent State University
http://etna.math.kent.edu

The integrals r_k and s_k are given in this case by (3.8) and (3.9), where

$$h_{1,1} = 0 \quad \text{and} \quad h_{k,\nu} = \frac{\Phi_k''(x_\nu)}{\Phi_k'(x_\nu)} = \frac{2q}{a} \sum_{\substack{i=1 \\ i\neq\nu}}^{k} \frac{1}{q^\nu - q^i}, \qquad 2 \le \nu \le k \le n.$$

COROLLARY 5.1. *The coefficients A_ν and B_ν, for $\nu = 1, \dots, n$, in the weighted Hermite quadrature formula*

$$\int_a^b w(x) f(x) \mathrm{d}x = \sum_{\nu=1}^{n} \left(A_\nu f(aq^{\nu-1})) + B_\nu f'(aq^{\nu-1}) \right) + R_n(f)$$

are given by

$$A_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(-\frac{2q}{a} \tau_{k,\nu} \mu_{m+1} + \left(1 + 2(2q^k - q^\nu)\tau_{k,\nu}\right)\mu_m \right)$$

and

$$B_\nu = \sum_{k=\nu}^{n} \frac{1}{(\Phi_k'(x_\nu))^2} \sum_{m=0}^{2k-2} \lambda_m^{(k-1)} \left(\mu_{m+1} - \frac{a}{q}(2q^k - q^\nu)\mu_m \right),$$

where $\mu_m = \int_a^b w(x) x^m \mathrm{d}x$,

$$\tau_{1,1} = 0, \qquad \tau_{k,\nu} = \sum_{\substack{i=1 \\ i\neq\nu}}^{k} \frac{1}{q^\nu - q^i}, \quad 2 \le \nu \le k \le n,$$

and $\lambda_m^{(k-1)}$ and $\Phi_k'(x_\nu)$ are computed by (5.6) and (5.5), respectively.

6. Standard Gauss-Christoffel quadrature rules. Let us consider the general form of the quadrature formula (3.1) with the coefficients A_ν and B_ν as

$$A_\nu = \int_a^b w(x) H_{\nu,0}(x) \mathrm{d}x \qquad \text{and} \qquad B_\nu = \int_a^b w(x) H_{\nu,1}(x) \mathrm{d}x,$$

in which the polynomials $H_{\nu,0}(x)$ and $H_{\nu,1}(x)$ are defined by

$$H_{\nu,0}(x) = \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 (1 - (x - 2x_k + x_\nu) h_{k,\nu})$$

and

$$H_{\nu,1}(x) = \sum_{k=\nu}^{n} \left(\frac{\Phi_{k-1}(x)}{\Phi_k'(x_\nu)} \right)^2 (x - 2x_k + x_\nu),$$

where $h_{k,\nu}$ is given by (2.8). According to (1.4), for these polynomials the following alternative expressions

$$H_{\nu,0}(x) = \frac{1 - h_{n,\nu}(x - x_\nu)}{\Phi_n'(x_\nu)^2} \left(\frac{\Phi_n(x)}{x - x_\nu} \right)^2 \qquad \text{and} \qquad H_{\nu,1}(x) = \frac{\Phi_n(x)}{\Phi_n'(x_\nu)^2} \cdot \frac{\Phi_n(x)}{x - x_\nu}$$

hold, and they can be represented in terms of the polynomials $\Phi_k(x)$, for $k = 0, 1, \dots$, as

$$H_{\nu,0}(x) = \frac{1 - h_{n,\nu}(x - x_\nu)}{\Phi_n'(x_\nu)^2} \left(\sum_{k=\nu}^{n} \gamma_{n,k}^{(\nu)} \Phi_{k-1}(x) \right)^2, \qquad \nu = 1, \dots, n,$$

ETNA
Kent State University
http://etna.math.kent.edu

and

$$H_{\nu,1}(x) = \frac{\Phi_n(x)}{\Phi_n'(x_\nu)^2} \sum_{k=\nu}^{n} \gamma_{n,k}^{(\nu)} \Phi_{k-1}(x), \qquad \nu = 1, \dots, n, \tag{6.1}$$

where

$$\gamma_{n,k}^{(\nu)} = \prod_{j=k+1}^{n} (x_\nu - x_j), \qquad \text{for} \quad \nu \le k \le n-1, \qquad \text{and} \quad \gamma_{n,n}^{(\nu)} = 1.$$

Let $\{\pi_k\}_{k=0}^{+\infty}$ be a system of monic polynomials orthogonal with respect to the inner product defined by

$$(f, g) = \int_a^b w(x) f(x) g(x) \mathrm{d}x, \qquad f, g \in L_w^2(a, b).$$

THEOREM 6.1. *The quadrature formula* (3.1) *reduces to the Gauss-Christoffel quadrature rule with respect to the weight function* $w : (a, b) \to \mathbb{R}^+$ *as*

$$\int_a^b w(x) f(x) \mathrm{d}x = \sum_{\nu=1}^{n} A_\nu f(x_\nu) + R_n(f),$$

if and only if the nodes $\{x_\nu\}_{\nu=1}^{n}$ *are such that*

$$(\Phi_n, \Phi_k) = \int_a^b w(x) \Phi_n(x) \Phi_{k-1}(x) \mathrm{d}x = 0, \qquad k = 1, 2, \dots, n. \tag{6.2}$$

Proof. Suppose that the orthogonality relations (6.2) hold. Since the span of the polynomials $\Phi_0(x), \Phi_1(x), \dots, \Phi_{n-1}(x)$ constitutes the space of all polynomials of degree at most $n-1$ (the set $\mathcal{P}_{n-1}$), this means that $\Phi_n(x)$ is orthogonal to $\mathcal{P}_{n-1}$ and therefore $\Phi_n(x) = \pi_n(x)$. Then, due to (6.1), the coefficients B_ν vanish, i.e.,

$$B_\nu = \int_a^b w(x) H_{\nu,1}(x) \mathrm{d}x = \frac{1}{\Phi_n'(x_\nu)^2} \sum_{k=\nu}^{n} \gamma_{n,k}^{(\nu)} (\Phi_n, \Phi_{k-1}) = 0, \qquad k = 1, 2, \dots, n,$$

and the coefficients A_ν reduce to Christoffel numbers (cf. [6, §1.4.2] and [15, p. 115]), i.e.,

$$\begin{aligned} A_\nu = \int_a^b w(x) H_{\nu,0}(x) \mathrm{d}x &= \int_a^b w(x) \left(1 - h_{n,\nu}(x - x_\nu)\right) \left(\frac{\pi_n(x)}{\pi_n'(x_\nu)(x - x_\nu)} \right)^2 \mathrm{d}x \\ &= \int_a^b w(x) \frac{\pi_n(x)}{\pi_n'(x_\nu)(x - x_\nu)} \mathrm{d}x = \lambda_{n,\nu}. \end{aligned}$$

Conversely, from the conditions $B_\nu = 0$, for $\nu = 1, \dots, n$, the orthogonality relations (6.2) follow directly. ∎

7. Gaussian quadrature rules using only function derivatives. In this section, we consider a class of quadrature formulas (3.1) for which the weight coefficients satisfy $A_\nu = 0$, for $\nu = 1, \dots, n$, i.e., when

$$\int_a^b w(x) f(x) \mathrm{d}x = \sum_{\nu=1}^{n} B_\nu f'(x_\nu) + R_n(f). \tag{7.1}$$

ETNA
Kent State University
http://etna.math.kent.edu

Note that in (7.1) a restriction on the functions f is, however, needed. A general class of quadratures with m-th derivative instead of the first one have been recently considered in [12, 16, 24]. Such quadratures need to be restricted to a class of functions f with a zero of order m at some point $x = \lambda\,(\in \mathbb{R})$, i.e., functions of the form $f(x) = (x-\lambda)^m g(x)$. Such a quadrature formula

$$\int_a^b w(x)f(x)\mathrm{d}x = \sum_{\nu=1}^{n} B_\nu f^{(m)}(x_\nu) + R_{n,m}(f),$$

is exact for all polynomials of degree at most $2n+m-1$ and is indeed equivalent to the Gaussian formula

$$\int_a^b w(x)(x-\lambda)^m g(x)\mathrm{d}x = \sum_{\nu=1}^{n} B_\nu \left(\frac{g(x)}{(x-\lambda)^m}\right)\Bigg|_{x=x_\nu}, \qquad g \in \mathcal{P}_{2n-1}.$$

Here, we are only interested in the case $m=1$. In order to satisfy the condition $f(\lambda)=0$, instead of (7.1), we can consider the following Gauss-Christoffel quadrature formula

$$\int_a^b w(x)(f(x)-f(\lambda))\mathrm{d}x = \sum_{\nu=1}^{n} B_\nu f'(x_\nu) + R_{n,1}(f). \tag{7.2}$$

The main idea in (7.2) is to transform the integral on the left-hand side to an integral of $f'(t)$ (or in general of $f^{(m)}(t)$) with respect to another weight function $t \mapsto \rho(t)$ supported on some interval on the real line and then to construct a Gaussian quadrature with respect to this weight function, i.e., we have

$$\int_a^b w(x)(f(x)-f(\lambda))\mathrm{d}x = \int_{\mathbb{R}} f'(t)\rho(t)\mathrm{d}t = \sum_{\nu=1}^{n} B_\nu f'(x_\nu) + R_{n,1}(f). \tag{7.3}$$

Following [16] (Lemma 2.2 and Theorem 3.1) we here distinguish three cases.

Case 1°: For $\lambda \le a$, the function $\rho(t)$ is supported and positive on (λ, b) and can be represented as

$$\rho(t) = \int_{\max\{a,t\}}^{b} w(x)\mathrm{d}x.$$

In this case, the quadrature formula (7.3) exists uniquely with all weights B_ν positive and all nodes x_ν belonging to the interval (λ, b).

Case 2°: For $\lambda \ge b$, the function $\rho(t)$ is supported and negative on (a, λ) and can be expressed as

$$\rho(t) = -\int_a^{\min\{b,t\}} w(x)\mathrm{d}x.$$

In this case, the quadrature formula (7.3) exists uniquely with all weights B_ν negative and all nodes x_ν belonging to the interval (a, λ).

Case 3°: For $a < \lambda < b$, the function $\rho(t)$ is supported on (a, b) and is negative on (a, λ) and positive on (λ, b) such that we have

$$\rho(t) = \begin{cases} -\int_a^t w(x)\mathrm{d}x, & t < \lambda, \\ \int_t^b w(x)\mathrm{d}x, & t > \lambda. \end{cases}$$

ETNA
Kent State University
http://etna.math.kent.edu

In the first two cases, 1° and 2°, the (monic) orthogonal polynomials π_k for $k = 0, 1, \ldots$, with respect to the positive (negative) weight function $t \mapsto \rho(t)$ on (λ, b) (and/or (a, λ)) exist uniquely and satisfy the three-term recurrence relation

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t) \qquad k = 0, 1, \ldots, \tag{7.4}$$

with $\pi_0(t) = 1$ and $\pi_{-1}(t) = 0$. The coefficient β_0 can be arbitrary, but it is convenient to put $\beta_0 = \mu_0 = \int_{\mathbb{R}} \rho(t)\mathrm{d}t$. Knowing the first N recursion coefficients in (7.4), we can obtain the nodes x_ν and the weights B_ν in (7.3) for any number of nodes $n \leq N$. In other words, the nodes x_ν are eigenvalues of the following symmetric tridiagonal Jacobi matrix (cf. [15, pp. 325–328])

$$J_n = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & \mathbf{O} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ \mathbf{O} & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}, \tag{7.5}$$

and the weight coefficients B_ν are given by $B_\nu = \beta_0 v_{\nu,1}^2$, $\nu = 1, \ldots, n$, where $v_{\nu,1}$ is the first component of the eigenvector $\mathbf{v}_\nu = [v_{\nu,1} \ \ldots \ v_{\nu,n}]^{\mathrm{T}}$ corresponding to the eigenvalue x_ν and normalized such that $\mathbf{v}_\nu^{\mathrm{T}} \mathbf{v}_\nu = 1$. The most popular method for solving this eigenvalue problem is the Golub-Welsch procedure obtained by a simplification of the QR algorithm [7]. Unfortunately, the coefficients in (7.5) are known explicitly only for some narrow classes of weight functions (e.g., for weights of the classical orthogonal polynomials). In other cases, the so-called *strongly non-classical cases*, the recursion coefficients must be constructed numerically (cf. [5, 6], [15, pp. 159–166]), though the main problem is a numerical instability of such algorithms. However, because of progress in symbolic computations and variable-precision arithmetic, the recursion coefficients can be directly generated by using the original Chebyshev method of moments (cf. [15, pp. 159–166]), and instability problems can be eliminated. Respectively, symbolic/variable-precision software for orthogonal polynomials and Gaussian (and similar) quadratures is available. In this regard, the MATHEMATICA package `OrthogonalPolynomials` (see [1] and [17]) is downloadable from the web site `http://www.mi.sanu.ac.rs/~gvm/`. Also, there is Gautschi's software in MATLAB (the packages `OPQ` and `SOPQ`).

In the last case 3°, according to [16, Lemma 2.2], we cannot claim the existence of orthogonal polynomials with respect to the variable-signed weight function $\rho(t)$ when $\lambda \in (a, b)$; see also [22]. However, if such (formal) orthogonal polynomials exist, the positions of their zeros can be characterized. Namely, then there is at most one zero outside the interval (a, b) (see [16, Lemma 2.3]). The above mentioned software can be applied for this case, too.

EXAMPLE 7.1. As an illustration, we here consider the integration of the function $f(x) = (x^2 + x)\cos(2\pi x)$ with respect to the weight function $w(x) = 1/\sqrt{x(1-x)}$ on $(0, 1)$. The exact value of this integral can be expressed in terms of the Bessel function of the first kind as

$$I(f) = \int_0^1 \frac{f(x)}{\sqrt{x(1-x)}}\mathrm{d}x = \frac{1}{4}(J_1(\pi) - 4\pi J_0(\pi)), \tag{7.6}$$

whose numerical value is

$$I(f) = 1.026\,958\,825\,993\,784\,301\,185\,457\,201\,824\,367\,747\,762\,156\,236\,945\,489\,773\ \ldots$$

ETNA
Kent State University
http://etna.math.kent.edu

Now, for different values of λ (see the mentioned cases $1°$–$3°$), we construct the quadrature formula (7.3). First, we compute the coefficients α_k and β_k for $k = 0, 1, \ldots, n-1$ (the lists `al` and `be`) in the three term-recurrence relation (7.4) in a symbolic form by the standard Chebyshev algorithm (cf. [15, pp. 160–162]) and use the list of the first $2n$ moments (`momen`) of the new weight function $t \mapsto \rho(t)$,

$$\mu_k = \int_{\mathbb{R}} t^k \rho(t) \mathrm{d}t, \qquad k = 0, 1, \ldots, 2n-1.$$

The above procedure can be implemented in the MATHEMATICA package `Orthogonal Polynomials` in a very simple way, i.e.,

```
<< orthogonalPolynomials`
  momen=Table[<expression for moments>,{k,0,59}];
  {al,be}=aChebyshevAlgorithm[momen, Algorithm -> Symbolic]
  pq[n_]:=aGaussianNodesWeights[n,al,be,
            WorkingPrecision -> 75,Precision -> 70]
  xB = Table[pq[n],{n,5,40,5}];
```

where we put $n = 30$ and `WorkingPrecision->75` in order to obtain very precise quadrature parameters (nodes and weights) with `Precision->70`. These parameters are calculated for $n = 5(5)30$, so that `xB[[1]][[1]]` and `xB[[1]][[2]]` give lists of nodes and weights for the five-point formula and `xB[[2]][[1]]` and `xB[[2]][[2]]` for the ten-point formula, etc.

The case $\lambda = -1$. Here, $\lambda < a = 0$ and

$$\rho(t) = \int_{\max\{0,t\}}^{1} \frac{1}{\sqrt{x(1-x)}} \mathrm{d}x = 2 \arccos\left(\sqrt{\max(0,t)}\right), \qquad t \in (-1, 1),$$

so that the moments are computed as

$$\mu_k = \int_{-1}^{1} t^k \rho(t) \mathrm{d}t = \frac{\pi}{k+1} \left(\frac{2k+1}{2^{2k+1}(k+1)} \binom{2k}{k} + (-1)^k \right), \qquad k = 0, 1, \ldots$$

The obtained recursive coefficients are

$$\alpha_0 = -\frac{5}{24}, \quad \alpha_1 = -\frac{119}{3432}, \quad \alpha_2 = -\frac{2588947}{500663592}, \quad \alpha_3 = -\frac{18183033185659091}{2601142907689630728},$$
$$\alpha_4 = -\frac{16660602648917486659428389005}{123745462574720546755256000048952}, \quad \text{etc.},$$

and

$$\beta_0 = \frac{3\pi}{2}, \quad \beta_1 = \frac{143}{576}, \quad \beta_2 = \frac{437643}{1635920}, \quad \beta_3 = \frac{106240541891341}{429030325805760},$$
$$\beta_4 = \frac{3948454526733515000396961}{154549110840746194607375 32}, \quad \text{etc.}$$

The obtained quadrature parameters for $n = 5$ are given in Table 7.1 (rounded to 13 digits in order to save space). The numbers in parenthesis indicate the decimal exponents. By applying these quadratures for $n = 5(5)30$ to the integral (7.6), we obtain the relative errors in the quadrature sums $Q_n(f; a) = \sum_{\nu=1}^{n} B_\nu f'(x_\nu)$ as

$$\mathrm{err}_n(f) = \left| \frac{Q_n(f) - I(f)}{I(f)} \right|,$$

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 7.1
Quadrature parameters x_ν and B_ν, $\nu = 1, \ldots, n$, for the five-point formula and $\lambda = -1$ and $\lambda = 0$.

	$\lambda = -1$		$\lambda = 0$	
ν	x_ν	B_ν	x_ν	B_ν
1	$-9.119264901649(-1)$	$6.984430414679(-1)$	$4.051762756031(-2)$	$2.854763299970(-1)$
2	$-5.677211568774(-1)$	1.402798250672	$2.065790376113(-1)$	$4.838056902117(-1)$
3	$-7.435701119674(-2)$	1.551924819771	$4.595835340537(-1)$	$4.512164268183(-1)$
4	$4.471396544544(-1)$	$8.276932280684(-1)$	$7.257355103578(-1)$	$2.706370295031(-1)$
5	$8.503502175341(-1)$	$2.315296404048(-1)$	$9.258643615236(-1)$	$7.966085026479(-2)$

TABLE 7.2
Relative errors $\mathrm{err}_n(f)$ in the quadrature sums $Q_n(f)$ with n nodes and $\lambda = -1$, $\lambda = 0$, and $\lambda = 1/4$.

λ	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$	$n = 30$
-1	$2.43(-1)$	$1.06(-7)$	$1.58(-16)$	$6.92(-27)$	$2.27(-38)$	$965(-51)$
0	$6.61(-4)$	$1.91(-13)$	$2.47(-25)$	$9.91(-39)$	$3.06(-53)$	$1.24(-68)$
$1/4$	$1.90(-3)$	$3.95(-14)$	$1.87(-25)$	$2.06(-38)$	$7.81(-54)$	$1.00(-68)$

which are given in Table 7.2. Note that in this case $f(-1) = 0$. The graph of the weight function $t \mapsto \rho(t)$ is displayed in Figure 7.1 (left).

A faster convergence of the quadrature sums can be achieved if one takes $\lambda = a = 0$. In this case, the support of the weight ρ is $[0, 1]$ and the recursive coefficients are respectively

$$\alpha_0 = \frac{3}{8}, \quad \alpha_1 = \frac{51}{104}, \quad \alpha_2 = \frac{133815}{269672}, \quad \alpha_3 = \frac{9334413039}{18745294856}, \quad \alpha_4 = \frac{161968155364209543}{324766924323367688},$$

$$\alpha_5 = \frac{5735389449861409808945886 2823}{1149095648120951853924928420 88}, \quad \text{etc.},$$

and

$$\beta_0 = \frac{\pi}{2}, \quad \beta_1 = \frac{13}{192}, \quad \beta_2 = \frac{2593}{40560}, \quad \beta_3 = \frac{951542397}{15060973760}, \quad \beta_4 = \frac{116488901494177}{1852006876475868},$$

$$\beta_5 = \frac{72231052115947193902875 25}{11508580083602755206592070 4}, \quad \text{etc.}$$

The corresponding quadrature parameters for $n = 5$ are also given in Table 7.1 and the relative errors in the quadrature sums for $n = 5, 10, 15, 20, 25, 30$ are presented in Table 7.2. Again, note that in this case $f(0) = 0$. The cases when $\lambda \geq b = 1$ are similar to the previous ones.

The case $\lambda = 1/4$. Here $\lambda \in (0, 1)$. According to the case 3°, the variable-signed weight function $\rho(t)$ is given by

$$\rho(t) = \begin{cases} -2 \arcsin\left(\sqrt{t}\right), & t < \lambda, \\ 2 \arccos\left(\sqrt{t}\right), & t > \lambda, \end{cases}$$

whose graph is shown in Figure 7.1 (right). An application of the package `OrthogonalPolynomials` gives the recursive coefficients for these formal orthogonal polynomials as

ETNA
Kent State University
http://etna.math.kent.edu

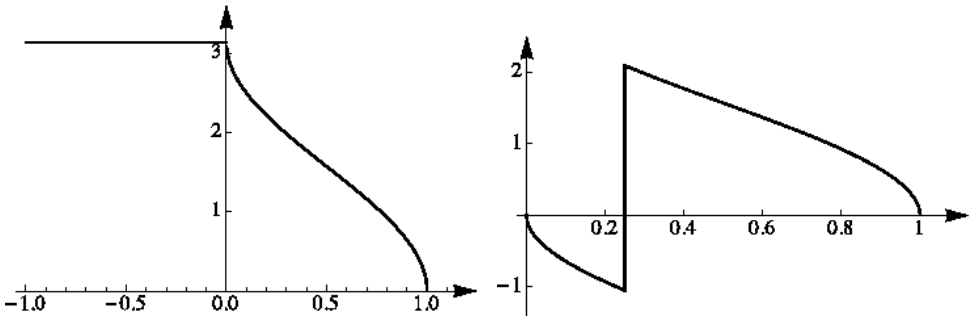

FIG. 7.1. *Graphs of the weight functions $t \mapsto \rho(t)$ for $\lambda = -1$ (left) and $\lambda = 1/4$ (right).*

$$\alpha_0 = \frac{5}{8},\ \ \alpha_1 = \frac{29}{8},\ \ \alpha_2 = -\frac{58397}{20152},\ \ \alpha_3 = \frac{15282294221}{20624181512},\ \ \alpha_4 = \frac{126387190198565645}{63888958325694728},$$

$$\alpha_5 = -\frac{258276715857820481324364942 05}{20768978505647158556943167032},\ \ \text{etc.},$$

and

$$\beta_0 = \frac{\pi}{4},\ \ \beta_1 = \frac{1}{192},\ \ \beta_2 = -\frac{2519}{240},\ \ \beta_3 = -\frac{82897911}{14213608640},\ \ \beta_4 = \frac{19656465118609}{2375528174673948},$$

$$\beta_5 = -\frac{85122912315090571300589 75}{34722595662518304217607 04},\ \ \text{etc.}$$

For example, we have

$$\pi_0(t) = 1,$$

$$\pi_1(t) = t - \frac{5}{8},$$

$$\pi_2(t) = t^2 - \frac{17}{4}t + \frac{217}{96},$$

$$\pi_3(t) = t^3 - \frac{27249}{20152}t^2 + \frac{88767}{201520}t - \frac{6191}{644864},$$

$$\pi_4(t) = t^4 - \frac{4284415}{2046862}t^3 + \frac{166006371}{114624272}t^2 - \frac{82709209}{229248544}t + \frac{53178473}{2619983360},$$

$$\pi_5(t) = t^5 - \frac{254161901255}{62426248888}t^4 + \frac{14109610232965}{2528263079964}t^3 - \frac{86691796853035}{26968139519616}t^2 + \frac{183835282649245}{251702635516416}t - \frac{726230965975195}{18122589757181952},\quad \text{etc.}$$

As we pointed out, at most one zero of $\pi_k(t)$, $k \geq 1$, can be outside the interval $(0, 1)$. For instance, the zeros of $\pi_5(t)$ (rounded to 20 decimal digits) are

$$\Big\{0.07850193478816297828 3,\ 0.41239004873647849422,\ 0.68442277490968688952,\ 0.91117498647961796404,\ 1.98490499339551402 11\Big\},$$

ETNA
Kent State University
http://etna.math.kent.edu

and the last one is outside the interval $(0,1)$. Since, the corresponding "weight" coefficients are

$$\Big\{-0.08647377160684188571,\ 0.45012324213407132278,\ 0.31863469789365847822,$$
$$0.10311387609344687634,\ 1.1888311351799918580\times 10^{-7}\Big\},$$

we see that the contribution of this node in the quadrature sum is approximately negligible. In the last row of Table 7.2, we computed the relative errors of the quadrature sums, which are obtained by this kind of formal orthogonal polynomials where $f(1/4)=0$.

Acknowledgments. The authors are deeply grateful to the anonymous referees for their insightful comments and constructive suggestions for improvements of this paper.

REFERENCES

[1] A. S. CVETKOVIĆ AND G. V. MILOVANOVIĆ, *The Mathematica package "OrthogonalPolynomials"*, Facta Univ. Ser. Math. Inform., 19 (2004), pp. 17–36.
[2] P. J. DAVIS, *Interpolation and Approximation*, Dover, New York, 1975.
[3] B. DELLA VECCHIA AND G. MASTROIANNI, *The Hermite interpolation*, Calcolo, 30 (1993), pp. 371–394.
[4] B. DELLA VECCHIA, G. MASTROIANNI, AND P. VÉRTESI, *Simultaneous approximation by Hermite interpolation of higher order*, J. Comput. Appl. Math., 50 (1994), pp. 233–240.
[5] W. GAUTSCHI, *On generating orthogonal polynomials*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 289–317.
[6] ———, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, New York, 2004.
[7] G. GOLUB AND J.H. WELSCH, *Calculation of Gauss quadrature rules*, Math. Comp., 23 (1969), pp. 221–230.
[8] M. IVAN, *A note on the Hermite interpolation polynomial for rational functions*, Appl. Numer. Math., 57 (2007), pp. 230–233.
[9] ———, *A note on the Hermite interpolation*, Numer. Algorithms, 69 (2015), pp. 517–522.
[10] R. KOEKOEK AND R. F. SWARTTOUW, *The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue*, Report no. 98–17, Faculty of Technical Mathematics and Informatics, Technical Universiteit Delft, Delft, 1998.
[11] G. G. LORENTZ, K. JETTER, AND S. D. RIEMENSCHNEIDER, *Birkhoff Interpolation*, Addison-Wessley, Reading, 1983.
[12] M. MASJED-JAMEI, *A new type of weighted quadrature rules and its relation with orthogonal polynomials*, Appl. Math. Comput., 188 (2007), pp. 154–165.
[13] M. MASJED-JAMEI, G. V. MILOVANOVIĆ, AND M. A. JAFARI, *Explicit forms of weighted quadrature rules with geometric nodes*, Math. Comput. Modelling, 53 (2011), pp. 1133–1139.
[14] ———, *Closed expressions for coefficients in weighted Newton-Cotes quadratures*, Filomat, 27 (2013), pp. 649–658.
[15] G. MASTROIANNI AND G. V. MILOVANOVIĆ, *Interpolation Processes: Basic Theory and Applications*, Springer, Berlin, 2008.
[16] G. V. MILOVANOVIĆ AND A. S. CVETKOVIĆ, *Gaussian quadrature rules using function derivatives*, IMA J. Numer. Anal., 31 (2011), pp 358–377.
[17] ———, *Special classes of orthogonal polynomials and corresponding quadratures of Gaussian type*, Math. Balkanica, 26 (2012), pp. 169–184.
[18] B. SADIQ AND D. VISWANATH, *Barycentric Hermite interpolation*, SIAM J. Sci. Comput., 35 (2013), pp. A1254–A1270.
[19] C. SCHNEIDER AND W. WERNER, *Hermite interpolation: the barycentric approach*, Computing, 46 (1991), pp. 35–51.
[20] Y. G. SHI, *A survey on Hermite interpolation of higher order*, in Paul Erdoš and his Mathematics I (Budapest, 1999), G. Halász, L. Lovász, M. Simonovits, and V. T. Sós, eds., Bolyai Soc. Math. Stud., 11, János Bolyai Math. Soc., Budapest, 2002, pp. 653–686.
[21] ———, *Theory of Birkhoff Interpolation*, Nova Science Publishers, Hauppauge, 2003.
[22] G. W. STRUBLE, *Orthogonal polynomials: variable-signed weight functions*, Numer. Math., 5 (1963), pp. 88–94.
[23] J. SZABADOS AND P. VÉRTESI, *Interpolation of Functions*, World Scientific, Singapore, 1990.
[24] B. D. WELFERT, *On quadrature formulae based on derivative collocation*, Appl. Math. Comput., 204 (2008), pp. 647–657.
[25] S. XIANG AND G. HE, *The fast implementation of higher order Hermite-Fejér interpolation*, SIAM J. Sci. Comput., 37 (2015), pp. A1727–A1751.