

Orthogonal polynomials and generalized Gauss-Rys quadrature formulae

Gradimir V. Milovanović^{1,2,*}, Nevena Vasović³

¹*Serbian Academy of Sciences and Arts, 11000 Beograd, Serbia*

²*University of Niš, Faculty of Sciences and Mathematics, 18000 Niš, Serbia*

³*University of Kragujevac, Faculty of Hotel Management and Tourism,
36210 Vrnjačka Banja, Serbia*

*Corresponding author: gvm@mi.sanu.ac.rs

Abstract

Orthogonal polynomials and the corresponding quadrature formulas of Gaussian type concerning the even weight function $\omega^\lambda(t; x) = \exp(-xt^2)(1 - t^2)^{\lambda-1/2}$ on $(-1, 1)$, with parameters $\lambda > -1/2$ and $x > 0$, are considered. For $\lambda = 1/2$ these quadrature rules reduce to the so-called Gauss-Rys quadrature formulas, which were investigated earlier by several authors, e.g., Dupuis *et al* 1976 and 1983; Sagar 1992; Schwenke 2014; Shizgal 2015; King 2016; Milovanović 2018, etc. In this generalized case, the method of modified moments is used, as well as a transformation of quadratures on $(-1, 1)$ with N nodes to ones on $(0, 1)$ with only $(N + 1)/2$ nodes. Such an approach provides a stable and very efficient numerical construction.

Keywords: Nodes; orthogonal polynomials; quadrature rule; recurrence relation; weights.

1. Introduction and preliminaries

In this paper, we consider a problem of construction of the quadrature formulas of Gaussian type on $[-1, 1]$, concerning the following two-parametric weight function

$$\omega^\lambda(t; x) = e^{-xt^2}(1 - t^2)^{\lambda-1/2}, \quad (1)$$

where x is a positive parameter and $\lambda > -1/2$, i.e.,

$$\int_{-1}^1 f(t)\omega^\lambda(t; x) dt = \sum_{\nu=1}^N A_\nu f(\tau_\nu) + R_N(f), \quad (2)$$

which are exact for all polynomials of degree at most $2N - 1$ (in notation \mathcal{P}_{2N-1}), i.e., with remainder term $R_N(f) \equiv R_N^\lambda(f; x) = 0$ for each $f \in \mathcal{P}_{2N-1}$. For an even number of nodes, i.e., when $N = 2n$, the quadrature sum in (2) can be written in the form

$$Q_{2n}(f; x) = \sum_{k=1}^n A_k [f(\tau_k) + f(-\tau_k)], \quad (3)$$

where $\tau_k = \tau_k^{(N)}(x; \lambda)$, $A_k = A_k^{(N)}(x; \lambda) > 0$, and $0 < \tau_1 < \dots < \tau_n < 1$.

The main tool for constructing weighted Gaussian quadrature rules are orthogonal polynomials with respect to the same weight function. In our case, let $\pi_k(t) \equiv \pi_k^\lambda(t; x)$ be polynomials orthogonal on $[-1, 1]$, with respect to the weight function given by (1). Since this weight function is even, the monic orthogonal polynomials $t \mapsto \pi_k^\lambda(t; x)$ satisfy the three-term recurrence relation for $k \geq 0$,

$$\pi_{k+1}^\lambda(t; x) = (t - \alpha_k)\pi_k^\lambda(t; x) - \beta_k\pi_{k-1}^\lambda(t; x), \quad (4)$$

with $\pi_0^\lambda(t; x) = 1$ and $\pi_{-1}^\lambda(t; x) = 0$. The recursion coefficients β_k are positive and depend on the parameters x and λ . Thus, $\beta_k = \beta_k^\lambda(x) > 0$, $k = 1, 2, \dots$, where $x > 0$ and $\lambda > -1/2$. Here, $\alpha_k = \alpha_k^\lambda(x) = 0$, because the weight function is even on $[-1, 1]$. The coefficient $\beta_0^\lambda(x)$ in (4) may be arbitrary, but it is conveniently defined as the first moment of the weight function $t \mapsto \omega^\lambda(t; x)$,

$$\beta_0^\lambda(x) = \int_{-1}^1 \omega^\lambda(t; x) dt = \sqrt{\pi} \frac{\Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)} {}_1F_1\left(\frac{1}{2}, \lambda + 1, -x\right), \quad (5)$$

where ${}_1F_1(a; b; z)$ is the Kummer confluent hypergeometric function; see the Kummer probability distribution (Ostrowska & Turan 2017).

Otherwise, in this paper, we use the generalized hypergeometric function ${}_pF_q$, defined by

$${}_pF_q(a_1, \dots, a_p; b_1, \dots, b_q; z) = \sum_{\nu=0}^{\infty} \frac{(a_1)_\nu \cdots (a_p)_\nu}{(b_1)_\nu \cdots (b_q)_\nu} \frac{z^\nu}{\nu!}$$

for different p and q , where the Pochhammer symbol $(\lambda)_\nu$ is given by

$$(\lambda)_\nu = \lambda(\lambda + 1) \cdots (\lambda + \nu - 1) = \frac{\Gamma(\lambda + \nu)}{\Gamma(\lambda)},$$

and $\Gamma(\lambda)$ is Euler's gamma function

$$\Gamma(\lambda) = \int_0^\infty t^{\lambda-1} e^{-t} dt \quad \text{for } \text{Re}(\lambda) > 0.$$

In Wolfram's MATHEMATICA the function ${}_pF_q$ is implemented as `HypergeometricPFQ` and suitable for both symbolic and numerical calculation. For $p = q + 1$, it has a branch cut discontinuity in the complex z plane running from 1 to ∞ . When $p \leq q$ this series converges for each $z \in \mathbb{C}$. For some recent results on this subject, especially on transformations, summations and some applications see (Milovanović *et al.* 2018, Milovanović & Rathie 2019).

The nodes $\pm\tau_k$ $k = 1, \dots, n$, in the quadrature formula (3) are zeros of the orthogonal polynomial $\pi_N^\lambda(t; x)$ of degree $N = 2n$, and the weight coefficients A_k , $k = 1, \dots, N$, are the corresponding Christoffel numbers (Gautschi 2004, Mastroianni & Milovanović 2008)).

This kind of quadrature formulas are very close to the so-called Rys quadrature formulas introduced in 1976 by Dupuis, Rys and King (Dupuis *et al.* 1976) as an attractive method in computational quantum chemistry, for evaluating two-electron repulsion integrals, which appear in molecular quantum mechanical calculations involving Gaussian Cartesian basis functions. It was shown that such integrals can be reduced to the one-dimensional integrals of the form $\int_0^1 \phi_m(t) \exp(-xt^2) dt$, where $\phi_m(t)$ are even algebraic polynomials of very high degree $2m$. In a recent paper H.K. King (King 2016) discussed existing strategies for evaluation of Rys nodes and weights; see (Sagar & Smith 1992). An efficient and stable method for constructing Gauss-Rys quadratures have been recently given in (Milovanović 2018).

As we mention at the beginning here we consider more general case with respect to the weight function (1), which is a product of Gegenbauer weight function and the Rys exponential function. Since our quadrature rules reduce to the Rys quadrature for $\lambda = 1/2$, we will call them as the *generalized Gauss-Rys quadrature formulae*.

In the case $x \rightarrow 0$, our problem is reduced to the Gauss-Gegenbauer rule, where $\omega^\lambda(t; 0) = (1 - t^2)^{\lambda-1/2}$, $\lambda > -1/2$, and $\pi_k^\lambda(t; 0) = \widehat{C}_k^\lambda(t)$ are monic Gegenbauer polynomials, which satisfy three-term recurrence relation (4), with the coefficients; see (Mastroianni & Milovanović 2008, p. 102)

$$\begin{aligned} \alpha_k^\lambda(0) &= 0 \quad (k \in \mathbb{N}), \quad \beta_0^\lambda(0) = \sqrt{\pi} \frac{\Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)}, \\ \beta_k^\lambda(0) &= \frac{k(2\lambda + k - 1)}{4(\lambda + k - 1)(\lambda + k)}, \quad k \in \mathbb{N}, \end{aligned} \quad (6)$$

except the case $\lambda = 0$, when $\beta_1^0(0) = 1/2$. Otherwise, our weight function $t \mapsto \omega^\lambda(t; x)$ belongs to Szegő's class; see Definition 2.2.1 in (Mastroianni & Milovanović 2008), because the integral

$$\int_{-1}^1 \frac{\log \omega^\lambda(t; x)}{\sqrt{1-t^2}} dt = -\frac{\pi}{2} \left[(2\lambda - 1) \log(4) + x \right]$$

is a finite number ($> -\infty$), and therefore we have the following asymptotic property

$$\lim_{k \rightarrow +\infty} \beta_k^\lambda(x) = \frac{1}{4}. \quad (7)$$

To construct the Gaussian formula (2), i.e., (3), with at most N nodes, using the Golub-Welsch algorithm (Golub & Welsch 1969), we need the first N recursive coefficients in (4), $\beta_k = \beta_k^\lambda(x)$, $k = 0, 1, \dots, N-1$, which must be constructed numerically for this non-classical case. Such approaches belong to the so-called *constructive theory of orthogonal polynomials* developed by Walter Gautschi in the 1980s; see (Gautschi 1982); (Gautschi 2004); (Milovanović 2014). In general, in numerical construction of the recursive coefficients an important aspect is the sensitivity of the problem concerning a small perturbation in the input. However, recent progress in variable-precision arithmetic and symbolic computation enables us to generate recursive coefficients directly by applying the original Chebyshev method of moments, but by using sufficiently high precision arithmetic to overcome the numerical instability. Sometimes, we can even obtain these coefficients in symbolic form (usually for small N). Today, symbolic/variable-precision software for orthogonal polynomials is available: Gautschi's package SOPQ in MATLAB (Gautschi 2005); (Gautschi 2018) and our MATHEMATICA package OrthogonalPolynomials (Cvetković & Milovanović 2004); (Milovanović & Cvetković 2012), and both packages are freely downloadable.

Let $\mu_k^\lambda(x)$ be moments defined by

$$\mu_k^\lambda(x) = \int_{-1}^1 t^k e^{-xt^2} (1-t^2)^{\lambda-1/2} dt, \quad k \in \mathbb{N}_0.$$

Evidently, $\mu_k^\lambda(x) = 0$ for odd k . Using the integral formula for the Kummer confluent hypergeometric function ${}_1F_1(a; b; z)$,

$${}_1F_1(a; b; z) = \frac{\Gamma(b)}{\Gamma(a)\Gamma(b-a)} \int_0^1 t^{a-1} (1-t)^{b-a-1} e^{-zt} dt, \quad (8)$$

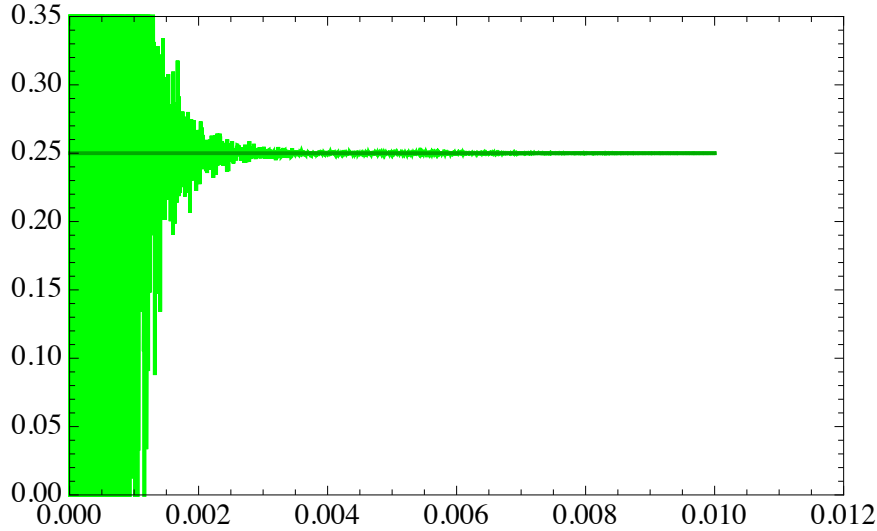


Fig. 1. Non-stability in calculating three-term recursive coefficients for small x (case $\beta_5^0(x)$).

we can get $\mu_k^\lambda(x)$, for each even k , in the form

$$\frac{\Gamma\left(\frac{k+1}{2}\right)\Gamma\left(\lambda+\frac{1}{2}\right)}{\Gamma\left(\frac{k}{2}+\lambda+1\right)} {}_1F_1\left(\frac{k+1}{2}; \frac{k}{2}+\lambda+1; -x\right).$$

In order to get N recurrence coefficients $\beta_k^\lambda(x)$, $k = 0, 1, \dots, N-1$ (the corresponding $\alpha_k^\lambda(x) = 0$, because the weight is an even function on $(-1, 1)$), we need the first $2N$ moments $\mu_k^\lambda(x)$. By the command `aChebyshevAlgorithm` in `MATHEMATICA` package `OrthogonalPolynomials`, in symbolic mode, we can obtain the recursion coefficients $\beta_k^\lambda(x)$ in symbolic form. For example, for $\lambda = 0$ and small $k \leq 10$, we get them in terms of the modified Bessel functions of the first kind $I_0(x/2)$ and $I_1(x/2)$,

$$\begin{aligned}\beta_0^0(x) &= \pi e^{-x/2} I_0\left(\frac{x}{2}\right), \\ \beta_1^0(x) &= \frac{I_0\left(\frac{x}{2}\right) - I_1\left(\frac{x}{2}\right)}{2I_0\left(\frac{x}{2}\right)}, \\ \beta_2^0(x) &= \frac{xI_0\left(\frac{x}{2}\right)^2 - 2I_1\left(\frac{x}{2}\right)I_0\left(\frac{x}{2}\right) - xI_1\left(\frac{x}{2}\right)^2}{2xI_0\left(\frac{x}{2}\right)\left(I_0\left(\frac{x}{2}\right) - I_1\left(\frac{x}{2}\right)\right)},\end{aligned}$$

etc. The obtained expressions for higher k are very complicated and unusable. Their computation for x near zero is not stable and requires an arithmetic of high precision. Non-stability in calculating $\beta_5^0(x)$ in standard *double precision* arithmetic, with machine precision $MP \approx 2.22 \times 10^{-16}$ (`$MachinePrecision` in the Wolfram `MATHEMATICA`), is presented in Figure 1. Only the use of high arithmetic, here with `WP=50` (`WorkingPrecision->50`), leads to a stable calculation (dark line in Figure 1). With such a calculation the recursive coefficients $\beta_k^0(x)$ for $k = 1, \dots, 6$ are shown in Figure 2, when x runs over $(0, 30)$.

According to the previous, we have to use numerical mode in our construction, i.e., the following commands

```
<<orthogonalPolynomials`
% mom=...(sequence of the length 2N)
{alpha,beta}=aChebyshevAlgorithm[mom, WorkingPrecision->WP];
```

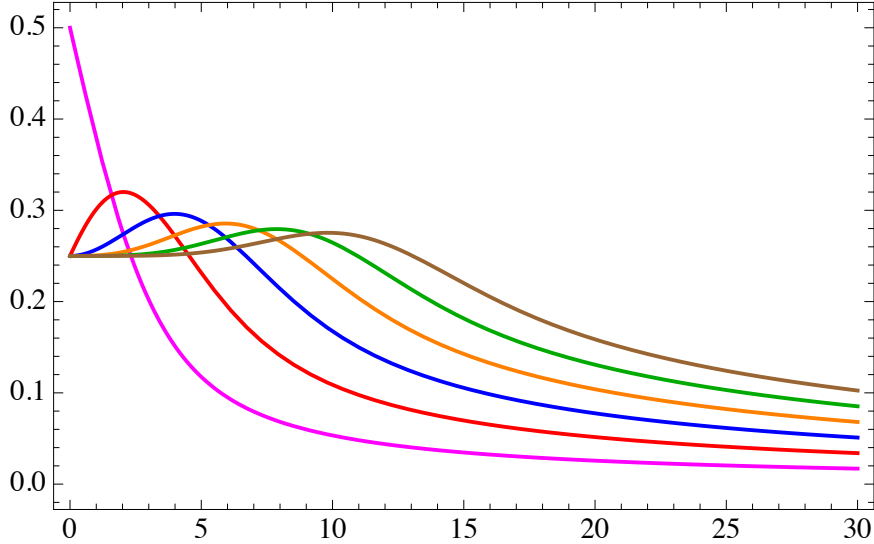


Fig. 2. The coefficients $\beta_k^0(x)$ for $x \in (0, 30)$ for $k = 1$ (magenta), $k = 2$ (red), $k = 3$ (blue), $k = 4$ (orange), $k = 5$ (green), and $k = 6$ (brown).

in order to get the sequences of the recurrence coefficients (of the length N), denoted by `alpha` (in this case, a zero sequence) and `beta`, with the maximal relative error

$$\text{err}_N^\lambda(x; \text{WP}) = \max_{0 \leq k \leq N-1} \left| \frac{\beta_k^\lambda(x) - \widehat{\beta}_k^\lambda(x)}{\widehat{\beta}_k^\lambda(x)} \right|.$$

Here, the “exact values” of the desired recurrence coefficients $\widehat{\beta}_k^\lambda(x)$ can be obtained using the same procedure, but with the higher working precision `WP1` (e.g., `WP1 = 2 WP`).

As in the special case ($\lambda = 1/2$) of the Rys polynomials (Milovanović 2018), this construction of the coefficients $\beta_k^\lambda(x)$ is unstable, again especially for small values of x . For example, to get the first $N = 50$ (100) coefficients for $x = 1/10$, with more than 16 exact decimal digits, i.e., when $\text{err}_N^\lambda(1/10; \text{WP}) < 10^{-16}$, we need the working precision at least `WP = 48` (86) for $\lambda = 0$, `WP = 46` (84) for $\lambda = 1$, and `WP` bigger than 105 (124) in the Legendre case $\lambda = 1/2$.

Therefore, in the next section, we construct polynomials orthogonal concerning the weight function $t \mapsto \omega^\lambda(\sqrt{t}; x)/\sqrt{t}$ on $(0, 1)$, using the method of modified moments (Gautschi 1982), as well as the corresponding quadrature rules of Gaussian type on $(0, 1)$. This approach enables us to have a stable and much easier construction of recursive coefficients and quadratures of the original problem on $(-1, 1)$, which is given in Section 3.

2. Orthogonal polynomials and Gaussian quadratures on (0,1)

Instead of construction the polynomials $t \mapsto \pi_k^\lambda(t; x)$ orthogonal on $(-1, 1)$, as well as the corresponding Gaussian formulas (2), with an even number of nodes $N = 2n$, with respect to the weight function defined in (1), in this section we use Theorem 2.2.11, p. 102, from (Mastroianni & Milovanović 2008), to transform the problem to the interval $(0, 1)$ for a new weight function $t \mapsto \Omega^\lambda(t; x)$, given by

$$\Omega^\lambda(t; x) = \frac{\omega^\lambda(\sqrt{t}; x)}{\sqrt{t}} = \frac{\exp(-xt)}{\sqrt{t}}(1-t)^{\lambda-1/2}.$$

Such an approach in construction the N -point Gaussian formulas, for the even weight function on a symmetric interval, needs only $N/2 = n$ nodes in the corresponding rules on the half

interval. In this manner, the influence of numerical instabilities in the process of construction can be significantly reduced, because the dimension of the Jacobi matrix is halved. Also, as we mentioned in the previous section, our construction is based on modified moments, which is an additional advantage over the direct construction of formulas on a symmetric interval.

Remark 1. If we want the rules (2), with an odd number of nodes $N = 2n + 1$, then the previous $\Omega^\lambda(t; x)$ should be replaced by the weight function $t \mapsto \sqrt{t} \omega^\lambda(\sqrt{t}; x)$. Such a construction is very similar to one for even N and we will consider only the case when N is even.

Let $p_k^\lambda(t; x)$ be monic orthogonal polynomials for the weight function $\Omega^\lambda(t; x)$ on $(0, 1)$. They satisfy the three-term recurrence relation for $k = 0, 1, \dots$; theorem 2.2.12, p. 102, in (Mastroianni & Milovanović 2008)

$$p_{k+1}^\lambda(t; x) = (t - a_k)p_k^\lambda(t; x) - b_k p_{k-1}^\lambda(t; x) \quad (9)$$

with $p_0^\lambda(t; x) = 1$, $p_{-1}^\lambda(t; x) = 0$. The recursion coefficients in (9) depend on λ and x and they are connected with ones in (4), i.e., $a_0 = \beta_1$ and

$$a_k = \beta_{2k} + \beta_{2k+1}, \quad b_k = \beta_{2k-1} \beta_{2k}. \quad (10)$$

According (7) we conclude that

$$\lim_{k \rightarrow +\infty} a_k = \frac{1}{2}, \quad \lim_{k \rightarrow +\infty} b_k = \frac{1}{16}. \quad (11)$$

Also, the parameters of the Gauss-Christoffel formula on $(0, 1)$,

$$\int_0^1 \Omega^\lambda(t; x) g(t) dt = \sum_{k=1}^n B_k g(\xi_k) + \widehat{R}_n(g), \quad (12)$$

which is exact for all $g \in \mathcal{P}_{2n-1}$, are connected with parameters of (3). Namely,

$$\pm \tau_k = \sqrt{\xi_k}, \quad A_k = \frac{1}{2} B_k, \quad k = 1, \dots, n. \quad (13)$$

For details see (Masjad-Jamei & Milovanović 2017).

To construct recursive coefficients in (9) we use the modified moments of the weight function $\Omega^\lambda(t; x)$ on $(0, 1)$ with respect to a system of polynomials $\{\phi_k\}$ ($\deg \phi_k = k$), chosen to be close in some sense to the desired orthogonal polynomials $p_k^\lambda(t; x)$. In our case an appropriate system of such polynomials are the monic Gegenbauer polynomials $\phi_k(t) = p_k^\lambda(t; 0) = \widehat{C}_{2k}^\lambda(\sqrt{t})$, $k \geq 0$.

These polynomials also satisfy a three-term recurrence relation

$$\phi_{k+1}(t) = (t - a_k^M) \phi_k(t) - b_k^M \phi_{k-1}(t) \quad (14)$$

for $k \geq 0$, with $\phi_0(t) = 1$ and $\phi_{-1}(t) = 0$. After some little calculations, we can obtain its recursive coefficients (Milovanović 2019):

$$\begin{aligned} a_0^M &= \frac{1}{2(\lambda + 1)}, & b_0^M &= \frac{\sqrt{\pi} \Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)}, \\ a_k^M &= \frac{4k^2 + 4\lambda k + \lambda - 1}{2(2k + \lambda - 1)(2k + \lambda + 1)}, & k &\geq 1, \\ b_1^M &= \frac{2\lambda + 1}{4(\lambda + 1)^2(\lambda + 2)}, \\ b_k^M &= \frac{k(2k - 1)(k + \lambda - 1)(2k + 2\lambda - 1)}{4(2k + \lambda - 2)(2k + \lambda - 1)^2(2k + \lambda)}, \end{aligned}$$

for $k \geq 2$.

For constructing the first n recurrence coefficients in (9), a_k and b_k , $k = 0, 1, \dots, n-1$, the method needs the first $2n$ modified moments

$$m_k^\lambda(x) = \int_0^1 \widehat{C}_{2k}^\lambda(\sqrt{t}) \Omega^\lambda(t; x) dt, \quad 0 \leq k \leq 2n-1.$$

2.1 Calculation of modified moments

In this subsection we calculate the modified moments.

Theorem 1. *Let $\widehat{C}_k^\lambda(t)$ be monic Gegenbauer polynomials orthogonal for the weight function $t \mapsto (1-t^2)^{\lambda-1/2}$, $\lambda > -1/2$, on $(-1, 1)$. The modified moments*

$$m_k^\lambda(x) = \int_0^1 \widehat{C}_{2k}^\lambda(\sqrt{t}) \frac{e^{-xt}}{\sqrt{t}} (1-t)^{\lambda-1/2} dt,$$

where $k = 0, 1, 2, \dots$, can be expressed via the Kummer confluent hypergeometric function as

$$m_k^\lambda(x) = \frac{(-1)^k \pi (2k + \lambda) (2k)! \Gamma(2k + 2\lambda + 1)}{2^{4k+2\lambda} (k + \lambda) k! \Gamma(2k + \lambda + 1)^2} x^k {}_1F_1\left(k + \frac{1}{2}; 2k + \lambda + 1; -x\right).$$

Alternatively,

$$m_k^\lambda(x) = \frac{(-1)^k e^{-x/2} x^k}{2^{2k+\lambda} (k + \lambda)_k} Q_k^\lambda(x), \quad k \in \mathbb{N}_0, \quad (15)$$

where $Q_k^\lambda(x)$ is an integral defined by

$$Q_k^\lambda(x) = \int_0^\pi e^{-\frac{x}{2} \cos \theta} \sin^{2k} \theta (1 - \cos \theta)^\lambda d\theta. \quad (16)$$

Proof. We start with a formula for Gegenbauer polynomials; see p. 529, Eq. 10, in (Prudnikov et al. 1986)

$$\begin{aligned} I &= \int_0^a x^{\alpha-1} (a^2 - x^2)^{\lambda-1/2} e^{-px^2} C_{2k+\varepsilon}^\lambda\left(\frac{x}{a}\right) dx \\ &= \frac{(-1)^k a^{\alpha+2\lambda-1}}{2(2k+\varepsilon)!} (2\lambda)_{2k+\varepsilon} \left(\frac{1+\varepsilon-\alpha}{2}\right)_k \\ &\quad \times \frac{\Gamma(\lambda + \frac{1}{2}) \Gamma(\frac{\alpha+\varepsilon}{2})}{\Gamma(b_2)} {}_2F_2(a_1, a_2; b_1, b_2; -a^2 p), \end{aligned} \quad (17)$$

which holds for $\varepsilon = 0$ or 1 , $a > 0$, $\operatorname{Re} \alpha > -\varepsilon$, and

$$\begin{aligned} a_1 &= \frac{\alpha}{2}, \quad a_2 = \frac{\alpha+1}{2}, \\ b_1 &= \frac{1+\alpha-\varepsilon}{2} - k, \quad b_2 = \frac{1+\alpha+\varepsilon}{2} + \lambda + k. \end{aligned}$$

Since

$$C_{2k}^\lambda(x) = \frac{2^{2k} (\lambda)_{2k}}{(2k)!} \widehat{C}_{2k}^\lambda(x),$$

putting $a = 1$, $\varepsilon = 0$ and $x = \sqrt{t}$, the integral I in (17) reduces to

$$\frac{2^{2k-1}(\lambda)_{2k}}{(2k)!} \int_0^1 t^{\alpha/2-1} (1-t)^{\lambda-1/2} e^{-pt} \widehat{C}_{2k}^\lambda(\sqrt{t}) dt,$$

and for $\alpha = 1$ and $p = x$ it becomes

$$I = \frac{2^{2k-1}(\lambda)_{2k}}{(2k)!} m_k^\lambda(x). \quad (18)$$

However, the right hand side in (17) is undetermined for $\alpha = 1$. Therefore, we will make some transformations of the expression on the right side and then let $\alpha \rightarrow 1$.

With $S_k \equiv S_k^{(\alpha, \lambda)}(x)$ we denote the expression

$$S_k = \left(\frac{1-\alpha}{2}\right)_k {}_2F_2(a_1, a_2; b_1, b_2; -x),$$

with parameters a_1, a_2, b_1, b_2 given before. Since

$$A := \left(\frac{1-\alpha}{2}\right)_k = (-1)^k \left(\frac{\alpha+1}{2} - k\right)_k = \frac{(-1)^k \Gamma(\frac{\alpha+1}{2})}{\Gamma(\frac{\alpha+1}{2} - k)},$$

we have that

$$\begin{aligned} S_k &= A \sum_{\nu=0}^{\infty} \frac{(\frac{\alpha}{2})_\nu (\frac{\alpha+1}{2})_\nu}{(\frac{\alpha+1}{2} - k)_\nu (\frac{\alpha+1}{2} + k + \lambda)_\nu} \frac{(-x)^\nu}{\nu!} \\ &= \sum_{\nu=0}^{\infty} \frac{(-1)^k (\frac{\alpha}{2})_\nu \Gamma(\frac{\alpha+1}{2} + \nu)}{\Gamma(\frac{\alpha+1}{2} - k + \nu) (\frac{\alpha+1}{2} + k + \lambda)_\nu} \frac{(-x)^\nu}{\nu!}. \end{aligned}$$

When $\alpha \rightarrow 1$, the first k terms in the sum $S_k^{(\alpha, \lambda)}(x)$ vanish, so that $S_k^{(1, \lambda)}(x) = \lim_{\alpha \rightarrow 1} S_k^{(\alpha, \lambda)}(x)$, after changing the index $\nu := \nu + k$, we get $S_k^{(1, \lambda)}(x)$ in the following form

$$(-1)^k \sum_{\nu=0}^{\infty} \frac{(\frac{1}{2})_{\nu+k} \Gamma(1 + \nu + k)}{\Gamma(1 + \nu) (1 + k + \lambda)_{\nu+k}} \frac{(-x)^{\nu+k}}{(\nu+k)!}.$$

Since $(\gamma)_{\nu+k} = (\gamma)_k (\gamma + k)_\nu$ it reduces to

$$\frac{x^k (\frac{1}{2})_k}{(1+k+\lambda)_k} \sum_{\nu=0}^{\infty} \frac{(\frac{1}{2} + k)_\nu}{(1+2k+\lambda)_\nu} \frac{(-x)^\nu}{\nu!},$$

i.e., $S_k^{(1, \lambda)}(x)$ can be expressed in terms of the Kummer confluent hypergeometric function

$$\frac{(\frac{1}{2})_k x^k}{(k+\lambda+1)_k} {}_1F_1\left(k + \frac{1}{2}; 2k + \lambda + 1; -x\right).$$

Thus, the right hand side of (17), for $a = 1$, $\varepsilon = 0$, $p = x$ and $\alpha \rightarrow 1$, becomes

$$\frac{(-1)^k (2\lambda)_{2k}}{2(2k)!} \frac{\Gamma(\lambda + \frac{1}{2}) \sqrt{\pi}}{\Gamma(k + \lambda + 1)} S_k^{(1, \lambda)}(x). \quad (19)$$

Finally, equating the expressions in (18) and (19) and using the well-known Legendre duplication formula

$$\Gamma(2z) = \frac{2^{2z-1}}{\sqrt{\pi}} \Gamma(z) \Gamma\left(z + \frac{1}{2}\right),$$

as well as the properties of gamma function, we obtain the modified moments $m_k^\lambda(x)$ as given in the statement of this Theorem.

Using the integral form (8) of the Kummer hypergeometric function ${}_1F_1(a; b; z)$, when $z = -x$, $a = k + \frac{1}{2}$ and $b = 2k + \lambda + 1$, after changes the variables $t = \cos^2 \theta$, we obtain the alternative expression (15) for the modified moments $m_k^\lambda(x)$. \square

Two interesting and important cases with the Chebyshev weights of the first kind ($\lambda = 0$) and the second kind ($\lambda = 1$) can be obtained as corollaries of the previous theorem.

Corollary 1. *Let $\widehat{T}_k(t)$ be monic Chebyshev polynomials of the first kind orthogonal with respect to the weight function $t \mapsto (1 - t^2)^{-1/2}$ on $(-1, 1)$. The modified moments*

$$m_k^0(x) = \int_0^1 \widehat{T}_{2k}(\sqrt{t}) \frac{e^{-xt}}{\sqrt{t(1-t)}} dt \quad (k \in \mathbb{N}_0)$$

can be expressed via the Kummer confluent hypergeometric function as

$$m_0^0(x) = \pi {}_1F_1\left(\frac{1}{2}; 1; -x\right) = \pi e^{-x/2} I_0\left(\frac{x}{2}\right)$$

and for $k \in \mathbb{N}$

$$m_k^0(x) = \frac{(-1)^k \pi}{2^{4k-1} k!} x^k {}_1F_1\left(k + \frac{1}{2}; 2k + 1; -x\right) \frac{(-1)^k \pi}{2^{2k-1}} e^{-x/2} I_k\left(\frac{x}{2}\right),$$

where $I_k(z)$ is the modified Bessel functions of the first kind and order k . Alternatively,

$$m_k^0(x) = \frac{(-1)^k e^{-x/2} x^k}{2^{2k} (k)_k} Q_k^0(x), \quad k \in \mathbb{N}_0,$$

where $Q_k^0(x)$ is the integral defined by Eq. (16).

Corollary 2. *Let $\widehat{U}_k(t)$ be monic Chebyshev polynomials of the second kind orthogonal with respect to the weight function $t \mapsto (1 - t^2)^{1/2}$ on $(-1, 1)$. The modified moments*

$$m_k^1(x) = \int_0^1 \widehat{U}_{2k}(\sqrt{t}) e^{-xt} \sqrt{\frac{1-t}{t}} dt \quad (k \in \mathbb{N}_0)$$

can be expressed via the Kummer confluent hypergeometric function as

$$m_k^1(x) = \frac{(-1)^k \pi}{2^{4k+1} k!} x^k {}_1F_1\left(k + \frac{1}{2}; 2k + 2; -x\right).$$

Alternatively,

$$m_k^1(x) = \frac{(-1)^k e^{-x/2} x^k}{2^{2k+1} (k+1)_k} Q_k^1(x) \quad (k \in \mathbb{N}_0),$$

where $Q_k^1(x)$ is the integral defined by Eq. (16).

2.2 Numerical construction

The MATHEMATICA package `OrthogonalPolynomials` provides also routines for a work with modified moments for generating the coefficients a_k and b_k in the relation (9). Alternative commands are

```
<<orthogonalPolynomials`
% akM=... (sequence of the length 2n)
% bkM=... (sequence of the length 2n)
% Mmom=...(sequence of the length 2n)
{a,b}=aChebyshevAlgorithmModified[Mmom,akM,bkM,
                                   WorkingPrecision->WP];
```

where akM and bkM are coefficients in the three-term recurrence relation (14), and $Mmom$ is the sequence of modified moments given in Theorem 1 by (15). For numerical calculating the integral $Q_k^\lambda(x)$, given in Eq. (16), we can use the standard command `NIntegrate` in MATHEMATICA, with the options

```
Method->"DoubleExponential" and WorkingPrecision->WP,
```

where WP is a given working precision.

For given λ , n and x , as well as the working precision WP , the coefficients a_k and b_k , $k = 0, 1, \dots, n-1$, in the recurrence relation (9) are obtained as the sequences a and b . Their maximal relative error is given by

$$\text{err}_n^\lambda(x; WP) = \max_{0 \leq k \leq n-1} \left\{ \left| \frac{a_k - \hat{a}_k}{\hat{a}_k} \right|, \left| \frac{b_k - \hat{b}_k}{\hat{b}_k} \right| \right\},$$

where, as before, the “exact values” of the desired coefficients \hat{a}_k and \hat{b}_k can be obtained using the same procedure, but with the higher working precision $WP1$. For example, if we take $\lambda = 0$ (see Corollary 1), $n = 100$ and $x = 1$, for $WP=30$ we get the first 100 recurrence coefficients a_k and b_k :

```
k          sequence a for x=1

0  0.378750193709599027324648823248
1  0.558108977120640683421160182517
2  0.500638815763627602857962430445
3  0.500002010402165855585412829707
4  0.500000003001355138913484171981
5  0.500000000002610206433841003986
6  0.50000000000000001484867909740311
7  0.50000000000000000000595403462627
8  0.50000000000000000000000177314407
9  0.50000000000000000000000000040763
10 0.50000000000000000000000000000007
>10 0.50000000000000000000000000000000

k          sequence b for x=1

0  2.02643806694935530514336305543
1  0.114048678184139268042120835991
```

```

2  0.0644019873749736944597342268427
3  0.0625100250180638822043294025279
4  0.0625000209807900172326264490836
5  0.0625000000234725623589669607001
6  0.0625000000000163246186141457655
7  0.06250000000000000077372313620488
8  0.0625000000000000000026589417479
9  0.062500000000000000000006928113
10 0.062500000000000000000000001416
>10 0.06250000000000000000000000000000

```

without loss of digits. As we can see the convergence of the previous sequences to the limit values (11) is very fast. These coefficients a_k and b_k , $k = 0, 1, \dots, 99$, give a possibility to construct Gaussian quadratures (12) for each $n \leq 100$, as well as the original rules (3) on $(-1, 1)$, i.e., (2), for each $N = 2n \leq 200$. Note that the rule (2), with $N = 200$ nodes is exact for all polynomials of degree at most $2N - 1 = 399$. The complete procedure is very fast.

We pay special attention to two interesting (Chebyshev) cases, when $\lambda = 0$ and $\lambda = 1$. In order to analyse stability in our construction we use two different arithmetics, WP=30 and the standard *double precision* arithmetic (WP=MP). In Table 1 we present the maximal relative errors of the recurrence coefficients $\text{err}_{100}^\lambda(x; \text{WP})$ for $x = 15, 20, 30$, when $\lambda = 0$ and $\lambda = 1$.

Table 1. Maximal relative errors of the recurrence coefficients $\text{err}_{100}^\lambda(x; \text{WP})$ for $x = 15, 20, 30$ in two different arithmetics

$\lambda = 0$			
WP	$x = 15$	$x = 20$	$x = 30$
30	$2. \times 10^{-29}$	6.30×10^{-27}	1.29×10^{-22}
MP	1.12×10^{-13}	3.05×10^{-11}	3.74×10^{-6}
$\lambda = 1$			
30	1.4×10^{-28}	3.03×10^{-26}	1.03×10^{-21}
MP	3.95×10^{-13}	1.26×10^{-10}	6.01×10^{-6}

The loss of digits practically does not exist for values of $x \leq 12$ for these values of λ . In both of these cases we have a loss of at most two, five or ten decimal digits, when $x = 15$, $x = 20$ or $x = 30$, respectively, regardless of which arithmetic we use. Thus, for $x < 12$, our method (the mapping of moments into recurrence coefficients) is *well-conditioned* and its condition number is near 1. Otherwise, roughly speaking, if the condition number of a mapping is 10^m , then approximately m decimal digits are lost. Therefore, we can conclude here that the condition number is approximately 10^2 , 10^5 and 10^{10} for the previously mentioned values of $x = 15$, 20 and 30. In such cases, if we need the accuracy of ℓ decimal digits in the recurrence coefficients a_k and b_k for each $k < n$, then we must use the working precision, which is at least $\text{WP} = \ell + m$ (Milovanović 2015). This means that for getting the first $n = 100$ recurrence coefficients with 16 exact decimal digits, our method needs only standard double precision arithmetics (WP=MP) when $x < 12$, but for larger x it needs $\text{WP} = \text{MP} + m$. For example, for $x \leq 30$ we should use $\text{WP} = 26$.

Now, for given λ and $X > 0$, we want to determine the coefficients $a_k(x)$ and $b_k(x)$ for each $x \in [0, X]$ and $k = 0, 1, \dots, n - 1$. Following Shizgal (Shizgal 2015), we first select a system

of points $S = \{x_\nu\}$ in the interval $[0, X]$, determine the corresponding values of the coefficients $a_k(x_\nu)$ and $b_k(x_\nu)$, $k = 0, 1, \dots, n-1$, and then construct the corresponding interpolating functions for each of these coefficients, which can be realized very easy in MATHEMATICA, using the function `Interpolation[{{x1, f1}, {x2, f2}, ...}]`. It constructs an interpolation of the function values f_ν corresponding to x values x_ν , and returns an `InterpolatingFunction` object, which can be used like any other pure function. Interpolation works by fitting polynomial curves between successive data points, by degree of the polynomial curves which can be specified by the option `InterpolationOrder->r`, with default setting `r=3`, which is quite satisfactory for our purpose.

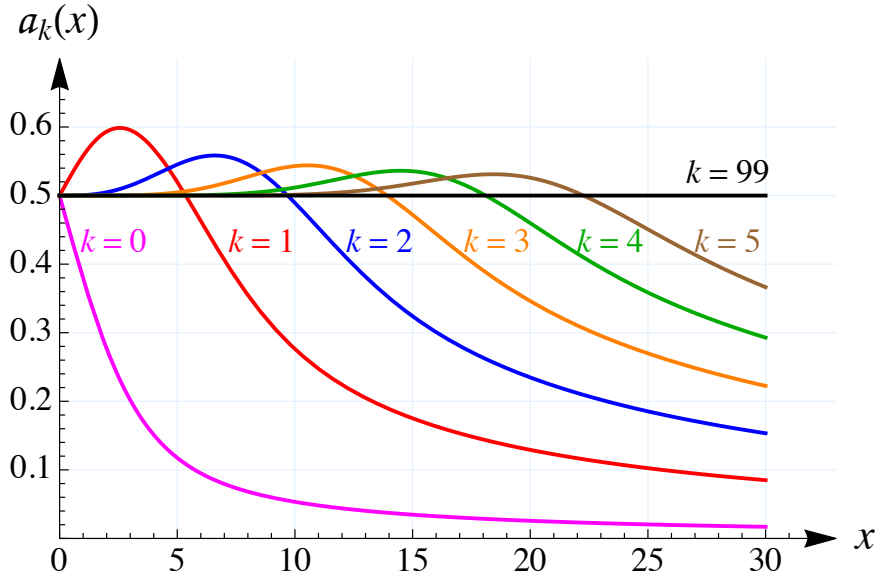


Fig. 3. The recurrence coefficients $a_k(x)$ for $x \in (0, 30)$ in the case $\lambda = 0$.

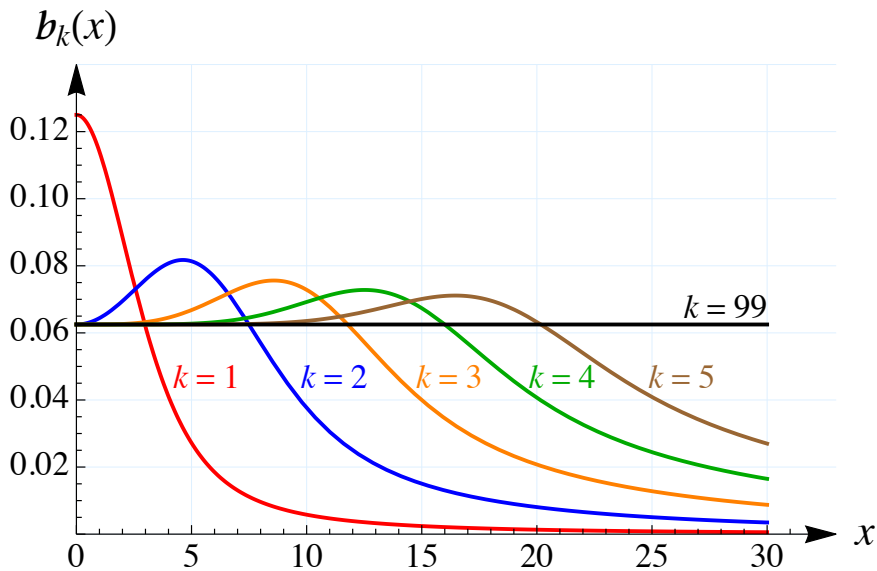


Fig. 4. The recurrence coefficients $b_k(x)$ for $x \in (0, 30)$ in the case $\lambda = 0$.

We illustrate this procedure for $\lambda = 0$, $X = 30$, and, because of simplicity, we take an equidistant system of points, e.g.,

$$S = \left\{ x_\nu = \frac{\nu}{10} \mid \nu = 0, 1, \dots, 300 \right\},$$

although it is also possible with fewer points with a significantly larger spacing in the second half of the interval. The obtained interpolating functions for our coefficients we will denote by again as functions $x \mapsto a_k(x)$ and $x \mapsto b_k(x)$, $k = 0, 1, \dots, 99$. In Figures 3 and 4 we present these coefficients. The corresponding graphics for $\lambda = 1$ are displayed in Figures 5 and 6.

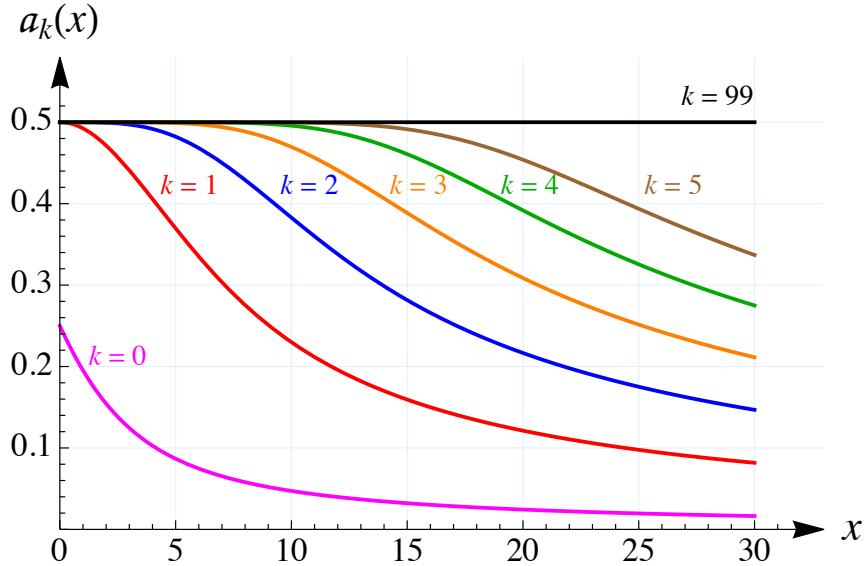


Fig. 5. The recurrence coefficients $a_k(x)$ for $x \in (0, 30)$ in the case $\lambda = 1$.

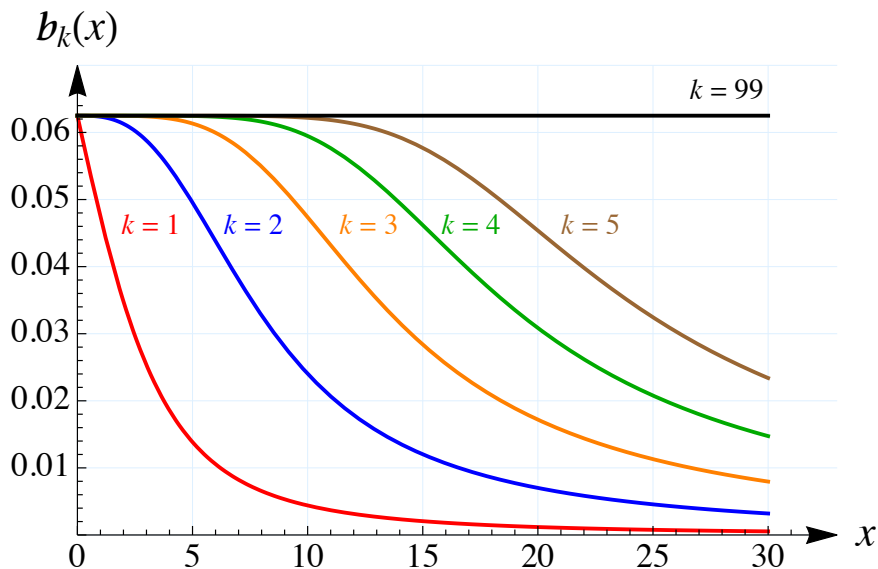


Fig. 6. The recurrence coefficients $b_k(x)$ for $x \in (0, 30)$ in the case $\lambda = 1$.

These coefficients enable us to obtain the parameters of the quadrature formula (12), the nodes ξ_k and the weight coefficients B_k , $k = 1, \dots, n$ for each n (in our example ≤ 100), using the Golub-Welsch algorithm (Golub & Welsch 1969), with the corresponding three-diagonal

Jacobi matrix $J_n(\Omega^\lambda(\cdot; x))$, given by

$$\begin{pmatrix} a_0(x) & \sqrt{b_1(x)} & & & \mathbf{0} \\ \sqrt{b_1(x)} & a_1(x) & \sqrt{b_2(x)} & & \\ & \sqrt{b_2(x)} & a_2(x) & \ddots & \\ & & \ddots & \ddots & \sqrt{b_{n-1}(x)} \\ \mathbf{0} & & & \sqrt{b_{n-1}(x)} & a_{n-1}(x) \end{pmatrix},$$

whose eigenvalues are the nodes ξ_k , $k = 1, \dots, n$, and the weight coefficients B_k can be obtained from the first component of the normalized eigenvectors corresponding to ξ_k . This algorithm is also included in the MATHEMATICA package `OrthogonalPolynomials`. The corresponding command is

```
{xi, B}=aGaussianNodesWeights[n, a, b,
WorkingPrecision->WP, Precision->PR]
```

where a and b are sequences of recurrence coefficients, and PR is a required precision of the sequences xi (nodes) and B (weight coefficients). This process is stable, so that we usually set $WP=PR+5$.

3. Gaussian quadratures and orthogonal polynomials on $(-1,1)$

With the obtained parameters of the n -point quadrature rule (12) on $(0, 1)$, ξ_k and B_k , we can get easily the quadrature parameters τ_k and A_k in the original quadrature formula (3) on $(-1, 1)$, using the relations (13). In this way, we get parameters in this $2n$ -point symmetric quadrature rule by solving the eigenvalue problem with the Jacobi matrix of the order only n .

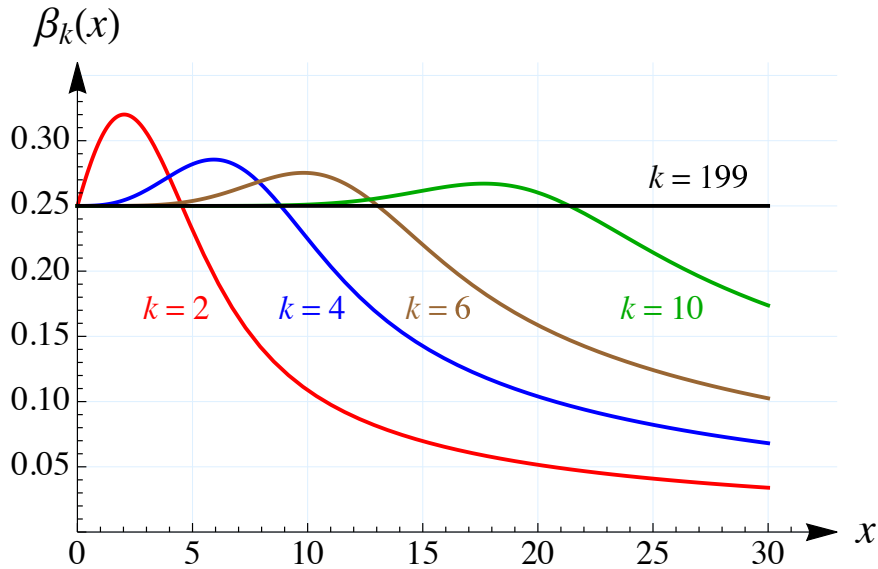


Fig. 7. Case $\lambda = 0$: The recurrence coefficients $x \mapsto \beta_k(x)$, for $k = 2, 4, 6, 10$ and 199 , when x runs over $[0, 30]$.

Also, we can easily get the coefficients β_k in the three-term recurrence relation (4) for polynomials $\pi_k^\lambda(t; x)$ orthogonal on $(-1, 1)$ with respect to the weight function $t \mapsto \omega^\lambda(t; x)$ given

in (1), using the relations (10). Namely, knowing a_k and b_k for $k = 0, 1, \dots, n - 1$, we have

$$\beta_0 = \sqrt{\pi} \frac{\Gamma(\lambda + \frac{1}{2})}{\Gamma(\lambda + 1)} {}_1F_1\left(\frac{1}{2}, \lambda + 1, -x\right), \beta_1 = a_0,$$

and

$$\beta_{2k} = \frac{b_k}{\beta_{2k-1}}, \beta_{2k+1} = a_k - \beta_{2k}, k = 1, \dots, n - 1,$$

for which we can also construct the corresponding interpolating functions on the set S as in the previous section for coefficients in the three-term recurrence relation (9). In Figures 7 and 8 we present graphics for some selected coefficients β_k in the cases when $\lambda = 0$ and $\lambda = 1$, respectively.

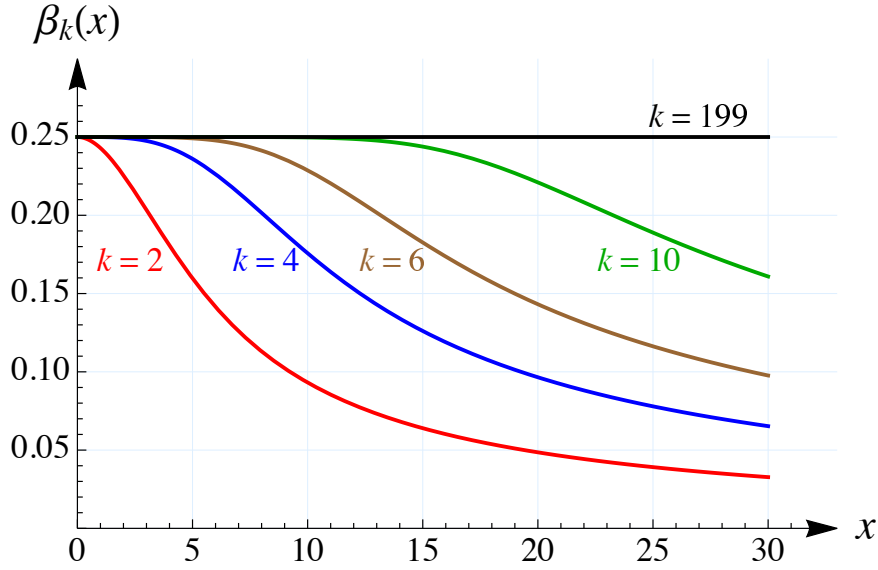


Fig. 8. Case $\lambda = 1$: The recurrence coefficients $x \mapsto \beta_k(x)$, for $k = 2, 4, 6, 10$ and 199 , when x runs over $[0, 30]$.

4. Conclusion

In this paper we propose a stable and efficient algorithm for constructing orthogonal polynomials and the corresponding quadrature formulas of Gaussian type with respect to the even weight function $\omega^\lambda(t; x) = \exp(-xt^2)(1 - t^2)^{\lambda-1/2}$ on $(-1, 1)$, with parameters $\lambda > -1/2$ and $x > 0$. For $\lambda = 1/2$ these quadrature rules reduce to the so-called Gauss-Rys quadrature formulas, which were investigated earlier by several authors. Two advantages of the proposed approach are the transformation of the original problem from $(-1, 1)$ to one on $(0, 1)$ and the use the method of modified moments for achieving the stability and reduction of numerical complexity of the algorithm. As a possible applications of these quadrature formulas we mention an application in solving Fredholm integral equations with the kernel $K(x, t) = e^{-xt^2}(1 - t^2)^{\lambda-1/2}$ on $[-1, 1]$, see (Asanov *et al.* 1917, Mastroianni & Milovanović 2009).

ACKNOWLEDGEMENTS

The work of the first author was supported in part by the Serbian Academy of Sciences and Arts (Φ -96). The authors are grateful to the referee for making certain useful suggestions.

References

- Asanov, A., Matanova, K. B., Asanov, R. A. (2017).** A class of linear and nonlinear Fredholm integral equations of the third kind. *Kuwait J. Sci.* **44** (1), 17–28.
- Cvetković, A. S., Milovanović, G. V. (2004).** The Mathematica package “OrthogonalPolynomials”. *Facta Univ. Ser. Math. Inform.* **19**, 17–36.
- Dupuis, M., Rys, J., King, H.F. (1976).** Evaluation of molecular integrals over Gaussian basis functions. *J. Chem. Phys.* **65**, 111–116.
- Gautschi, W. (1982).** On generating orthogonal polynomials. *SIAM J. Sci. Statist. Comput.* **3**, 282–317.
- Gautschi, W. (2004).** *Orthogonal Polynomials: Computation and Approximation*. Clarendon Press, Oxford.
- Gautschi, W. (2005).** Orthogonal polynomials (in Matlab). *J. Comput. Appl. Math.* **178**, 215–234.
- Gautschi, W. (2016).** *Orthogonal Polynomials in Matlab: Exercises and Solutions, Software-Environments-Tools*. SIAM, Philadelphia, PA.
- Gautschi, W. (2018).** *A Software Repository for Orthogonal Polynomials. Software, Environments and Tools*. Vol. 28. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
- Golub, G., Welsch, J. H. (1969).** Calculation of Gauss quadrature rules. *Math. Comp.* **23**, 221–230.
- King, H.F. (2016).** Strategies for evaluation of Rys roots and weights. *J. Phys. Chem. A* **120**, 9348–9351.
- Masjad-Jamei, M., Milovanović, G. V. (2017).** Construction of Gaussian formulas for even weight function. *Appl. Anal. Discrete Math.* **11**, 177–198.
- Mastroianni, G., Milovanović, G. V. (2008).** *Interpolation Processes – Basic Theory and Applications*. Springer Monographs in Mathematics, Springer – Verlag, Berlin –Heidelberg.
- Mastroianni, G., Milovanović, G. V. (2009).** Well-conditioned matrices for numerical treatment of Fredholm integral equations of the second kind. *Numer. Linear Algebra Appl.* **16**, 995–1011.
- Milovanović, G. V. (2014).** Chapter 11: Orthogonal polynomials on the real line. In: *Walter Gautschi: Selected Works with Commentaries, Volume 2* (C. Brezinski, A. Sameh, eds.), pp. 3–16, Birkhäuser, Basel.
- Milovanović, G. V. (2015).** Construction and applications of Gaussian quadratures with non-classical and exotic weight functions. *Stud. Univ. Babeş-Bolyai Math.* **60**, 211–233.
- Milovanović, G. V. (2018).** An efficient computation of parameters in the Rys quadrature formula. *Bull. Cl. Sci. Math. Nat. Sci. Math.* **43**, 39–64.

Milovanović, G. V. (2019). A note on extraction of orthogonal polynomials from generating function for reciprocal of odd numbers. *Indian J. Pure Appl. Math.* **50**, 15–22.

Milovanović, G. V., Cvetković, A. S. (2012). Special classes of orthogonal polynomials and corresponding quadratures of Gaussian type. *Math. Balkanica* **26**, 169–184.

Milovanović, G. V., Rathie, A. K. (2019). On a quadratic transformation due to Exton and its generalization. *Hacet. J. Math. Stat.* **48**(6), 1706–1711.

Milovanović, G. V., Parmar, R. K., Rathie, A. K. (2018). A study of generalized summation theorems for the series ${}_2F_1$ with an applications to Laplace transforms of convolution type integrals involving Kummer's functions ${}_1F_1$. *Appl. Anal. Discrete Math.* **12**, 257–272.

Ostrowska, S., Turan, M. (2017). On the powers of the Kummer distribution. *Kuwait J. Sci.* **44** (2), 1–8.

Prudnikov, A. P., Brychkov, Yu. A., Marichev, O. I. (1986). *Integrals and Series, Vol. 2 Special Functions*, Gordon & Breach Science Publisher, New York.

Rys, J., Dupuis, M., King, H. F. (1983). Computation of electron repulsion integrals using Rys quadrature method. *J. Comput. Chem.* **4**, 154–157.

Sagar, R. P., Smith, V. H. (1992). On the calculation of Rys polynomials and quadratures. *Int. J. Quant. Chem.* **42**(4), 827–836.

Schwenke, R. P. (2014). On the computation of high order Rys quadrature weights and nodes. *Comput. Phys. Comm.* **185**, 762–763.

Shizgal, B. D. (2015). A novel Rys quadrature algorithm for use in the calculation of electron repulsion integrals. *Comput. Theor. Chem.* **1074**, 178–184.

Submitted: 02/10/2020

Revised: 19/01/2021

Accepted: 19/01/2021

DOI: 10.48129/kjs.v49i1.10665