AN EFFICIENT COMPUTATION OF PARAMETERS IN THE RYS QUADRATURE FORMULA

GRADIMIR V. MILOVANOVIC

Dedicated to Professor Bogoljub Stankovic (1924–2018)

(Presented at the 8th Meeting, held on November 28, 2018)

Abstract. We present an efficient procedure for constructing the so-called Gauss-Rys quadrature formulas and the corresponding polynomials orthogonal on \((-1, 1)\) with respect to the even weight function \(w(t; x) = \exp(-xt^2)\), where \(x\) a positive parameter. Such Gauss-Rys quadrature formulas were investigated earlier e.g. by M. Dupuis, J. Rys, H.F. King [J. Chem. Phys. 65 (1976), 111 – 116; J. Comput. Chem. 4 (1983), 154 – 157], D.W. Schwenke [Comput. Phys. Comm. 185 (2014), 762 – 763], and B.D. Shizgal [Comput. Theor. Chem. 1074 (2015), 178 – 184], and were used to evaluate electron repulsion integrals in quantum chemistry computer codes. The approach in this paper is based to a transformation of quadratures on \((-1, 1)\) with \(N\) nodes to ones on \((0, 1)\) with only \([(N + 1)/2]\) nodes and their construction. The method of modified moments is used for getting recurrence coefficients. Numerical experiments are included.

AMS Mathematics Subject Classification (2000): 41A55, 65D30, 65D32, 33C45.

Key Words: numerical integration, Gaussian quadrature, orthogonal polynomials.
1. Introduction and preliminaries

The so-called Rys quadrature formulae have been introduced in 1976 by Dupius, Rys and King [3] as an attractive method in computational quantum chemistry for evaluating two-electron repulsion integrals,

\[ (ij|kl) = \int \int \phi_i(1)\phi_j(1) \frac{1}{r_{12}} \phi_k(2)\phi_l(2) \, d\tau_1 \, d\tau_2, \]

which appear in molecular quantum mechanical calculations involving Gaussian Cartesian basis functions. As it was explained in [3], [9], [25], and [27] it leads to the calculation of one-dimensional integrals of the form

\[ (ij|kl) = \int_0^1 f_m(t) \exp(-xt^2) \, dt, \quad (1.1) \]

where \( f_m(t) \) are even algebraic polynomials of degree \( 2m \) and the weight function is given by \( w(t; x) = \exp(-xt^2) \), where \( x \) a positive parameter. Because \( t \mapsto w(t; x) \) is an even weight function on \((-1, 1)\), the previous integral can be expressed as a half of the corresponding integral over the symmetric interval \((-1, 1)\).

The Rys quadrature formulas are Gaussian on the finite interval \((-1, 1)\) with respect to the exponential weight function \( w(t; x) = \exp(-xt^2) \). The corresponding (monic) polynomials with respect to the weight \( w(t; x) \) we denote by \( \pi_n(t; x) \), and they are known as Rys polynomials ([3], [25], [27], [26]). These polynomials are even or odd polynomials depending on the parity of \( n \). They satisfy the three-term recurrence relation

\[ \pi_{k+1}(t; x) = t\pi_k(t; x) - \beta_k \pi_{k-1}(t; x), \quad k = 1, 2, \ldots, \quad (1.2) \]

with \( \pi_0(t; x) = 1 \) and \( \pi_{-1}(t; x) = 0 \). The recursion coefficients depend on the parameter \( x \), \( \beta_k = \beta_k(x) > 0, k = 1, 2, \ldots \). The coefficient \( \beta_0 \) in (1.2) may be arbitrary, but is conveniently defined by (cf. [6], [12])

\[ \beta_0(x) = \int_{-1}^1 w(t; x) \, dt = \sqrt{\frac{\pi}{x}} \text{erf} \left( \frac{\sqrt{x}}{x} \right) = \frac{\sqrt{\pi} - \Gamma(1/2, x)}{\sqrt{x}}, \quad (1.3) \]

where \( \Gamma(a, z) \) is the incomplete gamma function defined by \( \Gamma(a, z) = \int_z^\infty t^{a-1} e^{-t} \, dt \).

In the case \( x \to 0 \), Rys formulas reduce to the well-known Gauss-Legendre rules. Evidently, \( w(t; 0) = 1 \), and therefore (cf. [14, p. 148])

\[ \beta_0(0) = 2, \quad \beta_k(0) = \frac{k^2}{4k^2 - 1}, \quad k = 1, 2, \ldots, \quad (1.4) \]
An efficient computation of parameters in the Rys quadrature formula

\[
\pi_k(t; 0) = \hat{P}_k(t) = \frac{2^k k!}{(k + 1)_k} P_k(t) = \frac{2^k}{(2k)_k} P_k(t),
\]

where \( P_k(t) \) are the standard Legendre polynomials. Note that

\[
\hat{P}_k(t) = 2^k k! \left( \frac{k}{2k} \right) P_k(t),
\]

are monic Legendre polynomials. Note that \( \hat{P}_k(t) = 2^k k! \left( \frac{k}{2k} \right) P_k(t) \).

When \( x \to +\infty \), these quadratures have an asymptotic behaviour like Gauss-Hermite quadratures [12, p. 325]. Let \( p_k(t; x) \) be an orthonormal polynomial, i.e., \( p_k(t; x) = \gamma_k(x) \pi_k(t; x) \), for which we have

\[
\gamma_k(x)^2 \int_{-1}^{1} \exp(-xt^2) \pi_k(t; x)^2 \, dt = 1,
\]

i.e.,

\[
\gamma_k(x)^2 \int_{-\sqrt{x}}^{\sqrt{x}} \exp(-\xi^2) \pi_k(\frac{\xi}{\sqrt{x}}; x)^2 \, d\xi = 1.
\]

When \( x \to +\infty \), we get

\[
\lim_{x \to +\infty} \sqrt{x} \beta_k(x) = \sqrt{\frac{k}{2}}
\]

and

\[
\lim_{x \to +\infty} \frac{\gamma_k(x)}{x^{1/4}} \pi_k\left( \frac{\xi}{\sqrt{x}}; x \right) = \frac{2^{k/2}}{\sqrt{k!} \sqrt{\pi}} \hat{H}_k(\xi),
\]

where \( \hat{H}_k(\xi) \) is the monic Hermite polynomial of degree \( k \).

Also, it is interesting to see that the weight function \( w(t; x) = \exp(-xt^2) \) belongs to Szegő’s class (see Definition 2.2.1 in [12, p. 103]), because of

\[
\int_{-1}^{1} \log w(t; x) \frac{1}{\sqrt{1 - t^2}} \, dt = \int_{-1}^{1} \frac{-xt^2}{\sqrt{1 - t^2}} \, dt = -\frac{\pi x}{2} > -\infty.
\]

As a consequence of it is the following asymptotic property of the recurrence coefficients \( \beta_k = \beta_k(x) \) in (1.2) (cf. [10])

\[
\lim_{k \to \infty} \beta_k(x) = \beta_{\infty}(x) = \frac{1}{4},
\]

In the sequel in this paper, we use the generalized hypergeometric function \( {}_pF_q \), defined by

\[
{}_pF_q \left( a_1, \ldots, a_p; b_1, \ldots, b_q; z \right) = \sum_{\nu=0}^{\infty} \frac{(a_1)_\nu \cdots (a_p)_\nu}{(b_1)_\nu \cdots (b_q)_\nu} \frac{z^\nu}{\nu!}.
\]
where the Pochhammer symbol \((\lambda)_\nu\) is given by

\[
(\lambda)_0 = 1, \quad (\lambda)_\nu = \lambda(\lambda + 1) \cdots (\lambda + \nu - 1) = \frac{\Gamma(\lambda + \nu)}{\Gamma(\lambda)},
\]

and \(\Gamma(\lambda)\) is Euler’s gamma function defined by

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

In particular, the so-called Gauss hypergeometric functions \(2F_1\) play a fundamental role in the applied mathematics and mathematical physics. In Wolfram’s MATHEMATICA the function \(pF_q\) is implemented as \(\text{HypergeometricPFQ}\) 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 \(\infty\). When \(p \leq q\) the above series on the right-hand side converges for each \(z \in \mathbb{C}\). For some recent results on this subject, especially on transformations, summations and some applications see [20], [21], [22].

According to (1.2), the Rys \(N\)-point Gaussian formulas

\[
I(f; x) = \int_{-1}^{1} w(t; x)f(t) \, dt = Q_N(f; x) + R_N(f),
\]

with the remainder term \(R_N(f)\), are symmetric rules, which are exact for all polynomials of degree at most \(2N - 1\), as well as for any odd function.

For example, for \(N = 2n\) this formula can be expressed in the form

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

where \(\tau_k = \tau_k^{(N)} = \tau_k^{(N)}(x), A_k = A_k^{(N)} = A_k^{(N)}(x) > 0\), and

\[
0 < \tau_1 < \cdots < \tau_n < 1.
\]

If \(N = 2n + 1\), the quadrature sum contains an additional term \(A_0f(0)\), i.e.,

\[
Q_{2n+1}(f; x) = A_0f(0) + \sum_{k=1}^{n} A_k(f(\tau_k) + f(-\tau_k)),
\]

where \(A_0 = A_0^{(N)} = A_0^{(N)}(x) > 0\).
Notice that the integral (1.1) can be computed exactly, except for rounding errors, by using the previous quadrature formulas as

\[(ij, kl) = \frac{1}{2} \int_{-1}^{1} f_m(t) \exp(-xt^2) \, dt = \frac{1}{2} Q_N(f_m; x),\]

providing \(N \geq m + 1\). Here,

\[
\frac{1}{2} Q_N(f_m; x) = \begin{cases} 
\sum_{k=1}^{N/2} A_k^{(N)} f_m(\tau_k^{(N)}), & N = 2n, \\
\frac{1}{2} A_0^{(N)} f_m(0) + \sum_{k=1}^{(N-1)/2} A_k^{(N)} f_m(\tau_k^{(N)}), & N = 2n + 1,
\end{cases}
\]

with a degree of precision \(d(N) = 2N - 1\), which is equal to \(4n \pm 1\) in these cases, respectively.

The authors of the previous mentioned papers [3], [25], [27], [26] interested also in polynomials \(p_k(t^2; x) = \pi_{2k}(t; x)\), which are orthogonal on \((0, 1)\).

A detailed discussion on methods for constructing Rys quadratures on \((-1, 1)\), as well as for constructing the so-called half range quadratures (with the same weight function on \((0, 1)\)), was recently presented by Shizgal [28], including a construction of orthogonal polynomials by using discretizing Stieltjes-Gautschi procedure (cf. [12, 162–166]). The classical Chebyshev method of moments is ill-conditioned.

In this paper we will first give some numerical experiments to show that the classical Chebyshev method is ill conditioned and almost inapplicable, and then we give some possibilities for its application using recent advances in symbolic computation and arithmetic of variable precision. However, the main part of our paper is an efficient procedure for constructing the Gauss-Rys quadrature formulas and the corresponding orthogonal polynomials based on a transformation of quadratures on \((-1, 1)\) with \(N\) nodes to ones on \((0, 1)\) with only \([N+1]/2\) nodes and their construction. The method of modified moments is used for getting recurrence coefficients. Numerical experiments are included.

2. Conditionality of the classical Chebyshev method

In a series of papers in the eighties of the last century (see [4], [6]), Walter Gautschi developed the so-called constructive theory of orthogonal polynomials on \(\mathbb{R}\), including effective algorithms for numerically generating orthogonal polynomials (method of modified moments, discretized Stieltjes-Gautschi procedure, Lanczos algorithm) and a detailed stability analysis of such algorithms, as well as several
new applications of orthogonal polynomials. He also provided software necessary for implementing these algorithms in Matlab (cf. [7]). This theory opened the door for extensive computational work on orthogonal polynomials and many their applications.

In general, in numerical construction of recursion coefficients an important aspect is the sensitivity of the problem with respect to small perturbation in the input. There is a simple algorithm, due to Chebyshev, which transforms the first $2N$ moments to $2N$ desired recursion coefficients (method of moments)

$$\mu = (\mu_0, \mu_1, \ldots, \mu_{2N-1}) \mapsto \rho = (\alpha_0, \ldots, \alpha_{N-1}, \beta_0, \ldots, \beta_{N-1}),$$

but its effectiveness depends critically on the conditioning of the mapping $K_N : \mathbb{R}^{2N} \to \mathbb{R}^{2N} (\mu \mapsto \rho)$. Usually it is ill-conditioned and practically, these calculations via moments in finite precision on a computer are quite ineffective because of the explosive growth of rounding errors.

However, recent progress in symbolic computation and variable-precision arithmetic now makes it possible to generate the coefficients in the three-term recurrence relation for orthogonal polynomials directly by using the original Chebyshev method of moments, but in a sufficiently high precision arithmetic in order to overcome the numerical instability.

Respectively symbolic/variable-precision software for orthogonal polynomials is available: Gautschi’s package SOPQ in MATLAB (see Appendix B in [7]) and our MATHEMATICA package OrthogonalPolynomials (see [2] and [18]), which is downloadable from the Web site in the Mathematical Institute of the Serbian Academy of Sciences and Arts: http://www.mi.sanu.ac.rs/~gvm/.

As we mentioned, the map $K_N$ is usually ill-conditioned, i.e., its condition number is much larger than one, $\text{cond } K_N \gg 1$. If the condition number is of order $10^m$, it roughly means a loss of $m$ decimal digits in results when the input data are perturbed by one units in the last digit. For example, if the working precision is WP decimal digits, e.g., WP $\approx$ MP $\approx 16$ and the condition number is $10^{10}$, then results will be accurate to only about $16 - 10 = 6$ digits! Here, MP denotes the $\text{MachinePrecision}$ number (notation in the Wolfram’s package MATHEMATICA), which is equal to 15.9546 ($\approx 16$).

Remark 2.1. In the so-called machine floating-point arithmetic an important number is $\epsilon_M = 2^{-n+1}$, where $n$ is the number of binary bits used in the internal representation of machine-precision floating-point numbers. It gives the difference between 1 and the next-nearest number representable as a machine-precision number (see [15, pp. 16–27]). Typical value of this number $\epsilon_M$ (machine epsilon, macheps or unit roundoff) in the double precision arithmetic ($n = 53$) is $\approx 2.22045 \times 10^{-16}$. In the Wolfram MATHEMATICA this constant is denoted by $\text{MachineEpsilon}$.
An efficient computation of parameters in the Rys quadrature formula

In the sequel we consider the construction of the orthogonal polynomials \( \pi_k(t; x) \) defined by the recurrence relation (1.2), i.e., the construction of the recurrence coefficients \( \beta_k \) (here, \( \alpha_k = 0 \) because the weight is an even function on \((-1, 1))

In order to get \( N \) recurrence coefficients \( \beta_k \), we need the first \( 2N \) moments \( \mu_k \), \( k = 0, 1, \ldots, 2N - 1 \), which can be expressed in terms of incomplete gamma functions,

\[
\mu_k = \int_{-1}^{1} t^k e^{-xt^2} \, dt = \begin{cases} 
  x^{-(k+1)/2} \left[ \Gamma \left( \frac{k+1}{2} \right) - \Gamma \left( \frac{k+1}{2}, x \right) \right], & k \text{ even} \\
  0, & k \text{ odd}
\end{cases}
\]

Taking concrete value of \( N \) and \( x \), as well as the working precision \( WP \), by using our \textsc{mathematica} package \texttt{OrthogonalPolynomials} (see [2] and [18]), with the following commands:

```mathematica
<< orthogonalPolynomials'
mom[n_, x_] := Table[
  If[k==0, Sqrt[Pi] Erf[Sqrt[x]]/Sqrt[x],
    If[OddQ[k], 0, x^(-(1+k)/2)(Gamma[(1+k)/2] - Gamma[(1+k)/2, x])]],
  {k, 0, 2n-1}];
momNx=mom[N, x];
{alpha, beta} = aChebyshevAlgorithmModified[momNx, WorkingPrecision -> WP];
```

we get the sequence of the recurrence coefficients, denoted by \( \beta_k \) (alpha is a zero sequence), with the maximal relative error

\[
\text{err}_N(x; WP) = \max_{0 \leq k \leq N-1} \left| \frac{\beta_k(x) - \hat{\beta}_k(x)}{\beta_k(x)} \right|.
\]

In the previous expression the exact values of the desired recurrence coefficients are denoted by \( \hat{\beta}_k(x) \) and their values can be obtained using the same procedure, but with the higher working precision \( WP1 \) (e.g., with \( WP1 = 2 WP \)).

**Table 1.** Maximal relative errors of the recurrence coefficients \( \beta_k(x) \), \( k = 1, \ldots, N - 1 \), for \( N = 10 \) and four values of \( x \) in two different arithmetics

<table>
<thead>
<tr>
<th>WP</th>
<th>( x = 1/10 )</th>
<th>( x = 1 )</th>
<th>( x = 10 )</th>
<th>( x = 25 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP</td>
<td>( 8.66 \times 10^4 )</td>
<td>( 1.4 \times 10^{-14} )</td>
<td>( 3. \times 10^{-15} )</td>
<td>( 0. \times 10^{-16} )</td>
</tr>
<tr>
<td>30</td>
<td>( 2.51 \times 10^{-25} )</td>
<td>( 1.4 \times 10^{-28} )</td>
<td>( 0. \times 10^{-30} )</td>
<td>( 0. \times 10^{-30} )</td>
</tr>
</tbody>
</table>
For a small value $N = 10$ we get the results with the maximal relative errors $\text{err}_N(x; \wp)$ presented in Table 1. As we can see in the the standard arithmetic, the effect of loss of digits appears for small values of $x (\leq 10)$. In particular, it is clear from the example with $\wp = 30$, the losses are two and five digits when $x = 1$ and $x = 1/10$, respectively. This means that the corresponding condition numbers of the mapping $K_{10}$ for $x = 1$ and $x = 1/10$ are approximately equal to $10^2$ and $10^5$, respectively. In Table 2 we present the approximative condition numbers of the mapping $K_N$, obtained by numerical experiments for the same values of $x$, in the construction of the recurrence coefficients $\beta_k(x), k = 0, 1, \ldots, N - 1$, when $N = 10, 20, 50$ and 100.

Table 2. Approximative values of the condition numbers of $K_N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$x = 1/10$</th>
<th>$x = 1$</th>
<th>$x = 10$</th>
<th>$x = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$10^5$</td>
<td>$10^2$</td>
<td>$&lt; 10$</td>
<td>$&lt; 10$</td>
</tr>
<tr>
<td>20</td>
<td>$10^{38}$</td>
<td>$10^{15}$</td>
<td>$10^9$</td>
<td>$10^5$</td>
</tr>
<tr>
<td>50</td>
<td>$10^{130}$</td>
<td>$10^{81}$</td>
<td>$10^{34}$</td>
<td>$10^{27}$</td>
</tr>
<tr>
<td>100</td>
<td>$10^{304}$</td>
<td>$10^{202}$</td>
<td>$10^{108}$</td>
<td>$10^{74}$</td>
</tr>
</tbody>
</table>

Table 3. Maximal relative errors of the recurrence coefficients $\beta_k(x), k = 1, \ldots, N - 1$, for $N = 50$ and four values of $x$ in different arithmetics

<table>
<thead>
<tr>
<th>$\wp$</th>
<th>$x = 1/10$</th>
<th>$x = 1$</th>
<th>$x = 10$</th>
<th>$x = 25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td></td>
<td></td>
<td>$1.65 \times 10^4$</td>
<td>$2.93 \times 10^{-4}$</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td>$5.21 \times 10^{-6}$</td>
<td>$7.62 \times 10^{-14}$</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td>$8.04 \times 10^{-17}$</td>
<td>$2.63 \times 10^{-24}$</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>$4.68 \times 10^{12}$</td>
<td>$4.31 \times 10^{-26}$</td>
<td>$3.52 \times 10^{-33}$</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>$4.50 \times 10^6$</td>
<td>$7.58 \times 10^{-37}$</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>$6.78 \times 10^{-2}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td></td>
<td>$1.22 \times 10^{-10}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>$3.01 \times 10^{-22}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>$4.28 \times 10^{12}$</td>
<td>$9.81 \times 10^{-30}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>$2.69 \times 10^{3}$</td>
<td>$1.61 \times 10^{-41}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>$1.87 \times 10^{-2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>$1.85 \times 10^{-9}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>$1.01 \times 10^{-18}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Some characteristic values of the maximal relative errors $\text{err}_N(x; \wp)$ obtained for $N = 50$ are presented in Table 3. For example, for getting all coefficients with more than 16
exact decimal digits (i.e., with \( \text{err}_{90}(x; \text{WP}) < 10^{-16} \)) we need the working precision at least \( \text{WP} = 150, 100, \) and \( 50, \) when \( x = 1/10, \ x = 1, \) and \( x \geq 10, \) respectively. When we need coefficients for \( N = 100, \) the corresponding mapping is very sensitive, with extremely high condition numbers (see the last row in Table 2). For example, for small values of \( x \leq 1/10 \) we need the working precision of several hundred, which is practically difficult to implement.

Therefore, in sequel we present an efficient procedure for constructing these coefficients.

3. Transformations of polynomials and quadratures to \([0, 1]\)

Let \( \pi_k(t; x) \) be orthogonal polynomials defined by the recurrence relation (1.2).

According to Theorem 2.2.1 \([12, \text{p. 102}]\), we can consider two sequences of monic polynomials:

\[
1^\circ \quad p_k(z; x) := \pi_{2k}(\sqrt{z}; x), \quad k = 0, 1, \ldots, \text{ which are orthogonal with respect to the weight function}
\]

\[
z \mapsto w_1(z; x) = \frac{w(\sqrt{z}; x)}{\sqrt{z}} = \frac{\exp(-xz)}{\sqrt{z}} \quad \text{on} \quad (0, 1);
\]

\[
2^\circ \quad q_k(z; x) := \pi_{2k+1}(\sqrt{z}; x)/\sqrt{z}, \quad k = 0, 1, \ldots, \text{ which are orthogonal with respect to the weight function}
\]

\[
z \mapsto w_2(z; x) = \sqrt{z} w(\sqrt{z}; x) = \sqrt{z} \exp(-xz) \quad \text{on} \quad (0, 1).
\]

Using these facts, the construction of the quadratures (1.8) and (1.9) can be significantly simplified.

These (monic) polynomials orthogonal on \((0, 1)\) satisfy the following three-term recurrence relations (see Theorem 2.2.12 in \([12, \text{p. 102}]\)),

\[
p_{k+1}(z; x) = (z - a_k)p_k(z; x) - b_kp_{k-1}(z; x), \quad k = 0, 1, \ldots, \tag{3.1}
\]

\[
q_{k+1}(z; x) = (z - c_k)q_k(z; x) - d_kq_{k-1}(z; x), \quad k = 0, 1, \ldots, \tag{3.2}
\]

with \( p_0(z; x) = q_0(z; x) = 1 \) and \( p_{-1}(z; x) = q_{-1}(z; x) = 0. \) The recursion coefficients in (3.1) and (3.2) can be expressed as

\[
a_0 = \beta_1, \quad a_k = \beta_{2k} + \beta_{2k+1}, \quad b_k = \beta_{2k-1}\beta_{2k} \tag{3.3}
\]

and

\[
c_0 = \beta_1 + \beta_2, \quad c_k = \beta_{2k+1} + \beta_{2k+2}, \quad d_k = \beta_{2k}\beta_{2k+1}, \tag{3.4}
\]

respectively, where \( \beta_k \) are coefficients in the three-term recurrence relation (1.2).

Quadrature formulas (1.8) and (1.9) on \((-1, 1)\) can be also connected with the corresponding ones on the interval \((0, 1)\) (cf. \([11], [13], [19], [17]\)).

Let \( P_N \) be a linear space of all algebraic polynomials of degree at most \( N. \) A subset of this space with only even polynomials will be denoted by \( P_N^e. \) We consider now two different cases, one with even \( N, \) and the second one with odd \( N. \)
CASE $N = 2n$. Since the symmetric formula (1.8) is exact for all odd functions and for $f \in \mathcal{P}_{2N-2}$ as a Gaussian rule, in our analysis it is enough to suppose that $f \in \mathcal{P}_{2N-2}$. Then, (1.8) reduces to
\[
\int_{0}^{1} w(t; x) f(t) \, dt = \sum_{k=1}^{n} A_{k}^{(N)} f(\tau_{k}^{(N)}) \quad (f \in \mathcal{P}_{2n-2}),
\]
i.e., after a change of variables $t := \sqrt{y}$, we obtain
\[
\int_{0}^{1} w(\sqrt{y}; x) g(y) \, \frac{dy}{\sqrt{y}} = 2 \sum_{k=1}^{n} A_{k}^{(N)} g(\tau_{k}^{(N)})^{2} \quad (g \in \mathcal{P}_{2n-1}),
\]
where $g(y) := f(\sqrt{y})$. As we can see, (3.5) represents a quadrature of Gaussian type with respect to the weight function $w_{1}(y; x) = y^{-1/2} \exp(-xy)$ on $(0, 1)$,
\[
\int_{0}^{1} y^{-1/2} e^{-xy} g(y) \, dy = \sum_{k=1}^{n} B_{k}^{(n)} g(y_{k}^{(n)}) \quad (g \in \mathcal{P}_{2n-1}),
\]
and parameters of (1.8) and (3.6) are in the following relations
\[
\tau_{k}^{(2n)} = \sqrt{y_{k}^{(n)}}, \quad A_{k}^{(2n)} = \frac{1}{2} B_{k}^{(n)}, \quad k = 1, \ldots, n. \tag{3.7}
\]

CASE $N = 2n + 1$. Suppose again that $f \in \mathcal{P}_{2N-2}$ and $g(z) := f(\sqrt{z})$. Formula (1.9) reduces to
\[
\int_{0}^{1} w(t; x) f(t) \, dt = \frac{1}{2} A_{0}^{(N)} f(0) + \sum_{k=1}^{n} A_{k}^{(N)} f(\tau_{k}^{(N)}) \quad (f \in \mathcal{P}_{4n}),
\]
i.e.,
\[
\int_{0}^{1} w(\sqrt{z}; x) g(z) \, \frac{dz}{\sqrt{z}} = A_{0}^{(N)} g(0) + \frac{2}{n} \sum_{k=1}^{n} A_{k}^{(N)} g(\tau_{k}^{(N)}) \quad (g \in \mathcal{P}_{2n}),
\]
which can be interpreted as the Gauss-Radau quadrature formula
\[
\int_{0}^{1} e^{-xz} g(z) \, \frac{dz}{\sqrt{z}} = C_{0}^{(n)} g(0) + \sum_{k=1}^{n} C_{k}^{(n)} g(z_{k}^{(n)}) \quad (g \in \mathcal{P}_{2n}), \tag{3.8}
\]
and parameters of (1.9) and (3.8) are in the following relations
\[
\tau_{k}^{(2n+1)} = \sqrt{z_{k}^{(n)}}, \quad A_{0}^{(2n+1)} = C_{0}^{(n)}, \quad A_{k}^{(2n+1)} = \frac{1}{2} C_{k}^{(n)}, \quad k = 1, \ldots, n. \tag{3.9}
\]

On the other side it is well-known that the nodes $z_{k}^{(n)}$ in the Gauss-Radau quadrature formula (3.8) are exactly zeros of the polynomial $q_{n}(z; x) = \prod_{k=1}^{n}(z - z_{k}^{(n)})$, which is orthogonal with respect to the weight function (cf. [12, p. 329])
\[
z \mapsto z e^{-xz} = w_{2}(z; x) \quad \text{on} \quad (0, 1),
\]
and the coefficients $c^{(n)}_k$ can be expressed in terms of the Christoffel numbers (weight coefficients) of the corresponding Gaussian formula

$$\int_0^1 w_2(z; x) g(z) \, dz = \sum_{k=1}^n W_k^{(n)} g(z_k^{(n)}) + R_n^G(g) \quad (R_n^G(P_{2n-1}) = 0), \quad (3.10)$$

i.e.,

$$c^{(n)}_0 = \mu_0 - \sum_{k=1}^n c_k^{(n)}, \quad c_k^{(n)} = \frac{W_k^{(n)}}{z_k^{(n)}}, \quad k = 1, \ldots, n, \quad (3.11)$$

where

$$\mu_0 = \int_0^1 e^{-xz} \sqrt{z} \, dz = \sqrt{\frac{\pi}{x}} \text{erf} \left( \sqrt{x} \right).$$

Note that also $\beta_0 = \mu_0$ (see (1.3)). The nodes and the weights in (3.8) can be also obtained by a little modification of the Golub-Welsch algorithm (see Remark 5.1.5 in [12, p. 329]).

4. A stable construction of the Rys polynomials and the corresponding polynomials orthogonal on $(0, 1)$

Instead of constructing the Rys polynomials $\pi_n(t; x)$, i.e., the recurrence coefficients $\beta_n = \beta_n(x)$ in (1.2), in this section we construct the polynomials $p_k(z; x)$ orthogonal on $(0, 1)$ with respect to the weight function $z \mapsto e^{-xz}/\sqrt{z}$, i.e., the coefficients $a_k$ and $b_k$ in their recurrence relation (3.1). In this manner, the influence of numerical instabilities in the process of construction can be significantly reduced. Also, in this way, when we construct Gaussian quadratures, the dimensions of the corresponding Jacobi matrices are halved.

In a similar way, we can also consider the case of the Gauss-Rys quadrature formula (1.7) for odd $N$, i.e., (1.8) when $N = 2n + 1$. In that case we first construct the coefficients $c_k$ and $d_k$ in the recurrence relation (3.2) for the polynomials $q_k(z; x)$ orthogonal on $(0, 1)$ with respect to the weight function $z \mapsto \sqrt{z} e^{-xz}$, and then we construct the Gaussian formula (3.8), i.e., (3.10). Finally, using (3.11) and (3.9) we obtain the parameters in the Gauss-Rys quadrature formula on $(-1, 1)$ for $N = 2n + 1$.

In the sequel we consider only the case with even number of nodes in the Gauss-Rys quadrature formula (1.7).

For constructing the recurrence coefficients in (3.1), we use the method of modified moments developed by Gautschi [4] (see also [12, pp. 160–162]). In order to have the first $n$ coefficients $a_k$ and $b_k$, $k = 0, 1, \ldots, n - 1$, in (3.1), this method needs the first $2n$ modified moments of the weight function $w_1(t; x) = t^{-1/2} \exp(-xt)$ 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 $\{p_k\} = \{p_k(t; x)\}$. We suppose that the polynomials $\phi_k$ are also monic and satisfy a three-term recurrence relation

$$\phi_{k+1}(t) = (t - a_k^M)\phi_k(t) - b_k^M \phi_{k-1}(t), \quad k = 0, 1, \ldots, \quad (4.1)$$

where $\phi_{-1}(t) = 0$ and $\phi_0(t) = 1$, with given coefficients $a_k^M \in \mathbb{R}$ and $b_k^M \geq 0$. 

An efficient computation of parameters in the Rys quadrature formula

49
Since for \( x = 0 \), \( \pi_n(t; 0) \) are monic Legendre polynomials \( \hat{P}_n(t) \) given by (1.5), the corresponding polynomials \( p_k(t; 0) \) orthogonal with respect to the weight function \( w_1(t; 0) = 1/\sqrt{t} \) on \((0, 1)\) are
\[
p_k(t; 0) = \pi_{2k}(\sqrt{t}; 0) = \hat{P}_{2k}(\sqrt{t}),
\]
and therefore we just take these polynomials as \( \phi_k(t) \) for calculating the modified moments
\[
\mu_k^M = \int_0^1 e^{-xt} \sqrt{t} \hat{P}_{2k}(\sqrt{t}) \, dt, \quad k = 0, 1, \ldots, 2n - 1.
\]
(4.2)

In this case for \( \phi_k(t) = \hat{P}_{2k}(\sqrt{t}) \), it is easy to see that the recurrence coefficients \( a_k^M \) and \( b_k^M \) in (4.1) can be expressed in the form
\[
a_k^M = \frac{8k^2 + 4k - 1}{(4k - 1)(4k + 3)} \quad (k \geq 0),
\]
\[
b_k^M = \frac{4k^2(2k - 1)^2}{(4k^2) - (4k - 2)(4k + 1)} \quad (k \geq 1)
\]
and \( b_0^M = 2 \).

2.1. Calculation of modified moments (4.2)

For calculating the integrals (4.2) we use the following formula
\[
\int_0^1 x^{a-1} e^{-px^2} P_{2n} \left( \frac{x}{a} \right) \, dx = \frac{(-1)^n a^{2n} (1 - \alpha + \varepsilon)/2)_n}{2((\alpha + \varepsilon)/2)_n+1} \quad (\varepsilon > 0, \quad a > 0)
\]
\[
\times {}_2F_2 \left( \begin{array}{c} \frac{\alpha + 1}{2}, \frac{1 + \alpha - \varepsilon}{2} \\ \frac{\alpha + \varepsilon}{2} + n + 1; \; \frac{\alpha}{2} + n + 1; \; -a^2 p \end{array} \right),
\]
which holds for \( \varepsilon = 0 \) or \( 1; \ a > 0, \Re \alpha > -\varepsilon \), where \( P_n \) is the Legendre polynomial. (We note here that there is a mistake in [24, p. 429, Eq. 9] for this formula. Namely, the denominator is given as \( 2((\alpha + \varepsilon)/2)_n \).)

First we take \( a = 1, \varepsilon = 0, \quad n = k, \quad x = \sqrt{t}, \) and then \( p = x \), so that we get
\[
\int_0^1 t^{a/2-1} e^{-xt} P_{2k}(\sqrt{t}) \, dt = \frac{(-1)^k ((1 - \alpha)/2)_k}{(\alpha/2)_{k+1}} \quad (\alpha > 0)
\]
\[
\times {}_2F_2 \left( \begin{array}{c} \frac{\alpha + 1}{2}, \frac{1 + \alpha}{2} - k, \; \frac{2 + \alpha}{2} + k; -x \end{array} \right),
\]
which is possible to express in terms of the Meijer \( G \)-function, defined by (cf. [1, p. 207], [23])
\[
G_{\alpha, n}^{m,n} \left( \begin{array}{c} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{array} \right) = G_{\alpha, n}^{m,n} \left( \begin{array}{c} a_1, \ldots, a_n; a_{n+1}, \ldots, a_p \\ b_1, \ldots, b_m; b_{m+1}, \ldots, b_q \end{array} \right)
\]
\[
= \frac{1}{2\pi i} \int_L \frac{\prod_{\nu=1}^m \Gamma(b_{\nu} - s) \prod_{\nu=1}^n \Gamma(1 - a_{\nu} + s)}{\prod_{\nu=m+1}^{n+1} \Gamma(1 - b_{\nu} + s) \prod_{\nu=1}^{n+1} \Gamma(a_{\nu} - s)} \, z^s \, ds,
\]
where an empty product is interpreted as 1, and parameters $a_{\nu}$ and $b_{\nu}$ are such that no pole of $\Gamma(b_{\nu} - s)$, $\nu = 1, \ldots, m$, coincides with any pole of $\Gamma(1-b_{\mu}+s)$, $\mu = 1, \ldots, n$. Here, $m$ and $n$ are such that $1 \leq m \leq q$ and $1 \leq n \leq p$. Roughly speaking, the contour $L$ separates the poles of functions $\Gamma(b_1 - s), \ldots, \Gamma(b_m - s)$ from the poles of $\Gamma(1-a_1 + s), \ldots, \Gamma(1-a_n + s)$.

Using the known relation

$$2F_2(a_1, a_2; b_1, b_2; z) = \frac{\Gamma(b_1)\Gamma(b_2)}{\Gamma(a_1)\Gamma(a_2)} G_{2,3}^{1,1} \left(-z \bigg| \begin{array}{c} 1-a_1, 1-a_2 \\ 0, 1-b_1, 1-b_2 \end{array} \right),$$

we obtain that

$$2F_2 \left( \frac{\alpha}{2}, \frac{1+\alpha}{2}; \frac{1+\alpha}{2}; k, \frac{2+\alpha}{2} ; -x \right) = \frac{\Gamma \left( \frac{1+\alpha}{2} - k \right) \Gamma \left( \frac{2+\alpha}{2} + k \right)}{\Gamma \left( \frac{\alpha}{2} \right) \Gamma \left( \frac{1+\alpha}{2} \right)} \times G_{2,3}^{1,1} \left( x \bigg| \begin{array}{c} 1-k, \frac{1-\alpha}{2} \\ 0, \frac{1-\alpha}{2} + k, -\frac{\alpha}{2} - k \end{array} \right).$$

Since $\Gamma(z) = (-1)^k(1-z)_k\Gamma(z - k)$, putting $z = (1+\alpha)/2$, we have that

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

so that

$$\int_0^1 t^{\alpha/2 - 1} e^{-xt} P_{2k}(\sqrt{t}) dt = G_{2,3}^{1,1} \left( x \bigg| \begin{array}{c} 1-k, \frac{1-\alpha}{2} \\ 0, \frac{1-\alpha}{2} + k, -\frac{\alpha}{2} - k \end{array} \right).$$

Now, using (1.5) and letting $\alpha \to 1$, we find the modified moments

$$\mu_k^M = \int_0^1 e^{-xt} P_{2k}(\sqrt{t}) dt = \frac{4^k (\frac{4}{2k})_k}{(\frac{1}{2k})_k} \times G_{2,3}^{1,1} \left( x \bigg| \begin{array}{c} \frac{3}{2}, 0 \\ 0, k, -\frac{1}{2} - k \end{array} \right).$$

(4.4)

After some transformations again to hypergeometric functions, (4.4) can be successively reduced to

$$\mu_k^M = \frac{4^k (\frac{4}{2k})_k}{(\frac{1}{2k})_k} \times F_1 \left( k + \frac{3}{2}, 2k + 3; -x \right) = \frac{4^k (\frac{4}{2k})_k}{(\frac{1}{2k})_k} \times \Gamma \left( \frac{2k + \frac{3}{2}}{2} \right) \frac{\Gamma \left( \frac{3}{2} \right)}{\Gamma \left( k + 1 \right) \Gamma \left( k + \frac{3}{2} \right)} \int_0^1 e^{-xt}(1-t)^k t^{-1/2} dt = \frac{2(-1)^k (4x)^k}{k! (\frac{4}{2k})_k} \int_0^{\pi/2} e^{-x \sin^2 \theta} \cos^{2k+1} \theta \sin^{2k} \theta d\theta.$$
i.e.,

\[ \mu_k^M = \frac{(-1)^k x^k}{k! (4k)} e^{-\frac{x^2}{2}} \int_0^\pi e^{\frac{x}{2} \cos \theta} \sin^{2k} \theta \cos \frac{\theta}{2} d\theta. \]

Typical graphics of the integrand \( t \mapsto g_k(t; x) = \exp\left(\frac{1}{2} x \cos \theta\right) \sin^{2k} \theta \cos(\theta/2) \) in the last integral are displayed in Figure 1 for \( k = 10 \) and four different values of \( x \). As we can see for small values of \( x \) the corresponding graphics almost coincide. This is more pronounced for larger values of \( x \), when \( k \) increases! For \( x = 1 \) and \( k = 10, 50, \) and 100, the corresponding graphics are presented in log-scale in Figure 2.

**Figure 1.** Graphics of the integrand \( t \mapsto g_{10}(t; x) \) when \( t \in (0, \pi) \) for \( x = 0, 1, 5, \) and 12.

**Figure 2.** Graphics of the integrand \( t \mapsto g_k(t; x) \) for \( k = 10, 50, \) and 100, when \( x = 1 \) (left) and \( x = 20 \) (right).
Due to such a behavior of the integrand $g_k(t; k)$, for the calculation of these integrals, i.e., the moments $\mu^M_k, k = 0, 1, \ldots, 2n - 1$, we can use the standard command \texttt{NIntegrate} in \textsc{Mathematica}, with the options

\begin{verbatim}
Method -> "DoubleExponential" and WorkingPrecision -> WP,
\end{verbatim}

where \texttt{WP} is a given working precision.

2.2. Calculation of the recurrence coefficients in (3.1)

For constructing the first $n$ recurrence coefficients $a_k$ and $b_k$ in (3.1), we need the first $2n$ modified moments $\mu^M_k, k = 0, 1, \ldots, 2n - 1$. It can be realised by using our \textsc{Mathematica} package \texttt{OrthogonalPolynomials} (see [2] and [18]), with the following commands:

\begin{verbatim}
<< orthogonalPolynomials'

akM = Table[(8k^2+4k-1)/((4k-1)(4k+3)), {k,0,2n-1}];

bkM = Table[If[k==0,2,
4k^2(2k-1)^2/((4k-3)(4k-1)^2(4k+1))], {k,0,2n-1}];

mmom = Table[(-1)^kx^k/(k!Binomial[4k,2k]) Exp[-x/2]
NIntegrate[Exp[x/2Cos[t]]Cos[t/2]Sin[t]^(2k),{t,0,Pi},
Method -> "DoubleExponential",WorkingPrecision->WP],
{k,0,2n-1}];

{ak,bk} = aChebyshevAlgorithmModified[mmom,akM,bkM,
WorkingPrecision -> WP];
\end{verbatim}

We only should specify $n$ and $x$, as well as the working precision \texttt{WP}. Note that \texttt{akM} and \texttt{bkM} are sequences $a^M_k$ and $b^M_k$ in the recurrence relation (4.1) given before by (4.3). The command

\begin{verbatim}
aChebyshevAlgorithmModified
\end{verbatim}

uses the sequence of modified moments \texttt{mmom} and these coefficients to produce the desired recurrence coefficients $a_k$ and $b_k$ in (3.1). They are here represented by the sequences $a_k$ and $b_k$, respectively.

For example, for $n = 100$ and $x = 1$ (with \texttt{WP=30}) we obtain the first 100 recurrence coefficients $a_k$ and $b_k$, and the possibility to construct all Gauss-Rys quadratures up to 200 nodes. The complete procedure is very fast and stable (without loss of digits when $x \leq 12$). In the following, we list only first 40 coefficients with 28 decimal digits to save space.

\begin{table}[h]
\centering
\caption{Recursion coefficients $a_k$ and $b_k$, $k = 0, 1, \ldots, 39$}
\begin{tabular}{llll}
\hline
$k$ & $a_k$ & $b_k$ \\
\hline
0 & 0.2537041018036844625448723502 & 1.493648265624854050798934872 \\
1 & 0.53737923178183433311648092673 & 0.06989448323719686660104213655 \\
\hline
\end{tabular}
\end{table}
Now, we will analyze the obtained numerical results in our example when \( n = 100 \) and the parameter \( x \) runs over \([0, 25]\).

Let \( \hat{a}_k \) and \( \hat{b}_k \), \( k = 0, 1, \ldots, n - 1 \), be exact values of the desired recurrence coefficients in (3.1), and \( a_k \) and \( b_k \), \( k = 0, 1, \ldots, n - 1 \), be their numerical values obtained using our procedure with the working precision \( \text{WP} \).

With \( \text{err}_n(\text{WP}) \) we denote the maximal relative error in the recurrence coefficients \( a_k \) and \( b_k \).
and \(b_k, k = 0, 1, \ldots, n - 1,\)

\[
\text{err}_n(WP) = \max_{0 \leq k \leq n-1} \left\{ \left| \frac{a_k - \hat{a}_k}{\hat{a}_k} \right|, \left| \frac{b_k - \hat{b}_k}{\hat{b}_k} \right| \right\}.
\]

Notice that for calculating this maximal relative error in recursive coefficients we need the exact coefficients \(\hat{a}_k\) and \(\hat{b}_k\), whose values can be well approximated with ones obtained by some better precision \(WP_1 > WP\). In our example (\(n = 100\)) we take \(WP_1 = 2WP\).

Using two different arithmetics, \(WP = 30\) and the standard double precision arithmetic (\(WP = MP\)), we obtain the maximal relative errors of the recurrence coefficients \(\text{err}_n(WP)\) for \(x = 13, 15, 20,\) and \(25\), which are given in Table 5.

<table>
<thead>
<tr>
<th>WP</th>
<th>(x = 13)</th>
<th>(x = 15)</th>
<th>(x = 20)</th>
<th>(x = 25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>(1.0 \times 10^{-29})</td>
<td>(1.20 \times 10^{-28})</td>
<td>(4.72 \times 10^{-26})</td>
<td>(4.84 \times 10^{-24})</td>
</tr>
<tr>
<td>MP</td>
<td>(7.0 \times 10^{-15})</td>
<td>(2.60 \times 10^{-13})</td>
<td>(7.14 \times 10^{-11})</td>
<td>(3.43 \times 10^{-9})</td>
</tr>
</tbody>
</table>

As we can see from Table 5, for \(x = 13, 15, 20\) and \(25\) we lose about one, two, four and six decimal digits, respectively, but for \(x \leq 12\), as we mentioned before, the loss of digits does not exist. Thus, in general, the accuracy of the obtained results (here, the recursion coefficients \(a_k\) and \(b_k\)) depends on the working precision, but also on the condition number of the mapping in the method of construction.

In our example, the method for \(x \leq 12\) is well-conditioned, because its condition number is near 1. But, for larger \(x\) this condition number increases exponentially like \(10^m\) (for example, for \(x = 25\) it is about \(10^6\), i.e., \(m = 6\)). Roughly speaking, if we need the accuracy of \(\ell\) decimal digits in the recurrence coefficients \(a_k\) and \(b_k\) for each \(k < n\), then we must use \(WP = \ell + m\) (cf. [16]).

2.3. Recursion coefficients \(a_k\) and \(b_k\) as functions of \(x\)

The procedure for calculating recurrence coefficients given in the previous subsection is very fast, so that for a given \(n\) and a finite set \(X\) of some selected values of \(x\) in an interval, we can calculate the recurrence coefficients \(a_k = a_k(x)\) and \(b_k = b_k(x), k = 0, 1, \ldots, n - 1,\) for each \(x \in X\).

Following Shizgal [28], we take

\[
X = \left\{ x_\nu = \frac{\nu}{10} \mid \nu = 0, 1, \ldots, 250 \right\}.
\]

After finding \(a_k(x_\nu)\) and \(b_k(x_\nu), k = 0, 1, \ldots, n - 1,\) for each point \(x = x_\nu\), we use these values to construct the corresponding interpolating functions for each of these coefficients, in
notations $\tilde{a}_k(x)$ and $\tilde{b}_k(x)$, which can be realized very easy in MATHEMATICA. In the sequel, these interpolating functions will be denoted simply without the tilda-symbol.

Figure 3. The coefficients $x \mapsto a_k(x)$ for $0 \leq k \leq 7$

Figure 4. The coefficients $x \mapsto a_k(x)$ for $8 \leq k \leq 13$

Graphics of the coefficients $x \mapsto a_k(x)$ in the recurrence relation (3.1) for $k = 0, 1, \ldots, 7$ are presented in Figure 3, and in Figure 4 ones for $k = 8, \ldots, 13$. 
Similarly, graphics of $x \mapsto b_k(x)$ on $[0, 25]$ are presented in Figures 5 and 6.

**Figure 5.** The coefficients $b_k(x)$ for $1 \leq k \leq 7$

**Figure 6.** The coefficients $x \mapsto b_k(x)$ for $8 \leq k \leq 13$
2.4. Recursion coefficients $\beta_k(x)$ in (1.2)

The coefficients $\beta_k(x)$ in the recurrence relation (1.2) can be obtained very easy by the coefficients $a_k(x)$ and $b_k(x)$, thanks to the relation (3.3). Namely,

$$\beta_0(x) = \sqrt{\frac{\pi}{x}} \text{erf}(\sqrt{x}), \quad \beta_1(x) = a_0(x),$$

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

Graphics of the coefficients $x \mapsto \beta_k(x), k = 1, \ldots, 12$, are displayed in Figure 7.

![Figure 7. The coefficients $x \mapsto \beta_k(x)$ for $1 \leq k \leq 12$](image)

In Figure 8 we present in details the graphics of these coefficients $x \mapsto \beta_k(x), k = 1, \ldots, 12$, from where we see their behavior.

In some cases, by using our MATHEMATICA package OrthogonalPolynomials we can also obtain the recurrence coefficients in symbolic form, taking the option Algorithm $\rightarrow$ Symbolic in the command aChebyshevAlgorithm. Often, however, the obtained expressions for higher $k$ become very complicated and useless. For example, taking $N = 7$ and replacing the last two lines in the MATHEMATICA code in Section 2 by

```math
momNx=mom[7,x];
{alpha,beta}=aChebyshevAlgorithm[momNx,Algorithm->Symbolic]
```

we obtain the expressions for the coefficients $\beta_k(x), k = 0, 1, \ldots, 6$. 

The first four of them are

\[
\beta_0(x) = \frac{\sqrt{\pi} \text{erf}(\sqrt{x})}{\sqrt{x}} , \quad \beta_1(x) = \frac{\sqrt{\pi} - 2\Gamma\left(\frac{3}{2}, x\right)}{2\sqrt{\pi} x \text{erf}(\sqrt{x})}, \\
\beta_2(x) = \frac{\text{erf}(\sqrt{x}) \left[3\pi - 4\sqrt{\pi} \Gamma\left(\frac{5}{2}, x\right)\right]}{2\sqrt{\pi} x \text{erf}(\sqrt{x})} \left[\sqrt{\pi} - 2\Gamma\left(\frac{7}{2}, x\right)\right] - \left[\sqrt{\pi} - 2\Gamma\left(\frac{5}{2}, x\right)\right]^2, \\
\beta_3(x) = \frac{\sqrt{\pi} \text{erf}(\sqrt{x}) \left\{3\pi - 15\sqrt{\pi} \Gamma\left(\frac{5}{2}, x\right) + 4 \left[3\sqrt{\pi} - 2\Gamma\left(\frac{7}{2}, x\right)\right] \Gamma\left(\frac{7}{2}, x\right) - G(x)\right\}}{x \left[\sqrt{\pi} - 2\Gamma\left(\frac{7}{2}, x\right)\right] \left[\text{erf}(\sqrt{x}) \left[3\pi - 4\sqrt{\pi} \Gamma\left(\frac{7}{2}, x\right)\right] - \left[\sqrt{\pi} - 2\Gamma\left(\frac{5}{2}, x\right)\right]^2\right]} ,
\]

where \( G(x) = 4 \left[\sqrt{\pi} - 2\Gamma\left(\frac{5}{2}, x\right)\right] \Gamma\left(\frac{5}{2}, x\right) \), but expressions for \( \beta_4(x), \beta_5(x), \) and \( \beta_6(x) \) are very complicated.

![Figure 8. The coefficients \( x \mapsto \beta_k(x) \) for \( 1 \leq k \leq 12 \)](image)

The obtained coefficients in this symbolic form are not stable for numerical calculations for small values of \( x \), because their numerators and denominators tend to zero when \( x \to 0 \). However, the behavior of \( x \mapsto \beta_k(x) \) near the origin can be seen from the following series expansions:

\[
\beta_0(x) = 2 - \frac{2x}{3} + \frac{x^2}{5} - \frac{x^3}{21} + \frac{x^4}{108} - \frac{x^5}{660} + O\left(x^6\right), \\
\beta_1(x) = \frac{1}{3} - \frac{4x}{45} + \frac{8x^2}{945} + \frac{16x^3}{14175} - \frac{32x^4}{93555} - \frac{1472x^5}{638512875} + O\left(x^6\right),
\]
\[
\beta_2(x) = \frac{4}{15} + \frac{32}{1375} x - \frac{272}{23625} x^2 - \frac{10496}{27286875} x^3 + \frac{2283968}{5320940625} x^4
\]
\[
- \frac{4322816}{558698765625} x^5 + O(x^6),
\]
\[
\beta_3(x) = \frac{9}{35} + \frac{4x}{1225} + \frac{3512}{1414875} x^2 - \frac{211408}{21589375} x^3 - \frac{2614112}{67595653125} x^4
\]
\[
+ \frac{20139796928}{1327240649109375} x^5 + O(x^6),
\]
\[
\beta_4(x) = \frac{16}{63} + \frac{256x}{218295} + \frac{61888x}{178783605} x^2 + \frac{125433856x^3}{619485191325} x^3 - \frac{8360266496x^4}{132693727981815}
\]
\[
- \frac{148533647084392x^5}{567832952809381384125} + O(x^6),
\]
\[
\beta_5(x) = \frac{25}{99} + \frac{500x}{891891} + \frac{9640x^2}{88297209} + \frac{344399920x^3}{13522982449977} + \frac{315524646176x^4}{25436729988406737}
\]
\[
- \frac{737067432518720x^5}{22915950046556293633} + O(x^6),
\]
\[
\beta_6(x) = \frac{36}{143} + \frac{32x}{102245} + \frac{242576x^2}{5219709495} + \frac{171547392x^3}{2363584496525} + \frac{8755594304x^4}{6084056849405535}
\]
\[
+ \frac{108699755410385408x^5}{17859382075926131188875} + O(x^6),
\]

Note that the free terms in these expansions are, in fact, the recurrence coefficients for the monic Legendre polynomials, \( \beta_k = k^2/(4k^2 - 1), \) \( k \geq 1 \) (see (1.4)).

**Remark 4.1.** From Figures 7 and 8 we can observe that the recurrence coefficients \( \beta_k(x) \) converge to the constant limit value \( 1/4 \) as \( k \to \infty \), which is in agreement with (1.6).

### 2.4. Construction of the Gaussian quadrature (3.6)

The obtained coefficients \( a_k \) and \( b_k, \) \( k = 0, 1, \ldots, n - 1, \) in (3.1) enable us to simply construct the parameters in the Gaussian formula (3.6) for each number of nodes less or equal to \( n \) (in our example \( n = 100 \)).

It is well known that the nodes \( y_k = y_k^{(n)} = y_k^{(n)}(x), \) \( k = 1, \ldots, n, \) in the Gaussian formula (3.6), i.e.,

\[
\int_0^1 e^{-xy} g(y) \, dy = \sum_{k=1}^n B_k^{(n)} g(y_k^{(n)}), \quad (g \in \mathcal{P}_{2n-1}),
\]

are eigenvalues of the symmetric tridiagonal *Jacobi matrix*, of order \( n \) associated with the
An efficient computation of parameters in the Rys quadrature formula

weight function \( y \mapsto w_1(y; x) \),

\[
J_n(w_1(\cdot; x)) = \begin{bmatrix}
    a_0(x) & \sqrt{b_1(x)} & \cdots & \sqrt{b_n(x)} \\
    \sqrt{b_1(x)} & a_1(x) & \cdots & \sqrt{b_{n-1}(x)} \\
    \sqrt{b_2(x)} & a_2(x) & \cdots & \sqrt{b_{n-2}(x)} \\
    \sqrt{b_{n-1}(x)} & \cdots & \sqrt{b_{n-2}(x)} & a_{n-1}(x) \\
\end{bmatrix}, \quad (4.5)
\]

and the weight coefficients (Christoffel numbers) \( B_k = B_k^{(n)} = B_k^{(n)}(x) \), \( k = 1, \ldots, n \), are given by \( B_k = b_0(x) v_k^1, k = 1, \ldots, n \), where \( v_k \) is the first component of the normalized eigenvector \( v_k = [v_{k,1}, \ldots, v_{k,n}]^T \) corresponding to the eigenvalue \( y_k \). \( J_n(w_1(\cdot; x))v_k = y_k v_k \), where \( v_k^T v_k = 1, k = 1, \ldots, n \).

![Figure 9](image_url) Positive nodes \( x \mapsto \tau_k(x), 0 \leq x \leq 25 \), of the 20-point Gauss-Rys quadrature

This eigenvalue problem can be easily solved by the Golub-Welsch procedure [8], which is implemented in several packages including Gautschi’s SOPQ in MATLAB and our MATLAB package OrthogonalPolynomials.

Thus, in our example, we can calculate Gaussian parameters (nodes and weights) for each \( n (\leq 100) \), using our MATLAB package OrthogonalPolynomials, with the command aGaussianNodesWeights. Then the parameters in the Gauss-Rys quadrature formula (1.7) for even \( N = 2n \), i.e.,

\[
\int_{-1}^{1} w(t; x) f(t) \, dt = \sum_{k=1}^{n} A_k(x) [f(\tau_k(x)) + f(-\tau_k(x))] + R_{2n}(f), \quad (4.6)
\]
are given by
\[ \tau_k(x) = \tau_k^{(2n)}(x) = \sqrt{y_k^{(n)}(x)}, \quad A_k(x) = A_k^{(2n)}(x) = \frac{1}{2} B_k^{(n)}(x), \quad k = 1, \ldots, n. \]

In this way, following our example from Subsection 2.2, we are able to construct the Gauss-Rys quadrature formulas (4.6), with even number of nodes up to \(2n \leq 200\).

![Figure 10. The weights \(x \mapsto A_k(x), 0 \leq x \leq 25\), of the 20-point Gauss-Rys quadrature](image)

The nodes \(\tau_k(x)\) and \(A_k(x)\) for the 20-point Gauss-Rys quadrature formulas are presented in Figures 9 and 10, respectively. Graphics of the weights are given in the log-scale.

The Gauss-Rys quadratures for odd nodes \((N = 2n + 1)\) can be obtained in a similar way, constructing the polynomials \(q_k(z;x) := \pi_{2k+1}(\sqrt{z};x)/\sqrt{z}, k = 0, 1, \ldots\), which are orthogonal on \((0, 1)\) with respect to the weight function \(z \mapsto w_2(z;x) = \sqrt{z} \exp(-xz)\), and the corresponding Gauss-Radau quadratures on \((0, 1)\).

**Acknowledgement.** The work of the author was supported in part by the Serbian Academy of Sciences and Arts (Φ-96) and by the Serbian Ministry of Education, Science and Technological Development (No. #OI 174015).

**REFERENCES**

An efficient computation of parameters in the Rys quadrature formula


