

Special Classes of Orthogonal Polynomials and Corresponding Quadratures of Gaussian Type

*Gradimir V. Milovanović*¹, *Aleksandar S. Cvetković*²

Presented at 6th International Conference “TMSF’ 2011”

In the first part of this survey paper we present a short account on some important properties of orthogonal polynomials on the real line, including computational methods for constructing coefficients in the fundamental three-term recurrence relation for orthogonal polynomials, and mention some basic facts on Gaussian quadrature rules. In the second part we discuss our MATHEMATICA package `OrthogonalPolynomials` (see [2]) and show some applications to problems with strong nonclassical weights on $(0, +\infty)$, including a conjecture for an oscillatory weight on $[-1, 1]$. Finally, we give some new results on orthogonal polynomials on radial rays in the complex plane.

MSC 2010: 33C47, 42C05, 41A55, 65D30, 65D32

Key Words: orthogonal polynomials, weight function, measure, moments, three-term recurrence relation, Gaussian quadratures, nodes, weights, software

1. Introduction to orthogonal polynomials on \mathbb{R}

The main concepts in the theory of orthogonal polynomials can be found in [16], [1], [5], [10]. Suppose $d\mu(t)$ is a positive measure on \mathbb{R} with finite or unbounded support, for which all moments $\mu_k = \int_{\mathbb{R}} t^k d\mu(t)$ exist and are finite. Then the inner product

$$(p, q) = \int_{\mathbb{R}} p(t)q(t)d\mu(t)$$

is well defined for any polynomials $p, q \in \mathcal{P}$ and gives rise to a unique system of monic orthogonal polynomials $\pi_k(t) \equiv \pi_k(d\mu; t) = t^k +$ terms of lower degree, $k = 0, 1, \dots$, such that

$$(\pi_k, \pi_n) = \|\pi_n\|^2 \delta_{kn} = \begin{cases} 0, & n \neq k, \\ \|\pi_n\|^2, & n = k. \end{cases}$$

Because of the property $(tf, g) = (f, tg)$, these polynomials satisfy the three-term recurrence relation

$$\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \dots, \quad (1)$$

with $\pi_0(t) = 1$ and $\pi_{-1}(t) = 0$, where $(\alpha_k) = (\alpha_k(d\mu))$ i $(\beta_k) = (\beta_k(d\mu))$ are sequences of recursion coefficients which depend on the measure $d\mu$. The coefficient β_0 may be arbitrary, but it is convenient to define it by $\beta_0 = \mu_0 = \int_{\mathbb{R}} d\mu(t)$. Unfortunately, these coefficients are known explicitly only for some narrow classes of orthogonal polynomials. For example, if the measure $d\mu$ is absolutely continuous, i.e., $d\mu(t) = w(t)dt$, and its weight function w satisfies Pearson's differential equation we have the so-called *classical* orthogonal polynomials (Jacobi, the generalized Laguerre, and Hermite polynomials). The recursion coefficients for such polynomials are known explicitly. For orthogonal polynomials for which these coefficients are not known we use the term *strong non-classical* polynomials.

The concept of orthogonality can be introduced also via a linear moment functional \mathcal{L} for which is $\mathcal{L}(t^k) = \mu_k$, $k \in \mathbb{N}_0$. In that case $\{\pi_n\}_{n \in \mathbb{N}_0}$ are called orthogonal polynomials with respect to the moment functional \mathcal{L} if

- $\deg \pi_n(t) = n$,
- $\mathcal{L}(\pi_n(t)\pi_m(t)) = 0$, $n \neq m$,
- $\mathcal{L}(\pi_n^2(t)) \neq 0$.

If the sequence of orthogonal polynomials exists for a given linear functional \mathcal{L} , then \mathcal{L} is called quasi-definite (regular) linear functional. The necessary and sufficient conditions for the existence of orthogonal polynomials $\{\pi_n\}$ with respect to the linear functional \mathcal{L} are that for each $n \in \mathbb{N}$ the Hankel determinants

$$\Delta_n = \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & & \mu_n \\ \vdots & & & \\ \mu_{n-1} & \mu_n & & \mu_{2n-2} \end{vmatrix} \neq 0.$$

If $\mathcal{L}(\pi_n(t)^2) > 0$, then such a functional \mathcal{L} is called positive definite. In that case, we can define $(p, q) = \mathcal{L}(p(t)\bar{q}(t))$, so that the orthogonality with respect to the moment functional \mathcal{L} is consistent with the standard definition of orthogonality with respect to an inner product. For orthogonality on \mathbb{R} we can put

$$\mathcal{L}(p(t)) = \int_{\mathbb{R}} p(t)d\mu(t) = \int_{\mathbb{R}} p(t)w(t) dt.$$

The coefficients in the three-term recurrence relation can be expressed in terms of Hankel determinants (or by Darboux’s formulae) as:

$$\alpha_k = \frac{\Delta'_{n+1}}{\Delta_{n+1}} - \frac{\Delta'_n}{\Delta_n} \left(= \frac{\mathcal{L}(t\pi_k(t)^2)}{\mathcal{L}(\pi_k(t)^2)} \right), \quad \beta_k = \frac{\Delta_{n-1}\Delta_{n+1}}{\Delta_n^2} \left(= \frac{\mathcal{L}(\pi_k(t)^2)}{\mathcal{L}(\pi_{k-1}(t)^2)} \right), \quad (2)$$

where Δ'_n denotes the determinant obtained from Δ_n when the last column $[\mu_{n-1} \ \mu_n \ \dots \ \mu_{2n-2}]^T$ is replaced by $[\mu_n \ \mu_{n+1} \ \dots \ \mu_{2n-1}]^T$.

Associated with the three-term recurrence relation is the Jacobi matrix

$$J(d\mu) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & \mathbf{0} \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \\ & \sqrt{\beta_2} & \alpha_2 & \ddots \\ \mathbf{0} & & \ddots & \ddots \end{bmatrix}.$$

Its leading principal minor matrix of order n will be denoted by $J_n(d\lambda)$.

In the constructive theory of (strong non-classical) orthogonal polynomials the basic computational problem is the following: *For a given measure $d\mu$ and for given $n \in \mathbb{N}$, generate the first coefficients $\alpha_k(d\mu)$ and $\beta_k(d\mu)$, $k = 0, 1, 2, \dots, n - 1$.* In numerical construction three approaches are well-known: *method of moments, Stieltjes procedure, and Lanczos algorithm.*

1.1. Method of (modified) moments. The recursion coefficients α_k and β_k in (1) can be computed from well-known formulae (2) in terms of Hankel-type determinants, but in that case an excessive complexity and an extreme numerical instability are appeared. To avoid these problems, one can attempt to use the so-called modified moments $m_k = \int_{\mathbb{R}} p_k(t)d\mu(t)$, $k = 0, 1, 2, \dots$, where p_k are some monic polynomials of degree k “close” in some sense to the desired polynomials π_k . Usually, we suppose that p_k satisfy a three-term recurrence relation of the form (1), with recursion coefficients $a_k (\in \mathbb{R})$ and $b_k (\geq 0)$ (instead of α_k and β_k). Then there is a unique map $\varrho: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ that takes the first $2n$ modified moments into the desired n recurrence coefficients α_k and β_k , i.e., $[m_k]_{k=0}^{2n-1} \mapsto [\alpha_k, \beta_k]_{k=0}^{n-1}$. An algorithm for realizing this map (modified Chebyshev algorithm) was given by Gautschi [3] (see also [5, pp. 76–78] and [10, pp. 160–162]).

For $a_k = b_k = 0$ we have $p_k(t) = t^k$ and the moments m_k reduce to the standard moments μ_k . This map for standard moments $[\mu_k]_{k=0}^{2n-1} \mapsto [\alpha_k, \beta_k]_{k=0}^{n-1}$ is severely ill-conditioned when n is large. Namely, it is very sensitive with respect to small perturbations in the moment information (in the first $2n$ moments). An analysis of such maps in detail can be found in the book [5, Chap. 2].

Using modified moments the corresponding map can become remarkably well-conditioned, especially for measures with a finite support.

2.2. Discretization methods. The basic idea for these methods is an approximation of the given measure $d\mu$ by a discrete N -point measure, usually using an appropriate quadrature rule,

$$d\mu(t) \approx d\mu_N(t) = \sum_{k=1}^N w_k \delta(t - x_k), \quad w_k > 0,$$

where δ is the Dirac delta function. Thereafter, the desired recursion coefficients are approximated by those of the discrete measure

$$\alpha_k(d\mu) \approx \alpha_k(d\mu_N), \quad \beta_k(d\mu) \approx \beta_k(d\mu_N).$$

For sufficiently large N , the approximate coefficients are computed by the *discretized Stieltjes-Gautschi procedure* or *Lanczos algorithm*. The corresponding inner product is a finite sum

$$(p, q)_N = \int_{\mathbb{R}} p(t)q(t) d\mu_N(t) = \sum_{k=1}^N w_k p(x_k)q(x_k).$$

In the first procedure, the computation of the recursive coefficients goes over Darboux's formulae for $k \leq n - 1$, where N is taken such that $N \gg n$. An alternative approach is the Lanczos algorithm, which is based on ideas of Lanczos and Rutishauser (for details see [5, pp. 97–98]).

2. Quadratures of Gaussian type

The n -point Gaussian quadrature formula

$$\int_{\mathbb{R}} f(t) d\mu(t) = \sum_{\nu=1}^n A_{\nu} f(\tau_{\nu}) + R_n(f),$$

is exact on the space of polynomials of degree at most $2n - 1$, i.e., $R_n(f) = 0$ for each $f \in \mathcal{P}_{2n-1}$. Numbers τ_{ν} are called nodes, and A_{ν} weights or Christoffel numbers. Node polynomial $\pi_n(t) = (t - \tau_1) \cdots (t - \tau_n)$ is orthogonal with respect to the measure $d\mu(t)$. The characterization of the Gauss-Christoffel quadratures via an eigenvalue problem for the Jacobi matrix has become the basis of current methods for generating this kind of quadratures. The most popular of them is one due to Golub and Welsch (see [8]). Their method is based on determining

the eigenvalues and the first components of the eigenvectors of the symmetric tridiagonal Jacobi matrix $J_n(d\mu)$.

Theorem 1. *The nodes τ_k in the Gauss-Christoffel quadrature rule, with respect to a positive measure $d\mu$, are the eigenvalues of the n -th order Jacobi matrix $J_n(d\mu)$, constructed by coefficients in the three-term recurrence relation for the monic orthogonal polynomials $\pi_n(d\mu; \cdot)$. The weights A_k are given by*

$$A_k = \beta_0 v_{k,1}^2, \quad k = 1, \dots, n,$$

where $\beta_0 = \mu_0 = \int_{\mathbb{R}} d\mu(t)$ and $v_{k,1}$ is the first component of the normalized eigenvector \mathbf{v}_k corresponding to the eigenvalue x_k ,

$$J_n(d\mu)\mathbf{v}_k = x_k\mathbf{v}_k, \quad \mathbf{v}_k^T \mathbf{v}_k = 1, \quad k = 1, \dots, n.$$

The Golub and Welsch procedure [8] was implemented in several programming packages including the most known ORTHPOL developed by Gautschi [4] in 1994.

3. Software

The previous mentioned package ORTPOL, written in FORTRAN, is a package of routines for generating orthogonal polynomials and Gauss-type quadrature rules and it was a yeast for a progress in this subject. Package OPQ was developed also by Gautschi in 2004, is written in MATLAB (see [6]) and is companion for the book [5]. There is also SOPQ written by the same author which implements some symbolic possibilities in MATLAB.

In this section we give a short account of our MATHEMATICA package `OrthogonalPolynomials` (see [2]). Package performs the construction of orthogonal polynomials and quadrature formulae. Also, this package has implemented almost all classes of orthogonal polynomials studied up to date.

3.1. Implementation of some symbolic algorithms. Chebyshev algorithm can be represented as the mapping of the sequence of moments of the measure $d\mu$, $\mu_k = \int_{\mathbb{R}} x^k d\mu(x)$, $k = 0, 1, \dots, 2n - 1$, into the coefficients of the three-term recurrence relation. Algorithm is rational and nonlinear and it can be represented using recurrence relation which uses only addition and multiplication of the operations. In a similar way the modified Chebyshev algorithm is also realized.

Laurie's algorithm can be expressed as the mapping between the three-term recurrence coefficients of the measure $d\mu$ into the three-term recurrence

coefficients from which it is possible to get the parameters (nodes and weights) of the Gauss–Kronrod quadrature formula using QR-algorithm. Algorithm is also nonlinear and rational; the only operations involved are addition and multiplication.

The Christoffel modification algorithms are ones which give answer to the following problems. Suppose we are given three-term recurrence coefficients α_k and β_k for the measure $d\mu$. What are the corresponding three-term recurrence coefficients for the measures $d\mu(x)/(z-x)$ and $(z-x)d\mu(x)$?

3.2. Implemented functions. For all supported classes of orthogonal polynomials, the package provides the basic information. Function operating on the classes are the following:

- **aThreeTermRecurrence**-function returns three-term recurrence coefficients of the referenced polynomial class. It is implemented in the format of the pure function. It can return coefficients of the three-term recurrence relation in the closed analytic form.

- **aNorm**-function returns the norm of monic polynomials of the referenced class. It is implemented in the format of the pure function. It is able to return closed analytic expression of the norm of the referenced polynomial class.

- **aNumerator** returns numerator polynomials of the given order for the referenced polynomial class. It is also implemented in the format of the pure function.

- **aKernel** returns the kernel polynomial of the referenced polynomial class. It is implemented in the format of the pure function.

Functions which are specific for the continuous class of polynomials are the following:

- **aWeight** returns the weight function with respect to which referenced class is orthogonal. It is also pure function.

- **aGetInterval** returns the interval of the orthogonality, i.e., the support of the measure.

Functions specific for the discrete polynomial class are the following:

- **aDistribution** represents the distribution function with respect to which referenced polynomials are orthogonal to, where distribution function is given by $\psi(x) = \int_{-\infty}^x d\mu(x)$.

- **aSupport** returns the supporting set of the measure. However it is clear that support can be an infinite set, that is why function produces a message about the supporting set and also returns few points of the supporting set. The number of returned points is given as the parameter of the function.

Keywords for the quadrature formulae are the following:

- **aGaussian** is a construction of the Gaussian quadrature formulae. It is possible to perform construction for all supported classes of the orthogonal polynomials. In the case the function **aNodesWeights** is called for the construction of this type of quadrature rules. Function **aGaussianNodesWeights** is called which performs computation. Function **aGaussianNodesWeights** has different calling formats. It is possible to call function **aGaussianNodesWeights** for the class of polynomials which is not supported and for which coefficients of the three-term recurrence relation are known, if the construction with QR-algorithm is wanted, or for which we know good starting values for the Pasquini algorithm.

- **aRadau** and **aLobatto** perform constructions of the Gauss-Radau and the Gauss-Lobatto quadrature formulae, respectively. It is possible to construct the Gauss-Radau (Gauss-Lobatto) quadrature formula calling directly the function **aRadauNodesWeights** (**aLobattoNodesWeights**) or calling the function **aNodesWeights** (**aNodesWeights**). It is possible to construct the Gauss-Radau (Gauss-Lobatto) quadrature formula for all supported polynomial classes for which this formula has a meaning.

- **aKronrod** performs the construction of Gauss-Kronrod quadrature formulae. It is possible to construct the Gauss-Kronrod quadrature formula directly calling the function **aKronrodNodesWeights** or by calling **aNodesWeights** with the keyword **aKronrod**. Construction can be performed for all supported polynomial classes for which the Gauss-Kronrod quadrature formula exists, i.e., for which the additional nodes of the Gauss-Kronrod formula are inside the supporting set of the measure. If the additional nodes of The Gauss-Kronrod quadrature formula are not inside the supporting set, a construction can be performed using Laurie algorithm.

- **aTuran** performs the construction of Gauss-Turan quadrature formulae. It is possible to construct the Gauss-Turan quadrature formula calling the function **aTuranNodesWeights** or **aGaussianNodesWeights** using the keyword **aTuran**. Construction is possible for all supported classes of the polynomials with a positive orthogonality measure.

- **aSigma** performs the construction of generalized Gaussian quadrature formulae for multiple nodes. It is possible to perform the construction using the function **aSigmaNodesWeights** or the function **aNodesWeights** using the keyword **aSigma**. Construction is possible for all supported classes of polynomials with the positive orthogonality measure.

4. Examples of some nonclassical weights on \mathbb{R}_+

In this section we consider some (strong) nonclassical measures $d\mu(x) = w(x)dx$ on \mathbb{R}_+ for which the recursive coefficients $\alpha_k(d\mu)$ and $\beta_k(d\mu)$, $k =$

$0, 1, \dots, n-1$, must be determined numerically. Usually, the method of (modified) moments is not applicable in a standard machine arithmetic for a sufficiently large n , because of ill-conditioned process. For this reason, a construction of recursive coefficients must be carefully realized by an application of the discretized Stieltjes-Gautschi procedure. Then, these coefficients can be used in a construction of Gaussian quadratures. Such nonclassical quadratures were used in the last two decades in many applications in physics, economics, etc. Some of them were also useful in summation of slowly convergent series (cf. [7], [11], [12]).

Today, however, by using software packages with capabilities of variable-precision arithmetic and with symbolic computations, it is possible to use directly the method of moments. Here, we mention a few such cases.

1° *One side exponential weight* $w(t) = \exp(-t^s)$. The moments are

$$\mu_k = \int_0^{+\infty} t^k w(t) dt = \frac{1}{s} \Gamma\left(\frac{k+1}{s}\right), \quad k \in \mathbb{N}_0, \quad s > 0.$$

Gamma function can be evaluated to arbitrary numerical precision in MATHEMATICA (see [15]). To obtain the three-term recursion coefficients using our package `OrthogonalPolynomials`, for example for $s = 4$ and $n \leq 40$ with `WorkingPrecision->80`, one only needs to execute the following commands:

```
In[1]:= << orthogonalPolynomials`
In[2]:= s = 4; mom = Table[Gamma[(k + 1) / s] / s, {k, 0, 80}];
In[3]:= {a1, be} = aChebyshevAlgorithm[mom, WorkingPrecision -> 80];
```

Taking the `WorkingPrecision` sufficiently large, for example to be 160, it is possible to get the maximal relative error in the previous obtained coefficients $\{a1, be\}$.

```
In[4]:= {a1160, be160} = aChebyshevAlgorithm[mom, WorkingPrecision -> 160];
In[5]:= N[Max[a1 / a1160 - 1, be / be160 - 1]]
Out[5]:= 1.21657 × 10-41
```

According this result we can conclude that at least 40 decimal digits in the coefficients $\{a1, be\}$ are exact. It means that we can compute the parameters (nodes and weights) in all n -point Gaussian formulae for $n \leq 40$ with the same precision, because the Golub–Welsch algorithm is well-conditioned. In

our package `OrthogonalPolynomials` this algorithm is realized by the function `aGaussianNodesWeights`.

The values of `{a1, be}` to 20 digits (in order to save space) are the following:

```
In[6]:= N[a1, 20]
Out[6]:= {0.48887053372346189882, 0.62818641522278629241,
0.72749846252899886372, 0.79316667767856890602,
0.84550068070369304183, 0.88947385732675106451, 0.92768720002591623186,
0.96165664416000938487, 0.99234724686775554666, 1.0204171215207083426,
1.0463372701512168196, 1.0704575902650673656, 1.0930458350503219779,
1.1143119096302551335, 1.1344237100022280517, 1.1535178319357899849,
1.1717070362388493989, 1.1890855902011758356, 1.2057331762846359415,
1.2217178089736984983, 1.2370980493338568790, 1.2519247122938861248,
1.2662422009683794311, 1.2800895623911959874, 1.2935013321592688828,
1.3065082170504410548, 1.3191376518033485778, 1.3314142571083574230,
1.3433602192753623498, 1.3549956072375764124, 1.3663386389969644140,
1.3774059069596455613, 1.3882125696011830242, 1.3987725153686710336,
1.4090985035455914421, 1.4192022858878477709, 1.4290947121207110513,
1.4387858218192631640, 1.4482849247441745112, 1.4576006713440198665}

In[7]:= N[be, 20]
Out[7]:= {0.90640247705547707798, 0.098994721290579866816,
0.12360391490491043276, 0.14843255336839211574,
0.17037577312509127245, 0.18993643166658240538, 0.20774334942915538575,
0.22418006982753043827, 0.23951438841450045697, 0.25393872995188821220,
0.26759613325099209474, 0.28059611114038540344, 0.29302459285560838809,
0.30495041336522100910, 0.31642969274957242875, 0.32750888063257769802,
0.33822692827534905141, 0.34861687395582608134, 0.35870702352900726373,
0.36852184526261103904, 0.37808265888878377732, 0.38740817374168433837,
0.39651491440644919575, 0.40541756128132547144, 0.41412922591731055504,
0.42266167575240335412, 0.43102551914423687151, 0.43923035893723932450,
0.44728492085744145841, 0.45519716159467914078, 0.46297436036204744636,
0.47062319691513053549, 0.47814981839813125783, 0.48555989691048566686,
0.49285867931994947221, 0.50005103056042679311, 0.50714147142587357943,
0.51413421169130767423, 0.52103317924773004456, 0.52784204582164591569}
```

By the commands

```
{n10, w10} = aGaussianNodesWeights[10, a1, be, WorkingPrecision -> 30,
Precision -> 20];
N[{n10, w10}, 20]
```

we obtain the parameters `{n10, w10}` of the 10-point Gaussian formula with respect to the exponential weight $w(t) = e^{-t^4}$ on $(0, +\infty)$:

```
{0.020910958172875507809, 0.10797068631427716171, 0.25562328785749218991,
0.44893748090362466665, 0.66900160738867166048, 0.89859468409016216374,
1.1272235953202294099, 1.3520533630572184144, 1.5771033460898674163,
1.8173858305642309824},
```

{0.053411949182865253872, 0.11931457846424081532, 0.17270920978259131748,
0.20156537177553722183, 0.18599086303177089234, 0.11992799310002205324,
0.045095152077344811358, 0.0079124481479888566100,
0.00047014632630322120166, 4.7651668126347300673*10^-6}}

Recently, quadratures with these exponential weights have been used in [9].

2° *Einstein's weight* $w(t) = \varepsilon(t) = t/(e^t - 1)$ on $(0, +\infty)$. The moments are

$$\mu_k = \int_0^{+\infty} t^k w(t) dt = (k+1)! \zeta(k+2), \quad k \in \mathbb{N}_0,$$

where zeta function can be evaluated to arbitrary numerical precision. Furthermore, for certain special arguments, **Zeta** (in MATHEMATICA) automatically evaluates to exact values. Thus, as in the previous case, a direct application of the method of moments gives recursion coefficients, as well as the parameters of quadratures.

Integrals with Einstein's weight frequently appear in solid state physics, e.g. the total energy of thermal vibration of a crystal lattice can be expressed in the form $\int_0^{+\infty} f(t) \varepsilon(t) dt$, where $f(t)$ is related to the phonon density of states. Also, such kind of integrals can be used to sum infinite series,

$$\sum_{k=1}^{+\infty} a_k = \int_0^{+\infty} f(t) \varepsilon(t) dt,$$

if the general term of the series, $a_k = -F'(k)$, is the negative derivative of the Laplace transform $F(s) = \int_0^{+\infty} f(t) e^{-st} dt$ evaluated at $s = k$ of some known function f . For details on these applications to summation of slowly convergent series see [7].

3° *Fermi's weight* $w(t) = \varphi(t) = 1/(e^t + 1)$ on $(0, +\infty)$. Yet, another weight which appears in the theory of slowly convergent series is the Fermi weight function $\varphi(t)$ on $(0, +\infty)$. The moments are given by

$$\mu_k = \int_0^{+\infty} \frac{t^k}{e^t + 1} dt = \begin{cases} \log 2, & k = 0, \\ (1 - 2^{-k}) k! \zeta(k+1), & k > 0. \end{cases}$$

Integrals with Fermi's weight are encountered in the dynamics of electrons in metals. Also, Gaussian quadratures with respect to this weight function can be used in summation of the slowly convergent series of the form $\sum_{k=1}^{+\infty} (-1)^k a_k$ (see [7]).

3° *Hyperbolic weight* $w(t) = 1/\cosh^2 t$ on $(0, +\infty)$. The moments are

$$\mu_k = \int_0^{+\infty} t^k w(t) dt = \begin{cases} 1, & k = 0, \\ \log 2, & k = 1, \\ C_k \zeta(k), & k \geq 2, \end{cases}$$

where $C_k = (2^{k-1} - 1)k!/4^{k-1}$. Computation of C_k is based on the following lemma:

Lemma 1. *For $k \geq 2$ we have*

$$\int_0^{+\infty} \frac{x^k}{\cosh^2 x} dx = k! \zeta(k) \frac{2^{k-1} - 1}{4^{k-1}}.$$

Proof. Starting from the equality (cf. [14, p. 361])

$$\int_0^{+\infty} x^{\alpha-1} (\tanh ax - 1) dx = 2^{1-\alpha} a^{-\alpha} \Gamma(\alpha) \zeta(\alpha) (2^{1-\alpha} - 1), \quad a > 0, \Re \alpha > 0,$$

an integration by parts gives

$$\int_0^{+\infty} x^{\alpha-1} (\tanh ax - 1) dx = -\frac{a}{\alpha} \int_0^{+\infty} \frac{x^\alpha}{\cosh^2 ax} dx.$$

Now, putting $a = 1$ and $\alpha = k \geq 2$ we get the desired result. ■

Quadrature formulae with respect to this hyperbolic weight function and an application to summation of slowly convergent series were discussed in [11] and [12].

4° *The weight* $w(t) = \exp(-t^s - t^{-s})$, $s > 0$ on $(0, +\infty)$. For this exotic weight function, the moments are

$$\mu_k = \int_0^{+\infty} t^k w(t) dt = \frac{2}{s} K_{(k+1)/s}(2),$$

where $K_r(z)$ is the modified Bessel function of the second kind. MATHEMATICA evaluates $K_r(z)$ using the following function `Bessel[r, z]`, and function K_r can be evaluated with arbitrary precision, so that we can directly apply the method of moments to computation of recursive coefficients. This weight function has an application in the weighted polynomial approximation on \mathbb{R}^+ .

5. A conjecture for an oscillatory weight on $[-1, 1]$

Software packages with capabilities of variable-precision arithmetic and symbolic computations, as our package `OrthogonalPolynomials`, enable us to investigate problems with exotic weights and identify some properties, and after that prove them or state some conjectures.

Let $d\mu(x) = xe^{im\pi x}\chi([-1, 1]; x) dx$, $m \in \mathbb{N}$, where $\chi(A; \cdot)$ is the characteristic function of the set A . In [13] we investigated the existence of orthogonal polynomials π_k with respect to the functional

$$\mathcal{L}(p) = \int_{-1}^1 p(x)d\mu(x), \quad \mu_k = \mathcal{L}(x^k), \quad k \in \mathbb{N}_0,$$

as well as several of their properties (three-term relation, differential equation, etc.). Also, we considered related quadrature rules and give applications of such quadrature rules to some classes of integrals involving highly oscillatory integrands. For example, we proved that *for every $m \in \mathbb{N}$ the monic orthogonal polynomials π_k exist uniquely and satisfy the three-term recurrence relation*

$$\pi_{k+1}(x) = (x - i\alpha_k)\pi_k(x) - \beta_k\pi_{k-1}(x), \quad k \in \mathbb{N}_0,$$

with $\pi_0(x) = 1$ and $\pi_{-1}(x) = 0$.

Using software `OrthogonalPolynomials` we determined analytic expressions for recursion coefficients for α_k and β_k ($k \leq 20$) and stated the following conjecture:

Conjecture 1. *Let $a_k(z)$ and $c_k(z)$ be algebraic polynomials with integer coefficients of degree r_k and s_k , respectively, i.e., $a_k(z) = A_k z^{r_k} + \dots$ and $c_k(z) = z^{s_k} + \dots$. If $\zeta = m\pi$ and $k \geq 2$, then*

$$\alpha_k = \frac{a_k(\zeta^2)}{\zeta c_{k-1}(\zeta^2)c_k(\zeta^2)}, \quad \beta_k = B_k \frac{c_{k-2}(\zeta^2)c_k(\zeta^2)}{\zeta^2 c_{k-1}(\zeta^2)^2},$$

where

$$A_n = \begin{cases} -\frac{k^2 - 1}{4} & (k \text{ odd}), \\ \frac{k^2 + 10k + 8}{4} & (k \text{ even}), \end{cases} \quad B_k = \begin{cases} 1 & (k \text{ odd}), \\ -k^2 & (k \text{ even}), \end{cases}$$

and

$$r_k = \frac{k(k+1)}{2}, \quad s_k = \begin{cases} \frac{(k+1)^2}{4} & (k \text{ odd}), \\ \frac{k(k+2)}{4} & (k \text{ even}), \end{cases}$$

Here we can mention that the complexity of expressions for α_k and β_k increases exponentially with k , but also there is an efficient algorithm for their numerical construction (see [13]).

6. Orthogonality on radial rays in \mathbb{C}

In this section we consider the following linear functional

$$\mathcal{L}(p) = \sum_{k=1}^n \int_0^1 p(r_k x e^{i\varphi_k}) d\mu_k(x),$$

where $r_k, k = 1, \dots, n$, are given real numbers and $\varphi_k, k = 1, \dots, n$, are angles from the interval $(-\pi, \pi]$, and $d\mu_k, k = 1, \dots, n$, are positive finite measures supported on $[0, 1]$ (see Fig. 1). It can be easily proved that \mathcal{L} is a linear functional acting on the linear space of all polynomials \mathcal{P} .

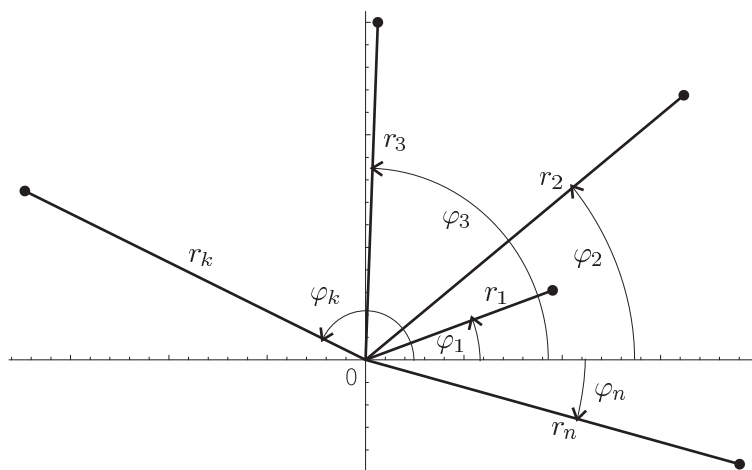


Figure 1: Radial rays in the complex plane \mathbb{C}

In the case $\varphi_k = 0, k = 1, \dots, n$, we have a case on the real line \mathbb{R} , with a positive definite functional. However, if at least one $\varphi_k \neq 0$, we get completely different situation. We cannot even state that \mathcal{L} is regular functional, i.e., the existence of the corresponding orthogonal polynomials is not granted. But, in some special cases we are able to prove that the functional \mathcal{L} is regular.

Suppose we have $d\mu_k = d\mu, k = 1, \dots, n$. Then, we can express the moments of \mathcal{L} in the following form

$$M_\nu = \mathcal{L}(z^\nu) = m_\nu \sum_{k=1}^n r_k^\nu e^{i\nu\varphi_k},$$

for $\nu \in \mathbb{N}_0$, where $m_\nu, \nu \in \mathbb{N}_0$, are moments of the measure $d\mu$.

Theorem 5. *Suppose $r_k \in \mathbb{Q}$ and $\varphi_k = k\zeta\pi, k = 1, \dots, n$, where $\zeta \in \mathbb{R} \setminus \mathbb{Q}$ is an algebraic number. If all of $d\mu_k$ are Lebesgue measures, then the associated moment functional $\mathcal{L}(p) = \sum_{k=1}^n \int_0^1 p(r_k x e^{ik\zeta\pi}) dx, p \in \mathcal{P}$, is regular.*

The proof relies on the Hilbert problem of proving transcendency of α^β for α and β algebraic, $\alpha \neq 0, 1$ and β irrational. Namely, we can use Gelfond result and choose $\alpha = -1$ and $\beta = \zeta$ in order to get transcendency of $e^{i\zeta\pi}$, which in turn guaranties that the Hankel determinants for the sequence of moments $M_\nu, \nu \in \mathbb{N}_0$, are non-vanishing.

In the rest of this paper, we present some numerical results. Consider linear functional of the following form

$$\mathcal{L}(p) = \sum_{k=1}^3 \int_0^1 p(x e^{ik5\sqrt{2}\pi/21}) dx, \quad p \in \mathcal{P}.$$

According to the previous theorem this functional is regular. The modulus of recursion coefficients for $k = 0, 1, \dots, 400$ are presented in Fig. 2. The corresponding $\arg \beta_k, k = 0, 1, \dots, 400$, are displayed in Fig. 3.

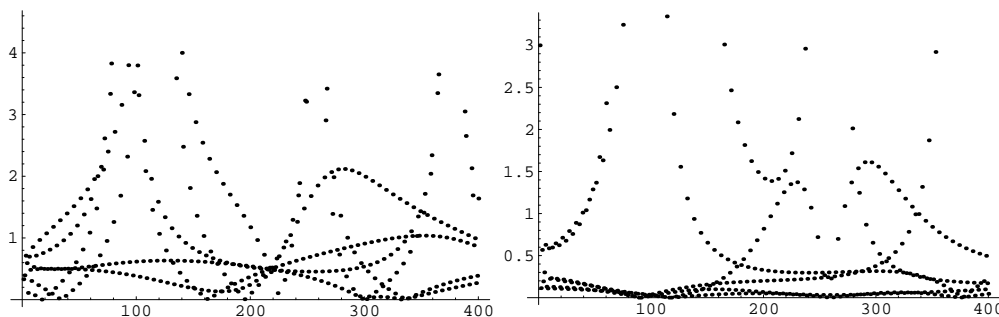
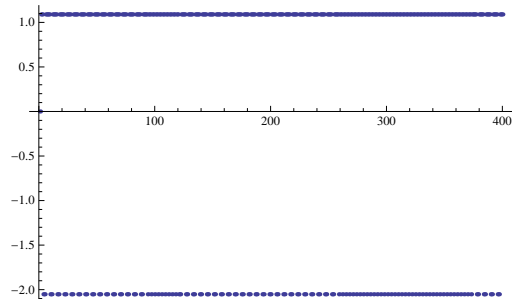
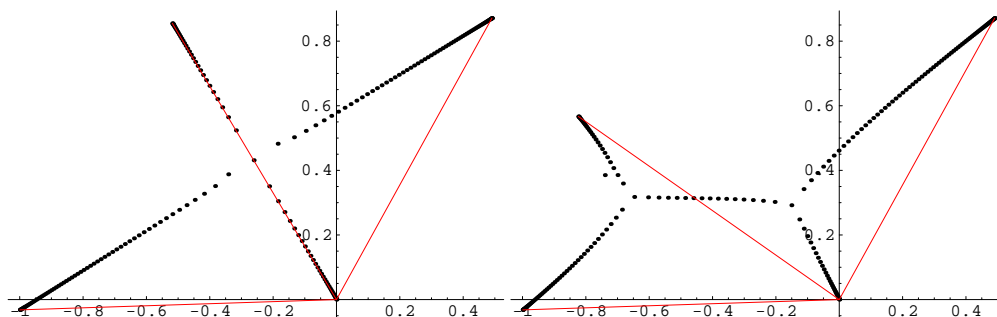


Figure 2: The coefficients $|\alpha_k|$ (left) and $|\beta_k|$ (right), $k = 0, 1, \dots, 400$

Finally, we present a distribution of zeros of the orthogonal polynomials $\pi_n(z)$. For example, if $n = 201$ all zeros of $\pi_{201}(z)$ are displayed in Fig. 4 (left). The rays are presented by solid lines. As we can see zeros are distributed symmetrically with respect to the given radial rays. But, a perturbation of the middle ray gives a modified distribution of zeros presented in Fig. 4 (right).

Under condition of the uniform boundedness of sequences of three-term recurrence coefficients (i.e., boundedness of the Jacobi operator J), the well-known results of H. Stahl can be applied. In that case the essential spectrum of

Figure 3: $\arg \beta_k$ for $k = 0, 1, \dots, 400$ Figure 4: Distribution of zeros of $\pi_{201}(z)$ for given rays (left) and for the case when the middle ray is perturbed (right)

J is given by the set of smallest capacity which contains ends of the rays, i.e., the points $r_k e^{i\varphi_k}$, $k = 1, \dots, n$, and the origin 0.

References

- [1] T.S. Chihara, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, New York (1978).
- [2] A.S. Cvetković, G.V. Milovanović, The Mathematica Package “OrthogonalPolynomials”, *Facta Univ. Ser. Math. Inform.* bf 19, (2004), 17–36.
- [3] W. Gautschi, On generating orthogonal polynomials, *SIAM J. Sci. Comput.* **3** (1982), 289–317.
- [4] W. Gautschi, Algorithm 726: ORTHPOL – A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules, *ACM Trans. Math. Software* **20** (1994), 21–62.
- [5] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Clarendon Press, Oxford (2004).

- [6] W. Gautschi: Orthogonal polynomials (in Matlab), *J. Comput. Appl. Math.* **178**, (2005), 215–234.
- [7] W. Gautschi, G.V. Milovanović, Gaussian quadrature involving Einstein and Fermi functions with an application to summation of series, *Math. Comp.* **44** (1985), 177–190.
- [8] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, *Math. Comp.* **23** (1969), 221–230.
- [9] A. Holly, A. Monfort, M. Rockinger, Fourth order pseudo maximum likelihood methods, *J. Econometrics* **162** (2011), 278–293.
- [10] G. Mastroianni, G.V. Milovanović, *Interpolation Processes – Basic Theory and Applications*, Springer Monographs in Mathematics, Springer-Verlag, Berlin (2008).
- [11] G.V. Milovanović, Summation of series and Gaussian quadratures, In: *Approximation and Computation* (R.V.M. Zahar, ed.), ISNM Vol. 119, Birkhäuser, Basel–Boston–Berlin (1994), 459–475.
- [12] G.V. Milovanović, Summation of series and Gaussian quadratures, II, *Numer. Algorithms* **10** (1995), 127–136.
- [13] G.V. Milovanović, A.S. Cvetković, Orthogonal polynomials and Gaussian quadrature rules related to oscillatory weight functions, *J. Comput. Appl. Math.* **179** (2005), 263–287.
- [14] A.P. Prudnikov, Yu.A. Brychkov, O.I. Marichev, *Integrals and Series, Vol. 1: Elementary Functions*, Gordon & Breach Science Publishers, New York (1986) [Translated from the Russian].
- [15] J.L. Spouge, Computation of the gamma, digamma and trigamma functions, *SIAM J. Numer. Anal.* **31** (1994), 931–944.
- [16] G. Szegő, *Orthogonal Polynomials*, Amer. Math. Soc. Colloq. Publ. **23**, 4th ed., Amer. Math. Soc., Providence, R. I. (1975).

¹ *Matematički Institut SANU*
Kneza Mihajla 36, 11001 Belgrade, SERBIA
e-mail: gvm@mi.sanu.ac.rs

Received: November 20, 2011

² *Faculty of Mechanical Engineering*
University of Belgrade
Kraljice Marije 16, 11120 Belgrade, SERBIA
e-mail: acvetkovic@mas.bg.ac.rs