Polynomials orthogonal with respect to the Binet and related weight functions

Walter Gautschi* and Gradimir V. Milovanović†

Abstract

Procedures based on moments are developed for computing the three-term recurrence relation for orthogonal polynomials relative to the Binet, generalized Binet, squared Binet, and related subrange weight functions. Monotonicity properties for the zeros of the respective orthogonal polynomials are also established.

2000 *Mathematics Subject Classification*: 33C47, 65D20

1 Introduction

The *Binet weight function* may be defined by

$$w_1(x) = -\log(1 - \mathrm{e}^{-|x|}) \quad \text{on } [-\infty, \infty]; \tag{1}$$

see [1, III, Eq. (5.4)], where the Binet *distribution* is defined by $w^B(x) = w_1(2\pi x)/(2\pi)$ and used in Binet's summation formula, *ibid.*, Eq. (5.15). More generally,

$$w_1(x; \alpha) = -\log(1 - \alpha\mathrm{e}^{-|x|}) \quad \text{on } [-\infty, \infty], \quad 0 < \alpha \le 1, \tag{2}$$

*Department of Computer Science, Purdue University, West Lafayette, IN 47907-2066, USA (wgautschi@purdue.edu)

†Mathematical Institute, Serbian Academy of Sciences and Arts, Belgrade, Serbia (gvm@mi.sanu.ac.rs)

is, what may be called, the *generalized Binet weight function*. We are interested in the polynomials orthogonal with respect to the weight functions (1) and (2), in particular in the recurrence formula

$$\pi_{k+1}(x) = (x - \alpha_k)\pi_k(x) - \beta_k \pi_{k-1}(x), \quad k = 0, 1, 2, \ldots\ , \quad \pi_{-1}(x) = 0, \tag{3}$$

satisfied by the respective monic polynomials. The coefficients α_k, β_k can be obtained by the classical Chebyshev algorithm, since the moments of both weight functions are known in terms of factorials and generalized polylogarithm functions. It is true that the classical Chebyshev algorithm is notoriously unstable, but we get around this problem by using sufficiently high precision. This is discussed for the Binet and generalized Binet weight functions in Section 2. The same can be done with the squares of the Binet and generalized Binet weight functions (Section 3), with the halfrange Binet and generalized Binet weight functions (Section 4), as well as with the squares of the halfrange weight functions (Section 5). Upper and lower subrange Binet weight functions are also considered in Section 6.

In the case of the generalized weight functions with parameter α, we prove that all zeros, resp. positive zeros when the weight function is symmetric, are monotonically decreasing as functions of α. They are shown to be monotonically increasing as functions of the upper or lower limit of the orthogonality interval. We do this by applying Markov's theorem and two variants thereof, and by a new related theorem of our own.

2 Binet and generalized Binet weight functions

2.1 Binet weight function

Since the weight function in (1) is symmetric with respect to the origin, its moments are

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd}, \\ -2\displaystyle\int_0^\infty x^k \log(1 - \mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even}, \end{cases} \tag{4}$$

Substituting $\mathrm{e}^{-x} = t$ in the integral of (4), one gets

$$\mu_k = 2(-1)^{k+1} \int_0^1 \log^k t \, \log(1-t) \, \frac{\mathrm{d}t}{t} \, ,$$

and thus

$$\mu_k = 2\, k!\, S_{k+1,1}(1) = 2\, k!\, \mathrm{Li}_{k+2}(1), \tag{5}$$

where

$$S_{n,p}(x) = \frac{(-1)^{n+p-1}}{(n-1)!p!} \int_0^1 \log^{n-1}(t) \log^p(1-xt) \, \frac{\mathrm{d}t}{t} \tag{6}$$

is the Nielsen generalized polylogarithm [7, Eq. (1.1)] and $\mathrm{Li}_n(x)$ the ordinary polylogarithm ([6, Eq. (1.1)]). We can thus apply the classical Chebyshev algorithm (cf., e.g., [2, §2.1.7]) in sufficiently high precision to generate any number N of recurrence coefficients α_k, β_k, $k = 0, 1, \ldots, N-1$, to any desired accuracy.

To implement this in Matlab, one needs, foremost, the routine `smom_binet.m`[1] that generates in `dig`-digit arithmetic the $2\,\mathtt{N}\times 1$ array `mom` of the first $2\,\mathtt{N}$ moments (4),

$$\mathtt{mom} = \mathtt{smom\_binet}(\mathtt{dig}, \mathtt{N}).$$

In addition, the routine `dig_binet.m` is provided that, with the command

$$[\mathtt{ab}, \mathtt{dig}] = \mathtt{dig\_binet}(\mathtt{N}, \mathtt{dig0}, \mathtt{dd}, \mathtt{nofdig}), \tag{7}$$

helps to determine the number `dig` of digits needed to obtain the $\mathtt{N}\times 2$ array `ab` of the first `N` recurrence coefficients α_k, β_k, $k = 0, 1, \ldots, \mathtt{N}-1$, to an accuracy of `nofdig` digits. The way this routine works is as follows: It first calculates the array `ab` with an estimated number `dig0` of digits (which is printed) and then successively increases (and prints) the number of digits in units of `dd` digits until the desired accuracy is achieved. If this happens after just one increment, the value of `dig0` must be lowered until at least two increments have occurred. The last value of `dig` printed can then be taken as the number of digits needed. A typical value of `dd` is 4. The command

$$\mathtt{ab} = \mathtt{sr\_binet}(\mathtt{dig}, \mathtt{nofdig}, \mathtt{N}), \tag{8}$$

[1] All Matlab routines and text files referenced in this paper can be accessed at `https://www.cs.purdue.edu/homes/wxg/archives/2002/codes/BINET.html`.

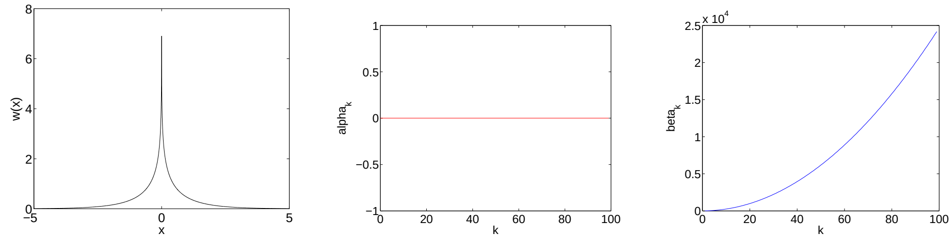

Figure 1: Binet weight function and its recurrence coefficients

finally, computes directly, in `dig`-digit arithmetic, the first `N` recurrence coefficients and places them to `nofdig` digits into the `N` $\times 2$ array `ab`.

Example 1. The first 100 recurrence coefficients to 32 digits of the Binet weight function.

With `N` $= 100$, `dig0` $= 56$, `dd` $= 4$, `nofdig` $= 32$, the routine (7) yields `dig` $= 64$ after two increments and also produces the 100×2 array `ab` of the first 100 recurrence coefficients to an accuracy of 32 digits. The α- and β-coefficients are depicted in the second and third plot of Fig. 1, the first showing the Binet weight function. The recurrence coefficients are also made available in the text file `coeff_binet.txt`, which can be loaded into the Matlab working window by the routine `loadvpa.m`. For the latter, see [3, p. ix]; see also [4, 2.3.8]. The same array `ab` can also be obtained directly with the routine (8), using `dig` $= 64$, `nofdig` $= 32$, and `N` $= 100$.

All α_k, of course, are zero, and the first β-coefficient obtained, $\beta_0 = \int_{\mathbb{R}} w(x)\mathrm{d}x$, is in complete agreement with the known moment $\mu_0 = \pi^2/3$. The remaining βs, multiplied by $(2\pi)^2$ (because of the difference in normalization) agree completely with those for $k = 1, 2, \ldots, 6$ given in [1, III, p. 721] in rational form, and with those for $k = 7, 8, \ldots, 16$ given to 15 decimals in the same reference, at least to 14, but usually to all 15 digits.

2.2 Generalized Binet weight function

2.2.1 Recurrence coefficients

The weight function (2), again being symmetric, has moments

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ -2\displaystyle\int_0^\infty x^k \log(1-\alpha \mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even.} \end{cases} \tag{9}$$

Similarly as in Subsection 2.1, one finds

$$\mu_k = 2\,k!\,S_{k+1,1}(\alpha) = 2\,k!\,\mathrm{Li}_{k+2}(\alpha), \quad k \text{ even.} \tag{10}$$

The moments (9) are now generated by the Matlab command `mom=smom_gbinet(dig,N,a)`, where `a` is the value of α and $0 < \alpha \leq 1$.

Example 2. The first 100 recurrence coefficients to 32 digits of the generalized Binet weight function for $\alpha = 1/2$.

The Matlab command `[ab,dig]=dig_gbinet(N,a,dig0,dd,nofdig)`, when run with `N` $=$ 100, `a` $= 1/2$, `dig0` $=$ 56, `dd` $=$ 4, `nofdig` $=$ 32, yields `dig` $=$ 64. The same command, or more directly, the command `ab=sr_gbinet(dig,nofdig,N,a)` with `dig` $=$ 64, produces the 100×2 array `ab` of the first 100 recurrence coefficients to an accuracy of 32 digits. They are depicted in the second and third plot of Fig. 2, the first showing the generalized Binet weight function for $\alpha = 1/2$. They are also made available in the text file `coeff_gbinet.txt`.

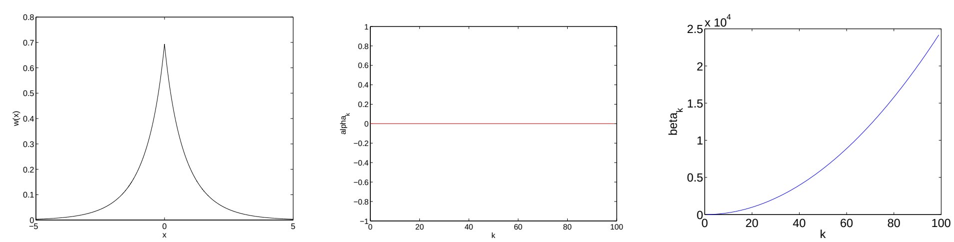

Figure 2: Generalized Binet weight function and its recurrence coefficients

2.2.2 Zeros of the orthogonal polynomials

The first objective of this subsection is to investigate the zeros of orthogonal polynomials depending on a parameter and to prove some monotonicity results. For this we use Markov's theorem and two simple corollaries thereof, as well as a new, but related, theorem. The second objective is to show appropriate plots.

We first recall Markov's theorem ([8, Theorem 6.12.1]).

Theorem 1 (A. Markov). *Let $w(x;\alpha)$ be a positive weight function on $[a,b]$, $-\infty \le a < b \le \infty$, depending on a parameter α, $\alpha_1 < \alpha < \alpha_2$. Assume that the first $2n$ moments of w and of $\frac{\partial w}{\partial \alpha}$ exist, and let π_n^α denote the monic polynomial of degree n orthogonal with respect to the weight function $w(\,\cdot\,;\alpha)$. Then each zero of π_n^α is an increasing (decreasing) function of α on (α_1,α_2) provided that*

$$\frac{1}{w(x;\alpha)}\frac{\partial w(x;\alpha)}{\partial \alpha} \tag{11}$$

is an increasing (decreasing) function of x on $[a,b]$.

Here are two simple corollaries to Markov's theorem.

Corollary 1. *Let $w(x;\alpha)$ be as in the theorem, and $w_r(x;\alpha) = [w(x;\alpha)]^r$, $r > 0$, have finite moments of order $\le 2n-1$. Then each zero of the nth-degree polynomial orthogonal with respect to the weight function w_r is increasing (decreasing) on (α_1,α_2) depending on whether (11) is increasing (decreasing) on $[a,b]$.*

Proof. We have

$$\frac{1}{w_r(x;\alpha)}\frac{\partial w_r(x;\alpha)}{\partial \alpha} = \frac{r\,[w(x;\alpha)]^{r-1}}{[w(x;\alpha)]^r}\frac{\partial w(x;\alpha)}{\partial \alpha} = \frac{r}{w(x;\alpha)}\frac{\partial w(x;\alpha)}{\partial \alpha}. \qquad \square$$

If $r < 0$, the type of monotonicity is reversed, from increasing to decreasing and vice versa.

Corollary 2. *Let $w(x;\alpha)$ be symmetric on $[-a,a]$, $0 < a \le \infty$, i.e., $w(-x;\alpha) = w(x;\alpha)$ for $0 \le x \le a$, but otherwise as in the theorem. Then each positive zero of π_n^α is increasing (decreasing) on (α_1,α_2) depending on whether (11) is increasing (decreasing) on $[0,a]$.*

Proof. Suppose first that $n = 2k$ is even. Then, as is well known (se, e.g., [2, Theorem 1.18]),

$$\pi_{2k}^{\alpha}(x;\alpha) = \pi_k^{+}(x^2;\alpha),$$

where $\pi_k^{+}(\,\cdot\,;\alpha)$ is orthogonal on $[0, a^2]$ with respect to the weight function $w^{+}(t;\alpha) = t^{-1/2}w(t^{1/2};\alpha)$ on $[0, a^2]$. Now the positive zeros of $\pi_{2k}^{\alpha}(x;\alpha)$ are the positive square roots of the zeros of $\pi_k^{+}(\,\cdot\,;\alpha)$, hence increasing (decreasing) on $[\alpha_1, \alpha_2]$ if the same is true for the zeros of $\pi_k^{+}(\,\cdot\,;\alpha)$. But

$$\frac{1}{w^{+}(t;\alpha)}\,\frac{\partial w^{+}(t;\alpha)}{\partial\alpha} = \frac{t^{-1/2}}{t^{-1/2}w(t^{1/2};\alpha)}\,\frac{\partial w(t^{1/2};\alpha)}{\partial\alpha} = \frac{1}{w(t^{1/2};\alpha)}\,\frac{\partial w(t^{1/2};\alpha)}{\partial\alpha}\,,$$

from which Corollary 2 follows.

For $n = 2k+1$ odd, the proof is similar, using $\pi_{2k+1}^{\alpha}(x;\alpha) = x\pi_k^{-}(x^2;\alpha)$, where $\pi_k^{-}(\,\cdot\,;\alpha)$ is orthogonal on $[0, a^2]$ with respect to the weight function $w^{-}(t) = t^{1/2}w(t^{1/2};\alpha)$.

$\square$

For later purposes, we consider the case where the parameter is not contained in the weight function, but is the upper limit of the interval of orthogonality, i.e., the (monic) polynomials $\{\pi_k\}$ are orthogonal on $[a, c]$, $-\infty \le a < c < \infty$, with respect to a weight function w,

$$\int_a^c \pi_k(x)\pi_\ell(x)w(x)\mathrm{d}x = 0, \quad k \neq \ell.$$

Theorem 2. *Let $w(x)$ be a positive weight function on $[a, c]$, $-\infty \le a < c < \infty$, having finite moments μ_k for $0 \le k \le 2n-1$. Then each zero $x_\nu = x_\nu(c)$ of π_n is a monotonically increasing function of c.*

Proof. The proof follows the same line of arguments as the proof of Markov's theorem given in [8, Theorem 6.12.1], being based on the Gauss quadrature formula

$$\int_a^c p(x)w(x)\mathrm{d}x = \sum_{\mu=1}^{n} \lambda_\mu(c)\,p(x_\mu(c)), \quad p \in \mathbb{P}_{2n-1}. \tag{12}$$

Differentiating (12) with respect to c, we have

$$p(c)w(c) = \sum_{\mu=1}^{n} \lambda_\mu(c)\,p'(x_\mu(c))\,\frac{\mathrm{d}x_\mu}{\mathrm{d}c} + \sum_{\mu=1}^{n} \frac{\mathrm{d}\lambda_\mu}{\mathrm{d}c}\,p(x_\mu(c)). \tag{13}$$

Let

$$p(x) = \frac{\pi_n^2(x)}{x - x_\nu}\,, \quad p'(x_\nu) = [\pi_n'(x_\nu)]^2.$$

Then, since $p'(x_\mu) = 0$ for $\mu \neq \nu$, we get from (13) that

$$\frac{\pi_n^2(c)}{c - x_\nu}\, w(c) = \lambda_\nu(c)\, [\pi_n'(x_\nu)]^2\, \frac{\mathrm{d}x_\nu}{\mathrm{d}c}\,. \tag{14}$$

Since on the right, both factors multiplying $\mathrm{d}x_\nu/\mathrm{d}c$ are positive, and on the left, $w(c) > 0$, $x_\nu < c$, it follows that $\mathrm{d}x_\nu/\mathrm{d}c > 0$.

□

Remark to Theorem 2. Theorem 2 is valid also if c is the lower limit of the orthogonality interval, by the same proof. Indeed, the left-hand side of Eq. (13) will then have a minus sign in front of it, and so does the left-hand side of Eq. (14). But now, $x_\nu > c$.

In the case at hand, the weight function is $w_1(x;\alpha)$ in (2), which is clearly symmetric on $(-\infty, \infty)$, so that according to Corollary 2 of Markov's theorem, the positive zeros of the generalized Binet polynomial π_n^α are increasing (decreasing) depending on whether

$$\frac{1}{w_1(x;\alpha)}\,\frac{\partial w_1(x;\alpha)}{\partial \alpha} = \frac{-1}{(\mathrm{e}^x - \alpha)\log(1 - \alpha \mathrm{e}^{-x})} \tag{15}$$

is increasing (decreasing) for x in $(0, \infty)$.

Let the right-hand side of (15), as a function of x, be denoted by $f(x)$ and the denominator by $g(x)$. Then

$$f(x) = \frac{-1}{g(x)}\,, \quad f'(x) = \frac{g'(x)}{g^2(x)}\,.$$

So the matter depends on whether $g'(x)$ is positive (negative) on $[0, \infty)$.

Using the product rule of differentiation, we have

$$
\begin{aligned}
g'(x) &= (\mathrm{e}^x - \alpha)\,\frac{\alpha \mathrm{e}^{-x}}{1-\alpha\mathrm{e}^{-x}} + \mathrm{e}^x \log(1-\alpha\mathrm{e}^{-x}) \\
&= (\mathrm{e}^x - \alpha)\,\frac{\alpha}{\mathrm{e}^x-\alpha} + \mathrm{e}^x \log(1-\alpha\mathrm{e}^{-x}) \\
&= \alpha + \mathrm{e}^x \log(1-\alpha\mathrm{e}^{-x}) \\
&= \mathrm{e}^x[\alpha\mathrm{e}^{-x} + \log(1-\alpha\mathrm{e}^{-x})].
\end{aligned}
$$

Letting $t = \alpha\mathrm{e}^{-x}$, $0 < t < 1$, and $y(t) = t + \log(1-t)$, we have $y(0) = 0$ and $y'(t) = -t/(1-t) < 0$, so that $y(t) < 0$ on $(0,1)$, i.e., the function in brackets is negative for x in $(0,\infty)$, that is, $g'(x) < 0$. Thus, all positive zeros of π_n^α are decreasing as a function of α in $(0,1)$.

In order to plot the zeros, we first use the Matlab routine `dig_gbinet.m` to determine the number `dig` of digits needed to obtain the first 30 recurrence coefficients to an accuracy of 6 digits (more than enough for plotting purposes). The result, for any α in $(0,1]$, is `dig` $= 16$. Once the respective variable-precision array `ab` has been obtained, one can revert to double precision for the rest of the computations.

The Matlab routine `plot_zeros_gbinet.m` for `N` $= 30$ plots the 15 positive zeros of π_n^α, and at the same time verifies their monotonic descent as functions of α. That descent is relatively slow, almost imperceptible; see the third plot in Fig. 3. The first two plots show the smallest and largest positive zero, plotted in a scale that makes their monotone descent visible. The plots, indeed, suggest not only monotonicity, but also concavity, and perhaps even complete monotonicity. In general, monotonicity was found to be consistently weaker the larger the zero. For example, when $n = 30$, the relative decrement of the smallest zero varies in absolute value between 4.41×10^{-3} and 4.52×10^{-1}, whereas the one for the largest zero varies between 3.83×10^{-6} and 1.59×10^{-5}.

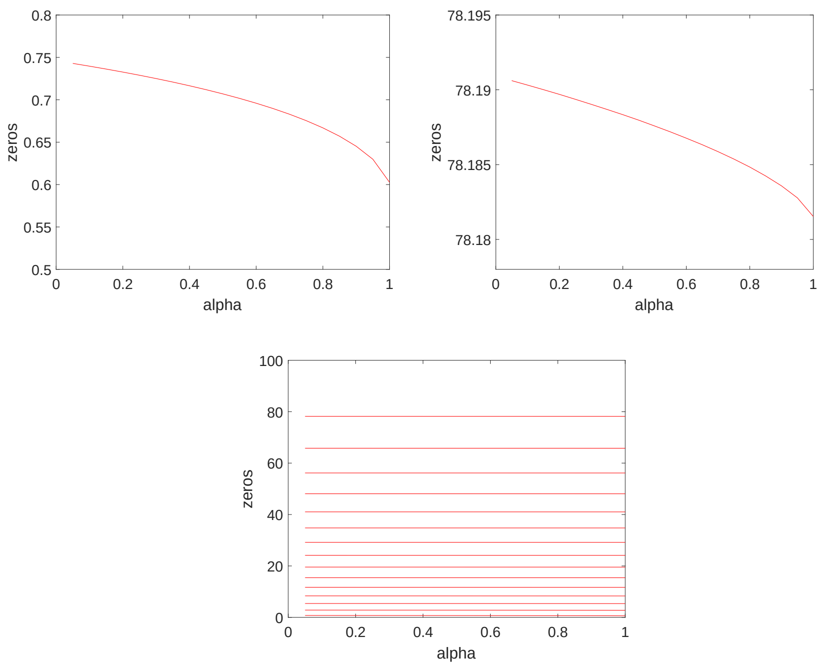

Figure 3: The positive zeros for $n = 30$ of the generalized Binet polynomials in dependence of the parameter α, $0 < \alpha \leq 1$; the smallest and largest positive zero (top), all positive zeros (bottom)

3 Squared Binet and squared generalized Binet weight functions

3.1 Squared Binet weight function

The *squared Binet weight function*,

$$w_2(x) = \log^2(1 - \mathrm{e}^{-|x|}) \quad \text{on } [-\infty, \infty], \tag{16}$$

being symmetric, has the moments

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd}, \\ 2\displaystyle\int_0^\infty x^k \log^2(1 - \mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even}. \end{cases} \tag{17}$$

Putting $e^{-x} = t$ in the integral of (17), we get

$$\mu_k = 2(-1)^k \int_0^1 \log^k t \, \log^2(1 - t) \, \frac{\mathrm{d}t}{t}\,.$$

For $k = 0$ we have $\mu_0 = 4\,\zeta(3)$ ([5, Eq. 4.261.12 for $n = 0$]) while for k (even) > 0

$$\mu_k = 4\,k!\,S_{k+1,2}(1),$$

with $S_{n,p}$ as defined in (6). We have ([7, pp. 39, 41] or [6, p. 1236])

$$s_n = S_{n-1,1}(1) = \sum_{j=1}^{\infty} \frac{1}{j^n} = \zeta(n)$$

and [7, Eq. (4.16)]

$$S_{n-1,2}(1) = \tfrac{1}{2}\,ns_{n+1} - \tfrac{1}{2}\,(s_2 s_{n-1} + s_3 s_{n-2} + \cdots + s_{n-1}s_2),$$

so that

$$\mu_k = 2k!\left[(k+2)\zeta(k+3) - \sum_{\nu=2}^{k+1} \zeta(\nu)\zeta(k+3-\nu)\right], \quad k(\text{even}) \; > 0. \tag{18}$$

The first `N` moments (18) are generated in `dig`-digit arithmetic by the Matlab command `mom=smom_sqbinet(dig,N)`.

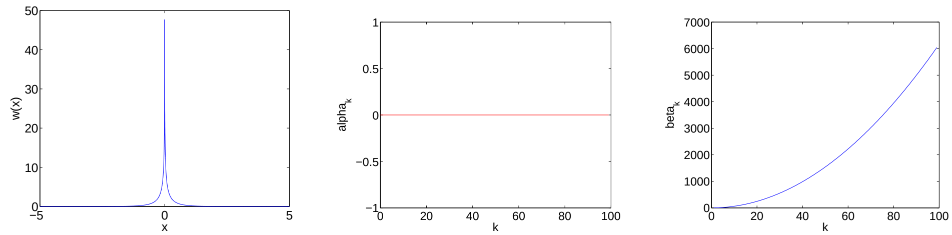

Figure 4: Squared Binet weight function and its recurrence coefficients

Example 3. The first 100 recurrence coefficients to 32 digits of the squared Binet weight function.

The Matlab command `[ab,dig]=dig_sqbinet(N,dig0,dd,nofdig)`, when run with `N`$=100$, `dig0` $=108$, `dd` $=4$, `nofdig` $=32$, yields `dig` $=116$. The same command, or more directly, the command `ab=sr_sqbinet(dig, nofdig,N)` with `dig` $=116$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 4, the first showing the squared Binet weight function. They are also made available in the text file `coeff_sqbinet.txt`; see also [4, 2.3.9].

3.2 Squared generalized Binet weight function

3.2.1 Recurrence coefficients

The *squared generalized Binet weight function*,

$$w_2(x;\alpha)=\log^2(1-\alpha\mathrm{e}^{-|x|})\quad\text{on } [-\infty,\infty],\ 0<\alpha\leq 1, \tag{19}$$

has the moments

$$\mu_k=\begin{cases} 0 & \text{if } k \text{ is odd,}\\ 2\displaystyle\int_0^\infty x^k\log^2(1-\alpha\mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even.}\end{cases} \tag{20}$$

Similarly as in Subsection 3.1, one finds

$$\mu_k=4\,k!\,S_{k+1,2}(\alpha), \tag{21}$$

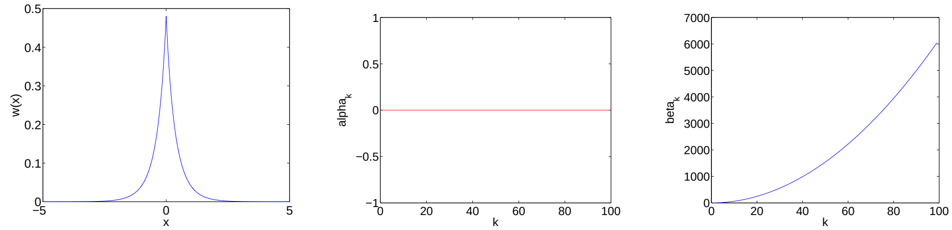

Figure 5: Squared generalized Binet weight function and its recurrence coefficients

where $S_{n,p}(x)$ is the Nielsen generalized polylogarithm function (6). We have [6, Eq. (2.9)]

$$S_{k+1,2}(\alpha) = \sum_{\nu=1}^{\infty} \left(\sum_{\mu=1}^{\nu} \frac{1}{\mu} \right) \frac{\alpha^{\nu+1}}{(\nu+1)^{k+2}}, \quad 0 < \alpha \leq 1. \tag{22}$$

The series converges fairly rapidly for all $k \geq 0$ provided α is not too close to 1.

The moments (20) are now generated by the Matlab command `mom=smom_sqg_binet(dig,N,a)`.

Example 4. The first 100 recurrence coefficients to 32 digits of the squared generalized Binet weight function for $\alpha = 1/2$.

The Matlab command `[ab,dig]=dig_sqgbinet(N,a,dig0,dd,nofdig)`, when run with `N` $= 100$, `a` $= 1/2$, `dig0` $= 56$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 64$. The same command, or more directly, the command `ab=sr_sqgbinet(dig,nofdig,N,a)` with `dig` $= 64$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digts. They are depicted in the second and third plot of Fig. 5, the first showing the squared generalized Binet weight function for $\alpha = 1/2$. They are also made available in the text file `coeff_sqgbinet.txt`.

3.2.2 Zeros of the orthogonal polynomials

By virtue of Corollary 1 to Markov's theorem and of what was proved in Subsection 2.2.2, the positive zeros of the squared generalized Binet polynomials all decrease monotonically. They behave similarly as those for the generalized Binet polynomials, but are only about half as large. For $n = 30$, the smallest and largest positive zero are shown in the first two plots of Fig. 6, and all 15 positive zeros in the third plot; cf. `plot_zeros_sqgbinet.m`.

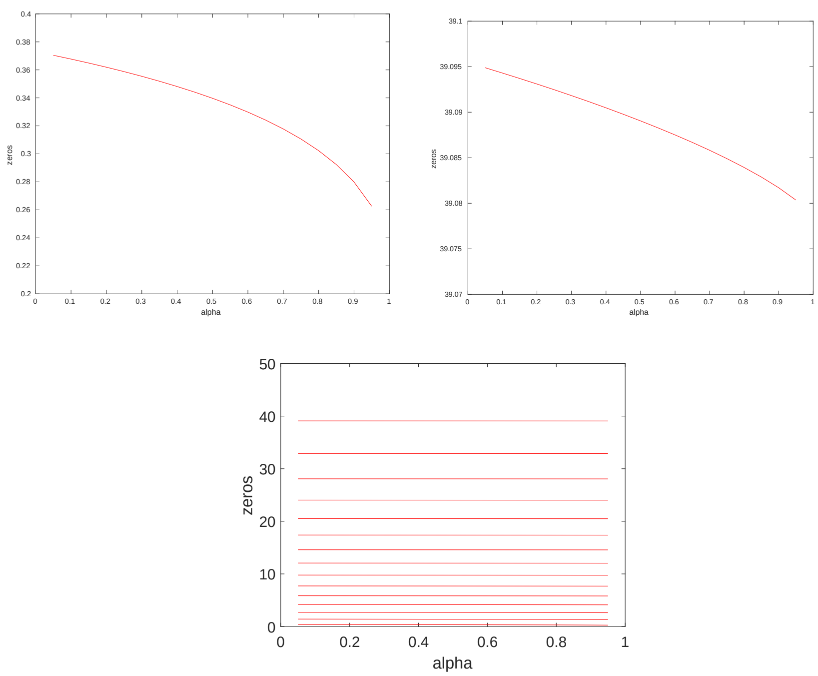

Figure 6: The positive zeros for $n = 30$ of the squared generalized Binet polynomials in dependence of the parameter α, $0 < \alpha < 1$; the smallest and largest positive zero (top), all positive zeros (bottom)

4 Halfrange Binet and halfrange generalized Binet weight functions

4.1 Halfrange Binet weight function

The halfrange Binet weight function is the weight function (1) supported on $[0,\infty]$. Its moments are (cf. Eqs. (4), (5))

$$\mu_k = k!\,\mathrm{Li}_{k+2}(1), \quad k = 0, 1, 2, \ldots \ . \tag{23}$$

They are generated by the Matlab command `mom=smom_hrbinet(dig,N)`.

Example 5. The first 100 recurrence coefficients to 32 digits of the halfrange Binet weight function.

The Matlab command `[ab,dig]=dig_hrbinet(N,dig0,dd,nofdig)`, when run with `N` $= 100$, `dig0` $= 116$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 124$. The same command, or more directly, the command `ab=sr_hrbinet(dig, nofdig,N)` with `dig` $= 124$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 7, the first showing the halfrange Binet weight function. They are also made available in the text file `coeff_hrbinet.txt`; see also [4, 2.9.23].

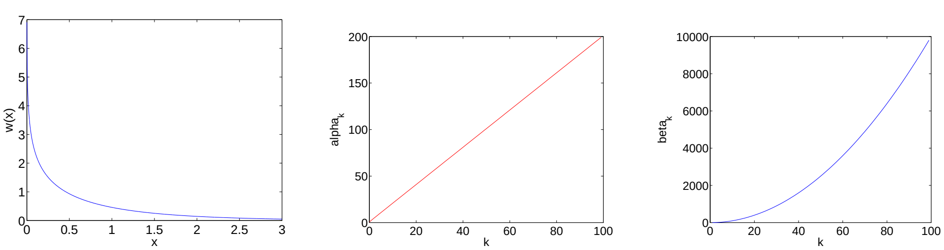

Figure 7: Halfrange Binet weight function and its recurrence coefficients

4.2 Halfrange generalized Binet weight function

4.2.1 Recurrence coefficients

The halfrange generalized Binet weight function is the weight function (2) supported on $[0,\infty]$. Its moments are (cf. Eqs. (9), (10))

$$\mu_k = k!\,S_{k+1,1}(\alpha) = k!\,\mathrm{Li}_{k+2}(\alpha), \quad k = 0, 1, 2, \ldots \ . \tag{24}$$

They are generated by the Matlab command `mom=smom_hrgbinet(dig,N,a)`, `a` $= \alpha$.

Example 6. The first 100 recurrence coefficients to 32 digits of the halfrange generalized Binet weight function for $\alpha = 1/2$.

The Matlab command `[ab,dig]=dig_hrgbinet(N,a,dig0,dd,nofdig)`, when run with `N` $= 100$, `a` $= 1/2$, `dig0` $= 120$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 128$. The same command, or more directly, the command `ab=sr_hrgbinet(dig,nofdig,N,a)` with `dig` $= 128$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 8, the first showing the halfrange generalized Binet weight function for $\alpha = 1/2$. They are also made available in the text file `coeff_hrgbinet.txt`.

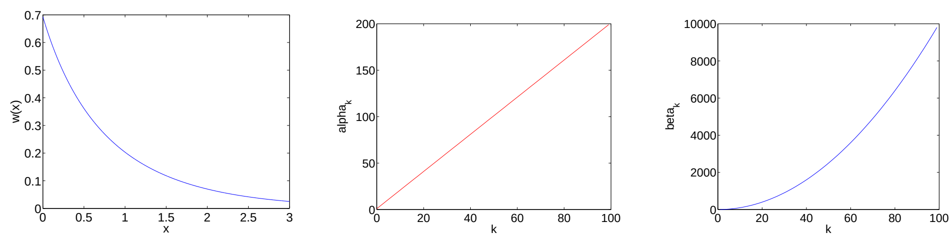

Figure 8: Halfrange generalized Binet weight function and its recurrence coefficients

4.2.2 Zeros of the orthogonal polynomials

By what was proved in Subsection 2.2.2, all zeros of the halfrange generalized Binet polynomials decrease monotonically as functions of the parameter α.

For $n = 15$, the smallest and largest zero are shown in the first two plots of Fig. 9, and all zeros in the third plot; cf. `plot_zeros_hrgbinet.m`.

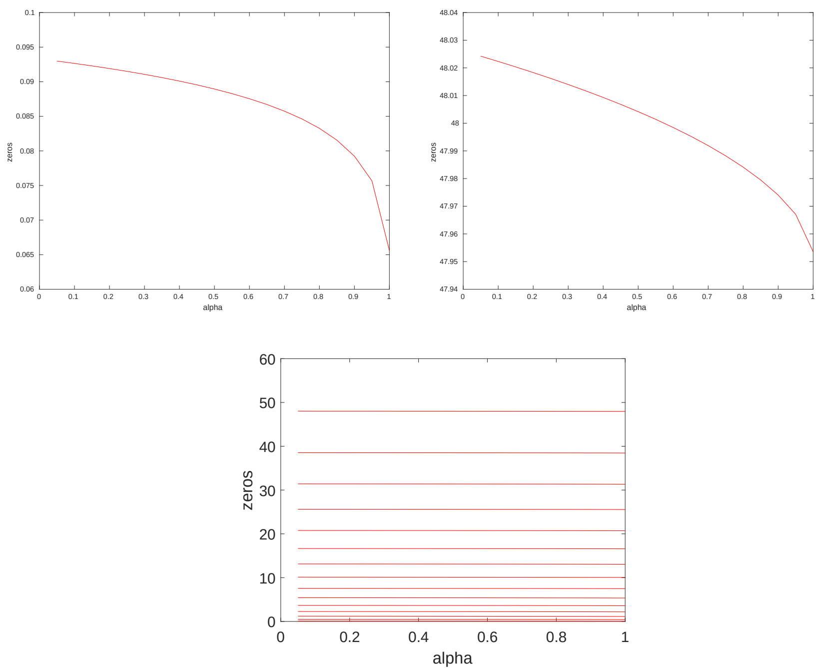

Figure 9: The zeros for $n = 15$ of the halfrange generalized Binet polynomials in dependence of the parameter α, $0 < \alpha \leq 1$; the smallest and largest zero (top), all zeros (bottom)

5 Halfrange squared Binet and halfrange squared generalized Binet weight functions

5.1 Halfrange squared Binet weight function

The halfrange squared Binet weight function is the weight function (16) supported on $[0, \infty]$. Its moments are (cf. Eqs. (17), (18))

$$\begin{aligned} &\mu_0 = 2\,\zeta(3), \\ &\mu_k = k!\left[(k+2)\zeta(k+3) - \sum_{\nu=2}^{k+1} \zeta(\nu)\zeta(k+3-\nu)\right], \quad k = 1, 2, 3, \dots\ . \end{aligned} \tag{25}$$

The first `N` of them are generated in `dig`-digit arithmetic by the Matlab command `mom=smom_hrsqbinet(dig,N)`.

Example 7. The first 100 recurrence coefficients to 32 digits of the halfrange squared Binet weight function.

The Matlab command `[ab,dig]=dig_hrsqbinet(N,dig0,dd,nofdig)`, when run with `N` $= 100$, `dig0` $= 160$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 168$. The same command, or more directly, the command `ab=sr_hrsqbinet(dig, nofdig,N)` with `dig` $= 168$, produces the $100{\times}2$ array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig, 10, the first showing the halfrange squared Binet weight function. They are also made available in the text file `coeff_hrsqbinet`.

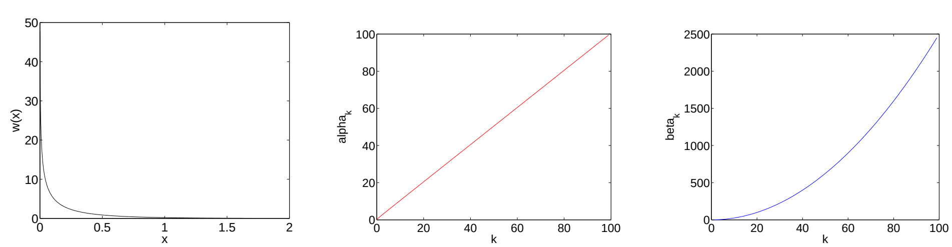

Figure 10: Halfrange squared Binet weight function and its recurrence coefficients

5.2 Halfrange squared generalized Binet weight function

5.2.1 Recurrence coefficients

The halfrange squared generalized Binet weight function is the weight function (19) supported on $[0, \infty]$. Its moments are (cf. Eqs. (20), (21))

$$\mu_k = 2\,k! S_{k=1,2}(\alpha), \quad k = 0, 1, 2, \dots \, . \tag{26}$$

They are generated by the Matlab command `mom=smom_hrsqgbinet(dig,N,a)`.

Example 8. The first 100 recurrence coefficients to 32 digits of the halfrange squared generalized Binet weight function for $\alpha = 1/2$.

The Matlab command `[ab,dig]=dig_hrsqgbinet(N,a,dig0,dd,nofdig)`, when run with `N` $= 100$, `a` $= 1/2$, `dig0` $= 116$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 124$. The same command, or more directly, the command `ab=sr_hrsqgbinet(dig,nofdig,N,a)` with `dig` $= 124$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 11, the first showing the halfrange squared generalized Binet weight function for $\alpha = 1/2$. They are also made available in the text file `coeff_hrsqgbinet`.

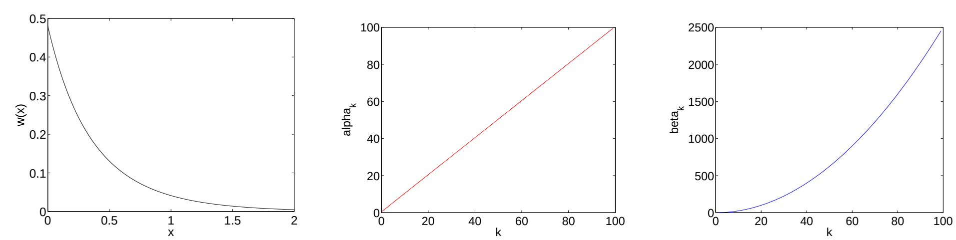

Figure 11: Halfrange squared generalized Binet weight function and its recurrence coefficients

5.2.2 Zeros of the orthogonal polynomials

Since, according to Subsection 4.2.2, all zeros of the halfrange generalized Binet polynomial are monotonically decreasing, the same is true, by Corollary 1 of Markov's theorem (cf. Subsection 2.2.2), for the square of the weight

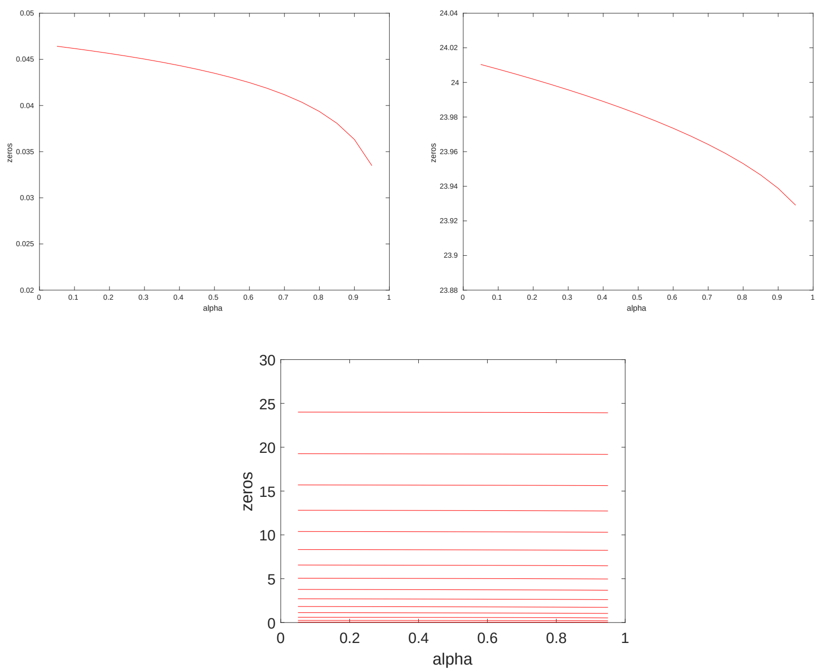

Figure 12: The zeros for $n = 15$ of the halfrange squared generalized Binet polynomials in dependence of the parameter α, $0 < \alpha < 1$; the smallest and largest zero (top), all zeros (bottom)

function. For $n = 15$, the smallest and largest zero are shown in the first two plots of Fig. 12, and all zeros in the third plot; cf. `plot_zeros_hrsqgbinet.m`.

6 Subrange Binet weight functions

6.1 An upper subrange Binet weight function

6.1.1 Recurrence coefficients

The weight function (1) is now assumed to be supported on the interval $[c, \infty]$, $0 < c < \infty$. The approach via moments,

$$\mu_k = -\int_c^\infty x^k \log(1 - e^{-x})\mathrm{d}x,$$

is still a valid option, giving, with the substitution of variables $t = e^{c-x}$,

$$\mu_k = \sum_{\nu=0}^{k} k^{(\nu)} c^{k-\nu} \mathrm{Li}_{\nu+2}(e^{-c}), \quad k = 0, 1, 2, \dots \, , \tag{27}$$

where

$$k^{(\nu)} = \begin{cases} 1 & \text{if } \nu = 0, \\ k(k-1)\cdots(k-\nu+1) & \text{if } \nu > 0, \end{cases}$$

is the descending factorial power and $\mathrm{Li}_n(x)$ the polylogarithm (cf. Subsection 2.1). The moments (27) are generated by the Matlab routine `smom_usrbinet.m`.

It is, however, considerably simpler, and hence faster, to make use of a linear translation of the upper subrange Binet weight function on $[c, \infty]$ to the halfrange generalized Binet weight function with parameter $\alpha = \mathrm{e}^{-c}$ (cf. Subsection 4.2). Denoting the recurrence coefficients of the latter by $a_k(\alpha)$, $b_k(\alpha)$, $k = 0, 1, 2, \dots$, it is easy to see that

$$\alpha_k = a_k(\alpha) + c, \quad \beta_k = b_k(\alpha), \quad k = 0, 1, 2, \dots, \quad \alpha = \mathrm{e}^{-c}. \tag{28}$$

The moments needed to generate the $a_k(\alpha)$, $b_k(\alpha)$ are then those in (24), which are definitely simpler than those in (27). They are produced by the Matlab command `mom=smom_usrbinet_alt(dig,N,c)`.

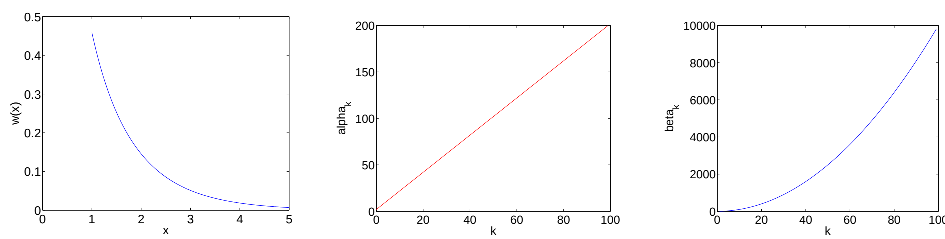

Figure 13: Upper subrange Binet weight function and its recurrence coefficients

Example 9. The first 100 recurrence coefficients to 32 digits of the upper subrange Binet weight function for $c = 1$.

The Matlab command `[ab,dig]=dig_usrbinet_alt(N,c,dig0,dd,nofdig)`, when run with `N` = 100, `c` = 1, `dig0` = 120, `dd` = 4, `nofdig` = 32, yields `dig` = 128. The same command, or more directly, the command `ab=sr_usrbinet_alt(dig,nofdig,N,c)` with $\texttt{dig} = 128$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 13, the first showing the upper subrange Binet weight function on $[1,\infty]$. They are also made available in the text file `coeff_usrbinet.txt`; see also [4, 2.9.25].

6.1.2 Zeros of the orthogonal polynomials

Our interest is now in the behavior of the zeros of the upper subrange Binet polynomials as functions of c. Here, the Remark to Theorem 2 of Subsection 2.2.2 applies, implying that all zeros are monotonically increasing. They are shown in Fig. 14.

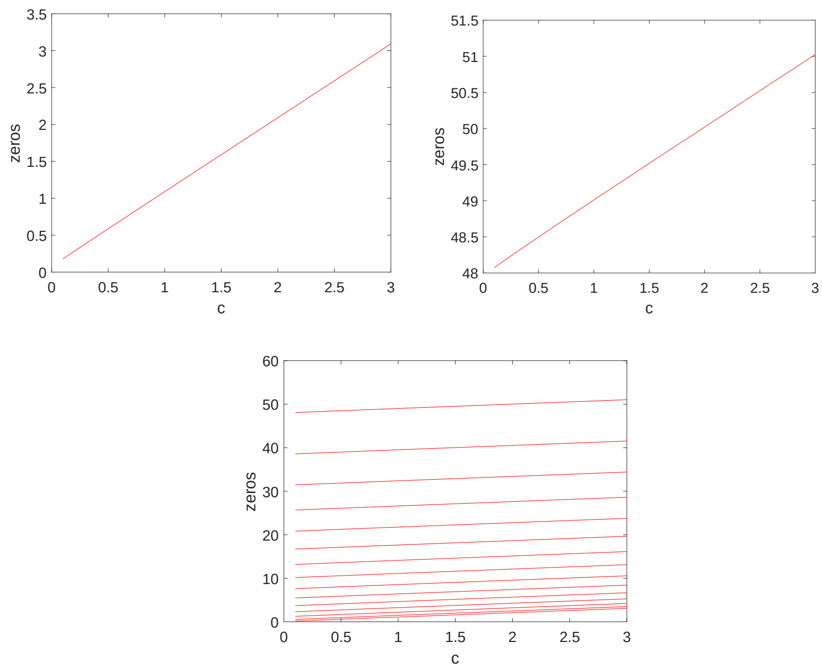

Figure 14: The zeros for $n = 15$ of the upper subrange Binet polynomials in dependence of the parameter c, $0 < c \leq 3$; the smallest and largest zero (top), all zeros (bottom)

6.2 A lower subrange Binet weight function

6.2.1 Recurrence coefficients

We consider here the weight function (1) supported on the interval $[0, c]$, $0 < c < \infty$. We take the simple approach of computing the respective moments as the difference between the halfrange and upper subrange moments,

$$\mu_k = \mu_k^{\mathrm{hr}} - \mu_k^{\mathrm{usr}}(c), \quad k = 0, 1, 2, \dots\ , \tag{29}$$

where μ_k^{hr} are the moments in (23), and $\mu_k^{\mathrm{usr}}(c)$ those in (27), although (29) may be subject to severe cancellation, especially if c is small. This must be compensated by an increase of the precision used to compute the moments.

The moments (29) are generated in `dig`-digit arithmetic by the Matlab command `mom=smom_lsrbinet(dig,N,c)`.

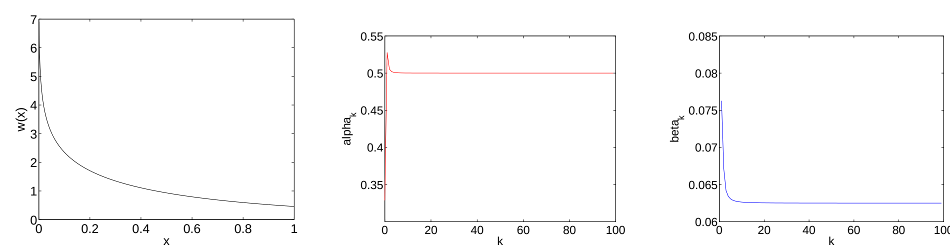

Figure 15: Lower subrange Binet weight function and its recurrence coefficients

Example 10. The first 100 recurrence coefficients to 32 digits of the lower subrange Binet weight function for $c = 1$.

The Matlab command `[ab,dig]=dig_lsrbinet(N,c,dig0,dd,nofdig)`, when run with `N` $= 100$, `c` $= 1$, `dig0` $= 520$, `dd` $= 4$, `nofdig` $= 32$, yields `dig` $= 528$. This large number of `dig` is due to extremely severe cancellation in (29), causing a loss of as many as 375 digits! The same command, or more directly, the command `ab=sr_lsrbinet(dig,nofdig,N,c)` with `dig` $= 528$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are depicted in the second and third plot of Fig. 15, the first showing the lower subrange Binet weight function on $[0, 1]$. They are also made available in the text file `coeff_lsrbinet.txt`; see also [4, 2.9.24].

6.2.2 Zeros of the orthogonal polynomials

By Theorem 2 of Subsection 2.2.2 all zeros of the lower subrange Binet polynomials are monotonically increasing. Using the routine `dig_lsrbinet.m`

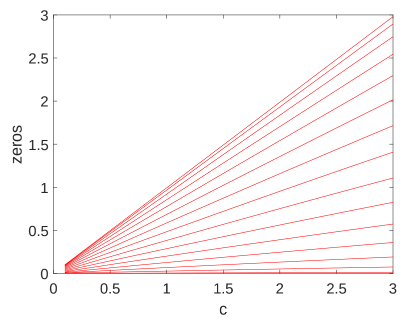

Figure 16: The zeros for $n = 15$ of the lower subrange Binet polynomials in dependence of the parameter c, $0 < c \leq 3$

with `N` $= 15$, it was found that `dig` $= 90$ digits are required to obtain the first 15 recurrence coefficients to an accuracy of 6 digits whenever $c \geq 1/10$. The plots produced by the routine `plot_zeros_lsrbinet.m` are shown in Fig. 16.

References

[1] Dahlquist, Germund. On summation formulas due to Plana, Lindelöf and Abel, and related Gauss–Christoffel rules, I. *BIT* 37 (1997), 256–295; II. *ibid.* 804–832; III. *ibid.* 39 (1999), 51–78.

[2] Gautschi, Walter. *Orthogonal polynomials: computation and approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2004.

[3] Gautschi, Walter. *Orthogonal polynomials in MATLAB: exercises and solutions*, Software, Environment, Tools, SIAM, Society of Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[4] Gautschi, Walter. *A software repository for orthogonal polynomials*, in preparation.

[5] GRADSHTEYN, I. S. AND I. M. RYZHIK. *Table of integrals, series, and products*, 5th ed., Academic Press, New York, 1994.

[6] KÖLBIG, K. S. Nielsen's generalized polylogarithyms, *SIAM J. Math. Anal.* 17 (1986), 1322–1358.

[7] KÖLBIG, K. S., J. A. MIGNACO, AND E. REMIDDI. On Nielsen's generalized polylogarithms and their numerical calculation, *BIT* 10 (1970), 38–74.

[8] SZEGÖ, GABOR. *Orthogonal polynomials*, 4th ed., Colloquium Publications, v. 23, American Mathematical Society, Providence, RI, 1975.