Numer Algor
DOI 10.1007/s11075-012-9556-z

ORIGINAL PAPER

Sub-range Jacobi polynomials

Walter Gautschi

Received: 3 February 2012 / Accepted: 28 February 2012

Abstract Orthogonal polynomials relative to the Jacobi weight function, but orthogonal on a strict subinterval of $[-1, 1]$, are studied, in particular with regard to their numerical computation. Related Gaussian quadrature rules are also considered.

Keywords Sub-range Jacobi polynomials · Computation of recurrence coefficients and special Gaussian quadrature rules

Mathematics Subject Classifications (2010) 33C47 · 65D20

1 Introduction

Let $w \not\equiv 1$ be one of the classical weight functions for orthogonal polynomials and I the associated interval of orthogonality. Given a strict subinterval $I_0 \subset I$, polynomials orthogonal on I_0 relative to w are here called *sub-range orthogonal polynomials* relative to w. Examples of such polynomials, occurring in physics and statistics, are the half-range Hermite polynomials orthogonal on $\mathbb{R}_+$ relative to the weight function $w(t) = e^{-t^2}$ (cf. [6, Examples 2.31 and 2.41], [5, Section 2.2(iii)]), the finite-range Hermite polynomials orthogonal with respect to the same weight function, but on $[-c, c]$ or $[0, c]$ $(c > 0)$, the latter being known as Maxwell's distribution, and the finite-range Laguerre polynomials orthogonal relative to the weight function $w(t) = e^{-t}$ on $[0, c]$ (cf. [5, Section 2.2,(iv)]).

W. Gautschi (✉)
Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-2066, USA
e-mail: wgautschi@purdue.edu

Recently, in connection with subperiodic trigonometric quadrature, Da Fies and Vianello [1] used sub-range Chebyshev polynomials orthogonal with respect to the weight function $w(t) = (1 - t^2)^{-1/2}$ on the interval $[-c, c]$, where $c = \sin(\omega/2)$, $0 < \omega < \pi$, and applied them to construct interesting Gaussian product formulae for integration over circular and annular sectors, circular zones, and circular lenses; see also [2]. In their computations they apply the modified Chebyshev algorithm (cf. [6, Section 2.1.7]) with Chebyshev moments of the weight function $(1 - c^2t^2)^{-1/2}$ on $[-1, 1]$; see [1, Section 2.1]. This work is specifically tailored to Chebyshev weight functions and rests heavily on the ability to accurately compute the modified moments—a nontrivial task. A more general, and at the same time simpler, approach is via discretization methods (cf. [6, Section 2.2.2]), which are used here to compute not only sub-range Chebyshev, but arbitrary sub-range Jacobi polynomials. The same method, in principle, can be applied to any weight function w.

2 Sub-range Jacobi polynomials and related Gaussian quadratures

Let

$$w(x) = (1 - x)^{\alpha}(1 + x)^{\beta}, \quad \alpha > -1,\ \beta > -1, \tag{2.1}$$

be the Jacobi weight function on $I = [-1, 1]$, and let the subinterval be $I_0 = [-c, c]$, $0 < c < 1$. Denote the corresponding (monic) sub-range Jacobi polynomials by $\pi_k(x)$, so that

$$\int_{-c}^{c} \pi_k(x)\pi_\ell(x)w(x)dx = 0 \quad \text{if } k \neq \ell.$$

Define

$$p_k(t) = \frac{1}{c^k}\pi_k(ct), \quad k = 0, 1, 2, \ldots . \tag{2.2}$$

These are monic polynomials satisfying the orthogonality relation

$$\int_{-1}^{1} p_k(t)p_\ell(t)w(ct)dt = 0 \quad \text{if } k \neq \ell, \tag{2.3}$$

and therefore satisfy a three-term recurrence relation

$$\begin{gathered} p_{k+1}(t) = (t - a_k)p_k(t) - b_k p_{k-1}(t), \quad k = 0, 1, 2, \ldots, \\ p_0(t) = 1, \quad p_{-1}(t) = 0, \end{gathered} \tag{2.4}$$

where by convention, $b_0 = \int_{-1}^{1} w(ct)dt$. The coefficients a_k, b_k (which depend on c) can easily be computed by the "general-purpose" discretization method described in [6, Section 2.2.2], using the software package OPQ[1] (for details,

---

[1] The package OPQ can be accessed at `OPQ.html` of the website http://www.cs.purdue.edu/archives/2002/wxg/codes, and all Matlab routines referenced in this paper at SRJAC of the same website.

see Section 3). Once they have been computed, the recurrence coefficients α_k, β_k of the sub-range Jacobi polynomials can be obtained by putting $t = x/c$ in (2.4) and multiplying through by c^{k+1}. In view of (2.2), i.e., $c^k p_k(x/c) = \pi_k(x)$, one gets

$$\alpha_k = ca_k, \quad k \geq 0; \qquad \beta_k = c^2 b_k, \quad k \geq 1. \tag{2.5}$$

Furthermore, by convention, $\beta_0 = \int_{-c}^{c} w(x)dx = c\int_{-1}^{1} w(ct)dt$, so that

$$\beta_0 = cb_0. \tag{2.6}$$

The first n recurrence coefficients a_k, b_k in (2.4) allow us to generate the n-point Gauss formula

$$\int_{-1}^{1} f(t)w(ct)dt = \sum_{k=1}^{n} w_k(c) f(t_k(c)), \qquad f \in \mathbb{P}_{2n-1}, \tag{2.7}$$

using the `OPQ` routine `gauss.m`. As c varies from 0 to 1, (2.7) describes a continuous transition from the Gauss–Legendre to the Gauss–Jacobi rule.

The same approach can be used to deal with asymmetric subintervals I_0, say, $I_0 = [-1, c]$, where $0 < c < 1$. Instead of the simple transformation $x = ct$, used in (2.2), one must use

$$x = x(t; c) := \frac{1}{2}(1 + c)t - \frac{1}{2}(1 - c), \quad -1 \leq t \leq 1.$$

The polynomials

$$p_k(t) = \frac{1}{\left[\frac{1}{2}(1 + c)\right]^k} \pi_k(x(t; c)), \quad k = 0, 1, 2, \ldots, \tag{2.8}$$

are then orthogonal on $[-1, 1]$ with respect to the weight function $w(x(t; c))$. If (2.4) is again their recurrence relation, with $b_0 = \int_{-1}^{1} w(x(t; c))dt = [\frac{1}{2}(1 + c)]^{-1} \int_{-1}^{c} w(x)dx$, we get for the recurrence coeficients α_k, β_k of the sub-range Jacobi polynomials

$$\alpha_k = a_k - \frac{1}{2}(1 - c), \quad k \geq 0; \qquad \beta_k = \left[\frac{1}{2}(1 + c)\right]^2 b_k, \quad k \geq 1, \tag{2.9}$$

and $\beta_0 = \int_{-1}^{c} w(x)dx = \frac{1}{2}(1 + c)b_0$.

For $c = 0$, we have

$$w(x(t; 0)) = 2^{-(\alpha+\beta)}(3 - t)^{\alpha}(1 + t)^{\beta}, \tag{2.10}$$

so that

$$\int_{-1}^{1} f(t)w(x(t; c))dt = \sum_{k=1}^{n} w_k(c) f(t_k(c)), \qquad f \in \mathbb{P}_{2n-1}, \tag{2.11}$$

now represents a continuous transition from the Gauss formula for the weight function (2.10) (when $c = 0$) to the Gauss–Jacobi formula (when $c = 1$).

3 Computational algorithms

The basic computational problem consists in computing, for any given integer n, the first n recursion coefficients a_k, b_k, $k = 0, 1, 2, \dots, n-1$, in (2.4). We propose to do this by (what in [4] is termed) *Stieltjes's procedure*, more precisely, by an appropriately discretized Stieltjes procedure.

If

$$(p, q) = \int_{-1}^{1} p(t)q(t)v(t; c)dt \tag{3.1}$$

is the underlying inner product (where $v(t; c) = w(ct)$ in the case of (2.3), and $v(t; c) = w(x(t; c))$ in the case of (2.8)), Stieltjes's procedure uses classical formulae for the recurrence coefficients a_k, b_k in terms of this inner product, in combination with the three-term recurrence relation (2.4) itself, to progressively generate the desired coefficients, starting with a_0, b_0 (cf. [6, Section 2.2.3.1]). It uses the inner product (3.1) only for p and q polynomials of progressively increasing degrees.

In the case $v(t; c) = w(ct)$, the integrand in (3.1) is a C^∞ function and thus amenable to numerical quadrature. As already suggested in [4], Fejér quadrature is quite suitable for this purpose and gives rise to the discrete inner product

$$(p, q)_N = \sum_{k=1}^{N} w_k^F w\left(ct_k^F\right) \cdot p\left(t_k^F\right) q\left(t_k^F\right), \tag{3.2}$$

where t_k^F, w_k^F are the nodes and weights of he N-point Fejér rule, that is, the interpolatory quadrature rule based on the N Chebyshev points $t_k^F = \cos\left(\frac{2k-1}{2N}\pi\right)$, $k = 1, 2, \dots, N$. The *discretized Stieltjes procedure* is simply Stieltjes's procedure in which (3.1) is replaced by (3.2). This produces certain approximations, $a_{k,N}$, $b_{k,N}$, $k = 0, 1, \dots, n-1$, of the desired recursion coefficients. Since Fejér's rule is interpolatory and positive, the procedure converges in the sense

$$a_{k,N} \to a_k, \quad b_{k,N} \to b_k, \quad k = 0, 1, \dots, n-1, \quad \text{as } N \to \infty. \tag{3.3}$$

The discretized Stieltjes procedure is implemented in the `OPQ` routine `mcdis.m` (multiple-component discretization), which allows for the interval of integration to be decomposed into any number of subintervals prior to discretization. In the present case, the one-component procedure suffices, i.e., Fejér's rules can be applied to the whole interval $[-1, 1]$. The Matlab routine that implements this procedure in the case of the sub-range Jacobi polynomials is `r_subjacobi.m` and is called by

$$\texttt{[ab, Ncap]} = \texttt{r\_subjacobi(n, eps0, c, alpha, beta)}. \tag{3.4}$$

Here, the input parameters are `n`—the number n of desired recursion coefficients, `eps0`—the desired relative accuracy, `alpha`, `beta`—the Jacobi parameters α, β, and `c`—the parameter c. The output is the $n \times 2$ array `ab` containing in the first column the n coefficients $\{a_k\}_{k=0}^{n-1}$ and in the second column $\{b_k\}_{k=0}^{n-1}$. The (optional) output parameter `Ncap` is a value of N in (3.3) that achieves the accuracy `eps0`.

In the case $v(t; c) = w(x(t; c))$, the matter is a little bit more complicated, since

$$v(t; c) = \left[\frac{1}{2}(1+c)\right]^{\alpha+\beta} \left(\frac{3-c}{1+c} - t\right)^{\alpha} (1+t)^{\beta}, \quad -1 < t < 1, \tag{3.5}$$

and the last factor has an algebraic singularity at $t = -1$ if β is not an integer. This calls for a two-component discretization, splitting the integral in (3.1) into two parts, one extended from -1 to 0, the other from 0 to 1. To discretize the second part, we can again apply the Fejér rule (transformed to the interval $[0, 1]$). For the first part (disregarding for the moment the constant factor) we must use Gauss–Jacobi quadrature on $[0, 1]$ by writing

$$\int_{-1}^{0} p(t)q(t) \left(\frac{3-c}{1+c} - t\right)^{\alpha} (1+t)^{\beta} dt = \int_{0}^{1} p(-x)q(-x) \left(\frac{3-c}{1+c} + x\right)^{\alpha} (1-x)^{\beta} dx$$

and applying to the second integral the Gauss–Jacobi rule on $[0, 1]$ with Jacobi parameters β and 0 (and obvious notation):

$$\begin{aligned} &\int_{0}^{1} p(-x)q(-x) \left(\frac{3-c}{1+c} + x\right)^{\alpha} (1-x)^{\beta} dx \\ &\qquad \approx \sum_{k=1}^{N} w_k^{GJ} p\left(-x_k^{GJ}\right) q\left(-x_k^{GJ}\right) \left(\frac{3-c}{1+c} + x_k^{GJ}\right)^{\alpha}. \end{aligned}$$

Multiplying this by the constant factor in (3.5) and adding the result to the N-point Fejér discretization of the second part,

$$\left[\frac{1}{4}(1+c)\right]^{\alpha+\beta} \sum_{k=1}^{N} \frac{1}{2} w_k^F p\left(\frac{1}{2}\left(t_k^F + 1\right)\right) q\left(\frac{1}{2}\left(t_k^F + 1\right)\right) \left(\frac{5-3c}{1+c} - t_k^F\right)^{\alpha} \left(3 + t_k^F\right)^{\beta},$$

will give the appropriate discrete inner product $(p,q)_N$. The routine `r_subjacobi0`, analogous to the routine (3.4), implements the discretized Stieltjes procedure in this case.

4 Numerical results

We first illustrate the transition from Gauss–Legendre to Gauss–Jacobi quadrature in (2.7). We take $n = 20$ and $\alpha = \beta = -1/2$, and for 100 equally-spaced values of c between 0 and 1 compute and plot the nodes and weights of the n-point Gauss formula (2.7) as functions of c. The results, produced by the routine `transG.m`, are shown in Fig. 1. It can be seen that the nodes are more or less constant, while the weights grow monotonically from the Legendre weights to the (constant) Chebyshev weights $w_k = \pi/n = .15707\ldots$, $k = 1, 2, \ldots, n$. Because of symmetry there are only $n/2 = 10$ curves in Fig. 1b. Figure 2 shows the analogous results for $\alpha = -\beta = -1/2$. Here, each Legendre weight (for $c = 0$) splits into a pair of distinct weights as c moves away from zero, one monotonically decreasing, the other monotonically increasing.

The analogous transitions for the Gauss formulae (2.11) are depicted in Figs. 3 and 4, which are produced by the same routine `transG.m`.

We next illustrate the "circle theorem" for Gauss (and other) quadratures, first enunciated by Davis and Rabinowitz [3] and later extended in [7]. According to this theorem, the nodes and weights of (2.7) and (2.11) satisfy

$$\frac{nw_k}{\pi w(ct_k)} \sim \sqrt{1-t_k^2} \text{ resp. } \frac{nw_k}{\pi w(x(t_k;c))} \sim \sqrt{1-t_k^2} \quad \text{as } n \to \infty \tag{4.1}$$

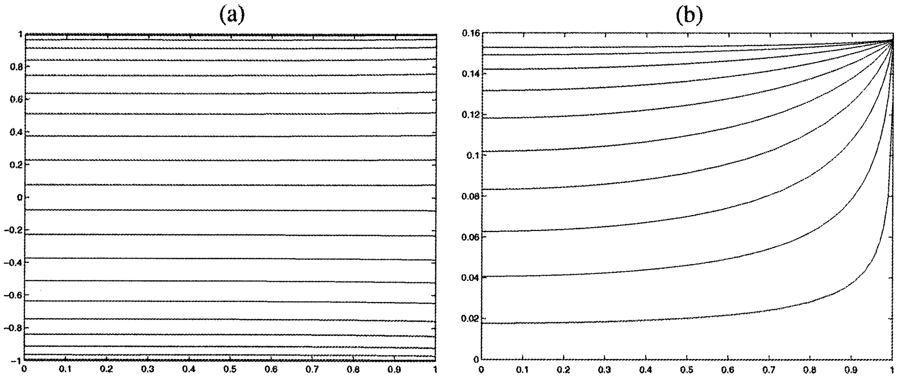

Fig. 1 Nodes (**a**) and weights (**b**) of (2.7) for $0 < c < 1$ when $\alpha = \beta = -1/2$

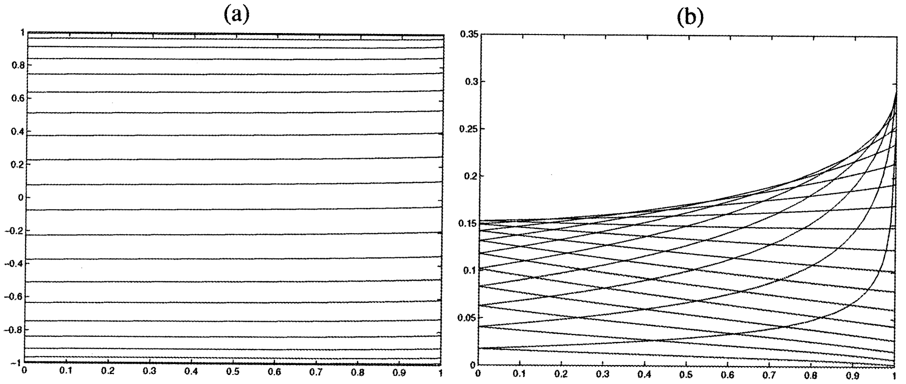

Fig. 2 Nodes (**a**) and weights (**b**) of (2.7) for $0 < c < 1$ when $\alpha = -\beta = -1/2$

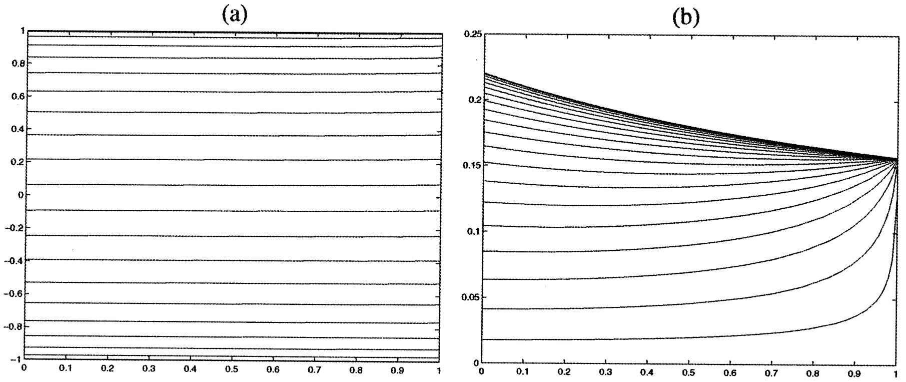

Fig. 3 Nodes (**a**) and weights (**b**) of (2.11) for $0 < c < 1$ when $\alpha = \beta = -1/2$

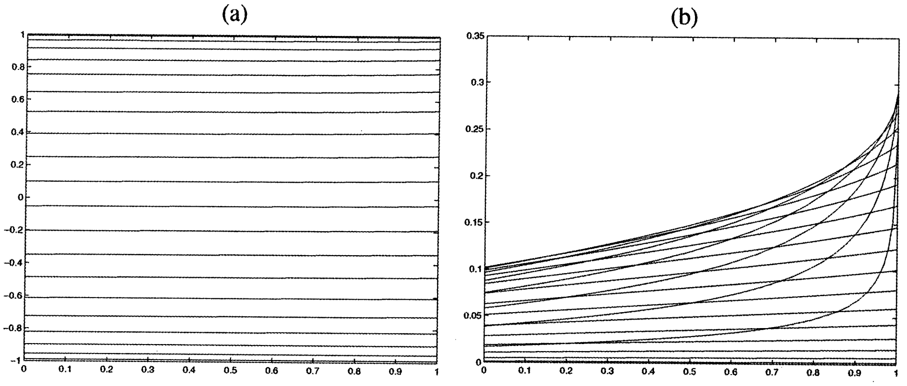

Fig. 4 Nodes (**a**) and weights (**b**) of (2.11) for $0 < c < 1$ when $\alpha = -\beta = -1/2$

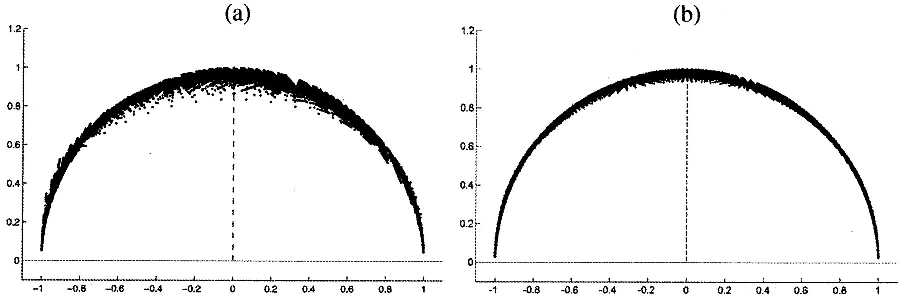

Fig. 5 The circle theorem for sub-range Jacobi weight functions

for all nodes t_k that lie in any compact subinterval of $[-1, 1]$. In Fig. 5, the quantities on the far left in (4.1) for the Gauss formula (2.7) are plotted against the nodes t_k for all α, β in the set $\mathcal{D} = \{\alpha, \beta = [-.9, -.7, -.5, -.3, -.1, 0, .1, .3, .5, .7, .9, 1.5, 3, 6], \beta \geq \alpha\}$, and $c = .1 : .1 : .9$, for $n = 20 : 5 : 40$ in (a), and for $n = 60 : 5 : 80$ in (b). The routine producing these graphs is `circle_thm.m`. For the Gauss formula (2.11) the results, as expected, are practically identical.

In the case of sub-range Chebyshev polynomials (cf. Section 1), we have checked our method against the one used in [1] for $c = \sin(\omega/2)$, $\omega = \pi/2, 9\pi/10, 99\pi/100$, and $n = 5 : 5 : 20$, $30 : 10 : 100$, 200, 300, and found excellent agreement; see `testFV.m`.

We also checked the performance of our routine `r_subjacobi.m` on sub-range Jacobi polynomials with $\alpha, \beta \in \mathcal{D}$. For the values of n and c shown in Table 1, we list Ncap, the maximum values of N over all $(\alpha, \beta) \in \mathcal{D}$ that yield a relative accuracy of `eps0` $= .5 \times 10^{-12}$. These values are by no means sharp, since convergence is checked in increments of N that become larger as N increases. Table 1 was produced by the routine `runsubjac.m`. Similar, often more favorable, results are obtained for the routine `r_subjacobi0`.

Table 1 Convergence behavior of the discretized Stieltjes procedure

$n = 10$		$n = 55$		$n = 100$		$n = 200$		$n = 300$	
c	Ncap	c	Ncap	c	Ncap	c	Ncap	c	Ncap
.100	41	.100	221	.100	401	.100	801	.100	1201
.500	51	.500	221	.500	401	.500	801	.500	1201
.900	91	.900	221	.900	401	.900	801	.900	1201
.990	231	.990	386	.990	501	.990	801	.990	1201
.999	591	.999	606	.999	901	.999	1001	.999	1501

Acknowledgement The author is grateful to Professor M. Vianello for providing him with a prepublication copy of [1] and related software.

References

1. Da Fies, G., Vianello, M.: Trigonometric Gaussian quadrature on subintervals of the period (2011). http://www.math.unipd.it/~marcov/pdf/trigauss.pdf. Accessed 8 March 2012
2. Da Fies, G., Vianello, M.: Algebraic cubature on planar lenses and bubbles. Dolomites Res. Notes Approx. **5**, 7–12 (2012)
3. Davis, P.J., Rabinowitz, P.: Some geometrical theorems for abscissas and weights of Gauss type. J. Math. Anal. Appl. **2**, 428–437 (1961)
4. Gautschi, W.: On generating orthogonal polynomials. SIAM J. Stat. Comput. **3**, 289–317 (1982)
5. Gautschi, W.: Computational aspects of orthogonal polynomials. In: Nevai, P. (ed.) Orthogonal Polynomials. NATO ASI Series, Series C: Mathematical and Physical Sciences, vol. 294, pp. 181–216. Kluwer, Dordrecht (1990)
6. Gautschi, W.: Orthogonal Polynomials: Computation and Approximation. Numerical Mathematics and Scientific Computation, Oxford University Press, New York (2004)
7. Gautschi, W.: The circle theorem and related theorems for Gauss-type quadrature rules. Electr. Trans. Numer. Anal. **25**, 129–137 (2006)