Contemporary Mathematicians

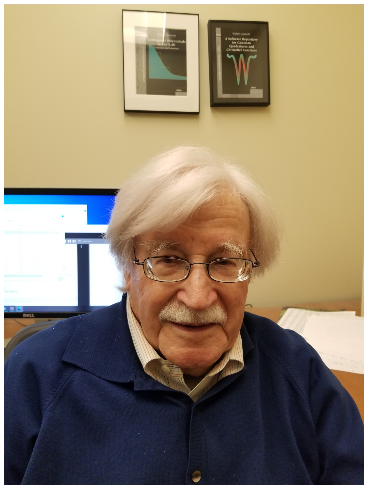
Joseph P.S. Kung
University of North Texas, USA
Editor

For further volumes:
http://www.springer.com/xeries/4817

Claude Brezinski • Ahmed Sameh
Editors

Walter Gautschi, Volume 4

Selected Works with Commentaries



Walter Gautschi, 2021

Contents

List of Contributors . . . . **xiii**

34 Publications **1**
Walter Gautschi

Part I Commentaries

35 Special Functions **7**
Javier Segura

35.1 Computational methods 7
35.1.1 Exponential integral 7
35.1.2 Repeated integrals of the coerror function 8
35.2 Zeros 8
35.2.1 Freud and subrange Freud polynomials 8
35.2.2 Ultraspherical polynomials 9
35.2.3 Subrange Jacobi polynomials 9
References 9

36 Approximation **11**
Miodrag M. Spalević

37 Orthogonal Polynomials on the Real Line **13**
Walter Van Assche

37.1 Polynomials orthogonal with respect to exponential integrals 13
37.2 Polynomials orthogonal with respect to cardinal B-spline weight functions 16
37.3 Binet-type polynomials and their zeros 17
37.4 Polynomials orthogonal relative to weight functions of Prudnikov type 19

37.5 Orthogonal polynomials relative to a generalized Marchenko–Passtur probability measure 21
37.6 Another look at polynomials orthogonal relative to exponential integral weight functions 23
References 24

38 Gauss-type Quadrature 27
Gradimir V. Milovanović

38.1 Introduction 27
38.2 Gauss–Turán quadrature 27
38.3 Gauss–Turán quadrature for Laguerre and Hermite weight functions 30
References 32

39 Ordinary Differential Equations 33
John Butcher

40 History 35
Gerhard Wanner

40.1 A historical note on Gauss–Kronrod quadrature 35
40.2 Interpolation before and after Lagrange 35
References 36

41 Miscellanea 37
Martin J. Gander

Part II Reprints

42 Papers on Special Functions **41**

42.1 [172] (with F. E. Harris and N. M. Temme), Expansions of the exponential integral in incomplete gamma functions, *Appl. Math. Lett.* 16, 1095–1099 (2003) 42

42.2 [212] Algorithm 957: Evaluation of the repeated integrals of the coerror function by halfrange Gauss–Hermite quadrature, *ACM Trans. Math. Software* 42, Article 9, 10 pages (2016) 48

42.3 [213] Monotonicity properties of the zeros of Freud and sub-range Freud polynomials, *Math. Comp.* 86, 855–864 (2017) 59

42.4 [216] On the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials, in *Frontiers in orthogonal polynomials and q-series*, 251–266, *World Sci. Publ.*, Hackensack, NJ, 2018 70

42.5 [218] On the zeros of subrange Jacobi polynomials, *Numer. Algorithms* 79, 759–768 (2018) 87

43 Paper on Approximation **99**

43.1 [217] A discrete top-down Markov problem in approximation theory, in *Frontiers in orthogonal polynomials and q-series* (M. Zuhair Nashed and Xin Li, eds.), 267–289, *World Sci. Publ.*, Hackensack, NJ, 2018 100

44 Papers on Orthogonal Polynomials on the Real Line **125**

44.1 [211] Polynomials orthogonal with respect to exponential integrals, *Numer. Algorithms* 70, 215–226 (2015) 126

44.2 [215] Polynomials orthogonal with respect to cardinal B-spline weight functions, *Numer. Algorithms* 76, 1099–1107 (2017) 139

44.3 [219] (with Gradimir V. Milovanović) Binet-type polynomials and their zeros, *Electron. Trans. Numer. Anal.* 50, 52-70 (2018) 149

44.4 [223] (with Gradimir V. Milovanović) Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure, *Numer. Algorithms* 88, 1233–1249 (2021) 169

44.5 [224] (with Gradimir V. Milovanović) Orthogonal polynomials relative to weight functions of Prudnikov type, *Numer. Algorithms*, to appear 187

44.6 [225] Another look at polynomials orthogonal relative to exponential integral weight functions, *Numer. Algorithms*, to appear 198

45 Paper on Gauss-type Quadrature 207

45.1 [209] High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions, *Numer. Algorithms* 67, 59–72 (2014) 208

46 Papers on Ordinary Differenial Equations 223

46.1 [15] (with H. A. Antosiewicz) Numerical methods in ordinary differential equations, Ch. 9 in *Survey of numerical analysis* (J. Todd, ed.), 314–346, McGraw–Hill, New York, 1962 224

46.2 [56] Comportement asymptotique des coefficients dans les formules d'intégration d'Adams, de Störmer et de Cowell, *Paris Sér. A-B*, 283, no. 10, Aiii, A787–A788, 1976 258

47 Papers on History 261

47.1 [177] A historical note on Gauss–Kronrod quadrature, *Numer. Math.* 100, 483–484 (2005) 262

47.2 [206] Interpolation before and after Lagrange, *Rend. Semin. Mat. Univ. Politec. Torino* 70, 347–368 (2012) 265

48 Paper on Miscellanea 289

48.1 [221] (with Ernst Hairer) On conjectures of Stenger in the theory of orthogonal polynomials, *J. Inequal. Appl.*, Paper No. 159 (2018), 27 pp. 290

List of Contributors

Walter Van Assche
Department of Mathematics
KU Leuven, Heverlee, Belgium

John C. Butcher
Department of Mathematics
The Univsersity of Auckland
Auckland, New Zealand

Martin Gander
Section de Mathématiques
Université de Genève
Genève, Switzerland

Nick Higham
School of Mathematics
The University of Manchester
Manchester, UK

Jacob Korevaar
Korteveg de Vries Institut
University of Amsterdam
Amsterdam, The Netherlands

Lisa Lorentzen
Institutt for Matematiske
Fag NTNU
Trondheim, Norway

Gradimir Milovanović
Matematički Institut SANU
Beograd, Serbia

Giovanni Monegato
Dipartimento di Matematica
Politecnico di Torino
Torino, Italy

Lothar Reichel
Department of Mathematical Sciences
Kent State University
Kent, OH, USA

Javier Segura
Departamento de Matematicas
Estadistica y Computacion
Universidad de Cantabria
Santander, Spain

Miodrag M. Spalević
Department of Mathematics
University of Belgrade
Belgrade, Serbia

Gerhard Wanner
Section de Mathématiques
Université de Genève
Genève, Switzerland

34

Publications (continuation of Section 4, Volume 1, including updated entries for the years 2012–2014)

Walter Gautschi

Books

B4. *Orthogonal polynomials in Matlab: exercises and solutions*, Software, Environments, Tools, SIAM, Philadelphia, 2016.

B5. *A software repository for orthogonal polynomials*, Software, Environments, Tools, SIAM, Philadelphia, 2018.

B6. *A software repository for Gaussian quadratures and Christoffel functions*, Software, Environments, Tools, SIAM, Philadelphia, 2021.

Publications

2012

204. *Numerical integration over the square in the presence of algebraic/logarithmic singularities with an application to aerodynamics*, Numer. Algorithms 61, 275–290.
205. *Sub-range Jacobi polynomials*, Numer. Algorithms 61, 649–657.
206. *Interpolation before and after Lagrange*, Rend. Semin. Mat. Univ. Politec. Torino 70, 347–368.

2013

207. *Repeated modifications of orthogonal polynomials by linear divisors*, Numer. Algorithms 63, 369–383.

208. *Neutralizing nearby singularities in numerical quadrature*, Numer. Algorithms 64, 417–425.

2014

209. *High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions*, Numer. Algorithms 67, 59–72.
210. *A brief summary of my scientific work and highlights of my career*, in *Walter Gautschi: selected works and commentaries, Vol.* 1 (C. Brezinski and A. Sameh, eds.), 9–17, Birkhäuser, Boston, MA.

2015

211. *Polynomials orthogonal with respect to exponential integrals.* Numer. Algorithms 70, 215–226.

2016

212. *Algorithm* 957: *Evaluation of the repeated integrals of the coerror function by half-range Gauss–Hermite quadrature*, ACM Trans. Math. Software 42, Article 9, 10 pages.

2017

213. *Monotonicity properties of the zeros of Freud and sub-range Freud polynomials: analytic and empirical results*, Math. Comp. 86, 855–864.
214. *Erratum to: Sub-range Jacobi polynomials*, Numer. Algorithms 74, 637.
215. *Polynomials orthogonal with respect to cardinal B-spline weight functions*, Numer. Algorithms 76, 1099–1107.

2018

216. *On the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials*, in *Frontiers in orthogonal polynomials and q-series* (M. Zuhair Nashed and Xin Li, eds.), 251–266, World Sci. Publ., Hackensack, NJ.
217. *A discrete top-down Markov problem in approximation theory*, in *Frontiers in orthogonal polynomials and q-series* (M. Zuhair Nashed and Xin Li, eds.), 267–289, World Sci. Publ., Hackensack, NJ.

218. *On the zeros of sub-range Jacobi polynomials*, Numer. Algorithms 79, 759–768.
219. (with Gradimir V. Milovanović) *Binet-type polynomials and their zeros*, Electron. Trans. Numer. Anal. 50, 52–70.
220. *Progress by accident: some reflections on my career*, SIAM News 51, no. 10, 7–8.

2019

221. *Kommentar* (*Interpolation des Logarithmus*) *zum Brief Leonhard Eulers an Daniel Bernoulli*, Opera Omnia IV/A3, Birkhäuser, Basel.
222. (with Ernst Hairer) *On conjectures of Stenger in the theory of orthogonal polynomials*, J. Inequal. Appl., Paper No. 159, 27 pp.

2021

223. (with Gradimir V. Milovanović) *Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure*, Numer. Algorithms 88, 1233–1249.
224. (with Gradimir V. Milovanović) *Orthogonal polynomials relative to weight functions of Prudnikov type*, Numer. Algorithms, to appear.
225. *Another look at polynomials orthogonal relative to exponential integral weight functions*, Numer. Algorithms, to appear.

Part I

Commentaries

In all commentaries, reference numbers preceded by "GA" refer to the numbers in the list of Gautschi's publications; see Section 4, Vol. 1 and Section 34, this volume. Numbers in boldface type indicate that the respective papers are included in these selected works.

35

Special functions

Javier Segura

Here, five papers are commented on, [GA**172**], [GA**212**], [GA**213**], [GA**216**], and [GA**218**] (see p. 31 of Vol. 1 and pp. 2–3 of this volume), all dealing with different aspects of special functions. They can be divided into two groups, one on computational methods and the other on zeros. Among the former are the papers [GA**172**] and [GA**212**], the first discussing the computation of the exponential integral by means of series expansion in incomplete gamma functions and the second the computation of the repeated integrals of the coerror function by halfrange Gauss–Hermite quadrature. In the second group of papers, the focus is on monotonicity properties, both analytic and empirical, for the zeros of classical and nonclassical orthogonal polynomials.

35.1 Computational methods

35.1.1 Exponential integral. In the paper [GA**172**], written jointly with F. E. Harris and N. M. Temme, the expansion of the exponential integral

$$E_1(z) + z + \ln\, z = \sum_{n=1}^{\infty} \frac{\gamma(n,z)}{n!} \tag{35.1}$$

in incomplete gamma functions is derived as a limit case (first $\lambda \downarrow 0$, then $a \to 0$) from the interesting identity

$$\gamma(a, \lambda z) = \lambda^a \sum_{n=0}^{\infty} \frac{\gamma(a+n, z)}{n!} (1-\lambda)^n$$

stated by Tricomi in [4] without proof, but proved in this paper. When compared with the known power series expansion

$$E_1(z) + z + \ln z = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{z^n}{n\, n!}\,, \tag{35.2}$$

one finds that they nicely complement one another in the sense that for large $|z|$, internal cancellations of terms in the series of (35.1) is negligent when $\arg z \approx 0$ but gets worse as $\arg z$ grows from 0 to π, whereas for the series in (35.2) it is just the other way around.

35.1.2 Repeated integrals of the coerror function. Computation of the repeated integrals of the coerror function,

$$f_n(x) = \imath^n \operatorname{erfc}(x), \quad n = 0, 1, 2, \ldots\,,$$

using recursion, has been considered by Gautschi previously in [GA**13**], [GA59], [GA60]; see §6.1.1 of Vol. 1. In [GA**212**] an entirely different approach is taken, namely numerical quadrature based on the integral representation

$$f_n(x) = \frac{2}{\sqrt{\pi}\, n!} \int_0^\infty t^n\, \mathrm{e}^{-(x^2+2xt)}\, \mathrm{e}^{-t^2} \mathrm{d}t,$$

where the integral is evaluated by halfrange Gauss–Hermite quadrature. This allows the computation of $f_n(x)$ for just one, or a few, values of n.

The potential occurrence of underflow and overflow, and the fact that the proposed (nonclassical) quadrature rules are currently available only up to 200 quadrature points and up to an accuracy of 32 digits, impose certain restrictions on the (n,x)-domain. These are carefully analyzed for the case where $f_n(x)$ is wanted to an accuracy of 12 or 30 decimal digits.

35.2 Zeros

35.2.1 Freud and subrange Freud polynomials. For zeros of an orthogonal polynomial that depends on a parameter τ, there is a classical result of A. Markov [3, Theorem 6. 12. 1] establishing monotonic growth of the zeros as a function of the parameter. The paper GA[**213**] provides concrete examples of this result related to subrange Freud polynomials orthogonal with respect to the weight function

$$w(x) = |x|^\mu\, \mathrm{e}^{-|x|^\nu}, \quad \mu > -1,\ \nu > 0$$

on intervals $x \in [0, c]$, $x \in [c, \infty]$, $x \in [-c, c]$, $x \in [-\infty, -c] \cup [c, \infty]$, where $c > 0$. The parameter in question here is $\tau = \nu$. Additional results, also for Freud polynomials on $x \in \mathbb{R}$, having a wider scope, are obtained by numerical computation.

35.2.2 Ultraspherical polynomials. Monotonicity and other properties for zeros of ultraspherical polynomials have received a great deal of attention in the literature. We mention, e. g., the work of Shafique, Muldoon, and Spigler [2] and that of Elbert and Siafarikas [1], culminating in the Ismail–Letessier–Askey (ILA) conjecture, proved in [2]. Gautschi's paper [GA**216**] examines to what extent this work can be extended, at least computationally, from monotonicity to complete monotonicity, which gives rise to a number of conjectures.

35.2.3 Subrange Jacobi polynomials. Subrange Jacobi polyhnomials are orthogonal with respect to the Jacobi weight function $w(x) = (1-x)^{\alpha}(1+x)^{\beta}$, $\alpha > -1$, $\beta > -1$, supported either on the symmetric subinterval $[-c, c]$, $0 < c < 1$, or on the asymmetric interval $[-1, c]$, $-1 < c < 1$. We may refer to them as symmetric resp. asymmetric subrange Jacobi polynomials. In the asymmetric case, it follows from Theorem 2.6 in [GA**219**] that all zeros of asymmetric subrange Jacobi polynomials are monotonically increasing as functions of c. In Theorem 1 of [GA**218**] the same is shown to be true in the symmetric case for all positive zeros if $\alpha = \beta$. The case $\alpha \neq \beta$ is more complicated. Assuming, without restriction of generality, that $\alpha < \beta$, the matter depends on the validity of a conjecture that is of independent interest, namely

$$\left[\frac{\pi_n(-c)}{\pi_n(c)}\right]^2 \left(\frac{1-c}{1+c}\right)^{\beta-\alpha} < 1, \quad n \geq 1,\ \alpha < \beta,\ 0 < c < 1, \tag{35.3}$$

where $\pi_n(\,\cdot\,)$ is the symmetric subrange Jacobi polynomial of degree n. If true, then the upper half of the zeros of π_n, if they are positive, increase monotonically as functions of c; cf. Theorem 2 in [GA**218**]. The conjecture (35.3) is known to be valid if $\alpha \leq 0$ and $\beta > 0$ (see Remark 2 on p. 763 of [GA**218**]).

References

[1] Elbert, Árpád and Panayiotis D. Siafarikas. Monotonicity properties of the zeros of ultraspherical polynomials, *J. Approx. Theory* **97** (1999), 3–39.

[2] Shafique, Ahmed, Martin E. Muldoon, and Renato Spigler. Inequalities and numerical bounds for zeros of ultraspherical polynomials, *SIAM J. Math. Anal.* **17** (1986), 1000–1007.

[3] Szegö, Gabor. *Orthogonal polynomials*, 4th ed., Colloquium Publ. 23, American Mathematical Society, Providence, RI, 1975.

[4] Tricomi, F. G. Asymptotische Eigenschaften der unvollständigen Gammafunktion, *Math. Z.* **53** (1950), 136–148.

36

Approximation

Miodrag M. Spalević

The paper here commented on is [GA**217**] (see p. 2 of this volume), an original contribution to approximation theory turning the classical Markov problem for a polynomial p of degree n around by seeking a bound of $|p|$ on $[-1, 1]$ if a bound of $\left|p^{(k)}\right|$ is known. Naturally, for this reversed Markov problem to be meaningful, k additional conditions for p and its first $k-1$ derivatives must be imposed. For these, Gautschi takes the initial conditions $p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0$. The problem then has an easy answer. More challenging and, at the same time, more interesting is a discrete version of this problem, which is what Gautschi is mainly dealing with here, namely

Problem (Discrete top-down Markov problem). *Given integers $n \geq 1$ and $1 \leq k \leq n$, and given $n-k+1$ distinct points $\mathbb{T}_n^{(k)} = \{\tau_\nu\}$ in $[-1, 1]$, $-1 \leq \tau_1 < \tau_2 < \cdots < \tau_{n-k+1} \leq 1$, consider the following class of polynomials of degree n,*

$$\begin{aligned}\mathbb{Q}_n^{(k)} = \{p \in \mathbb{P}_n : \ & p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0, \\ & \text{and } \left|p^{(k)}(\tau_\nu)\right| \leq 1, \ \nu = 1, 2, \ldots, n-k+1\}.\end{aligned}$$

For each $\nu = 1, 2, \ldots, n-k+1$, determine the maximum possible value $M_{n,\nu}^{(k)}$ of $|p(\tau_\nu)|$ when $p \in \mathbb{Q}_n^{(k)}$,

$$M_{n,\nu}^{(k)} = \max_{p \in \mathbb{Q}_n^{(k)}} |p(\tau_\nu)|, \quad \nu = 1, 2, \ldots, n-k+1. \tag{35.1}$$

Gautschi gives a nice and simple solution to this problem.

Denote by ℓ_μ the elementary Lagrange interpolation polynomials of degree $n-k$ for the points in $\mathbb{T}_n^{(k)}$, satisfying $\ell_\mu(\tau_\mu)=1$, $\ell_\mu(\tau_\nu)=0$ if $\nu\neq\mu$, and let $\mathbf{s}_\nu^{(k)}=[s_{\nu,1}^{(k)},s_{\nu,2}^{(k)},\ldots,s_{\nu,n-k+1}^{(k)}]$ be the vector with entries $s_{\nu,\mu}^{(k)}=1$ if the integral

$$I_{\nu,\mu}^{(k)}=\int_{-1}^{\tau_\nu}(\tau_\nu-\tau)^{k-1}\ell_\mu(\tau)\mathrm{d}\tau$$

is positive, $s_{\nu,\mu}^{(k)}=-1$ if it is negative, and an arbitrary value, for example $s_{\nu,\mu}^{(k)}=0$, if $I_{\nu,\mu}^{(k)}=0$. The desired quantity (35.1) is then expressed explicitly as

$$M_{n,\nu}^{(k)}=\frac{1}{(k-1)!}\sum_{\mu=1}^{n-k+1}\left|\int_{-1}^{\tau_\nu}(\tau_\nu-\tau)^{k-1}\ell_\mu(\tau)\,\mathrm{d}\tau\right|,\qquad \nu=1,2,\ldots,n-k+1,$$

the associated extremal polynomial being

$$p_{\nu,k}^*(t)=\frac{1}{(k-1)!}\int_{-1}^{t}(t-\tau)^{k-1}p_{n-k}(\tau;\mathbf{s}_\nu^{(k)})\mathrm{d}\tau,$$

where $p_{n-k}(\,\cdot\,;\mathbf{s}_\nu^{(k)})$ is the interpolation polynomial of degree $n-k$ passing through the points $(\tau_\mu,s_{\nu,\mu}^{(k)})$, $\mu=1,2,\ldots,n-k+1$.

Various examples are discussed numerically, with the points τ_ν being Gauss–Jacobi, in particular Gauss–Legendre, quadrature points, Gauss–Lobatto points for Jacobi weight functions, as well as equally spaced points on the open or closed interval from -1 to 1. Issues of interest are structural properties of the sign pattern matrix

$$\mathbf{S}_{n-k+1}=\begin{bmatrix}\mathbf{s}_1^{(k)}\\ \mathbf{s}_2^{(k)}\\ \vdots\\ \mathbf{s}_{n-k+1}^{(k)}\end{bmatrix}$$

and monotonicity properties of $M_{n,\nu}^{(k)}$ as a function of ν and of $\max_\nu M_{n,\nu}^{(k)}$ as a function of k or n. There are many opportunities here for additional work, for example, deriving the numerical results obtained, and more similar worthwhile results, by means of rigorous mathematical analysis.

Acknowledgement. I would like to express my thanks to the editors for their help in writing this commentary.

37

Orthogonal Polynomials on the Real Line

Walter Van Assche

During the past few decades, Walter Gautschi has developed a number of algorithms for computing recurrence coefficients for orthogonal polynomials and the related Gauss quadrature nodes and quadrature weights. These are all available and described in detail in his books and survey papers [GAB3], [GA**179**], [GAB4], [GAB5], [GAB6]. To fix notation, we will denote the orthonormal polynomials by p_n and the monic orthogonal polynomials by π_n. The recurrence relation for the monic polynomials is (§1.3 in [GAB3])

$$\pi_{n+1}(x) = (x - \alpha_n)\pi_n(x) - \beta_n \pi_{n-1}(x), \quad n = 0, 1, 2, \dots ,$$

with initial values $\pi_0(x) = 1$ and $\pi_{-1}(x) = 0$. Observe that β_0 is not needed, but Gautschi usually takes $\beta_0 = \mu_0$, the zero-moment of the orthogonality measure. The N-point quadrature nodes are the zeros of the orthogonal polynomial of degree N and are also equal to the eigenvalues of the symmetric tridiagonal Jacobi matrix containing the first N recurrence coefficients α_n, $0 \leq n \leq N-1$, on the diagonal and the square root of the $N-1$ coefficients β_n, $1 \leq n \leq N-1$, on the two side diagonals. The quadrature weights are related to the first components of the corresponding normalized eigenvectors and contain β_0 as a factor (Golub and Welsch, [7]). Hence the crucial task is to compute the recurrence coefficients.

37.1 Polynomials orthogonal with respect to exponential integrals [GA**211**]

The goal in this paper is to compute the recurrence coefficients for orthogonal polynomials with the exponential integral E_ν as weight function,

$$\int_0^\infty p_n(x)p_m(x)E_\nu(x)\mathrm{d}x = \delta_{m,n},$$

where

$$E_\nu(x) = \int_1^\infty \mathrm{e}^{-xt}\,\frac{\mathrm{d}t}{t^\nu}\,, \quad \nu > 0.$$

Integrals with this weight function, when ν is an integer, are of interest in radiative transfer problems in astronomy and astrophysics, and hence the relevant Gauss quadrature methods could be useful in that field. For $\nu = 1$, an eight-digit table of the 20-point Gauss quadrature formula has been published by Gautschi as early as 1968; see §(v) in [GA32].

In the present paper, Gautschi makes use of variable-precision codes from Chapter 2 of his book [GAB3] to compute as many of the recurrence coefficients as desired, to arbitrary precision. This is done not only for polynomials orthogonal on $[0,\infty]$, but also for subrange polynomials orthogonal on a finite interval $[0,c]$, $c > 0$. The software used comes from the package `OPQ` containing Matlab double-precision programs, and from the symbolic Matlab package `SOPQ`, which are available on Gautschi's website `https://www.cs.purdue.edu/archives/2002/wxg/codes/`.

The moments of the exponential integral weight function are easily obtained,

$$\mu_n = \int_0^\infty x^n E_\nu(x)\mathrm{d}x = \frac{n!}{\nu+n}\,, \quad n = 0,1,2,\ldots\,,$$

and are used in the symbolic version of the Chebyshev algorithm to compute the first N recurrence coefficients α_n, β_n, $0 \le n \le N-1$. Owing to the severe ill-conditioning of the process, the working precision has to be considerably larger than the target precision. Thus, for example, when $\nu = 1$ and $N = 40$, to obtain answers to an accuracy of 32 digits, one needs a working precision of 60 digits. The answers, in fact, are shown on p. 219 of Gautschi's paper. Interestingly, the recurrence coefficients α_n, with the exception of the first few, are very close, more so the larger n, to $2n$, which formally are the recurrence coefficients for the generalized Laguerre polynomial with parameter -1. Similarly for the recurrence coefficients β_n, which are close to $n(n-3)$, the generalized Laguerre coefficients with parameter -3; see Fig. 37.1. The reason for this may have something to do with the fact that $E_\nu(x) \approx \mathrm{e}^{-x}/x$ as $x \to \infty$, so that the exponential integral looks like the gamma density.

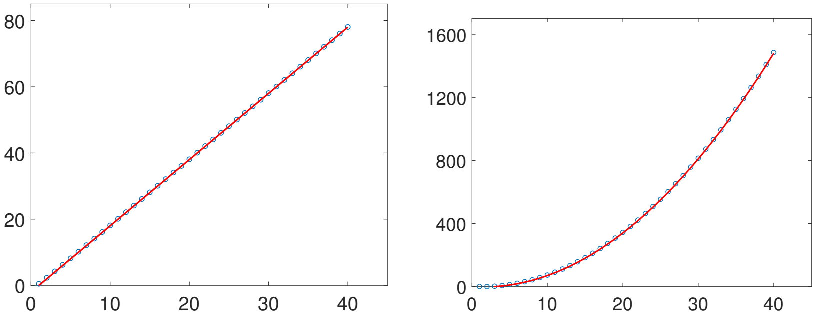


Figure 37.1: The recurrence coefficients α_n (left) and β_n (right) (indicated by circles) for $\nu = 1$. The lines in red indicate the recurrence coefficients of the generalized Laguerre polynomials with parameter -1 (left) and parameter -3 (right)

For the orthogonal polynomials on $[0, c]$,

$$\int_0^c p_n(x)p_m(x)E_\nu(x)\mathrm{d}x = \delta_{m,n},$$

the moments are

$$\mu_k = \int_0^c x^k E_\nu(x)\mathrm{d}x = \frac{1}{\nu+k}\left(\gamma(k+1,c) + \mathrm{e}^{\nu+k}\,\Gamma(1-\nu,c)\right),$$

where $\gamma(a,x)$ and $\Gamma(a,x)$ are the incomplete gamma functions,

$$\gamma(a,x) = \int_0^x t^{a-1}\mathrm{e}^{-t}\mathrm{d}t, \quad \Gamma(a,x) = \int_x^\infty t^{a-1}\mathrm{e}^{-t}\mathrm{d}t.$$

The latter can be computed with the Matlab Symbolic Toolbox, but for the former, Gautschi provides his own symbolic routine. The numerical conditioning is now worse, requiring a working precision of 85 digits to obtain the first 40 recurrence coefficients to 32 correct digits for moderately large values of c ($0 < c \leq 16$).

In Eq. (2.58) of [GAB4], Gautschi manages to find an explicit formula for the modified moments m_k of the weight function $E_\nu(x)$ on $[0, \infty]$ relative to an arbitrary (monic) polynomial $p_k(x)$ of degree k. He takes for p_k the monic Laguerre polynomial of degree k. Based on estimates for the numerical conditioning of the modified Chebyshev algorithm (producing the recurrence coefficients α_n, β_n from the modified moments), he then claims surprisingly large condition numbers, even larger than in the case of ordinary moments; see Tables 2.11 and 2.13 in [GAB4]. As communicated to me by Gautschi [4], he has verified these unusual claims by additional computations. He also pointed out two software issues, the first in the routine `run_sr_Enu.m`, where in the assignment statements for `ab1` and `ab2` the second argument on the right was missing and should be 32 in either case. Secondly, the routine `smmomEnu.m` was erroneous and has been replaced by a correct one. All this has been implemented in the routines on his website (see above).

In a later paper [GA**225**] (see §37.6) the computation of the three-term recurrence coefficients from modified moments is revisited. The comparison between the method using ordinary moments and modified moments was not fair since the analysis included the computation of the modified moments. See §37.6 for more comments.

37.2 Polynomials orthogonal with respect to cardinal B-spline weight functions [GA**215**]

B-splines play an important role in approximation by splines and multiresolution approximation, where integrals involving the cardinal B-spline appear frequently, and therefore quadrature formulas for these weight functions are indispensable. Milovanović [12] developed a method and an algorithm, based on the moments of the cardinal B-spline, to compute in symbolic form the recurrence coefficients of the orthogonal polynomials on $[0, m]$ for the cardinal B-spline function of order m. Gautschi had access to a preprint of Milovanović's paper and decided to look at another approach based on multicomponent discretization, which is described in §2.2.4 of his book [GAB3].

The cardinal B-spline φ_m of order $m \geq 1$ is a positive piecewise polynomial function which consists of polynomials of degree $m-1$ on each interval $[k-1, k]$ $(1 \leq k \leq m)$ chosen so as to make φ_m smooth, belonging to $C^{m-2}(\mathbb{R})$. It vanishes outside of $[0, m]$; see Fig. 37.2. For $m = 1$ the cardinal B-spline is constant equal to 1 on $[0, 1]$ and 0 elsewhere. The corresponding orthogonal polynomials are then the shifted Legendre polynomials on $[0, 1]$.

The cardinal B-spline φ_m can be (and has been) computed recursively by

$$\varphi_m(x) = \frac{1}{m-1}\left(x\,\varphi_{m-1}(x) + (m-x)\varphi_{m-1}(x-1)\right), \quad m \geq 2.$$

Matlab routines `cBspline.m` and `scBspline.m` are provided to compute φ_m in Matlab double- resp. variable-precision arithmetic. The multicomponent

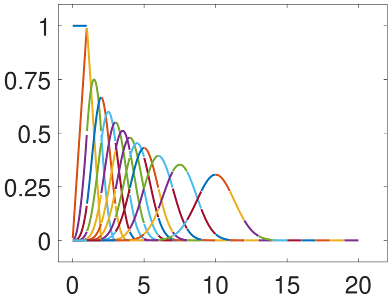


Figure 37.2: Cardinal B-splines for $m = 1:10, 12, 15, 20$

discretization procedure requires the evaluation of integrals of the form

$$\int_{k-1}^{k} \pi_\ell^2(t)\varphi_m(t)\mathrm{d}t, \qquad \int_{k-1}^{k} t\,\pi_\ell^2(t)\varphi_m(t)\mathrm{d}t$$

with $\ell \le n-1$, where n is the number of recurrence coefficient pairs wanted. These integrals can be computed exactly by Gauss quadrature on $[k-1, k]$ with $n + \lceil m/2 \rceil - 1$ points. The results produced in double precision are compared with the high-precision results of the moment-based method and are found to be quite accurate, the relative error always being close to machine precision. Three numerical examples illustrate the efficiency of the method.

The cardinal B-spline of order m is symmetric on the interval $[0, m]$ so that the recurrence coefficients α_k are all constant equal to $m/2$. The recurrence coefficients β_k converge to $m^2/16$, which is to be expected since the weight function belongs to Nevai's class for the interval $[0, m]$. This follows from the fact that $\varphi_m(x) > 0$ on $(0, m)$ (Rakhmanov's theorem; see, e.g., [2]). Gautschi mentions that convergence of the recurrence coefficients is a consequence of the weight function being in the Szegö class, which is correct, but the weight function also belongs to the Nevai class for the interval $[0, m]$, which is a much larger class. Gautschi also observes that convergence is slow, which is probably due to the fact that $\varphi_m(x)$ is very small near the endpoints 0 and m; see Fig. 37.2.

37.3 Binet-type polynomials and their zeros [GA**219**]

In the late 1990s, Dahlquist published some important work on summation formulas which involved Gauss quadrature using the Binet distribution $w^B(x) = w(2\pi x)/2\pi$, where

$$w(x) = -\log(1 - \mathrm{e}^{-|x|}), \quad x \in (\infty, \infty),$$

is the Binet weight function. In this paper [GA**219**], Gautschi and Milovanović also consider the generalized Binet weight function

$$w(x;\alpha) = -\log(1-\alpha \mathrm{e}^{-|x|}), \quad x \in (-\infty,\infty), \quad 0<\alpha<1.$$

They compute the recurrence coefficients and the zeros of orthogonal polynomials related to these and some other weight functions, such as the squares of these weight functions and the restrictions to $[0,\infty)$, $[c,\infty)$, $[0,c]$ and $[-c,c]$, where $c>0$. The Binet function w on $(-\infty,\infty)$ is considered in §2.1 of the paper, where the moments

$$\mu_k = 0 \ \text{ if } k \text{ is odd}, \quad \mu_k = 2\,k!\,\zeta(k+2) \ \text{ if } k \text{ is even}$$

are used in the Chebyshev algorithm to generate in sufficiently high precision the desired recurrence coefficients. As many as 64 working digits are used to compute the first 100 recurrence coefficients to 32 digits. The generalized Binet function $w(x;\alpha)$ is dealt with in §2.2. It is shown, in particular, that all positive zeros of the respective orthogonal polynomials are monotonically decreasing as functions of α. The proof uses Markov's theorem on the variation of the zeros of an orthogonal polynomial in dependence of a parameter contained in the weight function. In §3.1, the squared Binet function is considered, and in §3.2 the squared generalized Binet weight function, for which the same monotonicity property of the zeros is shown to hold as for generalized Binet functions in §2.2. Halfrange weight functions, supported on the positive real line $[0,\infty)$, are dealt with in §4, where all zeros of the corresponding generalized Binet polynomials are shown to decrease monotonically as functions of α. The squares of the Binet and generalized Binet function on $[0,\infty)$ are investigated in §5. The above monotonicity result for zeros again holds for halfrange squared generalized Binet polynomials. Finally, §6 deals with the restrictions to $[c,\infty)$, $c>0$, to $[0,c]$, and to $[-c,c]$. The dependence of the zeros on the parameter c is discussed for each of these cases. It is observed that the restriction to $[c,\infty)$, yielding upper subrange Binet polynomials, corresponds to a shift in the weight function in the sense that $w(x+c;\alpha) = w(x;\mathrm{e}^{-c}\alpha)$ for $x \geq 0$, so that the recurrrence coefficients α_k are shifted by the constant c. All zeros of the upper subrange Binet polynomials are now shown to be monotonically increasing as functions of c.

Cizek and Vrsay [1] formulated a conjecture about the asymptotic behavior of the recurrence coefficients, which was proved by Jones and Van Assche [9]. They, however, considered the weight function $w^B(x) = w(2\,\pi\,x)/2\pi$ and

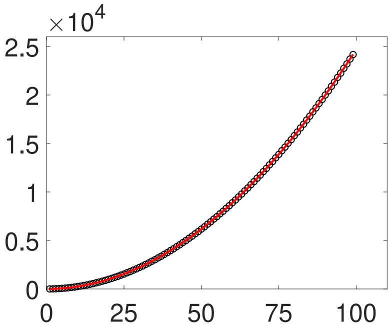


Figure 37.3: The recurrence coefficients β_k (indicated by black circles) for the Binet function w and their approximation (indicated by a red line)

its transformed version $w^B(\sqrt{x})/\sqrt{x}$ on $[0, \infty)$. Translated to the recurrence coefficients β_k for the Binet weight w, the asymptotic behavior is

$$\lim_{n\to\infty} \frac{\beta_n}{n^2} = \frac{\pi^2}{4}\,.$$

The recurrence coefficients β_k (black circles) are plotted in Fig. 37.3 together with $n^2\pi^2/4$ (solid red line), showing that this asymptotic behavior is indeed confirmed, not only for large n, but for all n. The recurrence coefficients are taken from the file `coeff_binet.txt` on Gautschi's website `https://www.cs.purdue.edu/archives/2002/wxg/codes/BINET.html`.

37.4 Polynomials orthogonal relative to weight functions of Prudnikov type [GA**224**]

At the seventh Spanish Symposium on Orthogonal Polynomials and Applications (VII SPOA) held at the University of Granada (Spain) in 1991, Prudnikov posed some problems [13, §9] about orthogonal polynomials with ultra-exponential weight functions $\xi(x, k)$ given by

$$\xi(x,k) = \frac{1}{2\pi \mathrm{i}} \int_{a-\mathrm{i}\infty}^{a+\mathrm{i}\infty} \frac{\Gamma^k(s)}{x^s}\,\mathrm{d}s, \quad a > 0,\ x > 0.$$

The right-hand side is the inverse of the Mellin transform, so that

$$\int_0^\infty x^{s-1}\xi(x,k)\mathrm{d}x = \Gamma^k(s),$$

and, in particular, the moments of $\xi(x,k)$ are

$$\mu_n = (n!)^k.$$

For $k = 1$ one has $\xi(x,1) = \mathrm{e}^{-x}$ and the corresponding orthogonal polynomials are the Laguerre polynomials $L_n(x)$. For $k = 2$ one has $\xi(x,2) = 2K_0(2\sqrt{x})$, where K_0 is the second-kind modified Bessel function of order 0. Prudnikov was interested in getting the generating function, a Rodrigues-type formula, the recurrence relation, and a differential equation for the corresponding orthogonal polynomials.

Gautschi and Milovanović consider the weight function

$$\rho_\nu(x) = 2\,x^{\nu/2}K_\nu(2\sqrt{x}), \quad x > 0,\ \nu \geq 0,$$

the case $\nu = 0$ corresponding to Prudnikov's weight function $\xi(x,2)$. The generalized Prudnikov weight is

$$w_\nu^\alpha(x) = x^\alpha \rho_\nu(x), \quad x > 0,\ \alpha > -1,$$

and its moments are

$$\mu_n = \Gamma(n+\alpha+\nu+1)\Gamma(n+\alpha+1).$$

Gautschi and Milovanović work out a procedure to compute the first N recurrence coefficients by means of the Chebyshev algorithm with sufficiently high precision. Their Matlab routines can be found on Gautschi's website `https://www.cs.purdue.edu/archives/2002/wxg/codes/PRUD.html`. Their analysis of the conditioning of the Chebyshev algorithm suggests that to compute $N = 100$ recurrence coefficients to an accuracy of 15 decimal digits requires 80 working digits.

Other related weight functions are also considered: Prudnikov-type weights of type 1,

$$w_\nu^+(x) = \mathrm{e}^{-x}\rho_\nu(x), \quad x > 0,\ \nu > -1,$$

and Prudnikov-type weights of type 2,

$$w_\nu^-(x) = x^{-1}\mathrm{e}^{-1/x}\rho_\nu(x), \quad x > 0,\ \nu \in \mathbb{R}.$$

The moments of Prudnikov-type weights of type 1 can be obtained in terms of a confluent hypergeometric function, and those for Prudnikov-type weights of type 2 in terms of a Meijer G-function The condition numbers for type 1 are substantially larger than for the generalized Prudnikov weight, and as a result, one will need to work with 100 digits to obtain the first $N = 100$ recurrence coefficients to an accuracy of 15 decimal digits. The condition numbers for the Prudnikov-type weights of type 2 are similar to those of the generalized Prudnikov weights. Gautschi and Milovanović also consider symmetric versions of the Prudnikov weights on $(-\infty, \infty)$ but with x replaced by $|x|$. They find that the condition numbers are then much lower and the number of required digits is about half of what was needed before.

It turns out that when $k = 2$, the *multiple* orthogonal polynomials [8], of interest in Hermite–Padé approximation and random matrix theory [15], with weights $x^\alpha \rho_\nu(x)$ and $x^\alpha \rho_{\nu+1}(x)$ are much more natural [14], and for them, one can indeed find a Rodrigues formula, a differential equation, and a recurrence relation, but not of order 2, but rather of order 3. When $k > 2$, the multiple orthogonal polynomials can be expressed in terms of Meijer G-functions and generalized hypergeometric functions. They appear in random matrix theory when one takes products of random matrices (Kuijlaars and Zhang [10]).

37.5 Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure [GA**223**]

The Marchenko–Pastur density is

$$w(x) = \frac{1}{2\pi c}\, x^{-1} \sqrt{(b-x)(x-a)}, \quad x \in [a,b],$$

when $0 < c < 1$, where $a = (\sqrt{c} - 1)^2$ and $b = (\sqrt{c} + 1)^2$, and

$$w(x) + \left(1 - \frac{1}{c}\right) \delta_0$$

when $c > 1$, where δ_0 is the Dirac delta function at $x = 0$ with mass 1. This in fact is the limiting distribution of the eigenvalues of the matrix XX^*, where X is an $n \times m$ random matrix with i.i.d. normal random variables (real, complex, or quaternion). When $m < n$, there will be an eigenvalue at 0 with multiplicity $n - m$, which gives rise to the δ_0 when $c > 1$ (given that $m = n/c$). The Marchenko–Pastur density also is the asymptotic distribution of the zeros of Laguerre polynomials $L_n^{a_n}(nx)$ when a_n/n has a limit, see [5] when $a_n = \alpha$ and [6] when $a_n = a\,n + \alpha$. This is not surprising since there is a connection between the random matrix XX^* and Laguerre polynomials, the average characteristic polynomial of XX^* being the Laguerre polynomial $L_n^{(m-n)}(x)$ when X involves complex normal random variables and $m \geq n$, and $x^{n-m}L_m^{(n-m)}(x)$ when $m < n$.

Gautschi and Milovanović generalize the Marchenko–Pastur density by replacing the exponents 1/2 for $b - x$ and $x - a$ by α and β, respectively, so that

$$w(x; \alpha, \beta, c) = \begin{cases} \dfrac{1}{c\,\mu_0}\, x^{-1}\,(b-x)^\alpha (x-a)^\beta + \left(1 - \dfrac{1}{c}\right)\delta_0 \ \text{ if } \ c > 1, \\ \dfrac{1}{\mu_0}\, x^{-1}\,(b-x)^\alpha (x-a)^\beta \ \text{ if } \ c < 1, \end{cases}$$

where

$$\mu_0(\alpha, \beta, c) = \int_a^b x^{-1}(b-x)^\alpha (x-a)^\beta \mathrm{d}x, \quad c > 0,$$

and the absolutely continuous part is supported on $[a, b]$. With an affine transformation $x = 2\sqrt{c}\,t + c + 1$ the interval $[a, b]$ is mapped to $[-1, 1]$ and the weight transforms to a weight of the form

$$\chi\, \frac{w^{(\alpha,\beta)}(t)}{t+g} + \left(1 - \frac{1}{c}\right)\delta_{-g},$$

where $w^{(\alpha,\beta)}$ is the Jacobi density $(1-t)^\alpha(1+t)^\beta$, $g = (c+1)/(2\sqrt{c})$ and χ a constant that makes this a probability density. We are dealing here with a modification of the Jacobi density known as a Geronimus transform; see, e.g., Zhedanov [16, Eq. (3.12)] or Maroni [11]. The Geronimus transform is the inverse of the Christoffel transform, which multiplies a given weight function by $t+g$. The Geronimus transform can also be viewed as an Uvarov transform (one divides the weight by $t+g$) and the addition of a mass point at $-g$. Such spectral transformations have been considered before in the literature, in particular by Gautschi in §2.4.1 of [GAB3]. But here, Gautschi and Milovanović do this in much detail for the generalized Marchenko–Pastur density, both analytically and numerically.

They first show that the monic orthogonal polynomials π_k^* for the weight on $[-1, 1]$ are given by

$$\pi_k^*(t) = \hat{P}_k^{(\alpha,\beta)}(t) + \gamma_k \hat{P}_{k-1}^{(\alpha,\beta)}(t),$$

where $\hat{P}_k^{(\alpha,\beta)}$ are the monic Jacobi polynomials and γ_k constants depending on α, β, c. This corresponds to Zhedanov's equation (3.9) in [16]. The coefficients γ_k are investigated in detail. It is shown that they are given by $\gamma_k = -p_k/p_{k-1}$, where the p_k satisfy the recurrence relation

$$p_{k+1} + (g + \alpha_k^J)p_k + \beta_k^J p_{k-1} = 0, \quad k \geq 1.$$

This is the three-term recurrence relation for monic Jacobi polynomials and variable $t = -g$. The general solution is a linear combination of $\hat{P}_k^{(\alpha,\beta)}(-g)$ and

$$\int_{-1}^{1} \frac{\hat{P}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t.$$

The latter is a minimal solution of the recurrence relation, for which Gautschi had developed numerical methods earlier; see §2.3 in [GAB3]. The recurrence coefficients α_k and β_k of interest can now be obtained in terms of the recurrence coefficients α_k^J and β_k^J of the monic Jacobi polynomials and the coefficients γ_k. Gautschi and Milovanović show that

$$\lim_{k\to\infty} \alpha_k = 1 + c, \quad \lim_{k\to\infty} \beta_k = c,$$

which also follows from the fact that the Marchenko–Pastur measure on $[a, b]$ belongs to the Nevai class [2].

Some examples are worked out in more detail, in particular those with $\alpha, \beta = \pm 1/2$, for which the recurrence coefficients α_k and β_k are given explicitly as functions of c. Remarkably, they converge almost instantaneously when $c < 1$, and in the case $\alpha = \beta = 1/2$ also when $c > 1$.

37.6 Another look at polynomials orthogonal relative to exponential integral weight functions [GA**225**]

In [GA**211**] Gautschi computed the recurrence coefficients for the orthogonal polynomials relative to the exponential integral weight function E_ν, see §37.1. He now realized that the comparison between the Chebyshev algorithm with ordinary and modified moments was misleading, since he included the computation of the modified moments in his analysis. For a fair comparison one must assume that the ordinary moments and the modified moments are given and the question then is how well both methods perform when computing the recurrence coefficients.

The conclusion is that for the weight function E_ν on the infinite interval $[0, \infty)$ the Chebyshev algorithm with modified moments (involving Laguerre polynomials with parameter $\alpha = -1/4$) is only marginally better than the method using ordinary moments. Both methods show a considerable amount of ill-conditioning, so that variable precision is necessary and Matlab double precision is not sufficient. For the weight function E_ν restricted to the finite interval $[0, c]$ the Chebyshev algorithm with modified moments (involving the Legendre polynomials on $[0, c]$) is perfectly stable and produces almost fully accurate results in Matlab double precision. There is however a difference in the runtime of the two algorithms in variable-precision arithmetic: the method with ordinary moments is about 30% faster than the method using modified moments.

The modified moments (in the case of the infinite support interval $[0, \infty)$) are evaluated here differently than before in [3, Exercise 2.26(b)] and in a manner applicable also to finite support intervals $[0, c]$. Specifically,

$$m_k = \int_0^\infty p_k(t) E_\nu(t) \mathrm{d}t = \int_0^\infty p_k(t)\, t^{\nu-1} \Gamma(1-\nu, t) \mathrm{d}t,$$

where $\Gamma(a, x)$ is the incomplete gamma function. Gautschi suggests to compute the latter integral using n-point Gauss quadrature relative to the novel and unusual weight function

$$v(x) = x^{\nu-1} \Gamma(1-\nu, x), \quad x > 0,$$

which will give exact results when $0 \leq k \leq 2n-1$. In a similar way one can compute the modified moments for the weight E_ν restricted to the finite interval $[0, c]$ by using Gauss quadrature relative to the weight function v restricted to $[0, c]$. Of course, this means that one has to compute the nodes and weights for these Gauss quadrature rules, for which the Chebyshev algorithm (in sufficiently high precision) is suggested. The moments for the weight v can be computed explicitly in terms of the gamma function (for the infinite interval) and the incomplete gamma function (for the finite interval).

References

[1] Cizek,J. and E. R. Vrscay. Asymptotic estimation of the coefficients of the continued fraction representing the Binet function, *C. R. Math. Rep. Acad. Sci. Canada* **4** (1982), 201–206.

[2] Denisov, Sergey A. On Rakhmanov's theorem for Jacobi matrices, *Proc. Amer. Math. Soc.* **132** (2004), 847–852.

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

[4] Gautschi, Walter. Personal communication by email of August 18, 2021.

[5] Gawronski, Wolfgang. On the asymptotic distribution of the zeros of Hermite, Laguerre, and Jonquière polynomials, *J. Approx. Theory* **50** (1987) 214–231.

[6] Gawronski, Wolfgang. Strong asymptotics and the asymptotic zero distribution of Laguerre polynomials $L_n^{(an+\alpha)}$ and Hermite polynomials $H_n^{(an+\alpha)}$, *Analysis* **13** (1993), 29–67.

[7] Golub, Gene H. and John H. Welsch. Calculation of Gauss quadrature rules. *Math. Comp.* **23** (1969), 221–230.

[8] Ismail, Mourad E. H. *Classical and quantum orthogonal polynomials in one variable*, Encyclopedia of Mathematics and its Applications 98, Cambridge University Press, Cambridge, 2005.

[9] Jones, William B. and Walter Van Assche. Asymptotic behavior of the continued fraction coefficients of a class of Stieltjes transforms including the Binet function, Orthogonal functions, moment theory, and continued fractions (Campinas, 1996), *Lecture Notes in Pure and Appl. Math.* **199**, Dekker, New York, 1998, 257–274.

[10] Kuijlaars, Arno B. J. and Lun Zhang. Singular values of products of Ginibre random matrices, multiple orthogonal polynomials and hard edge scaling limits, *Comm. Math. Phys.* **332** (2014), 759–781.

[11] Maroni, P. Sur la suite de polynômes orthogonaux associée à la forme $u = \delta_c + \lambda(x-c)^{-1}L$, *Period. Math. Hung.* **21** (1990),223–248.

[12] Milovanović, Gradimir V. Symbolic-numeric computation of orthogonal polynomials and Gaussian quadratures with respect to the cardinal B-spline, *Numer. Algorithms* **76** (2017), 333–347.

[13] Van Assche, Walter. Open problems, *J. Comput. Appl. Math.* **48** (1993), 225–243.

[14] Van Assche, W. and S. B. Yakubovich. Multiple orthogonal polynomials associated with Macdonald functions, *Integral Transform. Spec. Funct.* **9** (2000), 229–244.

[15] Zhang, Lun. A note on the limiting mean distribution of singular values for products of two Wishart random matrices, *J. Math. Phys.* **54** (8) (2013), 083303, 8 pp.

[16] Zhedanov, Alexei. Rational spectral transformations and orthogonal polynomials, *J. Comput. Appl. Math.* **85** (1997), 67–86.

38

Gauss-type Quadrature

Gradimir V. Milovanović

This commentary concerns the paper [GA**209**] (see p. 2 of this volume).

38.1 Introduction

In [3], I presented Gautschi's constructive theory of orthogonal polynomials on the real line, including effective algorithms for numerically generating orthogonal polynomials, a detailed stability analysis of these algorithms, as well as several new applications of orthogonal polynomials. In the last section of that article, I mentioned some extensions and applications of Gautschi's theory including a few of my own.

One of these applications is the construction of so-called s-orthogonal polynomials relative to any positive weight function (or measure $\mathrm{d}\lambda$) and the associated quadrature formulas with multiple nodes. This indeed is the topic of [GA**209**], where the measure $\mathrm{d}\lambda$ is either the Laguerre measure $e^{-t}\mathrm{d}t$ on $[0,\infty]$ or the Hermite measure $e^{-t^2}\mathrm{d}t$ on $[-\infty,\infty]$.

38.2 Gauss–Turán quadrature

Let $\mathbb{P}_n$ be the set of all algebraic polynomials of degree at most n and $\mathbb{P}$ be the set of all algebraic polynomials. The (monic) s-orthogonal polynomials

$$\pi_{n,s}(t) = t^n + \text{terms of lower degree} \quad (n \in \mathbb{N}_0,\ s \text{ a fixed integer in } \mathbb{N})$$

are characterized by the orthogonality relations

$$\int_{\mathbb{R}} t^\nu \, [\pi_{n,s}(t)]^{2s+1} \mathrm{d}\lambda(t) = 0, \quad \nu = 0, 1, \dots, n-1, \tag{38.1}$$

and play an important role in the construction of so-called Turán quadratures [4] with multiple nodes

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n}\sum_{\sigma=0}^{2s} \lambda_\nu^{(\sigma)} f^{(\sigma)}(\tau_\nu) + R_{n,s}(f), \tag{38.2}$$

where n and s are positive integers $n \geq 1$, $s > 0$, and the formula (38.2) has maximum degree of exactness $d = 2(s+1)n-1$, i.e., $R_{n,s}(f) = 0$ for all $f \in \mathbb{P}_{2(s+1)n-1}$. Indeed, the nodes τ_ν are the zeros of $\pi_{n,s}$ and the weights $\lambda_\nu^{(\sigma)}$, for each ν, the solution of an upper triangular system of $2s+1$ linear equations.

If $s = 0$, the polynomials $\pi_{n,s}$ are the ordinary (monic) polynomials $\pi_{n,s} = \pi_n$ orthogonal with respect to the measure $\mathrm{d}\lambda$ on $\mathbb{R}$, and (38.2) is the usual Gaussian quadrature rule.

In [1] we reinterpreted the conditions (38.1) as ordinary orthogonality conditions relative to the positive measure $\mathrm{d}\lambda_{n,s}(t) = [\pi_{n,s}(t)]^{2s}\mathrm{d}\lambda(t)$ (see also [GA**154**] and [2]), i.e.,

$$\int_{\mathbb{R}} t^\nu\, \pi_{n,s}(t)\mathrm{d}\lambda_{n,s}(t) = 0, \quad \nu = 0,1,\ldots,n-1. \tag{38.3}$$

Evidently, this defines $\pi_{n,s}$ implicitly. Nevertheless, for the measure $\mathrm{d}\lambda_{n,s}(t)$, where n and s are fixed, there exists a unique sequence of (monic) polynomials $\{\pi_{k,s}^{(n)}\}_{k\geq 0}$ ($\deg \pi_{k,s}^{(n)} = k$) such that

$$\int_{\mathbb{R}} t^j\, \pi_{k,s}^{(n)}(t)\mathrm{d}\lambda_{n,s}(t) = 0, \quad j = 0,1,\ldots,k-1. \tag{38.4}$$

Thus, $\{\pi_{k,s}^{(n)}\}_{k\geq 0}$ is a sequence of standard monic polynomials orthogonal with respect to the measure $\mathrm{d}\lambda_{n,s}(t)$ on $\mathbb{R}$ and therefore satisfies a three-term recurrence relation of the form

$$\begin{aligned} &\pi_{k+1,s}^{(n)}(t) = (t-\alpha_k)\pi_{k,s}^{(n)}(t) - \beta_k\pi_{k-1,s}^{(n)}(t), \quad k = 0,1,\ldots, \\ &\pi_{-1,s}^{(n)}(t) = 0, \quad \pi_{0,s}^{(n)}(t) = 1, \end{aligned} \tag{38.5}$$

where the recurrence coefficients depend on the measure $\mathrm{d}\lambda_{n,s}$. We denote them by $\alpha_{k,s}^{(n)} = \alpha_k^{(n)}$, $k = 0,1,2,\ldots$, and $\beta_{k,s}^{(n)} = \beta_k^{(n)}$, $k = 1,2,\ldots$, suppressing the dependence on s, or even simply by α_k and β_k as in (38.5).

Because of the uniqueness of this sequence of orthogonal polynomials $\{\pi_{k,s}^{(n)}\}_{k=0}^{\infty}$, by comparing (38.3) with (38.4), we conclude that its member with $k = n$ must be the s-orthogonal polynomial, i.e.,

$$\pi_{n,s}(t) = \pi_{n,s}^{(n)}(t), \tag{38.6}$$

so that we need the recurrence coefficients only for $k \leq n-1$ in order to get (38.6).

Now using Darboux's formulas

$$\alpha_k = \frac{\left(t\pi_{k,s}^{(n)}, \pi_{k,s}^{(n)}\right)}{\left(\pi_{k,s}^{(n)}, \pi_{k,s}^{(n)}\right)}, \quad \beta_k = \frac{\left(\pi_{k,s}^{(n)}, \pi_{k,s}^{(n)}\right)}{\left(\pi_{k-1,s}^{(n)}, \pi_{k-1,s}^{(n)}\right)}, \tag{38.7}$$

for $k \leq n-1$ (see [1]), where

$$(p, q) = (p, q)_{\mathrm{d}\lambda_{n,s}} = \int_{\mathbb{R}} p(t)q(t)\mathrm{d}\lambda_{n,s}(t), \tag{38.8}$$

and omitting indices ($\pi_k = \pi_{k,s}^{(n)}$), we can write the following system of $2n-1$ nonlinear equations,

$$\boldsymbol{f}_n(\boldsymbol{\varrho}_n) = \mathbf{0}, \quad \boldsymbol{\varrho}_n = [\alpha_0, \alpha_1, \ldots, \alpha_{n-1}; \beta_1, \ldots, \beta_{n-1}]^T, \tag{38.9}$$

where $\boldsymbol{f}_n = [f_1, f_2, \ldots, f_{2n-1}]^T$ and

$$\left.\begin{aligned} f_{2k+1} &= \int_{\mathbb{R}} (\alpha_k - t)\pi_k^2(t)\pi_n^{2s}(t)\mathrm{d}\lambda(t) = 0, \quad k = 0, 1, \ldots, n-1, \\ f_{2k} &= \int_{\mathbb{R}} \left[\beta_k \pi_{k-1}^2(t) - \pi_k^2(t)\right] \pi_n^{2s}(t)\mathrm{d}\lambda(t) = 0, \quad k = 1, \ldots, n-1. \end{aligned}\right\} \tag{38.10}$$

For solving the system of equations (38.9), the Newton–Kantorovich method, converging quadratically, was proposed in [1]. All integrals in (38.10), as well as the ones in the elements of the Jacobian, can be computed exactly, except for rounding errors, by using an $(s+1)n$-point Gauss–Christoffel quadrature formula with respect to the measure $\mathrm{d}\lambda(t)$. Thus, in all calculations, only the fundamental three-term recurrence relation (38.5) and the Gauss–Christoffel quadrature rule with respect to the measure $\mathrm{d}\lambda(t)$ are needed (see [1] or §3.1.3.2 in [GAB3]).

Using a stable construction of s-orthogonal polynomials, in a joint paper with Gautschi [GA**154**] we developed a method for constructing Gauss–Turán type quadrature formulae (38.2). This led to further progress in the theory of quadratures with multiple nodes.

As can be seen, the functions f_k, $k = 1, 2, \ldots, 2n-1$, in (38.10) are highly nonlinear functions of the $2n-1$ variables $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ and $\beta_1, \ldots, \beta_{n-1}$. Solving (38.9) with the Newton–Kantorovich method therefore requires sufficiently accurate initial approximations for the recurrence coefficients. This is the main problem in this area, which, in [GA**209**], is addressed by Gautschi in the case of Laguerre and Hermite weight functions.

38.3 Gauss–Turán quadrature for Laguerre and Hermite weight functions $s = 1, 2, \ldots, 5$

The values of s are restricted here to the first five integers, which is deemed sufficient for most applications. In the notations below, the parameter s is suppressed.

Consider first the Laguerre case. Here the recurrence coefficients α_k^L and β_k^L for the ordinary (monic) Laguerre polynomials are linear resp. quadratic in k, more precisely, $\alpha_k^L = 2k+1$ and $\beta_k^L = k^2$. Gautschi discovered that a similar property holds, at least approximately and irrespective of the value of s, also for the recurrence coefficients of interest here.

When $n = 1$ there is only one unknown $\alpha_0 = \alpha_0^{(1)}$, which is the only positive solution of the algebraic equation

$$f_1 = \int_0^\infty (\alpha_0 - t)[\pi_{1,s}(t)]^{2s} \mathrm{e}^{-t} \mathrm{d}t = \sum_{\nu=0}^{2s+1} (-1)^\nu (2s+2-\nu)_\nu \, \alpha_0^{2s+1-\nu} = 0,$$

where we used $\pi_{1,s}(t) = t - \alpha_0$.

When $n = 2$ there are three unknowns $\alpha_0^{(2)}$, $\alpha_1^{(2)}$ and $\beta_1^{(2)}$, when $n = 3$ five unknowns $\alpha_0^{(3)}$, $\alpha_1^{(3)}$, $\alpha_2^{(3)}$ and $\beta_1^{(3)}$, $\beta_2^{(3)}$, etc. Assuming $n = 1, 2, \ldots, N$, all unknowns may be arranged in a matrix $R = [A \,|\, B]$ of order $N \times (2N-1)$ formed with two block matrices A and B containing α- resp. β-coefficients.

That is,

$$R = \left[\begin{array}{cccccc|ccccc}
\alpha_0^{(1)} & & & & & & & & & & \\
\alpha_0^{(2)} & \alpha_1^{(2)} & & & & & \beta_1^{(2)} & & & & \\
\alpha_0^{(3)} & \alpha_1^{(3)} & \alpha_2^{(3)} & & & & \beta_1^{(3)} & \beta_2^{(3)} & & & \\
\vdots & \vdots & \vdots & \ddots & & & \vdots & \vdots & \ddots & & \\
\alpha_0^{(N-1)} & \alpha_1^{(N-1)} & \alpha_2^{(N-1)} & \cdots & \alpha_{N-2}^{(N-1)} & & \beta_1^{(N-1)} & \beta_2^{(N-1)} & \cdots & \beta_{N-2}^{(N-1)} & \\
\alpha_0^{(N)} & \alpha_1^{(N)} & \alpha_2^{(N)} & \cdots & \alpha_{N-2}^{(N)} & \alpha_{N-1}^{(N)} & \beta_1^{(N)} & \beta_2^{(N)} & \cdots & \beta_{N-2}^{(N)} & \beta_{N-1}^{(N)}
\end{array}\right].$$

Gautschi's discovery mentioned above can now be stated as follows: The elements on the diagonal and subdiagonals of the matrix A lie approximately on straight lines, whereas those on the diagonal and subdiagonals of the matrix B lie approximately on parabolas. Moreover, with a good deal of heuristics, Gautschi was able to determine the coefficients of the respective linear and quadratic functions. The approximate matrix $R_a \approx R$ with $N = 42$ so obtained provides sufficiently accurate initial approximations for the solution of the system (38.9) by the Newton–Kantorovich method (in sufficiently high-precision arithmetic) to succeed for all $2 \leq n \leq N$ when $s = 1$. For larger s this is true only if N is smaller, as small as $N = 15$ when $s = 5$.

In the case of the Hermite measure $\mathrm{d}\lambda(t) = e^{-t^2}\mathrm{d}t$, matters are similar, but simpler, insofar as the matrix A is the zero matrix by symmetry. Also, the integer N can now be taken as large as 90.

I take this opportunity to point out some inaccuracies and a mistake in the paper [GA**209**], neither of which affects the validity of the final results. On p. 62 of the paper, right after the "First Empirical Observation", it should have been stated that $k = 0$ in the left graph of Fig. 1 and $k = 1$ in the right graph. Likewise for Fig. 2 on p. 70. Also on top of p. 67, read $k = 0, 1, 2, \ldots, N-1$ instead of $k = 1, 2, \ldots, N-1$. Finally, the equation for $\Gamma(k + 1/2)$ in the textline after the third displayed formula on p. 69 should read $\Gamma(k + 1/2) = (1/2)_k \sqrt{\pi}$ and Eq. (4.1), accordingly,

$$Q_{2s+1}(\beta) = \sum_{k=0}^{2s+1} (-1)^k \binom{2s+1}{k} \left(\frac{1}{2}\right)_k \beta^{2s+1-k}.$$

The five equations following (4.1), however, are correct.

References

[1] Gradimir V. Milovanović, Construction of s-orthogonal polynomials and Turán quadrature formulae, in: *Numerical methods and approximation theory III (Niš, 1987)*, 311–328, Univ. Niš, Niš, 1988.

[2] Gradimir V. Milovanović, Quadratures with multiple nodes, power orthogonality, and moment-preserving spline approximation, *J. Comput. Appl. Math.* **127** (2001), 267–286.

[3] Gradimir V. Milovanović, Chapter 11: Orthogonal polynomials on the real line, in: *Walter Gautschi: Selected works with commentaries, Volume 2* (C. Brezinski and A. Sameh, eds.), 3–16, Birkhäuser, Basel, 2014.

[4] P. Turán, On the theory of the mechanical quadrature, *Acta Sci. Math. (Szeged)* **12** (1950), 30–37.

39

Ordinary Differential Equations

John Butcher

See Section 22, Volume 3.

40

History

Gerhard Wanner

40.1 A historical note on Gauss–Kronrod quadrature [GA**177**]
Many famous ideas attributed to famous authors have been found by someone else much earlier. As Walter Gautschi has discovered, the idea behind Gauss–Kronrod formulas is no exception from this rule and indeed has been put forward more than half a century earlier by Rudolf Skutsch [1]. Walter Gautschi does not speculate as to why this paper went unnoticed for such a long time. One reason, however, may well be that the paper was written in a style not all that easy to understand.

40.2 Interpolation before and after Lagrange [GA**206**]
Lagrange was born 1736 in Turin and died 1813 in Paris. For the occasion of Lagrange's 200th anniversary of death, the Department of Mathematics at the University of Turin invited Walter Gautschi to give a lecture in the series *Lezioni Lagrangiane*, which led to the present paper.

After a long career entirely consecrated to academic research of the highest level, without any teaching duties, Lagrange, then "le premier des savants d'Europe" (Fourier), found himself in front of a large heterogeneous class of "Citoyens" in the newly founded École Normale, who expected "des notions élémentaires", but instead "ils n'y ont trouvés que des notions académiques". Consequently, Lagrange had to lower continuously his level and finished his fifth and last lecture by explaining nicely Newton's interpolation formula and then his own version, now called "Lagrange's interpolation formula".

With more enthusiasm than Lagrange, Walter Gautschi took this formula as an occasion for an impressive overview of interpolation "before Lagrange":

Newton, a curious (and failed) attempt by Euler to interpolate the common logarithm from its values at the powers of ten [GA**186**], Waring [2] predating Lagrange by 16 years, and “after Lagrange”: the understanding of error and convergence (Cauchy, Runge) and many modern applications (barycentric formulas, Hermite–Fejér interpolation, quadrature, ODE’s and PDE’s).

References

[1] Skutsch, Rudolf. Über Formelpaare der mechanischen Quadratur, *Arch. Math. Phys.* **13** (1895), 78–83.

[2] Waring, Edward. Problems concerning interpolations, *Philos. Trans. Roy. Soc. London* **69** (1779), 59–67.

41

Miscellanea

Martin J. Gander

The paper to be commented on is [GA**222**]. I remember well when the work started which led to this joint paper by Walter Gautschi and Ernst Hairer: Walter was visiting Geneva and gave a seminar in our numerical analysis seminar series on June 12th 2018, which contained interesting conjectures going back to Stenger. After the seminar we had very lively discussions, and Ernst Hairer showed us on the board in my office a technique used in the theory of collocation Runge–Kutta methods in ordinary differential equations, which seemed related to the Stenger conjecture. After Walter returned back to Purdue University, the discussions continued and Walter and Ernst managed to prove the Stenger conjecture in several cases, but also found cases where it does not hold, which led to the present manuscript.

What are the Stenger conjectures? Consider the zeros x_j of an orthogonal polynomial of a fixed degree for a given weight function $w(x)$ on the interval $[a, b]$, and the Lagrange polynomials which equal one at one of the zeros and zero at all the others. If one integrates these Lagrange polynomials either between $[a, x_j]$ or $[x_j, b]$ and then puts the result into a matrix U resp. V indexed by the Lagrange polynomial index for the columns and the index of the zero for the rows, Stenger conjectured that the eigenvalues of the matrices U and V lie in the open right half of the complex plane. Walter and Ernst call this the restricted Stenger conjecture, and they also consider the case when, in the integration of the Lagrange polynomial, one uses the weight function w, which they call the extended Stenger conjecture, and which can also be considered for unbounded intervals. If the weight function is the identity, i.e. for Legendre polynomials, the restricted and extended Stenger conjectures coincide, and are simply called the Stenger conjecture.

In the present manuscript Walter and Ernst first prove that the Stenger conjecture holds, by using a symmetry argument that allows them to study only the U matrix. They next prove that the restricted Stenger conjecture also holds for the weight function $w(x) = 1 - x$ on $[-1, 1]$, i.e. a special case of the Jacobi weight functions, and again symmetry is used for the proof. At first one might think that the proof also works for slightly more general Jacobi weight functions, but this is not the case, and is illustrated by a counterexample using a Gegenbauer polynomial of degree five.

This counterexample is typical for many other weight functions as well, and Walter and Ernst give several conjectures for the restricted Stenger conjecture which are thoroughly based on computational evidence, in particular for Gegenbauer and Jacobi polynomials, and for algebraic/logarithmic weight functions, where in each case the conjecture only holds under additional conditions.

They then prove for the extended Stenger conjecture that it also holds for the special case of the Jacobi weight function above, and extensive numerical evidence is presented that suggests that the extended Stenger conjecture holds in more cases than the restricted one, namely for Gegenbauer and Jacobi polynomials, and also algebraic/logarithmic weight functions. In each case the eigenvalues are also plotted and one can see that they seem to converge to lie on specific curves in the complex plane as the matrices become larger.

Walter and Ernst then further investigate Laguerre and generalized Laguerre weight functions, and Hermite and generalized Hermite weight functions, and still the extended Stenger conjecture seems to hold. To push this further, Walter and Ernst present a case of a weight function supported on two disjoint intervals, so the function is not strictly positive, and the extended Stenger conjecture still seems to hold.

Finally, for discrete weight functions, and also block-discrete and epsilon-block-discrete weight functions, very carefully crafted numerical experiments show that the extended Stenger conjecture does not hold any more. So while it seems that the extended Stenger conjecture is more natural than the restricted one and holds in many more cases than the restricted one, it is still false for certain weight functions.

Part II

Reprints

42

Papers on Special Functions

172 (with F. E. Harris and N. M. Temme), Expansions of the exponential integral in incomplete gamma functions, Appl. Math. Lett. 16, 1095–1099 (2003)

212 Algorithm 957: Evaluation of the repeated integrals of the coerror function by half-range Gauss–Hermite quadrature, ACM Trans. Math. Software 42, Article 9, 10 pages (2016)

213 Monotonicity properties of the zeros of Freud and sub-range Freud polynomials, Math. Comp. 86, 855–864 (2017)

216 On the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials, in Frontiers in orthogonal polynomials and q-series (M. Zuhair Nashed and Xin Li, eds.), 251–266, World Sci.Publ., Hackensack, NJ, 2018

218 On the zeros of subrange Jacobi polynomials, Numer. Algorithms 79, 759–768 (2018)

42

42.1. [172] “Expansions of the exponential integral in incomplete gamma functions”

[172] “Expansions of the exponential integral in incomplete gamma functions”, *Appl. Math. Lett.* **16**, 1095–1099 (2003).

PERGAMON

Applied Mathematics Letters 16 (2003) 1095–1099

Applied Mathematics Letters

www.elsevier.com/locate/aml

Expansions of the Exponential Integral in Incomplete Gamma Functions

W. GAUTSCHI
Department of Computer Sciences, Purdue University
West Lafayette, IN 47907-1398, U.S.A.

F. E. HARRIS
Quantum Theory Project, University of Florida
P.O. Box 118435, Gainesville, FL 32611, U.S.A.

N. M. TEMME
Center for Mathematics and Computer Science (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

(Received and accepted August 2002)

Communicated by F. Brezzi

Abstract—An apparently new expansion of the exponential integral E_1 in incomplete gamma functions is presented and shown to be a limiting case of a more general expansion given by Tricomi in 1950 without proof. This latter expansion is proved here by interpreting it as a "multiplication theorem". A companion result, not mentioned by Tricomi, holds for the complementary incomplete gamma function and can be applied to yield an expansion connecting E_1 of different arguments. A general method is described for converting a power series into an expansion in incomplete gamma functions. In a special case, this provides an alternative derivation of Tricomi's expansion. Numerical properties of the new expansion for E_1 are discussed. © 2003 Elsevier Ltd. All rights reserved.

1. INTRODUCTION

The exponential integral

$$E_1(z) = \int_z^\infty \frac{e^{-t}}{t}\,dt \tag{1.1}$$

occurs widely in applications, most notably in quantum-mechanical electronic structure calculations. In view of the extremely large number of evaluations that are often required, there is a continuing interest in improving the efficiency of its calculation. In a search for better methods of evaluating E_1, one of us (F.E.H.) discovered the expansion

$$E_1(z) = -\gamma - \ln z + \sum_{n=1}^{\infty} \frac{\gamma(n,z)}{n!}, \tag{1.2}$$

F.E.H. acknowledges financial support from the U.S. National Science Foundation, Grant DMR-9980015. He is also employed at the Department of Physics, University of Utah.

0893-9659/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
doi: 10.1016/S0893-9659(03)00149-6

where

$$\gamma(a,z) = \int_0^z e^{-t}t^{a-1}\,\mathrm{d}t$$

is the incomplete gamma function (cf. [1, Section 6.5]). Another of us (W.G,) observed the relevance of an expansion given in 1950 by Tricomi, of which (1.2) is a limiting case.

2. AN EXPANSION OF TRICOMI

In 1950, Tricomi [2, equation (45)] stated without proof the expansion

$$\gamma(a,\lambda z) = \lambda^a \sum_{n=0}^{\infty} \frac{\gamma(a+n,z)}{n!}(1-\lambda)^n. \tag{2.1}$$

For any fixed complex $a \neq 0, -1, -2, \dots$, the left-hand side is analytic in the domain $\lambda z \in \mathbb{C}\backslash\mathbb{R}_-$, where $\mathbb{R}_-$ is the negative real axis; it is an entire function if a is a positive integer. For fixed a and z, the series in (2.1) converges for arbitrary complex λ.

An interesting proof derives from the observation that (2.1) is a "multiplication theorem" (see [3, Volume 1, Section 6.14]). Such theorems can usually be obtained when all derivatives of the function to be expanded can be expressed in terms of the same family of functions [3, Volume 1, Section 6.14]. In the present instance, we have the relation

$$\gamma(a+n,z) = (-1)^n z^{a+n} \frac{\mathrm{d}^n}{\mathrm{d}z^n}\left(z^{-a}\gamma(a,z)\right), \tag{2.2}$$

which follows readily from the integral representation

$$z^{-a}\gamma(a,z) = \int_0^1 e^{-zt}t^{a-1}\,\mathrm{d}t.$$

When using (2.2) in the right-hand side of (2.1), one obtains

$$\lambda^a z^a \sum_{n=0}^{\infty} \frac{h^n}{n!}\frac{\mathrm{d}^n}{\mathrm{d}z^n}\left(z^{-a}\gamma(a,z)\right), \qquad h = (\lambda - 1)z. \tag{2.3}$$

The series can be seen to be the Taylor expansion of $(z+h)^{-a}\gamma(a,z+h)$. Since $z + h = \lambda z$, expression (2.3) becomes

$$\lambda^a z^a (z+h)^{-a}\gamma(a,z+h) = \gamma(a,\lambda z).$$

This completes the proof of (2.1).

Multiplication theorems (and related addition theorems) are available for many other special functions, such as Bessel functions [4, Chapter 11; 1, p. 363; 5, Section 4.10; 6, Chapter 5, Section 5, Chapter 8, Section 6] and orthogonal polynomials [5, Section 9.8; 7, Section 4.10(7)]. Equation (2.1) is a special case of a multiplication theorem for confluent hypergeometric functions [3, Section 6.14].

3. DERIVATION OF (1.2) FROM TRICOMI'S EXPANSION

Separating out the first term on the right of (2.1) and bringing it to the left, we write Tricomi's result in the form

$$\frac{\gamma(a,\lambda z) - \lambda^a\gamma(a,z)}{\lambda^a} = \sum_{n=1}^{\infty} \frac{\gamma(a+n,z)}{n!}(1-\lambda)^n. \tag{3.1}$$

From the power series of $\gamma(a,\lambda z)$ (cf. [1, equation 6.5.29]) one gets

$$\frac{\gamma(a,\lambda z) - \lambda^a\gamma(a,z)}{\lambda^a} = \frac{z^a}{a}\left[1 - \frac{a}{a+1}\lambda z + \frac{a}{2a+4}(\lambda z)^2 + \cdots\right] - \gamma(a,z),$$

which, as $\lambda \downarrow 0$, has the limit $z^a/a - \gamma(a,z)$. Thus, by (3.1),

$$\frac{z^a}{a} - \gamma(a,z) = \sum_{n=1}^{\infty} \frac{\gamma(a+n,z)}{n!}. \tag{3.2}$$

If $\gamma(a,z)$ on the left is replaced by $\Gamma(a) - \Gamma(a,z)$ and $\Gamma(a)$ written as $\Gamma(a+1)/a$, equation (3.2) takes the form

$$\frac{z^a - \Gamma(a+1)}{a} + \Gamma(a,z) = \sum_{n=1}^{\infty} \frac{\gamma(a+n,z)}{n!}.$$

Now take the limit $a \downarrow 0$. Applying Bernoulli-l'Hospital's rule to the first term on the left and noting that $\Gamma'(1) = -\gamma$ and $\Gamma(0,z) = E_1(z)$, one gets

$$\ln z + \gamma + E_1(z) = \sum_{n=1}^{\infty} \frac{\gamma(n,z)}{n!},$$

which proves (1.2).

4. A COMPANION TO TRICOMI'S EXPANSION WITH AN APPLICATION TO THE EXPONENTIAL INTEGRAL

There is a companion result to (2.1), not mentioned by Tricomi, for the complementary incomplete gamma function,

$$\Gamma(a,\lambda z) = \lambda^a \sum_{n=0}^{\infty} \frac{\Gamma(a+n,z)}{n!}(1-\lambda)^n, \qquad |\lambda - 1| < 1. \tag{4.1}$$

This follows from (2.1) by inserting the definition $\gamma(a,z) = \Gamma(a) - \Gamma(a,z)$ in both sides of the expansion and noting that, by Taylor's series for λ^{-a} at $\lambda = 1$, one has

$$\sum_{n=0}^{\infty} \frac{\Gamma(a+n)(1-\lambda)^n}{n!} = \lambda^{-a}\Gamma(a), \qquad |\lambda - 1| < 1. \tag{4.2}$$

Equation (4.1) holds also for $a = 0$, by analytic continuation, and yields

$$E_1(\lambda z) = E_1(z) + \sum_{n=1}^{\infty} \frac{\Gamma(n,z)}{n!}(1-\lambda)^n, \qquad |\lambda - 1| < 1. \tag{4.3}$$

Here, the coefficients are elementary functions

$$\frac{\Gamma(n,z)}{n!} = \frac{1}{n} e^{-z} e_{n-1}(z), \qquad n \geq 1,$$

where $e_m(z) = 1 + z + z^2/2! + \cdots + z^m/m!$ are the partial sums of the exponential series. These can be generated by recursion in a stable fashion, at least when z is real (cf. [8]). If $\lambda = 1/2$, for example, then

$$E_1\left(\frac{z}{2}\right) = E_1(z) + \sum_{n=1}^{\infty} \frac{\Gamma(n,z)}{2^n n!}.$$

For positive z, the sum converges more rapidly than that of $(2^n n)^{-1}$.

5. OTHER EXPANSIONS IN INCOMPLETE GAMMA FUNCTIONS

We return to (3.2) and write it as

$$\frac{z^{a+k}}{a+k} = \sum_{n=k}^{\infty} \frac{\gamma(a+n,z)}{(n-k)!}, \qquad k \geq 0. \tag{5.1}$$

We next form the following series, with arbitrary d_k (subject to convergence), and insert (5.1) in it,

$$\begin{aligned} \sum_{k=0}^{\infty} \frac{d_k z^{a+k}}{k!(a+k)} &= \sum_{k=0}^{\infty} d_k \sum_{n=k}^{\infty} \binom{n}{k} \frac{\gamma(a+n,z)}{n!} \\ &= \sum_{n=0}^{\infty} \frac{\gamma(a+n,z)}{n!} \sum_{k=0}^{n} \binom{n}{k} d_k \\ &= \sum_{n=0}^{\infty} c_n \frac{\gamma(a+n,z)}{n!}, \end{aligned} \tag{5.2}$$

where

$$c_n = \sum_{k=0}^{n} \binom{n}{k} d_k. \tag{5.3}$$

Any power series that can be cast in the form given in the left-hand side of (5.2) can therefore be written as a series in incomplete gamma functions.

We illustrate the procedure by applying (5.2) to $\gamma(a, \lambda z)$, which has the power series expansion

$$\gamma(a, \lambda z) = \sum_{k=0}^{\infty} \frac{(-1)^k \lambda^{a+k} z^{a+k}}{k!(a+k)}.$$

In this example, $d_k = (-1)^k \lambda^{a+k}$, and from (5.3) we obtain $c_n = \lambda^a (1-\lambda)^n$, thereby recovering (2.1).

6. NUMERICAL PROPERTIES OF (1.2)

Tricomi [2, p. 148] expressed the thought that some of the series expansions he listed without proof, including (2.1), might prove useful also for computational purposes. We discuss here the computational merits of the series (1.2), which, as has been shown, is a limiting case of (2.1).

Compared with the power series in

$$E_1(z) = -\gamma - \ln z + \sum_{n=1}^{\infty} \frac{(-1)^{n-1} z^n}{n n!}, \tag{6.1}$$

the series in (1.2) has some definite drawbacks. In (6.1), the terms of the series can be generated recursively in forward direction, $n = 1, 2, 3, \ldots$, until they no longer contribute to the sum within the desired accuracy. This is not possible with (1.2). Although it is true that the terms in (1.2) also satisfy a forward recursion,

$$\begin{aligned} \gamma(n+1, z) &= n\gamma(n, z) - z^n e^{-z}, \qquad n = 1, 2, 3, \ldots, \\ \gamma(1, z) &= 1 - e^{-z}, \end{aligned} \tag{6.2}$$

the recursion becomes severely unstable as n exceeds $|z|$. (This can be shown by an analysis similar to the one in [9, Section 2.4].) To preserve numerical stability when $n > |z|$, one must generate $\gamma(\nu, z), \gamma(\nu-1, z), \ldots, \gamma(n, z)$ backwards with ν chosen sufficiently large, whereby $\gamma(\nu+1, z)$ may be replaced by zero. The choice of ν depends on the number of terms in (1.2) required for given accuracy, which has to be estimated *a priori*. Thus, summing the series to a prescribed accuracy is considerably more involved for the series in (1.2) than it is for the one in (6.1).

Another important consideration is internal cancellation of terms in a series. In this regard, the series in (1.2) and (6.1) complement each other. There is no significant cancellation of terms in either series if $|z|$ is relatively small, say $|z| \leq 5$. For larger values of $|z|$, the severity of cancellation increases with increasing $\arg z$ for the series in (1.2) and decreases with increasing $\arg z$ for the series in (6.1). Near the positive real axis ($\arg z \approx 0$) the series (1.2) is practically free of cancellation but subject to severe cancellation near the negative real axis ($\arg z \approx \pi$), more so the larger $|z|$. For the series (6.1), it is just the other way around.

With regard to speed of convergence, the two series are comparable, since for bounded z, as $n \to \infty$, one has $\gamma(n,z)/n! \sim z^n e^{-z}/(nn!)$ (cf. [10, Section 4.3, equation (3)]).

Another source of impaired accuracy is the cancellation that may occur when the series in (1.2), respectively, (6.1) is added to $-\gamma - \ln z$. This can be quite pronounced if $|z|$ is large and z near the positive real axis. The severity of the problem diminishes as $\arg z$ increases and becomes negligible near the negative real axis.

REFERENCES

1. M. Abramowitz and I.A. Stegun, Editors, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Reprint of the 1972 Edition, Dover, New York, (1992).
2. F.G. Tricomi, Asymptotische Eigenschaften der unvollständigen Gammafunktion, *Math. Z.* **53** (2), 136–148, (1950).
3. A. Erdélyi, W. Magnus, F. Oberhettinger and F.G. Tricomi, *Higher Transcendental Functions, Volumes 1–3*, Robert Krieger, Melbourne, FL, (1981).
4. G.N. Watson, *A Treatise on the Theory of Bessel Functions*, Cambridge Mathematical Library, Reprint of the Second (1944) Edition, Cambridge University Press, Cambridge, (1995).
5. G.E. Andrews, R. Askey and R. Roy, *Special Functions, Encyclopedia of Mathematics and Its Applications, Volume 71*, Cambridge University Press, Cambridge, (1999).
6. H. Hochstadt, *The Functions of Mathematical Physics*, Second Edition, Dover, New York, (1986).
7. G. Szegö, *Orthogonal Polynomials, American Mathematical Society, Colloquium Publications, Volume 23*, Fourth Edition, American Mathematical Society, Providence, RI, (1975).
8. W. Gautschi and B.J. Klein, Recursive computation of certain derivatives—A study of error propagation, *Comm. ACM* **13** (1), 7–9, (1970).
9. W. Gautschi, The computation of special functions by linear difference equations, In *Advances in Difference Equations (Veszprém, 1995)*, (Edited by S. Elaydi, I. Győri and G. Ladas), pp. 213–243, Gordon and Breach, Amsterdam, (1997).
10. F.G. Tricomi, *Funzioni Ipergeometriche Confluenti*, Edizioni Cremonese, Roma, (1954).

42.2. [212] “Algorithm 957: Evaluation of the repeated integrals of the coerror function by half-range Gauss–Hermite quadrature”

[212] “Algorithm 957: Evaluation of the repeated integrals of the coerror function by half-range Gauss–Hermite quadrature”, *ACM Trans. Math. Software* **42**, Article 9, 10 pages (2016).

Algorithm 957: Evaluation of the Repeated Integral of the Coerror Function by Half-Range Gauss–Hermite Quadrature

WALTER GAUTSCHI, Purdue University

Nonstandard Gaussian quadrature is applied to evaluate the repeated integral $\mathrm{i}^n\mathrm{erfc}\, x$ of the coerror function for $n \in \mathbb{N}_0$, $x \in \mathbb{R}$ in an appropriate domain of the (n, x)-plane. Relevant software in MATLAB is provided: in particular, two routines evaluating the function to an accuracy of 12 respective 30-decimal digits.

Categories and Subject Descriptors: G.1.4 [**Numerical Analysis**]: Quadrature and Numerical Differentiation; G.4 [**Mathematical Software**]: Algorithm Design and Analysis

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Half-range Gauss–Hermite quadrature, MATLAB software, repeated integral of the coerror function

ACM Reference Format:
Walter Gautschi. 2016. Algorithm 957: Evaluation of the repeated integral of the coerror function by half-range Gauss–Hermite quadrature. ACM Trans. Math. Softw. 42, 1, Article 9 (February 2016), 10 pages.
DOI: http://dx.doi.org/10.1145/2735626

1. INTRODUCTION

The integrals in the title are of considerable interest in physics and chemistry, notably in problems involving heat and mass transfer. They are traditionally evaluated by the three-term recurrence relation that they satisfy [Gautschi 1961; Amos 1973]. This involves, even if done carefully, controlled loss of accuracy [Gautschi 1977]. On the other hand, a whole sequence of integrals is produced, as may be required in some applications. Here, we propose a method based on quadrature that, involving the summation of a finite sum of positive terms, is perfectly stable and allows the computation of just one of these integrals. The quadrature entails nonclassical Gaussian integration and the half-range Hermite polynomials orthogonal with respect to the weight function $\exp(-t^2)$ on $[0, \infty]$. An important issue is the determination of a natural domain in the (n, x)-plane in which to evaluate the function.

2. INTEGRAL REPRESENTATION

The function in question is

$$f_n(x) = \mathrm{i}^n \text{ erfc } x, \quad n = -1, 0, 1, 2, \dots, \; x \in \mathbb{R}, \tag{2.1}$$

Author's address: W. Gautschi, Department of Computer Science, Purdue University, West Lafayette, IN 47907-2066; email: wgautschi@purdue.edu.

DOI: http://dx.doi.org/10.1145/2735626

where

$$f_{-1}(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}, \qquad (2.2)$$
$$f_n(x) = \int_x^\infty f_{n-1}(t)\mathrm{d}t, \quad n = 0, 1, 2, \ldots.$$

As is well known from calculus, f_n can be expressed as a single integral,

$$f_n(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \frac{(t-x)^n}{n!} e^{-t^2}\mathrm{d}t, \quad n \geq 0. \qquad (2.3)$$

Changing variables, $t - x = \tau$, we can write

$$f_n(x) = \frac{2}{\sqrt{\pi}} \int_0^\infty \frac{\tau^n}{n!} e^{-(\tau+x)^2}\mathrm{d}\tau, \qquad (2.4)$$

that is,

$$f_n(x) = \frac{2}{\sqrt{\pi}} \frac{e^{-x^2}}{n!} \int_0^\infty t^n e^{-2xt} e^{-t^2}\mathrm{d}t. \qquad (2.5)$$

Our method is based on this integral representation in which the integral will be evaluated by Gaussian quadrature; see Section 4.

Incidentally, the same integral representation can also be used to extend the meaning of the function $f_n(x)$ from nonnegative integer values of n to arbitrary real values $n = \nu > -1$ if the factorial $n!$ in Equation (2.5) is replaced by the gamma function $\Gamma(\nu + 1)$. Here, however, we will stick to integer values. Likewise, Equation (2.5) could be used as a basis for computing $f_n(z)$ for complex z. Again, we restrict ourselves here to real values of x.

Before developing computational methods, it seems worthwhile to figure out a natural (n, x)-domain in which to evaluate $f_n(x)$. This will be discussed in Section 3. At this point, we briefly note that when x is large negative, the integral in Equation (2.5) is prone to overflow. This can be avoided, or at least deferred, if one places the factor $\exp(-x^2)$ inside the integral, that is, if one uses the alternative form

$$f_n(x) = \frac{2}{\sqrt{\pi} n!} \int_0^\infty t^n e^{-(x^2+2xt)} e^{-t^2}\mathrm{d}t, \quad x < 0. \qquad (2.6)$$

We assume, without further notice, that this will be done in any computation involving negative values of x.

3. AN APPROPRIATE (*N*, *X*)-DOMAIN

In this section, we determine a natural domain $\mathcal{D}$ in which to evaluate the function $f_n(x)$. With regard to the variable n, we require that $n!$ does not overflow, which implies $n \leq 171$ for IEEE standard double precision. We shall assume, more conservatively, that $n \leq 150$.

When the variable x is nonnegative, the domain $\mathcal{D}$ is going to be the one in which $f_n(x)$, evaluated in MATLAB double-precision arithmetic, neither underflows nor overflows. This is discussed in Sections 3.1 and 3.2. Otherwise, when $x < 0$, the determining factor is not under- or overflow, but the numerical quadrature of the integral in Equation (2.6); see Section 3.3. For large negative values of x, this becomes more difficult and requires higher-order quadrature rules. Since we use nonstandard Gaussian quadrature rules, which currently are available only up to order 200, the domain $\mathcal{D}$, when $x < 0$, will be determined by the requirement that not more than 200 quadrature points be needed to evaluate the integral in Equation (2.6) to 12 significant decimal digits. Under- and overflow then ceases to be an issue. This is analyzed in Section 4.2.

3.1. Underflow

Since MATLAB's double-precision arithmetic conforms with the IEEE standard, the smallest positive floating-point number is `realmin` = 2.2251e–308; any positive number smaller than $\lambda = 2.2251 \times 10^{-308}$ gives rise to underflow.

We first observe that $f_n(x)$ never underflows when $x < 0$ (and $0 \le n \le 150$). Indeed, let $x = -\xi$, $\xi > 0$. Then, by Equation (2.3),

$$f_n(-\xi) = \frac{2}{\sqrt{\pi}\, n!} \int_{-\xi}^{\infty} (t+\xi)^n e^{-t^2} \mathrm{d}t$$
$$= \frac{2}{\sqrt{\pi}\, n!} \left(\int_0^{\infty} (t+\xi)^n e^{-t^2} \mathrm{d}t + \int_{-\xi}^{0} (t+\xi)^n e^{-t^2} \mathrm{d}t \right).$$

Here, the first integral on the right is greater than

$$\int_0^{\infty} t^n e^{-t^2} \mathrm{d}t = \frac{1}{2} \int_0^{\infty} \tau^{(n-1)/2} e^{-\tau} \mathrm{d}\tau = \frac{1}{2}\Gamma\left(\frac{n+1}{2}\right).$$

The second integral is

$$\int_0^{\xi} (\xi - \tau)^n e^{-\tau^2} \mathrm{d}\tau \ge 0,$$

so that

$$f_n(-\xi) \ge \frac{1}{\sqrt{\pi}} \frac{\Gamma((n+1)/2)}{\Gamma(n+1)} = \frac{1}{\sqrt{\pi}} \frac{\Gamma((n+1)/2)}{\Gamma(2\cdot(n+1)/2)}.$$

Using the duplication formula [Abramowitz and Stegun 1964, 6.1.18] for the gamma function, we have that

$$\frac{\Gamma((n+1)/2)}{\Gamma(n+1)} = \frac{\sqrt{2\pi}}{2^n \Gamma((n+2)/2)}, \tag{3.1}$$

thus,

$$f_n(-\xi) \ge \frac{\sqrt{2}}{2^n \Gamma((n+2)/2)} \ge \frac{\sqrt{2}}{2^{150}\Gamma(152/2)} > 3.99 \times 10^{-155} > \lambda, \quad \xi > 0. \tag{3.2}$$

Therefore, underflow cannot occur, and we can assume that $x \ge 0$. For each n, $0 \le n \le 150$, we determine experimentally the value $x = x^*(n)$ of x for which $f_n(x)$ underflows. The tool for this is the routine `quad_inerfc.m`[1] (calling on the routine `gauss.m` of the `OPQ` package [Gautschi 2014]) evaluating $f_n(x)$ by numerical quadrature of Equation (2.5); for details, see Section 4. This routine returns not only $y = f_n(x)$, but also two output variables `uflow` and `oflow`, which are 1 or 0 depending on whether or not underflow of y respective overflow has occurred. This is detected by the routine returning $y = 0$ respectively $y =$ `Inf`. Since we are interested in underflow, the procedure, then, is the following. We increase x from $x = 10$ in steps of 1 until the routine `quad_inerfc.m` returns `uflow` = 1 for the first time. We call b the corresponding value of x and let $a = b - 1$. We thus have an interval $[a, b]$, $a < b$, with the property that $y = f_n(a)$ does not underflow, but $f_n(b)$ does. We take this as the initial stage of a bisection method, which will halve the interval $[a, b]$ and check whether underflow does or does not occur

[1] All Matlab programs referenced in this paper can be accessed at http://dx.doi.org/10.4231/R7959FHP.

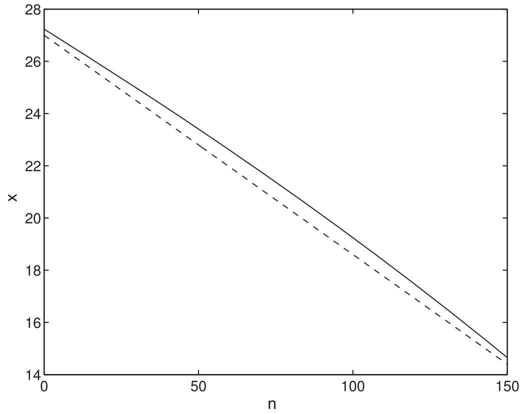


Fig. 1. The function $x_n^*(n)$ and its linear approximant.

at the midpoint m. If it does occur, we replace the right end point b; otherwise, we replace the left end point a, by m. We continue until the interval $[a, b]$ has become sufficiently small. (The number of bisection steps required can be determined *a priori* in a well-known manner; e.g., see Gautschi [2012, Section 4.3.1].) This is implemented in the routine `xunder.m`. The resulting function $x = x_n^*(n)$ has been computed in this manner for $n = 0 : 150$, and is plotted, using `run_xunder.m`, in Figure 1 as the solid line. It is slightly concave and decreasing, and is easily approximated from below by a straight line, as shown in Figure 1 by the dashed line. Its equation is $x = 27 - .084n$. Thus, the trapezoidal domain bounded on top by this straight line, on the left and right by the vertical lines $n = 0$ and $n = 150$, and at the bottom by the real line, is a close approximation to the desired domain $\mathcal{D}$ when $x \geq 0$. We denote it by $\mathcal{D}^+$.

3.2. Overflow

Overflow occurs if a number is produced that exceeds `realmax` $= 1.8 \times 10^{308}$. It is clear that overflow is not an issue when $x \geq 0$, since by Equation (2.5),

$$f_n(x) \leq \frac{2}{\sqrt{\pi}} \frac{e^{-x^2}}{n!} \int_0^\infty t^n e^{-t^2} \mathrm{d}t = \frac{1}{\sqrt{\pi}} e^{-x^2} \frac{\Gamma((n+1)/2)}{\Gamma(n+1)},$$

and using Equation (3.1),

$$f_n(x) \leq \frac{\sqrt{2}e^{-x^2}}{2^n \Gamma((n+2)/2)} \leq \sqrt{2}e^{-x^2} \leq \sqrt{2}, \quad x \geq 0, n = 0, 1, 2, \ldots.$$

For the case $x < 0$, as in Section 3.1, we let $x = -\xi$, $\xi > 0$, and first derive an upper bound for $f_n(-\xi)$. By Equation (2.4), we have that

$$\frac{\sqrt{\pi}}{2} n!\, f_n(-\xi) = \int_0^\infty t^n e^{-(t-\xi)^2} \mathrm{d}t, \quad \xi > 0.$$

Breaking up the integral on the right in two, one from 0 to ξ and the other from ξ to ∞, one finds that

$$\frac{\sqrt{\pi}}{2} n!\, f_n(-\xi) = \xi^{n+1} \left(\int_0^1 (1-t)^n e^{-\xi^2 t^2} \mathrm{d}t + \int_0^\infty (1+t)^n e^{-\xi^2 t^2} \mathrm{d}t \right),$$

and since the first integral is bounded by 1,

$$\frac{\sqrt{\pi}}{2} n!\, f_n(-\xi) < \xi^{n+1} \left(1 + \int_0^\infty (1+t)^n e^{-\xi^2 t^2} \mathrm{d}t \right). \tag{3.3}$$

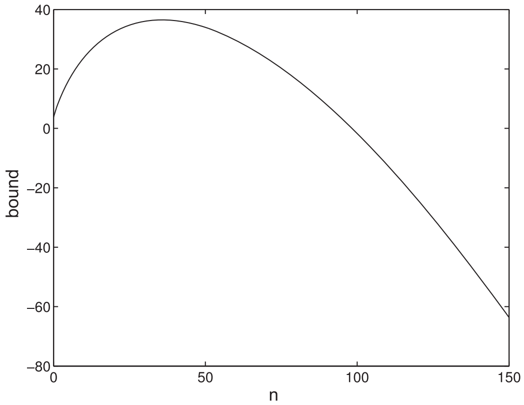


Fig. 2. Logarithm of the bound in Equation (3.4) when $\xi = 18.1$.

Here again, we split the integral in two, one from 0 to 1 and the other from 1 to ∞, using $1 + t \le 2$ in the former and $1 + t \le 2t$ in the latter, to get

$$\int_0^\infty (1+t)^n e^{-\xi^2 t^2} \mathrm{d}t < 2^n \left(1 + \int_1^\infty t^n e^{-\xi^2 t^2} \mathrm{d}t\right).$$

In the integral on the right, we change variables, $\xi^2 t^2 = \tau$, and find that

$$\int_1^\infty t^n e^{-\xi^2 t^2} \mathrm{d}t = \frac{1}{2\xi^{n+1}} \int_{\xi^2}^\infty \tau^{(n-1)/2} e^{-\tau} \mathrm{d}\tau$$
$$< \frac{1}{2\xi^{n+1}} \int_0^\infty \tau^{(n-1)/2} e^{-\tau} \mathrm{d}\tau = \frac{\Gamma((n+1)/2)}{2\xi^{n+1}}.$$

Inserting these inequalities in Equation (3.3) yields

$$\frac{\sqrt{\pi}}{2}\, n!\, f_n(-\xi) < \xi^{n+1}(1+2^n) + 2^{n-1}\Gamma((n+1)/2);$$

thus, using again Equation (3.1),

$$f_n(-\xi) < \frac{2(1+2^n)}{\sqrt{\pi}\, n!}\, \xi^{n+1} + \frac{\sqrt{2}}{\Gamma((n+2)/2)}. \tag{3.4}$$

In Section 4.2, it will be seen that ξ will have to be restricted to $\xi \le 18.1$ for the numerical evaluation of the integral in Equation (2.6) by Gaussian quadrature to be possible for all n with $0 \le n \le 150$. But then, the upper bound in Equation (3.4) is far from overflowing. The logarithm of the bound for $\xi = 18.1$ has the behavior shown in Figure 2 as a function of n. The routine `bound_Fn_neg.m` producing this plot also determines that the maximum occurs at $n = 37$ and has the value 36.5031. The maximum of $f_n(-\xi)$ when ξ is restricted as mentioned, therefore, is $\exp(36.5031) = 7.1303 \times 10^{15}$, way too small to cause overflow. Thus, the lower boundary of $\mathcal{D}$ must be taken to be above the straight line $x = -18.1$. A more precise boundary is determined in the next section.

3.3. The Lower Boundary of $\mathcal{D}$

Just as in Section 3.1, in which the upper boundary of $\mathcal{D}$ was determined experimentally to be the function $x = x^*(n)$—the smallest (positive) value of x for which $f_n(x)$ underflows—we now determine the lower boundary of $\mathcal{D}$ to be the function $x = x^{**}(n)$,

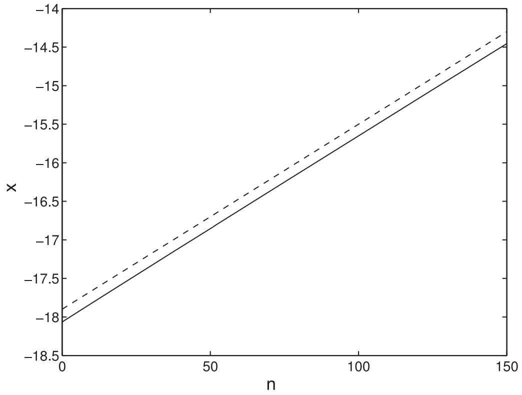


Fig. 3. The function $x_n^{**}(n)$ and its linear approximant.

the largest (negative) value of x for which Gauss quadrature fails on account of requiring more than 200 quadrature points for 12-digit accuracy. The routine for computing $x = x^{**}(n)$ is `xgo200.m`, resembling `xunder`, but calling on the routine `N_inerfc.m` that signals failure of Gauss quadrature by an output variable `go200`, which is 1 if Gauss quadrature fails, and 0 otherwise. The graph of the function $x = x^{**}(n)$, practically a straight line, as generated by the routine `run_xgo200.m`, is shown in Figure 3. As in Figure 1, it can be approximated from above by a straight line, $x = -17.9 + .024n$, which will be taken to be the lower boundary of $\mathcal{D}$.

4. THE QUADRATURE METHOD

Both integrals in Equations (2.5) and (2.6) can be evaluated by Gaussian quadrature relative to the weight function $w(t) = e^{-t^2}$ supported on the half-infinite interval $[0, \infty]$, that is, by half-range Gauss–Hermite quadrature. Thus, in the case of Equation (2.5), for example,

$$\int_0^\infty t^n e^{-2xt} w(t)\mathrm{d}t = \sum_{\nu=1}^N \lambda_\nu^G \big(\tau_\nu^G\big)^n e^{-2x\tau_\nu^G} + R_N, \tag{4.1}$$

where τ_ν^G, λ_ν^G are the N-point Gauss nodes and weights for the weight function w and R_N is the remainder term. The summation in Equation (4.1) involves only positive terms, so that the computation is perfectly stable.

With regard to the special (nonstandard) Gaussian quadrature formula required, it can be generated by well-known methods [Gautschi 2004, Section 3.1.1] from the eigenvalues and eigenvectors of the Jacobi matrix belonging to the half-range Hermite weight function. Related software in MATLAB is readily available to carry this out.

Specifically, the MATLAB command

$$\texttt{xw = gauss(N,ab)} \tag{4.2}$$

where `gauss.m` is a routine in the `OPQ` package [Gautschi 2014], generates the N nodes τ_ν^G and N weights λ_ν^G and returns them in the first respective second column of the $N\times 2$ array `xw`. The $N \times 2$ input array `ab` must contain the necessary data for the half-range Hermite polynomials, namely, in the first column the N coefficients $\{\alpha_k\}_{k=0}^{N-1}$, and in the second column the N coefficients $\{\beta_k\}_{k=0}^{N-1}$ in the three-term recurrence relation

$$\pi_{k+1}(t) = (t-\alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, \ldots, N-1,$$
$$\pi_0(t) = 1, \quad \pi_{-1}(t) = 0$$

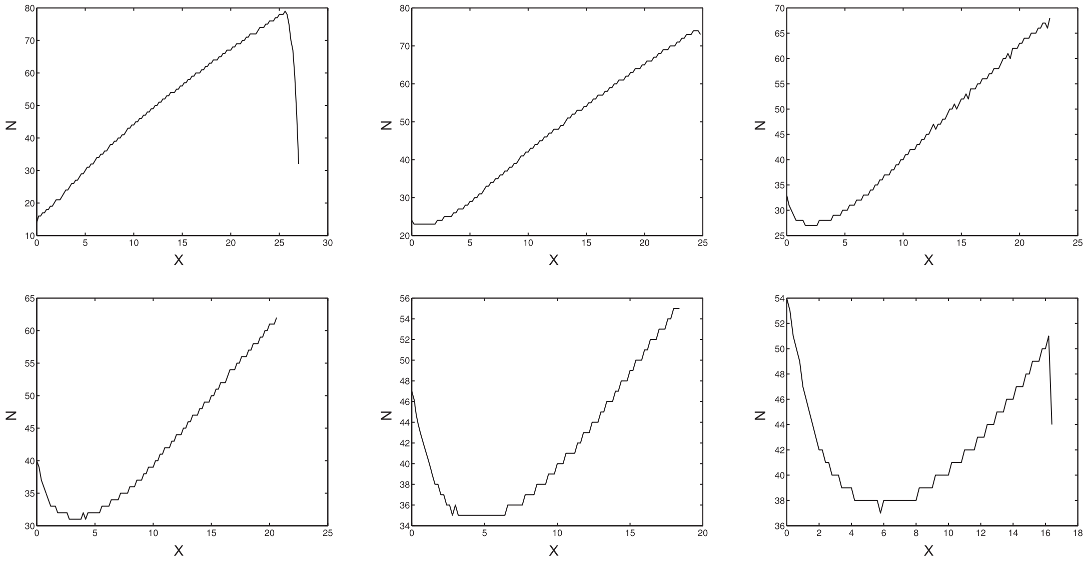


Fig. 4. The number N of Gauss points needed for 12 digit accuracy when the n-intervals, from top left to bottom right, are $[0, 25]$, $[25, 50]$, $[50, 75]$, $[75, 100]$, $[100, 125]$, and $[125, 150]$.

satisfied by these polynomials. The first 200 coefficients are available in the file `ab_hrhermite` to 32-digit accuracy and have been generated in Gautschi [to appear, Section 1.4] by the classical Chebyshev algorithm in high-precision arithmetic. The command[2]

$$\texttt{ab} = \texttt{loadvpa('ab\_hrhermite', 3,32,200,2)} \tag{4.3}$$

produces the symbolic 200×2 array `ab`, which, if necessary, can be converted to a MATLAB double-precision array `ab0` by the command `ab0=double(ab)`. The command (4.2) can thus be used to produce all N-point Gauss formulae for $N = 1, 2, \ldots, 200$.

It is important, however, to have some idea about the number N of Gauss points needed to achieve a prescribed accuracy. This is discussed in Sections 4.1 and 4.2 for $x \geq 0$ and $x < 0$, respectively, and for an accuracy requirement of 12 significant decimal digits. For 30-digit accuracy, see Section 5.

4.1. The Order of the Gauss Quadrature Rule Necessary for $x \geq 0$

In order to find an estimate $N^+(n, x)$ of the number $N(n, x)$ of quadrature points needed to obtain $f_n(x)$ for $x \geq 0$ accurate to 12 decimal digits, it is convenient to break up the n-interval $0 \leq n \leq 150$ into six subintervals,

$$I(n_0) = [n_0 \leq n \leq n_0 + 25], \quad n_0 = 0 : 25 : 125.$$

In each of these intervals, we determine the maximum value $N_{\max}(n_0)$ of N, extended over all $n \in I(n_0)$ and $x = 0 : 0.2 : 27$, $x \in \mathcal{D}^+$, using the routine `N_inerfc.m` and `quad_inerfc.m`. This is done in the script[3] `Nplus_pos.m`. The results are shown in the plots of Figure 4. In each plot, it is easy to bound N from above by a straight line

[2] The routine `loadvpa.m` is not yet part of the MATLAB symbolic toolbox, but has been developed by the MathWorks staff at the request of the author.

[3] In Matlab releases more recent than 2011, this routine may not work properly.

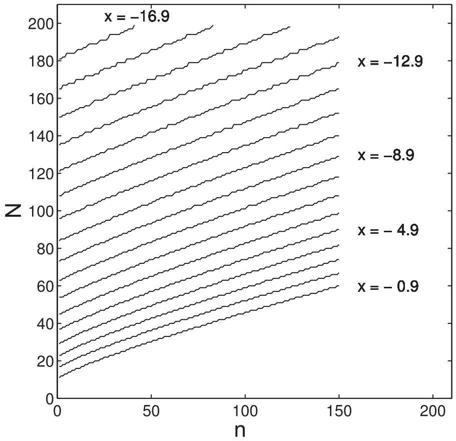


Fig. 5. The number N of Gauss points needed for 12 digit accuracy when $x < 0$.

$N^+ = N^+(n, x)$, $0 \le x \le 27$. Their equations are

$$N^+(n,x) = \begin{cases} \lceil 21.5 + 2.388x \rceil & (0 \le n \le 25), \\ \lceil 24 + 2.08x \rceil & (25 < n \le 50), \\ \lceil 33 + 1.549x \rceil & (50 < n \le 75), \\ \lceil 40 + 1.068x \rceil & (75 < n \le 100), \\ \lceil 47 + .4348x \rceil & (100 < n \le 125), \\ \lceil 54 - .1852x \rceil & (125 < n \le 150). \end{cases}$$

The bounds N^+ are fairly close to the true values of N when $n \le 50$, but can be quite a bit larger when $n > 50$. The results thus obtained for $f_n(x)$ may have an accuracy exceeding 12 decimal digits; for more on this, see Section 5.

4.2. The Order of the Gauss Quadrature Rule Necessary for $x < 0$

The routine Nplus_neg.m has the same objective as the routine Nplus_pos.m, namely, to find a suitable estimate $N^+(n, x)$ for $N(n, x)$ when $x < 0$. Contrary to the latter routine, however, it examines the values of N on the horizontal lines $x = -.9 : -1 : -16.9$ for $0 \le n \le 150$ such that $(n, x) \in \mathcal{D}$ (see Figure 3). The function $y = f_n(x)$ on the lower (straight-line) boundary behaves similar to its bound (Equation (3.4); see Figure 2): it has the value 2.000 at $n = 0$, a maximum 9.384×10^6 at $n = 17$, and the minimum 8.374×10^{-81} at $n = 150$.

The plots for $x = -.9 : -1 : -16.9$ are shown in Figure 5. Being rather regular, they allow us to derive a simple analytic expression for a suitable bound N^+, namely,

$$N^+(n, x) = \min(200, \lceil 10 + 10.5|x| + (.3223 + .00747|x|)n \rceil) \\ \text{if } (n, x) \in \mathcal{D},\ x < 0. \tag{4.4}$$

The condition in the if-clause is equivalent to $|x| \le 17.9 - .024n$, $0 \le n \le 150$, $x < 0$. As in Section 4.1, the bound can be fairly conservative, and yields answers that are often more accurate than the 12 digits striven for.

5. HIGH-PRECISION COMPUTATION

Since the Gauss quadrature rules needed are available to an accuracy of 32 decimal digits, we may as well try to extend the work described so far from the accuracy level of $\varepsilon_0 = .5 \times 10^{-12}$ to the one of, say, $\varepsilon_1 = .5 \times 10^{-30}$. To do so, we need symbolic variable-precision analogues of the five routines mentioned in Sections 4.1 and 4.2, that is, sgauss.m, sN_inerfc.m, squad_inerfc.m, sNplus_pos.m, and sNplus_neg.m, in order to produce plots analogous, but at a lower resolution, to those in Figures 4 and 5. We omit details and simply report on the results obtained for a suitable upper bound N^+.

When $x \geq 0$ and $0 \leq n \leq 150$, then

$$N^+(n,x) = \begin{cases} \lceil 43 + 3.92x \rceil & (0 \leq n \leq 25), \\ \lceil 43 + 4.042x \rceil & (25 \leq n \leq 50), \\ \lceil 39 + 4.143x \rceil & (50 \leq n \leq 75), \\ \lceil 49 + 3.333x \rceil & (75 \leq n \leq 100), \\ \lceil 58 + 2.722x \rceil & (100 \leq n \leq 125), \\ \lceil 67 + 1.733x \rceil & (125 \leq n \leq 150). \end{cases}$$

Otherwise, when $x < 0$, severe restrictions on x must be imposed. If the accuracy of 30 significant digits is to be maintained for all $0 \leq n \leq 150$, the limitation to 200 Gauss quadrature points essentially means that x must be restricted to $x > -1$. If we lower $\texttt{nmax} = 150$ to $\texttt{nmax} = 100$, the restriction is $x > -3.1$, and for $\texttt{nmax} = 50$, it is $x > -5.2$. Assuming, then, that $0 \leq n \leq 50$ and $0 > x \geq -5$, an appropriate choice of N^+ is found to be

$$N^+(n,x) = \begin{cases} \lceil (47 + 37|x|)/4 + .46n \rceil & (|x| \leq 4), \\ \lceil 57 + 2.36n \rceil & (|x| > 4). \end{cases}$$

6. CONCLUSION AND TESTS

We are now in a position to write the final (short) routines, inerfc.m and sinerfc.m, that evaluate $f_n(x)$ to at least 12-digit resp. 30-digit accuracy. The (n,x)-domain in the former routine is bounded on top by the descending straight line $x = 27 - .084n$, at the bottom by the ascending straight line $x = -17.9 + .024n$, and on the left and right by the vertical lines $n = 0$ and $n = 150$. The same domain is valid in the latter routine when $x \geq 0$, except that the boundary on top is the horizontal line $x = 27$. For $x < 0$, however, the vertical boundaries are at $n = 0$ and $n = 50$, and the one at the bottom at $x = -5$. Both routines assume that the calling program has downloaded the symbolic array ab by the command (4.3). The core of the routines is identical with the routines quad_inerfc.m and squad_inerfc.m, except for handling under- and overflow, which is no longer needed. Since our routine inerfc.m calls on the routine gauss.m in (4.2), with ab0=double(ab) in place of ab, the array ab0 must be declared as a global variable in the calling program. The same applies in the case of sinerfc.m and the file ab.

Extensive checks have been run using the routines run_inerfc.m and run_sinerfc.m to verify that the accuracy of 12 resp. 30 significant digits, in the respective domains, have indeed been achieved. This is done by running the routines twice, once with the estimated value $N = N^+$ of N, and once with $N = N^+ + 5$ (when ≤ 200) and comparing the two results to make sure that the former is as accurate as claimed. The points (n,x) chosen for testing the double-precision routine inerfc.m are

$$n = 0:25:150, \qquad x = \begin{cases} 0:.2:1, \\ 1.5:.5:5, \\ 5.5:.5:8, \\ 8.5:.5:12, \\ 13:19, \\ 20:27, \\ -.5:-.5:-17.5, \end{cases} \qquad (n,x) \in \mathcal{D}.$$

The maximum relative discrepancy errmax between the results of the two runs were observed to be 6.53×10^{-14}, 6.95×10^{-14}, 3.54×10^{-14}, 1.33×10^{-14}, 3.16×10^{-14}, 4.95×10^{-14} in the six intervals for $x \geq 0$, and 2.29×10^{-13} in the interval for $x < 0$. The same points are chosen for the variable-precision routine sinerfc.m, except for $x < 0$, where we take $n = 0 : 10 : 50$, $x = -.5 : -.5 : -5$, according to the restrictions we had to impose in this case. Here, the values for errmax were found to be 7.93×10^{-34}, 9.08×10^{-35}, 4.57×10^{-34}, 2.50×10^{-33}, 3.28×10^{-32}, 1.60×10^{-31}, for $x \geq 0$, and 6.78×10^{-31} for $x < 0$. As can be seen, the test results are quite satisfactory; they also confirm that the accuracy achieved in many parts of the (n, x)-domain is higher than the one requested, considerably so (by as many as nine digits) in the case of the variable-precision routine sinerfc.m.

To demonstrate how our routines are used, we refer the reader to the code in the file Table7_4.m in the accompanying software package, which recomputes Table 7.4 in Abramowitz and Stegun [1964] of the scaled function $2^n\Gamma(n/2 + 1)\mathrm{i}^n\mathrm{erfc}\, x$, using the routine inerfc.m, for $n = [1 : 6\ 10\ 11]$ and $x = 0 : .1 : 5$, and at the same time checks the results against 20-digit values produced by sinerfc.m, using dig $= 20$. All entries of Abramowitz and Stegun [1964, Table 7.4] were found to be correct to all six digits given, except for occasional end-figure errors of one unit. The maximum error of the 12-digit results generated by Table7_4.m was found to be 8.3×10^{-15}, so that, as expected, the actual accuracy produced by inerfc.m is almost two digits higher than requested.

REFERENCES

Milton Abramowitz and Irene A. Stegun, eds. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards, Applied Mathematics Series 55, Washington, D.C.

D. E. Amos. 1973. Bounds on Iterated Coerror Functions and Their Ratios. *Math. Comp*. 27 (1973), 413–427.

Walter Gautschi. 1961. Recursive Computation of the Repeated Integrals of the Error Function. *Math. Comp.* 15, 227–232. [Also in *Walter* Gautschi. 2014. *Selected Works with Commentaries*, Vol. 1, 260–265, Birkhäuser, New York, NY.]

Walter Gautschi. 1977. Evaluation of the Repeated Integrals of the Coerror Function. *ACM Trans. Math. Software* 3, 240–252.

Walter Gautschi. 2004. *Orthogonal Polynomials: Computation and Approximation*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, NY.

Walter Gautschi. 2012. *Numerical Analysis* (2d ed.). Birkhäuser, New York, NY.

Walter Gautschi. *Orthogonal Polynomials in Matlab: Exercises and Solutions*, Software, Environments, and Tools, SIAM, to appear.

Walter Gautschi. 2014. A Matlab Suite of Programs for Generating Orthogonal Polynomials and Related Quadrature Rules. Purdue University Research Repository. DOI:10.4231/R7959FHP

Received March 2013; revised October 2014; accepted February 2015

42.3. [213] “Monotonicity properties of the zeros of Freud and sub-range Freud polynomials”

[213] “Monotonicity properties of the zeros of Freud and sub-range Freud polynomials”, *Math. Comp.* **86**, 855–864 (2017).

MATHEMATICS OF COMPUTATION
Volume 86, Number 304, March 2017, Pages 855–864
http://dx.doi.org/10.1090/mcom/3181
Article electronically published on June 29, 2016

MONOTONICITY PROPERTIES OF THE ZEROS OF FREUD AND SUB-RANGE FREUD POLYNOMIALS: ANALYTIC AND EMPIRICAL RESULTS

WALTER GAUTSCHI

Abstract. Freud and sub-range Freud polynomials are orthogonal with respect to the weight function $w(t) = |t|^\mu \exp(-|t|^\nu)$, $\mu > -1$, $\nu > 0$, supported on the whole real line $\mathbb{R}$, resp. on strict subintervals thereof. The zeros of these polynomials are studied here as functions of ν and shown, analytically and empirically by computation, to collectively increase or decrease on appropriate intervals of the variable ν.

1. Introduction

Freud polynomials are commonly defined to be orthogonal with respect to the weight function

$$w(t) = w(t;\mu,\nu) = |t|^\mu e^{-|t|^\nu}, \quad \mu > -1,\ \nu > 0, \tag{1.1}$$

supported on the whole real line $\mathbb{R}$. Here we also consider "sub-range" Freud polynomials, which are orthogonal with respect to the same weight function (1.1), but on strict subintervals of $\mathbb{R}$. Specifically, *lower* and *upper symmetric sub-range Freud polynomials* are orthogonal on an interval $[-c, c]$, $0 < c < \infty$, resp. on two disjoint intervals $[-\infty, -c] \cup [c, \infty]$, and become ordinary Freud polynomials when $c \to \infty$, resp. $c \to 0$. Likewise, *lower* and *upper one-sided sub-range Freud polynomials* are orthogonal on $[0, c]$, $0 < c < \infty$, resp. on $[c, \infty]$, and become *half-range Freud polynomials* when $c \to \infty$, resp. $c \to 0$.

Our interest is in the zeros of these polynomials, in particular their monotonicity properties when considered functions of the parameter ν. Analytic results can be derived from a well-known theorem dealing with the dependence of the zeros of orthogonal polynomials on a parameter. While of limited scope, these results contain statements valid for arbitrary parameters $\mu > -1$ and arbitrary degrees n. They are presented and discussed in Section 2. A more comprehensive study of the zeros, at present, is possible only through experimental computation. Results obtained along these lines are described in Section 3. For computational details, however, we must refer to [2].

Received by the editor September 8, 2015.

2010 *Mathematics Subject Classification.* Primary 33C47, 33F05.

Key words and phrases. Freud polynomials, sub-range Freud polynomials, zeros of orthogonal polynomials, Matlab software.

2. Analytic results

The standard result for dealing with zeros of an orthogonal polynomial that depends on a parameter is Markov's theorem (Theorem 6.12.1 of [3]). We apply it here to Freud and sub-range Freud polynomials, where the parameter in question is ν.

2.1. One-sided sub-range Freud polynomials.

Theorem 1. (a) *Let $\nu_0 > 0$ and $0 < c \leq e^{-1/\nu_0}$. Denote by π_n the lower one-sided sub-range Freud polynomial of degree n orthogonal on $[0, c]$ with respect to the weight function*

$$w(t; c, \mu, \nu) = t^{\mu} e^{-t^{\nu}}, \quad t \in [0, c]. \tag{2.1}$$

Then all zeros of π_n are monotonically increasing on $[\nu_0, \infty)$ as functions of ν, for every $\mu > -1$ and $n = 1, 2, 3, \ldots$.

(b) *Let $\nu_0 > 0$ and $c \geq e^{-1/\nu_0}$. Denote by π_n the upper one-sided sub-range Freud polynomial of degree n orthogonal on $[c, \infty]$ with respect to the weight function*

$$w(t; c, \mu, \nu) = t^{\mu} e^{-t^{\nu}}, \quad t \in [c, \infty]. \tag{2.2}$$

Then all zeros of π_n are monotonically decreasing on $(0, \nu_0)$ as functions of ν, for every $\mu > -1$ and $n = 1, 2, 3, \ldots$.

Proof. (a) Let the zeros of π_n, in decreasing order, be

$$\tau_1(\nu) > \tau_2(\nu) > \cdots > \tau_n(\nu) > 0.$$

Then, according to Theorem 6.12.1 of [3], the regularity assumptions of which being all satisfied, the zero $\tau_k(\nu)$, for k fixed, is an increasing [decreasing] function of ν provided

$$f(t) := \frac{\partial w(t; c, \mu, \nu)/\partial \nu}{w(t; c, \mu, \nu)}, \quad 0 < t < c,$$

is an increasing [decreasing] function of t on $(0, c)$. An elementary computation will show that, irrespective of the value of μ,

$$f(t) = -t^{\nu} \ln t.$$

Now

$$f'(t) = -t^{\nu-1}(\nu \ln t + 1),$$

which is positive on the interval $(0, t_0)$ and negative on $t > t_0$, where $t_0 = e^{-1/\nu}$. Since by assumption $c \leq e^{-1/\nu_0}$, we have, for $\nu \geq \nu_0$,

$$0 < t < c \leq e^{-1/\nu_0} \leq e^{-1/\nu} = t_0, \tag{2.3}$$

hence $f'(t) > 0$ on $(0, c)$. By the cited theorem, therefore, $\tau_k(\nu)$ for each k is an increasing function of ν on $[\nu_0, \infty)$, as claimed.

(b) We now have, in place of (2.3), when $\nu < \nu_0$,

$$t \geq c \geq e^{-1/\nu_0} > e^{-1/\nu} = t_0,$$

so that $f'(t)$ is negative on (c, ∞), and the assertion follows as in part (a) from Theorem 6.12.1 of [3]. □

Part (a) of Theorem 1 is of limited scope insofar as it deals only with intervals of orthogonality $[0, c]$, $0 < c < 1$. (In the limit $c = 1$, that is, $\nu_0 = \infty$, it provides no information at all.) It is also of limited interest, since the zeros, in this case, are almost constant functions of ν (see Example 1 below). In this regard, part (b) of the theorem has a wider scope, covering intervals $[c, \infty]$, $0 < c < \infty$, and in the case $c \geq 1$, that is, $\nu_0 = \infty$, provides monotonicity information valid on the whole interval $0 < \nu < \infty$. (In the other limit case $c = 0$, that is, $\nu_0 = 0$, it again is devoid of content.)

We illustrate Theorem 1 by numerical examples. To compute the desired zeros, we first compute the first N recurrence coefficients of the respective orthogonal polynomials from the first $2N$ moments of the weight function, using the classical Chebyshev algorithm in sufficiently high precision (cf. [1, §2.1.7]). The moments are always expressible in terms of the gamma and incomplete gamma functions. Thereafter, the zeros of the orthogonal polynomial of degree n can be obtained (in ordinary working precision) for all $n \leq N$ by well-known eigenvalue/vector techniques (cf. [1, §3.1.1]).

Example 1. The zeros of π_n (of Theorem 1(a)) for $n = 15$ and $n = 30$, when $\nu_0 = 3$, $c = e^{-1/3} = .7165\ldots$, $\mu = 0$, and $3 \leq \nu \leq 10$.

Here, the monotone growth of the zeros is extremely slow. When $n = 15$, the slope is as small as 3.64×10^{-7} and never larger than 8.78×10^{-4}. For $n = 30$, the corresponding numbers are 4.77×10^{-8} and 4.52×10^{-4}. Thus, the zeros are practically constant as functions of ν. Plots of them are shown in Figure 1 for $n = 15$ and $n = 30$. It was determined that monotone growth of all zeros holds even for smaller values of ν, namely for $\nu \geq 1.6926$ when $n = 15$, and for $\nu \geq 1.7064$ when $n = 30$. Thus, Theorem 1(a) is not sharp with regard to the interval of monotonicity.

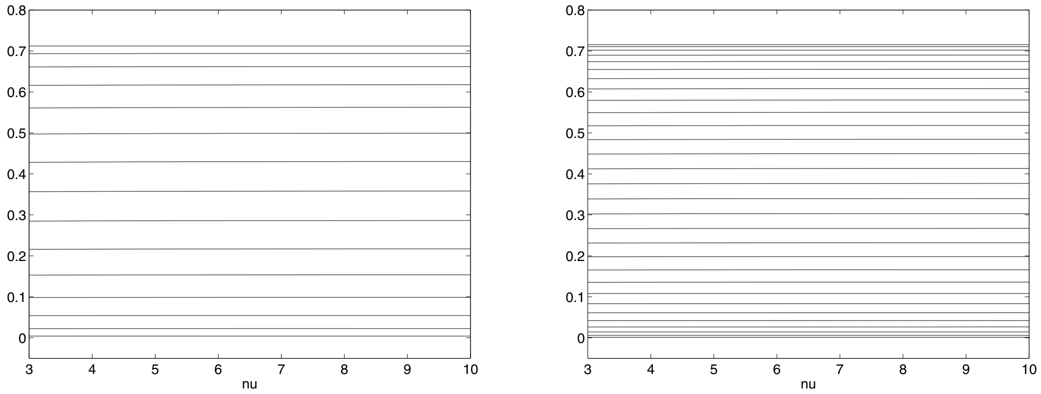


FIGURE 1. The zeros of π_n in the case $\nu_0 = 3$ of Theorem 1(a) for $n = 15$ (on the left) and $n = 30$ (on the right).

Example 2. The zeros of π_n (of Theorem 1(b)) for $n = 15$ and $c = 1$ ($\nu_0 = \infty$), $\mu = 0$, and $0 < \nu \leq 10$.

In Figure 2 the zeros of π_n are shown for $n = 15$, on the left when $0 < \nu < 2$, and on the right when $2 \leq \nu \leq 10$. In the former case, some of the zeros are very large, so that the plot is logarithmic in the y-axis. It is seen, and has been checked, that all zeros, as predicted by the theorem, are monotonically decreasing.

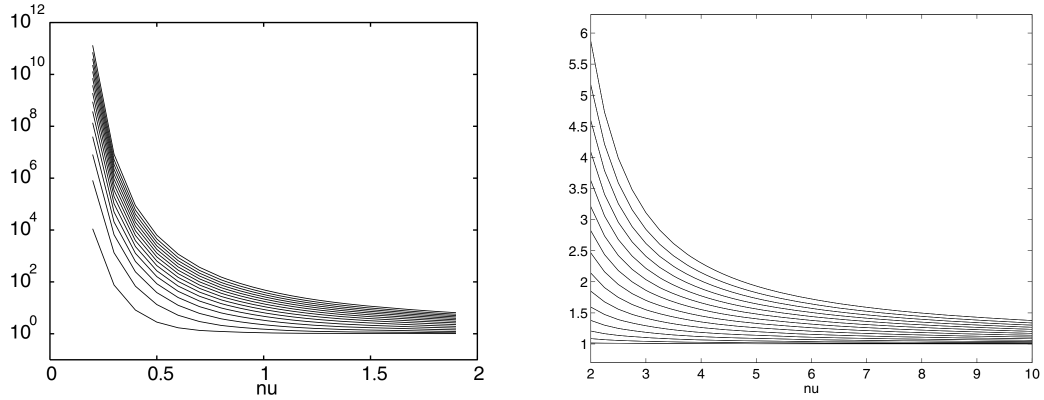


FIGURE 2. The zeros of π_n in the case $c = 1$ of Theorem 1(b) for $n = 15$ on $0 < \nu < 2$ (on the left) and $2 \leq \nu \leq 10$ (on the right).

The graphs look similar for values of c greater than 1 but, of course, lie above the horizontal line at height c. They require much higher precision (250-digit arithmetic when $c = 6$) to produce.

2.2. **Symmetric sub-range Freud polynomials.**

Theorem 2. (a) *Let* $\nu_0 > 0$ *and* $0 < c \leq e^{-1/(2\nu_0)}$. *Denote by* π_n^* *the symmetric sub-range Freud polynomial of degree* n *orthogonal on* $[-c, c]$ *with respect to the weight function*

$$w(t; c, \mu, \nu) = |t|^\mu e^{-|t|^\nu}, \quad t \in [-c, c]. \tag{2.4}$$

Then all positive zeros of π_n^* *are monotonically increasing on* $[2\nu_0, \infty]$ *as functions of* ν, *for every* $\mu > -1$ *and* $n = 2, 3, \dots$.

(b) *Let* $\nu_0 > 0$ *and* $c \geq e^{-1/(2\nu_0)}$. *Denote by* π_n^* *the symmetric sub-range Freud polynomial of degree* n *orthogonal on* $[-\infty, -c] \cup [c, \infty]$ *with respect to the weight function*

$$w(t; c, \mu, \nu) = |t|^\mu e^{-|t|^\nu}, \quad t \in [-\infty, -c] \cup [c, \infty]. \tag{2.5}$$

Then all positive zeros of π_n^* *are monotonically decreasing on* $(0, 2\nu_0)$ *as functions of* ν, *for every* $\mu > -1$ *and* $n = 2, 3, \dots$.

Proof. (a) Since in this case the weight function is even and the interval of orthogonality is symmetric with respect to the origin, the orthogonal polynomial of even degree $2n$ is $\pi^*_{2n}(t) = \pi^+_n(t^2)$ and the one of odd degree is $\pi^*_{2n+1}(t) = t\pi^-_n(t^2)$, where $\pi^\pm_n$ is orthogonal on $[0, c^2]$ relative to the weight function $w^\pm(t) = t^{\mp 1/2}w(t^{1/2})$ (cf. [1, Theorem 1.18]). Thus, the positive zeros of π^*_{2n}, resp. π^*_{2n+1}, are the square root of the zeros of π^+_n, resp. π^-_n. The weight functions for the latter polynomials are $t^{(\mu-1)/2}e^{-t^{\nu/2}}$, resp. $t^{(\mu+1)}e^{-t^{\nu/2}}$. To both of them, part (a) of Theorem 1 can be applied if ν is replaced by $\nu/2$ and c by c^2, showing that the square root of the zeros of $\pi^\pm_n$, hence also the zeros themselves, are monotonically increasing on $[\nu_0, \infty)$ if $c^2 \leq e^{-1/\nu_0}$ and $\nu/2 \geq \nu_0$, that is, if $c \leq e^{-1/(2\nu_0)}$ and $\nu \geq 2\nu_0$.

(b) The polynomials $\pi^\pm_n$ are now orthogonal on $[c^2, \infty]$ with respect to the weight function $w^\pm$. The proof then proceeds as in part (a), but applying part (b) of Theorem 1, again replacing ν by $\nu/2$ and c by c^2. □

As to the scope and sharpness of Theorem 2, here remarks similar to those after Theorem 1 also apply.

3. Empirical results

For simplicity, we concentrate on the case $\mu = 0$, but will indicate what effect other values of μ may have on our results. Also with regard to the range of ν-values, we will generally assume $0 < \nu \leq 10$, which seems to be the interval in which the more interesting monotonicity properties of the zeros play out.

As already noted, there are significant gaps in part (a) of the theorems of Section 2 with regard to intervals of orthogonality covered, and deficiencies in part (b) with regard to sharpness. Here, we fill the gaps and remove the deficiencies by numerical computation.

3.1. **Lower one-sided sub-range Freud polynomials.** The interval of orthogonality $[0, c]$, $0 < c < 1$, is covered by Theorem 1(a) of Section 2.1. It is not a particularly interesting case, since all zeros are essentially constant as functions of ν. The same is still true when $c = 1$ (the limiting case $\nu_0 = \infty$ of Theorem 1(a)), as is shown in Figure 3, depicting the zeros of π_n for $n = 15$ and $n = 30$.

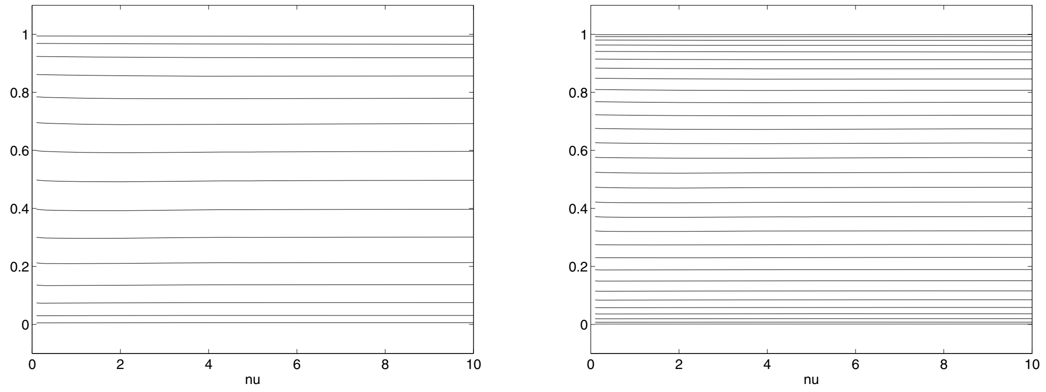


FIGURE 3. The zeros of π_n when $c = 1$ for $n = 15$ (on the left) and $n = 30$ (on the right).

To provide an idea of how the zeros behave when $c > 1$, we look at the case $c = 2$ and show graphs of them in Figure 4 for $n = 1, 7, 15$, and 30. The case $n = 1$ is somewhat special, the zero decreasing to a minimum value and increasing almost imperceptively thereafter. For $n > 1$, the appearance of the graphs resembles that of a waterfall, a gentle one when c is relatively small, and a more precipitous one for larger c; see, e.g., Figure 5, where $c = 6$. Although it may appear that all zeros are collectively decreasing, this is not quite true; there are exceptional intervals early on, when $\nu \leq \nu_1$, where $\nu_1 = 1.392, 1.420, 1.420$ for respectively $n = 7, 15, 30$, and also for ν much larger than 10. But all these exceptions occur in the flat parts of the graphs and are quite minute and not visible to the naked eye.

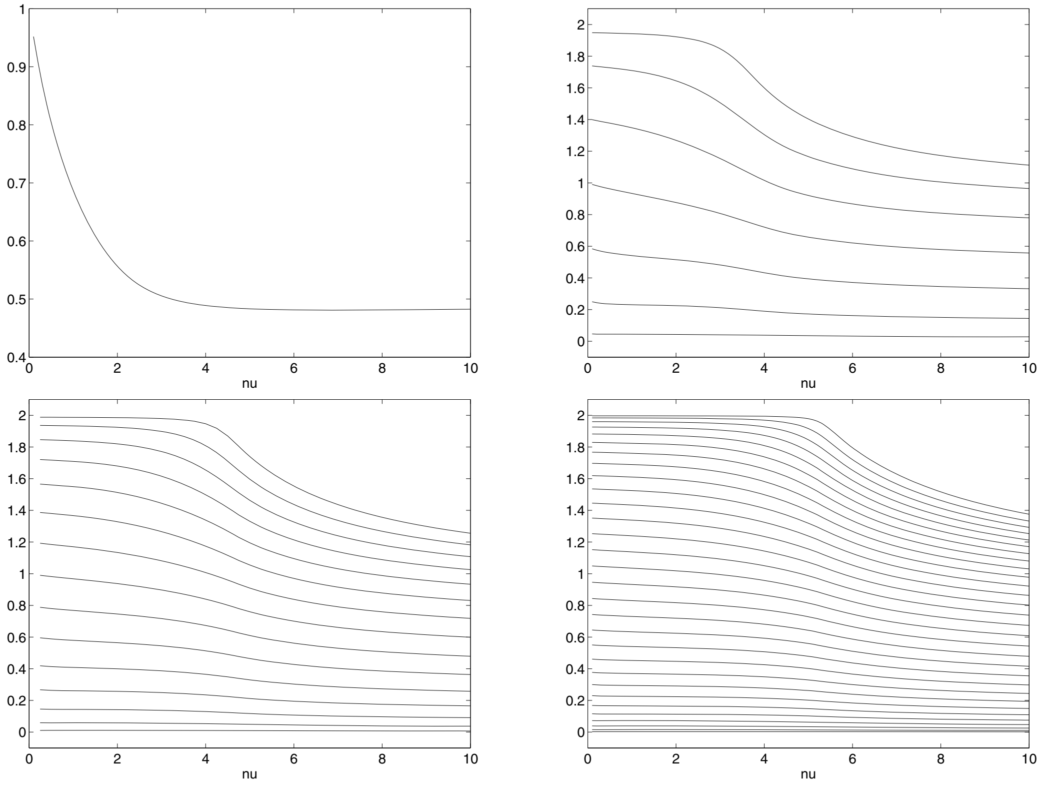


FIGURE 4. The zeros of π_n when $c = 2$ for $n = 1, 7, 15, 30$ (from top left to bottom right).

3.2. **Upper one-sided sub-range Freud polynomials.** Theorem 1(b) covers intervals $[c, \infty]$ with $0 < c < \infty$. It is sharp when $c \geq 1$ ($\nu_0 = \infty$), in which case all zeros decrease monotonically on $0 < \nu < \infty$. Plots of them have been shown in Figure 2, for $0 < \nu \leq 10$.

Here we wish to discuss in detail the sharpness of Theorem 1(b) for selected values of $c < 1$; specifically, for given ν_0 we compute the true interval $(0, \nu_0^*)$ on which all zeros decrease monotonically, for all $n \geq 1$ and all $\mu > -1$.

To begin with, we found evidence, by numerical experimentation, that any interval of monotonicity expands as either n, μ, or both, are increased. The least favorable case, therefore, is $n = 1$ and $\mu > -1$ very close to -1, say, $\mu = -.99999$.

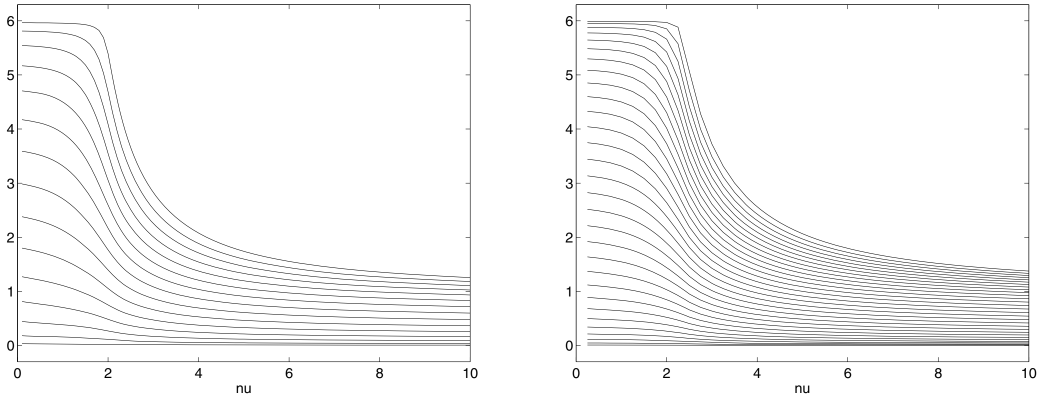


FIGURE 5. The zeros of π_n when $c = 6$ for $n = 15$ (on the left) and $n = 30$ (on the right).

In this case it is relatively straightforward to compute the desired interval $(0, \nu_0^*)$ as a function of ν_0. Results for selected values of ν_0 are shown in Table 1. It can be seen that these intervals are significantly larger than the intervals $(0, \nu_0)$ claimed in Theorem 1(b), but like the latter become smaller with decreasing ν_0, that is, decreasing c.

TABLE 1. Worst-case intervals $(0, \nu_0^*)$ of monotonic decrease of all zeros ($n = 1$ and $\mu \approx -1$).

ν_0	$c = e^{-1/\nu_0}$	ν_0^*	ν_0	$c = e^{-1/\nu_0}$	ν_0^*
6	.8464...	39.336	.6	.1887...	5.1103
4	.7788...	26.594	.5	.1353...	4.5158
2	.6065...	13.877	.4	.0820...	3.9393
1	.3678...	7.5703	.3	.0356...	3.3930
.9	.3291...	6.9481	.2	6.738×10^{-3}	2.8985
.8	.2865...	6.3295	.1	4.540×10^{-5}	2.4861
.7	.2396...	5.7162			

Notice that in the last few entries of Table 1 we are getting very close to the case of half-range Freud polynomials. The fact that the corresponding intervals $(0, \nu_0^*)$ remain finite, and even become a bit smaller, suggests that the zeros of the half-range Freud polynomials are not likely to collectively decrease for arbitrary $n \geq 1$ and $\mu > -1$. We will confirm and quantify this computationally in the next subsection.

3.3. **Half-range Freud polynomials.** Here we explore computationally how ν_0^*, the upper endpoint of the interval $(0, \nu_0^*)$ in which all zeros decrease monotonically, depends on n for $\mu = \mu_- = -.9999999$, about the least favorable value of μ, and also for $\mu = -1/2$, 0, 1/2, and 1. The results are shown in Table 2. Notice the extent of monotonic expansion of the interval $(0, \nu_0^*)$ when n and/or μ are increased. We can see from this table that, for example, all zeros of the half-range Freud polynomial

π_n, for any $\mu > -1$ (more precisely, $\mu \geq \mu_-$), decrease monotonically for all ν in the interval $(0, 10]$ when $n \geq 2$, and for all ν in the interval $(0, 100]$ when $n \geq 6$.

TABLE 2. The intervals $(0, \nu_0^*)$ of monotonic decrease of all zeros of half-range Freud polynomials π_n in dependence of n and μ.

$\mu = \mu_-$		$\mu = -1/2$		$\mu = 0$		$\mu = 1/2$		$\mu = 1$	
n	ν_0^*	n	ν_0^*	n	ν_0^*	n	ν_0^*	n	ν_0^*
1	2.1662	1	4.5574	1	6.8949	1	9.2204	1	11.541
2	11.541	2	15.371	2	19.204	2	23.039	2	26.874
3	26.874	3	32.233	3	37.593	3	42.955	3	48.318
4	48.318	4	55.207	4	62.098	4	68.990	4	75.882
5	75.882	5	84.303	5	92.725	5	101.15		
6	109.57								

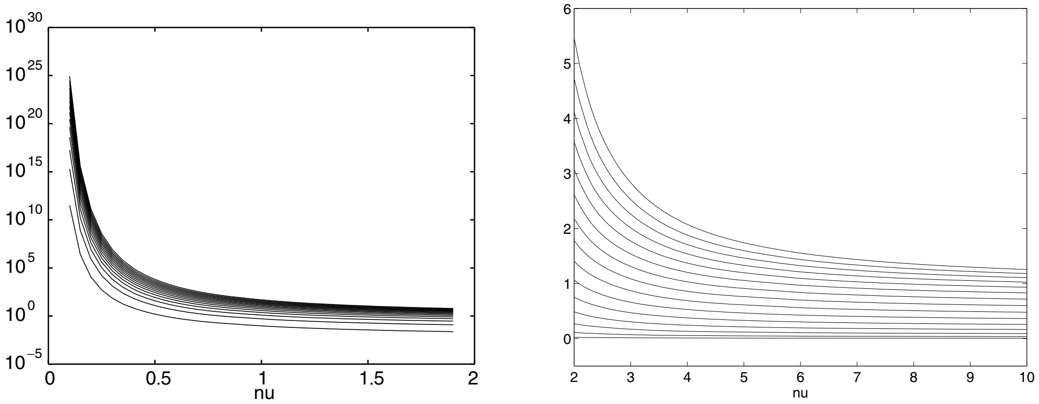


FIGURE 6. The zeros of the half-range Freud polynomial π_n of degree $n = 15$ on the interval $0 < \nu < 2$ (on the left) and $2 \leq \nu \leq 10$ (on the right) when $\mu = 0$.

We show plots of the zeros in Figure 6 for $n = 15$ and $\mu = 0$.

3.4. **Lower symmetric sub-range Freud polynomials.** Symmetric intervals $[-c, c]$ for $0 < c < 1$ are covered by Theorem 2(a). Since all zeros are then practically constant as functions of ν, even in the limit case $c = 1$ (that is, $\nu_0 = \infty$), when the theorem is devoid of content, the case $0 < c \leq 1$ is not of particular interest. For the more interesting cases $c > 1$, we again must rely on computational exploration.

One expects that the behavior of the positive zeros of π_n^* will be similar to that of all zeros of π_n in the asymmetric case. This is indeed borne out by numerical computation. One finds again the waterfall-like descent of all positive zeros, the steepness of the descent being larger the larger the parameter c. It does not seem necessary, therefore, to illustrate this pictorially.

3.5. **Upper symmetric sub-range Freud polynomials.** As in the asymmetric case of Section 3.2, also here in the symmetric case there is a need to sharpen part (b) of Theorem 2, that is, to determine, for given ν_0, the exact interval $(0, \nu_0^*)$ of monotone decrease of all positive zeros in the worst-case scenario of μ very close to -1 and $n = 1$. (Increasing μ and/or n, as in the asymmetric case, yields larger intervals $(0, \nu_0^*)$.) Results analogous to those in Table 1 are shown in Table 3.

TABLE 3. Worst-case intervals $(0, \nu_0^*)$ of monotonic decrease of all positive zeros ($n = 1$ and $\mu \approx -1$).

ν_0	$c = e^{-1/(2\nu_0)}$	ν_0^*	ν_0	$c = e^{-1/(2\nu_0)}$	ν_0^*
6	.9200...	78.671	.6	.4345...	10.221
4	.8824...	53.189	.5	.3678...	9.0315
2	.7788...	27.753	.4	.2865...	7.8786
1	.6065...	15.141	.3	.1888...	6.7860
.9	.5737...	13.896	.2	8.208×10^{-2}	5.7970
.8	.5352...	12.659	.1	6.737×10^{-3}	4.9722
.7	.4895...	11.432			

Here again, the last few entries, pertaining to cases very close to ordinary Freud polynomials, suggest that also the zeros of Freud polynomials are not likely to collectively decrease without some qualifications.

3.6. **Freud polynomials.** Computations analogous to those carried out in Section 3.3 have been made for the case of Freud polynomials. With notation as in Section 3.3, the results are shown in Table 4. Notice again the monotonic behavior of the intervals $(0, \nu_0^*)$ for increasing n and/or μ. It can be seen that all positive zeros of the Freud polynomial π_n^* decrease monotonically on the interval $(0, 10]$ if $n \geq 3$, and on the interval $(0, 100]$ if $n \geq 9$.

TABLE 4. The intervals $(0, \nu_0^*)$ of monotonic decrease of all positive zeros of Freud polynomials π_n^* in dependence of n and μ.

$\mu = \mu_-$		$\mu = -1/2$		$\mu = 0$		$\mu = 1/2$		$\mu = 1$	
n	ν_0^*	n	ν_0^*	n	ν_0^*	n	ν_0^*	n	ν_0^*
2	4.3325	2	6.7519	2	9.1147	2	11.457	2	13.790
3	13.790	3	16.117	3	18.441	3	20.762	3	23.082
4	23.082	4	26.910	4	30.741	4	34.574	4	38.408
5	38.408	5	42.242	5	46.077	5	49.913	5	53.749
6	53.749	6	59.106	6	64.465	6	69.825	6	75.186
7	75.186	7	80.548	7	85.910	7	91.273	7	96.636
8	96.636	8	103.52						
9	124.20								

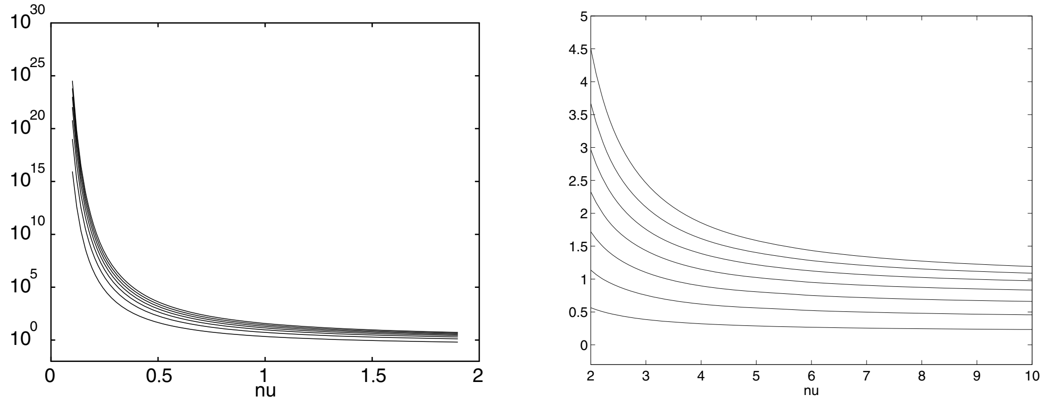


FIGURE 7. The positive zeros of the Freud polynomial π_n^* for $n = 15$ on $0 < \nu < 2$ (on the left) and $2 \leq \nu \leq 10$ (on the right).

Plots of the zeros for $n = 15$ are shown in Figure 7. In the process of producing these plots it was checked that all zeros indeed decrease on $(0, 10]$.

REFERENCES

[1] W. Gautschi, *Orthogonal Polynomials: Computation and Approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2004. MR2061539

[2] W. Gautschi, *Orthogonal polynomials in Matlab: Exercises and solutions*, Software, Environments, Tools, SIAM, Philadelphia, PA, 2016.

[3] G. Szegő, *Orthogonal Polynomials*, 4th ed., Colloquium Publications, vol. XXIII, American Mathematical Society, Providence, R.I., 1975. MR0372517

DEPARTMENT OF COMPUTER SCIENCE, PURDUE UNIVERSITY, WEST LAFAYETTE, INDIANA 47907-2066

E-mail address: wgautschi@purdue.edu

42.4. [216] “On the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials”

[216] “On the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials”, in *Frontiers in orthogonal polynomials and q-series* (M. Zuhair Nashed and Xin Li, eds.), 251–266, World Sci. Publ., Hackensack, NJ, 2018.

Chapter 13

On the Ismail–Letessier–Askey Monotonicity Conjecture for Zeros of Ultraspherical Polynomials

Walter Gautschi

Department of Computer Science, Purdue University
West Lafayette, IN 47907-2107, USA
wgautschi@purdue.edu

Dedicated to Mourad E. H. Ismail on his 70th birthday

A number of conjectured monotonicity properties for zeros of ultraspherical polynomials are reviewed, leading up to the Ismail–Letessier–Askey (ILA) conjecture of the title, which has been proved in 1999 by Elbert and Siafarikas. It is shown that two of the earlier conjectures are consequences of the ILA conjecture. Computational support is provided for strengthening several of these conjectures, including the ILA conjecture, from monotonicity to complete monotonicity.

Keywords: Ultraspherical polynomials; zeros; monotonicity.

Mathematics Subject Classification 2010: 33C45, 65D20

1. Introduction

Inequalities and monotonicity properties for zeros of orthogonal polynomials depending on a parameter is a classical subject; see, e.g., [24, Chapter 6]. The last three or four decades, however, have seen a considerable increase of activity in this area. Several approaches have been pursued concurrently. One is via differential equations, specifically the Sturm comparison theorem. Surveys and tutorials on this are given by L. Lorch, A. Laforgia, and

M. E. Muldoon in, respectively, [22, 20], and [21], and further applications to generalized Laguerre polynomials in [3], to Jacobi polynomials in [5, Section 3], and to ultraspherical polynomials in [1, 2]. Another approach uses Markov's theorem [24, Section 6.12], which has recently been applied in [11] in connection with Freud and sub-range Freud polynomials, and a slight extension thereof in [15, Section 3] with applications to Laguerre, Jacobi, and Meixner polynomials. An approach originating in physics, and promoted primarily by Mourad Ismail, makes use of the Hellmann–Feynman theorem, which looks at the zeros of orthogonal polynomials as eigenvalues of an operator depending on a parameter and states formulas for the derivatives of the eigenvalues with respect to that parameter. This is surveyed in the paper [19] and applied there, and in a number of other papers [14, 17, 18], to zeros of a variety of orthogonal polynomials, including birth and death process polynomials.

Finally, there is an entirely empirical approach based on numerical computation, which is an important vehicle to test conjectured inequalities and monotonicity properties, thereby providing stimuli for further analytical work. Examples of this are a series of papers, [6–8, 13] on the zeros of Jacobi polynomials and also the paper [11] already cited on the zeros of Freud polynomials. This is the approach used here in Section 5 to test conjectured higher monotonicity properties for zeros of ultraspherical polynomials. Ordinary monotonicity properties are surveyed in Section 2. Software tools used in this paper are briefly described in Section 4.

2. Ultraspherical Polynomials and the ILA Conjecture

Ultraspherical polynomials arise in the solution of Laplace's equation in high-dimensional spaces, when written in terms of hyperspherical coordinates and solved by the method of separation of variables. The polynomials, also named after the Austrian mathematician Leopold Gegenbauer (1849–1903), who introduced them in his doctoral thesis of 1875 and studied them in subsequent papers, are commonly denoted by $C_n^{(\lambda)}$ [23, equation 18.7.1] (but see also [24, Section 4.7], where the notation $P_n^{(\lambda)}$ is used), n being the degree and $\lambda > -1/2$ a parameter. They are orthogonal on the interval $(-1, 1)$ relative to the weight function $w_\lambda(t) = (1 - t^2)^{\lambda - 1/2}$, that is

$$\int_{-1}^{1} C_k^{(\lambda)}(t) C_\ell^{(\lambda)}(t) w_\lambda(t) \mathrm{d}t = 0 \quad \text{if } k \neq \ell. \tag{2.1}$$

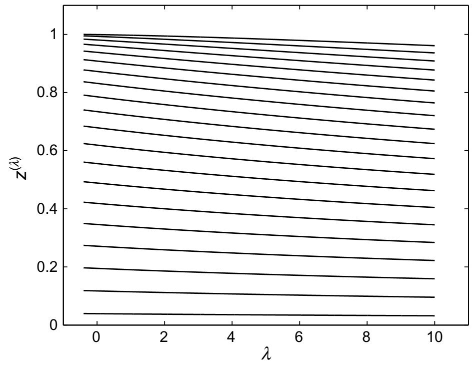


Fig. 1. The positive zeros of $C_n^{(\lambda)}$ for $n = 40$ and $-1/2 < \lambda \leq 10$.

Important special cases include Chebyshev polynomials of the first and second kind, corresponding to $\lambda = 0$ and $\lambda = 1$, and Legendre polynomials, corresponding to $\lambda = 1/2$.

Since the weight function w_λ is even, the polynomial $C_n^{(\lambda)}$ is even or odd, depending on whether n is even or odd, and the zeros therefore are symmetric, or antisymmetric, with respect to the origin. To study them, it thus suffices to look at the positive zeros $z_{n,k}^{(\lambda)}$ of $C_n^{(\lambda)}$. In Fig. 1 we show plots of them as functions of λ for $-0.49 \leq \lambda \leq 10$ and $n = 40$. It appears from the graphs, and has been verified computationally, that the zeros are all monotonically decreasing. This was already known to Stieltjes (see [24, p. 121]) and a proof using Markov's theorem in combination with quadratic transformation of hypergeometric functions is mentioned in [15, p. 188]. Laforgia [20] conjectured that the zeros multiplied by λ become monotonically increasing for all $\lambda > -1/2$ and all $n \geq 2$, and proved this for $0 < \lambda < 1$ using one of Szegö's formulations of the Sturm comparison theorem [24, Theorem 1.82.1]. We verified this computationally for $-1/2 < \lambda \leq 10$ (in steps of of 1/100) and for all n with $2 \leq n \leq 40$. Graphs for $n = 40$ are shown on the left of Fig. 2. Ahmed, Muldoon, and Spigler [1], also using Sturmian methods, proved that the zeros multiplied by $[\lambda + (2n^2 + 1)/(4n + 2)]^{1/2}$ are increasing when $-1/2 < \lambda \leq 3/2$, $n \geq 2$, which we verified computationally for the same values of λ and n as above and show graphs for $n = 40$ on the left of Fig. 3. This actually implies the validity of Laforgia's conjecture for $-1/2 < \lambda \leq 3/2$ by straightforward differentiation of $\lambda^{-1}[\lambda+(2n^2+1)/(4n+2)]^{1/2}[\lambda z_{n,k}^{(\lambda)}]$. Ismail and Letessier [16] conjectured

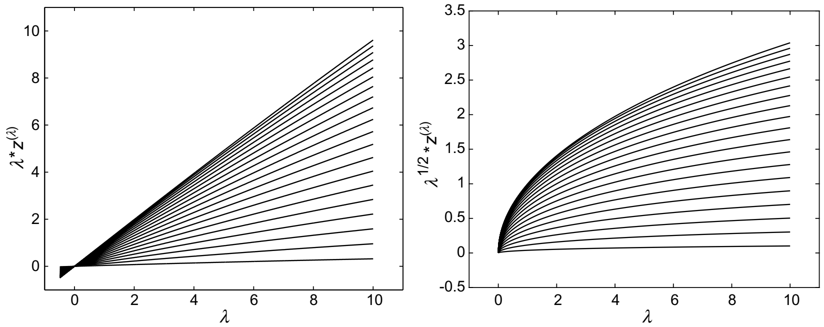


Fig. 2. The positive zeros of $C_n^{(\lambda)}$ multiplied by λ (left) for $n = 40$, $-1/2 < \lambda \leq 10$, and multiplied by $\sqrt{\lambda}$ (right) for $n = 40$, $0 \leq \lambda \leq 10$.

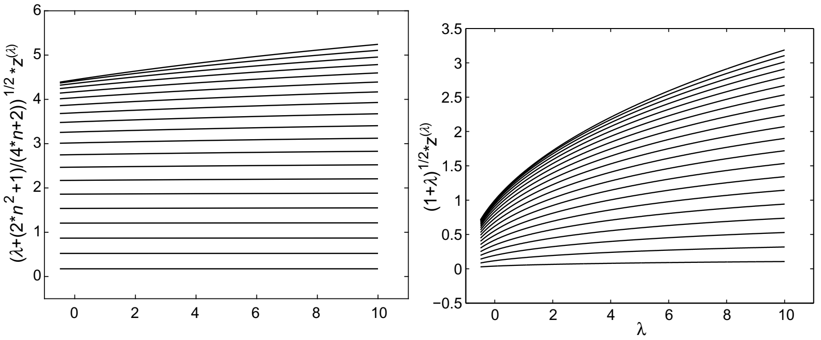


Fig. 3. The positive zeros of $C_n^{(\lambda)}$ multiplied by $[\lambda + (2n^2+1)/(4n+2)]^{1/2}$ (left) and multiplied by $(1+\lambda)^{1/2}$ (right) for $n = 40$, $-1/2 < \lambda \leq 10$.

that the zeros multiplied by $\sqrt{\lambda}$ are monotonically increasing for $\lambda \geq 0$ and proved this for the largest zero. We verified the conjecture for all $2 \leq n \leq 40$ and all positive zeros, for the same values of λ as before, but with $\lambda \geq 0$. Respective graphs are shown on the right of Fig. 2 for $n = 40$. Askey suggested that monotonic growth may also hold when the zeros are multiplied by $(1+\lambda)^{1/2}$, which has become known as the ILA conjecture, named after Ismail, Letessier, and Askey. We verified this conjecture for all $3 \leq n \leq 40$ (when $n = 2$ the product in question is constant equal to $1/\sqrt{2}$) and for $-1/2 < \lambda \leq 10$ in steps of $1/100$; see the graphs on the right of Fig. 3. The conjecture, in fact, has been proven by Elbert and Siafarikas [4] by showing that the Ahmed–Muldoon–Spigler result holds for all $\lambda > -1/2$, which in turn, by straightforward differentiation, implies the ILA conjecture.

3. Some Implications of the ILA Conjecture

With $z(\lambda)$ denoting any positive zero of $C_n^{(\lambda)}$, writing

$$f(\lambda) = \left(\frac{1+\lambda}{\lambda}\right)^{1/2}[\lambda^{1/2}z(\lambda)] = (1+\lambda^{-1})^{1/2}[\lambda^{1/2}z(\lambda)]$$

and using the ILA conjecture $f'(\lambda) > 0$ (proved by Elbert and Siafarikas), differentiation of the right-hand side yields

$$(1+\lambda^{-1})^{1/2}[\lambda^{1/2}z(\lambda)]' = f'(\lambda) + \tfrac{1}{2}\,(1+\lambda^{-1})^{-1/2}\lambda^{-3/2}z(\lambda) > 0 \quad \text{for } \lambda > 0,$$

that is, the validity of the Ismail–Letessier conjecture. Likewise, writing

$$f(\lambda) = \left(\frac{1+\lambda}{\lambda^2}\right)^{1/2}[\lambda z(\lambda)]$$

and differentiating yields

$$\left(\frac{1+\lambda}{\lambda^2}\right)^{1/2}[\lambda z(\lambda)]' = f'(\lambda) + \tfrac{1}{2}\,(1+\lambda)^{-1/2}(1+2/\lambda)z(\lambda) > 0 \quad \text{if } \lambda > 0,$$

proving Laforgia's conjecture for $\lambda > 0$. For $\lambda \le 0$, the conjecture is trivially true since $[\lambda z(\lambda)]' = z(\lambda) + \lambda z'(\lambda) > 0$ because of $z'(\lambda) < 0$.

4. Matlab Software for Orthogonal Polynomials

All computational work in this chapter was done by using the Matlab software package *Orthogonal Polynomials and Quadrature* (`OPQ`), located at

`http://dx.doi.org/10.4231/R7959FHP`,

and its symbolic (variable-precision) counterpart (`SOPQ`), located at

`http://dx.doi.org/10.4231/R7ZG6Q6T`.

We describe and illustrate here the routines most relevant for our purposes. Many other applications can be found in [10].

A system $\{\pi_k\}_{k=0}^{\infty}$ of monic polynomials $\pi_k(t) = t^k + \cdots$ is called *orthogonal* on the interval (a, b), $-\infty \le a < b \le \infty$, with respect to a positive weight function w, if (cf. also (2.1))

$$\int_a^b \pi_k(t)\pi_\ell(t)w(t)\mathrm{d}t = 0 \quad \text{when } k \neq \ell. \tag{4.1}$$

It is known that every such system satisfies a three-term recurrence relation

$$\begin{gathered} \pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \ldots, \\ \pi_{-1}(t) = 0, \quad \pi_0(t) = 1, \end{gathered} \tag{4.2}$$

with real $\alpha_k = \alpha_k(w)$ and positive $\beta_k = \beta_k(w)$ that depend on the weight function w. Thus, to obtain the polynomial π_n of degree n from (4.2) requires knowledge of the first n of these coefficients, $\alpha_0, \alpha_1, \ldots \alpha_{n-1}$ and $\beta_0, \beta_1, \ldots, \beta_{n-1}$, where β_0 can be arbitrary, but is conveniently defined to be $\beta_0 = \int_a^b w(t)\mathrm{d}t$. These are provided by the `OPQ` routine with syntax

$$\texttt{ab} = \texttt{r\_}\mathit{name}(\texttt{n}, \ldots), \tag{4.3}$$

where *name* specifies the name of the orthogonal polynomial, `n` the degree, and `ab` is the $n \times 2$ array of recurrence coefficients

$$\texttt{ab} = \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \\ \vdots & \vdots \\ \alpha_{n-1} & \beta_{n-1} \end{bmatrix}.$$

The dots on the right of (4.3) stand for possible parameters defining the orthogonal polynomial.

The polynomials that interest us here are exclusively *Jacobi polynomials*, which are orthogonal on the interval $(-1, 1)$ relative to the weight function $w(t) = (1 - t)^\alpha(1 + t)^\beta$, $\alpha > -1$, $\beta > -1$. More specifically, we are interested in *ultraspherical* polynomials, which are Jacobi polynomials with parameters $\alpha = \beta = \lambda - 1/2$, where $\lambda > -1/2$. Their first n recurrence coefficients are produced by the `OPQ` routine

```
ab=r_jacobi(n,lambda-1/2)
```

in Matlab double precision, and by the `SOPQ` routine

```
sab=sr_jacobi(dig,n,lambda-1/2)
```

in `dig`-digit precision. Thus, `ab=r_jacobi(5,1)` and `sab=sr_jacobi (42,5,1)`, which are the same as `ab=r_jacobi(5,1,1)` and `sab=sr_jacobi (42,5,1,1)`, produce

```
ab =

      0     1.333333333333333e+00
      0     2.000000000000000e-01
      0     2.285714285714286e-01
      0     2.380952380952381e-01
      0     2.424242424242424e-01
```

and

```
sab =

[ 0,  1.33333333333333333333333333333333333333333]
[ 0,                                          0.2]
[ 0, 0.228571428571428571428571428571428571428571]
[ 0, 0.238095238095238095238095238095238095238095]
[ 0, 0.242424242424242424242424242424242424242424]
```

that is, the first five recurrence coefficients for the ultraspherical polynomial $C_5^{(3/2)}$ in double and 42-digit precision.

There is no `OPQ` routine that specifically generates the zeros of orthogonal polynomials, but they can be generated by the Gauss quadrature routine

$$\mathtt{xw} = \mathtt{gauss}(\mathtt{n}, \mathtt{ab}), \tag{4.4}$$

which obtains in the first column of the n × 2 array `xw` the n nodes (in increasing order), and in the second column the corresponding weights, of the `n`-point Gaussian quadrature rule associated with the weight function w specified by the $n \times 2$ array `ab` of its first n recurrence coefficients. The n Gaussian nodes are nothing but the n zeros of the orthogonal polynomial of degree n relative to the weight function w. Thus, the n zeros of $C_n^{(\lambda)}$ in increasing order are obtained by the following Matlab routine:

```
function z=zeros_us(n,lambda)
%ZEROS_US Zeros of ultraspherical polynomials.
%  Z=ZEROS_US(N,LAMBDA) generates the N zeros in
%  increasing order of the Nth-degree ultraspherical
%  polynomial C_N^{(LAMBDA)}.
```

```
ab=r_jacobi(n,lambda-1/2); xw=gauss(n,ab);
z=xw(:,1);
```

The Matlab script

```
%PLOT_ZEROS_US

n=40;
n0=floor((n+1)/2)+1;
for k=n0:n
  si=(-.49:.01:10)';
%  si=(0:.01:10)';
  s=size(si); y=zeros(s);
  i=0;
  for lam=-.49:.01:10
%  for lam=0:.01:10
    i=i+1;
    z=zeros_us(n,lam);
    y(i,k)=z(k);
%    y(i,k)=lam*z(k);
%    y(i,k)=sqrt(lam)*z(k);
%    y(i,k)=(2*n^2+1+2*lam*(2*n+1))^(1/2)*z(k);
%    y(i,k)=(1+lam)^(1/2)*z(k);
  end
  for i=2:s(1)
    if y(i,k)>=y(i-1,k)
  %    if y(i,k)<=y(i-1,k)
      [n i k]
      [y(i,k) y(i-1,k)]
      error('wrong sign')
    end
  end
  plot(si,y(:,k),'LineWidth',1.5);set(gca,'FontSize',14)
  hold on
  axis([-1 11 0 1.1])
%  axis([-1 11 -1 11])
%  axis([-1 11 -.5 3.5])
%  axis([-1 11 -5 70])
%  axis([-1 11 -.5 3.5])
  xlabel('\lambda'); ylabel('z(\lambda)');
%  xlabel('\lambda'); ylabel('\lambda*z(\lambda)')
%  xlabel('\lambda'); ylabel('\lambda^{1/2}*z(\lambda)')
%  xlabel('\lambda');
%  ylabel('(2*n^2+1+2*\lambda*(2*n+1))^{1/2}*z(\lambda)')
%  xlabel('\lambda'); ylabel('(1+\lambda)^{1/2}*z(\lambda)')
end
```

not only produces all plots in the figures of Section 2 but also checks the validity of the relevant monotonicity properties by selecting the appropriate commands from those commented out or currently active.

5. Conjectured Higher Monotonicity Properties for Ultraspherical Polynomials

Let again $z(\lambda)$ be any positive zero of the nth-degree ultraspherical polynomial $C_n^{(\lambda)}$, $\lambda > -1/2$. (For simplicity of notation, we do not show the dependence of z on n.) The ILA conjecture, proved by Elbert and Siafarikas [4], states that

$$f(\lambda) = (1+\lambda)^{1/2} z(\lambda), \quad \lambda > -1/2, \tag{5.1}$$

is an increasing function of λ for $\lambda > -1/2$ and $n \geq 3$, even though $z(\lambda)$ is decreasing. Likewise, the Ahmed–Muldoon–Spigler result, as extended and proved by Elbert and Siafarikas (see end of Section 2), states that

$$g(\lambda) = \left(\lambda + \frac{2n^2+1}{4n+2}\right)^{1/2} z(\lambda), \quad \lambda > -1/2, \tag{5.2}$$

is an increasing function of λ for $\lambda > -1/2$ and $n \geq 2$. The Ismail–Letessier conjecture, as proved in Section 3, states that

$$h(\lambda) = \sqrt{\lambda}\, z(\lambda), \quad \lambda \geq 0, \tag{5.3}$$

is monotonically increasing for $\lambda \geq 0$ and $n \geq 2$, while Laforgia's conjecture, also proved in Section 3, states that

$$k(\lambda) = \lambda z(\lambda), \quad \lambda > -1/2, \tag{5.4}$$

is monotonically increasing for $\lambda > -1/2$ and $n \geq 2$.

The graphs of f and h on the right of Figs. 3 and 2, showing not only monotonicity, but also concavity of all positive zeros, suggest that these zeros satisfy also higher monotonicity properties. The same may be true for the function g and some of the larger zeros in the graph on the left of Fig. 3, and perhaps even for the function $k(\lambda)$.

5.1. The function $f(\lambda)$

Our conjecture for the function f is as follows.

Conjecture 5.1. *For all $n \geq 3$ the first derivative f' of the function f in* (5.1) *is completely monotone on* $(-1/2, \infty)$, *i.e.*,

$$(-1)^m f^{(m+1)}(\lambda) > 0, \quad m = 0, 1, 2, \ldots, \quad \lambda > -1/2. \tag{5.5}$$

The evidence we provide for this and the subsequent conjectures is for $n \leq 15$. It is based on divided differences. Thus, let $s(h) = \{s_j(h)\}_{j=1}^{\infty}$, $h > 0$, be the infinite sequence with

$$s_j(h) = -1/2 + jh, \quad j = 1, 2, 3, \ldots,$$

and $d_{j,m}(h;f) = [s_j(h), s_{j+1}(h), \dots, s_{j+m}(h)]f$ the mth-order divided difference of f relative to $m+1$ consecutive members of $s(h)$ starting with $s_j(h)$. Our objective is to verify computationally that for given integers $J > 1$ and $M > 0$

$$(-1)^m d_{j,m+1}(h;f) > 0, \quad j = 1, 2, \dots, J; \quad m = 0, 1, \dots, M. \tag{5.6}$$

Since the $(m+1)$th divided difference of f is equal to a positive constant times the $(m+1)$th derivative of f evaluated at some intermediate point (see, e.g., [9, equation (2.117)]), it is plausible, especially if h is small, that (5.6) implies (5.5), at least for the λ-range and the m-values indicated by (5.6), but very likely for all $\lambda > -1/2$ and all $m \geq 0$.

In principle, the computational validation of (5.6) is straightforward, but requires caution when m is large, because of numerical instability. By virtue of (see, e.g., [9, Chapter 2, Exercise 54])

$$\Delta^m f_j = m! h^m d_{j,m}(h;f),$$

where $\Delta^m f_j$ is the mth difference of f at s_j, the computation of divided differences and differences is equally stable. Because of cancellation errors, however, computing differences becomes increasingly unstable as m increases, and may even yield the wrong sign. Therefore, high-precision arithmetic is imperative, though time-consuming, when m is large.

We select $h = 0.02$, $M = 14$, and $J = 350$, which are constants that remain fixed for the remainder of this section. The choice of these constants covers divided differences of orders 1–15 and a λ-interval $-1/2 < \lambda \leq 13/2$.

In Matlab double-precision arithmetic, we were able to confirm (5.6) for all $3 \leq n \leq 15$ and $0 \leq m \leq 4$ in a matter of a few seconds runtime. The differences $\Delta^m f_j$ involved were never smaller in absolute value than 1.894×10^{-13}. For the values (n, m) not covered by these computations, that is, for $3 \leq n \leq 15$ and $5 \leq m \leq 14$, it took 36-digit arithmetic in symbolic Matlab and some 70 h of runtime to confirm (5.6). The absolutely smallest difference $\Delta^m f_j$ observed was 7.774×10^{-30}.

The computations described are implemented in the Matlab script[a] `conj_geg.m` for double precision arithmetic, and in the script `sconj_geg.m` for variable-precision arithmetic using the Matlab symbolic toolbox. The former script also uses the `OPQ` routines `r_jacobi.m` and `gauss.m`, computing respectively the recurrence coefficients of the relevant ultraspherical polynomials and the related Gaussian quadrature rules (hence, in particular, the positive zeros of the ultraspherical polynomials), and the latter

[a] All Matlab scripts referenced in this chapter are collected in [12].

script uses the corresponding variable-precision routines `sr_jacobi.m` and `sgauss.m`; cf. Section 4. An additional pair of routines is used, `dd.m` and `sdd.m`, generating (from the bottom up) the appropriate divided differences in double or variable precision; see, e.g., Machine Assignment 7(a) in [9, Chapter 2] and its solution on p. 153. The script `conj_geg.m` and the routine `dd.m` are shown below. Their variable-precision versions are straightforward translations from double-precision arithmetic to variable-precision arithmetic.

```
%CONJ_GEG

f0='%12.4e\n';
h=1/50; J=350; M=15;
for n=3:15
 fprintf('n=%4.0f\n',n)
 nh=floor(n/2); z=zeros(J,1);
 for k=1:nh
  for j=1:J
   b=-1+j*h;
%   b=-1/2-h+j*h;
   ab=r_jacobi(n,b);
   xw=gauss(n,ab);
   z(j)=sqrt(b+3/2)*xw(n+1-k,1);
%   z(j)=sqrt(b+1/2)*xw(n+1-k,1);
%   z(j)=-xw(n+1-k,1);
  end
  for m=1:M
   dmin=10^20; x=zeros(m+1,1); zm=zeros(m+1,1);
   for jm=1:J-m
    for mu=1:m+1
     x(mu)=-1+(jm+mu-1)*h;
     zm(mu)=z(jm+mu-1);
    end
    dm=dd(m,x,zm);
    if sign((-1)^m*dm)>0
     disp('wrong sign')
     fprintf(f0,h^m*factorial(m)*dm)
     error('quit')
    end
```

```
        if abs(dm)<dmin, dmin=abs(dm); end
      end
      fprintf(f0,h^m*factorial(m)*dmin)
    end
    fprintf('\n')
  end
end


function y=dd(n,x,f)
%DD Divided difference.
%   Y=DD(N,X,F) evaluates the Nth divided difference
%   Y of F, where X=[X_0,X_1,...,X_N]^T,
%   F=[F(X_0),F(X_1),...,F(X_N)]^T.

d=zeros(n+1,1);
d=f;
if n==0
  y=d(1);
  return
end
for j=1:n
  for i=n:-1:j
    d(i+1)=(d(i+1)-d(i))/(x(i+1)-x(i+1-j));
  end
end
y=d(n+1);
```

The script `conj_geg.m`, as listed, produces an error message "`wrong sign`" already for $n = 3$ and $m = 5$ and displays the delinquent difference. The latter is close in absolute value to the machine precision, in fact equal to 3.684×10^{-15}, and therefore unreliable. To produce the double-precision result mentioned above, the m-loop in the script has to be run only up to $m = 5$. The 36-digit confirmation of (5.6) is produced by the script `sconj_geg.m`.

5.2. The function $h(\lambda)$

The conjecture for the function h is as follows.

Conjecture 5.2. *For all $n \geq 2$ the first derivative h' of the function h in* (5.3) *is completely monotone on* $[0, \infty)$, *i.e.*,

$$(-1)^m h^{(m+1)}(\lambda) > 0, \quad m = 0, 1, 2, \ldots, \ \lambda \geq 0. \tag{5.7}$$

The numerical validation of Conjecture 5.2 is done by the same scripts `conj_geg.m` and `sconj_geg.m`, slightly modified to deal with the function h in place of f and the interval $\lambda \geq 0$ instead of $\lambda > -1/2$. Here, the double-precision routine does the job quickly for $2 \leq n \leq 15$ and $0 \leq m \leq 6$, and the 36-digit routine for the remaining cases in about 79 h runtime.

With regard to complete monotonicity, the functions g and k in (5.2), (5.4) do not quite measure up to the functions f and h in (5.1) and (5.3). It is true that for the single positive zero $z(\lambda)$, when $n = 2$ or $n = 3$, the functions g' and k' are indeed completely monotone, but for $n = 4$, computation suggests that for g this is true only for the larger of the two positive zeros and not for the other, and for the function k only for the smaller of the two zeros and not for the other.

5.3. The function $g(\lambda)$

Conjecture 5.3. *For all $n \geq 4$, the first derivative g' of the function g in* (5.2) *is completely monotone when $z(\lambda)$ is the largest positive zero of $C_n^{(\lambda)}$, but not so otherwise.*

This was verified in the same manner as Conjectures 5.1 and 5.2, with the same scripts `conj_geg.m`, `sconj_geg.m`, suitably modified. It took another 15 h of runtime.

For the positive zeros $z(\lambda)$ of g other than the largest, additional computations (in 36-digit arithmetic) suggest an "incomplete monotonicity" property, i.e., the existence of a positive integer m_0 such that

$$(-1)^m g^{(m+1)}(\lambda) > 0 \quad \text{for } 0 \leq m \leq m_0, \ \lambda > -1/2, \tag{5.8}$$

with the opposite inequality holding when $m = m_0 + 1$. If we denote the positive zeros of $C_n^{(\lambda)}$ in decreasing order by $z_{n,k}^{(\lambda)}$, $k = 1, 2, \ldots, \lfloor \text{n}/2 \rfloor$, then it is found for $4 \leq n \leq 30$ that

$$m_0 = 1 \text{ holds}$$

$$\text{for } k = 2 \text{ when } 17 \leq n \leq 30$$

$$m_0 = 2 \text{ holds}$$

$$\text{for } k = 2 \text{ when } 4 \leq n \leq 16$$

$$\text{for } k = 3 \text{ when } 6 \leq n \leq 30$$
$$\text{for } k = 4 \text{ when } 8 \leq n \leq 30$$
$$\text{for } k = 5 \text{ when } 17 \leq n \leq 30$$
$$\text{for } k = 6 \text{ when } 24 \leq n \leq 30$$
$$m_0 = 3 \text{ holds}$$
$$\text{for } k = 5 \text{ when } 10 \leq n \leq 16$$
$$\text{for } k = 6 \text{ when } 12 \leq n \leq 23$$
$$\text{for } 7 \leq k \leq 15 \text{ when } 2k \leq n \leq 30$$

5.4. The function $k(\lambda)$

It took another 60 h of runtime to verify the following conjecture for the function k.

Conjecture 5.4. *For* $4 \leq n \leq 10$, *the first derivative of the function* k *in* (5.4) *is incompletely monotone* (in the sense of (5.8), with $m_0 = 3, 2, 1$ for respectively $n = 4$, $5 \leq n \leq 6$, $n \geq 7$) *when* $z(\lambda)$ *is the largest positive zero of* $C_n^{(\lambda)}$, *and completely monotone otherwise. For* $11 \leq n \leq 15$, *both the largest and second-largest zero is incompletely monotone* (the former with $m_0 = 1$, and the latter with $m_0 = 8, 5, 3, 2$ for respectively $n = 11$, $n = 12$, $13 \leq n \leq 14$, $n = 15$), *while all the other zeros are completely monotone.*

This pattern likely continues for $n > 15$, with the first few positive zeros (in decreasing order) being incompletely monotone, and the remaining ones completely monotone.

5.5. The zeros $z(\lambda)$

It seems natural to ask whether higher monotonicity properties may hold also for the zeros themselves. After all, when $n = 2$, we have $-z'(\lambda) = (1/2\sqrt{2})(\lambda + 1)^{-3/2}$, which is clearly completely monotone, and the same is true for $n = 3$, where $-z'(\lambda) = (1/2\sqrt{2/3})(\lambda + 2)^{-3/2}$, suggesting that $-z'(\lambda)$ might be completely monotone for all $n \geq 2$,

$$(-1)^{m+1} z^{(m+1)}(\lambda) > 0, \quad m = 0, 1, 2, \ldots, \lambda > -1/2.$$

Computations (even in Matlab double precision), however, confirm this only for $m = 0$, the first counterexample occurring already when $n = 6$, $m = 1$, and $\lambda = -1/2 + h$, and others for $n = 7$, $m = 1$, and $\lambda = 0$, or $n = 8$, $m = 1$, and $\lambda = 1/2$.

Acknowledgments

The author is indebted to Mourad E. H. Ismail for the important reference [4].

References

[1] S. Ahmed, M. E. Muldoon and R. Spigler, Inequalities and numerical bounds for zeros of ultraspherical polynomials, *SIAM J. Math. Anal.* **17** (1986) 1000–1007.

[2] Á. Elbert and A. Laforgia, Some monotonicity properties of the zeros of ultraspherical polynomials, *Acta Math. Hung.* **48** (1986) 155–159.

[3] Á. Elbert and A. Laforgia, Monotonicity results on the zeros of generalized Laguerre polynomials, *J. Approx. Theory* **51** (1987) 168–174.

[4] Á. Elbert and P. Siafarikas, Monotonicity properties of the zeros of ultraspherical polynomials, *J. Approx. Theory* **97** (1999) 31–39.

[5] L. Gatteschi, On some approximations for the zeros of Jacobi polynomials, in *Approximation and computation*, International Series of Numerical Mathematics, Vol. 119 (Birkhäuser, Boston, MA, 1994), pp. 207–218.

[6] W. Gautschi, On a conjectured inequality for the largest zero of Jacobi polynomials, *Numer. Algorithms* **49** (2008) 195–198. [Also in *Selected Works*, Vol. 1, 518–521.]

[7] W. Gautschi, On conjectured inequalities for zeros of Jacobi polynomials, *Numer. Algorithms* **50** (2009) 93–96. [Also in *Selected Works*, Vol. 1, 523–526.]

[8] W. Gautschi, New conjectured inequalities for zeros of Jacobi polynomials, *Numer. Algorithms* **50** (2009) 293–296. [Also in *Selected Works*, Vol. 1, 528–531.]

[9] W. Gautschi, *Numerical Analysis*, 2nd edn. (Birkhäuser, New York, 2012).

[10] W. Gautschi, *Orthogonal Polynomials in Matlab: Exercises and Solutions*, Software, Environments, Tools (SIAM, Philadelphia, PA, 2016).

[11] W. Gautschi, Monotonicity properties of the zeros of Freud and sub-range Freud polynomials: analytic and empirical results, *Math. Comp.* **86** (2017) 855–864.

[12] W. Gautschi, Scripts for the Ismail–Letessier–Askey monotonicity conjecture for zeros of ultraspherical polynomials, *Purdue University Research Repository*, `doi:10.4231/R78G8HNQ`.

[13] W. Gautschi and P. Leopardi, Conjectured inequalities for Jacobi polynomials and their largest zeros, *Numer. Algorithms* **45** (2007) 217–230. [Also in *Selected Works*, Vol. 1, 503–516.]

[14] M. E. H. Ismail, The variation of zeros of certain orthogonal polynomials, *Adv. Appl. Math.* **8** (1987) 111–118.

[15] M. E. H. Ismail, Monotonicity of zeros of orthogonal polynomials, in *q-Series and Partitions*, ed. D. Stanton (Springer, New York, 1989), pp. 177–190.

[16] M. E. H. Ismail and J. Letessier, Monotonicity of zeros of ultraspherical polynomials, in *Orthogonal Polynomials and Their Applications*, Lecture Notes in Mathematics, Vol. 1329 (Springer, Berlin, 1988), Problem Section, pp. 329–330.

[17] M. E. H. Ismail and M. E. Muldoon, A discrete approach to monotonicity of zeros of orthogonal polynomials, *Trans. Amer. Math. Soc.* **323** (1991) 65–78.

[18] M. E. H. Ismail and R. Zhang, On the Hellmann–Feynman theorem and the variation of zeros of certain special functions, *Adv. Appl. Math.* **9** (1988) 439–446.

[19] M. E. H. Ismail and R. Zhang, The Hellmann–Feynman theorem and the variation of zeros of special functions, *Ramanujan International Symposium on Analysis* (Macmillan of India, New Delhi, 1989), pp. 151–183.

[20] A. Laforgia, Monotonicity properties for the zeros of orthogonal polynomials and Bessel functions, in *Orthogonal Polynomials and Applications* Lecture, Notes in Mathematics, Vol. 1171 (Springer, Berlin, 1985), pp. 267–277.

[21] A. Laforgia and M. E. Muldoon, Some consequences of the Sturm comparison theorem, *Amer. Math. Monthly* **93** (1986) 89–94.

[22] L. Lorch, Elementary comparison techniques for certain classes of Sturm–Liouville equations, in *Difference Equations*, Sympos. Univ. Upsaliensis Ann. Quingentesimum Celebrantis, No. 7 (Almqvist & Wiksell, Stockholm, 1977), pp. 12–133.

[23] F. W. J. Olver *et al.*, *NIST Handbook of Mathematical Functions* (US Department of Commerce and Cambridge University Press, 2010).

[24] G. Szegö, *Orthogonal Polynomials*, 4th edn. (American Mathematical Society, Providence, RI, 1975).

42.5. [218] "On the zeros of subrange Jacobi polynomials"

[218] "On the zeros of subrange Jacobi polynomials", *Numer. Algorithms* **79**, 759–768 (2018).

Numer Algor (2018) 79:759–768
https://doi.org/10.1007/s11075-017-0459-x

ORIGINAL PAPER

On the zeros of subrange Jacobi polynomials

Walter Gautschi[1]

Received: 27 October 2017 / Accepted: 7 December 2017 / Published online: 28 December 2017

Abstract All positive zeros of subrange Jacobi polynomials, orthogonal on $[-c, c]$, $0 < c < 1$, with respect to the weight function $w(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha > -1$, $\beta > -1$, are shown in the ultraspherical case $\alpha = \beta$, and partly conjectured in the general case $\alpha < \beta$, to be monotonically increasing as functions of c.

Keywords Subrange Jacobi polynomials · Zeros · Monotonicity

Mathematics Subject Classification 2010 33C47 · 65D20

1 Introduction

Subrange Jacobi polynomials are orthogonal with respect to the Jacobi weight function $w(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha > -1$, $\beta > -1$, supported either on the symmetric subinterval $[-c, c]$, $0 < c < 1$, or on an asymmetric interval $[-1, c]$, $-1 < c < 1$. Computational aspects of these orthogonal polynomials have been considered in [1] (under the more restrictive assumption $c > 0$ in the asymmetric case). Here we are interested in the zeros of these polynomials, in particular in monotonicity properties of the zeros when considered functions of c.

The asymmetric case is simpler, in this regard, since it has been proven, for general weight functions, that all zeros increase monotonically with c ([2, Theorem 2]). Here we consider the more interesting case of symmetric subintervals.

✉ Walter Gautschi
wgautschi@purdue.edu

[1] Department of Computer Science, Purdue University, West Lafayette, IN 47907-2066, USA

2 Symmetric subintervals

The monotonicity behavior of the zeros, as in [2], can be studied starting from the respective Gaussian quadrature formula,

$$\int_{-c}^{c} p(x)w(x)\mathrm{d}x = \sum_{\mu=1}^{n} \lambda_\mu(c)p(x_\mu(c)), \qquad p \in \mathbb{P}_{2n-1}, \tag{2.1}$$

where $x_\mu(c)$ are the zeros of the nth-degree subrange Jacobi polynomial and $\lambda_\mu(c)$ the corresponding Christoffel numbers. Differentiating with respect to c, we obtain

$$p(c)w(c) + p(-c)w(-c) = \sum_{\mu=1}^{n} \frac{\mathrm{d}\lambda_\mu(c)}{\mathrm{d}c}\, p(x_\mu(c)) + \sum_{\mu=1}^{n} \lambda_\mu(c)p'(x_\mu(c))\frac{\mathrm{d}x_\mu(c)}{\mathrm{d}c}. \tag{2.2}$$

The principal idea, going back to A. Markov (cf. [4, §6.12]), is to put here

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

where $\pi_n(x) = \pi_n^{(\alpha,\beta)}(x; c)$ is the (monic) subrange Jacobi polynomial of degree n, and $x_\nu = x_\nu(c)$ is some fixed zero of π_n. Clearly, $p(x_\mu) = 0$ for all μ, and also

$$p'(x_\mu) = 0 \ \text{ for } \mu \neq \nu, \quad \text{and} \quad p'(x_\nu) = [\pi_n'(x_\nu)]^2.$$

From (2.2), we therefore obtain

$$\pi_n^2(c)w(c)\left\{\frac{1}{c - x_\nu} - \left[\frac{\pi_n(-c)}{\pi_n(c)}\right]^2 \frac{w(-c)}{w(c)}\,\frac{1}{c + x_\nu}\right\} = \lambda_\nu(c)[\pi_n'(x_\nu)]^2\,\frac{\mathrm{d}x_\nu(c)}{\mathrm{d}c}. \tag{2.3}$$

In the special case $\alpha = \beta$ of ultraspherical polynomials, we have $w(-c) = w(c)$ and $\pi_n^2(-c) = \pi_n^2(c)$, so that (2.3) simplifies to

$$\pi_n^2(c)w(c)\left\{\frac{1}{c - x_\nu} - \frac{1}{c + x_\nu}\right\} = \lambda_\nu(c)[\pi_n'(x_\nu)]^2\,\frac{\mathrm{d}x_\nu}{\mathrm{d}c}\,. \tag{2.4}$$

Here,

$$\frac{1}{c - x_\nu} - \frac{1}{c + x_\nu} = \frac{2x_\nu}{c^2 - x_\nu^2}$$

is positive if $x_\nu > 0$. Since the other factors on the left of (2.4), as well as those on the right multiplying $\mathrm{d}x_\nu/\mathrm{d}c$, are all positive, it follows that $\mathrm{d}x_\nu/\mathrm{d}c > 0$. Thus, we have

Theorem 1 *All positive zeros of the subrange ultraspherical polynomials, orthogonal on* $[-c, c]$, $0 < c < 1$, *are monotonically increasing as functions of c.*

Remark to Theorem 1 From the proof of the theorem, it is clear that it holds for any symmetric weight function, i.e., for any positive function w satisfying $w(-x) = w(x)$ on $[-a, a]$, $0 < c < a \leq \infty$.

Fig. 1 Positive zeros of the subrange Chebyshev polynomial of degree $n = 30$ in dependence of the parameter c

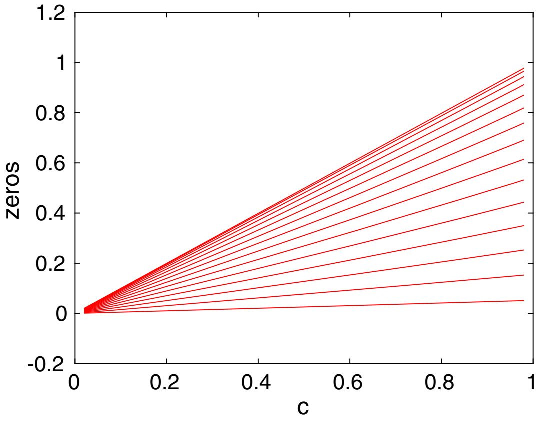


Figure 1, produced by the Matlab script `plot_zeros_srchebyshev.m`,[1] shows the positive zeros of the subrange Chebyshev polynomial ($\alpha = \beta = -1/2$) of degree $n = 30$ as functions of c.

In the general case $\alpha \neq \beta$, we may, and will, assume $\alpha < \beta$, because of $\pi_n^{(\alpha,\beta)}(x; c) = \pi_n^{(\beta,\alpha)}(-x; c)$. Since

$$\frac{w(-c)}{w(c)} = \left(\frac{1-c}{1+c}\right)^{\beta-\alpha},$$

we can write (2.3) in the form

$$\pi_n^2(c)(1-c)^\alpha(1+c)^\beta \left\{ \frac{1}{c - x_\nu} - \left[\frac{\pi_n(-c)}{\pi_n(c)}\right]^2 \left(\frac{1-c}{1+c}\right)^{\beta-\alpha} \frac{1}{c + x_\nu} \right\}$$
$$= \lambda_\nu(c)[\pi_n'(x_\nu)]^2 \frac{\mathrm{d}x_\nu}{\mathrm{d}c}, \tag{2.5}$$

so that the sign of the derivative $\mathrm{d}x_\nu/\mathrm{d}c$ is the same as the sign of the expression in curled brackets, which will be denoted by $\kappa_{n,\nu}(c)$,

$$\kappa_{n,\nu}(c) = \frac{1}{c - x_\nu} - \left[\frac{\pi_n(-c)}{\pi_n(c)}\right]^2 \left(\frac{1-c}{1+c}\right)^{\beta-\alpha} \frac{1}{c + x_\nu}. \tag{2.6}$$

It should be noted that $\kappa_{n,\nu}(1) = 1/(1 - x_\nu)$, which is positive since $x_\nu < 1$. Thus, in the unlikely scenario where c is very close to 1, some or even all zeros may be monotonically increasing near $c = 1$.

With regard to the factor multiplying the second term on the right of (2.6), we make the following interesting conjecture.

Conjecture *For any $n \geq 1$, $\alpha > -1$, $\beta > -1$ with $\alpha < \beta$, and for any c with $0 < c \leq 1$, there holds*

$$\omega := \left[\frac{\pi_n(-c)}{\pi_n(c)}\right]^2 \left(\frac{1-c}{1+c}\right)^{\beta-\alpha} < 1, \tag{2.7}$$

[1] All Matlab scripts and routines referenced in this paper can be accessed at the website http://www.cs.purdue.edu/archives/2002/wxg/codes/ZSRJAC.html.

where $\pi_n(\cdot) = \pi_n^{(\alpha,\beta)}(\cdot\,; c)$ *is the subrange Jacobi polynomial of degree* n *orthogonal on* $[-c, c]$ *relative to the weight function* $w(x) = (1-x)^\alpha(1+x)^\beta$.

Empirical evidence in support of the conjecture is provided in Appendix B.

3 Subrange Chebyshev polynomials of the fourth kind

This is the case of Jacobi polynomials with parameters $\alpha = -1/2$, $\beta = 1/2$, which merits special attention. We assume the zeros x_ν of the nth-degree polynomial ordered increasingly,

$$-c < x_1(c) < x_2(c) < \cdots < x_n(c) < c. \tag{3.1}$$

The Matlab script `plot_zeros_srchebyshev4.m` computes and plots the zeros of subrange Jacobi polynomials. In the case at hand, all zeros for $n = 30$ are shown at the bottom of Fig. 2 for $.02 \le c \le .98$. The plot suggests that the negative zeros are monotonically decreasing, and the positive zeros monotonically increasing as functions of c. Actually, this is not quite true. It is true for all zeros unless c is close to 1. For the values $c = .02 : .02 : .98$ of c, exceptions have been observed only for the largest negative zeros. When n is even, they are monotonically decreasing for c less than about .89, consistently for all (even) n, and monotonically increasing thereafter. When n is odd, they are decreasing for c less than about .979, and increasing thereafter. This is illustrated for $n = 30$ in the top-left plot of Fig. 2, and for $n = 29$ in the top-right plot. The behavior described is consistent with the remark made after (2.6).

4 Subrange Jacobi polynomials with parameters $\alpha < \beta$

In the case now under consideration, the upper half of the zeros display a more coherent pattern than the lower half, both with regard to the sign of the zeros and their monotone behavior as functions of c.

For the upper half of the zeros, we have

Theorem 2 *Assuming the conjecture in* Section 2 *is true, all zeros* x_ν, $\nu = \lfloor n/2 \rfloor + 1 : n$, *if positive, of the subrange Jacobi polynomial of degree* n, $n \ge 1$, *orthogonal on* $[-c, c]$, $0 < c < 1$, *with respect to the weight function* $w(x) = (1-x)^\alpha(1+x)^\beta$, $\alpha < \beta$, *are monotonically increasing as functions of* c.

Proof From (2.6) and (2.7), we have

$$\frac{1}{c - x_\nu} - \frac{\omega}{c + x_\nu} = \frac{(1-\omega)c + (1+\omega)x_\nu}{c^2 - x_\nu^2} > 0,$$

since $0 < x_\nu < c$ by assumption, and $0 < \omega < 1$ by the conjecture of Section 2. Thus, the assertion of Theorem 2 follows from what was said after (2.5). □

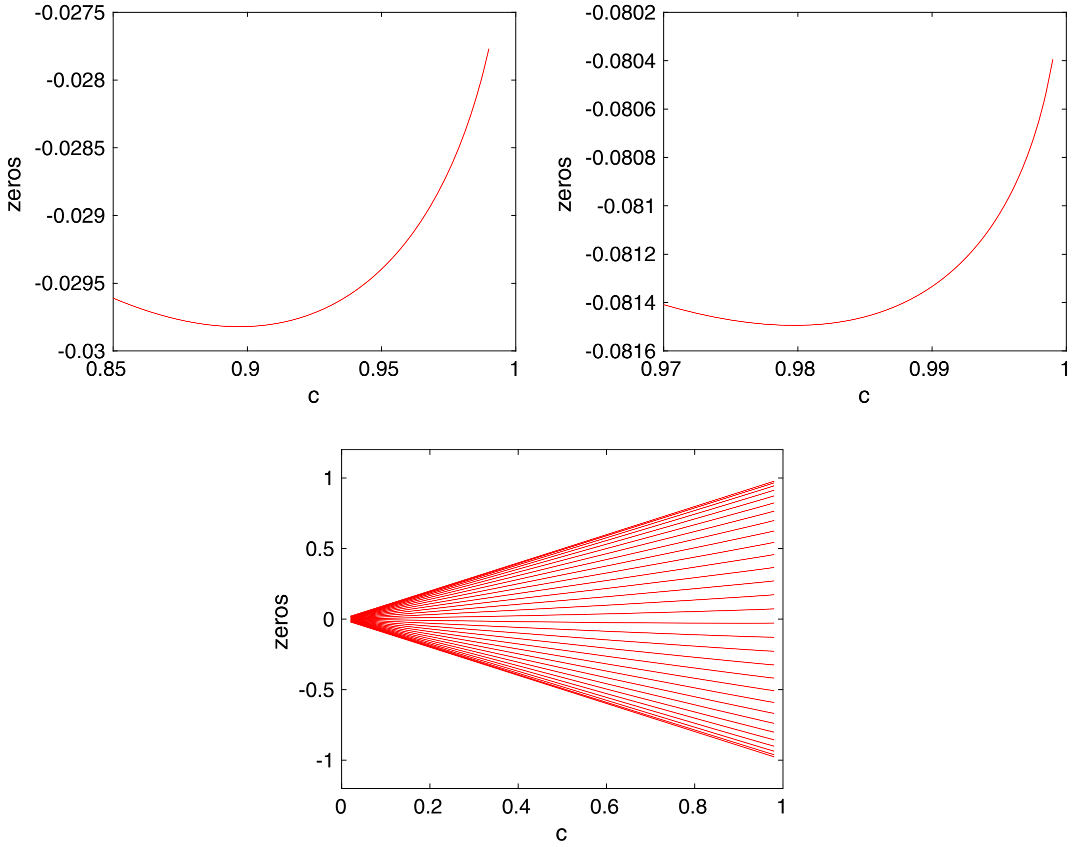


Fig. 2 Zeros of the subrange Chebyshev polynomial of the fourth kind in dependence of the parameter c; $n = 30$, largest negative zero (top left); $n = 29$, largest negative zero (top right); $n = 30$, all zeros (bottom)

Remark 1. All zeros x_ν in question were found, using the Matlab script `mon_zeros_subjacobi.m` (which also verifies monotonicity of the zeros), to be positive when $n \leq 30$, $\alpha, \beta = -.8 : .2 : 10$, $\alpha < \beta$, and $c = .02 : .02 : .98$. The script may take many hours to run.

2. The conjecture in Section 2 has been proven to be true [3] for Jacobi parameters $\alpha \leq 0$, $\beta \geq 0$. In this case, the claim in Theorem 2 also follows from A. Markov's theorem; see Appendix A.

Plots of the zeros x_ν, $\nu = \lfloor n/2 \rfloor + 1 : n$, for $n = 30$, $\alpha = -.8$, $\beta = 10$, are produced by the Matlab script `plot_zeros_subjacobi.m` and shown on the left of Fig. 3. The plot on the right shows the zeros x_ν, $\nu = 1 : \lfloor n/2 \rfloor$, for the same values of n, α, and β. Here, as was determined by the Matlab script `run_anal_zeros_subjacobi.m`, the zeros start out decreasing monotonically, but at some point, sooner the larger the zero, turn around and begin to increase. The two largest zeros even become positive, the first at about $c = .78$, the other already at about $c = .3$. The same pattern, perhaps in a less pronounced form, persists for smaller values of $\beta - \alpha$.

Both plots were produced by the routine `plot_zeros_subjacobi.m`.

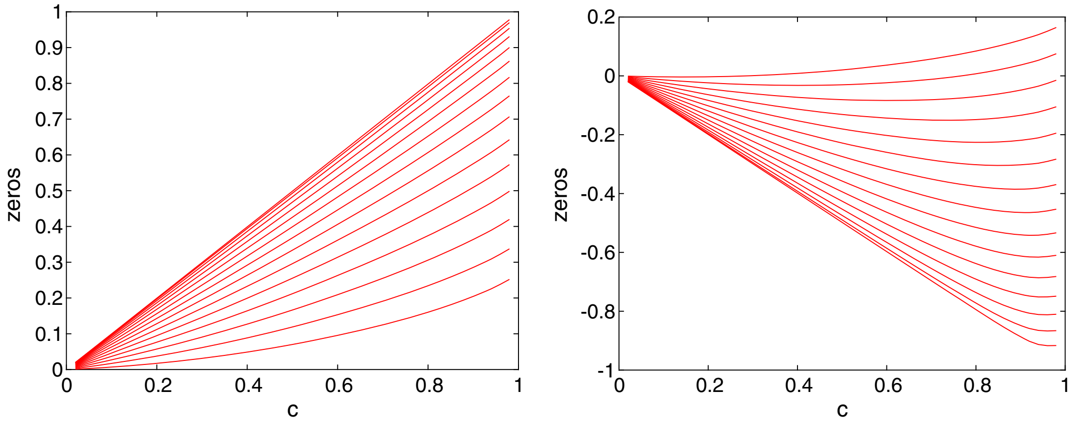


Fig. 3 Zeros of the subrange Jacobi polynomial of degree $n = 30$ for $\alpha = -.8$, $\beta = 10$ in dependence of the parameter c; upper half of the zeros (left); lower half of the zeros (right)

Appendix A: A partial result

Denote the subrange Jacobi polynomial of degree n, orthogonal on $[-c, c]$, $0 < c < 1$, relative to the Jacobi weight function $w^{(\alpha,\beta)}(x) = (1-x)^\alpha(1+x)^\beta$, by $\pi_n(x) = \pi_n^{(\alpha,\beta)}(x; c)$. Let $p_n(x) = p_n^{(\alpha,\beta)}(x; c)$ be the monic polynomial of degree n orthogonal on $[-1, 1]$ relative to the weight function $v^{(\alpha,\beta)}(x; c) = w^{(\alpha,\beta)}(cx)$, $-1 \le x \le 1$. Then, as is easily seen (also see [1, §2]),

$$p_n^{(\alpha,\beta)}(x; c) = \frac{1}{c^n}\,\pi_n^{(\alpha,\beta)}(cx; c). \tag{A.1}$$

Since $p_n^{(\alpha,\beta)}$ is a polynomial orthogonal with respect to the weight function $w(x; c) = (1-cx)^\alpha(1+cx)^\beta$, $x \in (-1, 1)$, depending on a parameter, we can apply A. Markov's theorem (cf. [4, §6.12]). According to this theorem, all zeros x_ν of $p_n^{(\alpha,\beta)}$ are monotonically increasing functions of c in $(0, 1)$ provided the Markov function

$$M(x; c) = \frac{\partial w(x; c)/\partial c}{w(x; c)}$$

is monotonically increasing as a function of x in $(-1, 1)$. We have

$$\begin{aligned}\frac{\partial w(x; c)}{\partial c} &= \alpha(1-cx)^{\alpha-1}(-x)(1+cx)^\beta + \beta x(1-cx)^\alpha(1+cx)^{\beta-1}\\ &= x\left\{\beta(1-cx)^\alpha(1+cx)^{\beta-1} - \alpha(1-cx)^{\alpha-1}(1+cx)^\beta\right\},\end{aligned}$$

so that

$$M(x; c) = x\left\{\frac{\beta}{1+cx} - \frac{\alpha}{1-cx}\right\}, \quad -1 < x < 1.$$

Since

$$\left(\frac{x}{1+cx}\right)' = \frac{1}{(1+cx)^2}, \quad \left(\frac{x}{1-cx}\right)' = \frac{1}{(1-cx)^2}$$

(primes indicate derivatives with respect to x), there holds

$$M'(x;c) = \frac{\beta}{(1+cx)^2} - \frac{\alpha}{(1-cx)^2}\,.$$

Therefore, if $\alpha \leq 0$ and $\beta \geq 0$ and not both, α and β, are zero, in which case Theorem 1 applies, we have $M'(x;c) > 0$ on $(-1,1)$. Hence, by Markov's theorem, all zeros of the orthogonal polynomial $p_n^{(\alpha,\beta)}$ are monotonically increasing as functions of c. Since the zeros of $\pi_n^{(\alpha,\beta)}$ are $\xi_\nu = c\,x_\nu$ and

$$\frac{\mathrm{d}\xi_\nu}{\mathrm{d}c} = x_\nu + c\,\frac{\mathrm{d}x_\nu}{\mathrm{d}c},$$

it follows that all positive zeros of $\pi_n^{(\alpha,\beta)}(\,\cdot\,;c)$ are monotonically increasing functions of c in (0, 1).

Appendix B: Empirical evidence supporting the conjecture in Section 2

Using (A.1), we get from (2.7)

$$\omega = \left[\frac{p_n^{(\alpha,\beta)}(-1;c)}{p_n^{(\alpha,\beta)}(1;c)}\right]^2 \left(\frac{1-c}{1+c}\right)^{\beta-\alpha}. \tag{B.1}$$

Since $w^{(\alpha,\beta)}(0) \equiv 1$, the polynomial $p_n^{(\alpha,\beta)}(x;c)$, as $c \downarrow 0$, tends to the monic Legendre polynomial of degree n, and hence $\omega = 1$ for $c = 0$. What we conjecture, in fact, is that for any $n \geq 1$ and any α, β with $\beta > \alpha > -1$, the quantity ω in (B.1) decreases monotonically as a function of c, $c \in (0, 1]$. Since $\omega = 1$ for $c = 0$, it then follows that $0 < \omega < 1$ for c in $(0, 1]$, as conjectured in (2.7).

To prove this analytically, by differentiating ω with respect to c, is virtually impossible, and even unlikely to succeed in symbolic computation, since the dependence of $p_n^{(\alpha,\beta)}(\,\cdot\,;c)$ on c is very complicated. However, we provide rather compelling empirical evidence that ω indeed decreases as a function of c.

We begin by computing and plotting ω for $n = 30$, for $c = 0 : .01 : 1$, and for selected values of α, β, namely the seven values $\alpha = [-1/2, 0, 1/2, 1, 2, 5, 10]$ of α and for each α the five values $\beta = \alpha + [1/2, 1, 2, 5, 10]$ of β. The seven plots are produced by the Matlab routine `conj_omega.m`, which also verifies monotonicity.

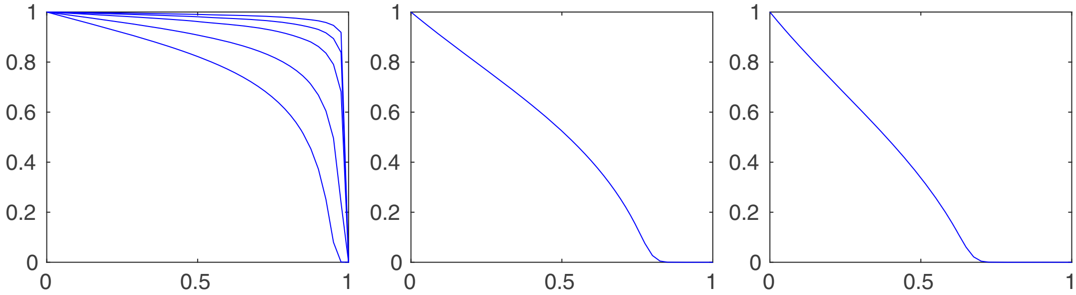


Fig. 4 The function ω in dependence of c for $n = 30$ and selected values of (α, β), $\beta > \alpha$

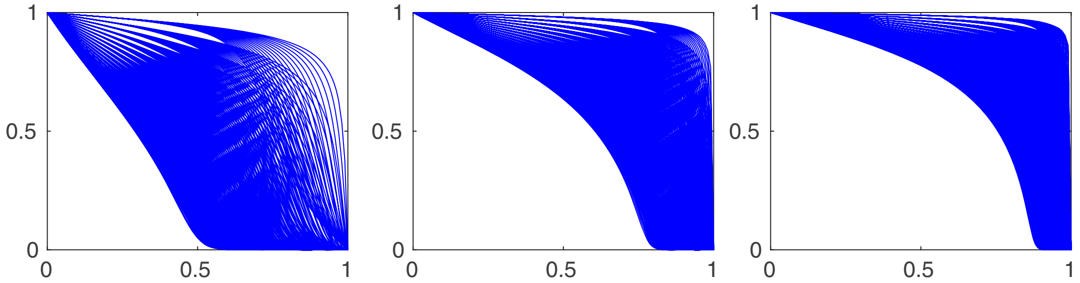


Fig. 5 The function ω in dependence of c for $n = 10$, $n = 30$, $n = 50$ (from left to right), and for integer values of (α, β), $\beta > \alpha$

Since all these plots look quite alike, we show, in the far left of Fig. 4, only the fourth plot as a representative for all seven. The same routine also plots ω for $\alpha = -1/2$, $\beta = 30$ and for $\alpha = 5$, $\beta = 50$, the result of which is shown in the second and third plot of Fig. 4. The flatness at the bottom of these curves, of course, is caused by the second factor in (B.1).

Clearly, when $\beta = \alpha$, we have $\omega = 1$, since not only the second factor in (B.1), but also the first, is equal to 1. Thus, when β is close to, but larger than α, inevitably ω will be close to 1 for quite a while, but eventually will have to vanish at $c = 1$ owing to the second factor vanishing there. This explains the behavior of the curves near the top (corresponding to $\beta \approx \alpha$) of the first plot in Fig. 4, and also makes plausible the behavior of ω shown in the other two plots of the figure, typical for values of α and β far apart.

The plots in Fig. 5 show ω for $\alpha = 0 : 20$, $\beta = \alpha + [1 : 20]$, that is, for integer values of α, β, for which the computation (by 1-component discretization) of the polynomials $p_n^{(\alpha,\beta)}(\,\cdot\,; c)$ is fast, since the weight function $w^{(\alpha,\beta)}(cx)$ is a polynomial of degree $\alpha + \beta$; cf. [1, §2]. The three plots, produced by the Matlab script `run0_omega.m`, are respectively for $n = 10$, $n = 30$, and $n = 50$.

The evidence for the validity of our conjecture produced so far relates mostly to the rapid descent of ω to 0 with c approaching 1; see, however, Fig. 4. The more delicate behavior for small c, say $0 \le c \le .1$, requires further scrutiny. Figure 6, produced by the script `run1_omega.m`, shows plots of ω for c in the interval [0, .1], for $\beta = \alpha + [1 : 20]$ (from top to bottom), superimposed as α varies from 0 to 20 in steps of 1. It can be seen that each plot is practically a linearly decreasing

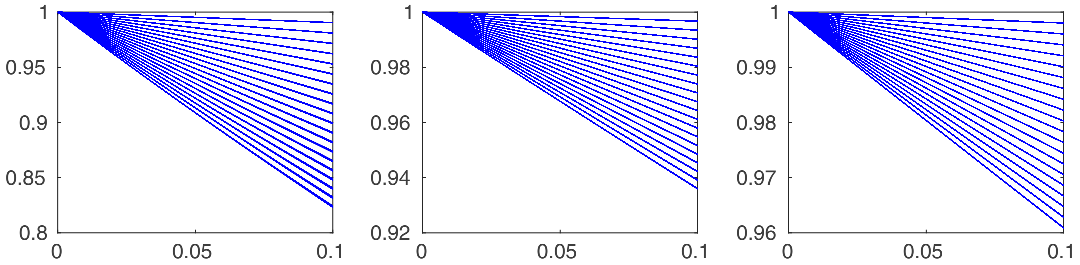


Fig. 6 The function ω in dependence of c on [0, .1] for $n = 10$, $n = 30$, $n = 50$ (from left to right), and in each plot for $\beta = \alpha + [1 : 20]$ (where $\alpha = 0 : 20$)

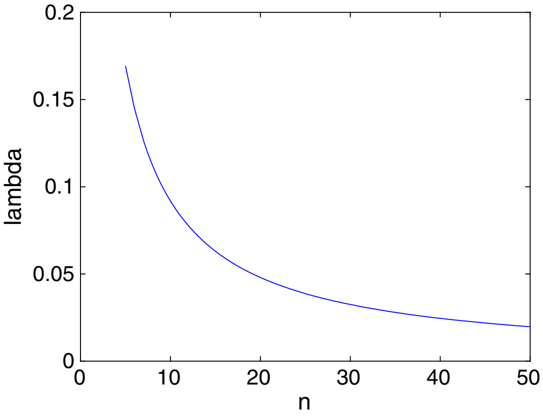


Fig. 7 The coefficient $\lambda(n)$ in (B.2)

Table 1 Approximation errors for ω

n	maxerr	α	β	c	minerr	α	β	c
5	3.94 (− 2)	0	20	.1	5.45 (− 6)	11	26	.06
10	9.94 (− 3)	0	20	.1	2.84 (− 7)	0	14	.06
15	4.44 (− 3)	0	20	.1	6.08 (− 7)	17	37	.04
20	2.50 (− 3)	0	20	.1	1.17 (− 8)	19	32	.07
25	1.61 (− 3)	0	20	.1	1.29 (− 7)	8	21	.06
30	1.12 (− 3)	0	20	.1	1.10 (− 7)	8	22	.05
35	8.22 (− 4)	0	20	.1	2.22 (− 7)	15	26	.09
40	6.30 (− 4)	0	20	.1	1.87 (− 8)	10	28	.03
45	4.98 (− 4)	0	20	.1	2.23 (− 8)	0	13	.04
50	4.03 (− 4)	0	20	.1	1.82 (− 8)	3	13	.06

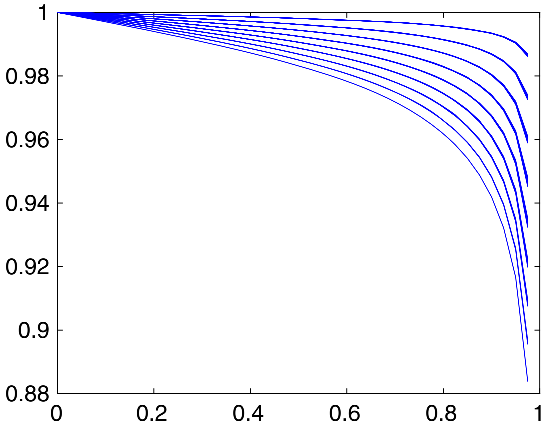


Fig. 8 The function ω in dependence of c for $n = 30$ and selected values of (α, β) in the domain $0 \geq \beta > \alpha > -1$

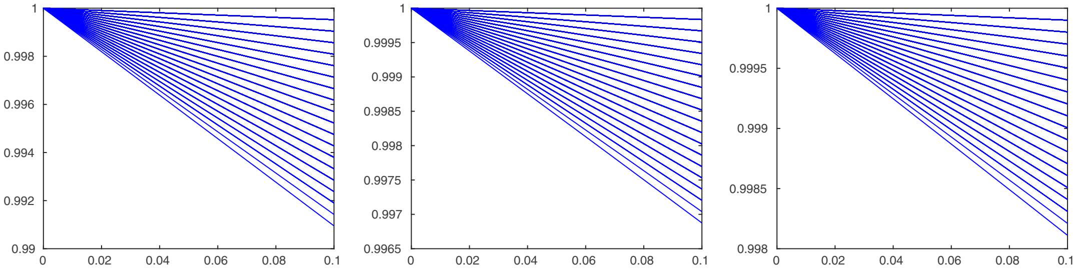


Fig. 9 The function ω in dependence of c on $[0, .1]$ for $n = 10$, $n = 30$, $n = 50$ (from left to right), and in each plot for $\beta = \alpha + [.05 : .05 : 1] \leq 0$ (where $\alpha = -.95 : .05 : -.05$)

function of c and (judging from the thickness of the curve) essentially independent of α, aslight dependence showing only when c is near .1 and β near 20. The slopes of the curves, using the script `run2_omega.m`, are found to depend essentially linearly on $\beta - \alpha$, with coefficient $-\lambda$ depending less on β the larger n. Taking appropriate mean values, one finds

$$\omega \approx 1 - \lambda(n)(\beta - \alpha)c, \quad 0 \leq c \leq .1, \ \ 5 \leq n \leq 50. \tag{B.2}$$

The coefficients $\lambda(n)$ for $n = 5 : 50$ are generated by the script `run2_omega.m` and plotted by the script `plot_lambda.m`. The curve produced is shown in Fig. 7.

The maximum and minimum relative error in ω, taken over all $\alpha = 0 : 20$, $\beta = \alpha + [1 : 20]$, $c = .01 : .01 : .1$, are shown in Table 1, where the columns headed by α, β, c show the values of these parameters at which the maxima resp. minima are attained.

As can be seen, the maximum error, attained consistently at $\alpha = 0$ and at the largest values of β and c, decreases from about 4×10^{-2} to 4×10^{-4} as n increases from 5 to 50, whereas the minimum error, not unexpectedly, is always attained near $c = 0$.

The (α, β)-domain covered so far is $\beta > \alpha \geq 0$. It remains to look at the small triangular domain $0 \geq \beta > \alpha > -1$, since for $\alpha < 0$, $\beta > 0$ the conjecture has been proved; see Remark 2 in Section 4. The evidence in this case is very similar to the one already presented. In fact, the findings analogous to those depicted in the first plot of Fig. 4, but for $\alpha, \beta = -.9 : .1 : 0$, $\beta > \alpha$, are shown in Fig. 8 produced by the script `conj_omega0.m`, while those analogous to Fig. 6 are shown in Fig. 9 produced by the script `run1_omega0.m`.

References

1. Gautschi, W.: Sub-range Jacobi polynomials. Numer. Algorithms **61**, 649–657 (2012)
2. Gautschi, W., Milovanović, G.V.: Binet-Type Polynomials and Their Zeros, in preparation
3. Milovanović, G.V.: Personal Communication. Email October 22 (2017)
4. Szegö, G.: Orthogonal polynomials, 4th edn., Colloquium Publications, vol. 23. American Mathematical Society, Providence (1975)

43

Paper on Approximation

[217] A discrete top-down Markov problem in approximation theory, in: Frontiers in orthogonal polynomials and q-series (M. Zuhair Nashed and Xin Li, eds.), 267–289, World Sci. Publ., Hackensack, NJ, 2018

43.1. [217] “A discrete top-down Markov problem in approximation theory”

[217] “A discrete top-town Markov problem in approximation theory”, in: *Frontiers in orthogonal polynomials and q-series* (M. Zuhair Nashed and Xin Li, eds.), 267–289 (2018).

Chapter 14

A Discrete Top-Down Markov Problem in Approximation Theory

Walter Gautschi

Department of Computer Science, Purdue University
West Lafayette, IN 47907-2107, USA
wgautschi@purdue.edu

The Markov brothers' inequalities in approximation theory concern polynomials p of degree n and assert bounds for the kth derivatives $|p^{(k)}|$, $1 \le k \le n$, on $[-1,1]$, given that $|p| \le 1$ on $[-1,1]$. Here we go in the other direction, seeking bounds for $|p|$, given a bound for $|p^{(k)}|$. For the problem to be meaningful, additional restrictions on p must be imposed, for example, $p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0$. The problem then has an easy solution in the continuous case, where the polynomial and their derivatives are considered on the whole interval $[-1,1]$, but is more challenging, and also of more interest, in the discrete case, where one focuses on the values of p and $p^{(k)}$ on a given set of $n-k+1$ distinct points in $[-1,1]$. Analytic solutions are presented and their fine structure analyzed by computation.

Keywords: Markov problem; discrete; top-down.

Mathematics Subject Classification 2010: 41A10, 41A17

1. Introduction

For a polynomial p of degree $n = 2$ such that

$$|p(t)| \le 1 \quad \text{for } -1 \le t \le 1, \tag{1.1}$$

the 19th-century chemist Dmitri I. Mendeleev (the creator of the periodic table of elements) raised the question of what this implies with regard to the magnitude of $|p'(t)|$ on $[-1,1]$. He found the answer to be 4 and told this to one of his colleagues, the mathematician Andrei A. Markov, who in turn generalized and solved the question in 1890 for arbitrary n by showing that

$$|p'(t)| \leq n^2 \quad \text{for } -1 \leq t \leq 1. \tag{1.2}$$

For higher, say kth-order, derivatives, $1 \leq k \leq n$, the same problem was solved by the younger brother Vladimir A. Markov, who showed, rather remarkably, that

$$|p^{(k)}(t)| \leq \frac{n^2(n^2-1^2)\cdots[n^2-(k-1)^2]}{1\cdot 3\cdots(2k-1)} \quad \text{for } -1 \leq t \leq 1. \tag{1.3}$$

The results of the brothers Markov have generated a great deal of interest in the approximation theory community and led to a considerable body of literature; see, e.g., the historical review of Shadrin [5].

We may call these problems “bottom-up”, since they go from the 0th derivative up to the kth derivative. Here we look at *top-down Markov problems*, assuming

$$|p^{(k)}(t)| \leq 1 \quad \text{for } -1 \leq t \leq 1, \tag{1.4}$$

and seeking a bound for $|p(t)|$ on $[-1,1]$. Since we can always add a polynomial of degree $k-1$ to p without affecting (1.4), top-down Markov problems become meaningful only if additional restrictions are imposed on the polynomial p. The simplest ones are to prescribe the values of the polynomial and its first $k-1$ derivatives to be zero at the left endpoint of the interval, $p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0$. Even then, the answer is fairly trivial, since $p(t)$ is the k-times iterated integral from -1 to t of $p^{(k)}$, i.e., because of the initial conditions imposed,

$$p(t) = \int_{-1}^{t} \frac{(t-\tau)^{k-1}}{(k-1)!}\, p^{(k)}(\tau)\mathrm{d}\tau. \tag{1.5}$$

Therefore, by (1.4),

$$|p(t)| \leq \int_{-1}^{t} \frac{(t-\tau)^{k-1}}{(k-1)!}\,\mathrm{d}\tau = \frac{(t+1)^k}{k!} \leq \frac{2^k}{k!} \quad \text{for } -1 \leq t \leq 1. \tag{1.6}$$

The bound is sharp, being attained when $p^{(k)}(t) \equiv 1$.

This straightforward answer is probably the reason why top-down Markov problems have never been given any attention. However, if we replace (1.4) by

$$|p^{(k)}(\tau_\nu)| \le 1, \quad \nu = 1, 2, \ldots, n-k+1, \tag{1.7}$$

for a set of $n-k+1$ distinct points $\{\tau_\nu\}$ in $[-1,1]$ and, as before, assume $p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0$, we can ask, for each $\nu = 1, 2, \ldots, n-k+1$, the question of bounding $|p(\tau_\nu)|$ over all polynomials satisfying the constraints imposed. This problem is meaningful and not without interest; we call it a *discrete top-down Markov problem*. To the best of our knowledge, the problem has never been studied before. The special case $k = 1$ of the problem, however, came up recently in work of Hager *et al.* [4] on orthogonal collocation methods in optimal control. The n points $\{\tau_\nu\}$ in this application are the zeros of the Legendre polynomial of degree n, and the authors conjectured for $|p(\tau_n)|$ the bound $\tau_n + 1$ and corresponding extremal polynomial $p^*(t) = t + 1$ and the same bound for $|p(\tau_\nu)|$, $1 \le \nu < n$.

Thus, the problem we propose is the following.

Problem 1.1 (Discrete Top-Down Markov Problem). *Given integers $n \ge 1$ and $1 \le k \le n$, and given $n-k+1$ distinct points $\mathbb{T}_n^{(k)} = \{\tau_\nu\}$ in $[-1,1]$, $-1 \le \tau_1 < \tau_2 < \cdots < \tau_{n-k+1} \le 1$, consider the following class of polynomials of degree n,*

$$\begin{aligned}\mathbb{Q}_n^{(k)} = \{p \in \mathbb{P}_n :\, & p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0, \\ & \textit{and } |p^{(k)}(\tau_\nu)| \le 1 \quad \textit{for } \nu = 1, 2, \ldots, n-k+1\}.\end{aligned} \tag{1.8}$$

For each ν, $\nu = 1, 2, \ldots, n-k+1$, determine the maximum possible value $M_{n,\nu}^{(k)}$ of $|p(\tau_\nu)|$ when $p \in \mathbb{Q}_n^{(k)}$,

$$M_{n,\nu}^{(k)} = \max_{p \in \mathbb{Q}_n^{(k)}} |p(\tau_\nu)|, \quad \nu = 1, 2, \ldots, n-k+1. \tag{1.9}$$

It may be instructive to look at two very special cases, in which not only the continuous, but also the discrete problem has an easy solution. The first case is $n = k = 1$, and $\tau_1 \in (-1, 1]$. Here we are dealing with the class of linear functions $p(t)$ that vanish at $t = -1$. In the continuous problem one adds the condition $|p'(t)| \le 1$ on the interval $[-1,1]$, and in the discrete problem, the condition $|p'(\tau_1)| \le 1$ at some point τ_1 in $(-1,1]$. In both problems the extremal polynomials are $p(t) = t + 1$, for which

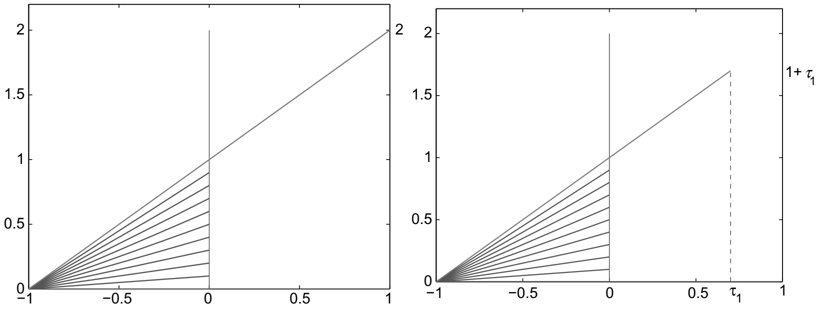


Fig. 1. Continuous vs. discrete top-down Markov problem in the case $n = k = 1$.

$\max_{|t|\leq 1} |p(t)| = 2$ in the continuous case, and $M_{1,1}^{(1)} = 1 + \tau_1$ in the discrete case; see Fig. 1.

The second special case is $k = n$, $\tau_1 \in (-1, 1]$. Here, the initial conditions in the problem yield $p(t) = p^{(n)}(-1)(t+1)^n/n!$ and the assumption about $p^{(n)}$, which is a constant, is $|p^{(n)}| \leq 1$. Therefore, $\max_{|t|\leq 1} |p(t)| = 2^n/n!$ in the continuous problem, and

$$M_{n,1}^{(n)} = \frac{(1+\tau_1)^n}{n!} \quad (k = n) \tag{1.10}$$

in the discrete one, the extremal polynomials being $(t+1)^n/n!$ in both cases.

In the general case, explicit answers to the discrete top-down problem can still be given, but seem to require computational work to get detailed information about their properties for concrete choices of the support points. In Section 2 we give a complete solution of our problem and introduce terminology for further study. Computer software to evaluate the solution is provided in Section 3, and specific examples are discussed in Section 4.

2. Solution of the Problem

We present the solution of our problem in the form of a theorem.

Theorem 2.1. *Let ℓ_μ, $\mu = 1, 2, \ldots, n-k+1$, be the elementary Lagrange interpolation polynomials of degree $n-k$ associated with the $n-k+1$ support* *points of* $\mathbb{T}_n^{(k)}$,

$$\ell_\mu(\tau) = \prod_{\substack{\kappa=1\\ \kappa\neq\mu}}^{n-k+1} \frac{\tau-\tau_\kappa}{\tau_\mu-\tau_\kappa},\ \mu = 1,2,\dots,n-k+1 \quad \textit{if}\ k<n;$$

$$\ell_1(\tau) = 1 \quad \textit{if}\ k=n. \tag{2.1}$$

Let $\mathbf{s}_\nu^{(k)} = [s_{\nu,1}^{(k)}, s_{\nu,2}^{(k)}, \dots, s_{\nu,n-k+1}^{(k)}]$ *be the vector with entries* $s_{\nu,\mu}^{(k)} = 1$ *if the integral*

$$I_{\nu,\mu}^{(k)} = \int_{-1}^{\tau_\nu} (\tau_\nu-\tau)^{k-1}\ell_\mu(\tau)\mathrm{d}\tau \tag{2.2}$$

is positive, $s_{\nu,\mu}^{(k)} = -1$ *if it is negative, and an arbitrary value, for example* $s_{\nu,\mu}^{(k)} = 0$*, otherwise. Define the polynomial* $p_{\nu,k}^*$ *of degree* n *by*

$$p_{\nu,k}^*(t) = \frac{1}{(k-1)!}\int_{-1}^{t} (t-\tau)^{k-1} p_{n-k}(\tau;\mathbf{s}_\nu^{(k)})\mathrm{d}\tau, \tag{2.3}$$

where $p_{n-k}(\,\cdot\,;\mathbf{s}_\nu^{(k)})$ *is the interpolation polynomial of degree* $\le n-k$ *passing through the* $n-k+1$ *points* $(\tau_\mu, s_{\nu,\mu}^{(k)})$*. Then*

$$M_{n,\nu}^{(k)} = \frac{1}{(k-1)!}\sum_{\mu=1}^{n-k+1}\left|\int_{-1}^{\tau_\nu}(\tau_\nu-\tau)^{k-1}\ell_\mu(\tau)\mathrm{d}\tau\right|,$$

$$\nu = 1,2,\dots,n-k+1, \tag{2.4}$$

and $p_{\nu,k}^*$ *in* (2.3) *is the associated extremal polynomial, for which*

$$M_{n,\nu}^{(k)} = p_{\nu,k}^*(\tau_\nu), \quad \nu = 1,2,\dots,n-k+1. \tag{2.5}$$

Proof. Let $p \in \mathbb{Q}_n^{(k)}$, $1 \le k \le n$. Because of $p(-1) = p'(-1) = \cdots = p^{(k-1)}(-1) = 0$, one has (cf. (1.5))

$$p(t) = \int_{-1}^{t} \frac{(t-\tau)^{k-1}}{(k-1)!}\, p^{(k)}(\tau)\mathrm{d}\tau. \tag{2.6}$$

Since the $(n-k+1)$-point Lagrange interpolation formula produces an exact identity when applied to any polynomial of degree $\le n-k$, we can write

$$p^{(k)}(\tau) = \sum_{\mu=1}^{n-k+1} p^{(k)}(\tau_\mu)\ell_\mu(\tau), \tag{2.7}$$

with ℓ_μ as defined in (2.1). Therefore, by (2.6),

$$p(t) = \int_{-1}^{t} \frac{(t-\tau)^{k-1}}{(k-1)!} \sum_{\mu=1}^{n-k+1} p^{(k)}(\tau_\mu)\ell_\mu(\tau)\mathrm{d}\tau,$$

that is,

$$p(t) = \frac{1}{(k-1)!} \sum_{\mu=1}^{n-k+1} p^{(k)}(\tau_\mu) \int_{-1}^{t} (t-\tau)^{k-1}\ell_\mu(\tau)\mathrm{d}\tau. \tag{2.8}$$

Now consider the case $t = \tau_\nu$. We have

$$|p(\tau_\nu)| = \frac{1}{(k-1)!} \left| \sum_{\mu=1}^{n-k+1} p^{(k)}(\tau_\mu) \int_{-1}^{\tau_\nu} (\tau_\nu-\tau)^{k-1}\ell_\mu(\tau)\mathrm{d}\tau \right|,$$
$$\nu = 1, 2, \ldots, n-k+1. \tag{2.9}$$

The only way we can control the magnitude of this expression is by choosing the values $p^{(k)}(\tau_\mu)$ subject to the constraints of the problem. To maximize $|p(\tau_\nu)|$, we select $p^{(k)}(\tau_\mu) = 1$ if the integral in (2.9), that is, $I_{\nu,\mu}^{(k)}$ in (2.2), is positive, $p^{(k)}(\tau_\mu) = -1$ if it is negative, and an arbitrary value $p^{(k)}(\tau_\mu)$ otherwise. This is always possible, since $p^{(k)}$ is a polynomial of degree $n-k$, and yields (2.4).

The extremal polynomial, by construction and the definition of the sign vector $\mathbf{s}_\nu^{(k)}$ is (2.8) with $p^{(k)}(\tau_\mu) = s_{\nu,\mu}^{(k)}$, $\mu = 1, 2, \ldots, n-k+1$, that is, (2.3). □

We remark that the integral (2.2) may indeed vanish for some μ, in which case $s_{\nu,\mu}^{(k)}$ can be chosen arbitrarily. A simple example is $n = 2$, $k = 1$, and $\tau_1 = -1/3$, $\tau_2 = 1/3$. Then $I_{2,2}^{(1)} = \int_{-1}^{1/3} \ell_2(\tau)\mathrm{d}\tau = \int_{-1}^{1/3}[(3/2)\tau + 1/2]\mathrm{d}\tau = 0$. If we choose $s_{2,2}^{(1)} = \sigma$, with σ arbitrary, one gets from (2.3) and (2.5),

$$M_{2,2}^{(1)} = \int_{-1}^{1/3} p_1(\tau; \mathbf{s}_2^{(1)})\mathrm{d}\tau = \int_{-1}^{1/3} \left[\frac{3}{2}(\sigma-1)\tau + 1 + \frac{1}{2}(\sigma-1)\right]\mathrm{d}\tau$$
$$= \left[\frac{3}{4}(\sigma-1)\tau^2 + (1 + \frac{1}{2}(\sigma-1))\tau\right]_{-1}^{1/3} = \frac{4}{3},$$

which is the correct answer, regardless of the choice of σ.

Note that for $k = n$, since $\ell_1(\tau) = 1$, we get from (2.4)

$$M_{n,1}^{(n)} = \frac{(1+\tau_1)^n}{n!},$$

in agreement with (1.10).

The value of $M_{n,\nu}^{(k)}$ in (2.4) could be further estimated by

$$\begin{aligned} M_{n,\nu}^{(k)} &< \frac{1}{(k-1)!}\sum_{\mu=1}^{n-k+1}\int_{-1}^{\tau_\nu}(\tau_\nu-\tau)^{k-1}|\ell_\mu(\tau)|\mathrm{d}\tau \\ &< \frac{(\tau_\nu+1)^{k-1}}{(k-1)!}\int_{-1}^{\tau_\nu}\sum_{\mu=1}^{n-k+1}|\ell_\mu(\tau)|\mathrm{d}\tau \\ &= \frac{(\tau_\nu+1)^{k-1}}{(k-1)!}\int_{-1}^{\tau_\nu}\lambda_{n-k+1}(\tau)\mathrm{d}\tau \\ &< \frac{(\tau_\nu+1)^{k}}{(k-1)!}\Lambda_{n-k+1}, \quad \nu = 1,2,\ldots,n-k+1, \end{aligned}$$

where $\lambda_{n-k+1}(\tau)$ is the Lebesgue function for $(n-k+1)$-point polynomial interpolation and $\Lambda_{n-k+1} = \max_{-1\le\tau\le1}\lambda_{n-k+1}(\tau)$ the Lebesgue constant. However, when k is fixed and n large, this may be a gross over-estimation, since, even in the best of circumstances, Λ_{n-k+1} grows logarithmically to ∞ as $n\to\infty$ whereas the bounds $M_{n,\nu}^{(k)}$, as will be seen in Section 4.1.2, are often smaller than 2.

The sign pattern for the integrals $I_{\nu,\mu}^{(k)}$ in (2.2) is characterized by the $(n-k+1)\times(n-k+1)$ "sign-pattern matrix"

$$\mathbf{S}_{n-k+1} = [s_{i,j}^{(k)}] = \begin{bmatrix} \mathbf{s}_1^{(k)} \\ \mathbf{s}_2^{(k)} \\ \vdots \\ \mathbf{s}_{n-k+1}^{(k)} \end{bmatrix}, \tag{2.10}$$

where each row vector $\mathbf{s}_\nu^{(k)}$ exhibits the signs of the integrals $I_{\nu,\mu}^{(k)}$ for $\mu = 1,2,\ldots,n-k+1$. The matrix $\mathbf{S}_{n-k+1}$ is uniquely determined by the $n-k+1$ support points τ_ν if all integrals $I_{\nu,\mu}^{(k)}$ are different from zero. Its features greatly influence the way the bounds $M_{n,\nu}^{(k)}$ behave.

We illustrate this in the case $k = 1$ and τ_ν the zeros of the Legendre polynomial of degree n. In this case (and in many others, cf. the end of

Section 4.1.2), the matrix $\mathbf{S}_n$ is found to have the following characteristic structure: all elements on the diagonal and below are 1, and those in successive upper side diagonals are $-1, +1, -1, \dots,$

$$\begin{aligned} s_{i,j} &= 1 \quad \text{if } j \le i, \\ s_{i,i+r} &= (-1)^r \quad \text{if } r = 1, 2, \dots, n-i. \end{aligned} \tag{2.11}$$

We call this the *canonical sign pattern*. Thus, in successive rows of $\mathbf{S}_n$, the number of sign changes diminishes by 1 from one row to the next, being $n-1$ in the first row, and 0 in the last. In particular, for $\nu = n$, by (2.5) and (2.3), since $p_{n-1}(\,\cdot\,; \mathbf{s}_n^{(1)}) \equiv 1$, we get the bound $M_{n,n}^{(1)} = 1 + \tau_n$, which is slightly smaller than the one for $k = 1$ in the continuous case (cf. (1.6)). But is $M_{n,n}^{(1)}$ the largest of all individual bounds $M_{n,\nu}^{(k)}$?

The graphs in Fig. 2, produced by the script[a] `plot_pnmk_leg20.m`, show the polynomials $p_{n-1}(\,\cdot\,; \mathbf{s}_\nu^{(1)})$ in the case $n = 20$, for $\nu = 1, 7, 13$, and 19.

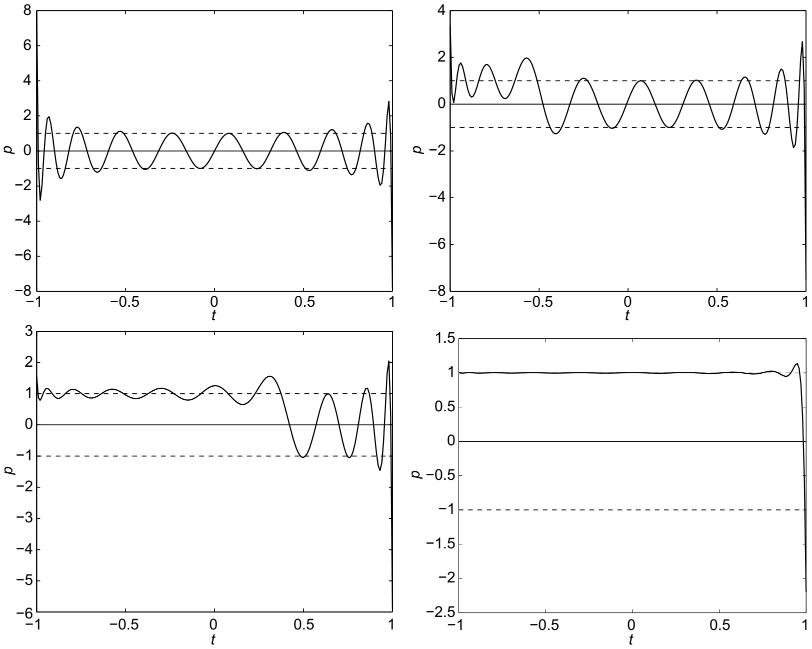


Fig. 2. The polynomial $p_{19}(\,\cdot\,; \mathbf{s}_\nu^{(1)})$ for $\nu = 1, 7, 13, 19$ (from top left to bottom right).

[a] All Matlab scripts referenced in this chapter are collected in [3].

As expected, the oscillations (about 0) diminish with increasing ν and also shift to the right. Combined with the fact that the interval of integration in (2.2) also increases, this contributes to the bounds $M_{n,\nu}^{(k)}$ likely to exhibit monotonic behavior as functions of ν.

Definition 2.2. The set $\mathbb{T}_n^{(k)}$ of $n-k+1$ support points τ_ν is said to have *Property* M_k ("M for "monotone") if

$$0 \leq M_{n,1}^{(k)} < M_{n,2}^{(k)} < \cdots < M_{n,n-k+1}^{(k)}. \tag{2.12}$$

A collection of sets $\{\mathbb{T}_n^{(k)}\}$, $k = 1, 2, \ldots, n$, is said to have *Property M*, if each set $\mathbb{T}_n^{(k)}$ has Property M_k.

An explicit criterion for the validity of Property M_1 can be given in the case $n = 2$, $k = 1$, using (2.5) and (2.3). One finds

$$M_{2,1}^{(1)} = (1 + \tau_1)\left[1 + \frac{1 + \tau_1}{\tau_2 - \tau_1}\right], \quad M_{2,2}^{(1)} = 1 + \tau_2,$$

so that Property M_1 holds precisely if

$$\frac{\tau_2 - \tau_1}{1 + \tau_1} > 1,$$

that is, if the interval $[\tau_1, \tau_2]$ is larger than the interval $[-1, \tau_1]$. In particular, it holds when $\tau_1 = -1$. Also note that Property M_n, involving only one bound, $M_{n,1}^{(n)}$, always holds trivially.

In the case where Property M_k holds, we may further distinguish between the largest bound $M_{n,n-k+1}^{(k)}$ to be normal or nonnormal according to the following definition.

Definition 2.3. When Property M_k holds, the largest bound $M_{n,n-k+1}^{(k)}$ is called *normal* if the associated sign-pattern vector $\mathbf{s}_{n-k+1}^{(k)}$ consists of 1s only, so that, by (2.5) and (2.3) with $\nu = n - k + 1$,

$$M_{n,n-k+1}^{(k)} = \frac{(\tau_{n-k+1} + 1)^k}{k!} \quad \text{(normal)}. \tag{2.13}$$

It is said to be *supnormal* (respectively, *subnormal*) if $M_{n,n-k+1}^{(k)}$ is larger (respectively smaller) than $(\tau_{n-k+1} + 1)^k/k!$.

The fine structure of the solution as defined by these concepts will be examined for concrete support points $\mathbb{T}_n^{(k)}$ by computation in Section 4.

3. Software

A main piece of software, in Matlab double precision, for computing the quantities $M_{n,\nu}^{(k)}$ is the routine Mnu.m, shown below, using (2.4). It first transforms the required integrals to integrals extended over $[-1, 1]$ before computing them (exactly) by $\lfloor(n+1)/2\rfloor$-point Gaussian quadrature. We also prepared an analogous symbolic routine sMnu.m that can be run in variable-precision arithmetic to check the accuracy of the double-precision answers. Both routines call on routines ellmu.m resp. sellmu.m evaluating the elementary Lagrange interpolation polynomials. A number of additional Matlab routines and scripts, which are identified in the proper context, have been written, making freely use of routines from the software package OPQ [2, Section 1], for example the routine gauss.m in the program displayed below. Among them is the routine pstar.m evaluating the extremal polynomial $p_{\nu,k}^{*}$ of (2.3).

Here, we list the principal routines Mnu.m and pstar.m.

```
function [M,S]=Mnu
%MNU The vector of the quantities in Eq. (1.9) and the
%sign-pattern matrix S of Eq. (2.10).
%  Given the (n-k+1)x1 array tau of support points, the
%  function [M,S]=Mnu generates the (n-k+1)x1 array M of
%  the quantities M_{n,1},M_{n,2},...,M_{n,n-k+1} in Eq. (2.4),
%  using floor((n+1)/2)-point Gaussian quadrature to (exactly)
%  compute the integrals in Eq. (2.4). The global variable ab0
%  is the N0x2 array, N0=floor((N+1)/2), of recurrence
%  coefficients for monic Legendre polynomials, useful when
%  the routine is run for many values of n<=N.

global n k tau ab0
M=zeros(n-k+1,1); S=zeros(n-k+1);
n0=floor((n+1)/2);
xw0=gauss(n0,ab0); x0=xw0(:,1); w0=xw0(:,2);
for nu=1:n-k+1
  L=0;
  for mu=1:n-k+1
    if tau(nu)==-1
      s=0;
    else
      t=.5*(tau(nu)-1+(tau(nu)+1)*x0); y=ellmu(mu,t);
      s=sum(w0.*(tau(nu)-t).^(k-1).*y);
    end
    if abs(s)<1e-14
      S(nu,mu)=0;
    elseif s>0
```

```
   S(nu,mu)=1;
  else
   S(nu,mu)=-1;
  end
  L=L+abs(s);
 end
 M(nu)=.5*(tau(nu)+1)*L/factorial(k-1);
end

function y=pstar(nu,t)
%PSTAR Extremal polynomial
%   Y=PSTAR(NU,T) evaluates the extremal polynomial p_{NU,K}^*(T),
%   given the (N-K+1)x1 array TAU of support points and the
%   (N-K+1)x(N-K+1) array S of sign patterns. The variable T has
%   to be a single value.

global n k tau S ab0
xw0=gauss(floor((n+1)/2),ab0); x0=xw0(:,1); w0=xw0(:,2);
sx=size(x0); t1=(t-1+(t+1)*x0)/2;
pnmk=zeros(sx);
for mu=1:n-k+1
 pnmk=pnmk+S(nu,mu)*ellmu(mu,t1);
end
y=sum(w0.*(t-t1).^(k-1).*pnmk);
y=(t+1)*y/(2*factorial(k-1));
```

4. Examples

4.1. Zeros of Jacobi polynomials

In this subsection, $\mathbb{T}_n^{(k)} = \{\tau_\nu\}$ will be the zeros of the Jacobi polynomial $P_{n-k+1}^{(\alpha,\beta)}$ of degree $n-k+1$ with parameters $\alpha > -1$, $\beta > -1$.

4.1.1. *Property M*

We first try to find out in which part of the (α,β)-plane Property M holds for $\{\mathbb{T}_n^{(k)}\}$. We do this by letting β, for fixed α, move upwards from $\beta = -0.9$ (where Property M holds) in steps of 0.1 until Property M fails for the first time. If this happens at $\beta = \beta_0$, we will have two values of β, $\beta_{\text{low}} = \beta_0 - 0.1$ and $\beta_{\text{high}} = \beta_0$, at the first of which Property M holds, and at the second of which it does not. We then apply a bisection type method to determine a more accurate value of β^* such that Property M holds for $\beta < \beta^*$ and fails to hold for $\beta > \beta^*$. If we do this for $\alpha = -0.9{:}0.1{:}5$ and plot β^* vs. α, we obtain a curve in the (α,β)-plane, $\beta = \beta^*(\alpha)$, $-0.9 \leq \alpha \leq 5$, below of which Property M holds, and above of which it does not. This

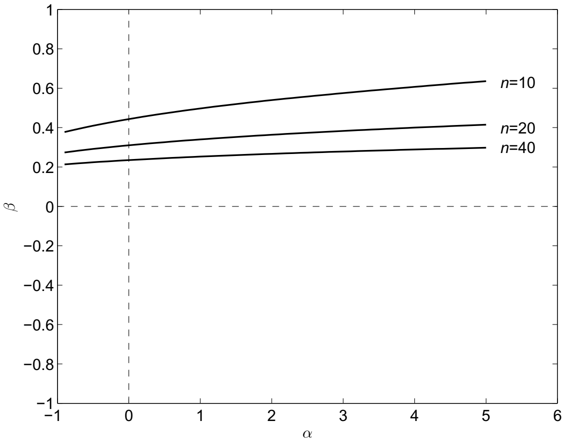


Fig. 3. Domains $\mathcal{D}_M$ for zeros of Jacobi polynomials

is implemented in the script `Mnu_test2.m` and the routine `PropM.m`, for $n = 10$, 20, and 40. The curves produced are shown in Fig. 3, generated by the script `plot_bestar.m`.

The domains $\mathcal{D}_M$ in the (α,β)-plane in which Property M holds are quite large, containing most likely the whole strip $-1 < \alpha < \infty$, $-1 < \beta \leq 0.15$. The important case $\alpha = \beta = 0$ of Legendre polynomials is well within this domain. It looks as if the upper boundary, when $n \to \infty$, would become a horizontal line at the height of about 0.15, an expectation that was reinforced when the script `Mnu_test2.m` was run with $n = 80$.

It may be thought that the discrete Markov problem yields solutions that are similar to those of the continuous problem, that is, producing bounds for $|p|$ nearly equal, or less than 2. The present discussion provides an opportunity to dispel this notion. As can be seen from Fig. 3, Property M holds in a domain that extends far (probably infinitely far) to the right and includes the positive real axis. So, let us look at the following exotic example.

Example 4.1. The case $\alpha = 20$, $\beta = 0$.

For simplicity, we take $k = 1$ and $n = 10$. The bounds $M^{(1)}_{10,\nu}$, produced by the script `Ex4_1.m`, are then as shown in Table 1. We can see that $M^{(1)}_{10,10}$ is anything but "nearly equal, or less than to 2"! (See also Example 4.2.)

Table 1. Bounds for $\alpha = 20$, $\beta = 0$.

ν	$M^{(1)}_{10,\nu}$
1	1.801846843051301e−02
2	5.842152063398374e−02
3	1.268621727178132e−01
4	2.226277805868339e−01
5	3.440754243906574e−01
6	4.890625971135910e−01
7	6.552254810765432e−01
8	1.088942247153418e+00
9	4.231399671654690e+00
10	4.323112886064376e+01

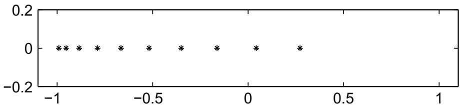


Fig. 4. Zeros of the Jacobi polynomial $P_{10}^{(20,0)}$.

The corresponding sign-pattern matrix is

$$\mathbf{S}_{10}^{(1)} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \end{bmatrix}.$$

Interestingly, it has the canonical ±1 Toeplitz pattern in the upper triangular part, but quite a few -1s in the lower triangular part. The support points τ_ν, plotted by the script **`plot_tau20.m`** and displayed in Fig. 4, are considerably "left-heavy" and therefore allow the interpolation polynomial $p_9(\,\cdot\,;\mathbf{s}_{10}^{(1)})$, plotted by the script **`plot_pnmk_jac10.m`**, in the

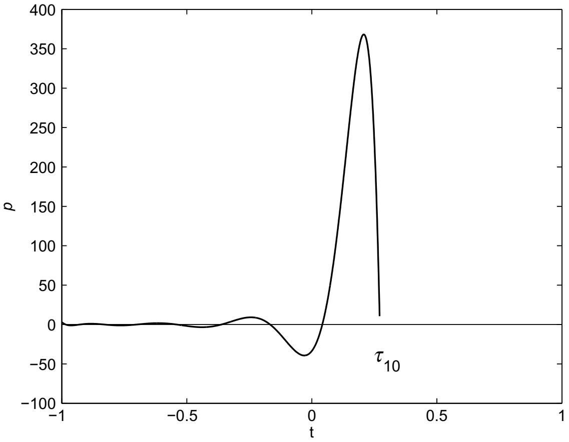


Fig. 5. The interpolation polynomial $p_9(\,\cdot\,;\mathbf{s}_{10}^{(1)})$ on $[-1,\tau_{10}]$.

last quarter of the interval $[-1,1]$ to unimpededly soar to great heights in absolute value. Just before $t=\tau_{10}$ it has a large hump as shown in Fig. 5, causing the last bound in Table 1 to be unusually large.

Returning to Fig. 3, what can be said about the bounds $M_{n,\nu}^{(k)}$ (for $n=10,20,40$) when β is above the respective curve in Fig. 3? The matter is then more complicated since the zeros of the Jacobi polynomials, when β becomes large, become increasingly "right-heavy" and, as a result, at least when $k=1$, the interpolation polynomial p_{n-1} in (2.3) is large (possibly very large) at the left end of the interval $[-1,1]$, regardless of the value of ν. This makes the bounds $M_{n,\nu}^{(1)}$ large for all ν, and the issue of monotonicity becomes blurred. There may no longer prevail any particular pattern with regard to the relative magnitudes of the bounds. This changes, however, when k becomes larger, because of the mitigating influence of the factor $(\tau_\nu-\tau)^{k-1}$ in (2.3). Indeed, already when $k=2$, the bounds are monotonically increasing for $n=10,20,40$, at least when $\beta\leq 10$, but very likely even beyond that.

When $k=1$, it is meaningful to ask for what values of α, β we have $M_{n,\nu}^{(1)}\leq M_{n,n}^{(1)}$ for all $1\leq\nu<n$. This was determined by the script `Mnu_test5.m` and the routine `PropMax.m` similarly as before and led to the curves (in red) of Fig. 6, below of which this maximum property holds. They were produced by the script `plot_bestarMax.m`. The figure, at the bottom, also reproduces the three curves of Fig. 3.

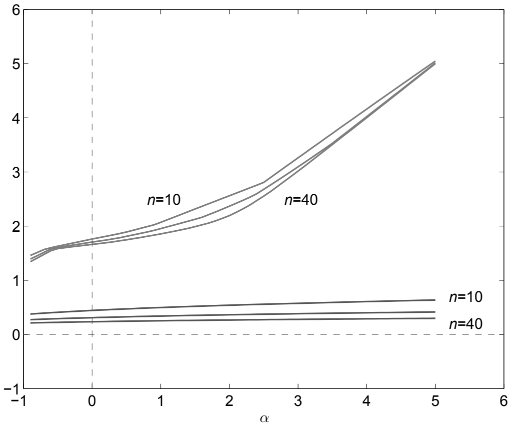


Fig. 6. Domains for zeros of Jacobi polynomials in which $M_{n,n}^{(1)}$ is the largest bound.

Here is another somewhat exotic example.

Example 4.2. The case $\alpha = 0$, $\beta = 5$ (above the red curves in Fig. 6) and $k = 1$.

Taking again $n = 10$, the script `Ex4_2.m` now obtains uniformly large bounds shown in Table 2 and the sign-pattern matrix

$$\mathbf{S}_{10}^{(1)} = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

As can be seen, the columns of $\mathbf{S}_{10}^{(1)}$ are alternately all 1s and all −1s. The support points τ_ν, plotted by the script `plot_tau5.m` and shown in Fig. 7, are now "right-heavy", as already mentioned, causing the interpolation

Table 2. Bounds for $\alpha = 0$, $\beta = 5$.

ν	$M_{10,\nu}^{(1)}$
1	9.451771476864684e+01
2	9.197235521181210e+01
3	9.282549856814146e+01
4	9.241442728995621e+01
5	9.263482633418995e+01
6	9.252393392561106e+01
7	9.256151952490204e+01
8	9.257534088982851e+01
9	9.252776896050197e+01
10	9.259039977486584e+01

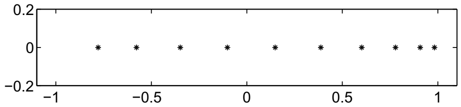


Fig. 7. Zeros of the Jacobi polynomial $P_{10}^{(0,5)}$.

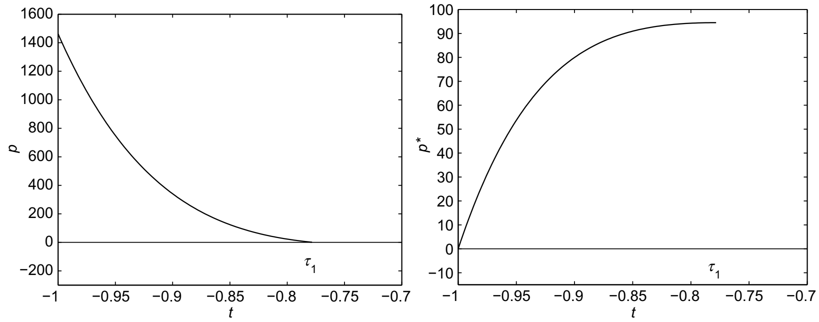


Fig. 8. The interpolation polynomial $p_9(\,\cdot\,;\mathbf{s}_\nu^{(1)})$ and extremal polynomial $p_{\nu,1}^*$ on $[-1,\tau_1]$.

polynomial $p_9(\,\cdot\,;\mathbf{s}_\nu^{(1)})$, which, by the way, is the same for each ν, to be very large at the left end of the interval $[-1,1]$, as shown on the left of Fig. 8. The extremal polynomial $p_{\nu,1}^*(t)$ (also the same for each ν), accordingly, raises quickly to a large number at $t = \tau_1$, see the graph on the right of Fig. 8, and stays at that level for the rest of the interval $[-1,1]$, which explains the uniformly large magnitude of all the bounds in Table 2. The graphs of Fig. 8 were produced in the script `Ex4_2.m`.

4.1.2. *Normality of* $M^{(k)}_{n,n-k+1}$ *for* $k = 1$

Now that we know of a fairly large domain $\mathcal{D}_M$ (cf. Fig. 3) of the (α,β)-plane in which Property M holds, i.e., in which the bounds $M^{(k)}_{n,1}, M^{(k)}_{n,2}, \ldots, M^{(k)}_{n,n-k+1}$ are monotonically increasing for each k, it is of interest to determine in which subdomain $\mathcal{D}_N$ the largest of these, $M^{(k)}_{n,n-k+1}$, is normal according to Definition 2.3 in Section 2. Because then, by (2.3) and (2.5), we will have

$$0 \leq M^{(k)}_{n,1} < M^{(k)}_{n,2} < \cdots < M^{(k)}_{n,n-k} < \frac{(1+\tau_{n-k+1})^k}{k!}, \quad 1 \leq k \leq n. \tag{4.1}$$

We analyze this only in the case $k = 1$, since for $k > 1$, normality of the last bound is more the exception than the rule.

Before we proceed, let us digress just a little. Normality of $M^{(1)}_{n,n}$, by definition, means that the sign-pattern vector $\mathbf{s}^{(1)}_n$ consists of all 1s, hence, by (2.2),

$$\int_{-1}^{\tau_n} \ell_\mu(\tau)\mathrm{d}\tau > 0, \quad \mu = 1, 2, \ldots, n. \tag{4.2}$$

Since τ_n, the largest zero of $P^{(\alpha,\beta)}_n$, is usually very close to 1, the inequality (4.2) is almost the same as $\int_{-1}^{1} \ell_\mu(\tau)\mathrm{d}\tau > 0$, which would mean that the Newton–Cotes quadrature formula on $[-1, 1]$ with the nodes being the zeros of the Jacobi polynomial $P^{(\alpha,\beta)}_n$ is positive. This is a property not easy to establish rigorously, but has been studied in the 1970s by R. Askey, and in the 1980s by G. Sottas (for references, see [1]). In particular, a conjecture of Askey, slightly revised in [1, Section 4.1], is positivity of the Newton–Cotes formulas in the (α, β)-domain $\mathcal{D}_{NC}$ shown in Fig. 9.

We have used our script `Mnu_test3.m` and routine `normal.m` to verify (4.2) for $n = 10, 20, 40$ on a grid on $\mathcal{D}_{NC}$ with spacing 0.01. We were successful in all cases except for various values of β near the left boundary $\alpha = -1$ (more precisely, when $\alpha = -0.99$).

Resuming our analysis of normality, we now proceed similarly as in Section 4.1.1, but search for fixed β, $-1 < \beta \leq 0.2$, for the alpha-value α^* for which normality of $M^{(1)}_{n,n}$ holds for all $\alpha < \alpha^*$, but does not hold for $\alpha > \alpha^*$. We can then develop plots of $\alpha = \alpha^*(\beta)$, or by reversing, plots of $\beta = \beta^*(\alpha)$, to delineate the domain $\mathcal{D}_N$. The results for $n = 10, 20, 40$, obtained by the script `Mnu_test3.m` and plotted by `plot_bestar_low.m`,

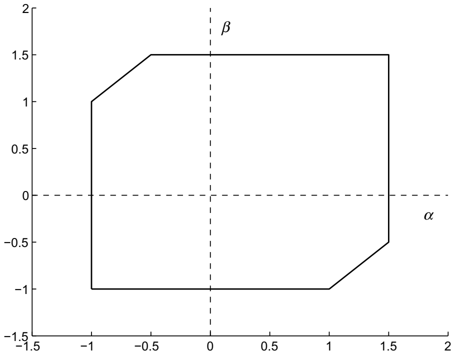


Fig. 9. Positivity domain $\mathcal{D}_{NC}$ for Newton–Cotes formulas with nodes equal to the zeros of the Jacobi polynomial $P_n^{(\alpha,\beta)}$.

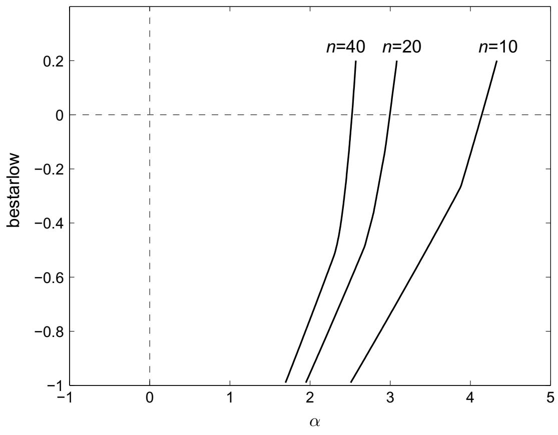


Fig. 10. Domains $\mathcal{D}_N$ for zeros of Jacobi polynomials when $k = 1$.

are shown in Fig. 10. Here, it appears that as $n \to \infty$, there might be a vertical limit boundary at about $\alpha = 1.3$. To the right of the curves, as was determined by the script `Mnu_test4.m`, the bound $M_{n,n}^{(1)}$ is consistently supnormal (cf. Definition 2.3 in Section 2), for α as large as 10, and probably remains so for all $\alpha > 10$.

By comparing the domains $\mathcal{D}_M$ of Fig. 3 with the domains $\mathcal{D}_N$ of Fig. 10, and taking approximate intersections, one can conclude with confidence that, when $k = 1$, the inequalities (4.1) hold in the rectangular domain

$-1 < \alpha \leq 1, -1 < \beta \leq 0.15$, for any value of n. In the rectangle $-1 < \alpha \leq 1$, $-1 < \beta \leq 0.5$, we used the script `Mnu_test1.m` to verify that (on a grid with spacing 0.05) the sign-pattern matrix $\mathbf{S}_{n-k+1}^{(k)}$, when $k = 1$, rather remarkably, is always canonical (cf. Section 2), at least for $n \leq 40$, but very likely for all $n \geq 2$.

4.1.3. *Monotonicity of $M_n^{(k)}$*

The script `Mnu_test6.m` was written to explore in what part of the (α, β)-plane the inequalities

$$M_n^{(1)} > M_n^{(2)} > \cdots > M_n^{(n)} \tag{4.3}$$

hold, where

$$M_n^{(k)} = \max_{1 \leq \nu \leq n-k+1} M_{n,\nu}^{(k)}.$$

It was found that, for $2 \leq n \leq 40$, the inequalities (4.3) are valid on a grid with spacing 0.05 within the square $Q = \{\alpha, \beta : |\alpha| < 0.5,\ |\beta| < 0.5\}$, suggesting that they may hold for all $\alpha, \beta \in Q$, perhaps even for all $n \geq 2$.

With regard to monotonicity in the variable n,

$$M_2^{(k)} < M_3^{(k)} < M_4^{(k)} < \cdots, \tag{4.4}$$

the script `Mnu_test7.m`, analogous to `Mnu_test6.m`, suggests that, at least for $n \leq 40$, the inequalities are valid on the square $-1 < \alpha, \beta \leq 1$ when $k = 1$, on the square $-0.5 \leq \alpha, \beta < 0.5$ when $k = 2$, and on $|\alpha|, |\beta| \leq 0.25$ when $k = 3$ and $n \geq 3$. They hold for still larger values of k and $n \geq k$, in sufficiently smaller (α, β)-domains, in particular for Legendre polynomials $(\alpha = \beta = 0)$, when $4 \leq k \leq 26$, but not when $k = 27$, where the inequality fails to hold for $n = 32, 33$.

4.2. Gauss–Lobatto quadrature points

In this subsection, $\mathbb{T}_n^{(k)} = \{\tau_\nu\}$ are the $(n - k + 1)$-point Gauss–Lobatto quadrature points relative to the Jacobi weight function when $n - k \geq 2$, the points $\tau_1 = -1$, $\tau_2 = 1$, when $n - k = 1$, and the point $\tau_1 = -1$, when $n - k = 0$. Results analogous to those in Section 4.1 can be obtained by making relatively minor changes to the software used before.

4.2.1. *Property M*

The scripts `Mnu_test2.m`, `plot_bestarlob.m` in combination with the routine `PropM.m`, adjusted to deal with Gauss–Lobatto points, yield Fig. 11,

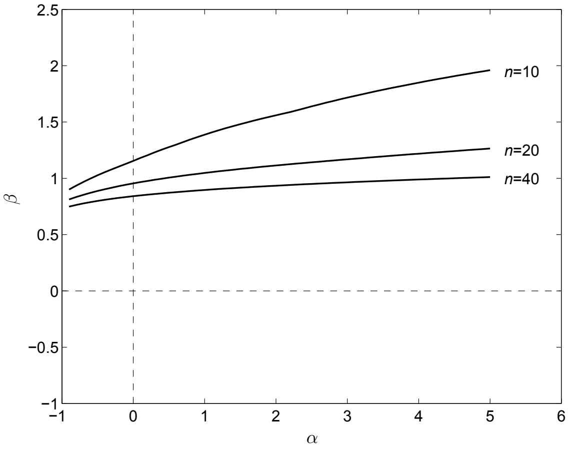


Fig. 11. Domains $\mathcal{D}_M$ for Gauss–Lobatto points.

showing the domains $\mathcal{D}_M$ (below the curves) in which Property M holds. They extend to β-values considerably larger than those in Fig. 3, but otherwise there is not much change.

The analogue of Fig. 6, for Gauss–Lobatto quadrature points, is Fig. 12, produced by the routine `plot_bestarlobMax.m` and the suitably adjusted script `Mnu_test5.m` and routine `PropMax.m`, showing the domains (below the red curves) in which $M_{n,\nu}^{(1)} \leq M_{n,n}^{(1)}$ for $\nu = 1, 2, \ldots, n-1$.

4.2.2. *Normality of* $M_{n,n-k+1}^{(k)}$ *for* $k = 1$

The analogue of Fig. 10 showing the domains $\mathcal{D}_N$ (to the left of the curves) in which $M_{n,n}^{(1)}$ is normal, is Fig. 13, produced by the routine `plot_bestarlob_low.m` and the suitably adjusted script `Mnu_test3.m` and routine `normal.m`.

Comparison of the domains $\mathcal{D}_M$ and $\mathcal{D}_N$ again allows us to conclude with confidence that the inequalities (4.1) with $k = 1$ hold for all $n \geq 2$ in the rectangular domain $-1 < \alpha \leq 1$, $-1 < \beta \leq 0.5$, except, possibly, when β is very close to -1.

The startling property of canonical sign-pattern matrices $\mathbf{S}_n^{(1)}$ that we noted at the end of Section 4.1.2 to hold for zeros of Jacobi polynomials does no longer hold for all n when the support points are Gauss–Lobatto

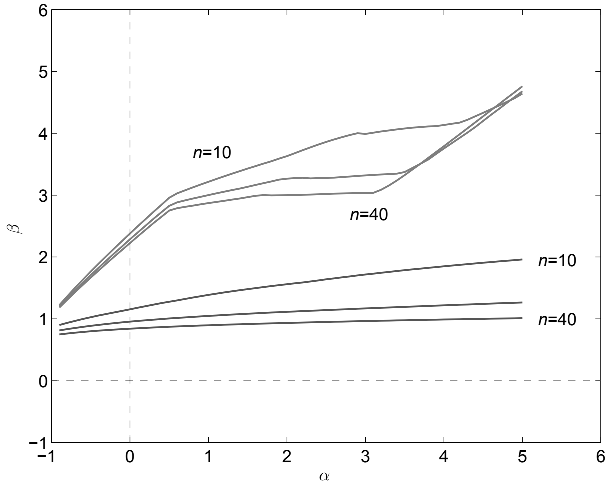


Fig. 12. Domains for Gauss–Lobatto points in which $M_{n,n}^{(1)}$ is the largest bound.

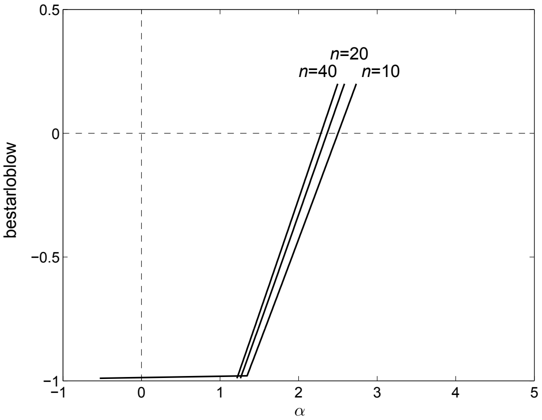


Fig. 13. Domains $\mathcal{D}_N$ for Gauss–Lobatto points when $k = 1$.

points, not even in the special case of Legendre polynomials. In that case, however, it does hold sporadically for selected values of n; those $\leq$100 are shown in Table 3. When it does not hold, the offending elements in the matrix $\mathbf{S}_n^{(1)}$ are consistently located in the first row, either in position 2 through n, or in position $1, 3, 5, \ldots, 2\lfloor(n+1)/2\rfloor - 1$.

Table 3. Values of $n \leq 100$ for which $\mathbf{S}_n^{(1)}$ is canonical for Gauss–Lobatto points in the case $\alpha = \beta = 0$.

3	10	17	23	35	39	46	61	70	76	84	92
5	12	19	25	36	40	52	64	73	80	87	94
6	13	20	30	37	41	56	68	74	82	89	
8	16	21	32	38	45	58	69	75	83	91	

4.3. Equally spaced points

4.3.1. *Equally spaced points on* $(-1, 1)$

In this subsection, $\mathbb{T}_n^{(k)}$ will be the points $\tau_\nu = -1 + 2\nu/(n - k + 2)$, $\nu = 1, 2, \ldots, n - k + 1$, equally spaced in the open interval $(-1, 1)$.

The following properties are found to hold: The canonical sign-pattern matrix $\mathbf{S}_{n-k+1}$ is the $(n - k + 1) \times (n - k + 1)$ matrix whose columns are alternately all 1s and all -1s (cf. also Example 4.2),

$$s_{i,j} = (-1)^{j-1}, \quad 1 \leq i, j \leq n - k + 1, \tag{4.5}$$

disregarding occasional zero elements. This was verified by the script `Mnu_test1_equal.m` and the routine `Prop_canon_equal.m` for all $n \leq 100$.

With regard to the bounds $M_{n,\nu}^{(k)}$, the properties are markedly different depending on whether k is equal to 1 or larger than 1.

In the case $k = 1$, the interpolation polynomial p_{n-1} in (2.3), which by the previous property is the same for each ν, is even if n is odd, and odd if n is even. As a consequence, by (2.5), we have

$$\begin{aligned} M_{n,\nu}^{(1)} &= M_{n,n+1-\nu}^{(1)}, \quad \nu = 1, 2, \ldots, n \quad (n \text{ even}), \\ M_{n,\nu}^{(1)} - M_{n,(n+1)/2}^{(1)} &= -[M_{n,n+1-\nu}^{(1)} - M_{n,(n+1)/2}^{(1)}], \\ &\qquad \nu = 1, 2, \ldots, n \quad (n \text{ odd}), \end{aligned} \tag{4.6}$$

that is symmetry when n is even, and anti-symmetry with respect to the middle bound $M_{n,(n+1)/2}^{(1)}$ when n is odd.

In the case $k > 1$, we consistently have Property M_k,

$$M_{n,1}^{(k)} < M_{n,2}^{(k)} < \cdots < M_{n,n-k+1}^{(k)} \quad (1 < k < n), \tag{4.7}$$

as was verified by the script `Mnu_test2_equal0.m` and routine `Prop_mon_equal.m` for all $n \leq 100$.

4.3.2. *Equally spaced points on* $[-1, 1]$

Here, $\mathbb{T}_n^{(k)}$ are the points $\tau_\nu = -1 + 2(\nu - 1)/(n - k)$, $\nu = 1, 2, \ldots, n - k + 1$, where $n \geq 2$, $1 \leq k \leq n - 1$, which are equally spaced in the closed interval $[-1, 1]$.

In this case, no canonical sign-pattern matrix $\mathbf{S}_{n-k+1}$ could be detected. It was found, however, by the script `Mnu_test2_equalC.m` and the same routine `Prop_mon_equal.m` as before, that Property M_k, that is (4.7), holds precisely if

$$(2 \leq n \leq 5 \text{ and } 1 \leq k \leq n - 1) \quad \text{or} \quad (n \geq 6 \text{ and } 2 \leq k \leq n - 1),$$

at least as long as $n \leq 100$, but very likely also for $n > 100$.

Acknowledgments

I am indebted to Mourad E. H. Ismail for bringing the work [4] of W. W. Hager *et al.* to my attention and to W. W. Hager for a critical reading of earlier drafts of the paper.

References

[1] W. Gautschi, Moments in quadrature problems, *Comput. Math. Appl.* **33** (1997) 105–118.

[2] W. Gautschi, *Orthogonal Polynomials in Matlab: Exercises and Solutions*, Software, Environments, and Tools (SIAM, 2016).

[3] W. Gautschi, Scripts for a discrete top-down Markov problem in approximation theory. *Purdue University Research Repository*, `doi:10.4231/R74Q7RX0`.

[4] W. W. Hager, H. Hou and A. V. Rao. Convergence rate for a Gauss collocation method applied to unconstrained optimal control, *J. Optim. Theory Appl.* **169**(3) (2016) 801–824.

[5] A. Shadrin, Twelve proofs of the Markov inequality, in *Approximation Theory: A Volume Dedicated to Borislav Bojanov* (Prof. M. Drinov Academic Publishing House, Sofia, 2004), pp. 233–298.

44

Papers on Orthogonal Polynomials on the Real Line

211 Polynomials orthogonal with respect to exponential integrals, Numer. Algorithms 70, 215–226 (2015)
215 Polynomials orthogonal with respect to cardinal B-spline weight functions, Numer. Algorithms 76, 1099–1107 (2017)
219 (with Gradimir V. Milovanović) Binet-type polynomials and their zeros, Electron. Trans. Numer. Anal. 50, 52–70 (2018)
223 (with Gradimir V. Milovanović) Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure, Numer. Algorithms 88, 1233–1249 (2021)
224 (with Gradimir V. Milovanović) Orthogonal polynomials relative to weight functions of Prudnikov type, Numer. Algorithms, to appear
225 Another look at polynomials orthogonal relative to exponential integral weight functions, Numer. Algorithms, to appear

44.1. [211] “Polynomials orthogonal with respect to exponential integrals”

[211] “Polynomials orthogonal with repect to exponential integrals”, *Numer. Algorithms* **70**, 215–226 (2015).

Numer Algor (2015) 70:215–226
DOI 10.1007/s11075-014-9943-8

ORIGINAL PAPER

Polynomials orthogonal with respect to exponential integrals

Walter Gautschi

Received: 12 May 2014 / Accepted: 23 November 2014 / Published online: 19 December 2014

Abstract Moment-based methods and related Matlab software are provided for generating orthogonal polynomials and associated Gaussian quadrature rules having as weight function the exponential integral E_ν of arbitrary positive order ν supported on the positive real line or on a finite interval $[0, c]$, $c > 0$. By using the symbolic capabilities of Matlab, allowing for variable-precision arithmetic, the codes provided can be used to obtain as many of the recurrence coefficients for the orthogonal polynomials as desired, to any given accuracy, by choosing d-digit arithmetic with d large enough to compensate for the underlying ill-conditioning.

Keywords Orthogonal polynomials · Exponential integrals · Chebyshev algorithm · Matlab software

Mathematics Subject Classification (2010) 33C47 · 65D30

1 Introduction

Integrals having as weight function the exponential integral $E_\nu(x) = \int_1^\infty e^{-xt}\mathrm{d}t/t^\nu$, $\nu > 0$, on the positive real line, $x \in \mathbb{R}_+$, or on a finite interval, $x \in [0, c]$, $c > 0$, are of interest in radiative transfer when $\nu = m \geq 1$ is an integer. Early attempts by A. Reiz to compute them by means of 2- and 3-point Gaussian quadrature are reported on, and tables to 3–4 digits provided in, the monograph [2, §23] of 1950. More accurate values to 7 digits, for $m = 1 : 5$, also for 4-point quadratures (if $m \leq 3$), are published in [10]. The case of finite intervals $[0, c]$, $c > 0$,

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

was dealt with in [9], where 3- and 4-point Gaussian quadrature rules are tabulated for $m = 1$, $m = 2$, and $c = .1 : .1 : 1.0, 1.2 : .2 : 4.0, 4.5 : .5 : 8.0, 9 : 15$ to 4–6 decimal digits. Further progress has been made in the late 1960s and early 1970s. In [4, p. 436], 8-digit values were published for the 20-point Gaussian quadrature rule in the case $m = 1$ and support interval $[0, \infty]$. (Some of the terminal figures are in error by one or a few units, and two terminal figures in the case of the smallest node.) They were obtained from the respective orthogonal polynomial, generated by the author's discretization procedure. Ad hoc methods were used in [3] to compute the same 20-point formula to 12 (correct) digits; see Table 4 in the cited reference. To the best of our knowledge, no further work on this topic has been published since.

In the present paper we take advantage of the intervening progress made in the constructive theory of orthogonal polynomials, especially the development of variable-precision codes (for example, see [5, Ch. 2; 6–8]), to compute as many of the respective orthogonal polynomials as desired, to arbitrarily high accuracy, and for arbitrary positive values of ν. Once the first n recurrence coefficients of these orthogonal polynomials have been obtained, to whatever accuracy, this immediately provides access to all corresponding k-point Gaussian quadrature rules, $1 \le k \le n$, to the same accuracy.

2 The weight function E_ν, $\nu > 0$, on $[0, \infty]$

Our approach is to compute the first n orthogonal polynomials (that is, their recurrence coefficients) from the first $2n$ moments of the weight function. It is known (cf. [5, §2.1]) that the underlying numerical condition of the process is typically very bad. We determine estimates of the respective condition numbers empirically as a function of n and ν, and use this information to choose the number of digits in variable-precision arithmetic that are needed to compensate for the loss of accuracy due to ill-conditioning.

The software we need comes from two packages of Matlab programs, the package `OPQ` for ordinary Matlab double-precision computation, and the symbolic Matlab package `SOPQ` for variable-precision computation. The former can be downloaded from (http://dx.doi.org/10.4231/R7959FHP), and the latter from the file `SOPQ.html` in (https://www.cs.purdue.edu/archives/2002/wxg/codes/). All other Matlab routines and files referenced in this paper reside on the same website (https://www.cs.purdue.edu/archives/2002/wxg/codes/) in the file `POEXPINT`.

The `OPQ` routine `chebyshev.m` resp. the `SOPQ` routine `schebyshev.m` implements the Chebyshev algorithm (cf. [5, §2.1.7]) in ordinary Matlab double precision resp. symbolic/variable precision. It uses the moments (or modified moments) of the weight function to generate the recurrence coefficients α_k, β_k of the corresponding (monic) orthogonal polynomials π_k,

$$\begin{aligned}\pi_0(t) &= 1, \quad \pi_{-1} = 0,\\ \pi_{k+1}(t) &= (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0, 1, 2, \ldots .\end{aligned} \tag{2.1}$$

For the weight function $E_\nu(x)$, $\nu > 0$, on $[0, \infty]$, the moments are especially simple,

$$\begin{aligned}\mu_k &= \int_0^\infty t^k E_\nu(t)\mathrm{d}t = \int_0^\infty t^k \mathrm{d}t \int_1^\infty \frac{e^{-tu}}{u^\nu}\,\mathrm{d}u \\ &= \int_1^\infty \frac{\mathrm{d}u}{u^\nu}\int_0^\infty t^k e^{-ut}\mathrm{d}t = \int_1^\infty \frac{\mathrm{d}u}{u^\nu}\int_0^\infty \left(\frac{\tau}{u}\right)^k e^{-\tau}\frac{\mathrm{d}\tau}{u} \\ &= k!\int_1^\infty \frac{\mathrm{d}u}{u^{\nu+k+1}}\,,\end{aligned}$$

that is,

$$\mu_k = \frac{k!}{\nu+k}\,, \quad k = 0, 1, 2, \ldots\,, \tag{2.2}$$

or, evaluated recursively,

$$\begin{aligned}\mu_k &= k\frac{\nu+k-1}{\nu+k}\,\mu_{k-1}, \quad k = 1, 2, \ldots\,, \\ \mu_0 &= \frac{1}{\nu}\,.\end{aligned} \tag{2.3}$$

This is implemented in the symbolic Matlab routine smomEnu.m,

```
%SMOMENU
%
function mom=smomEnu(dig,N,nu)
digits(dig);
mom=vpa(zeros(2*N,1)); snu=vpa(nu);
mom(1)=1/snu;
for k=1:2*N-1
  mom(k+1)=(k*(snu+k-1)/(snu+k))*mom(k);
end
```

The routine sr_Enu.m can then be used to produce in dig-digit arithmetic the desired recurrence coefficients.

```
%SR_ENU
%
function ab=sr_Enu(dig,N,nu)
digits(dig);
mom=smomEnu(dig,N,nu);
ab=schebyshev(dig,N,mom');
%ab
```

The first N coefficients $\alpha_k, \beta_k, k = 0, 1, \ldots, N-1$, are returned in the first, resp. second, column of the $N \times 2$ array ab. The double-precision analogue of these routines are momEnu.m and r_Enu.m.

Insight into the conditioning of the process can be obtained by comparing dig0-digit results with dig1-digit results, dig1≫dig0. If the absolute value of the relative discrepancies are divided by $.5 \times 10^{\text{-dig0}}$, this gives fairly reliable estimates of the condition numbers involved. The routine `condEnu.m`,

```
%CONDENU
% function [cond,lcond]=condEnu(N,nu)
f0='%8.0f %12.4e %12.4e\n';
dig0=32; dig1=48;
cond=zeros(N,1); lcond=zeros(N,1);
tic
ab0=sr_Enu(dig0,N,nu);
ab1=sr_Enu(dig1,N,nu);
toc
err=subs(abs((ab0-ab1)./ab1));
eps0=.5*10^(-dig0); c=err/eps0;
for n=1:N
  cond(n)=max(c(n,1),c(n,2));
  lcond(n)=log10(cond(n));
% fprintf(f0,n,cond(n),lcond(n))
end
plot((2:N)',lcond(2:N));set(gca,'FontSize',14)
axis([0 42 -10 30])
hold on
```

computes condition numbers (`cond`) and their base-10 logarithms (`lcond`) for $n = 1 : N$ and $\nu > 0$. With `dig0`=32, `dig1`=48, $N = 40$, and $\nu = m = 1 : 5$, it produces condition numbers whose logarithm to base 10 are plotted in Fig. 1. It can be seen that they are relatively insensitive to the particular values of m. For $n = 40$, one must expect a condition number of the order 10^{28}.

Suppose now that we want the first 40 recurrence coefficients accurate to 32 decimal digits. According to Fig. 1, we would need a working precision of about $28 + 32 = 60$ digits. Indeed, with `dig=60`, the maximum relative errors in the recurrence coefficients, as determined by `run_sr_Enu.m`,

```
%RUN_SR_ENU
%
f0='%8.0f %12.4e\n';
N=40;
disp('       m      err')
for m=1:5
  ab1=sr_Enu(60,N,m);
  ab2=sr_Enu(68,N,m);
  err=max(max(subs(abs((ab1-ab2)./ab2))));
  fprintf(f0,m,err)
end
%vpa(ab1,32)
%xw=sgauss(60,10,ab1(1:10,:))
%xw=sgauss(60,20,ab1(1:20,:))
```

are as follows,

m	maxerr
1	4.31(–34)
2	4.37(–33)
3	6.27(–34)
4	1.03(–33)
5	8.34(–34)

which should be sufficient to get 32 correct significant digits.

When $m = 1$, the result for `ab1`, rounded to 32 significant digits, is shown below.

```
ab1 =
[ 0.5,                                 1.0                              ]
[ 2.3,                                 0.41666666666666666666666666666667]
[ 4.2346938775510204081632653061224,   2.6133333333333333333333333333333]
[ 6.1991728899520653289533363162975,   6.7549479383590170762182423990004]
[ 8.1760287637692264292699317909546,  12.870010159219961694306927947982]
[ 10.159422455949061212251865164385,  20.968994696423940837568730033392]
[ 12.146763033936533447023591073777,  31.057006630936801273572544890070]
[ 14.136701121356747300993633150615,  43.136954134511657222301719712458]
[ 16.128455652762927469396958109060,  57.210668436970122333167134416654]
[ 18.121539082321841623319665520066,  73.279386245227974471651536115340]
[ 20.115629289875337430432917374549,  91.343987098052228799027821345184]
[ 22.110503786849051613944146941994, 111.40512190727557194747483078212]
[ 24.106003312846896348414485714247, 133.46328782636285119890864794111]
[ 26.102010469568827217617261947703, 157.51887442601060579760842747446]
[ 28.098436563081088211003295397313, 183.57219352081384881254732894186]
[ 30.095213170671117434692239911288, 211.62349918475516842026740943688]
[ 32.092286550163055045719930642234, 241.67300161999396798681139359024]
[ 34.089613824796098217389455425773, 273.72087703234213807690738866538]
[ 36.087160313621981482166959580906, 307.76727483055256287082400203899]
[ 38.084897621989801207054976553018, 343.81232298324062012406026193200]
[ 40.082802248959441625185033380225, 381.85613207730455797274502171017]
[ 42.080854554029524135012861596701, 421.89879844200106929659826020412]
[ 44.079037978530591076723802015299, 463.94040658821755047472100884584]
[ 46.077338450692653888311066112066, 507.98103113750578852555793347317]
[ 48.075743925290319796558902187815, 554.02073836527156158227868335729]
[ 50.074244023312606062481483917045, 602.05958744825363192208568506608]
[ 52.072829746951336550343267893564, 652.09763148259506560518308426530]
[ 54.071493251985063900848805331302, 704.13491832195542993299333320163]
[ 56.070227664382534174913972818239, 758.17149127300924746037495398821]
[ 58.069026931320505158868382006277, 814.20738967686282687226088956636]
[ 60.067885699236479946825157355040, 872.24264939842125436311466446771]
[ 62.066799213304285759238786765347, 932.27730324088594660987318727380]
[ 64.065763234022944743541101952352, 994.31138129890265056912092917402]
[ 66.064773967579491409986408284564, 1058.3449112610896454434310771559]
[ 68.063828007376281312836671975265, 1124.3779186705290316748905178572]
[ 70.062922284667550360700504894342, 1192.4104271501374685786630141720]
[ 72.062054026674477715769790223624, 1262.4424585985283766786742605580]
[ 74.061220720875801808371126074796, 1334.4740333609488353760830067157]
[ 76.060420084426113329606309812572, 1408.5051703790570651814974233587]
[ 78.059650037853873322599060693536, 1484.5358873226525872793900026879]

>>
```

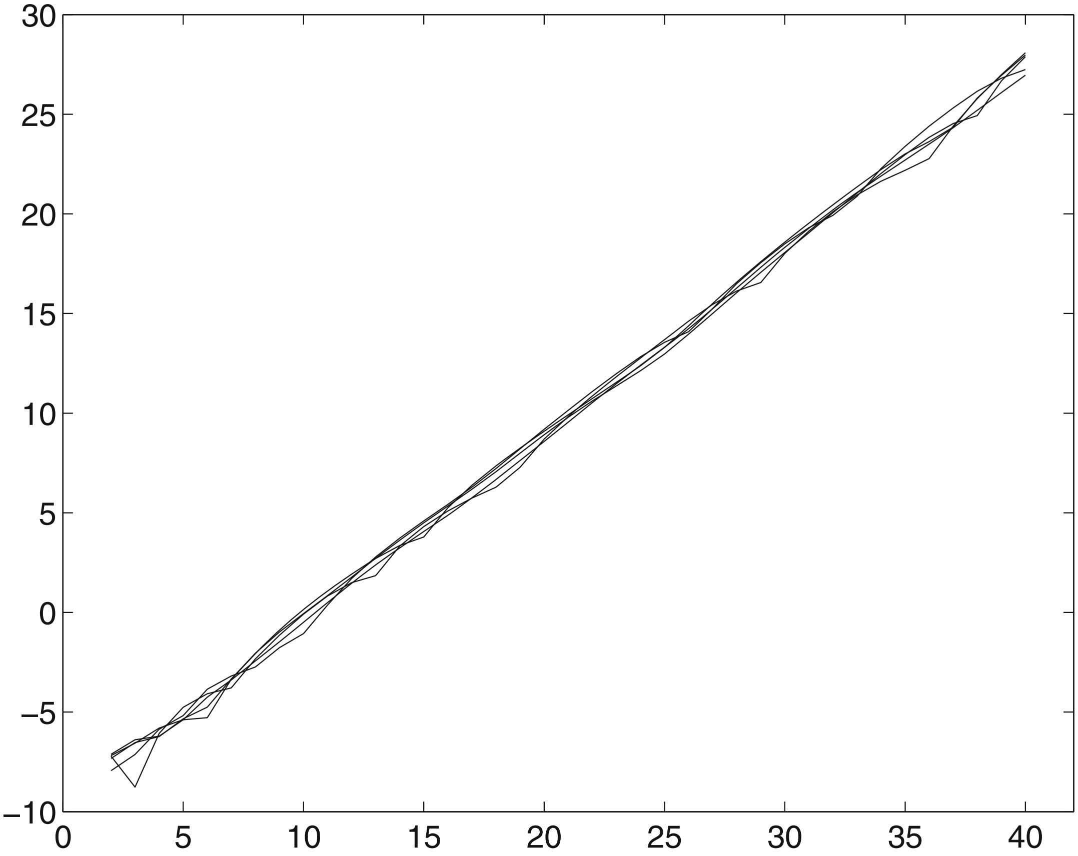


Fig. 1 Condition numbers for $E_m(x)$, $m = 1 : 5$ on $[0, \infty]$

It took 34 seconds to produce (on an Intel Pentium D 830 3GHz Dual core processor with 1GB DDR2 non-ECC SDRAM 2DIM memory). The first 12 lines in the output, incidentally, confirm the 12-digit results in Table 3 of [3], and also, if used in the `SOPQ` routine `sgauss.m`, those in Table 4 except for two end-figure errors of one unit.

In order to check the 7-digit results in [10] (for the N-point Gaussian quadrature rules, $N \leq 4$), ordinary Matlab double precision is more than enough. This is done by the routine `Reiz.m`, which calls on the `OPQ` routine `gauss.m`,

```
%REIZ
%
N=4;
for m=1:5
  fprintf('                         m=%1.0f\n',m)
  ab=r_Enu(N,m);
  for n=2:N
    xw=gauss(n,ab)
    fprintf('\n')
  end
end
```

There is agreement with the results in the cited reference except for occasional end-figure discrepancies of one unit.

We could have applied Chebyshev's algorithm also with *modified* moments,

$$\mu_k = \int_0^\infty p_k(t) E_\nu(t) \mathrm{d}t, \quad k = 0, 1, 2, \dots ,$$

for example with p_k being Laguerre polynomials. It was found, however, that the procedure involved is considerably slower and moreover, surprisingly, the underlying conditioning worse than in the case of ordinary moments.

3 The weight function E_ν, $\nu > 0$, on $[0, c]$, $c > 0$

In the case of a finite interval $[0, c]$, the moments are

$$\mu_k = \int_0^c t^k E_\nu(t) \mathrm{d}t, \quad k = 0, 1, 2, \dots .$$

By elementary manipulations, using the definition of E_ν and an interchange of integrals, one finds

$$\mu_k = \int_1^\infty t^{-(\nu+k+1)} \gamma(k+1, ct)\, \mathrm{d}t, \tag{3.1}$$

where $\gamma(a, x)$ is the incomplete gamma function

$$\gamma(a, x) = \int_0^x t^{a-1} e^{-t} \mathrm{d}t.$$

Integration by parts in (3.1), integrating the first factor and differentiating the second, yields

$$\mu_k = \frac{1}{\nu + k} [\gamma(k+1, c) + c^{\nu+k}\, \Gamma(1-\nu, c)], \quad k = 0, 1, 2, \dots , \tag{3.2}$$

where

$$\Gamma(a, x) = \int_x^\infty t^{a-1} e^{-t} \mathrm{d}t.$$

For large c, we have $\gamma(k+1, c) \sim k!$ and $\Gamma(1-\nu, c) \sim c^{-\nu} e^{-c}$ (cf. [1, 6.5.32]), so that, for fixed k,

$$\mu_k \sim \frac{1}{\nu + k} (k! + c^k e^{-c}), \quad c \to \infty,$$

recovering (2.2) in the limit as $c \to \infty$.

In order to compute μ_k from (3.2) in variable-precision arithmetic, one needs symbolic routines for incomplete gamma functions. The Matlab Symbolic Toolbox has a routine for $\Gamma(a, x)$, but currently none for $\gamma(a, x)$. While the latter function could be obtained in terms of the former, this would entail, in the context of (3.2), very large cancellation errors, especially when k is large. To avoid this, we wrote our own symbolic routine `sgammastar.m` for Tricomi's form $\gamma^*(a, x)$ of $\gamma(a, x)$,

```
%SGAMMASTAR Symbolic counterpart of gammastar.m
%
function y=sgammastar(dig,a,x)
syms y y0 t
digits(dig)
eps0=.5*10^(-dig);
a=vpa(a,dig); x=vpa(x,dig);
if rem(subs(a),1)==0 & subs(a)<0
  y=x^(-subs(a));
  return
end
t=vpa(1/gamma(a+1),dig); y0=vpa(0); y=t; k=0;
while subs(abs((y-y0)/y))>eps0
  k=k+1; y0=y; t=vpa(x*t/(a+k),dig); y=y0+t;
end
y=vpa(exp(-x),dig)*y;
```

which is an entire function in both a and x. It uses the power series ([1, 6.5.29])

$$\gamma^*(a, x) = e^{-x} \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(a + k + 1)},$$

which allows a very stable and rapid evaluation when $x > 0$ (but not excessively large) and $a \geq 0$, all terms of the series being positive. The function $\gamma(a, x)$ can then be obtained from $\gamma^*(a, x)$ by $\gamma(a, x) = \Gamma(a)x^a\gamma^*(a, x)$.

The resulting symbolic routine for μ_k, $k = 0, 1, \ldots, 2n - 1$, is `smomEnufin.m`.

```
%SMOMENUFIN Symbolic counterpart of momEnufin.m
%
function mom=smomEnufin(dig,n,nu,c)
digits(dig);
c=vpa(c,dig);
mom=vpa(zeros(2*n,1));
y=feval(symengine,'igamma',1-nu,c);
y=vpa(y,dig);
for k=1:2*n
  y1=gamma(vpa(k,dig))*c^k*sgammastar(dig,k,c);
  mom(k)=(y1+c^(nu+k-1)*y)/vpa(nu+k-1,dig);
end
```

There is, however, again the question of conditioning. We can estimate condition numbers similarly as in Section 2 by a routine named `condEnufin.m`. It uses a routine `sr_Enufin.m` analogous to the routine `sr_Enu` of Section 2.

```
     %CONDENUFIN
     %
    function [cond,lcond]=condEnufin(N,nu,c)
    f0='%8.0f %12.4e %12.4e\n';
    dig0=64; dig1=120;
    cond=zeros(N,1); lcond=zeros(N,1);
    tic
    ab0=sr_Enufin(dig0,N,nu,c);
    ab1=sr_Enufin(dig1,N,nu,c);
    toc
    err=subs(abs((ab0-ab1)./ab1));
    eps0=.5*10^(-dig0); c=err/eps0;
    for n=1:N
      cond(n)=max(c(n,1),c(n,2));
      lcond(n)=log10(cond(n));
%      fprintf(f0,n,cond(n),lcond(n))
    end
    plot((2:N)',lcond(2:N));set(gca,'FontSize',14)
    axis([0 42 -10 60])
```

In the routine `run_condEnufin.m`,

```
%RUN_CONDENUFIN
%
N=40; m=1;
for c=[.1 .5 1 2 4 8 16 32 64 128]
  if c>.1&m==1, hold on; end
  condEnufin(N,m,c);
end
```

for each $m = 1 : 5$ we ran it with $N = 40$, $c = [.1\ .5\ 1\ 2\ 4\ 8\ 16\ 32\ 64\ 128]$ and plotted the graphs analogous to the ones in Fig. 1 superimposed on each other; they are shown in Fig. 2 for for $m = 1$ (on top) and $m = 5$ (at the bottom).

As in the case $c = \infty$ (cf. Fig. 1), it is evident that the condition numbers are relatively insensitive to the choice of m. For moderately large values of c (say, $0 < c \leq 16$), and $n = 40$, the condition number is of the order 10^{53}, suggesting that, to obtain results to 32 correct digits, the number d of digits to be used should be about 85. As c becomes larger, the condition numbers slowly approach the ones for $c = \infty$, being of the order 10^{46} when $c = 32$, of the order 10^{39} when $c = 64$, and of the order 10^{27} when $c = 128$.

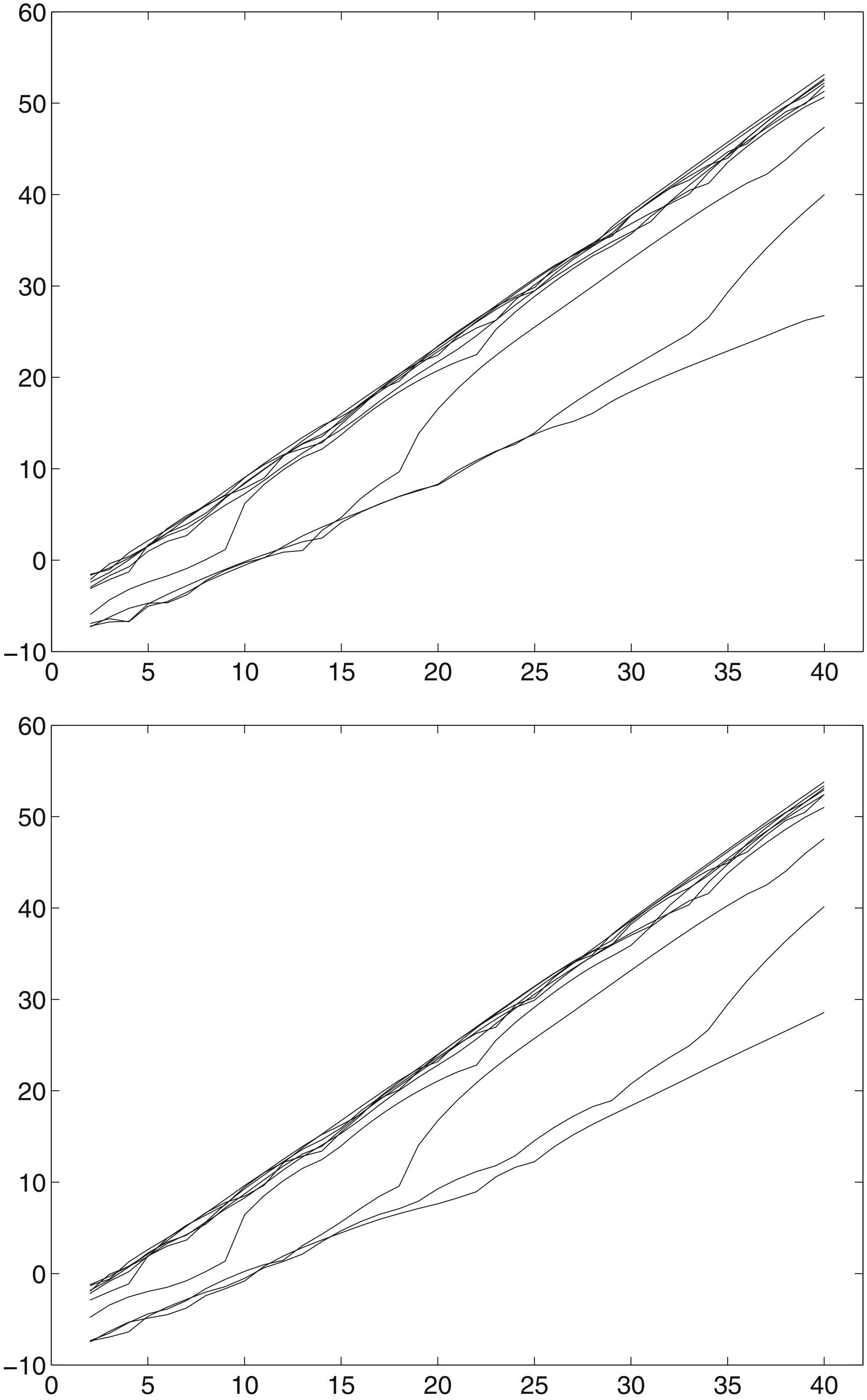


Fig. 2 Condition numbers for $E_m(x)$, $m = 1$ and $m = 5$, on $[0, c]$, $c = [.1\ .5\ 1\ 2\ 4\ 8\ 16\ 32\ 64\ 128]$

Actually, the routine `run_sr_Enufin.m`,

```
%RUN_SR_ENUFIN
%
N=40; m=1; c=.1;
tic
ab1=sr_Enufin(82,N,m,c);
ab2=sr_Enufin(90,N,m,c);
toc
err=max(subs(abs((ab1-ab2)./ab2)))
%ab1
```

can be used to determine by trial and error the exact number of digits d needed. In the first call to the routine `sr_Enufin.m`, the number of digits (the first input argument) is guessed (82 in the above display for $m = 1$, $c = .1$), and then adjusted, either upward or downward, in accordance with the maximum error produced. In the second call to `sr_Enufin.m`, the number of digits is kept fixed (at 90). The results agree very closely with the estimates given above: for the values $c = [16\ 32\ 64\ 128]$ (and $N = 40$), the required number d of digits is shown in Table 1 for $m = 1 : 5$.

Note also that the results are pracically independent of m in the range $1 \le m \le 5$.

To conclude, we use the routine `Kegel.m`, calling on the double-precision routine `r_Enufin.m` and the `OPQ` routine `gauss.m`,

```
%KEGEL
%
f0='%8.0f %5.1f %12.8f %12.8f %12.8f\n';
f1='%27.8f %12.8f %12.8f\n';
f2='%8.0f %5.1f %12.8f %12.8f %12.8f %12.8f\n';
f3='%27.8f %12.8f %12.8f %12.8f\n';
disp('    m  c         nodes/weights')
for m=1:2
  for c=[.1:.1:1 1.2:.2:4 4.5:.5:8 9:15]
    ab=r_Enufin(4,m,c);
    xw=gauss(3,ab); x=xw(:,1); w=xw(:,2);
    fprintf(f0,m,c,x(1),x(2),x(3))
    fprintf(f1,w(1),w(2),w(3))
    xw=gauss(4,ab); x=xw(:,1); w=xw(:,2);
    fprintf(f2,m,c,x(1),x(2),x(3),x(4))
    fprintf(f3,w(1),w(2),w(3),w(4))
    fprintf('\n')
  end
end
```

to reproduce Tables 1–4 of [9] (for the N-point Gaussian quadrature rules, $N \le 4$). Spotchecking the output supports the author's claim of accuracy up to one or two units in the last decimal digit.

Table 1 Number d of digits needed to get results to 32 correct decimal digits

m	c	d	m	c	d	m	c	d	m	c	d	m	c	d
1	16	83	2	16	84	3	16	84	4	16	84	5	16	84
	32	79		32	79		32	79		32	79		32	79
	64	72		64	72		64	72		64	72		64	72
	128	61		128	61		128	60		128	60		128	60

References

1. Abramowitz, M., Stegun, I.A. (eds.): Handbook of Mathematical Functions with Formulas, Graphs, and Marthematical Tables, National Bureau of Standards, Appl. Math. Ser. 55, U. S. Government Printing Office, Washington. D. C. (1964)
2. Chandrasekhar, S.: Radiative Transfer, the International Series of Monographs on Physics. Oxford Univeristy Press, Oxford (1950)
3. Danloy, B.: Numerical construction of Gaussian quadrature formulas for $\int_0^1 (-\mathrm{Log}\, x) \cdot x^\alpha \cdot f(x) \cdot dx$ and $\int_0^\infty E_m(x) \cdot f(x) \cdot dx$. Math. Comp. **27**, 861–869 (1973)
4. Gautschi, W.: Algorithm 331—Gaussian quadrature formulas. Comm. ACM **11**, 432–436 (1968)
5. Gautschi, W.: Orthogonal Polynomials: Computation and Approximation, Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
6. Gautschi, W.: Variable-precision recurrence coefficients for nonstandard orthogonal polynomials. Numer. Algorithms **52**, 409–418 (2009). Also in Selected Works, vol. 2, 266–275
7. Gautschi, W.: Sub-range Jacobi polynomials. Numer. Algorithms **61**, 275–290 (2012). Also in Selected Works, vol. 2, 277–285
8. Gautschi, W.: Repeated modifications of orthogonal polynomials by linear divisors. Numer. Algorithms **63**, 369–383 (2013). Also in Selected Works, vol. 2, 287–301
9. Kegel, W.H.: Zur numerischen Berechnung der Integrale $\int_0^\tau f(x)K_n(x)dx$, Z. Astrophys **54**, 34–40 (1962)
10. Reiz, A.: Quadrature formulae for the numerical calculation of mean intensities and fluxes in a stellar atmosphere. Arkiv Astronomi **1**, 147–153 (1950)

44.2. [215] “Polynomials orthogonal with respect to cardinal B-spline weight functions”

[215] “Polynomials orthogonal with respect to cardinal B-spline weight functions”, *Numer. Algorithms* **76**, 1099–1107 (2017).

Numer Algor (2017) 76:1099–1107
DOI 10.1007/s11075-017-0298-9

ORIGINAL PAPER

Polynomials orthogonal with respect to cardinal B-spline weight functions

Walter Gautschi[1]

Received: 1 November 2016 / Accepted: 19 February 2017 / Published online: 28 February 2017

Abstract A stable and efficient discretization procedure is developed to compute the recurrence coefficients for orthogonal polynomials whose weight function is a polynomial cardinal spline of order $m \geq 1$. The procedure is compared with a symbolic moment-based method developed recently by G. V. Milovanović. Numerical examples are provided for illustration.

Keywords Orthogonal polynomial · B-spline weight function · Recurrence coefficients

1 Introduction

In the interesting paper [4], G.V. Milovanović developed techniques and software in Mathematica to obtain in symbolic form the first 60 recurrence coefficients for polynomials orthogonal on $[0, m]$ with respect to the cardinal B-spline weight function of order m. The method used is based on the moments of the weight function, from which the recurrence coefficients of the orthogonal polynomials are obtained by the classical Chebyshev algorithm [1, §2.1.7]. We implement the same approach in symbolic Matlab, which allows us to generate in variable-precision arithmetic as many recurrence coefficients as desired, using sufficiently many working digits. To obtain, for example, the first 100 pairs of coefficients to an accuracy of 16 digits, the number of working digits required is 164, essentially regardless of the value of m (at least in the range $2 \leq m \leq 20$ and probably beyond). This large number of digits is due to

✉ Walter Gautschi
wgautschi@purdue.edu

[1] Purdue University, West Lafayette, IN 47907, USA

the highly ill-conditioned nature of the mapping from the first 200 moments to the first 100 pairs of recurrence coefficients. More details are provided in Section 2.

The problem, actually, may be viewed as a textbook example for another approach based on multicomponent discretization. The cardinal B-spline of order m consists of m different polynomial pieces of degree $m-1$ supported on the m intervals $I_k=[k-1,k]$, $k=1,2,\ldots,m$. The pieces are joined together to form a piecewise polynomial function of smoothness class $\mathbb{C}^{m-2}[\mathbb{R}]$. The intervals I_k are taken to be the m component intervals of the multicomponent discretization method [1, §2.2.4], which will produce exact results for the first n recurrence coefficients if the discretization on each interval I_k is carried out by $(n+\lceil m/2\rceil-1)$-point Gauss quadrature. Moreover, the method is quite stable if a stable method is used to evaluate the B-spline on each interval I_k. The fact that ordinary Matlab double precision yields highly accurate results at a cost incomparably lower than the one involved in the moment-based approach requiring variable-precision arithmetic, is the main advantage of our alternative approach. A variable-precision version of the latter is made available, if needed. All this is discussed in Section 3. Numerical illustrations are provided in Section 4.

All Matlab scripts used in this paper are downloadable from https://dx.doi.org/10.4231/R7NG4NKC.

2 Moment-based method

The moments $\mu_k^{(m)}=\int_0^m t^k\varphi_m(t)\mathrm{d}t$ of the cardinal B-spline φ_m of order m are easily obtained recursively by [4, Eq.(8)]

$$\mu_k^{(m)}=\frac{m}{m+k}\left(k\mu_{k-1}^{(m)}+\mu_k^{(m-1)}\right),\, k=0,1,2,\ldots,\qquad m\geq 1, \tag{2.1}$$

using the boundary conditions

$$\mu_{-1}^{(m)}=0,\,\mu_0^{(m)}=1,\,m\geq 1;\quad \mu_0^{(0)}=1,\,\mu_k^{(0)}=0,\,k\geq 1. \tag{2.2}$$

Indeed, each entry $\mu_k^{(m)}$ (in the (k,m)-plane) can be computed from the entries immediately to the left and immediately below. This is implemented in the symbolic Matlab routine `smom_Bspline.m`. The symbolic routine `schebyshev.m` (cf. [2, Appendix B]) then applies Chebyshev's algorithm to generate the desired recurrence coefficients $\alpha_k^{(m)}$, $\beta_k^{(m)}$ in

$$\begin{aligned}\pi_{k+1}(t)&=(t-\alpha_k^{(m)})\pi_k(t)-\beta_k^{(m)}\pi_{k-1}(t),\quad k=0,1,2,\ldots,\\ \pi_{-1}(t)&=0,\quad \pi_0(t)=1\end{aligned} \tag{2.3}$$

for the monic orthogonal polynomials $\pi_k(\,\cdot\,)=\pi_k(\,\cdot\,;\varphi_m)$.

The number of decimal digits (in units of 4) needed to obtain the first n pairs of recurrence coefficients to 16-digit accuracy, as determined by the routines `dig_Bspline.m`, `run_dig_Bspline.m` is shown in Table 1 for selected values of n and m.

Table 1 Number of decimal digits required for 16-digit recurrence coefficients

n	$m=2$	4	6	8	10	12	15	20
20	40	40	40	40	44	44	44	44
40	72	72	72	72	72	72	72	72
60	100	100	100	100	100	100	100	100
80	132	132	128	128	128	128	128	132
100	164	160	160	160	160	160	160	160

The symbolic Matlab routine `sr_cBspline_cheb.m` then computes the first n pairs of recurrence coefficients in `dig`-digit arithmetic.

3 Method based on multicomponent discretization

The method to be described, already used for splines of order 2 in [2, Exercise 2.16], requires the evaluation of cardinal B-splines φ_m of order m. The spline can be defined in terms of the divided difference of order m relative to the integer knots $0, 1, 2, \ldots, m$ of the truncated power function

$$\omega_m(t; x) = (t-x)_+^{m-1} = \begin{cases} (t-x)^{m-1} & \text{if} t \geq x, \\ 0 & \text{otherwise} \end{cases}$$

considered a function of t. Thus,

$$\varphi_m(x) = m[0, 1, 2, \ldots, m]\omega_m(\,\cdot\,; x). \tag{3.1}$$

It is a polynomial of degree $m-1$ on each interval $I_k = [k-1, k]$, $k = 1, 2, \ldots, m$, and overall a piecewise polynomial function on $\mathbb{R}$ of smoothness class $\mathbb{C}^{m-2}[\mathbb{R}]$, when $m \geq 2$. It vanishes identically outside the interval $[0, m]$. When $m = 1$, then φ_m equals 1 on $[0, 1]$ and 0 elsewhere.

Numerically, the evaluation of $\varphi_m(x)$ as defined in (3.1) is subject to loss of accuracy due to cancelation errors, which become worse the larger m. A more stable evaluation can be based on the recurrence relation [4, Eq. (5)],

$$\varphi_m(x) = \frac{1}{m-1}\left(x\varphi_{m-1}(x) + (m-x)\varphi_{m-1}(x-1)\right), m \geq 2 \quad x \in \mathbb{R}. \tag{3.2}$$

Indeed, if we define

$$\varphi_{m,k} = \varphi_m|_{x \in I_k}, \quad k = 1, 2, \ldots, m,$$

where $\varphi_{m,k} = 0$ if $k \leq 0$ or $k \geq m+1$, then by (3.2)

$$\begin{gathered} \varphi_{1,1} = 1, \\ \text{and for } \mu = 2, 3, \ldots, m \\ \varphi_{\mu,k} = \frac{1}{\mu-1}\left(x\varphi_{\mu-1,k} + (\mu-x)\varphi_{\mu-1,k-1}\right), k = 1, 2, \ldots, \mu. \end{gathered} \tag{3.3}$$

This is implemented in the Matlab routines `cBspline.m` and `scBspline.m`, which compute the m values of the cardinal B-spline φ_m of order m at the points $k-1+x$, $0<x<1$, $k=1,2,\dots,m$, in Matlab double resp. variable-precision arithmetic. The former is used in the script `plot_cBspline.m` to compute and plot the cardinal B-splines for $m=1:10$ and $m=12,15,20$. The plots are shown in Fig. 1.

We now use the method (3.3) of evaluating φ_m in the multicomponent discretization procedure (cf. [1, §2.2.4], [2, p. 8]), which in our application requires the evaluation of integrals

$$\int_{I_k} p(t)\varphi_m(t)\mathrm{d}t, \qquad k=1,2,\dots,m, \tag{3.4}$$

for polynomials $p(t)$, which are either $\pi_\nu^2(t)$ or $t\pi_\nu^2(t)$ with $\nu \le n-1$, where n is the number of pairs of recurrence coefficients that we want to compute. Since φ_m on each interval I_k is a polynomial of degree $m-1$, the integrand in (3.4) is a polynomial of degree at most $2n-1+m-1=2n+m-2$, which can be integrated exactly by $(n+\lceil m/2\rceil-1)$-point Gauss quadrature (relative to the interval I_k). Our method, therefore, produces exact answers in exact arithmetic, and turns out to be effective also in finite-precision arithmetic. It is implemented in the routine `r_cBspline_dis.m` producing the first n pairs of recurrence coefficients for the polynomials orthogonal with respect to the weight function φ_m. The corresponding symbolic routine is `sr_cBspline_dis.m`. An effort has been made to render these routines as efficient as possible. Thus, for example, when n and m

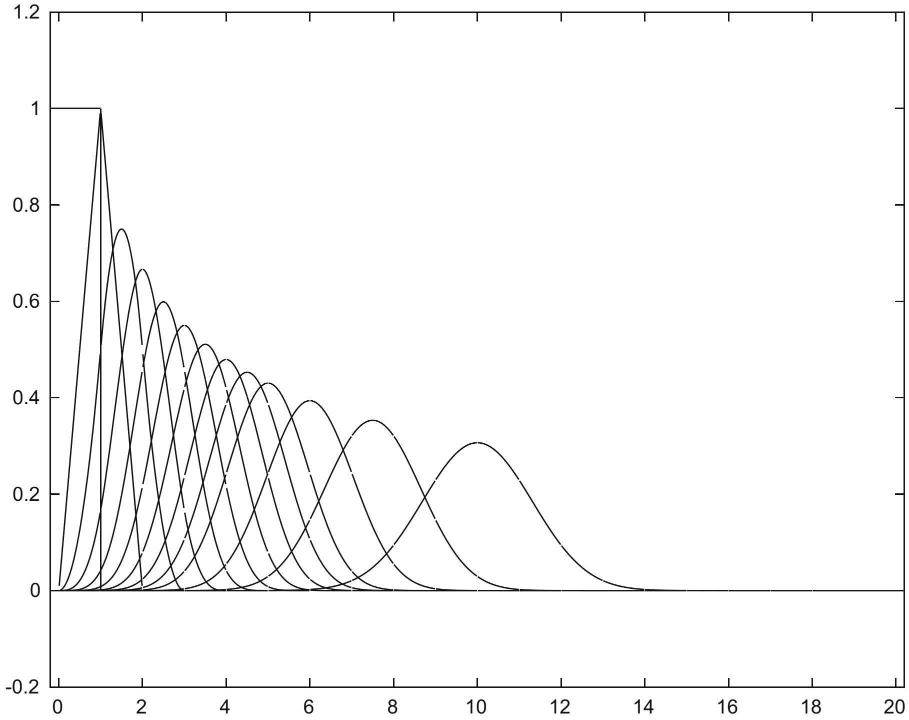


Fig. 1 Cardinal B-splines for $m=1:10, 12, 15, 20$

are given, the quadrature formula used to carry out the discretization in the routine `mcdis.m` resp. `smcdis.m` is generated only once at the beginning of the routines `r_cBspline_dis.m`, `sr_cBspline_dis.m` and transmitted to the routines `quad_cBspline.m`, `quad_scBspline.m` in `mcdis.m`, `smcdis.m` as a global variable. Even so, the symbolic routine `sr_cBspline_dis.m` may be slow, especially if m is large, even slower than `sr_cBspline_cheb.m`, because of m calls to the routine `scBspline.m` evaluating the spline function φ_m in variable-precision arithmetic. The double-precision routine `r_cBspline_dis.m`, however, is quite fast.

In order to establish the stability and efficiency of the procedure, we compare the results produced by the double-precision routine `r_cBspline_dis.m` with the results produced by the `dig`-digit routine `sr_cBspline_cheb.m` (with `dig` as in Table 1) and the 18-digit routine `sr_cBspline_dis.m`. The same values of n are used as in Table 1 and the values $m = [2 : 6, 10, 20]$. We select Stieltjes's method to obtain the recurrence coefficients from the discretized measure, since Lanczos's method is much slower in this application. The relative error in the β-coefficients was found to be always close to machine precision, never larger than 1.3×10^{-14} and as small as 6.32×10^{-16}. The errors are displayed in Table 2 and below each error the timings in seconds, first for the variable-precision Chebyshev algorithm and the 18-digit multicomponent discretization procedure, and next to them for the double-precision multicomponent discretization routine. Evidently, the two former are substantially larger than the last, because both use variable-precision arithmetic. Among the two variable-precision routines, the second is usually considerably faster, unless m is large. Table 2 was generated by the script `test_cBspline.m`.

The recurrence coefficients $\beta_k^{(m)}$ were observed to converge (however slowly) to limit values as $k \to \infty$, while by symmetry, $\alpha_k^{(m)} = m/2$ for all k. The existence of the limit $\beta_\infty^{(m)} = \lim_{k\to\infty} \beta_k^{(m)}$ in fact is a consequence of our weight function being in the Szegö class (for the latter, cf. [3]), so that the α- and β-coefficients for the

Table 2 Accuracy of the multicomponent discretization procedure and timings for variable-precision Chebyshev and 18-digit discretization vs double-precision discretization

n	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 10$	$m = 20$
20	2.56 (–15)	1.81 (–15)	1.94 (–15)	1.33 (–15)	6.66 (–16)	6.32 (–16)	8.92 (–16)
	[8, 4, .07]	[7, 4, .01]	[8, 5, .002]	[8, 7, .003]	[8, 8, .003]	[10, 20, .003]	[13, 86, .07]
40	8.11 (–15)	3.57 (–15)	2.90 (–15)	2.61 (–15)	1.42 (–15)	1.36 (–15)	2.34 (–15)
	[31, 8, .07]	[31, 9, .01]	[32, 12, .005]	[34, 16, .005]	[36, 20, .005]	[39, 44, .006]	[40, 180, .08]
60	7.44 (–15)	4.55 (–15)	3.79 (–15)	3.29 (–15)	2.98 (–15)	1.45 (–15)	1.10 (–15)
	[74, 15, .08]	[75, 19, .01]	[77, 24, .007]	[78, 30, .007]	[80, 37, .007]	[84, 82, .008]	[89, 284, .01]
80	8.00 (–15)	5.73 (–15)	5.45 (–15)	4.42 (–15)	4.75 (–15)	2.88 (–15)	2.24 (–15)
	[138, 27, .13]	[142, 32, .02]	[145, 40, .01]	[148, 50, .01]	[151, 68, .01]	[158, 155, .06]	[171, 454, .02]
100	1.30 (–14)	8.30 (–15)	6.33 (–15)	4.55 (–15)	3.17 (–15)	2.44 (–15)	1.79 (–15)
	[230, 43, .08]	[232, 52, .03]	[236, 62, .02]	[239, 88, .02]	[241, 100, .03]	[250, 219, .02]	[267, 632, .03]

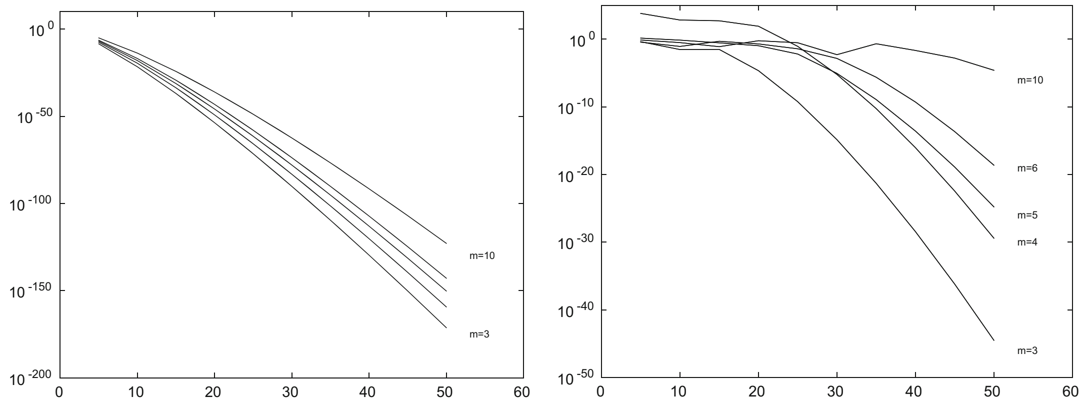


Fig. 2 Relative errors of Gauss quadrature of the integral I_{1a} (*left*) and the integral I_{1b} (*right*)

(monic) polynomials orthogonal on $[-1, 1]$ have limits equal, respectively, to 0 and 1/4, hence to $m/2$ and $m^2/16$ for polynomials orthogonal on $[0, m]$. Thus,

$$\beta_\infty^{(m)} = \frac{m^2}{16}\,. \tag{3.5}$$

This can be confirmed, at least to about 3 or more decimal places, by running the routine `r_cBspline_dis.m` with $N = 500$; cf. the script `run_r_cBspline.m`. (For N that large, Stieltjes's procedure experiences underflow problems, in contrast to Lanczos's procedure, which works flawlessly.)

4 Numerical illustrations

Example 1 (a) The integral

$$I_{1a} = \int_0^m 10 \sin x\, \varphi_m(x) \mathrm{d}x, \quad m = 3, 4, 5, 6, 10.$$

(b) The integral

$$I_{1b} = \int_0^m \left(1 - 2\sin\frac{19x}{3}\right)^3 \sinh\left(1 - \frac{x}{2}\right) \varphi_m(x) \mathrm{d}x.$$

This is Example 1 from [4]. As shown in Fig. 2, the relative errors of n-point Gaussian quadrature, $n = 5 : 5 : 50$, relative to the weight function φ_m, on the left for the integral I_{1a} and on the right for the integral I_{1b}, are very much in agreement with those found in [4]. The figures were produced by the

Table 3 Timings (in s) for Example 1

Ex	$m = 3$	4	5	6	10
1a	62.6	67.1	74.7	83.3	157
1b	51.3	56.3	63.7	71.6	124

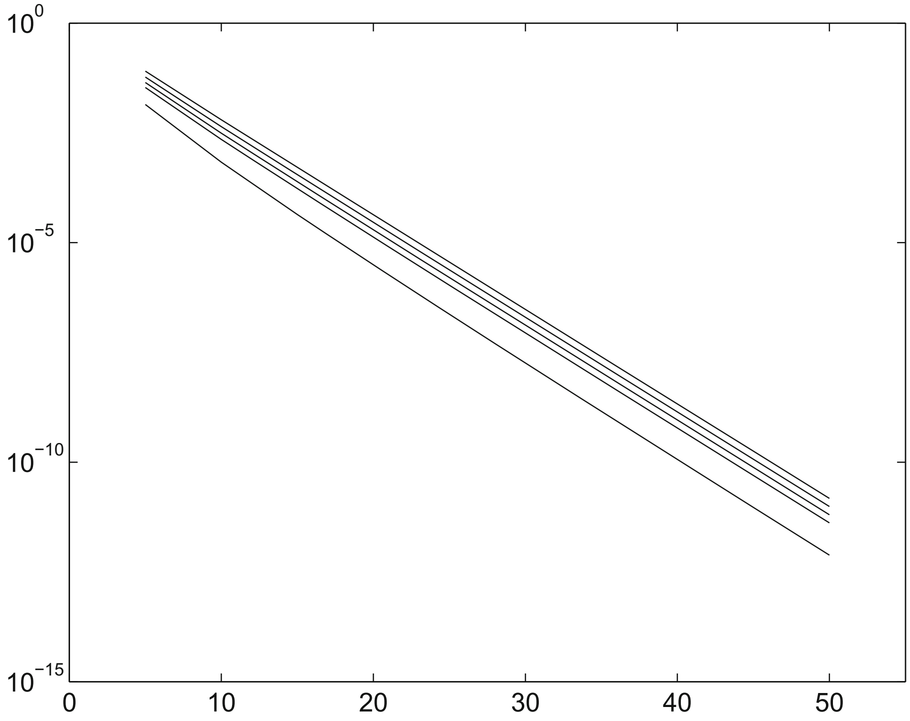


Fig. 3 Relative errors of Gauss quadrature of the integral I_2

scripts `cBspline_Ex1a.m` and `cBspline_Ex1b.m` in 200-digit resp. 50-digit variable-precision arithmetic. Timings (in seconds) for each curve are shown in Table 3.

Example 2 The integral

$$I_2 = \int_0^m \frac{10}{1 + 16\left(\frac{2x}{m} - 1\right)^2}\,\varphi_m(x)\mathrm{d}x.$$

This is Example 2 in [4]. The script `cBspline_Ex2.m`, using the double-precision routine `r_cBspline_dis.m`, produces graphs of the relative errors shown in Fig. 3 and corresponding timings in Table 4, similarly as in Example 1. Note, however, the much smaller run times, due to the exclusive use of double-precision arithmetic. The graphs are in agreement with those in [4].

The integrands in Example 1 are analytic functions, while the one in Example 2 has two conjugate complex poles relatively close to the real axis, about halfway

Table 4 Timings (in s) for Example 2

Ex	$m = 3$	4	5	6	10
2	.0203	.0203	.0199	.0202	.0220

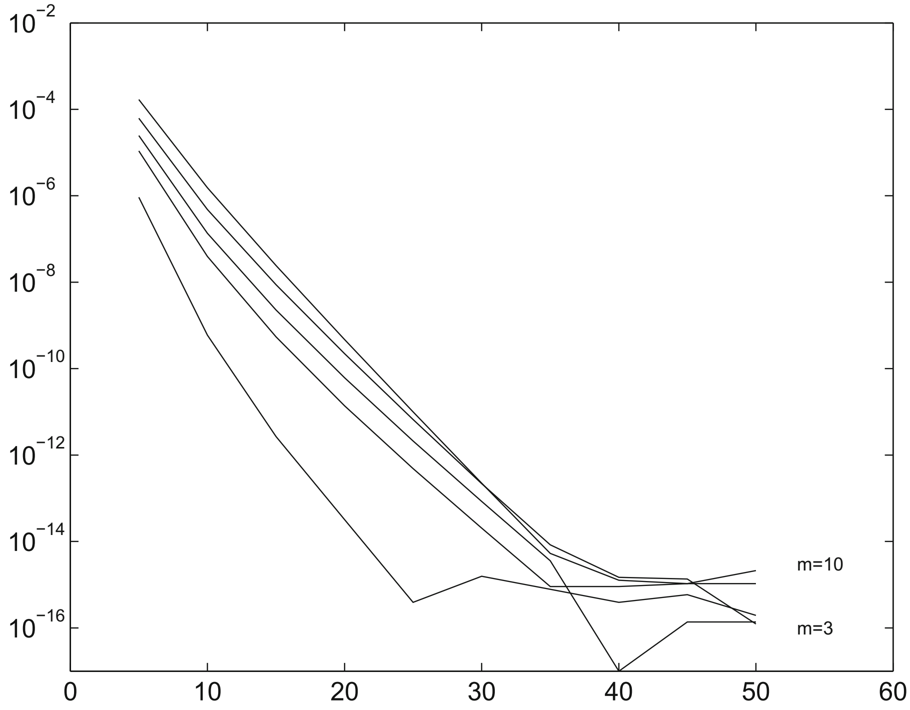


Fig. 4 Relative errors of Gauss quadrature of the integral I_3

through the interval $[0, m]$. In the next example, we take an integrand having an algebraic singularity to the left, and close to, the interval $[0, m]$.

Example 3 The integral

$$I_3 = \int_0^m (1 + 10x)^{-1/2}\varphi_m(x)\mathrm{d}x.$$

Gauss quadrature converges surprisingly fast: relevant graphs for $m = 3, 4, 5, 6, 10$ and timings, produced by `cBspline_Ex3.m`, are shown above (Fig. 4) and below (Table 5).

In all examples, the "exact" answers, for computing errors, were obtained by high-precision Gaussian quadrature involving 65 points and 200-digit, 50-digit arithmetic, respectively, in Example 1a, 1b, and 100 points and double-precision arithmetic in Examples 2 and 3.

Table 5 Timings (in s) for Example 3

Ex	$m = 3$	4	5	6	10
3	.0203	.0204	.0200	.0203	.0222

Acknowledgments The author was privileged to have seen a preprint of [4] before its publication. He is also indebted to Professor G. V. Milovanović for the reference [3] in connection with (3.5).

References

1. Gautschi, W.: Orthogonal polynomials: Computaion and Approximation, Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
2. Gautschi, W.: Orthogonal Polynomials in Matlab: Exercises and Solutions, Software, Environments, and Tools. SIAM, Philadelphia, PA (2016)
3. Lubinsky, D.S.: Asymptotics of orthogonal polynomials: some old, some new, some identities. In Proceedings of the International Conference on Rational Approximation, ICRA99 (Antwerp). Acta. Appl. Math. **61**, 207–256 (2000)
4. Milovanović, G.V.: Symbolic numeric computation of orthogonal polynomials and Gaussian quadratures with respect to the cardinal B-spline. to appear

44.3. [219] (with Gradimir V. Milovanović) “Binet-tpe polynomials and their zeros”

[219] (with Gradimir V. Milovanović) “Binet-type polynomials and their zeros”, *Electron. Trans. Numer. Anal.* **50**, 52–70 (2018).

Electronic Transactions on Numerical Analysis.
Volume 50, pp. 52–70, 2018.
Copyright © 2018, Kent State University.
ISSN 1068–9613.
DOI: 10.1553/etna_vol50s52

BINET-TYPE POLYNOMIALS AND THEIR ZEROS*

WALTER GAUTSCHI† AND GRADIMIR V. MILOVANOVIĆ‡

Abstract. Procedures based on moments are developed for computing the three-term recurrence relations 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.

Key words. Binet weight function, orthogonal polynomials, zeros, monotonicity

AMS subject classifications. 33C47, 65D20

1. Introduction.

The *Binet weight function* may be defined by

$$w_1(x) = -\log(1 - \mathrm{e}^{-|x|}) \quad \text{on } [-\infty, \infty]. \tag{1.1}$$

It has been introduced, in connection with a number of summation formulas [2, 3, 4], in [4, eq. (5.4)], where the Binet *distribution* is defined by $w^B(x) = w_1(2\pi x)/(2\pi)$, and has been used in Binet's summation formula, *ibid.*, eq. (5.15). More generally,

$$w_1(x; \alpha) = -\log(1 - \alpha \mathrm{e}^{-|x|}) \qquad \text{on } [-\infty, \infty], \;\; 0 < \alpha < 1, \tag{1.2}$$

is what may be called the *generalized Binet weight function*. We are interested in the polynomials orthogonal with respect to the weight functions (1.1) and (1.2), in particular, in the recurrence formulas

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

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, respectively 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.

*Received December 19, 2017. Accepted March 4, 2018. Published online on November 15, 2018. Recommended by L. Reichel. The research of G. V. Milovanović is supported in part by the Serbian Academy of Sciences and Arts (No. Φ-96).

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

‡Serbian Academy of Sciences and Arts, Belgrade, Serbia & University of Niš, Faculty of Sciences and Mathematics, 18000 Niš, Serbia (gvm@mi.sanu.ac.rs).

2. Binet and generalized Binet weight functions. Moment-related methods and their implementation, both in Matlab and Mathematica, are considered in Section 2.1 for the Binet weight function and in Section 2.2.1 for generalized Binet weight functions. Section 2.2.2 is devoted to a study of the zeros of orthogonal polynomials depending on a parameter and, in particular, to the monotone behavior of the zeros of generalized Binet polynomials π_n^α when considered as functions of the parameter α.

2.1. The Binet weight function. Since the weight function in (1.1) is symmetric with respect to the origin, its moments are

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ -2\displaystyle\int_0^\infty x^k \log(1-\mathrm{e}^{-x})\,\mathrm{d}x & \text{if } k \text{ is even.} \end{cases} \tag{2.1}$$

Substituting $\mathrm{e}^{-x}=t$ in the integral of (2.1), one gets

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

and thus

$$\mu_k = 2\,k!\,S_{k+1,1}(1) = 2\,k!\,\mathrm{Li}_{k+2}(1) = 2\,k!\,\zeta(k+2), \qquad k \ \text{ even,} \tag{2.2}$$

where

$$S_{n,p}(x) = \frac{(-1)^{n+p-1}}{(n-1)!p!}\int_0^1 \log^{n-1}(t)\log^p(1-xt)\,\frac{\mathrm{d}t}{t} \tag{2.3}$$

is the Nielsen generalized polylogarithm [11, eq. (1.1)] and $\mathrm{Li}_n(x)$ the ordinary polylogarithm [10, eq. (1.1)]. We can thus apply the classical Chebyshev algorithm (cf., e.g., [5, §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`N` moments (2.1),

```
mom=smom_binet(dig,N).
```

In addition, the routine `dig_binet.m` is provided, which, by the command

(2.4)
```
[ab,dig]=dig_binet(N,dig0,dd,nofdig),
```

helps to determine the number `dig` of digits needed to obtain the `N`×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, then 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

(2.5)
```
ab=sr_binet(dig,nofdig,N),
```

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

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

EXAMPLE 2.1 (The first 100 recurrence coefficients to 32 digits of the Binet weight function). With `N` $=$ 100, `dig0` $=$ 56, `dd` $=$ 4, `nofdig` $=$ 32, the routine (2.4) 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 displayed in the second and third plot of Figure 2.1, the first showing the Binet weight function. The recurrence coefficients are also made available in the textfile `coeff_binet.txt`, which can be loaded into the Matlab working window by the routine `loadvpa.m`. For the latter, see [6, p. ix]; see also [8, 2.3.8]. The array `ab` can also be obtained directly with the routine (2.5) using `dig` $=$ 64, `nofdig` $=$ 32, and `N` $=$ 100.

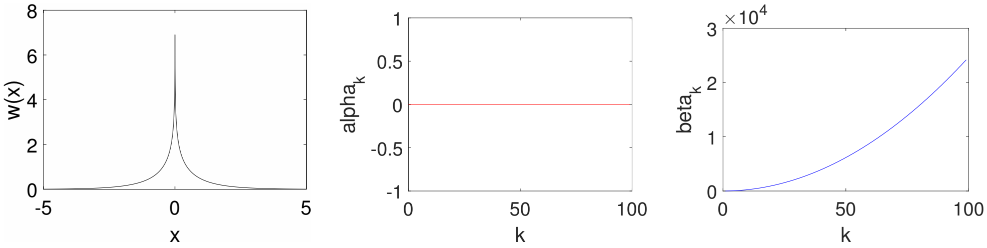


FIG. 2.1. *The Binet weight function and its recurrence coefficients.*

The same 100 recurrence coefficients have been obtained in symbolic form by the Mathematica package `OrthogonalPolynomials` ([1, 13]) using the commands

```
<<orthogonalPolynomials`
momGB[k_,alpha_]:=If[OddQ[k],0,2 k!PolyLog[k+1,1,alpha]];
mom = Table[momGB[k,1], {k,0,199}];
{alBSym,beBSym}=aChebyshevAlgorithm[mom,Algorithm->Symbolic]
```

The β_k, for $0 \leq k \leq 11$, so obtained are given in [12, p. 457] in rational form and those for $12 \leq k \leq 39$ to 60 decimals on page 458 of the same reference. There is complete agreement to all 32 digits between the coefficients obtained in Matlab and those obtained in Mathematica rounded to 32 digits. The fact that the recurrence coefficients are rational numbers multiplied by π^2 makes the computation in symbolic Mathematica extremely fast. For example, on a MacBook Pro Retina OSX 10.12.6 laptop, the first 100 recurrence coefficients in symbolic form are obtained in 1.04 s. These symbolic formulae can then be used to compute the coefficients to arbitrary precision. For example, `N[beBSym,32]` produces the β-coefficients to 32 digits in 1.1 ms, and `N[beBSym,500]` yields the same coefficients to 500 digits in 1.3 ms. If one uses the numerical calculation option in the Chebyshev algorithm,

```
{alB,beB}=aChebyshevAlgorithm[mom,WorkingPrecision->58]
```

with 58-digit working precision (`WP`), then the first 100 recurrence coefficients are obtained to 32 digits in 77.3 ms. With 86-digit `WP`, they are obtained to 60 digits in 81.8 ms and with 160-digit `WP` to 135 digits in 84.8 ms. In contrast, Matlab, on a Dell Optiplex 790 computer, takes 186 s to compute the same 100 coefficients to 32 digits.

2.2. The generalized Binet weight function.

2.2.1. The recurrence coefficients. The weight function (1.2), again being symmetric, has moments

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ -2\displaystyle\int_0^\infty x^k \log(1-\alpha e^{-x})dx & \text{if } k \text{ is even.} \end{cases} \tag{2.6}$$

Similarly as in Section 2.1, one finds

$$\mu_k = 2\,k!\,S_{k+1,1}(\alpha) = 2\,k!\,\mathrm{Li}_{k+2}(\alpha), \qquad k \text{ even.} \tag{2.7}$$

The moments (2.6) are generated by the Matlab command

```
mom=smom_gbinet(dig,N,a),
```

where a is the value of α and $0 < \alpha < 1$.

EXAMPLE 2.2 (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 displayed in the second and third plot of Figure 2.2, the first showing the generalized Binet weight function for $\alpha = 1/2$. They are also made available in the textfile coeff_gbinet.txt.

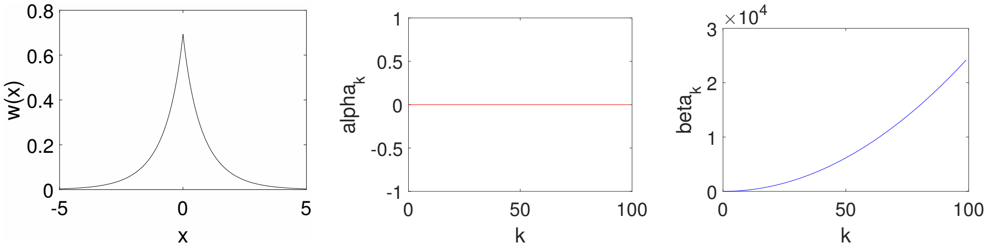


FIG. 2.2. *The generalized Binet weight function for $\alpha = 1/2$ and its recurrence coefficients.*

Unlike the case $\alpha = 1$, when the recurrence coefficients are simply rational numbers multiplied by π^2, in the general case $0 < \alpha < 1$, the symbolic expressions of the recurrence coefficients β_k become rapidly more complicated with increasing k and hence, the runtime correspondingly larger. Yet, using the numerical calculation option in both the computation of the moments and Chebyshev's algorithm yields fast algorithms similarly to those described in Example 2.1.

2.2.2. The zeros of orthogonal polynomials. The first objective of this section 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 provide appropriate plots.

We first recall Markov's theorem ([14, Theorem 6.12.1]).

THEOREM 2.3 (A. Markov). *Let $w(x;\alpha)$ be a positive weight function on $[a,b]$, $-\infty \leq a < b \leq \infty$, 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*

$$M(x;\alpha) := \frac{1}{w(x;\alpha)}\,\frac{\partial w(x;\alpha)}{\partial \alpha} \tag{2.8}$$

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

Here are two simple corollaries to Markov's theorem.

COROLLARY 2.4. *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 $\leq 2n-1$. 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 (2.8) 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 $r < 0$, then the type of monotonicity is reversed, from increasing to decreasing and vice versa.

COROLLARY 2.5. *Let $w(x;\alpha)$ be symmetric on the interval $[-a,a]$, $0 < a \leq \infty$, i.e., $w(-x;\alpha) = w(x;\alpha)$, for $0 \leq x \leq a$, but otherwise as in Theorem 2.3. Then each positive zero of π_n^α is increasing (decreasing) on (α_1,α_2) depending on whether (2.8) is increasing (decreasing) on $[0,a]$.*

Proof. Suppose first that $n = 2k$ is even. Then, as is well known (see, e.g., [5, 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 they are increasing (decreasing) on $[\alpha_1,\alpha_2]$ if the same is true for the zeros of $\pi_k^+(\,\cdot\,;\alpha)$. But there holds

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

from which Corollary 2.5 follows.

For odd n, 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 $[a,c]$, $-\infty \leq a < c < \infty$, with respect to a weight function w,

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

THEOREM 2.6. *Let $w(x)$ be a positive weight function on $[a,c]$, $-\infty \leq a < c < \infty$, having finite moments μ_k, for $0 \leq k \leq 2n-1$. Then each zero $x_\nu = x_\nu(c)$ of π_n is monotonically increasing as a function of c.*

Proof. The proof follows the same line of arguments as the proof of Markov's theorem given in [14, Theorem 6.12.1] being based on the Gauss quadrature formula

$$\int_a^c p(x)w(x)\mathrm{d}x = \sum_{\mu=1}^{n} \lambda_\mu(c)\, p(x_\mu(c)), \qquad p \in \mathbb{P}_{2n-1}. \tag{2.9}$$

Differentiating (2.9) with respect to c, we have

$$p(c)w(c) = \sum_{\mu=1}^{n} \lambda_\mu(c)\, p'(x_\mu(c))\, \frac{\mathrm{d}x_\mu}{\mathrm{d}c} + \sum_{\mu=1}^{n} \frac{\mathrm{d}\lambda_\mu}{\mathrm{d}c}\, p(x_\mu(c)). \tag{2.10}$$

Let

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

Then, since $p(x_\mu) = 0$ for all μ and $p'(x_\mu) = 0$ for $\mu \neq \nu$, we get from (2.10) that

$$\frac{\pi_n^2(c)}{c - x_\nu}\, w(c) = \lambda_\nu(c)\, [\pi_n'(x_\nu)]^2\, \frac{\mathrm{d}x_\nu}{\mathrm{d}c}\, . \tag{2.11}$$

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

REMARK 2.7. Theorem 2.6 is valid also if c is the lower limit of the orthogonality interval, by the same proof. Indeed, the left-hand side of equation (2.10) will then have a minus sign in front of it and so does the left-hand side of (2.11). But now, $x_\nu > c$.

Returning to Theorem 2.3, the weight function is $w_1(x;\alpha)$ in (1.2), which is clearly symmetric on $(-\infty, \infty)$, so that according to Corollary 2.5, the positive zeros of the generalized Binet polynomial π_n^α are increasing (decreasing) depending on whether

$$\frac{1}{w_1(x;\alpha)}\, \frac{\partial w_1(x;\alpha)}{\partial \alpha} = \frac{-1}{(\mathrm{e}^x - \alpha)\log(1 - \alpha \mathrm{e}^{-x})} \tag{2.12}$$

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

Let the right-hand side of (2.12), as a function of x, be denoted by $f(x)$ and the denominator by $g(x)$. Then

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

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, we have the following theorem:

THEOREM 2.8. *All positive zeros of the generalized Binet polynomial π_n^α are monotonically decreasing as functions of α.*

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 $\texttt{dig} = 16$. Once the respective variable-precision array `ab` has been obtained, one can revert to double precision for the rest of the computations.

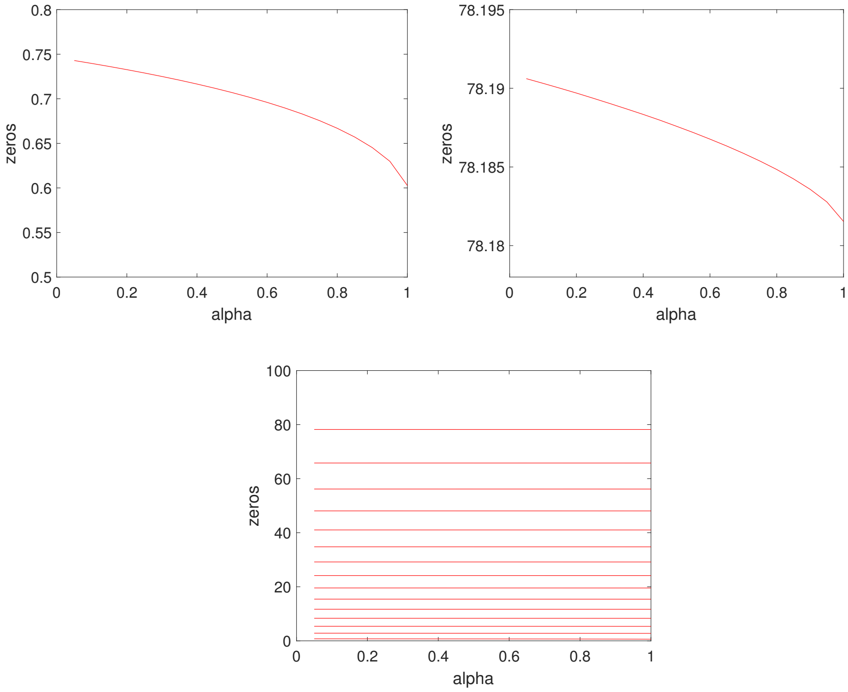


FIG. 2.3. *The positive zeros for $n = 30$ of the generalized Binet polynomial π_n^α in dependence of the parameter α, $0 < \alpha \leq 1$; the smallest and largest positive zero (top); all positive zeros (bottom).*

The Matlab script `plot_zeros_gbinet.m` with $\texttt{N} = 30$ 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 small, almost imperceptible; see the third plot in Figure 2.3. 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}.

3. Squared Binet and squared generalized Binet weight functions.

Moment-related methods and their implementation, both in Matlab and Mathematica, are considered in Section 3.1 for the squared Binet weight function and in Section 3.2.1 for squared generalized Binet weight functions. Section 3.2.2 deals with zeros of squared generalized Binet polynomials.

3.1. The squared Binet weight function.

The *squared Binet weight function*,

$$w_2(x) = \log^2(1 - \mathrm{e}^{-|x|}) \qquad \text{on } [-\infty, \infty], \tag{3.1}$$

being symmetric, has the moments

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ 2\displaystyle\int_0^\infty x^k \log^2(1-\mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even.} \end{cases} \tag{3.2}$$

Putting $e^{-x} = t$ in the integral of (3.2), 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 $\mu_0 = 4\,\zeta(3)$ ([9, 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 (2.3). We have ([11, pp. 39, 41] or [10, p. 1236])

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

and [11, eq. (4.16)]

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

so that

$$\mu_k = 2k!\left[(k+2)\zeta(k+3) - \sum_{\nu=2}^{k+1} \zeta(\nu)\zeta(k+3-\nu)\right], \qquad k \text{ (even) } > 0. \tag{3.3}$$

The first N moments (3.3) are generated in dig-digit arithmetic by the Matlab command

```
mom=smom_sqbinet(dig,N).
```

EXAMPLE 3.1 (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 displayed in the second and third plot of Figure 3.1, the first showing the squared Binet weight function. They are also made available in the textfile coeff_sqbinet.txt; see also [8, 2.3.9].

Symbolic computation in Mathematica of the first 200 moments is accomplished by the command

```
  momSGB = Table[If[k == 0, 4 Zeta[3], If[OddQ[k], 0, 2k!
                ((k+2)Zeta[k+3]-Sum[Zeta[v]Zeta[k+3-v],
                {v,2,k+1}])]], {k,0,199}];
```

taking 170.1 ms to run. With the numerical calculation option in Chebyshev's algorithm using working precision WP $= 108$ yields the first 100 recurrence coefficients to 32 digits in 136.2 ms, with WP $= 136$ to 60 digits in 147.7 ms, and with WP $= 196$ to 120 digits in 168.4 ms.

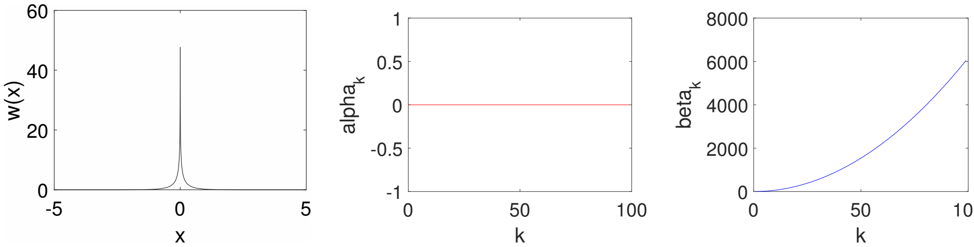


FIG. 3.1. *The squared Binet weight function and its recurrence coefficients.*

3.2. The squared generalized Binet weight function.

3.2.1. The recurrence coefficients. The *squared generalized Binet weight function*,

$$w_2(x;\alpha) = \log^2(1-\alpha\mathrm{e}^{-|x|}) \qquad \text{on } [-\infty,\infty],\ 0<\alpha<1, \tag{3.4}$$

has the moments

$$\mu_k = \begin{cases} 0 & \text{if } k \text{ is odd,} \\ 2\displaystyle\int_0^\infty x^k \log^2(1-\alpha\mathrm{e}^{-x})\mathrm{d}x & \text{if } k \text{ is even.} \end{cases} \tag{3.5}$$

Similarly as in Section 3.1, one finds

$$\mu_k = 4\,k!\,S_{k+1,2}(\alpha), \tag{3.6}$$

where $S_{n,p}(x)$ is the Nielsen generalized polylogarithm function (2.3). We have [10, eq. (2.9)]

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

The series converges fairly rapidly for all $k \ge 0$ provided α is not too close to 1. The moments (3.5) are generated by the Matlab command

```
mom=smom_sqg_binet(dig,N,a).
```

EXAMPLE 3.2 (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 digits. They are displayed in the second and third plot of Figure 3.2, the first showing the squared generalized Binet weight function for $\alpha = 1/2$. They are also made available in the textfile `coeff_sqgbinet.txt`; see also [8, 2.3.11].

Symbolic computation in Mathematica of the first 200 moments is accomplished by the command

```
momSGB=Table[If[OddQ[k],0,4k!PolyLog[k+1,2,1/2]],{k,0,199}];
```

taking 3.3 ms to run (equally fast for every $\alpha < 1$). With the numerical calculation option in Chebyshev's algorithm using working precision `WP` $=64$, the first 100 recurrence coefficients to 32 digits are obtained in 2.12 s, with `WP` $=92$ to 60 digits in 2.90 s, and with `WP` $=152$ to 120 digits in 5.55 s.

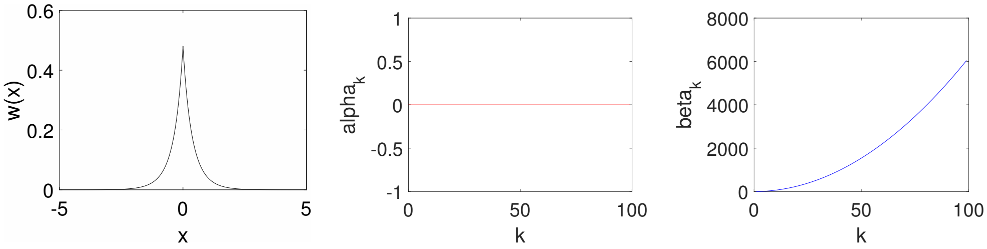


FIG. 3.2. *The squared generalized Binet weight function for $\alpha = 1/2$ and its recurrence coefficients.*

3.2.2. The zeros of the orthogonal polynomials. By virtue of Corollary 2.4 to Markov's theorem and of what was proved in Section 2.2.2, we have

THEOREM 3.3. *All positive zeros of the squared generalized Binet polynomial π_n^α are monotonically decreasing as functions of α.*

The zeros 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 presented in the first two plots of Figure 3.3 and all 15 positive zeros in the third plot; cf. the Matlab script `plot_zeros_sqgbinet.m`.

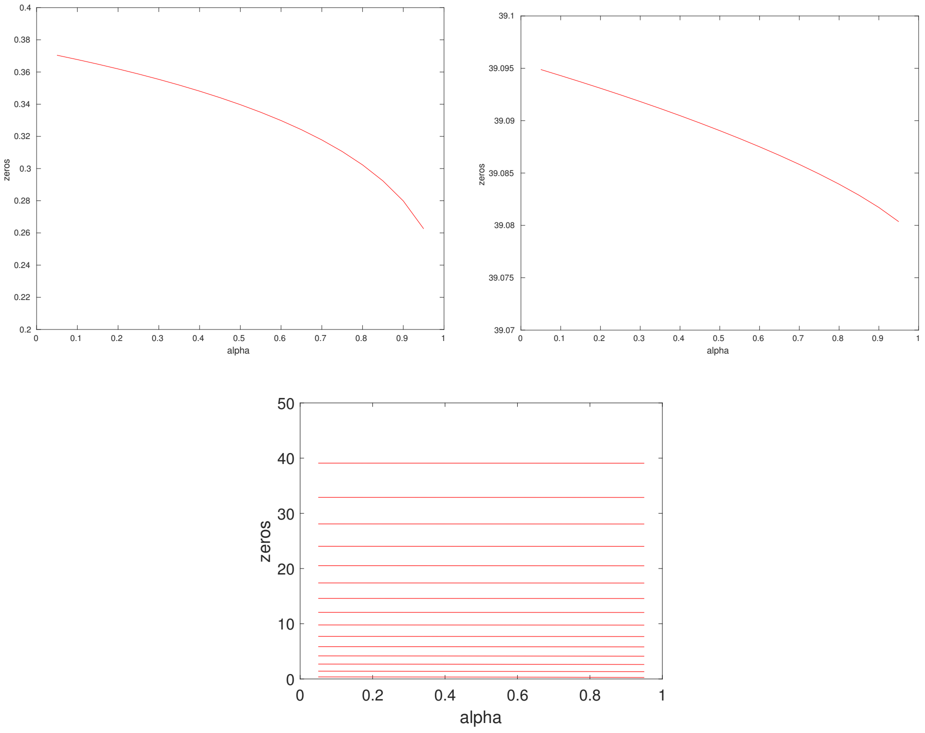


FIG. 3.3. *The positive zeros for $n = 30$ of the squared generalized Binet polynomials in dependence of the parameter α, $0 < \alpha < 1$; the smallest and largest positive zero (top); all positive zeros (bottom).*

4. Halfrange Binet and halfrange generalized Binet weight functions. Moment-related methods and their implementation in Matlab are considered in Section 4.1 for the halfrange Binet weight function and in Section 4.2.1 for halfrange generalized Binet weight functions. Section 4.2.2 deals with zeros of halfrange generalized Binet polynomials.

4.1. The halfrange Binet weight function. The halfrange Binet weight function is the weight function (1.1) supported on $[0,\infty]$. Its moments are (cf. equations (2.1), (2.2))

$$\mu_k = k!\,\zeta(k+2), \qquad k=0,1,2,\dots \tag{4.1}$$

They are generated by the Matlab command

```
mom=smom_hrbinet(dig,N).
```

EXAMPLE 4.1 (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 displayed in the second and third plot of Figure 4.1, the first showing the halfrange Binet weight function. They are also made available in the textfile `coeff_hrbinet.txt`; see also [8, 2.9.26].

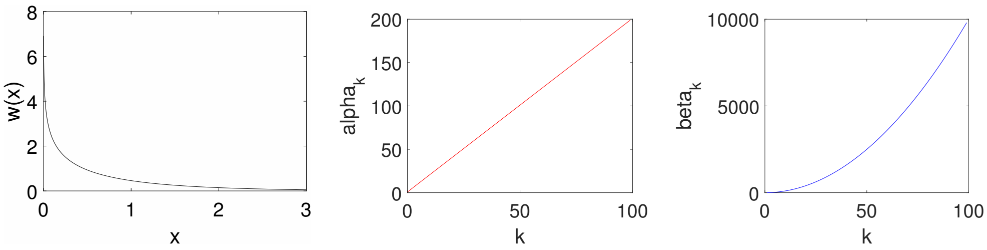


FIG. 4.1. *The halfrange Binet weight function and its recurrence coefficients.*

4.2. The halfrange generalized Binet weight function.

4.2.1. The recurrence coefficients. The halfrange generalized Binet weight function is the weight function (1.2) supported on $[0,\infty]$. Its moments are (cf. equations (2.6), (2.7))

$$\mu_k = k!\,S_{k+1,1}(\alpha) = k!\,\mathrm{Li}_{k+2}(\alpha), \quad k=0,1,2,\dots \tag{4.2}$$

They are generated by the Matlab command

`mom=smom_hrgbinet(dig,N,a)`, `a` $=\alpha$.

EXAMPLE 4.2 (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` $=128$. As before, the same command or 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 displayed in the second and third plot of Figure 4.2, the first showing the halfrange generalized Binet weight function for $\alpha=1/2$. They are also made available in the textfile `coeff_hrgbinet.txt`; see also [8, 2.9.30].

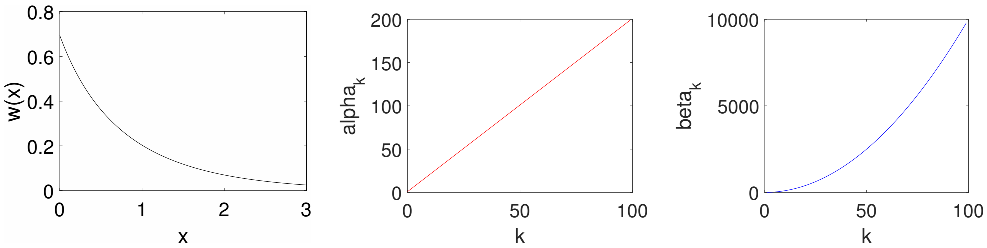


FIG. 4.2. *The halfrange generalized Binet weight function for $\alpha = 1/2$ and its recurrence coefficients.*

4.2.2. The zeros of the orthogonal polynomials. By what was proved in Section 2.2.2, we have

THEOREM 4.3. *All zeros of the halfrange generalized Binet polynomial π_n^α are monotonically decreasing as functions of α.*

For $n = 15$, the smallest and largest zero are presented in the first two plots of Figure 4.3 and all zeros in the third plot; cf. `plot_zeros_hrgbinet.m`.

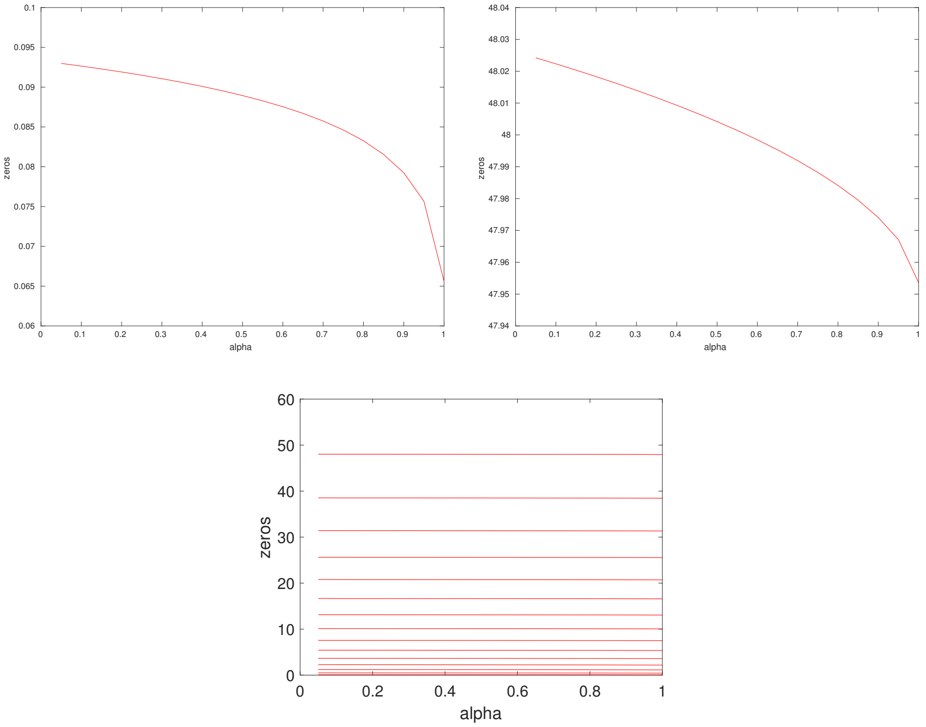


FIG. 4.3. *The zeros for $n = 15$ of the halfrange generalized Binet polynomials in dependence of the parameter α, $0 < \alpha \leq 1$; the smallest and largest zero (top); all zeros (bottom).*

5. Halfrange squared Binet and halfrange squared generalized Binet weight functions.

Moment-related methods and their implementation in Matlab are considered in Section 5.1 for the halfrange squared Binet weight function and in Section 5.2.1 for halfrange squared generalized Binet weight functions. Section 5.2.2 deals with zeros of halfrange squared generalized Binet polynomials.

5.1. The halfrange squared Binet weight function. The halfrange squared Binet weight function is the weight function (3.1) supported on $[0,\infty]$. Its moments are (cf. equations (3.2), (3.3))

$$\mu_0 = 2\,\zeta(3),$$
$$\mu_k = k!\left[(k+2)\zeta(k+3) - \sum_{\nu=2}^{k+1}\zeta(\nu)\zeta(k+3-\nu)\right], \qquad k=1,2,3,\dots$$

The first `N` of them are generated in `dig`-digit arithmetic by the Matlab command

```
mom=smom_hrsqbinet(dig,N).
```

EXAMPLE 5.1 (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 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are displayed in the second and third plot of Figure 5.1, the first showing the halfrange squared Binet weight function. They are also made available in the textfile `coeff_hrsqbinet.txt`; see also [8, 2.9.31].

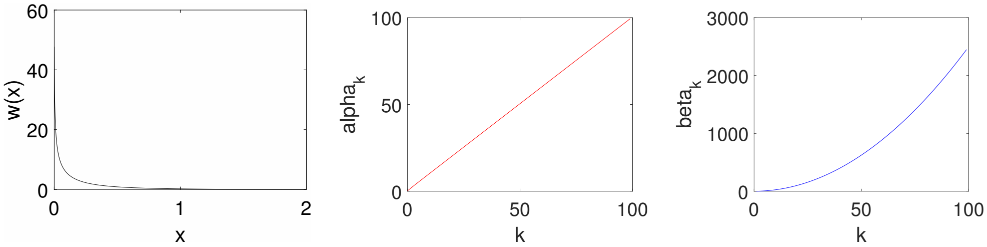


FIG. 5.1. *The halfrange squared Binet weight function and its recurrence coefficients.*

5.2. The halfrange squared generalized Binet weight function.

5.2.1. The recurrence coefficients. The halfrange squared generalized Binet weight function is the weight function (3.4) supported on $[0,\infty]$. Its moments are (cf. equations (3.5), (3.6))

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

They are generated by the Matlab command

```
mom=smom_hrsqgbinet(dig,N,a).
```

EXAMPLE 5.2 (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 $\texttt{N} = 100$, $\texttt{a} = 1/2$, $\texttt{dig0} = 116$, $\texttt{dd} = 4$, $\texttt{nofdig} = 32$, yields $\texttt{dig} = 124$. The same command or the direct command `ab=sr_hrsqgbinet(dig,nofdig,N,a)` with $\texttt{dig} = 124$ produces the $100{\times}2$ array `ab` of the first 100 recurrence coefficients to 32 digits. They are displayed in the second and third plot of Figure 5.2, the first showing the halfrange squared generalized Binet weight function for $\alpha = 1/2$. They are also made available in the textfile `coeff_hrsqgbinet.txt`; see also [8, 2.9.32].

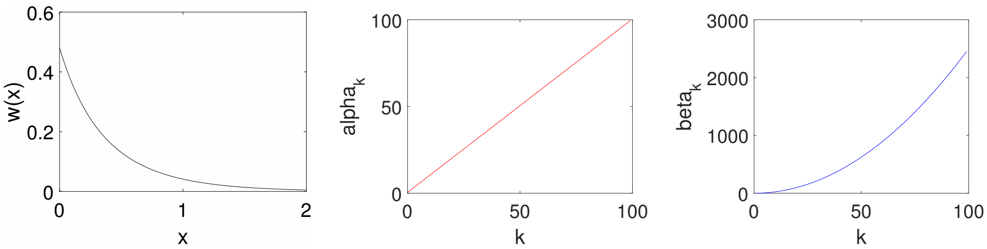


FIG. 5.2. *The halfrange squared generalized Binet weight function for $\alpha = 1/2$ and its recurrence coefficients*

5.2.2. The zeros of the orthogonal polynomials. Since, according to Section 4.2.2, all zeros of the halfrange generalized Binet polynomial are monotonically decreasing, the same is true, by Corollary 2.4 of Markov's theorem (cf. Section 2.2.2), for the square of the weight function. Thus we have

THEOREM 5.3. *All zeros of the halfrange squared generalized Binet polynomial π_n^α are monotonically decreasing as functions of α.*

For $n = 15$, the smallest and largest zero are displayed in the first two plots of Figure 5.3 and all zeros in the third plot; cf. `plot_zeros_hrsqgbinet.m`.

6. Subrange Binet weight functions.

Moment-related methods and their implementation in Matlab are considered in Section 6.1.1 for an upper subrange Binet weight function, in Section 6.2.1 for a lower subrange Binet weight function, and in Section 6.3.1 for a lower symmetric subrange Binet weight function. Sections 6.1.2, 6.2.2, and 6.3.2 deal with the zeros of the respective subrange Binet polynomials.

6.1. An upper subrange Binet weight function.

6.1.1. The recurrence coefficients. The weight function (1.1) 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 $t = e^{c-x}$,

$$\mu_k = \sum_{\nu=0}^{k} k^{(\nu)} c^{k-\nu} \mathrm{Li}_{\nu+2}(e^{-c}), \qquad k = 0, 1, 2, \dots, \tag{6.1}$$

where

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

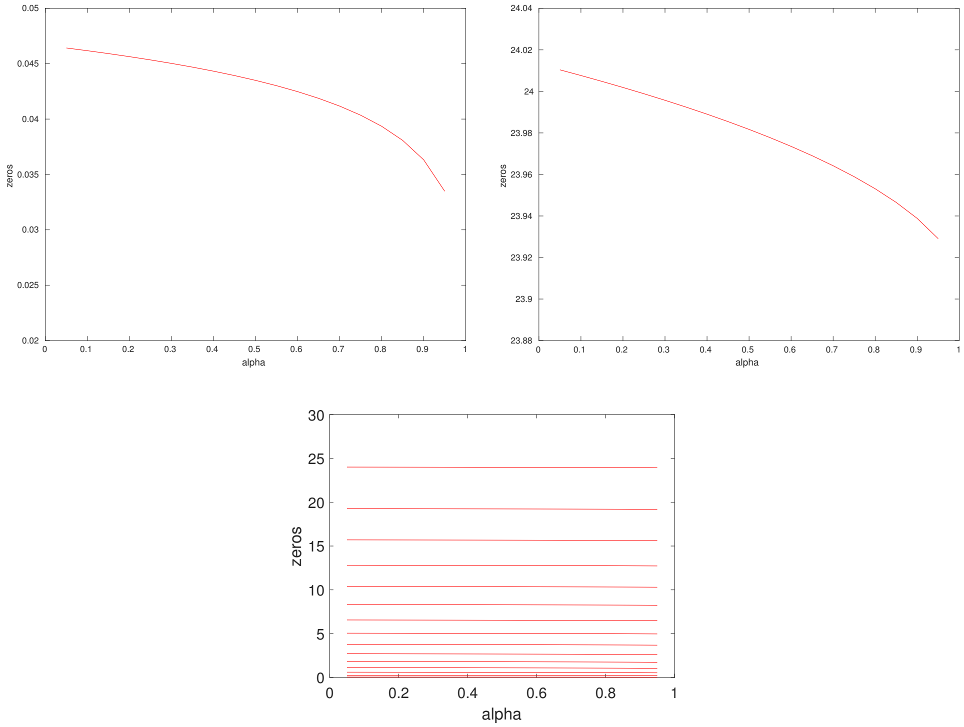


FIG. 5.3. *The zeros for* $n = 15$ *of the halfrange squared generalized Binet polynomials in dependence of the parameter* α, $0 < \alpha < 1$*; the smallest and largest zero (top); all zeros (bottom).*

is the descending factorial power and $\mathrm{Li}_n(x)$ the polylogarithm (cf. Section 2.1). The moments (6.1) 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. Section 4.2). Denoting the recurrence coefficients of the latter by $a_k(\alpha)$, $b_k(\alpha)$, $k = 0, 1, 2, \ldots$, it is easy to see that

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

The moments needed to generate the $a_k(\alpha)$, $b_k(\alpha)$ are then those in (4.2), which are definitely simpler than those in (6.1). They are produced by the Matlab command

```
mom=smom_usrbinet_alt(dig,N,c).
```

EXAMPLE 6.1 (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` $= 128$. As before, this or the more direct command `ab=sr_usrbinet_alt(dig,nofdig,N,c)` with `dig` $= 128$, `c` $= 1$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are displayed in the second and third plot of Figure 6.1, the first showing the upper subrange Binet weight function on $[1, \infty]$. They are also made available in the textfile `coeff_usrbinet.txt`; see also [8, 2.9.28].

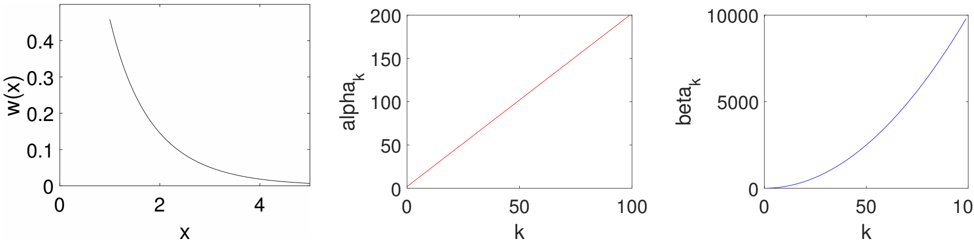


FIG. 6.1. *The upper subrange Binet weight function and its recurrence coefficients.*

6.1.2. The 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, applying Remark 2.7 to Theorem 2.6 of Section 2.2.2 gives the following result.

THEOREM 6.2. *All zeros of the upper subrange Binet polynomial π_n, orthogonal on $[c, \infty]$, are monotonically increasing as functions of c.*

The zeros for $n = 15$ are displayed in Figure 6.2; cf. `plot_zeros_usrbinet.m`.

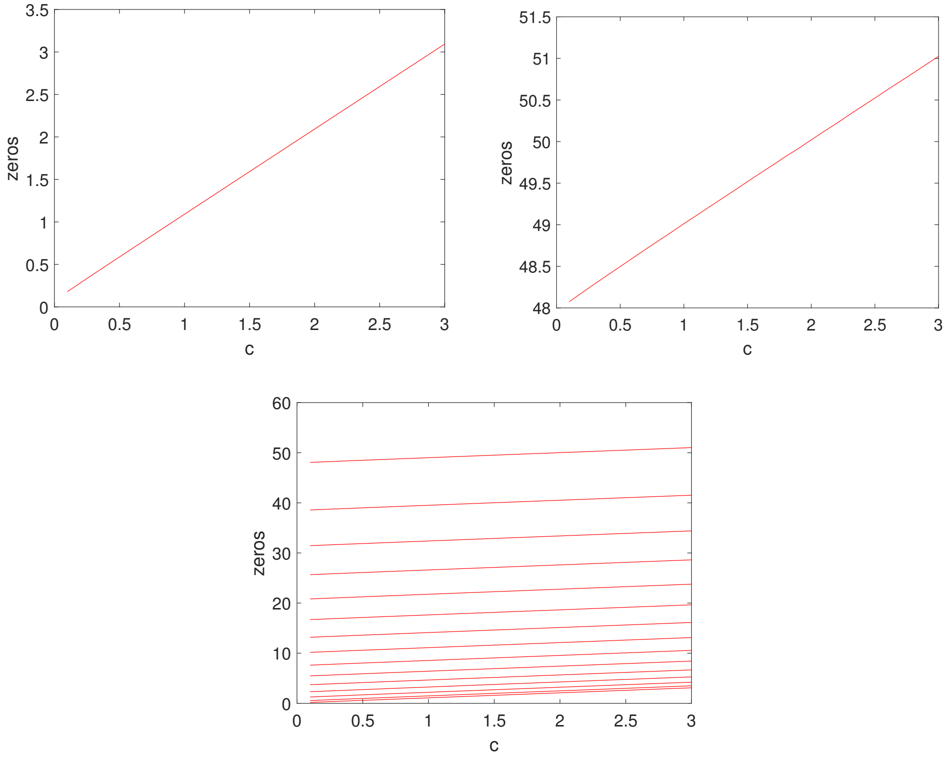


FIG. 6.2. *The zeros for $n = 15$ of the upper subrange Binet polynomials in dependence of the parameter c, $0 < c \leq 3$; the smallest and largest zero (top); all zeros (bottom).*

6.2. A lower subrange Binet weight function.

6.2.1. The recurrence coefficients. We consider here the weight function (1.1) 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,

$$\mu_k = \mu_k^{\text{hr}} - \mu_k^{\text{usr}}(c), \qquad k = 0, 1, 2, \ldots, \tag{6.2}$$

where μ_k^{hr} are the moments in (4.1) and $\mu_k^{\text{usr}}(c)$ those in (6.1) although (6.2) 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 (6.2) are generated in `dig`-digit arithmetic by the Matlab command

```
mom=smom_lsrbinet(dig,N,c).
```

EXAMPLE 6.3 (The first 100 recurrence coefficients to 32 digits of the lower subrange Binet weight function on $[0, c]$ for $c = 1$). The Matlab command

```
[ab,dig]=dig_lsrbinet(N,c,dig0,dd,nofdig),
```

when run with $\texttt{N} = 100$, $\texttt{c} = 1$, $\texttt{dig0} = 520$, $\texttt{dd} = 4$, $\texttt{nofdig} = 32$, yields $\texttt{dig} = 528$. This large number of `dig` is due to extremely severe cancellation in (6.2) causing a loss of as many as 375 digits! The same or the direct command `ab=sr_lsrbinet(dig,nofdig,N,c)` with $\texttt{dig} = 528$, $\texttt{c} = 1$, produces the 100×2 array `ab` of the first 100 recurrence coefficients to 32 digits. They are displayed in the second and third plot of Figure 6.3, the first showing the lower subrange Binet weight function on $[0, 1]$. They are also made available in the textfile `coeff_lsrbinet.txt`; see also [8, 2.9.27].

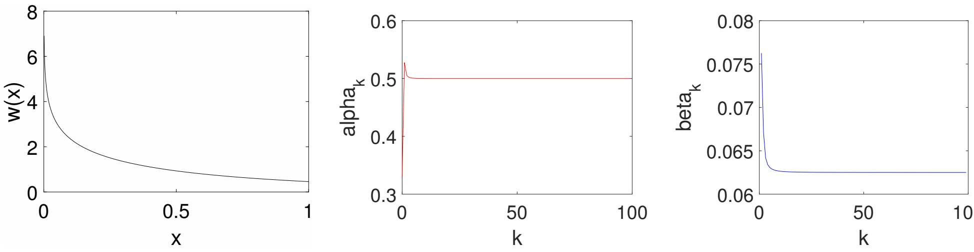


FIG. 6.3. *Lower subrange Binet weight function with $c = 1$ and its recurrence coefficients.*

6.2.2. The zeros of the orthogonal polynomials. By Theorem 2.6 of Section 2.2.2, we have

THEOREM 6.4. *All zeros of the lower subrange Binet polynomial π_n, orthogonal on $[0, c]$, are monotonically increasing as functions of c.*

Using the routine `dig_lsrbinet.m` with $\texttt{N} = 15$, it was found that $\texttt{dig} = 90$ digits are required to obtain the first 15 recurrence coefficients to an accuracy of 6 digits whenever $c \geq 1/10$. The zeros obtained are displayed in Figure 6.4; cf. `plot_zeros_lsrbinet.m`.

6.3. A lower symmetric subrange Binet weight function.

6.3.1. The recurrence coefficients. Here, the weight function (1.1) is supported on the interval $[-c, c]$, $0 < c < \infty$. The moments μ_k, therefore, are 0 if k is odd and twice of those in (6.2) if k is even, and they are generated by the routine `smom_lssrbinet.m`.

EXAMPLE 6.5 (The first 100 recurrence coefficients to 32 digits of the lower symmetric subrange Binet weight function on $[-c, c]$ for $c = 1$). The Matlab routine

```
dig_lssrbinet(N,c,dig0,dd,nofdig),
```

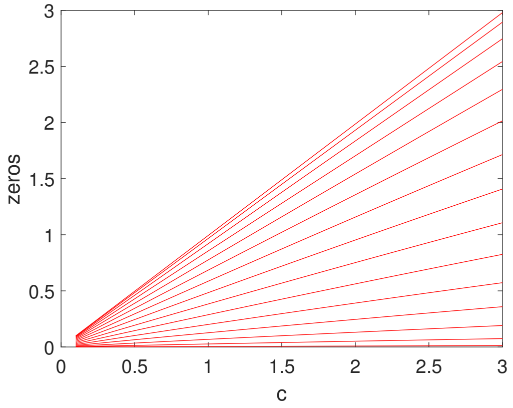


FIG. 6.4. *The zeros for $n = 15$ of the lower subrange Binet polynomial π_n, orthogonal on $[0, c]$, in dependence of the parameter c,* $0 < c \leq 3$.

run with $\texttt{N} = 100$, $\texttt{c} = 1$, $\texttt{dig0} = 460$, $\texttt{dd} = 4$, $\texttt{nofdig} = 32$, yields $\texttt{dig} = 468$ as the number of digits needeed to obtain the 100×2 array `ab` of the desired recurrence coefficients. The same array can be obtained by the command `ab=sr_lssrbinet(dig,nofdig,N,c)` with $\texttt{dig} = 468$, $\texttt{c} = 1$. The recurrence coefficients are displayed in the second and third plot of Figure 6.5, the first showing the lower symmetric subrange Binet weight function on $[-1, 1]$, and they are also made available to 32 digits in the textfile `coeff_lssrbinet.txt`; see also [8, 2.9.29].

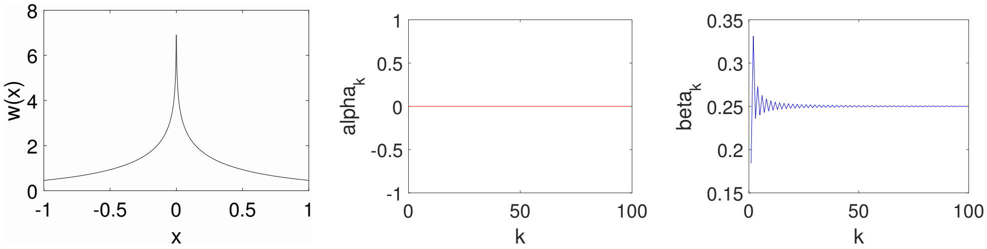


FIG. 6.5. *The lower symmetric subrange Binet weight function with* $c = 1$ *and its recurrence coefficients.*

6.3.2. The zeros of the orthogonal polynomials.

Since the lower symmetric subrange Binet weight function is symmetric on $[-c, c]$, we can apply the Remark 2.7 to Theorem 2.3 to obtain:

THEOREM 6.6. *All positive zeros of the lower symmetric subrange Binet polynomial π_n, orthogonal on $[-c, c]$, $0 < c < \infty$, are monotonically increasing as functions of c.*

Plots of the positive zeros for $n = 30$ are displayed in Figure 6.6; cf. the Matlab script `plot_zeros_lssrbinet.m`.

REFERENCES

[1] A. CVETKOVIĆ AND G. V. MILOVANOVIĆ, *The Mathematica package "OrthogonalPolynomials"*, Facta Univ. Ser. Math. Inform., 19 (2004), pp. 17–36.

[2] G. DAHLQUIST, *On summation formulas due to Plana, Lindelöf and Abel, and related Gauss–Christoffel rules, I.*, BIT, 37 (1997), pp. 256–295.

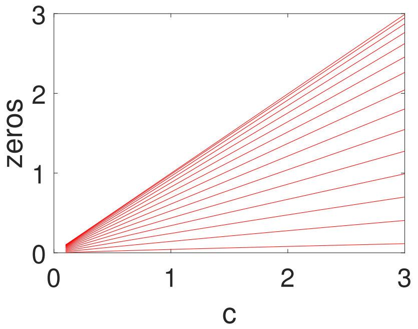


FIG. 6.6. *The positive zeros for $n = 30$ of the lower symmetric subrange Binet polynomial π_n, orthogonal on $[-c, c]$, in dependence of the parameter c, $0 < c \leq 3$.*

[3] ———, *On summation formulas due to Plana, Lindelöf and Abel, and related Gauss–Christoffel rules, II.*, BIT, 37 (1997), pp. 804–832.

[4] ———, *On summation formulas due to Plana, Lindelöf and Abel, and related Gauss–Christoffel rules, III.*, BIT, 39 (1999), pp. 51–78.

[5] W. GAUTSCHI, *Orthogonal Polynomials: Computation and Approximation*, Oxford University Press, New York, 2004.

[6] ———, *Orthogonal Polynomials in MATLAB: Exercises and Solutions*, Society of Industrial and Applied Mathematics, Philadelphia, 2016.

[7] ———, *On the zeros of subrange Jacobi polynomials*, Numer. Algorithms, 79 (2018), pp. 759–768.

[8] ———, *A Software Repository for Orthogonal Polynomials*, e-book, Society of Industrial and Applied Mathematics, Philadelphia, 2018.

[9] I. S. GRADSHTEYN AND I. M. RYZHIK. *Table of Integrals, Series, and Products*, 5th ed., Academic Press, Boston, 1994.

[10] K. S. KÖLBIG, *Nielsen's generalized polylogarithyms*, SIAM J. Math. Anal. 17 (1986), pp. 1232–1258.

[11] K. S. KÖLBIG, J. A. MIGNACO, AND E. REMIDDI, *On Nielsen's generalized polylogarithms and their numerical calculation*, Nordisk Tidskr. Informationsbehandling (BIT), 10 (1970), pp. 38–73.

[12] G. V. MILOVANOVIĆ, *Summation formulas of Euler–Maclaurin and Abel-Plana: Old and new results and applications*, in Progress in Approximation Theory and Applicable Complex Analysis , N. K. Govil, R. Mohapatra. M. Qazi, and G. Schmeisser, eds., Springer Optim. Appl., 117, Springer, Cham, 2017, pp. 429–461.

[13] G. V. MILOVANOVIĆ AND A. S. CVETKOVIĆ, *Special classes of orthogonal polynomials and corresponding quadratures of Gaussian type*, Math. Balkanica, 26 (2013), pp. 169–184.

[14] G. SZEGÖ, *Orthogonal Polynomials*, 4th ed., American Mathematical Society, Providence, 1975.

44.4. [223] (with Gradimir V. Milovanović) “Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure”

[223] (with Gradimir V. Milovanović) “Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure”, *Numer. Algorithms* **88**, 1233–1249 (2021).

Numerical Algorithms (2021) 88:1233–1249
https://doi.org/10.1007/s11075-021-01073-1

ORIGINAL PAPER

Orthogonal polynomials relative to a generalized Marchenko–Pastur probability measure

Walter Gautschi[1] · **Gradimir V. Milovanović**[2,3]

Received: 30 July 2020 / Accepted: 14 January 2021 / Published online: 18 March 2021

Abstract
The Marchenko–Pastur probability measure, of interest in the asymptotic theory of random matrices, is generalized in what appears to be a natural way. The orthogonal polynomials and their three-term recurrence relation for this generalized Marchenko–Pastur measure are obtained in explicit form, analytically as well as symbolically using Mathematica. Special cases involve Chebyshev polynomials of all four kinds. Supporting Matlab software is provided.

Keywords Orthogonal polynomials · Generalized Marchenko–Pastur measure · Three-term recurrence relation

Mathematics subject classification (2010) 15B52 · 3304 · 33C47

1 Introduction

In 1967 (see [8]), the Ukrainian mathematicians Vladimir Alexandrovich Marchenko (b. 1922) and Leonid Andreevich Pastur (b. 1937), working on the asymptotic theory of large random matrices, came to consider a probability measure now known as the Marchenko–Pastur measure or the Marchenko–Pastur law. Here, entirely out of curiosity and without any applications in mind, we generalize the measure in the same way as Jacobi polynomials are a generalization of Chebyshev polynomials (of the

✉ Walter Gautschi
wgautschi@purdue.edu

Gradimir V. Milovanović
gvm@mi.sanu.ac.rs

[1] Department of Computer Science, Purdue University, West Lafayette, IN 47907-2107, USA

[2] Serbian Academy of Sciences and Arts, Kneza Mihaila 35, Belgrade, 11000, Serbia

[3] Faculty of Sciences and Mathematics, University of Niš, Niš, 18000, Serbia

second kind). We are interested in the orthogonal polynomials relative to this generalized measure and in their three-term recurrence relation. Both are obtained explicitly in Sections 3 and 4. In special cases, the orthogonal polynomials are identified in Section 5 in terms of Chebyshev polynomials of all four kinds. Double-precision and variable-precision Matlab routines for generating the recurrence coefficients of the generalized Marchenko–Pastur measure are also provided.

2 The generalized Marchenko–Pastur measure

The Marchenko–Pastur measure, as formulated in [1, 7, 9], is supported on the interval $[a, b]$, where

$$a = (\sqrt{c} - 1)^2, \quad b = (\sqrt{c} + 1)^2, \quad c > 0, \tag{2.1}$$

and defined, if $c > 1$, by the density function

$$w(x) = \frac{1}{2\pi c} x^{-1}[(b - x)(x - a)]^{1/2} + \left(1 - \frac{1}{c}\right)\delta_0, \quad a \le x \le b, \tag{2.2}$$

where δ_0 is the Dirac delta function at $x = 0$ with mass 1, and, if $c < 1$, by

$$w(x) = \frac{1}{2\pi c} x^{-1}[(b - x)(x - a)]^{1/2}, \quad a \le x \le b. \tag{2.3}$$

A natural generalization is to replace the exponent 1/2 of $b - x$ and $x - a$ by α and β, respectively, as in the case of Jacobi measures, assuming $\alpha > -1$, $\beta > -1$. The constant $(2\pi)^{-1}$ multiplying $x^{-1}[(b - x)(x - a)]^{1/2}$ in (2.2) happens to be a normalization factor when $c > 1$. Likewise, $(2\pi c)^{-1}$ is the normalization factor when $c < 1$ (cf. (5.5)). Thus, our generalization of the Marchenko–Pastur measure is

$$w(x; \alpha, \beta, c) = \begin{cases} \dfrac{1}{c\,\mu_0} x^{-1}\,(b - x)^{\alpha}(x - a)^{\beta} + \left(1 - \dfrac{1}{c}\right)\delta_0 \ \text{if } c > 1, \\ \dfrac{1}{\mu_0} x^{-1}\,(b - x)^{\alpha}(x - a)^{\beta} \ \text{if } c < 1, \end{cases} \quad a \le x \le b, \tag{2.4}$$

where

$$\mu_0 = \mu_0(\alpha, \beta, c) = \int_a^b x^{-1}\,(b - x)^{\alpha}(x - a)^{\beta}\,\mathrm{d}x, \quad c > 0. \tag{2.5}$$

The weight function w in (2.4) is clearly a probability measure, for $0 < c < 1$ as well as for $c > 1$, in the sense that $\int_a^b w(x; \alpha, \beta, c)\mathrm{d}x = 1$. The case $c = 1$ can be transformed to a Jacobi weight function with Jacobi parameters $\alpha, \beta - 1$ (if $\beta > 0$). This case is classical and well known, and therefore will not be considered any further.

The monic polynomials orthogonal with respect to the weight function (2.4) will be denoted by $\pi_k(x)$, $k = 0, 1, 2, \ldots$. As is well known, they satisfy a three-term recurrence relation

$$\begin{aligned} \pi_{k+1}(x) &= (x - \alpha_k)\pi_k(x) - \beta_k\pi_{k-1}(x), \quad k = 0, 1, 2, \ldots, \\ \pi_{-1}(x) &= 0, \quad \pi_0(x) = 1. \end{aligned} \tag{2.6}$$

Our interest is in the coefficients α_k, β_k. We use the convention $\beta_0 = \int_a^b w(x;\alpha,\beta,c)\mathrm{d}x$.

Notice that, when $c < 1$, the factor $1/\mu_0$ in (2.4) is unimportant, as regards orthogonal polynomials, since it has no effect on them whatsoever. Not so when $c > 1$, where the constant $1/(c\mu_0)$ multiplies only the continuous part of the measure. A change of that constant will therefore also change the orthogonal polynomials.

3 Orthogonal polynomials and recurrence coefficients for (2.4) when $c > 1$

3.1 Orthogonal polynomials on $[-1, 1]$

Associated with the weight function (2.4) is the inner product

$$(u,v) = \int_a^b \left[\frac{1}{c\mu_0}x^{-1}(b-x)^\alpha(x-a)^\beta + \left(1-\frac{1}{c}\right)\delta_0\right]u(x)v(x)\mathrm{d}x, \qquad (3.1)$$

where μ_0 is given by (2.5). Changing variables

$$x = 2\sqrt{c}\,t + c + 1, \qquad (3.2)$$

mapping the interval $[-1, 1]$ onto $[a, b]$, the constant μ_0 and inner product (3.1) become

$$\mu_0 = (2\sqrt{c})^{\alpha+\beta}\int_{-1}^1 \frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t, \qquad (3.3)$$

respectively

$$(u,v)^* = \frac{1}{c}\,\frac{\displaystyle\int_{-1}^1 \frac{u(t)v(t)}{t+g}\,w^{(\alpha,\beta)}(t)\mathrm{d}t}{\displaystyle\int_{-1}^1 \frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t} + \left(1-\frac{1}{c}\right)u(-g)v(-g), \qquad (3.4)$$

where $w^{(\alpha,\beta)}(t)$ is the Jacobi weight function and

$$g = \frac{c+1}{2\sqrt{c}}. \qquad (3.5)$$

Note that g is invariant with respect to the transformation $c \mapsto 1/c$ and $g > 1$ for all $c \neq 1$.

We denote the monic polynomials orthogonal with respect to the inner product (3.4) by π_k^*. Inserting (3.2) into (2.6) and using $\pi_k(2\sqrt{c}\,t + c + 1) = (2\sqrt{c})^k\pi_k^*(t)$ yields the recurrence relation for the π_k^*,

$$\pi_{k+1}^*(t) = (t - \alpha_k^*)\pi_k^*(t) - \beta_k^*\pi_{k-1}^*(t), \qquad (3.6)$$

where

$$\alpha_k^* = \frac{\alpha_k - (c+1)}{2\sqrt{c}}, \quad \beta_k^* = \frac{\beta_k}{4c}, \quad k \geq 0. \qquad (3.7)$$

For later use, we obtain $\alpha_0^* = \alpha_0/(2\sqrt{c}) - g$ in (3.7) more explicitly, noting that

$$\alpha_0 = \int_a^b x\, w(x; \alpha, \beta, c)\mathrm{d}x = \frac{1}{c\mu_0}\int_a^b (b-x)^{\alpha}(x-a)^{\beta}\mathrm{d}x, \tag{3.8}$$

the term with the Dirac function being zero because of the factor x in the first integral of (3.8). Applying the change of variables (3.2) to the second integral in (3.8) as well as to the integral in (2.5) representing μ_0 gives

$$\alpha_0 = \frac{2^{\alpha+\beta+2}}{\sqrt{c}}\,\frac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}\left(\int_{-1}^{1}\frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t\right)^{-1}. \tag{3.9}$$

The integral in large parentheses can be identified with Mathematica to be

$$\int_{-1}^{1}\frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t = \frac{2^{\alpha+\beta+1}}{g-1}\,B(\alpha+1, \beta+1)\,{}_2F_1(1, \beta+1; \alpha+\beta+2; -2/(g-1)), \tag{3.10}$$

where B is Euler's beta integral and ${}_2F_1$ the hypergeometric function. Therefore, by (3.7) for $k = 0$,

$$\alpha_0^* = \frac{g-1}{c}\,\frac{1}{{}_2F_1(1, \beta+1; \alpha+\beta+2; -2/(g-1))} - g, \quad g = \frac{c+1}{2\sqrt{c}}. \tag{3.11}$$

Since π_k^* is orthogonal with respect to the inner product (3.4) to all polynomials of degree $k-1$, putting $u(t) = \pi_k^*(t)$ and $v(t) = v_j(t) = (t+g)^j$ in (3.4) gives

$$(\pi_k^*, v_j)^* = \frac{1}{c}\,\frac{\displaystyle\int_{-1}^{1}\pi_k^*(t)(t+g)^{j-1}w^{(\alpha,\beta)}(t)\mathrm{d}t}{\displaystyle\int_{-1}^{1}\frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t} + \left(1-\frac{1}{c}\right)\pi_k^*(-g)\delta_{j,0} = 0,$$
$$j = 0, 1, 2, \ldots, k-1, \tag{3.12}$$

where $\delta_{j,0}$ is the Kronecker delta. The relations for $j = 1, 2, \ldots, k-1$ imply that

$$\int_{-1}^{1}\pi_k^*(t)(t+g)^{j-1}w^{(\alpha,\beta)}(t)\mathrm{d}t = 0, \quad j = 1, 2, \ldots, k-1,$$

that is, π_k^* is orthogonal with respect to the Jacobi weight function to all polynomials of degree $\le k-2$, hence:

Theorem 1 *The monic orthogonal polynomials π_k^* relative to the inner product* (3.4) *are*

$$\pi_k^*(t) = \overset{\circ}{P}{}_k^{(\alpha,\beta)}(t) + \gamma_k \overset{\circ}{P}{}_{k-1}^{(\alpha,\beta)}(t), \quad k \ge 1, \tag{3.13}$$

where $\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)$ denote the monic Jacobi polynomials and γ_k are constants depending on α, β, c.

The constants γ_k can be determined from the orthogonality relation (3.12) with $j = 0$

$$\int_{-1}^{1} \frac{\pi_k^*(t)}{t+g}\, w^{(\alpha,\beta)}(t)\,\mathrm{d}t + (c-1)\pi_k^*(-g)\int_{-1}^{1} \frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t = 0, \tag{3.14}$$

as will be shown in the next subsection.

3.2 Determination of the coefficients γ_k

Let α_k^J, β_k^J be the recurrence coefficients for the monic Jacobi polynomials

$$\begin{aligned} &\overset{\circ}{P}{}_{k+1}^{(\alpha,\beta)}(t) = (t-\alpha_k^J)\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t) - \beta_k^J \overset{\circ}{P}{}_{k-1}^{(\alpha,\beta)}(t). \quad k = 0, 1, 2, \dots, \\ &\overset{\circ}{P}{}_0^{(\alpha,\beta)}(t) = 1, \quad \overset{\circ}{P}{}_{-1}^{(\alpha,\beta)}(t) = 0, \end{aligned} \tag{3.15}$$

where [5, Table 1.1][1]

$$\begin{aligned} \alpha_k^J &= \frac{\beta^2-\alpha^2}{(2k+\alpha+\beta)(2k+\alpha+\beta+2)}, \quad k \ge 0, \\ \beta_0^J &= \frac{2^{\alpha+\beta+1}\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}, \ \beta_k^J = \frac{4k(k+\alpha)(k+\beta)(k+\alpha+\beta)}{(2k+\alpha+\beta)^2((2k+\alpha+\beta)^2-1)}, \\ &\qquad k \ge 1, \end{aligned} \tag{3.16}$$

except in the case $k = 0$ and $\alpha + \beta = 0$, when $\alpha_0^J = -\alpha$, and in the case $k = 1$ and $\alpha + \beta + 1 = 0$, when $\beta_1^J = -2\alpha(\alpha + 1)$.

Theorem 2 *The recurrence coefficients α_k^*, β_k^* in (3.6), expressed in terms of the γ_k and the recurrence coefficients in (3.16), are*

$$\begin{aligned} &\alpha_0^* = \alpha_0^J - \gamma_1, \quad \alpha_k^* = \alpha_k^J + \gamma_k - \gamma_{k+1}, \quad k \ge 1, \\ &\beta_k^* = \beta_k^J + \gamma_k(\alpha_{k-1}^J - \alpha_k^J) - \gamma_k(\gamma_k - \gamma_{k+1}), \quad k \ge 1. \end{aligned} \tag{3.17}$$

Alternatively, for $k \ge 2$,

$$\beta_k^* = \frac{\gamma_k}{\gamma_{k-1}}\,\beta_{k-1}^J. \tag{3.18}$$

Proof Using (3.13) in (3.6), we have

$$\begin{aligned} \overset{\circ}{P}{}_{k+1}^{(\alpha,\beta)}(t) + \gamma_{k+1}\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t) = (t-\alpha_k^*)\left[\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t) + \gamma_k \overset{\circ}{P}{}_{k-1}^{(\alpha,\beta)}(t)\right] \\ - \beta_k^*\left[\overset{\circ}{P}{}_{k-1}^{(\alpha,\beta)}(t) + \gamma_{k-1}\overset{\circ}{P}{}_{k-2}^{(\alpha,\beta)}(t)\right]. \end{aligned} \tag{3.19}$$

[1] In the formula for β_0^J at the bottom of Table 1.1 of [5], the denominator should read $\Gamma(\alpha+\beta+2)$ instead of $\Gamma(\alpha+\beta+1)$.

Expressing $t\,\mathring{P}_k^{(\alpha,\beta)}$ and $t\,\mathring{P}_{k-1}^{(\alpha,\beta)}$ from (3.15) in terms of the monic Jacobi polynomials and putting the results into (3.19) yields

$$\begin{aligned}\left[\alpha_k^J-\alpha_k^*+\gamma_k-\gamma_{k+1}\right]\mathring{P}_k^{(\alpha,\beta)}(t)\;+\left[\beta_k^J-\beta_k^*+\gamma_k(\alpha_{k-1}^J-\alpha_k^*)\right]\mathring{P}_{k-1}^{(\alpha,\beta)}(t)&\\ +\left[\gamma_k\beta_{k-1}^J-\gamma_{k-1}\beta_k^*\right]\mathring{P}_{k-2}^{(\alpha,\beta)}(t)\equiv 0.&\end{aligned}\tag{3.20}$$

Since orthogonal polynomials are linearly independent, all coefficients in (3.20) must vanish, that is,

$$\begin{aligned}&\left.\begin{aligned}\alpha_k^*&=\alpha_k^J+\gamma_k-\gamma_{k+1},\\ \beta_k^*&=\beta_k^J+\gamma_k(\alpha_{k-1}^J-\alpha_k^*)=\beta_k^J+\gamma_k(\alpha_{k-1}^J-\alpha_k^J-\gamma_k+\gamma_{k+1})\end{aligned}\right\}\quad k\ge 1,\\ &\beta_k^*\;=\frac{\gamma_k}{\gamma_{k-1}}\,\beta_{k-1}^J,\;k\ge 2.\end{aligned}\tag{3.21}$$

This proves (3.17) for $k\ge 1$ and (3.18).

Noting that by (3.13) there holds $\pi_1^*(t)=\mathring{P}_1^{(\alpha,\beta)}(t)+\gamma_1$, and $\mathring{P}_1^{(\alpha,\beta)}(t)=t-\alpha_0^J$ by (3.15), we get $\pi_1^*(t)=t-\alpha_0^J+\gamma_1$. On the other hand, by (3.6), $\pi_1^*(t)=t-\alpha_0^*$, so that $\alpha_0^*=\alpha_0^J-\gamma_1$, which is the first relation in (3.17). □

Now the first relation in (3.17), together with (3.11) and (3.16) for $k=0$, yields

$$\gamma_1=\frac{\beta-\alpha}{\alpha+\beta+2}-\frac{g-1}{c}\,\frac{1}{{}_2F_1(1,\beta+1;\alpha+\beta+2;-2/(g-1))}+gg=\frac{c+1}{2\sqrt{c}}\,.\tag{3.22}$$

Let

$$p_k=\int_{-1}^{1}\frac{\mathring{P}_k^{(\alpha,\beta)}(t)}{t+g}\,w^{(\alpha,\beta)}(t)\mathrm{d}t+(c-1)\mathring{P}_k^{(\alpha,\beta)}(-g)\int_{-1}^{1}\frac{w^{(\alpha,\beta)}(t)}{t+g}\,\mathrm{d}t,\qquad k=0,1,2,\ldots,\quad c>1,\tag{3.23}$$

where the integral on the far right is known; see (3.10). If in (3.12) with $j=0$, we replace π_k^* by (3.13) and then solve for γ_k yields the remaining γ's,

$$\gamma_k=-\frac{p_k}{p_{k-1}},\quad k=2,3,\ldots\,.\tag{3.24}$$

To obtain a recurrence relation for the p_k, note first of all that from (3.23), we have

$$p_0=c\,I,\quad I=\int_{-1}^{1}\frac{w^{(\alpha,\beta}(t)}{t+g}\,\mathrm{d}t,\quad c>1.\tag{3.25}$$

Next, divide both sides of (3.15) by $t+g$ and integrate from -1 to 1 with weight function $w^{(\alpha,\beta)}$. We obtain, using (3.23) with k replaced by $k-1$, k, $k+1$, and (3.15) with $t=-g$,

$$\begin{aligned} p_{k+1} &= \int_{-1}^{1} \frac{t\,\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t - \alpha_k^J p_k - \beta_k^J p_{k-1} \\ &+ (c-1)I\left[\overset{\circ}{P}{}_{k+1}^{(\alpha,\beta)}(-g) + \alpha_k^J \overset{\circ}{P}{}_k^{(\alpha,\beta)}(-g) + \beta_k^J \overset{\circ}{P}{}_{k-1}^{(\alpha,\beta)}(-g)\right] \\ &= \int_{-1}^{1} \frac{t\,\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t - \alpha_k^J p_k - \beta_k^J p_{k-1} - (c-1)I\, g\overset{\circ}{P}{}_k^{(\alpha,\beta)}(-g) \end{aligned}$$

or, writing $t/(t+g) = 1 - g/(t+g)$,

$$p_{k+1} = \int_{-1}^{1} \overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)\, w^{(\alpha,\beta)}(t)\mathrm{d}t - (g+\alpha_k^J)p_k - \beta_k^J p_{k-1}.$$

Therefore, by orthogonality,

$$p_{k+1} = \frac{2^{\alpha+\beta+1}\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}\,\delta_{k,0} - (g+\alpha_k^J)p_k - \beta_k^J p_{k-1}, \quad k=0,1,2,\ldots, \tag{3.26}$$

where $\delta_{k,0}$ is the Kronecker delta and $p_{-1}=0$. If we restrict k to $k\geq 1$, the inhomogeneous term in (3.26) disappears, and we have

$$p_{k+1} + (g+\alpha_k^J)p_k + \beta_k^J p_{k-1} = 0, \quad k=1,2,3,\ldots. \tag{3.27}$$

3.3 The three-term recurrence relation (3.27)

In this subsection, it does not matter whether $c>1$ or $c<1$ since the parameter g in (3.27) is invariant with respect to the transformation $c \mapsto 1/c$.

The difference equation (3.27) is of Poincaré type, i.e., its coefficients

$$g+\alpha_k^J \to g, \quad \beta_k^J \to 1/4$$

have finite limits as $k\to\infty$, and the characteristic polynomial is

$$z^2 + \frac{c+1}{2\sqrt{c}}z + \frac{1}{4}. \tag{3.28}$$

The two zeros

$$z_1 = \frac{-1}{2\sqrt{c}}, \quad z_2 = \frac{-\sqrt{c}}{2},$$

having different moduli, implies (cf. [4, Theorem 2.2]) that there are two linearly independent solutions $p_k^{(1)}$, $p_k^{(2)}$ of (3.27) such that

$$\frac{p_{k+1}^{1)}}{p_k^{(1)}} \to \frac{-1}{2\sqrt{c}}, \quad \frac{p_{k+1}^{(2)}}{p_k^{(2)}} \to \frac{-\sqrt{c}}{2} \quad \text{as } k\to\infty. \tag{3.29}$$

One is dominant and the other minimal (see [4, p. 25] for terminology).

Theorem 3 *Let*

$$f_k = \int_{-1}^{1} \frac{\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t, \quad g_k = \overset{\circ}{P}{}_k^{(\alpha,\beta)}(-g), \quad k = 0, 1, 2, \ldots . \tag{3.30}$$

Both f_k and g_k are solutions of the difference equation (3.27)*, the first being minimal and the other dominant.*

Proof The fact that f_k and g_k are solutions of (3.27) is readily verified. To show minimality of f_k, it suffices to show that

$$\lim_{k\to\infty} r_k = 0, \tag{3.31}$$

where

$$r_k = \frac{1}{\overset{\circ}{P}{}_k^{(\alpha,\beta)}(-g)} \int_{-1}^{1} \frac{\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t. \tag{3.32}$$

Since normalization of the Jacobi polynomials in (3.32) is irrelevant, we may drop the circle on top of the P's. Using the Cauchy–Schwarz inequality, we then have

$$\begin{aligned} |r_k| &= \frac{1}{|P_k^{(\alpha,\beta)}(-g)|} \left| \int_{-1}^{1} \frac{P_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t \right| \\ &\le \frac{1}{|P_k^{(\alpha,\beta)}(-g)|} \, ||P_k^{(\alpha,\beta)}||\, ||(\cdot\ + g)^{-1}||, \end{aligned}$$

where $||u|| = \sqrt{\int_{-1}^{1} u^2(t) w^{(\alpha,\beta)}(t)\mathrm{d}t}$. Since $||P_k^{(\alpha,\beta)}|| = \mathrm{O}(1/\sqrt{k})$ (cf. [10, p. 132]), $||(\cdot\ + g)^{-1}||$ is a positive constant not depending on k, and $|P_k^{(\alpha,\beta)}(-g)| = P_k^{(\beta,\alpha)}(g) \to \infty$ when $g > 1$ (cf. [15, Theorem 8.21.7]), there indeed holds (3.31). □

3.4 Recurrence coefficients

The three-term recurrence relation (2.6) of interest can now be obtained from (3.7),

$$\begin{aligned} \alpha_k &= 2\sqrt{c}\,\alpha_k^* + c + 1, \quad k \ge 0, \\ \beta_0 &= 1, \quad \beta_k = 4\,c\,\beta_k^*, \quad k \ge 1, \end{aligned} \tag{3.33}$$

with α_0^* given by (3.11) and α_k^*, β_k^*, $k \ge 1$, by (3.21).

Theorem 4 *The recurrence coefficients α_k, β_k satisfy*

$$\lim_{k\to\infty} \alpha_k = 1 + c, \quad \lim_{k\to\infty} \beta_k = c. \tag{3.34}$$

Proof The coefficients α_k^*, β_k^* in (3.6) have limits 0 resp. 1/4 as $k \to \infty$ (see [14], [12], [3, Theorem 4]). Therefore, letting $k \to \infty$ in (3.33) yields (3.34). □

4 Orthogonal polynomials and recurrence coefficients for (2.4) when $c < 1$

The analysis is essentially the same as in the previous section, once the delta function and the factor $1/c$ in (2.4) have been removed. This requires, however, a few adjustments. Specifically, delete the factor c multipying ${}_2F_1$ both in (3.22) and (3.11). Moreover, replace p_k in (3.23) by

$$p_k = \int_{-1}^{1} \frac{\overset{\circ}{P}{}_k^{(\alpha,\beta)}(t)}{t+g}\, w^{(\alpha,\beta)}(t)\mathrm{d}t, \quad k = 0, 1, 2, \ldots, \quad c < 1, \tag{4.1}$$

which, as shown in Theorem 3, is a minimal solution of (3.27).

The coefficients γ_k, $k \geq 2$, in Theorem 1 are again determined by (3.24), but now with p_k given in (4.1).

Theorem 5 *The coefficients γ_k in Theorem 1 satisfy*

$$\lim_{k\to\infty} \gamma_k = \sqrt{c}/2, \quad c \neq 1. \tag{4.2}$$

Proof If $c > 1$, the limit (4.2), by (3.24), is the limit

$$-\lim_{k\to\infty} \frac{p_{k+1}}{p_k} \tag{4.3}$$

for the dominant solution p_k of (3.27) in (3.23), that is, $\sqrt{c}/2$, and if $c < 1$, the limit (4.3) for the minimal solution in (4.1), that is, again $\sqrt{c}/2$. □

Remarks to Theorem 5

1. When $\alpha = \beta = 1/2$ and both $c > 1$ and $c < 1$, the limit in (4.2) is attained instantaneously, that is, $\gamma_1 = \gamma_2 = \gamma_3 = \cdots$. The same is true if $\alpha = -\beta = \pm 1/2$ and $c < 1$.
2. When $\alpha = \beta = -1/2$ and $c < 1$, the limit in (4.2) is attained almost instantaneously, that is, $\gamma_2 = \gamma_3 = \cdots$. (High-precision computation may be required to make this visible.)

Computation of the minimal solution p_k of the recurrence relation (3.27) in forward direction is unstable, more so the smaller c. A stable algorithm to compute p_k for $0 \leq k \leq N$ is as follows (cf. [4, Eq. (3.9)]). Select an integer $\nu > N$ and apply the following backward/forward recursion,

$$\begin{aligned} &r_\nu^{(\nu)} = -\sqrt{c}/2, \quad r_{k-1}^{(\nu)} = \frac{-\beta_k^J}{g + \alpha_k^J + r_k^{(\nu)}}, \quad k = \nu, \nu-1, \ldots 1, \\ &f_0^{(\nu)} = p_0, \quad f_k^{(\nu)} = r_{k-1}^{(\nu)} f_{k-1}^{(\nu)}, \quad k = 1, 2, 3, \ldots, N, \end{aligned} \tag{4.4}$$

where (cf. [4, Eq. (3.10)]) $r_\nu^{(\nu)}$ is taken to be the right-hand limit in (3.29). Moreover (cf. (3.10)),

$$p_0 = \frac{2^{\alpha+\beta+1}}{g-1} \frac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)}\, {}_2F_1(1, \beta+1; \alpha+\beta+2; -2/(g-1)).$$

Table 1 Value of ν required in (4.4) to obtain 13-digit accuracy when using Matlab double-precision arithmetic in the case of the Legendre weight function

c	N	ν	c	N	ν	c	N	ν
.01	10	14	.1	10	19	.5	10	40
	20	24		20	28		20	49
	50	54		50	58		50	78
	100	103		100	107		100	126
.05	10	17	.3	10	27	.8	10	103
	20	26		20	36		20	112
	50	56		50	66		50	140
	100	105		100	115		100	187

Since $f_k^{(\nu)} \to p_k, k = 0, 1, 2, \ldots, N$, as $\nu \to \infty$, one applies (4.4) for a sequence of increasing ν-values, $\nu > N$, until two successive $f_k^{(\nu)}$ agree to the desired precision. In the special cases of Chebyshev weight functions of all four kinds, this is not necessary since the $r_{k-1}^{(\nu)}$, owing to the essentially constant values of α_k^J, β_k^J in these cases, do not depend on ν nor on k (except for the single value of r_0 in the case of the Chebyshev weight function of the first kind). Besides, the p_k are explicitly known in these cases (cf. Section 5).

When the algorithm (4.4) is run in Matlab double precision, the smallest value of ν guaranteeing 13-digit accuracy is shown in Table 1 for the case $\alpha = \beta = 0$ and for selected values of c and N.

In the case of c very close to 1 (hence ν very large), one may get away with forward recursion and only moderate increase, if any, of the working precision.

The procedure described in this and the previous section is implemented in the Matlab functions `GMP.m` and `sGMP.m` in double resp. variable precision; visit https://www.cs.purdue.edu/archives/2002/wxg/codes/GMP.html.

5 Examples

The general results of Sections 3 and 4 are here specialized to the cases $\alpha, \beta = \pm 1/2$ and $\alpha = \beta = 0$. In the former case, the desired recurrence coefficients α_n, β_n are obtained by the procedures of Sections 3 and 4 in symbolic form as functions of n and c, making use of the symbolic capabilities of Mathematica. In the latter case, numerical values are provided for $0 \le n \le 9$, $c = 4$, and $c = 1/4$.

5.1 Chebyshev weight function of the first kind

Here, $\alpha = \beta = -1/2$. The hypergeometric function needed in (3.10), (3.11), and (3.22) is [13, Eq. 15.4.6]:

$$_2F_1(1, 1/2; 1; -x) = {_2F_1}(1/2, 1; 1; -x) = (1 + x)^{-1/2},$$

and therefore, since $x = 2/(g-1) = 4\sqrt{c}/(\sqrt{c}-1)^2$,

$$ {}_2F_1(1, 1/2, ; 1; -2/(g-1)) = \begin{cases} (\sqrt{c}-1)/(\sqrt{c}+1) & \text{if } c > 1, \\ (1-\sqrt{c})/(1+\sqrt{c}) & \text{if } c < 1. \end{cases} $$

Using (3.3) and (3.10), we get

$$ \mu_0 = \begin{cases} \pi/(c-1) & \text{if } c > 1, \\ \pi/(1-c) & \text{if } c < 1. \end{cases} \tag{5.1} $$

With regard to the recurrence relation (3.26), we have, by (3.25), (3.10), and the first paragraph of Section 4,

$$ p_0 = \begin{cases} 2\,c^{3/2}\pi/(c-1) & \text{if } c > 1, \\ 2\sqrt{c}\,\pi/(1-c) & \text{if } c < 1. \end{cases} $$

From (3.26) with $k = 0$ and $k = 1$, noting that $\alpha_k^J = 0$, $\beta_1^J = 1/2$, we get

$$ p_1 = \begin{cases} -(1+c^2)\,\pi/(c-1) & \text{if } c > 1, \\ -2\,c\,\pi/(1-c) & \text{if } c < 1, \end{cases} $$

$$ p_2 = \begin{cases} (c^3 - c^2 + c + 1)\pi/(2\sqrt{c}(c+1)) & \text{if } c > 1, \\ c^{3/2}\pi/(1-c) & \text{if } c < 1. \end{cases} $$

Finally, since $\beta_k^J = 1/4$ for $k \geq 2$,

$$ p_{k+1} = -\frac{c+1}{2\sqrt{c}}\,p_k - \frac{1}{4}\,p_{k-1}, \quad k = 2, 3, \ldots, \tag{5.2} $$

which is a three-term recurrence relation with constant coefficients and characteristic polynomial (3.28). This allows us to explicitly obtain the solution p_k of (5.2) with the above starting values p_1, p_2. The result, when $c > 1$, is

$$ p_k = 2\pi\sqrt{c}\,(-1)^k \left[\left(\frac{\sqrt{c}}{2}\right)^k + \frac{c+1}{c-1}\left(\frac{1}{2\sqrt{c}}\right)^k \right], \quad c > 1, \quad k = 3, 4, \ldots, \tag{5.3} $$

and when $c < 1$

$$ p_k = \frac{4\pi\sqrt{c}}{1-c}(-1)^k \left(\frac{\sqrt{c}}{2}\right)^k, \quad c < 1, \quad k = 3, 4, \ldots. \tag{5.4} $$

Once we have p_k, we get all the γ_k from (3.24) and (3.22), which in turn gives

$$ \alpha_0^* = \begin{cases} -(c^2+1)/(2\,c^{3/2}) & \text{if } c > 1, \\ -1/\sqrt{c} & \text{if } c < 1 \end{cases} $$

from (3.11) and α_k^*, β_k^* for $k \geq 1$ from (3.21), and thus α_k, β_k from (3.33). All this can be done numerically as well as symbolically.

The symbolic results are displayed in Table 2.

It is evident again, as was proved in Theorem 4, that α_n and β_n have limits $c+1$ resp. c as $n \to \infty$, but what is noteworthy, and follows from Remark 2 to Theorem 5, is that in the case $c < 1$, these limits are attained almost instantaneously.

In order to affirm the validity of the data shown in Table 2 and in the subsequent Tables 3, 4, and 5, we used the modified Chebyshev algorithm [5, Section 2.1.7] based on the modified moments $m_k = \int_a^b w(x; \alpha, \beta, c)(x-a)^k \mathrm{d}x$, $k = 0, 1, 2, \ldots$,

Table 2 Recurrence coefficients for the measure (2.4) in the case of the Chebyshev weight function of the first kind

	n	$[\alpha_n, \beta_n]$
$c > 1$	0	$\left[\frac{c-1}{c}, 1\right]$
	1	$\left[c+2+\frac{(c-1)^2}{c(c^2+1)}, \frac{(c-1)(c^2+1)}{c^2}\right]$
	> 1	$\left[c+1-\frac{(c-1)^3(c+1)c^{n-1}}{(c^n-c^{n-1}+c+1)(c^{n+1}-c^n+c+1)},\right.$ $\left. c\,\frac{(c^{n-1}-c^{n-2}+c+1)(c^{n+1}-c^n+c+1)}{(c^n-c^{n-1}+c+1)^2}\right]$
$c < 1$	0	$[1-c, 1]$
	1	$[2c+1, 2c(1-c)]$
	> 1	$[c+1, c]$

and implemented in the first author's SOPQ package [6, Section 2.1] as well as in the second author's Mathematica package *Orthogonal polynomials* [2, 11]. When run in sufficiently high precision (to combat possible instabilities), the results obtained are in agreement with those obtained by evaluating the symbolic expressions in Table 2 for selected values of c, both with $c > 1$ and $c < 1$, and for n as large as 100.

5.2 Chebyshev weight function of the second kind

This is the original case $\alpha = \beta = 1/2$. The hypergeometric function needed here is [13, Eq. 15.4.17]

$$ {}_2F_1(1, 3/2; 3; -x) = 4\left(1+\sqrt{1+x}\right)^{-2}, $$

where

$$ x = \frac{2}{g-1} = \frac{4\sqrt{c}}{(\sqrt{c}-1)^2}. $$

Table 3 Recurrence coefficients for the measure (2.4) in the case of the Chebyshev weight function of the second kind

	n	$[\alpha_n, \beta_n]$
$c > 1$	0	$[1, 1]$
	> 0	$[c+1, c]$
$c < 1$	0	$[1, 1]$
	> 0	$[c+1, c]$

A short calculation yields

$$ {}_2F_1(1, 3/2; 3; -2/(g-1)) = \begin{cases} (\sqrt{c}-1)^2/c & \text{if } c > 1, \\ (\sqrt{c}-1)^2 & \text{if } c < 1. \end{cases} $$

Therefore, by (3.3) and (3.10),

$$ \mu_0 = \begin{cases} 2\pi & \text{if } c > 1, \\ 2\pi\, c & \text{if } c < 1. \end{cases} \tag{5.5} $$

The recurrence relation (5.2) in this case is

$$ p_{k+1} = -\frac{c+1}{2\sqrt{c}}\, p_k - \frac{1}{4}\, p_{k-1}, \quad k = 1, 2, 3, \dots, \tag{5.6} $$

where

$$ p_0 = \sqrt{c}\,\pi, \quad p_1 = -c\,\pi/2, $$

both, for $c > 1$ and $c < 1$. Hence, similarly as in Section 5.1,

$$ p_k = \pi\,\sqrt{c}\,(-1)^k \left(\frac{\sqrt{c}}{2}\right)^k, \quad k = 2, 3, 4, \dots, \quad c \neq 1 \tag{5.7} $$

The recurrence coefficients α_k^*, β_k^*, $k \geq 0$, and thus also the α_k, β_k, can now be obtained as in the previous subsection. Table 3 displays the symbolic results. Convergence in (3.34) is now truly instantaneous, not only in the case $c < 1$ but also when $c > 1$; cf. Remark 1 to Theorem 5.

5.3 Chebyshev weight function of the third kind

In this case, $\alpha = -1/2$, $\beta = 1/2$, and the relevant hypergeometric function is [13, Eq. 15.4.18]

$$ {}_2F_1(1, 3/2; 2; -x) = \frac{2}{1 + x + \sqrt{1+x}}, $$

which, for $x = 2/(g-1) = 4\sqrt{c}/(\sqrt{c}-1)^2$ becomes

$$ {}_2F_1(1, 3/2; 2; -2/(g-1)) = \begin{cases} (\sqrt{c}-1)^2/(\sqrt{c}\,(1+\sqrt{c})) & \text{if } c > 1, \\ (\sqrt{c}-1)^2/(1+\sqrt{c}) & \text{if } c < 1. \end{cases} $$

Therefore, by (3.3) and (3.10),

$$ \mu_0 = \begin{cases} 2\pi/(1+\sqrt{c}) & \text{if } c > 1, \\ 2\pi\sqrt{c}/(1+\sqrt{c}) & \text{if } c < 1. \end{cases} \tag{5.8} $$

From (3.25), (3.10), and from (3.26) with $k = 0$, noting that $\alpha_0^J = 1/2$, we have

$$ p_0 = \begin{cases} 2\pi\, c/(1+\sqrt{c}) & \text{if } c > 1, \\ 2\pi\,\sqrt{c}/(1+\sqrt{c}) & \text{if } c < 1, \end{cases} \qquad p_1 = \begin{cases} \pi\,(1 - c - c^{3/2})/(1+\sqrt{c}) & \text{if } c > 1, \\ -\pi\, c/(1+\sqrt{c}) & \text{if } c < 1, \end{cases} \tag{5.9} $$

and

$$ p_{k+1} = -\frac{c+1}{2\sqrt{c}}\, p_k - \frac{1}{4}\, p_{k-1}, \quad k = 1, 2, 3, \dots. \tag{5.10} $$

Table 4 Recurrence coefficients for the measure (2.4) in the case of the Chebyshev weight function of the third kind

	n	$[\alpha_n, \beta_n]$
$c > 1$	0	$\left[\frac{\sqrt{c}+1}{\sqrt{c}}, 1\right]$
	1	$\left[c+1+\frac{(\sqrt{c}+1)(c-1)^2}{\sqrt{c}\,(c^{3/2}+c-1)}, \frac{(\sqrt{c}+1)(c^{3/2}+c-1)}{c}\right]$
	> 1	$\left[c+1+\frac{(\sqrt{c}+1)(c-1)^2c^{n-1}}{(c^{n-1/2}+c^{n-1}-1)(c^{n+1/2}+c^n-1)},\right.$ $\left.c\,\frac{(c^{n-3/2}+c^{n-2}-1)(c^{n+1/2}+c^n-1)}{(c^{n-1/2}+c^{n-1}-1)^2}\right]$
$c < 1$	0	$[1+\sqrt{c}, 1]$
	1	$[c+1, c\,(1+\sqrt{c})]$
	> 1	$[c+1, c]$

When $c > 1$, the solution of (5.10) with starting values (5.9) is

$$p_k = \frac{2\pi\sqrt{c}}{1+\sqrt{c}}(-1)^k\left[(1+\sqrt{c})\left(\frac{\sqrt{c}}{2}\right)^k - \left(\frac{1}{2\sqrt{c}}\right)^k\right], \quad k = 2, 3, 4, \ldots, \tag{5.11}$$

while for $c < 1$, it is

$$p_k = \frac{2\pi\sqrt{c}}{1+\sqrt{c}}(-1)^k\left(\frac{\sqrt{c}}{2}\right)^k, \quad k = 2, 3, 4, \ldots. \tag{5.12}$$

In the same way as before, knowledge of p_k eventually yields the recurrence coefficients α_k, β_k. They are shown, as functions of c, in Table 4.

5.4 Chebyshev weight function of the fourth kind

This is the case $\alpha = 1/2$, $\beta = -1/2$, where [13, Eq. 15.4.17]

$${}_2F_1(1, 1/2; 2; -x) = {}_2F_1(1/2, 1; 2; -x) = 2\left(1+\sqrt{1+x}\right)^{-1},$$

and, with $x = 2/(g-1) = 4\sqrt{c}/(\sqrt{c}-1)^2$

$${}_2F_1(1, 1/2; 2; -2/(g-1)) = \begin{cases} (\sqrt{c}-1)/\sqrt{c} & \text{if } c > 1, \\ 1-\sqrt{c} & \text{if } c < 1. \end{cases}$$

By (3.3) and (3.10), there follows

$$\mu_0 = \begin{cases} 2\pi/(\sqrt{c}-1) & \text{if } c > 1, \\ 2\pi\sqrt{c}/(1-\sqrt{c}) & \text{if } c < 1. \end{cases} \tag{5.13}$$

Table 5 Recurrence coefficients for the measure (2.4) in the case of the Chebyshev weight function of the fourth kind

	n	$[\alpha_n, \beta_n]$
$c > 1$	0	$\left[\dfrac{\sqrt{c}-1}{\sqrt{c}}, 1\right]$
	1	$\left[c+1-\dfrac{(\sqrt{c}-1)(c-1)^2}{\sqrt{c}\,(c^{3/2}-c+1)}, \dfrac{(\sqrt{c}-1)(c^{3/2}-c+1)}{c}\right]$
	> 1	$\left[c+1-\dfrac{(\sqrt{c}-1)(c-1)^2\,c^{n-1}}{(c^{n-1/2}-c^{n-1}+1)(c^{n+1/2}-c^n+1)}, c\,\dfrac{(c^{n-3/2}-c^{n-2}+1)(c^{n+1/2}-c^n+1)}{(c^{n-1/2}-c^{n-1}+1)^2}\right]$
$c < 1$	0	$[1-\sqrt{c}, 1]$
	1	$[c+1, c(1-\sqrt{c})]$
	> 1	$[c+1, c]$

The recurrence relation (3.26), for $k \geq 1$, is

$$p_{k+1} = -\frac{c+1}{2\sqrt{c}}\,p_k - \frac{1}{4}\,p_{k-1}, \quad k = 1, 2, 3, \ldots, \tag{5.14}$$

where, as in Section 5.3, but with $\alpha_0^J = -1/2$,

$$p_0 = \begin{cases} 2\pi\, c/(\sqrt{c}-1) \ \text{if } c > 1, \\ 2\pi\sqrt{c}/(1-\sqrt{c}) \ \text{if } c < 1, \end{cases} \quad p_1 = \begin{cases} -(c^{3/2}-c+1)\pi/(\sqrt{c}-1) \ \text{if } c > 1, \\ -c\,\pi/(1-\sqrt{c}) \ \text{if } c < 1. \end{cases}$$

The explicit solution of (5.14), when $c > 1$, is

$$p_k = 2\pi\,\sqrt{c}\,(-1)^k\left[\left(\frac{\sqrt{c}}{2}\right)^k + \frac{1}{\sqrt{c}-1}\left(\frac{1}{2\sqrt{c}}\right)^k\right], \quad k = 2, 3, 4, \ldots, \tag{5.15}$$

and when $c < 1$,

$$p_k = \frac{2\pi\,\sqrt{c}}{1-\sqrt{c}}(-1)^k\left(\frac{\sqrt{c}}{2}\right)^k, \quad k = 2, 3, 4, \ldots. \tag{5.16}$$

The desired recurrence coefficients α_k, β_k can now be obtained as in the three previous subsections. They are displayed as functions of c in Table 5.

5.5 Legendre weight function

Here, $\alpha = \beta = 0$, and by (3.3):

$$\mu_0 = \log\left(\frac{\sqrt{c}+1}{\sqrt{c}-1}\right)^2, \quad c \neq 1. \tag{5.17}$$

Table 6 Recurrence coefficients for the measure (2.4), $c = 4$, in the case of the Legendre weight function

n	α_n	β_n
0	.910239226626837	1.00000000000000
1	5.39383052925145	3.72266068344396
2	4.85035318634382	4.81975085119498
3	4.91545178723217	4.43942238158180
4	4.96405975959264	4.20474171054654
5	4.98480915733151	4.10065191759388
6	4.99299025623952	4.05587961485571
7	4.99635149648618	4.03506402338317
8	4.99786955634723	4.02419082363368
9	4.99863659426219	4.01782940485962

The recurrence relation (3.26), for $k \geq 1$, is

$$p_{k+1} = -\frac{c+1}{2\sqrt{c}}\, p_k - \frac{k^2}{4k^2-1}\, p_0, \quad k = 1, 2, 3, \dots, \tag{5.18}$$

where

$$p_0 = \log\left(\frac{\sqrt{c}+1}{\sqrt{c}-1}\right)^2 \times \begin{cases} c \text{ if } c > 1, \\ 1 \text{ if } c < 1, \end{cases} \quad p_1 = 2 - (c+1)/(2\sqrt{c})\, p_0,$$

and the desired recurrence coefficients α_k, β_k follow as described in Section 3.4: first, α_0^* and α_0 are obtained from (3.7) with $k = 0$ in combination with (3.9) (where the integral is equal to the above value of μ_0); this gives γ_1 from the first relation in (3.17), the remaining γ's coming from (3.24); finally, α_k^*, β_k^* for $k \geq 1$ are obtained from (3.21), and thus α_k, β_k from (3.33).

Table 7 Recurrence coefficients for the measure (2.4), $c = 1/4$, in the case of the Legendre weight function

n	α_n	β_n
0	.910239226626837	1.00000000000000
1	1.32084322743042	.309263583593324
2	1.26054739936794	.263830285912563
3	1.25362002567957	.256207550577421
4	1.25168096856576	.253540027022905
5	1.25091993817511	.252292444551082
6	1.25055903507149	.251607291494291
7	1.25036545798308	.251190105642641
8	1.25025216974473	.250917040883073
9	1.25018139891252	.25072846766016

An attempt to generate the recurrence coefficients α_n, β_n in symbolic form proved to be unfeasible because of the rapidly increasing complexity of the symbolic expressions for the α_n and β_n as n increases. We therefore restrict ourselves to obtaining numerical values of the first ten recurrence coefficients for $c = 4$ and $c = 1/4$ using the routine GMP.m. They are shown in Table 6 for $c = 4$ and in Table 7 for $c = 1/4$.

Convergence in Table 6 is seen to be monotone increasing for α_n and monotone decreasing for β_n, except at the very beginning. In Table 7, α_n and β_n both decrease monotonically.

Acknowledgements The authors are indebted to Walter Van Assche for helpful remarks and for the references [3, 12, 14].

Funding The work of the second author was supported in part by the Serbian Academy of Sciences and Arts (Φ-96).

References

1. Chafai, D.: http://djalil.chafai.net/blog/2011/01/29/the-marchenko-pastur-law/
2. Cvetković, A.S., Milovanović, G.V.: The Mathematica package "Orthogonal Polynomials". Facta Univ. Ser. Math. Inform. **19**, 17–36 (2004)
3. Denisov, S.A.: On Rakhmanov's theorem for Jacobi matrices. Proc. Amer. Math. Soc. **132**(3), 847–852 (2004)
4. Gautschi, W.: Computational aspects of three-term recurrence relations. SIAM Rev. **9**, 32–82 (1967)
5. Gautschi, W.: Orthogonal Polynomials: Computation and Approximation. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
6. Gautschi, W.: Orthogonal polynomials in MATLAB: exercises and solutions. Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2016)
7. Jonsson, D.: Some limit theorems for the eigenvalues of a sample covariance matrix. J. Multivariate Anal. **12**, 1–38 (1982)
8. Marchenko, V.A., Pastur, L.A.: Distribution of eigenvalues for some sets of random matrices (Russian). Mat. Sb. **72**, 507–536 (1967). [English translation in Math. USSR-Sbornik 1 (1967), 457–453.]
9. Marco, A., Martinez, J.-J.: A total positivity property of the Marchenko–Pastur law. Electr. J. Linear Algebra **30**, 105–117 (2015)
10. Mastroianni, G., Milovanović, G.V.: Interpolation Processes: Basic Theory and Applications. Springer Monographs in Mathematics. Springer, Berlin (2008)
11. Milovanović, G.V., Cvetković, A.S.: Special classes of orthogonal polynomials and corresponding quadratures of Gaussian type. Math. Balkanica **26**, 169–184 (2012)
12. Nevai, P.G.: Orthogonal polynomials. Mem. Amer. Math. Soc. **18**(213), v + 185 pp (1979)
13. Olver, F.W.J., Lozier, D.W., Boisvert, R.F., Clark, C.W. (eds.): NIST Handbook of Mathematical Functions. U. S. Department of Commerce and Cambridge University Press, Cambridge (2010)
14. Rakhmanov, E.A.: The asymptotic behavior of the ratio of orthogonal polynomials. II. (Russian). Mat. Sb. (N. S.) **118**(160), 104–117 (1982). [English translation in Math. USSR-Sbornik 46 (1983), 105–117.]
15. Szegö, G. Orthogonal Polynomials, Colloquium Publications, 4th edn., vol. 23. American Mathematical Society, Providence (1975)

44.5. [224] (with Gradimir V. Milovanović) “Orthogonal polynomials relative to weight functions of Prudnikov type”

[224] (with Gradimir V. Milovanović) “Orthogonal polynomials relative to weight functions of Prudnikov type”, *Numer. Algorithms*, to appear.

Orthogonal polynomials relative to weight functions of Prudnikov type

Walter Gautschi[*] Gradimir V. Milovanović[†]

Abstract

Moment-based methods are used to generate the three-term recurrence relation for polynomials orthogonal with respect to the Prudnikov, the generalized Prudnikov, and Prudnikov-type weight functions and their symmetric extensions. All procedures developed are implemented, and made available, in Matlab software.

Keywords: Orthogonal polynomials, Prudnikov-type weight functions, three-term recurrence relation

MSC 2020 – Mathematics Subject Classification System: 15B52, 3304, 33C47

1 Introduction

Let

$$\rho_\nu(x) = 2\, x^{\nu/2} K_\nu(2\sqrt{x}), \quad x > 0, \quad \nu \in \mathbb{R},$$

where K_ν is the second-kind modified Bessel function of order ν ([6, §10.25]). *Prudnikov polynomials* [10, §3] are polynomials orthogonal with respect to the weight function

$$w_\nu(x) = \rho_\nu(x), \quad 0 < x < \infty, \quad \nu \geq 0. \tag{1.1}$$

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

[†] Serbian Academy of Sciences and Arts, 11000 Belgrade, Kneza Mihaila 35, Serbia and University of Niš, Faculty of Sciences and Mathematics, 18000 Niš, Serbia (gvm@mi.sanu.ac.rs)

A. P. Prudnikov (1927–1999) advocated their study in [7, Problem 2] and dealt with the case $\nu = 0$ in [*ibid.*, Problem 1]. S. Yakubovich, in [10], considered also *generalized Prudnikov polynomials* orthogonal with respect to the weight function

$$w_\nu^\alpha(x) = x^\alpha \rho_\nu(x), \quad 0 < x < \infty, \quad \alpha > -1, \tag{1.2}$$

as well as *Prudnikov-type polynomials*, in [9, §2] of type 1, orthogonal with respect to

$$w_\nu^+(x) = \mathrm{e}^{-x} \rho_\nu(x), \quad 0 < x < \infty, \quad \nu > -1, \tag{1.3}$$

and in [9, §3] of type 2, orthogonal with respect to

$$w_\nu^-(x) = x^{-1}\mathrm{e}^{-1/x} \rho_\nu(x), \quad 0 < x < \infty, \quad \nu \in \mathbb{R}. \tag{1.4}$$

Symmetric extensions of all these polynomials will also be considered, where x in the weight function is replaced by $|x|$, and the support interval is extended to the whole real line.

Multiple orthogonal polynomials relative to the pair $x^\alpha\rho_\nu(x)$, $x^\alpha\rho_{\nu+1}(x)$ of weight functions are studied in [8]; see also [1] and [11, §IIA].

Weight functions involving modified Bessel functions $K_\nu(x)$ (rather than $K_\nu(2\sqrt{x})$) have been used previously in connection with wave functions for nonlocal potentials [5]; see also [2, Exercise 2.32], [3, §2.1.3], [4, §2.1.3].

The object of this note is to develop the respective orthogonal polynomials and their symmetric extensions, in particular, to obtain the three-term recurrence relations they satisfy and to provide related Matlab software. The approach used in all cases is the classical Chebyshev algorithm, computing the recurrence coefficients from the moments of the weight function. Because of the underlying ill-conditioning, high-precision computation is required.

2 Moments

The nth-order moment of the generalized Prudnikov weight function is

$$\mu_n^{\nu,\alpha} = \int_0^\infty x^n w_\nu^\alpha(x)\mathrm{d}x = \Gamma(n+\alpha+\nu+1)\Gamma(n+\alpha+1), \ \ \nu \ge 0, \ \ \alpha > -1, \tag{2.1}$$

as follows from [10, Eq. (2.4)] where μ is replaced by $n+\alpha$. For the first Prudnikov-type weight function, the moment of order n is

$$\mu_n^+ = \int_0^\infty x^n w_\nu^+(x)\,\mathrm{d}x = \int_0^\infty x^n \mathrm{e}^{-x}\rho_\nu(x)\,\mathrm{d}x = 2\int_0^\infty x^{n+\nu/2}\mathrm{e}^{-x}K_\nu(2\sqrt{x})\,\mathrm{d}x, \qquad \nu > -1.$$

Using Mathematica 12.2, one finds (cf. also [9, Eq. (1.8)])

$$\mu_n^+ = n!\Gamma(n+\nu+1)U(n+\nu+1,\nu+1,1),$$

where $U(a,b,x)$ is the confluent hypergeometric function (also known as Tricomi's function) or, in terms of generalized hypergeometric functions, $U(a,b,x) = x^{-a}\,{}_2F_0(a,a-b+1;-;-x^{-1})$ [6, §13.6(vi)]. Thus,

$$\mu_n^+ = n!\Gamma(n+\nu+1)\,{}_2F_0(n+\nu+1,n+1;-;-1). \tag{2.2}$$

For the second Prudnikov-type weight function, the moments have been given in [9, Eq. (1.9)], though involving (in the last line of the equation) the gamma function at a nonpositive integer, that is, ∞. We have, however,

$$\mu_n^- = \int_0^\infty x^n w_\nu^-(x)\mathrm{d}x = 2\int_0^\infty x^{n+\nu/2-1}\mathrm{e}^{-1/x}K_\nu(2\sqrt{x})\mathrm{d}x, \quad \nu\in\mathbb{R}. \tag{2.3}$$

Here, the second integral can be expressed in terms of the Meijer G-function, for which we use here the Matlab notation on the right of

$$G_{p,q}^{m,n}\left(z;\ \begin{matrix} a_1,a_2,\dots,a_p \\ b_1,b_2,\dots,b_q \end{matrix}\right) = \texttt{meijerG}\,(\,[a_1,\dots,a_n],[a_{n+1},\dots,a_p],[b_1,\dots b_m],[b_{m+1},\dots,b_q],z) \qquad 0\le n\le p,\ 0\le m\le q.$$

(The content between brackets may be empty. For example, if $p=0$, the first two arguments of `meijerG` are empty, or only the second one if $p=n$.)

To begin with, the term on the far right of (2.3) can be written as a Mellin transform of the function $f(x) = 2\,\mathrm{e}^{-1/x}K_\nu(2\sqrt{x})$,

$$\mu_n^- = (\mathcal{M}\,f)(s) = \int_0^\infty x^{s-1}\,f(x)\mathrm{d}x, \quad s = n+\nu/2.$$

Using the Mathematica 12.2 command

```
MellinTransform[2 BesselK[v,2Sqrt[x]]Exp[-1/x],x,s]
```

yields
(2.4)

$$(\mathcal{M}\, f)(s) = \frac{2^{2s-2}}{\pi^{3/2}}\, \texttt{meijerG}\ (\ [1/2, 1, (2-2s-\nu)/4, (4-2s-\nu)/4, \\ (2-2s+\nu)/4, (4-2s+\nu)/4], [\ \ ], [\ \ ], [\ \ ], 64).$$

Letting

$$a_1 = 1, \quad a_2 = 1 - s - \nu/2, \quad a_3 = 1 - s + \nu/2 \quad \text{and} \quad p = n = 3,\ q = m = 0,$$

the right-hand side of (2.4) is

$$\frac{2^{p+1-n-a_1-a_2-a_3}}{\pi^{n-p/2}}\, \texttt{meijerG}\ (\ [\ a_1/2, (a_1+1)/2, a_2/2, (a_2+1)/2, a_3/2, (a_3+1)/2], \\ [\ \ ], [\ \ ], [\ \ ], 4^p),$$

which, by [6, Eq. 16.19.4] and $s = n + \nu/2$, equals

$$\texttt{meijerG}\ (\ [a_1, a_2, a_3], [\ ], [\ ], [\ ], 1) = \texttt{meijerG}\ (\ [1, 1-(n+\nu), 1-n], [\ ], [\ ], [\ ], 1).$$

Thus, simplifying by employing [6, Eq. 16.19.1], we get

$$\mu_n^- = \texttt{meijerG}\ (\ [\ \ ], [\ \ ], [0, n+\nu, n], [\ \ ], 1). \tag{2.5}$$

The moments of the symmetric extension of all the weight functions above are twice the moments stated, if n is even, and zero if n is odd.

3 Orthogonal polynomials and recurrence coefficients

It is well known that (monic) orthogonal polynomials π_k relative to a positive weight function $w(x)$ on some finite or infinite interval $[a, b]$ satisfy a three-term recurrence relation

$$\begin{gathered} \pi_{k+1}(x) = (x - \alpha_k)\pi_k(x) - \beta_k \pi_{k-1}(x), \quad k = 0, 1, 2, \ldots, \\ \pi_{-1} = 0, \quad \pi_0 = 1, \end{gathered} \tag{3.1}$$

where the coefficients α_k are real and $\beta_k > 0$. Conventionally, β_0 is taken to be $\beta_0 = \int_a^b w(x)\mathrm{d}x$. For any computational work with orthogonal polynomials, knowledge of this recurrence relation, that is, of its recurrence coefficients, is indispensable.

There is Matlab software available that generates to any given accuracy the first N recurrence coefficients α_k, β_k, $k = 0, 1, 2, \ldots, N-1$, and places them into the first, respectively second, column of an $N \times 2$ array `ab`; see, e.g., [2, §1]. The procedure used here is the Chebyshev algorithm, generating the first N recurrence coefficients from the first $2N$ moments of the weight function. Given the ill-conditioned nature of this proposition, it is important to know how many working digits are required to obtain all N coefficients β_k (and thus, presumably, also all $\alpha_k \neq 0$) to a given relative accuracy. For the three weight functions of Prudnikov type, this is answered by the Matlab routines `dig_gprudnikov.m`, `dig_prudnikov_type.m`, and for the respective symmetric extensions by `dig_gprudnikov_symm.m`, `dig_prudnikov_type_symm.m`. These routines not only provide the desired number `dig` of working digits, but also the respective array `ab` of recurrence coefficients to the accuracy requested. Once this number `dig` of required working digits is known, the array `ab` can be generated directly by the routines `sr_gprudnikov.m`, `sr_prudnikov_type.m`, `sr_gprudnikov_symm.m`, `sr_prudnikov_type_symm.m`.

For all Matlab routines needed, visit

`https://www.cs.purdue.edu/archives/2002/wxg/codes/PRUD.html`.

3.1 Generalized Prudnikov polynomials

Our target precision for the recurrence coefficients, in this and the next two subsections, is 15-digit accuracy. For generalized Prudnikov polynomials, the results of the routine `dig gprudnikov.m` are shown in Table 1 for selected values of ν and for $\alpha = \pm 1/2$. (Other values of $\alpha > -1$, including $\alpha = 0$, in the range from $-.9$ to 10.6, have led to basically the same results, except, occasionally, somewhat larger ones, but never by more than 3 units.) It can be seen that, for each N shown, the results are more or less the same, which means that the underlying conditioning is essentially independent of the parameters ν and α. The results, in fact, suggest condition numbers of the order $10^{10} - 10^{12}$ when $N = 25$, $10^{28} - 10^{30}$ when $N = 50$, and $10^{63} - 10^{67}$ when $N = 100$.

N	ν	α	`dig`	N	ν	α	`dig`	N	ν	α	`dig`
25	0	$-1/2$	25	50	0	$-1/2$	43	100	0	$-1/2$	78
		$1/2$	25			$1/2$	44			$1/2$	79
	$1/3$	$-1/2$	27		$1/3$	$-1/2$	45		$1/3$	$-1/2$	81
		$1/2$	27			$1/2$	45			$1/2$	82
	$2/3$	$-1/2$	27		$2/3$	$-1/2$	45		$2/3$	$-1/2$	81
		$1/2$	27			$1/2$	45			$1/2$	82
	1	$-1/2$	25		1	$-1/2$	43		1	$-1/2$	80
		$1/2$	25			$1/2$	43			$1/2$	80
	3	$-1/2$	25		3	$-1/2$	44		3	$-1/2$	80
		$1/2$	26			$1/2$	43			$1/2$	81
	6	$-1/2$	26		6	$-1/2$	44		6	$-1/2$	81
		$1/2$	26			$1/2$	43			$1/2$	81

Table 1: The number `dig` of digits required in the Chebyshev algorithm to obtain the first N recurrence coefficients of the generalized Prudnikov polynomials to an accuracy of 15 decimal digits

We used our routine `sr_gprudnikov(dig,nofdig,N,nu,alpha)` with `dig` =30, `nofdig`=18, `N`=11, `nu`=`alpha`=0, to check Table 9.1 in [7] containing the values of $a_n = \sqrt{\beta_n}$ for $n = 1, 2, \ldots, 10$. Agreement to all digits was observed except for the last digit, which occasionally is off by one unit.

3.2 Prudnikov-type polynomials of the first type

Here the results of the routine `dig_prudnikov_type.m` with `type`=1 (for the weight function w_ν^+) are shown in Table 2 for selected values of $\nu > -1$. As before in the case of generalized Prudnikov polynomials, the results are practically independent of ν.

The condition numbers for $N = 25, 50, 100$ are now about 10^{20}, 10^{40}, and 10^{85}, that is, substantially larger than in the case of generalized Prudnikov weight functions.

N	ν	dig	N	ν	dig	N	ν	dig
25	$-2/3$	34	50	$-2/3$	55	100	$-2/3$	102
	$-1/3$	35		$-1/3$	55		$-1/3$	102
	0	35		0	54		0	100
	$1/3$	35		$1/3$	55		$1/3$	102
	$2/3$	35		$2/3$	55		$2/3$	102
	1	35		1	54		1	100
	3	35		3	54		3	101
	6	35		6	54		6	101

Table 2: The number dig of digits required in the Chebyshev algorithm to obtain the first N recurrence coefficients of the Prudnikov-type polynomials of type 1 to an accuracy of 15 decimal digits

N	ν	dig	N	ν	dig	N	ν	dig
25	-6	23	50	-6	42	100	-6	79
	-3	25		-3	42		-3	79
	-1	25		-1	43		-1	79
	$-2/3$	26		$-2/3$	44		$-2/3$	81
	$-1/3$	26		$-1/3$	44		$-1/3$	81
	0	24		0	42		0	79
	$1/3$	26		$1/3$	44		$1/3$	81
	$2/3$	26		$2/3$	44		$2/3$	81
	1	24		1	43		1	79
	3	26		3	43		3	80
	6	25		6	44		6	80

Table 3: The number dig of digits required in the Chebyshev algorithm to obtain the first N recurrence coefficients of the Prudnikov polynomials of type 2 to an accuracy of 15 decimal digits

3.3 Prudnikov-type polynomials of the second type

Here the results of the routine dig_prudnikov_type.m with type=2 (for the weight function w_ν^-) are shown in Table 3 for selected values of $\nu \in \mathbb{R}$. They are quite similar to the ones in §3.1 where applicable, and so is the degree of ill-contitioning.

Knowing the degree of ill-conditioning, it is easy to estimate the number of digits needed to get any desired accuracy. Thus, for example, when $N = 100$, to get 32-digit accuracy will require something like 32+67=99 digits in the case of generalized Prudnikov weight functions and Prudnikov weight functions of type 2, and 32+87=119 digits for Prudnikov weight functions of type 1. Both these numbers have been corroborated numerically.

3.4 Polynomials orthogonal relative to the symmetric extension of weight functions of Prudnikov type

Symmetry usually lowers condition numbers. This is the case here, where, compared with the case of generalized Prudnikov polynomials in subsection 3.1, the number `dig` of required digits is now about one half of those in Table 1 when N is 25 and 50, and even somewhat smaller when $N = 100$. More specifically, `dig` is never greater than 16, 22, 35 for, respectively, $N = 25, 50, 100$ and parameters ν and α as in Table 1. Similarly, for symmetric Prudnikov-type polynomials of type 1, the largest numbers `dig` are 20, 27, 45 for, respectively $N = 25, 50, 100$ and for ν as in Table 2, and for symmetric Prudnikov-type polynomials of type 2 they are 15, 21, 34 for ν as in Table 3.

Acknowledgments. The work of the second author was supported in part by the Serbian Academy of Sciences and Arts (Φ-96). The authors thank Walter Van Assche for drawing their attention to related work on multiple orthogonal polynomials in [8], [1], and [11].

References

[1] Coussement, Els and Walter Van Assche. Some properties of multiple orthogonal polynomials associated with Macdonald functons. Proceedings of the Fifth International Symposium on Orthogonal Polynomials, Special Functions and their Applications (Patras, 1999). *J. Comp. Appl. Math.* 133 (2001), 253–261.

[2] Gautschi, Walter. *Orthogonal polynomials in MATLAB. Exercises and solutions*, Software, Environments, and Tools, 26. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2016.

[3] Gautschi, Walter. *A software repository for orthogonal polynomials*, Software, Environments, and Tools, 28. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2018.

[4] Gautschi, Walter. *A software repository for Gaussian quadratures and Christoffel functions*, Software, Environments, and Tools, 32. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2021.

[5] Gordon, R. G. Constructing wavefunctions for nonlocal potentials, *J. Chem. Phys.* 52 (1970), 6211–6217.

[6] Olver, Frank W. J. et al., eds. *NIST Handbook of Mathematical Functions*, National Institute of Standards and Technology, and Cambridge University Press, 2010.

[7] Van Assche, Walter. Open problems. Proceedings of the Seventh Spanish Symposium on Orthogonal Polynomials and Applications (VII SPOA) (Granada, 1991). *J. Comput. Appl. Math.* 48 (1993), 225–243.

[8] Van Assche, W. and S. B. Yakubovich. Multiple orthogonal polynomials associated with Macdonald functions, *Integral Transforms Spec. Funct.* 9 (2000), 229–244.

[9] Yakubovich, S. Orthogonal polynmials with the Prudnikov-type weights, *Complex Anal. Oper. Theory* 14 (2020), no. 1, Art. 26, 27 pp.

[10] Yakubovich, S. Orthogonal polynomials with ultra-exponential weight functions: An explicit solution to the Ditkin–Prudnikov problem, *Constr. Approx.* 53 (2021), 1–38.

[11] Zhang, Lun. A note on the limiting mean distribution of singular values for products of two Wishart random matrices, *J. Math. Phys.* 54 (2013), no. 8, 083303, 8 pp.

44.6. [225] “Another look at polynomials orthogonal relative to exponential integral weight functions”

[225] “Another look at polynomials orthogonal relative to exponential integral weight functions”, *Numer. Algorithms*, to appear.

Another look at polynomials orthogonal relative to exponential integral weight functions

Walter Gautschi*

Abstract

The computation of the three-term recurrence relation for the orthogonal polynomials in the title from modified moments is revisited and related Matlab software provided.

Keywords Orthogonal polynomials · Exponential integral weight function · Recurrence coefficients · Matlab software
Mathematics Subject Classification (2010) 33C47 · 65D30

1 Introduction

Orthogonal polynomials relative to the exponential integral weight function, also called generalized exponential integral [3, §8.19],

$$w(x) = E_\nu(x), \quad E_\nu(x) = \int_1^\infty \frac{\mathrm{e}^{-xt}}{t^\nu}\mathrm{d}t, \quad \nu > 0,$$

have recently been considered in [1] and [2, Exercise 2.26], for the infinite support interval $[0, \infty]$ as well as for a finite one, $[0, c]$, $c > 0$. In the former case, to compute recurrence coefficients, we applied both, the Chebyshev

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

algorithm based on ordinary moments, and a modified Chebyshev algorithm based on appropriate modified moments. It was stated that, surprisingly, the former performed better than the latter. In retrospect, we realize that this statement is misleading in so far as the analysis dealt not only with the performance of the Chebyshev algorithm but included also the performance of generating the modified moments, which happened to be a process subject to massive cancellation errors. To make a fair comparison of the Chebyshev algorithm with the modified Chebyshev algorithm, one must assume that the moments, resp. modified moments, which are input variables to these aLgorithms, are given accurately, and that their generation should not be part of the analysis. If done so, the modified Chebyshev algorithm will remain the superior one, even, to a lesser extent, in the case of infinite intervals, as will be documented in this article. At the same time we will develop a more satisfactory way of generating modified moments.

2 Modified moments

For ordinary moments, see [2, Exercise 2.26 (a) and (c)].

2.1 Analysis

We need the first $2n$ modified moments m_k in order to obtain the $n \times 2$ array `ab` of recurrence coefficients for the weight function w by means of the modified Chebyshev algorithm.

2.1.1 Infinite support interval $[0, \infty)$

The modified moments, relative to any monic polynomials p_k of degree k, are given by

$$m_k = \int_0^\infty p_k(t)E_\nu(t), \quad E_\nu(t) = \int_1^\infty \frac{\mathrm{e}^{-tu}}{u^\nu}\,\mathrm{d}u, \quad k = 0, 1, 2, \ldots, 2n-1,$$

where $\nu > 0$. Thus,

$$m_k = \int_0^\infty p_k(t)\,\mathrm{d}t \int_1^\infty \frac{\mathrm{e}^{-tu}}{u^\nu}\,\mathrm{d}u.$$

Changing variables, $tu = s$, in the inner integral yields

$$\begin{aligned} m_k = & \int_0^\infty p_k(t)\,\mathrm{d}t \int_t^\infty \left(\frac{s}{t}\right)^{-\nu} \mathrm{e}^{-s}\,\frac{\mathrm{d}s}{t} \\ = & \int_0^\infty p_k(t)\,t^{\nu-1}\mathrm{d}t \int_t^\infty s^{-\nu}\mathrm{e}^{-s}\mathrm{d}s, \end{aligned}$$

that is,

$$m_k = \int_0^\infty p_k(t)\,t^{\nu-1}\,\Gamma(1-\nu,t)\,\mathrm{d}t, \tag{2.1}$$

where $\Gamma(a,x)$ is the upper incomplete gamma function [3, Eq. 8.2.2]. Since $k \le 2n-1$ and the p_k are polynomials of degree at most $2n-1$, the integral in (2.1) can be evaluated exactly by the n-point Gauss quadrature rule relative to the weight function

$$v(x) = x^{\nu-1}\,\Gamma(1-\nu,x), \quad \nu > 0, \tag{2.2}$$

supported on the interval $[0,\infty)$. Thus, if the nodes and weights of this quadrature rule are $\tau_i^{(n)}$ and $\lambda_i^{(n)}$, both depending on ν, we have

$$m_k = \sum_{i=1}^n \lambda_i^{(n)} p_k(\tau_i^{(n)}), \quad k = 0,1,2,\ldots,2n-1. \tag{2.3}$$

For the polynomials p_k we will take monic generalized Laguerre polynomials with parameter $a = -1/4$.

2.1.2 Finite support interval $[0,c]$, $c > 0$

The modified moments, relative to any monic polynomials p_k of degree k, are now given by

$$m_k = \int_0^c p_k(t) E_\nu(t)\mathrm{d}t, \quad E_\nu(t) = \int_1^\infty \frac{\mathrm{e}^{-tu}}{u^\nu}\,\mathrm{d}u, \quad k = 0,1,2,\ldots 2n-1.$$

In the same way as in §2.1.1 one finds

$$m_k = \int_0^c p_k(t)\,t^{\nu-1}\,\Gamma(1-\nu,t)\,\mathrm{d}t, \tag{2.4}$$

that is, the same relation as in (2.1) but with the weight function (2.2) supported on $[0, c]$. Therefore, (2.3) holds again, with the $\tau_i^{(n)}$ and $\lambda_i^{(n)}$, both depending on ν and c, the nodes and weights of the n-point Gauss quadrature rule for the weight function (2.2) supported on $[0, c]$.

For the polynomials p_k we will take the monic Legendre polynomials on the interval $[0, c]$.

It remains to show how the arrays `ab` of recurrence coefficients for the weight functions (2.2), supported either on $[0, \infty)$ or on $[0, c]$, and hence the required Gaussian quadrature rules, can be obtained. We propose to use, in either case, the Chebyshev algorithm in sufficiently high precision. For this, we need the moments of the weight function (2.2),

$$\mu_k(v) = \int_0^c t^{k+\nu-1}\,\Gamma(1-\nu, t)\,\mathrm{d}t, \quad k = 0, 1, 2, \ldots, 2n-1.$$

These can be obtained explicitly, using integration by parts. One finds, for $k = 0, 1, 2, \ldots, 2n-1$,

(2.5)
$$\mu_k(v) = \frac{1}{k+\nu}\begin{cases} \Gamma(k+1) \ \text{ if } c = \infty, \\ \\ \big[\Gamma(1-\nu, c)\,c^{k+\nu} + \Gamma(k+1) - \Gamma(k+1, c)\big] \ \text{ if } 0 < c < \infty. \end{cases}$$

The Chebyshev algorithm has been applied, with working precision of as much as 500 digits, to compute 32-digit $N \times 2$ arrays `ab` with $N = 100$ for $\nu = 1/2, 1, 2$ and $c = 1, 3, 5, \infty$; see §2.2.

2.2 Matlab implementation and numerical results

All our numerical results are consistently assuming a target precision of 15 digits, that is, about Matlab double precision.

2.2.1 Infinite support interval $[0, \infty)$

We begin with presenting the Matlab software used for computing the $n \times 2$ array `ab` of the desired recurrence coefficients from either the moments of the weight function w or from the modified moments. The Matlab function generating the first $2n$ moments is `smom_Enu.m`[1] while the first $2n$ modified moments are generated by the function `smmom_Enu.m`. Both are used, respectively, in the function `dig_Enu.m` and `dig_Enumod.m` to produce the $n \times 2$ array `ab` of recurrence coefficients of the desired orthogonal polynomials to a prescribed accuracy along with the number `dig` of digits required to achieve this accuracy. If this number is known, the array `ab` can be obtained more directly by the function `sr_Enu.m` resp. `sr_Enumod.m`.

For the values 1/2, 1, 2 of the parameter ν, the auxiliary $N \times 2$ arrays `ab` of the recurrence coefficients for the weight function v in (2.2) have been computed for $N = 100$ to an accuracy of 32 digits and are made available in the text files `ab_powigaminf_1half.txt`, `ab_powigaminf_1.txt`, and `ab_powigaminf_2.txt`. For any other values of the parameter ν, they can easily be produced by the function `dig_powigaminf.m`.

Computation of the moments by the routine `smom_Enu.m` is quite stable since they are simple rational functions; see [2, Eq. (2.56)]. It is found, likewise, using `dig_mmEnu.m`, that the computation of the modified moments by the routine `smmom_Enu.m` is also fairly stable, involving a loss of no more than 1 or 2 digits. With regard to the number of digits needed to obtain the recurrence coefficients to an accuracy of 15 digits, our routines `dig_Enu.m` and `dig_Enumod.m` found them to be as shown in Table 1, where the numbers in parentheses refer to moments, and those preceding them to modified moments. It can be seen, in accord with previous experience with unbounded

n	$\nu = 1$	$\nu = 1/2$	$\nu = 2$
20	23 (25)	25 (24)	24 (25)
40	43 (44)	44 (44)	43 (44)
60	62 (64)	63 (64)	62 (64)
80	81 (83)	82 (81)	80 (82)
100	100 (101)	102 (101)	100 (101)

Table 1: Number of digits required to obtain the recurrence coefficients accurate to 15 digits when the interval is $[0, \infty]$

[1] All Matlab functions and text files relevant to implement the work in this paper, are accessible on the website `https://www.cs.purdue.edu/archives/2002/wxg/codes/ALAOPEXPINT.html`.

intervals, that the modified Chebyshev algorithm is only marginally better than the ordinary Chebyshev algorithm, and sometimes even a bit worse, and both are facing a considerable amount of ill-conditioning. Variable-precision arithmetic is therefore a necessity here, and ordinary Matlab double-precison arithmetic not an option.

2.2.2 Finite support interval $[0, c]$, $c > 0$

The moments of the weight function w can be expressed explicitly in terms of incomplete gamma functions, see [2, Eq. (2.61)], while the modified moments require Gaussian quadrature in (2.4) and therefore are more time-consuming to compute. The Matlab functions generating the first $2n$ of them are now `smom_Enufin.m` resp. `smmom_Enufin.m`. As before, they are used in the functions `dig_Enufin.m` resp. `dig_Enufinmod.m` to produce the $n \times 2$ array `ab` of recurrence coefficients for the desired orthogonal polynomials to a prescribed accuracy along with the number `dig` of digits required to achieve this accuracy. If this number is known, one can use directly the functions `sr_Enufin.m` and `sr_Enufinmod.m`. In the latter, the modified moments are computed in `(dig+2)`-digit arithmetic rather than `dig`-digit arithmetic as in the rest of the routine. The computation of the moments and modified moments, as in §2.2.1, is found, using `dig_mEnufin.m` and `dig_mmEnufin.m`, to be again quite stable. Computing the recurrence coefficients from the modified moments, however, is now fundamentally different. This is shown in Table 2, generated for $\nu = 1$ and $c = 1, 3, 5$ by the Matlab functions `dig_Enufin.m` and `dig_Enufinmod`, both in variable-precision arithmetic. (For $\nu = 1/2$ and $\nu = 2$, the results are practically identical, being different only occasionally

n	$c = 1$	$c = 3$	$c = 5$
20	16 (40)	16 (40)	16 (38)
40	16 (69)	16 (69)	16 (69)
60	16 (99)	16 (99)	16 (99)
80	16 (129)	16 (130)	16 (130)
100	16 (160)	16 (160)	16 (160)

Table 2: Number of digits required to obtain the recurrence coefficients accurate to 15 digits when the interval is $[0, c]$ and $\nu = 1$

by one unit.) As is clearly visible, the routine `dig_Enufinmod`, using modified moments, is perfectly stable, and in fact, when run entirely in Matlab double precision, produces almost fully accurate results. Thus, if an accuracy of 13 decimal digits is acceptable, no variable-precision routine is needed in this case, and the Matlab double-precision function `ab=r_Enufinmod.m` is adequate and substantially faster.

If we compare the runtime of the two routines `sr_Enufin.m` and `sr_Enufinmod.m`, both run in variable-precision arithmetic with the appropriate number of digits, we find that the former is faster by about 30% than the latter, evidently because of the more complicated nature of modified moments. So one may prefer to use the former routine rather than the latter.

3. The author declares that he has no conflict of interest.

References

[1] Gautschi, Walter. Polynomials orthogonal with respect to exponential integrals, *Numer. Algorithms* 70 (2015), 215–226.

[2] Gautschi, Walter. *Orthogonal polynomials in MATLAB: exercises and solutions*, Software, Environments, and Tools, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[3] Olver, Frank W. J. et al., eds. *NIST handbook of mathematical functions*, National Institute of Standards and Technology (NIST) and Cambridge University Press, 2010.

45

Paper on Gauss-type Quadrature

209 High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions, Numer. Algorithms 67, 59–72 (2014)

45.1. [209] “High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions”

[209] “High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions”, *Numer. Algorithms* **67**, 59–72 (2014).

Numer Algor (2014) 67:59–72
DOI 10.1007/s11075-013-9774-z

ORIGINAL PAPER

High-precision Gauss–Turán quadrature rules for Laguerre and Hermite weight functions

Walter Gautschi

Received: 12 July 2013 / Accepted: 20 September 2013 / Published online: 2 October 2013

Abstract Procedures and corresponding Matlab software are presented for generating Gauss–Turán quadrature rules for the Laguerre and Hermite weight functions to arbitrarily high accuracy. The focus is on the solution of certain systems of nonlinear equations for implicitly defined recurrence coefficients. This is accomplished by the Newton–Kantorovich method, using initial approximations that are sufficiently accurate to be capable of producing n-point quadrature formulae for n as large as 42 in the case of the Laguerre weight function, and 90 in the case of the Hermite weight function.

Keywords Gauss–Turán quadrature rules · Computation of nodes and weights

Mathematics Subject Classification (2010) 65D32

1 Introduction

We recall that a Gauss–Turán quadrature rule for any positive weight function (or measure $\mathrm{d}\lambda$) is a formula of the type

$$\int_{\mathbb{R}} f(t)\mathrm{d}\lambda(t) = \sum_{\nu=1}^{n}\sum_{\sigma=0}^{2s} \lambda_\nu^{(\sigma)} f^{(\sigma)}(\tau_\nu) + R_{n,s}(f), \tag{1.1}$$

where n and s are positive integers $n \geq 1$, $s > 0$, and the formula has maximum degree of exactness $d = (2s+2)n - 1$,

$$R_{n,s}(f) = 0 \quad \text{for all } f \in \mathbb{P}_{(2s+2)n-1}, \tag{1.2}$$

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

where $\mathbb{P}_k$ denotes the set of polynomials of degree $\leq k$ (cf., e.g., [1, Section 3.1.3.1]). If $s = 0$, this is the usual Gaussian quadrature rule. We are interested here in computing the nodes τ_ν and weights $\lambda_\nu^{(\sigma)}$ for values of n as large as possible, and for $1 \leq s \leq 5$, which is probably sufficient for most practical applications.

The task is usually divided into two parts: the calculation of the nodes τ_ν, and the calculation of the weights $\lambda_\nu^{(\sigma)}$. The former necessarily requires the solution of nonlinear systems of equations. Early work on this subject ([6]) is focused directly on the n nodes τ_ν and a system of n nonlinear equations satisfied by them. The same is true for more recent work such as [3–5] (dealing with more general Gauss–Turán type formulae with multiplicities varying from node to node). Here we use a different system of equations, originally proposed by Milovanović in [2] (see also [1, Section 3.1.3.2]), namely a system of $2n-1$ nonlinear equations satisfied by certain recurrence coefficients of implicitly defined orthogonal polynomials. With regard to the weights $\lambda_\nu^{(\sigma)}$, they are computed for each fixed ν from an upper triangular system of $2s+1$ linear equations.

The main problem to be considered in this work is the solution of the system of $2n-1$ nonlinear equations for the $2n-1$ recurrence coefficients in question. We propose to use the Newton–Kantorovich method for this purpose. For this, as is well known, it is critical to have at hand sufficiently accurate initial approximations. In the case of Laguerre and Hermite weight functions, considered here, this happens to be relatively simple, owing to a startling property satisfied by the desired recurrence coefficients.

It is well known that the three-term recurrence relation for the monic Laguerre polynomials,

$$L_{k+1}(x) = \left(x - \alpha_k^L\right) L_k(x) - \beta_k^L L_{k-1}(x), \quad k = 0, 1, 2, \dots,$$

has coefficients $\alpha_k^L = 2k+1$ and $\beta_k^L = k^2$, that is, the α-coefficients are linear in k and the β-coefficients quadratic. Surprisingly, a similar property holds, at least approximately, for the recurrence coefficients of interest here. Likewise, the recurrence coefficients for the Hermite polynomials are $\alpha_k^H = 0$, $\beta_k^H = \frac{1}{2}k$, that is, the β-coefficients are linear in k, while the α-coefficients vanish by symmetry. Something similar happens to be true for the coefficients of interest here, at least approximately in the case of the β-coefficients. These remarkable properties allow us to obtain very accurate initial approximations for Newton's method, giving rise to convergence (to machine precision or better) in as few as 4–8 iterations and for n as large as 42 in the Laguerre case (when $s = 1$), i.e., for systems of as many as 83 nonlinear equations, and even for larger n in the Hermite case. Considering that only ten years ago the author had convergence problems already for $n = 5$ resp. $n = 7$ (cf. [1, Example 3.24]), this constitutes real progress!

There is, however, another difficulty to cope with: the increasingly (with n and s) ill-conditioned and normwise large Jacobian matrices in Newton's method. High-precision computation, therefore, is essential in this context.

2 Numerical approach

For fixed n and s, the nodes $\tau_\nu = \tau_{\nu,s}^{(n)}$ of (1.1) are the zeros of the (monic) polynomial $\pi_{n,s}$ of degree n, which is the last member of the sequence of polynomials $\{\pi_{k,s}\}$ (all depending on n) defined by

$$\begin{aligned} \pi_{k+1,s}(t) &= (t-\alpha_{k,s})\pi_{k,s}(t) - \beta_{k,s}\pi_{k-1,s}(t), \quad k=0,1,2,\ldots,n-1, \\ \pi_{0,s}(t) &= 1, \quad \pi_{-1,s}=0, \end{aligned} \tag{2.1}$$

where $\alpha_{k,s} = \alpha_{k,s}^{(n)}$, $\beta_{k,s} = \beta_{k,s}^{(n)}$ are suitable real resp. positive coefficients. (The superscripts are to emphasize that all these quantities depend on n. Also, we set here $\beta_{0,s}^{(n)} = 0$.). This looks like the three-term recurrence relation for orthogonal polynomials. That's in fact what it is, but the orthogonal polynomials $\pi_{0,s}, \pi_{1,s}, \ldots, \pi_{n,s}$ are defined implicitly, their measure being $\mathrm{d}\lambda_{n,s}(t) = [\pi_{n,s}(t)]^{2s}\mathrm{d}\lambda(t)$ (cf. [1, Section 3.1.3.2]). The $2n-1$ coefficients $\alpha_k = \alpha_{k,s}^{(n)}$, $\beta_k = \beta_{k,s}^{(n)}$ satisfy a system of $2n-1$ nonlinear equations,

$$\boldsymbol{f}(\boldsymbol{\rho}_n) = \boldsymbol{0}, \quad \boldsymbol{\rho}_n^{\mathrm{T}} = [\alpha_0, \alpha_1, \ldots, \alpha_{n-1}; \beta_1, \ldots, \beta_{n-1}], \tag{2.2}$$

where each component f_ν of $\boldsymbol{f} = [f_1, f_2, \ldots, f_{2n-1}]^{\mathrm{T}}$ is a highly nonlinear function of the $2n-1$ variables $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}; \beta_1, \ldots, \beta_{n-1}$, namely (cf. [1, Eq. (3.1.66)])

$$\begin{aligned} f_{2\nu+1} &= \int_{\mathbb{R}} (\alpha_\nu - t)\pi_{\nu,s}^2(t)\pi_{n,s}^{2s}(t)\mathrm{d}\lambda(t), \quad \nu = 0,1,\ldots,n-1, \\ f_{2\nu} &= \int_{\mathbb{R}} \left(\beta_\nu \pi_{\nu-1,s}^2(t) - \pi_{\nu,s}^2(t)\right)\pi_{n,s}^{2s}\mathrm{d}\lambda(t), \quad \nu = 1,\ldots,n-1. \end{aligned} \tag{2.3}$$

Once this system is solved, the nodes τ_ν are computed as the eigenvalues of the Jacobi matrix of order n associated with the three-term recurrence relation (2.1). The weights $\lambda_\nu^{(\sigma)}$, $\sigma = 0,1,\ldots,2s$, for each ν, can be determined as the solution of an upper triangular system of $2s+1$ linear equations,

$$\hat{\boldsymbol{A}}\hat{\boldsymbol{\lambda}} = \hat{\boldsymbol{\mu}}, \quad \hat{\boldsymbol{A}} \in \mathbb{R}^{(2s+1)\times(2s+1)}, \quad \hat{\boldsymbol{\lambda}} = \left[\lambda_\nu^{(\sigma)}\right]_{\sigma=0}^{2s}, \tag{2.4}$$

where $\hat{\boldsymbol{\mu}}$ is a vector, and $\hat{\boldsymbol{A}}$ an upper triangular Toeplitz matrix, both computable from the nodes $\tau_1, \tau_2, \ldots, \tau_n$ and depending on ν (cf. [1, Eq. (3.1.68)]).

In the next two sections, the procedure, as applied to the Laguerre and Hermite weight functions, is described in more detail. It suffices to discuss the procedure for solving the system of nonlinear equations (2.2), which is the core problem. The remaining part of the computation, i.e., the computation of the nodes τ_ν and weights $\lambda_\nu^{(\sigma)}$, is straightforward and does not require further discussion.

3 Gauss–Turán formulae for the Laguerre weight function

In this section, we let throughout $\mathrm{d}\lambda(t) = e^{-t}\mathrm{d}t$ on $[0,\infty]$. In a first subsection, Section 3.1, we look at the case $n=1$, which involves just one nonlinear (in fact, algebraic) equation. In general, the problem consists of solving the system (2.2) of

$2n-1$ nonlinear equations for $n = 1, 2, \ldots, N$, where N is a given (hopefully large) integer. Each of these systems will have a solution which we denote by

$$\alpha_0^{(n)}, \alpha_1^{(n)}, \ldots, \alpha_{n-1}^{(n)}; \ \beta_1^{(n)}, \ldots, \beta_{n-1}^{(n)}, \qquad n = 1, 2, \ldots, N, \tag{3.1}$$

suppressing the dependence on s. Our objective is to compute the matrix R of dimension $N \times (2N-1)$ containing in the nth row the solution (3.1) followed by zeros if $n < N$, that is,

$$R = \begin{bmatrix} \alpha_0^{(1)} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_0^{(2)} & \alpha_1^{(2)} & \beta_1^{(2)} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \alpha_0^{(3)} & \alpha_1^{(3)} & \alpha_2^{(3)} & \beta_1^{(3)} & \beta_2^{(3)} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & 0 & 0 & 0 \\ \alpha_0^{(N-1)} & \alpha_1^{(N-1)} & \alpha_2^{(N-1)} & \cdots & \alpha_{N-2}^{(N-1)} & \beta_1^{(N-1)} & \beta_2^{(N-1)} & \cdots & \beta_{N-2}^{(N-1)} & 0 & 0 \\ \alpha_0^{(N)} & \alpha_1^{(N)} & \alpha_2^{(N)} & \cdots & \alpha_{N-2}^{(N)} & \alpha_{N-1}^{(N)} & \beta_1^{(N)} & \cdots & \beta_{N-3}^{(N)} & \beta_{N-2}^{(N)} & \beta_{N-1}^{(N)} \end{bmatrix}. \tag{3.2}$$

Notice that the kth subdiagonal of R, $k = 0, 1, \ldots, N-1$, contains the elements $\alpha_{m-1}^{(m+k)}$, $m = 1, 2, \ldots, N-k$, the last subdiagonal containing just one element, $\alpha_0^{(N)}$. The analogous β-elements, $\beta_m^{(m+k)}$, $m = 1, 2, \ldots, N-k$, lie on what we call the kth staircase-supdiagonal, $k = 1, 2, \ldots, N-1$, since to get from one of these elements to the next, one has to go down vertically one place and horizontally to the right two places (not one place as for ordinary diagonals). The last of these staircase-diagonals again constains just one element, $\beta_1^{(N)}$. The elements on these diagonals resp. staircase-diagonals were found to have the following striking properties.

First Empirical Observation. (a) *The elements $\alpha_{m-1}^{(m+k)}$, $m = 1, 2, \ldots, N-k$, on the* kth *subdiagonal of R, $k = 0, 1, \ldots, N-1$, for all practical purposes, and irrespective of the value of s, lie on a straight line, i.e., are a linear function of m*. (For $k = N-2$ and $k = N-1$, this of course is a trivial statement). (b) *The elements $\beta_m^{(m+k)}$, $m = 1, 2, \ldots, N-k$, on the kth staircase-supdiagonal of R, $k = 1, 2, \ldots, N-1$, for all practical purposes, and irrespective of the value of s, lie on a parabola, i.e., are a quadratic function of m*. (For $k = N-1, N-2, N-3$, this again is a trivial statement).

This empirical observation is illustrated in Fig. 1 for $s = 1$ and $N = 40$. (The reason for this, as already hinted at in Section 1, may well be that the recursion coefficients α_k^L and β_k^L for the Laguerre polynomials are exactly linear resp. quadratic functions of k.)

Second Empirical Observation. *The linear and quadratic functions in* (a) *resp.* (b) *of the First Empirical Observation are more or less independent of k.*

In Section 3.2 we describe a sequential row-by-row computation of the matrix R, using the results obtained for the α's and β's in the $(n-1)$st row as initial approximations for the corresponding α's and β's in the nth row and estimating the new $\alpha_{n-1}^{(n)}$, $\beta_{n-1}^{(n)}$, which have no matching entries in the preceding row, by an interpolation

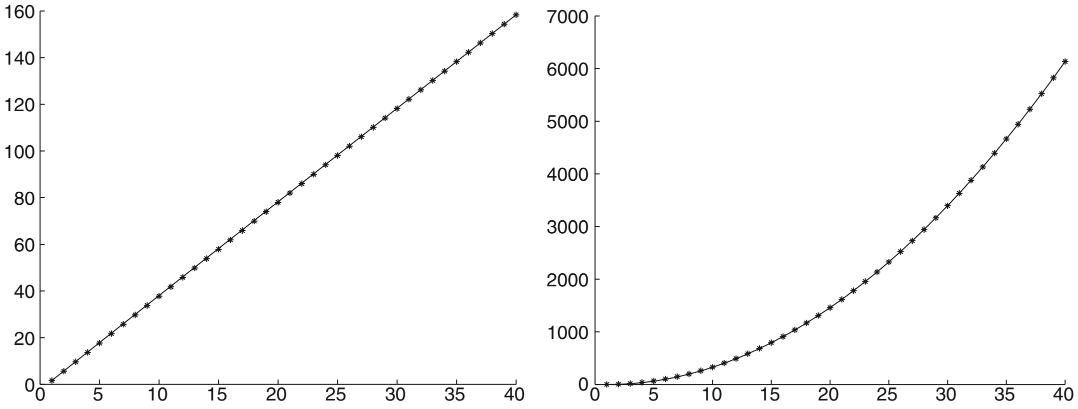


Fig. 1 Linear approximation for $\alpha_{n-1,1}^{(n)}$, $1 \le n \le 40$, (*left*) and quadratic approximation for $\beta_{n,1}^{(n+1)}$, $1 \le n \le 39$, (*right*). The true (computed) values are indicated by *stars*, the approximations by a *solid line*

process. Examining the results so obtained led to the discovery of the two empirical observations above and the determination of the coefficients of the respective linear and quadratic functions. Further examination of these coefficients will suggest a close analytic approximation R_a of R, the first $2n-1$ entries in the nth row of R_a providing excellent initial approximations for the solution of (2.2). This is incorporated in Section 3.3 into a final production code.

3.1 The case $n = 1$

Here, there is just one unknown, $\alpha_0 = \alpha_{0,s}^{(1)}$. By (2.1) for $k = 0$, we have $\pi_{1,s}(t) = t - \alpha_0$, and therefore, by (2.3) for $\nu = 0$,

$$f_1 = \int_0^\infty (\alpha_0 - t)\pi_{1,s}^{2s}(t)\mathrm{d}\lambda(t) = \int_0^\infty (\alpha_0 - t)^{2s+1} e^{-t}\mathrm{d}t.$$

Thus, by the binomial formula, we have $f_1 = 0$ if

$$P_{2s+1}(\alpha) = 0, \quad s \ge 1, \tag{3.3}$$

where $\alpha = \alpha_0$ and

$$P_{2s+1}(\alpha) = \sum_{k=0}^{2s+1} (-1)^k (2s+2-k)_k\, \alpha^{2s+1-k}. \tag{3.4}$$

For the first five values of s, one finds

$$\begin{aligned}
P_3(\alpha) &= \alpha^3 - 3\alpha^2 + 6\alpha - 6 \ \ (s = 1),\\
P_5(\alpha) &= \alpha^5 - 5\alpha^4 + 20\alpha^3 - 60\alpha^2 + 120\alpha - 120 \ \ (s = 2),\\
P_7(\alpha) &= \alpha^7 - 7\alpha^6 + 42\alpha^5 - 210\alpha^4 + 840\alpha^3 - 2520\alpha^2 + 5040\alpha - 5040 \ \ (s = 3),\\
P_9(\alpha) &= \alpha^9 - 9\alpha^8 + 72\alpha^7 - 504\alpha^6 + 3024\alpha^5 - 15120\alpha^4 + 60480\alpha^3 - 181440\alpha^2\\
&\quad +362880\alpha - 362880 \ \ (s = 4),\\
P_{11}(\alpha) &= \alpha^{11} - 11\alpha^{10} + 110\alpha^9 - 990\alpha^8 + 7920\alpha^7 - 55440\alpha^6 + 332640\alpha^5\\
&\quad -1663200\alpha^4 + 6652800\alpha^3 - 19958400\alpha^2 + 39916800\alpha - 39916800 \ \ (s = 5).
\end{aligned}$$

Each of these polynomials has exactly one positive zero $\alpha = \alpha_0$, namely

$$\begin{aligned} \alpha_0 &= 1.596071637983321\ldots \quad (s=1),\\ \alpha_0 &= 2.180607124035125\ldots \quad (s=2),\\ \alpha_0 &= 2.759002709962271\ldots \quad (s=3),\\ \alpha_0 &= 3.333551485269048\ldots \quad (s=4),\\ \alpha_0 &= 3.905451757616910\ldots \quad (s=5). \end{aligned} \tag{3.5}$$

3.2 Numerical procedure for exploration

The procedure[1] described in this section, named `sturan_laguerre_explore.m`, is strictly a preliminary one, serving mainly as a vehicle for exploration. The required recurrence coefficients $\left[\alpha_{0,s}^{(n)}, \alpha_{1,s}^{(n)}, \ldots, \alpha_{n-1,s}^{(n)}; \beta_{1,s}^{(n)}, \ldots, \beta_{n-1,s}^{(n)}\right]$ will be generated successively for $n = 1, 2, 3, \ldots$. When $n = 1$, we need only $\alpha_{0,s}^{(1)}$, which is computed for $s = 1, 2, \ldots, 5$ to the desired (high) accuracy by Newton's method applied to the algebraic equation (3.3), using as initial approximation the respective value of α_0 in (3.5) truncated to Matlab double precision.

For each $n > 1$, we need to solve the nonlinear system (2.2). We do this by applying the Newton–Kantorovich method, a description of which can be found in [1, Section 3.1.3.2]. To compute $\boldsymbol{\rho}_n = \left[\alpha_{0,s}^{(n)}, \alpha_{1,s}^{(n)}, \ldots, \alpha_{n-1,s}^{(n)}; \beta_{1,s}^{(n)}, \ldots, \beta_{n-1,s}^{(n)}\right]$, we use the already computed $\boldsymbol{\rho}_{n-1}$ to provide the initial approximations

$$\alpha_{0,s}^{(n)} \approx \alpha_{0,s}^{(n-1)}, \ldots, \alpha_{n-2,s}^{(n)} \approx \alpha_{n-2,s}^{(n-1)}; \quad \beta_{1,s}^{(n)} \approx \beta_{1,s}^{(n-1)}, \ldots, \beta_{n-2,s}^{(n)} \approx \beta_{n-2,s}^{(n-1)}. \tag{3.6}$$

We still need, however, initial approximations for $\alpha_{n-1,s}^{(n)}, \beta_{n-1,s}^{(n)}$.

Since a quadratic function is uniquely determined by three of its values, we begin by computing $\boldsymbol{\rho}_n$ for $n = 2, 3, 4$, using the procedure outlined above. Suitable initial approximations for $\alpha_{n-1}^{(n)}, \beta_{n-1}^{(n)}$, required for $\boldsymbol{\rho}_n$, are sought in terms of the corresponding Laguerre coefficients $\alpha_{n-1}^L, \beta_{n-1}^L$. By simple trial-and-error we find

$$\left.\begin{aligned} \alpha_1^{(2)} &\approx 1.9\alpha_1^L, & \beta_1^{(2)} &\approx 3.2\beta_1^L\\ \alpha_2^{(3)} &\approx 2.0\alpha_2^L, & \beta_2^{(3)} &\approx 3.7\beta_2^L\\ \alpha_3^{(4)} &\approx 2.0\alpha_3^L, & \beta_3^{(4)} &\approx 3.9\beta_3^L \end{aligned}\right\} \quad s = 1,$$

$$\left.\begin{aligned} \alpha_1^{(2)} &\approx 2.8\alpha_1^L, & \beta_1^{(2)} &\approx 6.7\beta_1^L\\ \alpha_2^{(3)} &\approx 2.9\alpha_2^L, & \beta_2^{(3)} &\approx 8.2\beta_2^L\\ \alpha_3^{(4)} &\approx 3.0\alpha_3^L, & \beta_3^{(4)} &\approx 8.6\beta_3^L \end{aligned}\right\} \quad s = 2,$$

[1] All Matlab procedures referenced or used in this paper can be downloaded from www.cs.purdue.edu/archives/2002/wxg/codes/HPGT.html.

$$\left.\begin{array}{ll}\alpha_1^{(2)} \approx 3.7\alpha_1^L, & \beta_1^{(2)} \approx 11.3\beta_1^L \\ \alpha_2^{(3)} \approx 3.9\alpha_2^L, & \beta_2^{(3)} \approx 14.4\beta_2^L \\ \alpha_3^{(4)} \approx 3.9\alpha_3^L, & \beta_3^{(4)} \approx 15.3\beta_3^L\end{array}\right\} \quad s = 3,$$

$$\left.\begin{array}{ll}\alpha_1^{(2)} \approx 4.6\alpha_1^L, & \beta_1^{(2)} \approx 17.1\beta_1^L \\ \alpha_2^{(3)} \approx 4.8\alpha_2^L, & \beta_2^{(3)} \approx 22.2\beta_2^L \\ \alpha_3^{(4)} \approx 4.9\alpha_3^L, & \beta_3^{(4)} \approx 23.8\beta_3^L\end{array}\right\} \quad s = 4,$$

$$\left.\begin{array}{ll}\alpha_1^{(2)} \approx 5.5\alpha_1^L, & \beta_1^{(2)} \approx 24.1\beta_1^L \\ \alpha_2^{(3)} \approx 5.8\alpha_2^L, & \beta_2^{(3)} \approx 31.8\beta_2^L \\ \alpha_3^{(4)} \approx 5.9\alpha_3^L, & \beta_3^{(4)} \approx 34.2\beta_3^L\end{array}\right\} \quad s = 5.$$

The procedure with these initial approximations converges in 6–7 Newton iterations, even for accuracies as high as 64 decimal digits.

Now that $\alpha_{n-1,s}^{(n)}$, $\beta_{n-1,s}^{(n)}$ have been accurately computed for $n = 2, 3, 4$, we determine the linear function in the First Empirical Observation, part (a), by finding the linear interpolant (in the variable n) taking on the values $\alpha_{1,s}^{(2)}$ and $\alpha_{3,s}^{(4)}$ at $n = 2$ and $n = 4$. Likewise, the quadratic function in part (b) is taken to be the quadratic interpolant based on the values $\beta_{1,s}^{(2)}$, $\beta_{2,s}^{(3)}$, and $\beta_{3,s}^{(4)}$ at $n = 2, 3$, and 4. We obtain (to 4 decimals):

$$\left.\begin{array}{l}\alpha_{n-1}^{(n)} \approx 4.0747n - 2.4358 \\ \beta_{n-1}^{(n)} \approx 4.1540n^2 - 9.0846n + 4.803\end{array}\right\} \quad s = 1,$$

$$\left.\begin{array}{l}\alpha_{n-1}^{(n)} \approx 6.1534n - 3.8822 \\ \beta_{n-1}^{(n)} \approx 9.4751n^2 - 21.3243n + 11.4453\end{array}\right\} \quad s = 2,$$

$$\left.\begin{array}{l}\alpha_{n-1}^{(n)} \approx 8.2335n - 5.333 \\ \beta_{n-1}^{(n)} \approx 16.9644n^2 - 38.7275n + 20.9285\end{array}\right\} \quad s = 3,$$

$$\left.\begin{array}{l}\alpha_{n-1}^{(n)} \approx 10.3142n - 6.786 \\ \beta_{n-1}^{(n)} \approx 26.6224n^2 - 61.2983n + 33.2533\end{array}\right\} \quad s = 4,$$

$$\left.\begin{array}{l}\alpha_{n-1}^{(n)} \approx 12.3954n - 8.2407 \\ \beta_{n-1}^{(n)} \approx 38.4493n^2 - 89.0384n + 48.4185\end{array}\right\} \quad s = 5.$$

These initial approximations, along with those in (3.6), can now be used to run our procedure for $n > 4$ (cf. `runSTL.m`). The largest value N of n for which it worked satisfactorily is shown in Table 1 for the five values of s, along with the condition number of the Jacobian when $n = N$, the magnitude of its absolutely largest element, and the number of digits used in the computation. (Numbers in parentheses indicate

Table 1 Performance data for the explorative procedure in the case of the Laguerre weight function

s	N	Cond	Max element	Digits
1	42	1.2(58)	2.1(250)	80
2	32	1.4(51)	6.2(297)	68
3	20	4.6(36)	1.7(236)	64
4	18	3.0(35)	2.0(275)	64
5	15	3.3(35)	6.8(275)	64

exponents of 10.) By "satisfactory" we mean that the $\alpha_{n-1,s}^{(n)}$ and $\beta_{n-1,s}^{(n)}$ follow the linear resp. quadratic growth with n shown in Fig. 1. What typically happens, as n is increased in steps of 1, is that the condition of the Jacobian matrix in Newton's method gets worse and some of its elements rapidly become very large, so much so that eventually overflow may occur when computing the condition number in Matlab double precision, or Newton's method may fail to converge within 20 iterations. Such breakdowns are usually preceded by a change in the growth pattern shown in Fig. 1, which thus signals trouble ahead.

For each value of s we compute not only the matrix R for the appropriate number N, but also the matrix C of dimension $N \times 5$, where $c_{k+1,1}$, $c_{k+1,2}$ ($k \geq 0$) are the coefficients of the $N-1$ linear functions

$$\alpha_{n-1}^{(n+k)} \approx c_{k+1,1}n + c_{k+1,2}, \; 1 \leq n \leq N-k, \quad k = 0, 1, \ldots, N-2, \tag{3.7}$$

approximating the α-coefficients along the kth subdiagonal of R, with $c_{N,1} = 0$, $c_{N,2} = \alpha_0^{(N)}$, and $c_{k,3}$, $c_{k,4}$, $c_{k,5}$ are the coefficients of the $N-3$ quadratic functions

$$\beta_n^{(n+k)} \approx c_{k,3}n^2 + c_{k,4}n + c_{k,5}, \; 1 \leq n \leq N-k, \quad k = 1, 2, \ldots, N-3, \tag{3.8}$$

approximating the β-coefficients shown on the left, with $c_{N-2,3} = 0$, $c_{N-1,4} = 0$, $c_{N-1,5} = \beta_1^{(N)}$ and $c_{N,3} = c_{N,4} = c_{N,5} = 0$.

The quality of these approximations is quite remarkable and tends to get better with larger k. Selected values of the maximum relative error for the α's and β's, taken over all n with $1 \leq n \leq N-k$, are shown in Table 2 for $s = 1$. They are similar for larger values of s.

A careful look at the matrix C not only confirmed the First and Second Empirical Observations stated at the beginning of Section 3, but also revealed further patterns that allowed us to express the required initial approximations as functions of m and s (at least for $1 \leq s \leq 5$). Specifically, with regard to the α-coefficients, we found that

Table 2 Quality of the approximations (3.7) and (3.8) when $s = 1$ for selected values of k

k	Maxerr α	Maxerr β
0	1.7(−2)	–
1	1.2(−2)	5.5(−2)
5	2.4(−3)	2.0(−2)
15	3.5(−4)	3.7(−3)
35	1.8(−4)	1.2(−5)

for $k = 1, 2, \ldots, N-1$, $1 \le s \le 5$, with $N = N(s)$ as in Table 1,

$$\alpha_{m-1}^{(m+k)} \approx 2(s+1)m - \tfrac{1}{2}(3s+2), \quad m = 1, 2, \ldots, N-k. \tag{3.9}$$

As to the β-coefficients, we found that for $k = 1, 2, \ldots, N-1$, $1 \le s \le 5$, with $N = N(s)$ as in Table 1,

$$\beta_m^{(m+k)} \approx (s+1)m\left((s+1)m - \tfrac{1}{2}\,s\right), \quad m = 1, 2, \ldots, N-k. \tag{3.10}$$

The fact that the approximations in both (3.9) and (3.10) are independent of k is in close agreement with reality when k is relatively large, but less so otherwise.

3.3 A production code for the Laguerre weight function

All ingredients for a production code having been assembled in Section 3.2, it remains to incorporate them into a final routine, named `sturan_laguerre.m`. From (3.9), (3.10) we get for the n-point Gauss–Turán formula the following initial approximations for the $2n-1$ recursion coefficients,

$$\alpha_{\nu-1}^{(n)} \approx 2(s+1)\nu - \tfrac{1}{2}(3s+2), \quad \nu = 1, 2, \ldots, n \quad (1 \le n \le N) \tag{3.11}$$

and

$$\beta_\nu^{(n)} \approx (s+1)\nu\left((s+1)\nu - \tfrac{1}{2}\,s\right), \quad \nu = 1, 2, \ldots, n-1 \;\; (2 \le n \le N). \tag{3.12}$$

These initial approximations have been found quite satisfactory for all $1 \le s \le 5$ and $n \le N$ (where $N = N(s)$ as in Table 1), with only one exception, $s = 5$ and $n = 7$, in which case Newton's iteration failed to converge within 20 iterations. For this reason, we provide in the routine `sturan_laguerre.m` the more accurate approximations

$$\begin{aligned} \alpha_{\nu-1}^{(n)} &\approx c_{n-\nu+1,1}\,\nu + c_{n-\nu+1,2}, \quad && \nu = 1, 2, \ldots, n, \\ \beta_\nu^{(n)} &\approx c_{n-\nu,3}\,\nu^2 + c_{n-\nu,4}\,\nu + c_{n-\nu,5}, \quad && \nu = 1, 2, \ldots, n-1, \end{aligned} \tag{3.13}$$

in terms of the matrix C. Newton's method then converges without exceptions, taking typically 6–8 iterations (but 10 iterations when $n = 6$).

Table 3, produced by the routine `timingSTL.m` and run on an Intel^R, Core^{TM}2, CPU at 1.86GHz with 2 GB main memory, is intended to provide some more information about timing and the number of Newton iterations in the cases $s = 1, 2, \ldots, 5$ for selected values n. The letter "c" in the heading of the table gives the order of magnitude (in powers of 10) of the condition number of the Jacobian matrix in Newton's method. (These were obtained during the explorative work described in Section 3.2.) The letter "d" stands for the number of digits used in the computation. The three values chosen for d are such that the results are expected to have 8, 16, and 32 correct decimal digits. In reality, and in conformity with the nature of Newton's method, they are generally considerably more accurate. In some cases, marked by an asterisk in Table 3, the estimated number d, however, is not sufficient to produce satisfactory results and must be enlarged by as little as 4 (when $s = 3$) and by as much as 20 (when $s = 5$). The reason for this is not entirely clear, but probably has to do with the

Table 3 Timing (in minutes) and number of Newton iterations for the Laguerre weight function in the cases $s = 1 : 5$

s	n	c	$d = c + 8$	$d = c + 16$	$d = c + 32$
1	9	16	2 [5]	2 [6]	3 [7]
	20	35	12 [5]	17 [6]	20 [7]
	31	49	40 [5]	51 [6]	61 [7]
	42	58	130 [5]	187 [7]	215 [8]
2	8	17	1 [5]	2 [6]	2 [7]
	16	34	19 [6]	12 [7]	14 [8]
	24	40	63 [6]	36 [7]	42 [8]
	32	52	100 [7]*	102 [7]	117 [8]
3	5	10	1 [5]	1 [6]	1 [7]
	10	23	4 [6]	4 [7]	5 [8]
	15	34	11 [6]	12 [7]	14 [8]
	20	37	30 [8]*	30 [8]*	34 [9]
4	6	14	1 [6]	1 [7]	1 [8]
	10	24	4 [6]	5 [7]	6 [8]
	14	34	13 [8]*	13 [8]	15 [9]
	18	36	27 [8]*	26 [8]*	30 [9]
5	6	17	2 [9]	2 [10]	2 [11]
	9	23	4 [7]	5 [8]	5 [9]
	12	30	9 [7]*	8 [6]	9 [7]
	15	35	3 [1]*	5 [2]*	7 [3]

Jacobian matrix in Newton's method not only being ill-conditioned, but also having extremely large elements (of the order 10^{275}).

The timings shown in Table 3 are in minutes and include the computation of the weights. The numbers in brackets indicate how many iterations it took for Newton's method to converge. All results are checked by recomputations with d increased by 4 and verifying consistency of the results.

It is to be noted that in the last line for $s = 5$, the number of iterations is exceptionally small. This is so because the initial approximations used in this case, based as they are on the matrix C, are the true values accurate to Matlab double precision, owing to the interpolation scheme adopted to produce the approximations in (3.7), (3.8).

4 Gauss–Turán formulae for the Hermite weight function

In this section we let $\mathrm{d}\lambda(t) = e^{-t^2}\mathrm{d}t$ on $\mathbb{R}$. Because of symmetry, $\alpha_{\nu,s}^{(n)} = 0$ for $\nu = 0, 1, \ldots, n-1$ and all n and s. The first set of n equations $f_{2\nu+1} = 0$ in (2.3), therefore, is trivially satisfied. We need to be concerned only with the second set of $n-1$ equations $f_{2\nu} = 0$, assuming $n \geq 2$. (The Gauss–Turán formula (1.1) in the case $n = 1$ is simply obtained by Taylor expansion of $f(t)$ at the point $t = \tau_1 = 0$.)

4.1 The case $n = 2$

There is just one unknown, $\beta_1 = \beta_{1,s}^{(2)}$, the first nonlinear equation being $f_2 = 0$. From (2.3) with $\nu = 1$, we have

$$f_2 = \int_{\mathbb{R}} (\beta_1 - t^2)\pi_{2,s}^{2s}(t)\mathrm{d}\lambda(t) = 2\int_0^\infty (\beta_1 - t^2)^{2s+1} e^{-t^2}\,\mathrm{d}t.$$

Letting $t^2 = \tau$ and $\beta = \beta_1$, we get

$$f_2 = \int_0^\infty (\beta - \tau)^{2s+1} \tau^{-1/2} e^{-\tau}\,\mathrm{d}\tau,$$

hence, by the binomial formula,

$$f_2 = \sum_{k=0}^{2s+1} (-1)^k \binom{2s+1}{k} \Gamma(k + 1/2)\beta^{2s+1-k}.$$

Since $\Gamma(k + 1/2) = (-1)^k(-1/2)_k\sqrt{\pi}$, the equation $f_2 = 0$ yields the algebraic equation

$$Q_{2s+1}(\beta) = 0, \quad s \geq 1,$$

where

$$Q_{2s+1}(\beta) = \sum_{k=0}^{2s+1} \binom{2s+1}{k} (-1/2)_k\, \beta^{2s+1-k}. \tag{4.1}$$

For the first five values of s, one finds, after clearing all fractions,

$$\begin{aligned}
Q_3(\beta) &= 8\beta^3 - 12\beta^2 + 18\beta_1 - 15 \quad (s = 1),\\
Q_5(\beta) &= 32\beta^5 - 80\beta^4 + 240\beta^3 - 600\beta^2 + 1050\beta - 945 \quad (s = 2),\\
Q_7(\beta) &= 128\beta^7 - 448\beta^6 + 2016\beta^5 - 8400\beta^4 + 29400\beta^3 - 79380\beta^2\\
&\quad + 145530\beta - 135135 \quad (s = 3),\\
Q_9(\beta) &= 512\beta^9 - 2304\beta^8 + 13824\beta^7 - 80640\beta^6 + 423360\beta^5 - 1905120\beta^4\\
&\quad + 6985440\beta^3 - 19459440\beta^2 + 36486450\beta - 34459425 \quad (s = 4),\\
Q_{11}(\beta) &= 2048\beta^{11} - 11264\beta^{10} + 84480\beta^9 - 633600\beta^8 + 4435200\beta^7\\
&\quad - 27941760\beta^6 + 153679680\beta^5 - 713512800\beta^4 + 2675673000\beta^3\\
&\quad - 7581073500\beta^2 + 14404039650\beta - 13749310575 \quad (s = 5).
\end{aligned}$$

Each of these polynomials has exactly one real zero, namely

$$\begin{array}{ll}
1.053573782217666\ldots & (s = 1),\\
1.609349223985799\ldots & (s = 2),\\
2.165699416895352\ldots & (s = 3),\\
2.722281296487453\ldots & (s = 4),\\
3.278979167347799\ldots & (s = 5).
\end{array} \tag{4.2}$$

4.2 A production code for the Hermite weight function

Preliminary explorative work is similar to what was done for the Laguerre weight function in Section 3.2, but simpler, because all α-coefficients are zero and only the β-coefficients need to be determined. We therefore omit details. Rather than writing new software specifically for symmetric measures (which, in fact, would be preferable), we simply use existing software and take zero initial approximations for the α's.

What was found empirically can again be expressed in two observations.

First Empirical Observation *The elements $\beta_m^{(m+k)}$, $m = 1, 2, \ldots, N-k$, on the* kth *staircase-supdiagonal of R, $k = 1, 2, \ldots, N-1$, for all practical purposes, and irrespective of the value of s, lie on a straight line, i.e., are linear functions of m.* (For $k = N-2$ and $k = N-1$, this is a trivial statement.)

This is illustrated in Fig. 2 for $s = 1$ and $N = 40$.

Second Empirical Observation *The linear functions of the First Empirical Observation are more or less independent of k.*

The coefficients of these linear functions are stored in a $(N-1) \times 2$ array C, so that

$$\beta_m^{(m+k)} \approx c_{k,1}m + c_{k,2}, \quad m = 1, 2, \ldots, N-k, \tag{4.3}$$

where for $k = N-1$ (hence $m = 1$),

$$c_{N-1,1} = 0, \quad c_{N-1,2} = \beta_1^{(N)}. \tag{4.4}$$

Setting $m+k = n$, we can write $\beta_{n-k}^{(n)} \approx c_{k,1}(n-k)+c_{k,2}$, and since $1 \le n-k \le n-1$ (the first inequality coming from $m \ge 1$, the second from $k \ge 1$), letting $n-k = \nu$, we get

$$\beta_\nu^{(n)} \approx c_{n-\nu,1}\nu + c_{n-\nu,2}, \quad \nu = 1, 2, \ldots, n-1, \tag{4.5}$$

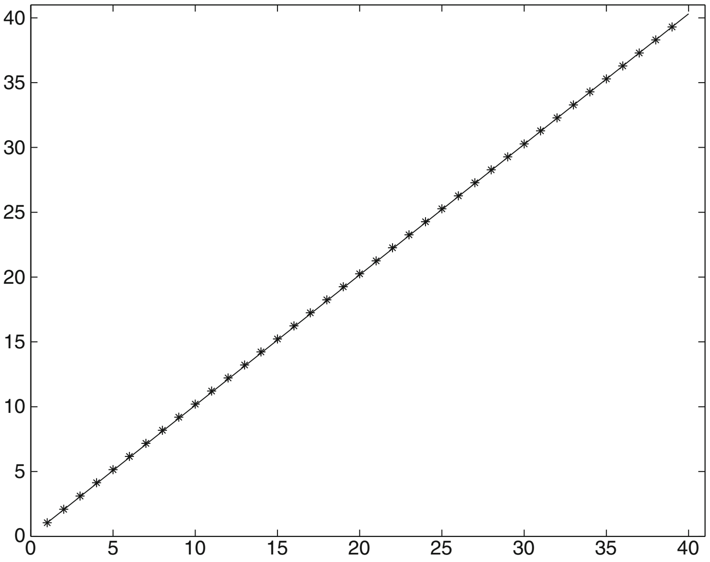


Fig. 2 Linear approximation for $\beta_{n-1,1}^{(n)}$, $2 \le n \le 40$

Table 4 Performance data for the production procedure in the case of the Hermite weight function, including timing (in hours)

s	N	Cond	Max element	Digits	Timing
1	90	6.9(65)	6.4(274)	86	65.8
2	65	2.2(53)	3.9(304)	72	45.0
3	49	5.1(47)	1.4(307)	72	16.5
4	39	1.9(40)	6.9(306)	64	8.1
5	32	1.7(36)	3.7(301)	64	5.7

which provides good initial approximations for Newton's method. (Note that (4.5) is in fact an equality when $n = N$ and $\nu = 1$, because of (4.4)). By the Second Empirical Observation, the coefficients $c_{n-\nu,1}$ and $c_{n-\nu,2}$ are practically independent of ν. The coefficients, however, depend on s. Looking at the numerical values stored in C, we find that with an error of at most a few percent, there holds

$$\beta_\nu^{(n)} \approx \frac{s+1}{2}\,\nu, \quad 1 \le s \le 5. \tag{4.6}$$

Using the simple initial approximations (4.6), one finds that the largest value of N for which the procedure works satisfactorily is now considerably larger than in the case of the Laguerre weight function, but the run times are correspondingly large.

Table 5 Timing (in minutes) and number of Newton iterations for the Hermite weight function in the cases $s = 1:5$

s	n	c	$d = c + 8$	$d = c + 16$	$d = c + 32$
1	9	6	1 [4]	2 [5]	2 [6]
	20	20	12 [5]	17 [6]	20 [7]
	31	31	39 [5]	50 [6]	59 [7]
	42	37	128 [5]	156 [6]	184 [7]
2	8	7	1 [5]	2 [6]	2 [7]
	16	17	9 [5]	11 [6]	13 [7]
	24	25	25[5]	32 [6]	39 [7]
	32	33	86 [6]	101 [7]	118 [8]
3	5	4	1 [5]	1 [6]	1 [7]
	10	10	3 [5]	4 [6]	5 [7]
	15	18	11 [6]	13 [7]	15 [8]
	20	27	23 [6]	27 [7]	31 [8]
4	6	6	1 [6]	1 [7]	2 [8]
	10	11	4 [6]	5 [7]	6 [8]
	14	17	11 [6]	13 [7]	14 [8]
	18	26	21 [6]	24 [7]	28 [8]
5	6	7	1 [6]	2 [7]	2 [8]
	9	10	4 [6]	5 [7]	5 [8]
	12	15	8 [6]	10 [7]	11 [8]
	15	21	17 [7]	19 [8]	21 [9]

This is shown in Table 4, the analogon to Table 1 in Section 3.2, which is produced by the routine `runSTH.m`, using `sturan_hermite.m`. For $s = 1$, it is possible that values of N larger than 90 may still be satisfactory. We have not examined this in detail because of the very large run time involved. The number of Newton iterations is relatively modest: 7 for $s \leq 3$ and 8 for $s = 4$ and 5.

To facilitate comparison with the Laguerre case (and to avoid very long run times), we provide more information about timings and number of Newton iterations for the same values of s and n as in Table 3. They are produced by the routine `timingSTH.m` and listed in Table 5. It can be seen that run times, and often also the number of Newton iterations, are generally smaller than in the Laguerre case.

Acknowledgments The author is indebted to Gradimir V. Milovanović for useful comments.

References

1. Gautschi, W.: Orthogonal polynomials: computation and approximation. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2004)
2. Milovanović, G.V.: Construction of s-orthogonal polynomials and Turán quadrature formulae. In: Milovanović, G.V. (ed.) Numerical methods and approximation III (Niš, 1987), pp. 311–328. Univ. Niš, Niš (1988)
3. Milovanović, G.V., Spalević, M.M.: Quadrature formulae connected to σ-orthogonal polynomials. J. Comput. Appl. Math **240**, 619–637 (2002)
4. Milovanović, G.V., Spalević, M.M., Cvetković, A.S.: Calculation of Gaussian-type quadratures with multiple nodes. Math. Comput. Model. **39**, 325–347 (2004)
5. Shi, Y.G., Xu, G.: Construction of σ-orthogonal polynomials and gaussian quadrature formulas. Adv. Comput. Math. **27**, 79–94 (2007)
6. Stroud, A.H., Stancu, D.D.: Quadrature formulas with multiple Gaussian nodes. J. SIAM Numer. Math. Ser. B **2**, 129–143 (1965)

46

Papers on Ordinary Differential Equations

15 (with H. A. Antosiewicz), Numerical methods in ordinary differential equations, Ch. 9 in: Survey of numerical analysis (J. Todd, ed.), 314–346, McGraw–Hill, New York, 1962

56 Comportement asymptotique des coefficients dans les formules d'intégration d'Adams, de Störmer et de Cowell, C. R. Acad. Sci. Paris Sér. A-B, 283, no. 10, Aiii, A787–A788 (1976)

46.1. [15] (with H. A. Antosiewicz) “Numerical methods in ordinary differential equations”

[15] “Numerical methods in ordinary differential equations”, Ch. 9 in *Survey of numerical analysis* (J. Todd. ed.), 314–346, McGraw–Hill, New York, 1962.

9

Numerical Methods in Ordinary Differential Equations

HENRY A. ANTOSIEWICZ
PROFESSOR OF MATHEMATICS
UNIVERSITY OF SOUTHERN CALIFORNIA
LOS ANGELES

WALTER GAUTSCHI
OAK RIDGE NATIONAL LABORATORY

9.1 Introduction

We consider, throughout, methods for the numerical solution of the initial-value problem for systems of real ordinary differential equations of first order,

$$\frac{dy^\mu}{dx} = f^\mu(x, y^1, \ldots, y^m), \qquad y^\mu(x_0) = y_0{}^\mu \qquad (\mu = 1, 2, \ldots, m).$$

Our aim is to give the fundamental principles on which most of these methods are based, rather than to attempt a complete presentation of all such methods. For a more detailed treatment and a discussion of methods for the numerical solution of boundary-value and eigenvalue problems, we refer to the books listed in the bibliography—for example, to those of Collatz, where further references to the ample literature on the subject can be found.

In the following we always write the basic system in vector form,*

$$\frac{dy}{dx} = f(x,y), \qquad y(x_0) = y_0, \tag{9.1}$$

* To attain more symmetry, some authors write (9.1) as

$$\frac{dw}{dx} = F(w), \qquad w(x_0) = w_0,$$

by setting $w = \begin{pmatrix} x \\ y \end{pmatrix}$, $F = \begin{pmatrix} 1 \\ f \end{pmatrix}$, $w_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$. This artifice appears to go back to Nyström ([46], p. 9).

and denote by $\|y\|$ the usual euclidean length of y. Without further mention, we tacitly assume that there is a set $I \times R$ containing (x_0,y_0) in its interior on which $f(x,y)$ is continuous and satisfies

$$\|f(x,u) - f(x,v)\| \leq K \|u - v\| \tag{9.2}$$

for some constant $K > 0$, so that there exists on some interval $[x_0,X] \subset I$ a unique solution $y(x)$ with $y(x_0) = y_0$. Recall that, if $f(x,y)$ is of class C^s, $s \geq 1$, on $I \times R$, then $y(x)$ is of class C^{s+1} on $[x_0,X]$ and, if R is convex and $\|\partial f(x,y)/\partial y^\mu\|$ is bounded on $I \times R$ for each $\mu = 1, 2, \ldots, m$, then (9.2) holds.

All methods for the numerical solution of (9.1) considered in the following yield, for a sequence $\{x_n\}$ of abscissas $x_n > x_0$, a sequence $\{y_n\}$ of vectors y_n which approximate to the (exact) vectors $y(x_n)$ of the desired solution. In the first part, Secs. 9.2 to 9.5, we consider the so-called one-step methods, which define y_{n+1} as a function of x_{n+1}, x_n, y_n. A famous representative of this group of methods is Kutta's generalization of Simpson's rule. The second part, Secs. 9.6 to 9.10, is devoted to multistep methods, in which y_{n+1} is defined as a function of x_{n+1}, $x_{n-\rho}$, $y_{n-\rho}$, $\rho = 0, 1, \ldots, k$ for some $k > 0$. Here, the classical example is Adams' method.

The kind of difficulty one is likely to face is illustrated by the simple equation,

$$\frac{dy}{dx} = \begin{pmatrix} 0 & 1 \\ 10a^2 & 9a \end{pmatrix} y, \qquad y(0) = \begin{pmatrix} 1 \\ -a \end{pmatrix}, \tag{9.3}$$

in which $a > 0$ is some constant. Its general solution is

$$y(x) = c_1 e^{-ax}\begin{pmatrix} 1 \\ -a \end{pmatrix} + c_2 e^{10ax}\begin{pmatrix} 1 \\ 10a \end{pmatrix},$$

and the particular solution desired is obtained for $c_1 = 1$, $c_2 = 0$. No matter what approximate method is used, it will introduce a c_2-component due to round-off errors, and if at $x = \xi$ this component is η, it will be ηe^{20a} at $x = \xi + 2$. Thus, it will ultimately totally overshadow the desired solution and lead to entirely spurious results, regardless of how many decimals are carried in the computation.

Though in a particular problem the situation may not be so bad, this example should serve nevertheless as a warning: numerical integration should not be undertaken when no definite information about the desired solution is available. Careful analysis of the problem at hand must always precede the start of computations.

RUNGE-KUTTA METHODS

9.2 Method of Taylor's Series and General Runge-Kutta Method

Every one-step method can be written in the form

$$y_{n+1} = y_n + h_n\varphi(x_n,y_n;h_n) \qquad (n = 0, 1, 2, \ldots), \tag{9.4}$$

where $\varphi(x,y;h)$ is a vector-valued function and $h_n = x_{n+1} - x_n$. The choice of φ should be "reasonable" in the sense that for fixed $(x,y) \in I \times R$

$$\varphi(x,y;h) \to f(x,y) \qquad \text{as } h \to 0. \tag{9.5}$$

If $y(x)$ is the exact solution of the differential equation and

$$r(x,h) = y(x) + h\varphi(x,y(x);h) - y(x + h), \tag{9.6}$$

then for any fixed $x \in [x_0,X]$

$$r(x,h) = o(h) \qquad \text{as } h \to 0.$$

The vector $r(x,h)$ is called the *truncation error* at the point x. If p is the largest integer p' with the property that

$$r(x,h) = O(h^{p'+1}) \qquad \text{as } h \to 0, \tag{9.7}$$

then p is called the *order* of the method (9.4).

The simplest choice of φ satisfying (9.5) is

$$\varphi(x,y;h) = f(x,y).$$

The corresponding method,*

$$y_{n+1} = y_n + hf(x_n,y_n), \tag{9.8}$$

was proposed by Euler in 1768. If $f \in C^1$, it has the order $p = 1$, since

$$r(x,h) = y(x) + hf[x,y(x)] - y(x + h) = y(x) + hy'(x) - y(x + h) = O(h^2).$$

This also shows that the method, in essence, makes use of the first two terms of Taylor's series.

A natural extension of (9.8), then, is the so-called *method of Taylor's series* (also proposed by Euler), which takes into account the first $(p + 1)$ terms of Taylor's series. To describe it, let $f \in C^p$, $p > 1$, and set

$$f^{[0]}(x,y) = f(x,y), f^{[k+1]}(x,y) = \frac{\partial f^{[k]}(x,y)}{\partial x} + \frac{\partial f^{[k]}(x,y)}{\partial y} f(x,y)$$
$$(k = 0, 1, \ldots, p - 2) \tag{9.9}$$

* In this and in similar formulas in the sequel, we write simply h instead of h_n, with the understanding that h may or may not depend on n.

where $\partial f^{[k]}/\partial y$ denotes the $m \times m$ matrix having as columns $\partial f^{[k]}/\partial y^{\mu}$ ($\mu = 1, 2, \ldots, m$). Clearly,

$$f^{[k]}(x,y(x)) = y^{(k+1)}(x). \tag{9.10}$$

Taking, then,

$$\varphi(x,y;h) = \sum_{k=0}^{p-1} \frac{h^k}{(k+1)!} f^{[k]}(x,y),$$

we are led to a method of order p, since, by (9.6), (9.10),

$$r(x,h) = y(x) + h \sum_{k=0}^{p-1} \frac{h^k}{(k+1)!} y^{(k+1)}(x) - y(x+h) = O(h^{p+1}).$$

Although for special systems of differential equations, notably linear systems, the method of Taylor's series may be used quite efficiently (for an example, see Chap. 2), its applicability in more general cases is rather limited, because of the rapidly increasing complexity of the expressions in (9.9). It was Runge [52] who, in 1895, first pointed out a possibility of evading successive differentiations and of preserving at the same time the increased accuracy afforded by Taylor's series. Runge's method was subsequently improved by Heun [26] and Kutta [32].

Kutta's proposal (somewhat more general than a similar one made by Heun) consists in setting up φ with undetermined parameters as follows:

$$\begin{aligned} \varphi(x,y;h) &= \sum_{s=1}^{r} \alpha_s k_s, \\ k_1(x,y) &= f(x,y), \\ k_s(x,y;h) &= f\left(x + \mu_s h,\ y + h \sum_{j=1}^{s-1} \lambda_{sj} k_j\right) \qquad (s = 2, \ldots, r). \end{aligned} \tag{9.11}$$

Given r, the number of "substitutions" into f, the parameters α_s, μ_s, λ_{sj} are to be determined so as to make the order p of (9.11) as large as possible.

We note that in (9.11) $f \in C^q$ implies $\varphi \in C^q$ for h sufficiently small.

Expanding the local truncation error formally into Taylor's series,

$$r(x,h) = \sum_{k=0}^{\infty} \frac{1}{k!} \left[\frac{\partial^k r(x,h)}{\partial h^k}\right]_{h=0} h^k,$$

and noting from (9.6) that $r(x,0) = 0$ and that

$$\left[\frac{\partial^k r(x,h)}{\partial h^k}\right]_{h=0} = k\left[\frac{\partial^{k-1}\varphi(x,y(x);h)}{\partial h^{k-1}}\right]_{h=0} - y^{(k)}(x) \qquad (k > 0),$$

we find that Kutta's method is of order p, if $f \in C^p$ and

$$\left[\frac{\partial^{k-1}\varphi(x,y(x);h)}{\partial h^{k-1}}\right]_{h=0} - \frac{1}{k} y^{(k)}(x) \quad \begin{cases} = 0 & \text{for } 1 \le k \le p, \\ \ne 0 & \text{for } k = p+1. \end{cases} \tag{9.12}$$

More detailed calculations show that the p identities in (9.12) are equivalent to a set of, in general, nonlinear equations for the parameters α_s, μ_s, λ_{sj}. For each r, there will be a largest value of p, $p = p^*(r)$, for which these equations are solvable, and it turns out that

$$p^*(r) = r \qquad \text{if } 1 \leq r \leq 4.$$

The corresponding solutions have a certain degree of freedom, which is 1 for $r = 2$ and 2 for $r = 3$ and $r = 4$. Their actual derivation, however, is of considerable length, and the calculations become rapidly more complex as r is further increased. For example, if $r = 5$, Kutta [32] obtains 16 equations in 15 unknowns, and it appears as yet uncertain whether these equations are dependent. Thus, $p^*(5) \geq 4$; similarly one knows only that $p^*(6) \geq 5$. Corresponding formulas of order 4 and 5 are listed in [46]. Formulas of order 6 utilizing eight substitutions are derived in [28,29].

9.3 Examples of Runge-Kutta Formulas

To derive a particular set of Runge-Kutta formulas for a fixed $r(\geq 1)$, one proceeds as follows. First one obtains, in terms of the partial derivatives of $f(x,y)$, the partial derivatives of $\varphi(x,y;h)$ with respect to h, by using (9.11), and the (total) derivatives of $y(x)$, by using (9.1); this requires $(p - 1)$ differentiations if a method of order p is desired. Then one substitutes these expressions into (9.12) and satisfies the p identities in (9.12) by equating to zero the coefficients of the various partial derivatives of $f(x,y)$. Under the natural assumption that

$$\mu_s = \sum_{j=1}^{s-1} \lambda_{sj} \qquad (s = 2, \ldots, r), \tag{9.13}$$

one finds, for example, for $p = 1$, 2, or 3, that the first, the first two, or all three of the following equations must be satisfied:

$$\sum_{s=1}^{r} \alpha_s = 1, \tag{9.14}$$

$$\sum_{s=2}^{r} \alpha_s\mu_s = \tfrac{1}{2}, \tag{9.15}$$

$$\sum_{s=2}^{r} \alpha_s\mu_s^2 = 2\sum_{s=3}^{r} \alpha_s \sum_{i=2}^{s-1} \lambda_{si}\mu_i = \tfrac{1}{3}. \tag{9.16}$$

Suppose we require only a method of order $p \geq 1$, so that solely (9.14) must hold. If $r = 1$, then (9.13) drops out, and $\alpha_1 = 1$, which yields Euler's point-slope formula (9.8). If $r > 1$, we have to satisfy r equations with $2r - 1 + \tfrac{1}{2}r(r - 1)$ variables, which leaves us $r - 1 + \tfrac{1}{2}r(r - 1)$ free parameters.

Similarly, for a Runge-Kutta method of order $p \geq 2$, both (9.14) and (9.15) must hold, which requires $r \geq 2$. There are $r - 2 + \frac{1}{2}r(r-1)$ degrees of freedom in the choice of the parameters. For example, if $r = 2$, we have

$$\alpha_1 = 1 - \alpha_2, \qquad \mu_2 = \lambda_{21} = \frac{1}{2\alpha_2}.$$

Note that $r = 2$ violates the second relation in (9.16) so that $p^*(2) = 2$.

In the *modified Euler-Cauchy method* (or the improved point-slope formula) the choice $\alpha_2 = 1$ is made, for which (9.4) reads

$$y_{n+1} = y_n + hf[x_n + \tfrac{1}{2}h,\ y_n + \tfrac{1}{2}hf(x_n, y_n)]. \tag{9.17}$$

In *Heun's method* (or the improved Euler-Cauchy method) $\alpha_2 = \frac{1}{2}$ is chosen, for which (9.4) reduces to

$$y_{n+1} = y_n + \tfrac{1}{2}h\{f(x_n, y_n) + f[x_n + h, y_n + hf(x_n, y_n)]\}. \tag{9.18}$$

Runge-Kutta formulas of order $p \geq 3$ can be obtained if $r \geq 3$; they can be chosen from a set of formulas with degree of freedom $[r - 4 + \frac{1}{2}r(r-1)]$. If $r = 3$, they are given by

$$\alpha_1 = \frac{2 - 3(\mu_2 + \mu_3) + 6\mu_2\mu_3}{6\mu_2\mu_3},$$

$$\alpha_2 = \frac{2 - 3\mu_3}{6\mu_2(\mu_2 - \mu_3)}, \qquad \alpha_3 = \frac{2 - 3\mu_2}{6\mu_3(\mu_3 - \mu_2)},$$

$$\lambda_{21} = \mu_2, \qquad \lambda_{31} = \frac{\mu_3(3\mu_2 - \mu_3 - 3\mu_2^2)}{\mu_2(2 - 3\mu_2)}, \qquad \lambda_{32} = \frac{\mu_3(\mu_3 - \mu_2)}{\mu_2(2 - 3\mu_2)},$$

as follows from (9.13) to (9.16). If $\mu_2 = \frac{1}{3}$, $\mu_3 = \frac{2}{3}$ is chosen, then (9.4) becomes

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{4}h(k_1 + 3k_3), \\ k_1 &= f(x_n, y_n), \\ k_3 &= f\{x_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}hf[x_n + \tfrac{1}{3}h, y_n + \tfrac{1}{3}hf(x_n, y_n)]\}, \end{aligned} \tag{9.19}$$

which is also referred to as *Heun's formula*. The choice $\mu_2 = \frac{1}{2}$, $\mu_3 = 1$ yields *Kutta's third-order rule*:

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{6}h(k_1 + 4k_2 + k_3), \\ k_1 &= f(x_n, y_n), \\ k_2 &= f[x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hf(x_n, y_n)], \\ k_3 &= f\{x_n + h, y_n - hf(x_n, y_n) + 2hf[x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hf(x_n, y_n)]\}. \end{aligned} \tag{9.20}$$

To find Runge-Kutta formulas of order $p = 4$, we must have $r \geq 4$. If $r = 4$, there are obtained 11 equations containing 13 parameters. A particular solution suggested by Kutta is the set of constants

$$\begin{gathered}
\alpha_1 = \alpha_4 = \tfrac{1}{6}, \qquad \alpha_2 = \alpha_3 = \tfrac{1}{3}, \\
\mu_2 = \mu_3 = \tfrac{1}{2}, \qquad \mu_4 = 1, \\
\lambda_{21} = \lambda_{32} = \tfrac{1}{2}, \qquad \lambda_{43} = 1, \qquad \lambda_{31} = \lambda_{41} = \lambda_{42} = 0.
\end{gathered}$$

This is the choice made in Runge-Kutta's method, for which (9.4) reads

$$\begin{aligned}
y_{n+1} &= y_n + \tfrac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), \\
k_1 &= f(x_n, y_n), \\
k_2 &= f(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_1), \\
k_3 &= f(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_2), \\
k_4 &= f(x_n + h, y_n + hk_3).
\end{aligned} \tag{9.21}$$

Note, that, if f is independent of y, both formulas (9.20) and (9.21) reduce to Simpson's rule.

Gill [19] proposes a solution for the same 11 equations under the additional requirement that the vectors $y_n + h(\lambda_{i1}k_1 + \lambda_{i2}k_2)$, $i = 3, 4$, and $y_n + h(\alpha_1 k_1 + \alpha_2 k_2)$ be linearly dependent. He adopts, from among others, the set of constants

$$\begin{gathered}
\alpha_1 = \alpha_4 = \tfrac{1}{6}, \qquad \alpha_2 = \tfrac{1}{3}(1 - \sqrt{\tfrac{1}{2}}), \qquad \alpha_3 = \tfrac{1}{3}(1 + \sqrt{\tfrac{1}{2}}), \\
\mu_2 = \mu_3 = \tfrac{1}{2}, \qquad \mu_4 = 1, \\
\lambda_{21} = \tfrac{1}{2}, \qquad \lambda_{31} = -\tfrac{1}{2} + \sqrt{\tfrac{1}{2}}, \qquad \lambda_{32} = 1 - \sqrt{\tfrac{1}{2}}, \\
\lambda_{41} = 0, \qquad \lambda_{42} = -\sqrt{\tfrac{1}{2}}, \qquad \lambda_{43} = 1 + \sqrt{\tfrac{1}{2}},
\end{gathered}$$

with which (9.4) becomes

$$\begin{aligned}
y_{n+1} &= y_n + \tfrac{1}{6}h[k_1 + 2(1 - \sqrt{\tfrac{1}{2}})k_2 + 2(1 + \sqrt{\tfrac{1}{2}})k_3 + k_4], \\
k_1 &= f(x_n, y_n), \\
k_2 &= f(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}hk_1), \\
k_3 &= f(x_n + \tfrac{1}{2}h, y_n - (\tfrac{1}{2} - \sqrt{\tfrac{1}{2}})hk_1 + (1 - \sqrt{\tfrac{1}{2}})hk_2), \\
k_4 &= f(x_n + h, y_n - \sqrt{\tfrac{1}{2}}hk_2 + (1 + \sqrt{\tfrac{1}{2}})hk_3).
\end{aligned} \tag{9.22}$$

Still other formulas are described in [31] and in [13, 38], where also their adoption for use on high-speed computing machines is discussed.

More accurate variants of Runge-Kutta's method, involving also substitutions into partial derivatives of f, are given in [17, 36, 47]. Further one-step methods with

$$\varphi(x,y;h) = \sum_{s=1}^{r} \alpha_s f(x + \mu_s h, y + \mu_s h \bar{\varphi}_s(x,y;\mu_1 h, \ldots, \mu_r h)),$$

where the $\bar{\varphi}_s$ represent auxiliary one-step methods, are considered in [23, 25].

9.4 Differential Equations of Higher Order

Because of their importance we consider in more detail systems of second-order differential equations

$$v'' = g(x,v,v'), \qquad v(x_0) = v_0, \qquad v'(x_0) = v'_0, \tag{9.23}$$

where v and g are vector-valued functions and g is sufficiently smooth on a suitable set $I \times R \times S$.

The system (9.23) can be written as a first-order system [of twice the size of (9.23)] if we introduce the column vector $w = v'$ and write

$$y = \begin{pmatrix} v \\ w \end{pmatrix}, \qquad f(x,y) = \begin{pmatrix} w \\ g(x,v,w) \end{pmatrix}, \qquad y_0 = \begin{pmatrix} v_0 \\ w_0 \end{pmatrix}. \tag{9.24}$$

Then (9.23) is equivalent to

$$y' = f(x,y), \qquad y(x_0) = y_0. \tag{9.25}$$

Any method of Sec. 9.3 can now be applied to the system (9.25). The resulting formulas may then again be separated into their v and w parts to obtain formulas for the computation of v and $w = v'$.

For the Runge-Kutta formula (9.21) we find, after a short calculation,

$$\begin{aligned} v_{n+1} &= v_n + hv'_n + \tfrac{1}{6}h^2(l_1 + l_2 + l_3), \\ v'_{n+1} &= v'_n + \tfrac{1}{6}h(l_1 + 2l_2 + 2l_3 + l_4), \end{aligned} \tag{9.26}$$

where

$$\begin{aligned} l_1 &= g(x_n,v_n,v'_n), \\ l_2 &= g(x_n + \tfrac{1}{2}h,\ v_n + \tfrac{1}{2}hv'_n,\ v'_n + \tfrac{1}{2}hl_1), \\ l_3 &= g(x_n + \tfrac{1}{2}h,\ v_n + \tfrac{1}{2}hv'_n + \tfrac{1}{4}h^2 l_1,\ v'_n + \tfrac{1}{2}hl_2), \\ l_4 &= g(x_n + h,\ v_n + hv'_n + \tfrac{1}{2}h^2 l_2,\ v'_n + hl_3). \end{aligned} \tag{9.27}$$

By construction, this method is of order 4, in the sense that

$$v_1 - v(x_0 + h) = O(h^5), \qquad v'_1 - v'(x_0 + h) = O(h^5). \tag{9.28}$$

In analogy with Kutta's procedure [see (9.11)], one could start from the outset with a system of formulas of type (9.26), (9.27) and use undetermined coefficients to search for more accurate or more convenient formulas. This was done in 1925 by Nyström [46], who sets

$$\begin{aligned} v_{n+1} &= v_n + hv'_n + h^2 \sum_{s=1}^{r} \alpha_s l_s, \\ v'_{n+1} &= v'_n + h \sum_{s=1}^{r} \beta_s l_s, \end{aligned} \tag{9.29}$$

with $l_1 = g(x_n, v_n, v'_n)$,

$$l_s = g(x_n + \mu_s h,\, v_n + \mu_s h v'_n + h^2 \sum_{j=1}^{s-1} \lambda_{sj} l_j,\, v'_n + h \sum_{j=1}^{s-1} \kappa_{sj} l_j) \qquad (s = 2, \ldots, r). \tag{9.30}$$

He finds, among others, the following particular simple set of formulas:

$$\begin{aligned} l_1 &= g(x_n, v_n, v'_n), \\ l_2 &= g(x_n + \tfrac{1}{2}h,\, v_n + \tfrac{1}{2}hv'_n + \tfrac{1}{8}h^2 l_1,\, v'_n + \tfrac{1}{2}hl_1), \\ l_3 &= g(x_n + \tfrac{1}{2}h,\, v_n + \tfrac{1}{2}hv'_n + \tfrac{1}{8}h^2 l_1,\, v'_n + \tfrac{1}{2}hl_2), \\ l_4 &= g(x_n + h,\, v_n + hv'_n + \tfrac{1}{2}h^2 l_3,\, v'_n + hl_3), \end{aligned} \tag{9.31}$$

from which v_{n+1}, v'_{n+1} is obtained as in (9.26). The resulting method is also of order 4 but in general involves fewer calculations than (9.27), since the first two arguments in l_2 and l_3 are the same.

A method which extends Nyström's method (9.26), (9.31) to systems of mth-order equations

$$v^{(m)} = g(x, v, v', \ldots, v^{(m-1)}), \qquad v^{(\mu)}(x_0) = v_0{}^{(\mu)} \qquad (\mu = 0, 1, \ldots, m-1) \tag{9.32}$$

was developed by Zurmühl [72, 73]. To describe it, let

$$T_\mu(\alpha) = v_n{}^{(\mu)} + \alpha h v_n{}^{(\mu+1)} + \cdots + \frac{(\alpha h)^{m-\mu-1}}{(m-\mu-1)!} v_n{}^{(m-1)} \qquad (0 \le \mu \le m-1)$$

denote truncated Taylor series for the μth derivative at $x = x_n + \alpha h$, based on the available derivatives at $x = x_n$. Using the short-hand notation $g(x, u_\mu)$ for $g(x, u_0, u_1, \ldots, u_{m-1})$, set

$$\begin{aligned} l_1 &= g[x_n, T_\mu(0)], \\ l_2 &= g\left[x_n + \tfrac{1}{2}h,\, T_\mu(\tfrac{1}{2}) + \frac{(h/2)^{m-\mu}}{(m-\mu)!} l_1\right], \\ l_3 &= g\left[x_n + \tfrac{1}{2}h,\, T_\mu(\tfrac{1}{2}) + \frac{(h/2)^{m-\mu}}{(m-\mu)!} l^{(\mu)}\right], \qquad l^{(\mu)} = \begin{cases} l_1 \ (0 \le \mu \le m-2) \\ l_2 \ (\mu = m-1), \end{cases} \\ l_4 &= g\left[x_n + h,\, T_\mu(1) + \frac{h^{m-\mu}}{(m-\mu)!} l_3\right]. \end{aligned} \tag{9.33}$$

Then

$$v_{n+1}^{(\mu)} = T_\mu(1) + \frac{h^{m-\mu}}{(m-\mu)!}\,\frac{1}{(m-\mu+1)(m-\mu+2)}\,[(m-\mu)^2 l_1$$
$$+ 2(m-\mu)(l_2+l_3) + (2-m+\mu)l_4] \qquad (\mu = 0, 1, \ldots, m-1). \tag{9.34}$$

It was shown by Zurmühl that in case v is a scalar function one has

$$v_1{}^{(\mu)} - v^{(\mu)}(x_0+h) = O(h^{p_\mu}), \qquad p_\mu = \begin{cases} m+3-\mu & (0 \le \mu \le m-2) \\ 5 & (\mu = m-1). \end{cases} \tag{9.35}$$

Thus, if $m > 2$, the function $v(x)$ is obtained with a local error of order h^{m+3}, and the order of accuracy in the sequence of derivatives decreases successively by 1 until it reaches $O(h^5)$ for the $(m-2)$nd derivative. It must be noted, however, that the accumulated error of v over a large number of steps is nevertheless of order h^4 for all m. Compare in this connection the remarks in [55].

Further Runge-Kutta methods for (9.32), involving also partial derivatives of g, are listed in [7, 17].

9.5 Error Analysis

For any one-step method we have, from (9.4) and (9.6),

$$y_{n+1} = y_n + h\varphi(x_n, y_n; h),$$
$$y(x_{n+1}) = y(x_n) + h\varphi(x_n, y(x_n); h) - r(x_n, h),$$

where $r(x,h)$ is the local truncation error. Thus, if $\epsilon_n = y_n - y(x_n)$ denotes the error vector at the nth step, we find by subtraction

$$\epsilon_{n+1} = \epsilon_n + h[\varphi(x_n, y_n; h) - \varphi(x_n, y(x_n); h)] + r(x_n, h). \tag{9.36}$$

This shows that the increase of the accumulated error is composed of two parts: the truncation error $r(x_n,h)$ and the contribution arising from the second term on the right. It is, in fact, the nature of this additional term which is decisive for the behavior of ϵ_n as n becomes large.

To see this, write

$$\varphi(x_n, y_n; h) - \varphi(x_n, y(x_n); h) = \frac{\partial\varphi}{\partial y}\,[y_n - y(x_n)] = \frac{\partial\varphi}{\partial y}\,\epsilon_n,$$

where the derivatives making up the matrix $(\partial\varphi/\partial y)$ are taken at suitable points on the line segment between y_n and $y(x_n)$. We obtain then, from (9.36),

$$\epsilon_{n+1} = \left(I + h\frac{\partial\varphi}{\partial y}\right)\epsilon_n + r(x_n, h). \tag{9.37}$$

The behavior of ϵ_n for large n, therefore, depends mainly on the matrices $I + h(\partial\varphi/\partial y)$, which are all close to the unit matrix I. If they have eigenvalues which are consistently (i.e., for all n) larger than 1, then one expects the sequence of error norms $\|\epsilon_n\|$ to increase ultimately like a geometric progression. If all eigenvalues are consistently less than 1, then the error norms will remain bounded. In view of this, the first term on the right in (9.37) is often called the "propagation error." Obviously, in case of simple quadratures, where $\partial\varphi/\partial y = 0$, there is no geometric propagation of errors.

An estimation of the propagation error for Runge-Kutta methods was already given by Runge [53]. Suppose that,* in $I \times R$,

$$\left\| \frac{\partial f(x,y)}{\partial y} \right\| \leq K,$$

and consider, for example, the Runge-Kutta method (9.21), with

$$\begin{aligned} \varphi(x,y;h) &= \sum_{s=1}^{4} \alpha_s k_s(x,y), \\ k_1(x,y) &= f(x,y), \\ k_s(x,y) &= f(x + \mu_s h, y + \lambda_s h k_{s-1}) \qquad (s > 1), \end{aligned} \tag{9.38}$$

$$\begin{aligned} \alpha_1 &= \tfrac{1}{2}\alpha_2 = \tfrac{1}{2}\alpha_3 = \alpha_4 = \tfrac{1}{6}, \\ \mu_2 &= \mu_3 = \tfrac{1}{2}\mu_4 = \tfrac{1}{2}, \\ \lambda_2 &= \lambda_3 = \tfrac{1}{2}\lambda_4 = \tfrac{1}{2}. \end{aligned} \tag{9.39}$$

Then, for any two points (x,y), $(x,\bar{y})$ in $I \times R$, we have

$$\|k_1(x,y) - k_1(x,\bar{y})\| = \left\| \frac{\partial f}{\partial y}(y - \bar{y}) \right\| \leq K\,\|y - \bar{y}\|,$$

$$\begin{aligned} \|k_s(x,y) - k_s(x,\bar{y})\| &= \left\| \frac{\partial f}{\partial y}\{y - \bar{y} + \lambda_s h[k_{s-1}(x,y) - k_{s-1}(x,\bar{y})]\} \right\| \\ &\leq K[\|y - \bar{y}\| + \lambda_s h\,\|k_{s-1}(x,y) - k_{s-1}(x,\bar{y})\|]. \end{aligned}$$

From this we obtain successively

$$\begin{aligned} \|k_2(x,y) - k_2(x,\bar{y})\| &\leq K(1 + \lambda_2 hK)\,\|y - \bar{y}\|, \\ \|k_3(x,y) - k_3(x,\bar{y})\| &\leq K(1 + \lambda_3 hK + \lambda_3\lambda_2 h^2K^2)\,\|y - \bar{y}\|, \\ \|k_4(x,y) - k_4(x,\bar{y})\| &\leq K(1 + \lambda_4 hK + \lambda_4\lambda_3 h^2K^2 + \lambda_4\lambda_3\lambda_2 h^3K^3)\,\|y - \bar{y}\|. \end{aligned}$$

* For a different assumption in this connection (relative to scalar differential equations) see [11].

Therefore, by (9.38) and (9.39),

$$\begin{aligned}\|\varphi(x,y;h) - \varphi(x,\bar{y};h)\| &\le K[\alpha_1 + \alpha_2(1 + \lambda_2 hK) \\ &+ \alpha_3(1 + \lambda_3 hK + \lambda_3\lambda_2 h^2K^2) + \alpha_4(1 + \lambda_4 hK + \lambda_4\lambda_3 h^2 K^2 \\ &+ \lambda_4\lambda_3\lambda_2 h^3K^3)]\,\|y - \bar{y}\| = K(1 + \tfrac{1}{2}hK + \tfrac{1}{6}h^2K^2 + \tfrac{1}{24}h^3K^3)\,\|y - \bar{y}\|.\end{aligned}$$

Letting now $x = x_n, y = y_n, \bar{y} = y(x_n)$, we find for the propagation term in (9.36) the following estimation

$$\begin{aligned}&\|\epsilon_n + h\{\varphi(x_n,y_n;h) - \varphi(x_n,y(x_n);h)\}\| \\ &\qquad \le \left(1 + hK + \frac{(hK)^2}{2!} + \frac{(hK)^3}{3!} + \frac{(hK)^4}{4!}\right)\|\epsilon_n\|. \qquad (9.40)\end{aligned}$$

The estimation of the truncation error is considerably more laborious. A first rigorous estimate for the Runge-Kutta method (9.38), (9.39) is stated in [4] and proved in [5] by Bieberbach. We give it here for the case of a scalar first-order differential equation. In $I \times R$, let

$$f \in C^4, \qquad |f| \le N, \qquad \left|\frac{\partial^l f}{\partial x^i\, \partial y^k}\right| \le \frac{M}{N^{k-1}} \qquad (0 < l \le 4, 0 \le k \le 4). \qquad (9.41)$$

Then, for $x_n < X$,

$$|r(x_n,h)| \le MN(3.7 + 5.4M + 1.3M^2 + .017M^3)h^5. \qquad (9.42)$$

Similar bounds are derived in [5] and [66] also for arbitrary systems of differential equations. Analogous bounds for the truncation error of Zurmühl's method can be found in [18] (see also [8]).

Under the slightly different assumptions

$$\left|\frac{\partial^l f}{\partial x^i\, \partial y^k}\right| \le \frac{M^l}{N^{k-1}} \qquad (0 \le l \le 4) \qquad (9.43)$$

(which contain, for $l = 0$, the assumption $|f| \le N$) Lotkin [35] derives the "asymptotic" estimate

$$r(x_n,h) = \rho h^5 + O(h^6), \qquad |\rho| \le \tfrac{73}{720} NM^4. \qquad (9.44)$$

The constant M in (9.43) may be obtained by forming

$$\max_{I \times R}\left|\frac{\partial^l f}{\partial x^{l-k}\, \partial y^k}\right| = K_{lk}, \qquad \max_{0 \le k \le l} N^{k-1}K_{lk} = L_l, \qquad \max_{1 \le l \le 4} L_l^{1/l} = M$$

in this order.

Estimates which are based on a priori bounds for $y^{(5)}(x)$, $(d/dx)f_y[x, y(x)]$, $f_{yy}[x,y(x)]$ are given in [1], and further estimates in [7, 10].

The estimation (9.40) of the propagation error can now be combined with any estimation

$$\|r(x_n,h)\| \leq Th^5$$

of the truncation error to give a bound for the total accumulated error in Runge-Kutta's method. In fact, setting

$$1 + hK + \frac{(hK)^2}{2!} + \frac{(hK)^3}{3!} + \frac{(hK)^4}{4!} = 1 + hP,$$

we obtain, from (9.36) and (9.40),

$$\|\epsilon_{n+1}\| \leq (1 + hP)\,\|\epsilon_n\| + Th^5 \qquad (n = 0, 1, 2, \ldots),$$

and therefore, if $\epsilon_0 = 0$ and $P > 0$,

$$\|\epsilon_n\| \leq \frac{Th^4}{P}\,[(1 + hP)^n - 1].$$

In the light of example (9.3) it is not surprising that estimations of this kind, being necessarily satisfied in every instance, may lead to rather conservative bounds. For example, if we apply (9.41), (9.42) to the equation

$$y' = f(x), \tag{9.45}$$

that is, to a simple problem of quadrature, we obtain

$$|r(x_n,h)| \leq 3.7Hh^5(1 + \cdots), \qquad H = \max_{1 \leq l \leq 4} |f^{(l)}|, \tag{9.46}$$

where dots indicate additional positive terms. On the other hand, the well-known remainder term for Simpson's rule, to which Runge-Kutta's method reduces in this case, leads to the estimate

$$|r(x_n,h)| \leq \frac{H^*h^5}{2880}, \qquad H^* = \max |f^{(4)}|. \tag{9.47}$$

Comparison between (9.46) and (9.47) shows that, for quadratures, Bieberbach's bound is too large by a factor of at least 10^4. Somewhat better is the bound which derives from (9.44) [about 300 times too large in the case of (9.45)], but it has the drawback of being justified only for h sufficiently small.

In view of these difficulties, present-day efforts tend to appraise the error more realistically by stochastic methods [25, 50]. In practice, one makes use of various asymptotic devices to estimate the accuracy, notably Richardson's "deferred approach" to the limit [51] (see also [16, 21]).

DIFFERENCE METHODS

9.6 Linear Multistep Methods

The methods so far discussed replace the first-order system of differential equations (9.1) by a first-order system of difference equations (9.4). The local error committed thereby could be made as small as $O(h^5)$, but not smaller without considerable additional effort. We now replace (9.1) by difference equations of higher order, which will permit us to decrease the local error to any desired order of magnitude.

More specifically, we consider *linear multistep methods*, that is, methods which define y_{n+1} by a linear combination of vectors y_{n-s+1}, hf_{n+1}, hf_{n-s+1}, $s = 1, 2, \ldots, k$, where

$$f_n = f(x_n, y_n) \qquad (n = 0, 1, 2, \ldots). \tag{9.48}$$

We write these methods in the form

$$y_{n+1} + \alpha_1 y_n + \cdots + \alpha_k y_{n-k+1} = h(\beta_0 f_{n+1} + \beta_1 f_n + \cdots + \beta_k f_{n-k+1}), \tag{9.49}$$

where it is assumed that α_s, β_s are real constants, $k > 1$, $|\alpha_k| + |\beta_k| > 0$, and $h = x_{n+1} - x_n$ is independent of n. Once k initial vectors $y_0, y_1, \ldots, y_{k-1}$ are known, the relation (9.49) can be used to obtain successively all desired approximations y_n for $n \geq k$. The k initial vectors must be found independently by some other method (e.g., by a Runge-Kutta method).

It is customary to call the formula (9.49) *open* if $\beta_0 = 0$ and *closed* if $\beta_0 \neq 0$. Open formulas define the "new" approximation y_{n+1} explicitly in terms of "old" approximations, whereas closed formulas contain y_{n+1} implicitly as argument in f_{n+1}. In the latter case the relation (9.49) represents a system of m, in general, nonlinear equations for the m components of y_{n+1}. The existence and uniqueness of a solution of such a system, and a practical method of solving it, are discussed in Sec. 9.8.

We associate with the "k-step" method (9.49) the linear functional L defined by

$$Lw(t) = \sum_{s=0}^{k} [\alpha_s w(k - s) - \beta_s w'(k - s)], \qquad \alpha_0 = 1. \tag{9.50}$$

Here $w(t)$ is considered to be a scalar function differentiable in $[0,k]$. We call k the *index* of L and say that the functional L is of *order* p if $L(t^r) = 0$ for $r = 0, 1, \ldots, p$ but $L(t^{p+1}) \neq 0$.*

* In place of the powers $\{t^r\}$ one may consider other systems of functions $\{\omega_r(t)\}$ and define analogously the order of L with respect to $\{\omega_r\}$. For the case $\omega_r(t) = \exp(\lambda_r t)$ with suitable constants λ_r, see [6].

If $w \in C^{p+1}$ in $[0,k]$, then, by Taylor's theorem,

$$w(t) = \sum_{r=0}^{p} \frac{w^{(r)}(0)}{r!} t^r + \frac{1}{p!} \int_0^k (t-u)^p{}_+ w^{(p+1)}(u)\, du$$

where $\qquad (t-u)_+ = \max\,(0, t-u).$

Every linear functional L of order p can therefore be represented in the form

$$Lw = \frac{1}{p!} \int_0^k \lambda_p(u) w^{(p+1)}(u)\, du, \qquad \lambda_p(u) = L[(t-u)^p{}_+], \qquad (9.51)$$

in which all characteristics of L are collected in the first factor of the integrand and all those of w in the second factor. If $\lambda_p(u)$ does not change sign in $[0,k]$, then (9.51) can be simplified to

$$Lw = l_{p+1} w^{(p+1)}(\tau), \qquad l_{p+1} = \frac{L(t^{p+1})}{(p+1)!}, \qquad 0 < \tau < k. \qquad (9.51')$$

In fact, by the mean-value theorem,

$$Lw = \frac{w^{(p+1)}(\tau)}{p!} \int_0^k \lambda_p(u)\, du,$$

and by (9.51), with $w(t) = t^{p+1}/(p+1)!$,

$$\int_0^k \lambda_p(u)\, du = \frac{L(t^{p+1})}{p+1}. \qquad (9.52)$$

The expressions in (9.51) and (9.51′) are referred to as *remainder terms* of the functional L.

In analogy with (9.6) we define for the method (9.49) the local truncation error at the point x to be the vector

$$r(x,h) = y^*(x,h) - y(x+h),$$

where y^* is the solution of

$$y^* + \sum_{s=1}^{k} \alpha_s y(x - sh + h) - h\left[\beta_0 f(x+h, y^*) + \sum_{s=1}^{k} \beta_s y'(x - sh + h)\right] = 0.$$

If the functional L in (9.50) has order p, then our method (9.49) is of order p in the sense of (9.7). More precisely, it can be shown that if $f \in C^p$, then

$$\|r(x,h)\| = |l_{p+1}|\ \|y^{(p+1)}(x)\|\ h^{p+1} + o(h^{p+1}).$$

Multistep methods of a more general form than (9.49) have also been considered. Some of them make use of certain "advanced" points (x_{n+l}, y_{n+l}), $l > 1$, others of partial derivatives of f. For a study of these we refer to [9, 15, 33, 44].

9.7 Multistep Methods of Maximal Order

We turn now to the question of determining suitable functionals (9.50). In order to obtain a functional L of order p, we must have

$$L(t^r) = \sum_{s=0}^{k} \alpha_s (k-s)^r - r \sum_{s=0}^{k} \beta_s (k-s)^{r-1} = 0 \quad (r = 0, 1, \ldots, p). \tag{9.53}$$

Since $\alpha_0 = 1$, these relations represent an inhomogeneous system of $(p + 1)$ linear equations in $(2k + 1)$ unknowns α_s, β_s. If $p < 2k$, such a system has always infinitely many solutions, so that there is an infinite number of functionals L with index k and order $p < 2k$. On the other hand, it can be shown that (9.53) has a unique solution if $p = 2k$ and no solution if $p > 2k$. The corresponding functional L of maximum order can be constructed in closed form by using Hermite's interpolation formula (details are given in [14], numerical values in [56]).

The resulting k-step methods of order $2k$ are mainly of theoretic interest, in spite of their high local accuracy. It turns out that the corresponding recurrence (9.49) is very sensitive to small disturbances, which are quickly amplified during repeated application of (9.49). This phenomenon, known as numerical instability, is examined in more detail in Sec. 9.10. Here, we mention only that the stability properties of (9.49) depend on the location of the zeros of the polynomial

$$a(z) = z^k + \alpha_1 z^{k-1} + \cdots + \alpha_k, \tag{9.54}$$

which is called the *generating polynomial* of the functional L. In fact, a necessary condition for stability is that all zeros of $a(z)$ be located within or on the unit circle and that all zeros on the unit circle be simple [14, 59]:

$$\text{if} \qquad a(\zeta) = 0, \qquad \text{then} \qquad \begin{cases} \text{either } |\zeta| < 1 \\ \text{or } |\zeta| = 1 \quad \text{and} \quad a'(\zeta) \neq 0. \end{cases} \tag{9.55}$$

Following Dahlquist [14], we call a functional L (and also the corresponding multistep method) *stable* if the generating polynomial $a(z)$ of L satisfies condition (9.55).

From the practical point of view, therefore, the interesting question is to maximize the order of a functional L among all stable functionals with given index. We can assume, henceforth, that L has order $p \geq 0$, so that certainly

$$a(1) = 1 + \alpha_1 + \alpha_2 + \cdots + \alpha_k = 0. \tag{9.56}$$

We note then, first of all, the following theorem.

Theorem 9.1. *If for a given generating polynomial* (9.54) *there exists a corresponding functional L with index k and order $p \geq k + 1$, then it is unique.*

The proof follows readily from (9.53), in which the α_s are to be considered as prescribed and the β_s as unknowns.

A functional L of the type indicated in Theorem 9.1 always exists. Indeed, let us write (9.50) equivalently in the form

$$\begin{aligned} Lw(t) &= \sum_{s=0}^{k} [\alpha_s w(k-s) - \gamma_{ks}\nabla^s w'(k)] \\ &= a(E)E^{-k}w(k) - \sum_{s=0}^{k}\gamma_{ks}\nabla^s w'(k), \end{aligned} \tag{9.57}$$

where E, ∇ denote the displacement and the backward difference operator, respectively. By means of the formal calculus of operators, we transform (9.57) into

$$Lw(t) = \left\{-\frac{a[1/(1-\nabla)](1-\nabla)^k}{\ln(1-\nabla)} - \sum_{s=0}^{k}\gamma_{ks}\nabla^s\right\}w'(k),$$

which certainly holds whenever $w(t)$ is a polynomial. Let

$$-\frac{a[1/(1-z)](1-z)^k}{\ln(1-z)} = \sum_{s=0}^{\infty} c_{ks}z^s, \tag{9.58}$$

and define

$$\gamma_{ks} = c_{ks} \qquad (s = 0, 1, \ldots, k). \tag{9.59}$$

Then

$$Lw(t) = \left(\sum_{s=k+1}^{\infty} c_{ks}\nabla^s\right)w'(k). \tag{9.60}$$

Hence, $Lw = 0$ if w is a polynomial of degree $\leq k+1$; that is, L is of order $p \geq k+1$.

If L is stable, it can be shown, moreover, that $c_{k,k+1} \neq 0$ unless k is even and

$$\alpha_{k-s} = -\alpha_s \qquad (s = 0, 1, \ldots, k;\ k \text{ even}), \tag{9.61}$$

in which case $c_{k,k+1} = 0$, $c_{k,k+2} \neq 0$. In view of (9.60) and Theorem 9.1, this implies the following result.

Theorem 9.2 (*Dahlquist* [14]). *The maximal order among all stable functionals L with index k is equal to $k+1$ if k is odd and is equal to $k+2$ if k is even.*

It is characteristic of a stable functional of even index and maximal order that its generating polynomial $a(z)$ has the zeros $z = 1$ and $z = -1$ and that all other zeros, if any, are on the unit circle arranged in conjugate pairs. In fact, (9.61) is equivalent to $z^k a(1/z) + a(z) \equiv 0$; hence $a(1) = a(-1) = 0$, and $a(\zeta) = 0$ implies $a(1/\zeta) = 0$. Since no zero is allowed to fall outside the unit circle, we have $|\zeta| = 1$ for each zero ζ, and then $1/\zeta = \bar{\zeta}$ is a zero whenever ζ is.

Conversely, if $a(z)$ has the zeros

$$\zeta_1 = 1, \qquad \zeta_2 = -1, \qquad \zeta_{2r-1} = \bar{\zeta}_{2r} = e^{i\theta_r}$$
$$(r = 2, \ldots, k/2; 0 < \theta_r < \pi),$$

then
$$a(z) = (z^2 - 1) \prod_{r=2}^{k/2} [z^2 - (2 \cos \theta_r) z + 1],$$

so that

$$z^k a(1/z) + a(z) = (1 - z^2) \prod_{r=2}^{k/2} [1 - (2 \cos \theta_r) z + z^2]$$
$$+ (z^2 - 1) \prod_{r=2}^{k/2} [z^2 - (2 \cos \theta_r) z + 1] \equiv 0.$$

The gain in order, if k is even, can be explained by the existence of an expansion of L in central differences,

$$Lw = \sum_{s=0}^{k} \alpha_s w(k - s) - \sum_{s=0}^{k/2} \tau_{ks}\, \delta^{2s} w'(\tfrac{1}{2}k) \qquad (k \text{ even}; \alpha_{k-s} = -\alpha_s), \quad (9.62)$$

where the coefficients τ_{ks} are found from*

$$\frac{a(u)}{u^{(k/2)} \ln u} = \sum_{s=0}^{\infty} \tau_{ks} z^{2s}, \qquad u = 1 + \tfrac{1}{2}z^2 + z\sqrt{1 + \tfrac{1}{4}z^2}. \quad (9.63)$$

Formula (9.62), for example, contains for $k = 2$, $a(z) = z^2 - 1$, Simpson's formula

$$w(2) - w(0) - \tfrac{1}{3}[w'(2) + 4w'(1) + w'(0)] = -\tfrac{1}{90} w^{(5)}(\tau), \quad (9.64)$$

and for $k = 4$, $a(z) = z^4 + z^3 - z - 1$ a formula due to Dahlquist [14]:

$$w(4) + w(3) - w(1) - w(0) - 3[w'(3) + w'(1)$$
$$+ \tfrac{1}{10}\, \delta^4 w'(2)] = -\tfrac{1}{140} w^{(7)}(\tau). \quad (9.65)$$

The remainder terms are obtained from (9.51′), which is applicable in these cases, as can be verified.

Stable functionals L of maximal order are always closed [14]. Open functionals of index k and order k can be constructed by a formula analogous to (9.57):

$$Lw = \sum_{s=0}^{k} \alpha_s w(k - s) - \sum_{s=0}^{k-1} \sigma_{ks} \nabla^s w'(k - 1), \quad (9.66)$$

where
$$\frac{(1 - z)^{k-1} a[1/(1 - z)]}{-\ln (1 - z)} = \sum_{s=0}^{\infty} \sigma_{ks} z^s. \quad (9.67)$$

* Note that the left-hand side in (9.63) is an even function of z, since $u(-z) = 1/u(z)$ and

$$\frac{a(1/u)}{u^{-(k/2)} \ln(1/u)} = \frac{-u^k a(1/u)}{u^{(k/2)} \ln u} = \frac{a(u)}{u^{(k/2)} \ln u}$$

by virtue of (9.61).

For example, if $k = 4$ and $a(z) = z^4 - 1$, we obtain from (9.66) and (9.51′),

$$w(4) - w(0) - \tfrac{4}{3}[2w'(3) - w'(2) + 2w'(1)] = \tfrac{14}{45}w^{(5)}(\tau). \qquad (9.68)$$

In view of the representations (9.57) and (9.66), multistep methods are often called *finite-difference methods.* Extensive lists or various examples of difference methods can be found in [22, 48, 65, 69]. The classical difference methods are based on the functional L with $a(z) = z^k - z^{k-1}$ and are discussed in more detail in Sec. 9.9.

9.8 Predictor-Corrector Schemes

We return now to the question of solving the equation

$$\begin{aligned} y_{n+1} + \alpha_1 y_n + \cdots + \alpha_k y_{n-k+1} = h[\beta_0 f(x_{n+1}, y_{n+1}) \\ + \beta_1 f_n + \cdots + \beta_k f_{n-k+1}] \qquad (\beta_0 \neq 0) \qquad (9.69) \end{aligned}$$

for the vector y_{n+1}, assuming that the vectors $y_n, y_{n-1}, \ldots ; f_n, f_{n-1}, \ldots$ are known. We shall show that, for h sufficiently small, (9.69) has a unique solution, which can be obtained by a method of successive approximations.

Theorem 9.3. *If*

$$|\beta_0|\, Kh < 1, \qquad (9.70)$$

$K > 0$ *being a constant satisfying* (9.2), *then Eq.* (9.69) *has a unique solution* y_{n+1}. *If we define the iteration*

$$\begin{aligned} y_{n+1}^{[\nu+1]} + \alpha_1 y_n + \cdots + \alpha_k y_{n-k+1} = h[\beta_0 f(x_{n+1}, y_{n+1}^{[\nu]}) \\ + \beta_1 f_n + \cdots + \beta_k f_{n-k+1}] \qquad (\nu = 0, 1, 2, \ldots), \\ (9.71) \end{aligned}$$

then $y_{n+1}^{[\nu]} \to y_{n+1}$ *as* $\nu \to \infty$ *for any initial vector* $y_{n+1}^{[0]}$. *Moreover,*

$$\|y_{n+1}^{[\nu]} - y_{n+1}\| \leq \frac{(|\beta_0|\, Kh)^\nu}{1 - |\beta_0|\, Kh} \|y_{n+1}^{[1]} - y_{n+1}^{[0]}\|. \qquad (9.72)$$

(It is tacitly assumed that $y_{n+1}^{[0]} \in R$ and that, with any $y_{n+1}^{[\nu]} \in R$, also $y_{n+1}^{[\nu+1]} \in R$, where R is a closed set in euclidean space E_m.)

Proof. Define the operator

$$Tu = h\beta_0 f(x_{n+1}, u) - \sum_{s=1}^{k} (\alpha_s y_{n-s+1} - h\beta_s f_{n-s+1}),$$

which maps R into itself, and let $\mu = |\beta_0|\, Kh$; by assumption, $\mu < 1$. Then, for any two vectors u and v of R, we have, by (9.2),

$$\|Tu - Tv\| = \|h\beta_0[f(x_{n+1}, u) - f(x_{n+1}, v)]\| \leq \mu\, \|u - v\|,$$

so that T is a contraction operator. For such operators it is well known that the iteration $y_{n+1}^{[\nu+1]} = Ty_{n+1}^{[\nu]}$, that is, (9.71), converges to a unique solution y_{n+1} of (9.69) for which (9.72) holds.

Practical use of Theorem 9.3 is made in the *predictor-corrector methods.* These consist in "predicting" an initial approximation $y_{n+1}^{[0]}$ to y_{n+1} by means of an open multistep formula and "correcting" it then successively by means of the iteration (9.71), based on a closed multistep formula. If L_1 and L_2 denote the functionals corresponding to the predictor and corrector formulas, respectively, then they are usually chosen in such a way that L_1 and L_2 have the same order p but that L_2 has a smaller remainder term than L_1; that is,

$$\text{order } L_1 = \text{order } L_2 = p, \qquad |L_1(t^{p+1})| > |L_2(t^{p+1})|. \tag{9.73}$$

As an example of such a predictor-corrector scheme, we mention *Milne's method* [40], which uses the functional in (9.68) for L_1 and the functional in (9.64) for L_2:

$$\begin{aligned} y_{n+1}^{[0]} &= y_{n-3} + \tfrac{4}{3}h(2f_n - f_{n-1} + 2f_{n-2}), \\ y_{n+1}^{[\nu+1]} &= y_{n-1} + \tfrac{1}{3}h[f(x_{n+1}, y_{n+1}^{[\nu]}) + 4f_n + f_{n-1}] \quad (\nu = 0, 1, 2, \ldots). \end{aligned} \tag{9.74}$$

Here, $p = 4$ and $L_1[t^5] = {}^{112}\!/_3$, $L_2[t^5] = -{}^4\!/_3$. Other pairs of predictor and corrector formulas are given in the next section. See also [24, 57], for alternative formulas, and [30], for a procedure of changing the step length in Milne's method.

The number of iterations required to obtain y_{n+1} with a given accuracy is seen from (9.72) to depend, in part, on the size of Kh. Since this quantity should be kept considerably less than unity, not more than two or three iterations should turn out to be necessary. If more are needed, there is evidence that the step length h is too large.

We note also that the relations

$$y_{n+1}^{[\nu+1]} = y_{n+1}^{[\nu]} + h\beta_0[f(x_{n+1}, y_{n+1}^{[\nu]}) - f(x_{n+1}, y_{n+1}^{[\nu-1]})] \qquad (\nu \geq 1) \tag{9.75}$$

can be substituted for (9.71) to compute the second iterate and all higher iterates.

The predictor-corrector scheme described above for systems of first-order differential equations applies also to a single differential equation of order $m > 1$,

$$v^{(m)} = g(x, v, v', \ldots, v^{(m-1)}), \tag{9.76}$$

if (9.76) is transformed in the usual manner into a system of m first-order equations. One would predict then the unknown function v and all its first $(m - 1)$ derivatives, before any corrections are applied. In practice, however, one predicts only once, namely, the value $v_{n+1}^{(m-1)}$, by a predictor formula P, say, and from then on uses a corrector formula C to obtain $v_{n+1}^{(m-2)}$, $v_{n+1}^{(m-3)}$, . . . , v_{n+1} in this order. The values obtained are then inserted into the differential equation (9.76), giving $v_{n+1}^{(m)}$, after which $v_{n+1}^{(m-1)}$ can be recalculated by the corrector formula C. If this new value differs from that obtained previously by P, the cycle is repeated.

9.9 Adams' Method

Difference methods which correspond to the generating polynomial

$$a(z) = z^k - z^{k-1} \qquad (k > 0)$$

are always stable in the sense of Sec. 9.7 since $a(z)$ has the simple zero $z = 1$ and all remaining zeros are zero if $k > 1$. The corresponding open and closed methods of index k and maximal order are called *Adams' extrapolation method* and *Adams' interpolation method*, respectively. The latter is of order $k + 1$, by virtue of Theorem 9.2, whereas the former is of order k, as follows from (9.66) and Theorem 9.1.

The functional L_1 for the extrapolation method is obtained from (9.66) and (9.67),

$$L_1 w = w(k) - w(k-1) - \sum_{s=0}^{k-1} \sigma_s \nabla^s w'(k-1), \tag{9.77}$$

where the constants σ_s are determined by the expansion

$$\frac{z}{-(1-z)\ln(1-z)} = \sum_{s=0}^{\infty} \sigma_s z^s. \tag{9.78}$$

Here, the coefficients no longer depend on k, which has the practical advantage that the index and order of L_1 can be simultaneously increased by simply adding more terms to the sum in (9.77).

Adams' extrapolation method of order k can thus be written in the form

$$y_{n+1} = y_n + h \sum_{s=0}^{k-1} \sigma_s \nabla^s f_n. \tag{9.79}$$

The coefficients σ_s allow the following representation as a definite integral:

$$\sigma_s = \int_0^1 \binom{t+s-1}{s} dt \qquad (s = 0, 1, 2, \ldots). \tag{9.80}$$

In fact, since L_1 is of order k, we have, for $k \geq 1$,

$$0 = L_1\left[\int_0^t \binom{u}{k-1} du\right] = \int_0^k \binom{t}{k-1} dt - \int_0^{k-1} \binom{t}{k-1} dt$$

$$- \sum_{s=0}^{k-1} \sigma_s \binom{k-1-s}{k-1-s} = \int_{k-1}^k \binom{t}{k-1} dt - \sum_{s=0}^{k-1} \sigma_s,$$

where the well-known relation

$$\nabla^s \binom{t}{r} = \binom{t-s}{r-s}$$

was used. Hence

$$\sum_{s=0}^{k-1} \sigma_s = \int_{k-1}^{k} \binom{t}{k-1}\,dt,$$

and since k is arbitrary,

$$\sigma_k = \sum_{s=0}^{k} \sigma_s - \sum_{s=0}^{k-1} \sigma_s = \int_{k}^{k+1} \binom{t}{k}\,dt - \int_{k-1}^{k} \binom{t}{k-1}\,dt$$

$$= \int_{k-1}^{k} \left[\binom{t+1}{k} - \binom{t}{k-1}\right] dt = \int_{k-1}^{k} \binom{t}{k}\,dt = \int_{0}^{1} \binom{t+k-1}{k}\,dt,$$

which proves (9.80) for $s \geq 1$. If $s = 0$, then $\sigma_0 = 1$ follows directly from (9.78).

As to the remainder term of L_1, it can be shown that

$$L_1 w = \sigma_k w^{(k+1)}(\tau), \qquad 0 < \tau < k. \tag{9.81}$$

For the interpolation method we obtain the corresponding functional L_2 from (9.57) to (9.59),

$$L_2 w = w(k) - w(k-1) - \sum_{s=0}^{k} \gamma_s \nabla^s w'(k), \tag{9.82}$$

where

$$\frac{z}{-\ln(1-z)} = \sum_{s=0}^{\infty} \gamma_s z^s. \tag{9.83}$$

By an argument similar to that following (9.80) one finds

$$\gamma_s = \int_0^1 \binom{t+s-2}{s}\,dt \qquad (s = 0, 1, 2, \ldots). \tag{9.84}$$

Also, the remainder term of L_2 can be written in the form

$$L_2 w = \gamma_{k+1} w^{(k+2)}(\tau), \qquad 0 < \tau < k. \tag{9.85}$$

If we increase by 1 the index (and order) of the functional L_1 in (9.77), we can combine it with L_2 to form a pair of predictor and corrector formulas (both having order $k + 1$). This leads to the following method:

$$\begin{aligned} y_{n+1}^{[0]} &= y_n + h \sum_{s=0}^{k} \sigma_s \nabla^s f_n, \\ y_{n+1}^{[\nu+1]} &= y_n + h \sum_{s=0}^{k} \gamma_s \nabla^s f_{n+1}^{[\nu]} \qquad (\nu = 0, 1, 2, \ldots), \end{aligned} \tag{9.86}$$

where $f_{n+1}^{[\nu]} = f(x_{n+1}, y_{n+1}^{[\nu]})$. It is easily verified from (9.80) and (9.84) that

$$|\gamma_{s+1}| < \sigma_{s+1} \qquad (s \geq 1),$$

so that the criterion (9.73) for a proper choice of predictor and corrector formulas is fulfilled.

Adams [3] derives both formulas (9.86) by integration of appropriate interpolation polynomials. He does not use the first formula but predicts $y_{n+1}^{[0]}$ by extrapolation of the last difference retained and successive "advancing" of the difference table.

As indicated in (9.75), the calculation for the iteration in (9.86) can be shortened by using

$\sigma_k \quad \sum_{s=0}^{k} \gamma_s$

$$y_{n+1}^{[\nu+1]} = y_{n+1}^{[\nu]} + h\gamma_0(f_{n+1}^{[\nu]} - f_{n+1}^{[\nu-1]}) \qquad \text{for } \nu \geq 1.$$

In the case where $y_{n+1}^{[0]}$ is predicted according to (9.86), also the first iterate can be obtained directly by means of the formula [58]:

$0 \quad \sigma_k$

$$y_{n+1}^{[1]} = y_{n+1}^{[0]} + h\gamma_k \nabla^{k+1} f_{n+1}^{[0]}. \tag{9.87}$$

In order to use (9.87), however, one has to carry along an extra column of the $(k + 1)$st differences.

A detailed error analysis for Adams' method was first given* by von Mises [41] and since then has been the subject of numerous investigations (e.g., [2, 37, 39, 43, 61 to 64, 67, 68]). We refer the reader to these original papers or to the treatments in the texts by Hildebrand and Collatz.

9.10 Stability of Difference Methods

In numerical work it is usually impossible to carry out all required calculations with unlimited precision. One is forced to round numbers to finitely many figures and, if infinite processes are involved, to reduce these to finite ones. Thus, in the case of multistep method (9.49), the actual results z_n one obtains do not satisfy (9.49), that is,

$$\sum_{s=0}^{k} \alpha_s y_{n+1-s} = h \sum_{s=0}^{k} \beta_s f(x_{n+1-s}, y_{n+1-s}), \tag{9.88}$$

but rather

$$\sum_{s=0}^{k} \alpha_s z_{n+1-s} = h \sum_{s=0}^{k} \beta_s f(x_{n+1-s}, z_{n+1-s}) + r_n, \tag{9.89}$$

where the vectors r_n are "small" in norm. If the initial vectors $z_\kappa (\kappa = 0, 1, \ldots, k - 1)$ differ from y_κ only slightly and if all r_n are sufficiently small, one expects that the vectors z_n will also differ only slightly from y_n for all n. Regarding (9.89) as a perturbation of (9.88), one may then say, loosely speaking, that (9.88) is stable if it is indeed insensitive to small perturbances r_n.

In the following we consider a somewhat stronger notion of stability, which is uniform with respect to h in the sense that, if $z_\kappa - y_\kappa$ and r_n are suitably restricted, stability takes place no matter how small h. Since

* Actually Tamarkin [35'] has already a convergence proof for the Adams-Bashforth method.

(9.88) will generally be used for $n = k - 1, k, \ldots, N - 1$, with N such that $x_0 + Nh < X$, this implies that we shall have stability for arbitrarily large N.

We say that the perturbations in (9.89) are of class $C(\delta)$ if, for some $h_0 > 0$,

$$\sum_{\kappa=0}^{k-1} \|z_\kappa - y_\kappa\| + \sum_{n=k-1}^{N-1} \|r_n\| \leq \delta \tag{9.90}$$

uniformly for all $h \leq h_0$ and all N with $x_0 + Nh < X$. For example, perturbations such that $\|z_\kappa - y_\kappa\| \leq \delta/2k$, $\|r_n\| \leq \delta h/[2(X - x_0)]$ are of class $C(\delta)$.

The multistep method (9.88) is then called stable with respect to a class Γ of functions $f(x,y)$ if for each $f \in \Gamma$ and for any $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that the errors produced by perturbations of class $C(\delta)$ satisfy $\|z_n - y_n\| \leq \epsilon\,(n = k, k + 1, \ldots, N)$ uniformly for all $h \leq h_0$ and all N with $x_0 + Nh < X$.†

In the following we consider the class Γ^* of functions f continuous on $I \times R$ and satisfying (9.2).

Theorem 9.4.‡ *A multistep method* (9.88) *is stable relative to the class* Γ^* *if and only if condition* (9.55) *is fulfilled.*

Before proving Theorem 9.4, we establish the following useful lemma on linear nonhomogeneous difference equations with constant coefficients.

Lemma 9.1. *Let* e_n *satisfy the difference equation*

$$\sum_{s=0}^{k} \alpha_s e_{n+1-s} = g_n \qquad (n \geq k - 1, \alpha_0 = 1) \tag{9.91}$$

and denote by $H_{n\kappa}(\kappa = 0, 1, \ldots, k - 1)$ *the solutions of the corresponding homogeneous equation* (*with* $g_n = 0$) *satisfying the initial conditions*

$$H_{n\kappa} = \delta_{n\kappa} \qquad (n = 0, 1, \ldots, k - 1). \tag{9.92}$$

($\delta_{n\kappa}$ *is the Kronecker symbol.*) *Then*

$$e_n = \sum_{\kappa=0}^{k-1} H_{n\kappa} e_\kappa + \sum_{\nu=k-1}^{n-1} H_{n+k-\nu-2,k-1} g_\nu \qquad (n \geq k). \tag{9.93}$$

Proof of Lemma 9.1. It is clear from (9.91) that e_n must be of the form

$$e_n = \sum_{\kappa=0}^{k-1} K_{n\kappa} e_\kappa + \sum_{\nu=k-1}^{n-1} L_{n\nu} g_\nu \qquad (n \geq k). \tag{9.94}$$

† For other definitions of stability, see, for example, [34, 49].

‡ Theorem 9.4 is essentially due to Dahlquist [14]. We follow here, with minor deviations, the exposition given in [27].

By considering the special case where all $g_\nu = 0$ and $e_\kappa = \delta_{\kappa\lambda}$ for some fixed λ, $0 \leq \lambda < k$, we infer from (9.94) that $e_n = K_{n\lambda}$ and from the uniqueness of the corresponding solution that

$$K_{n\lambda} = H_{n\lambda} \qquad (n \geq k;\ \lambda = 0, 1, \ldots, k-1).$$

Considering next the case where $g_\nu = \delta_{\nu\mu}$ for some fixed $\mu \geq k-1$ and $e_\kappa = 0$ for $\kappa = 0, 1, \ldots, k-1$, we conclude from (9.94) that $L_{n\nu}$ is a solution of

$$\sum_{s=0}^{k} \alpha_s L_{n+1-s} = \delta_{n\nu} \qquad (n \geq k-1) \tag{9.95}$$

satisfying the initial conditions

$$L_\kappa = 0 \qquad (\kappa = 0, 1, \ldots, k-1). \tag{9.96}$$

For all $n < \nu$, (9.95) is homogeneous, so that $L_{n\nu} = 0$ $(n \leq \nu)$, because of (9.96). Setting $n = \nu$ in (9.95), we find $L_{\nu+1,\nu} = 1$. For $n > \nu$, (9.95) is again homogeneous. Hence, $L_{n\nu}$ is a solution, for $n > \nu$, of the homogeneous difference equation associated with (9.91), satisfying

$$L_{\nu-k+2,\nu} = L_{\nu-k+3,\nu} = \cdots = L_{\nu\nu} = 0, \qquad L_{\nu+1,\nu} = 1.$$

Since $H_{n+k-\nu-2,k-1}$ is another such solution, it follows that

$$L_{n\nu} = H_{n+k-\nu-2,k-1}.$$

Lemma 9.1 is proved.

Proof of Theorem 9.4. Let

$$e_n = z_n - y_n \qquad (n = 0, 1, 2, \ldots).$$

Subtracting (9.88) from (9.89), we find

$$\begin{gathered}\sum_{s=0}^{k} \alpha_s e_{n+1-s} = g_n,\\ g_n = h\sum_{s=0}^{k} \beta_s[f(x_{n+1-s}, z_{n+1-s}) - f(x_{n+1-s}, y_{n+1-s})] + r_n.\end{gathered} \tag{9.97}$$

Thus, each component e_n^μ of e_n satisfies a difference equation (9.91) with g_n^μ as inhomogeneous term.

To prove the necessity of (9.55), we need only consider the particular case where $f(x, y) \equiv 0$ and $r_n = 0$ $(n \geq k-1)$. Then $g_n = 0$, and e_n^μ is a solution of the homogeneous difference equation associated with (9.91). The general solution of this equation is a linear combination (with, in general, complex coefficients) of the k solutions

$$\zeta_i^n n^j \qquad (j = 0, 1, \ldots, q_i - 1;\ i = 1, 2, \ldots, q), \tag{9.98}$$

where the ζ_i denote the distinct zeros of the generating polynomial $a(z)$ and the q_i denote the respective multiplicities. Suppose (9.55) is not fulfilled. Then either $|\zeta_i| > 1$ for some i, or $|\zeta_i| = 1$ and $j > 0$ for some i and j. In both cases the corresponding solution (9.98) is not bounded in absolute value as $n \to \infty$. It can always be made a component of e_n^μ by a suitable choice of the initial values e_κ^μ. Since, furthermore, e_n^μ does not depend on h, we can choose h so small and n correspondingly so large as to make $|e_n^\mu|$, and thus $\|e_n\|$, as large as we please, no matter how small the initial errors are. This contradicts stability as defined above.

To prove the sufficiency part of the theorem, we assume that (9.55) holds. Then all solutions (9.98) are bounded as $n \to \infty$. Since the solutions $H_{n\kappa}$ of Lemma 9.1 are linear combinations of the solutions (9.98), it follows that

$$|H_{n\kappa}| \leq H \qquad (n \geq 0, \kappa = 0, 1, \ldots, k - 1)$$

for some $H \geq 1$. From (9.93), isolating the term with $\nu = n - 1$ in the second sum, we then have

$$|e_n^\mu| \leq H\left(\sum_{\kappa=0}^{k-1} |e_\kappa^\mu| + |g_{n-1}^\mu| + \sum_{\nu=k-1}^{n-2} |g_\nu^\mu|\right).$$

Summing over μ,

$$\sum_{\mu=1}^{m} |e_n^\mu| \leq H\left(\sum_{\kappa=0}^{k-1} \sum_{\mu=1}^{m} |e_\kappa^\mu| + \sum_{\mu=1}^{m} |g_{n-1}^\mu| + \sum_{\nu=k-1}^{n-2} \sum_{\mu=1}^{m} |g_\nu^\mu|\right).$$

Since $\|u\| \leq \sum_{\mu=1}^{m} |u^\mu| \leq \sqrt{m}\,\|u\|$ for any vector $u \in E_m$, we find

$$\|e_n\| \leq H\sqrt{m}\left(\sum_{\kappa=0}^{k-1} \|e_\kappa\| + \|g_{n-1}\| + \sum_{\nu=k-1}^{n-2} \|g_\nu\|\right).$$

From (9.97) and (9.2) we obtain the estimates

$$\|g_{n-1}\| \leq h\beta K\,\|e_n\| + h\beta K \sum_{s=1}^{k} \|e_{n-s}\| + \|r_{n-1}\|,$$

$$\|g_\nu\| \leq h\beta K \sum_{s=0}^{k} \|e_{\nu+1-s}\| + \|r_\nu\|,$$

where $\beta = \max_s |\beta_s|$, so that, after some grouping together, and further enlargement,

$$(1 - h\beta KH\sqrt{m})\,\|e_n\| \leq h\beta KH(k + 1)\sqrt{m} \sum_{\nu=0}^{n-1} \|e_\nu\|$$

$$+ H\sqrt{m}\left(\sum_{\kappa=0}^{k-1} \|e_\kappa\| + \sum_{\nu=k-1}^{n-1} \|r_\nu\|\right).$$

Assume now that $h \leq h_0$, $1 - h_0\beta KH\sqrt{m} > 0$ and that the perturbations in (9.89) are of class $C(\delta)$. Then

$$\|e_n\| \leq hA \sum_{\nu=0}^{n-1} \|e_\nu\| + B\delta \qquad (n = 1, 2, \ldots), \tag{9.99}$$

where $A = \dfrac{\beta KH(k+1)\sqrt{m}}{1 - h_0\beta KH\sqrt{m}} > 0, \qquad B = \dfrac{H\sqrt{m}}{1 - h_0\beta KH\sqrt{m}} > 1.$

Observe that the difference equation

$$E_n = hA \sum_{\nu=0}^{n-1} E_\nu + B\delta \qquad (n = 1, 2, \ldots), \qquad E_0 = B\delta \tag{9.100}$$

has the solution

$$E_n = B\delta(1 + hA)^n.$$

Obviously, $\|e_0\| \leq E_0$, and subtracting (9.100) from (9.99), we get

$$\|e_n\| - E_n \leq hA \sum_{\nu=0}^{n-1} (\|e_\nu\| - E_\nu) \qquad (n = 1, 2, \ldots).$$

Therefore, by induction, $\|e_n\| - E_n \leq 0$. Thus, for $n \leq N$,

$$\|e_n\| \leq E_n \leq E_N \leq B\delta(1 + hA)^{(X-x_0)/h} \leq B\delta e^{A(X-x_0)}.$$

Choosing then $\delta = B^{-1}e^{-A(X-x_0)}\epsilon$, we obtain $\|e_n\| \leq \epsilon$, which proves stability. This completes the proof of Theorem 9.4.

As interesting as stability is the question of convergence, that is, the question of whether $\max\limits_{0 \leq n \leq N} \|z_n - y(x_n)\| \to 0$ as $h \to 0$ whenever

$$\sum_{\kappa=0}^{k-1} \|z_\kappa - y(x_\kappa)\| + \sum_{n=k-1}^{N-1} \|r_n\| = O(h).$$

Convergence in this sense implies stability. Conversely, it can be shown [14, 27] that *every stable method of order* ≥ 1 *is also convergent.* For further results along these lines, see [20].

A stable method, though producing satisfactory results for h sufficiently small, may very well fail to do so if h is not small enough. This is likely to occur if the solution of the differential equation decreases exponentially, whereas the approximating difference equation has solutions which increase exponentially. Instabilities of this weaker type are studied in [12, 45, 49, 54, 70, 71].

REFERENCES

Among the books specifically devoted to the numerical solution of differential equations are those numbered 6, 13, 14, 22, 41, and 46 in the bibliography in Sec. 2.43, together with

Š. E. Mikeladze, "New Methods of Integration of Differential Equations and their Applications to Problems in the Theory of Elasticity," Gosudarstv. Izdat. Tehn.-Teor. Lit., Moscow, 1951 (in Russian).

H. von Sanden, "Praxis der Differentialgleichungen," 4th ed., Walter de Gruyter & Co., Berlin, 1955.

Various texts on numerical analysis pay substantial attention to the present subject, notably those numbered 9, 27, 29, 35, 36, 38, 48, 51, 59, 70, 71, and 73 in the bibliography in Sec. 2.43.

Among the other relevant works are the following:

1. J. Albrecht, Beiträge zum Runge-Kutta-Verfahren, *Z. Angew. Math. Mech.*, vol. 35, pp. 100–110, 1955.

2. N. S. Bahvalov, On the Estimation of the Error in the Numerical Integration of Differential Equations by Adams' Extrapolation Method, *Dokl. Akad. Nauk SSSR*, vol. 104, pp. 683–686, 1955 (in Russian).

3. F. Bashforth and J. C. Adams, "An Attempt to Test the Theories of Capillary Action . . . with an Explanation of the Method of Integration Employed," Cambridge University Press, London, 1883.

4. L. Bieberbach, "Theorie der Differentialgleichungen," Springer-Verlag, Berlin, 1923.

5. L. Bieberbach, On the Remainder of the Runge-Kutta Formula in the Theory of Ordinary Differential Equations, *Z. Angew. Math. Phys.*, vol. 2, pp. 233–248, 1951.

6. P. Brock and F. J. Murray, The Use of Exponential Sums in Step by Step Integration, *Math. Tables Aids Comput.*, vol. 6, pp. 63–78, ^1952. ^ pp 138-150,

7. E. Bukovics, Eine Verbesserung und Verallgemeinerung des Verfahrens von Blaess zur numerischen Integration gewöhnlicher Differentialgleichungen, *Österreich. Ing.-Arch.*, vol. 4, pp. 338–349, 1950.

8. E. Bukovics, Beiträge zur numerischen Integration, II, III, *Monatsh. Math.*, vol. 57, pp. 333–350, 1954; vol. 58, pp. 258–265, 1954.

9. V. Capra, Nuove formule per l'integrazione numerica delle equazioni differenziali ordinarie del 1° e del 2° ordine, *Univ. e Politec. Torino. Rend. Sem. Mat.*, vol. 16, pp. 301–359, 1956–1957.

10. V. Capra, Valutazione degli errori nella integrazione numerica dei sistemi di equazioni differenziali ordinarie, *Atti Accad. Sci. Torino. Cl. Sci. Fis. Mat. Nat.*, vol. 91, pp. 188–203, 1956–1957.

11. J. W. Carr, III, Error Bounds for the Runge-Kutta Single-step Integration Process, *J. Assoc. Comput. Mach.*, vol. 5, pp. 39–44, 1958.

12. L. Collatz, Über die Instabilität beim Verfahren der zentralen Differenzen für Differentialgleichungen zweiter Ordnung, *Z. Angew. Math. Phys.*, vol. 4, pp. 153–154, 1953.

13. S. D. Conte and R. F. Reeves, A Kutta Third-order Procedure for Solving Differential Equations Requiring Minimum Storage, *J. Assoc. Comput. Mach.*, vol. 3, pp. 22–25, 1956.

14. G. Dahlquist, Convergence and Stability in the Numerical Integration of Ordinary Differential Equations, *Math. Scand.*, vol. 4, pp. 33–53, 1956.

15. G. Dahlquist, Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations, *Kungl. Tekn. Högsk. Handl. Stockholm,* no. 130, pp. 1–87, 1959.

16. W. J. Duncan, Technique of the Step-by-step Integration of Ordinary Differential Equations, *Phil. Mag.*, ser. 7, vol. 34, pp. 493–509, 1948.

17. E. Fehlberg, Eine Methode zur Fehlerverkleinerung beim Runge-Kutta-Verfahren, *Z. Angew. Math. Mech.*, vol. 38, pp. 421–426, 1958.

18. W. Gautschi, Über den Fehler des Runge-Kutta-Verfahrens für die numerische Integration gewöhnlicher Differentialgleichungen *n*-ter Ordnung, *Z. Angew. Math. Phys.*, vol. 6, pp. 456–461, 1955.

19. S. Gill, A Process for the Step-by-step Integration of Differential Equations in an Automatic Digital Computing Machine, *Proc. Cambridge Philos. Soc.*, vol. 47, pp. 96–108, 1951.

20. A. D. Gorbunov and B. M. Budak, On the Convergence of Certain Finite Difference Processes for the Equations $y' = f(x, y)$ and $y'(x) = f(x, y(x), y(x - \tau(x)))$, *Dokl. Akad. Nauk SSSR*, vol. 119, pp. 644–647, 1958 (in Russian).

21. S. Gorn and R. Moore, "Automatic Error Control; The Initial Value Problem in Ordinary Differential Equations," Ballistic Research Laboratory Report 893, 1954.

22. "Interpolation and Allied Tables," H.M. Stationery Office, London, 1956.

23. P. C. Hammer and J. W. Hollingsworth, Trapezoidal Methods of Approximating Solutions of Differential Equations, *Math. Tables Aids Comput.*, vol. 9, pp. 92–96, 1955.

24. R. W. Hamming, Stable Predictor-Corrector Methods for Ordinary Differential Equations, *J. Assoc. Comput. Mach.*, vol. 6, pp. 37–47, 1959.

25. P. Henrici, Theoretical and Experimental Studies on the Accumulation of Error in the Numerical Solution of Initial Value Problems for Systems of Ordinary Differential Equations, pp. 36–44 in *Proc. Intern. Conf. on Information Processing, Paris*, 1959, UNESCO, Paris, 1960.

26. K. Heun, Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Variable, *Z. Math. Phys.*, vol. 45, pp. 23–38, 1900.

27. T. E. Hull and W. A. J. Luxemburg, Numerical Methods and Existence Theorems for Ordinary Differential Equations, *Numer. Math.*, vol. 2, pp. 30–41, 1960.

28. A. Hut̆a, Une amélioration de la méthode de Runge-Kutta-Nyström pour la résolution numérique des équations différentielles du premier ordre, *Acta Fac. Nat. Univ. Comenian. Math.*, vol. 1, pp. 201–224, 1956.

29. A. Hut̆a, Contribution à la formule de sixième ordre dans la méthode de Runge-Kutta-Nyström, *Acta Fac. Nat. Univ. Comenian. Math.*, vol. 2, pp. 21–24, 1957.

30. G. H. Keitel, An Extension of Milne's Three-point Method, *J. Assoc. Comput. Mach.*, vol. 3, pp. 212–222, 1956.

31. J. Kuntzmann, Deux formules optimales du type de Runge-Kutta, *Chiffres*, vol. 2, pp. 21–26, 1959.

32. W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, *Z. Math. Phys.*, vol. 46, 435–453, 1901.

33. E. Lindelöf, Remarques sur l'intégration numérique des équations différentielles ordinaires, *Acta Soc. Sci. Fenni. Nova Ser. A*, vol. 2, pp. 1–21, 1938.

34. W. Liniger, Zur Stabilität der numerischen Integrationsmethoden für Differentialgleichungen, Dissertation, Lausanne, 1956; Zürich, 1957.

35. M. Lotkin, On the Accuracy of Runge-Kutta's Method, *Math. Tables Aids Comput.*, vol. 5, pp. 128–133, 1951.

36. M. Lotkin, A New Integrating Procedure of High Accuracy, *J. Math. Phys.*, vol. 31, pp. 29–34, 1952.

37. S. M. Lozinskiĭ, Error Estimation in the Numerical Integration of Ordinary Differential Equations, I, *Izv. Vysš. Učebn. Zavedeniĭ Mat.*, no. 5, pp. 52–90, 1958 (in Russian).

38. D. W. Martin, Runge-Kutta Methods for Integrating Differential Equations on High-speed Digital Computers, *Comput. J.*, vol. 3, pp. 118–123, 1958.

39. P. Matthieu, Über die Fehlerabschätzung beim Extrapolationsverfahren von Adams, I, II, *Z. Angew. Math. Mech.*, vol. 31, pp. 356–370, 1951; vol. 33, pp. 26–41, 1953.

40. W. E. Milne, Numerical Integration of Ordinary Differential Equations, *Amer. Math. Monthly*, vol. 33, pp. 455–460, 1926.

41. R. von Mises, Zur numerischen Integration von Differentialgleichungen, *Z. Angew. Math. Mech.*, vol. 10, pp. 81–92, 1930.

42. A. R. Mitchell and J. W. Craggs, Stability of Difference Relations in the Solution of Ordinary Differential Equations, *Math. Tables Aids Comput.*, vol. 7, pp. 127–129, 1953.

43. E. Mohr, Über das Verfahren von Adams zur Integration gewöhnlicher Differentialgleichungen, *Math. Nachr.*, vol. 5, pp. 209–218, 1951.

44. I. S. Muhin, Application of the Markov-Hermite Interpolation Polynomials for the Numerical Integration of Ordinary Differential Equations, *Prikl. Mat. Meh.*, vol. 16, pp. 231–238, 1952 (in Russian).

45. I. S. Muhin, On the Accumulation of Errors in Numerical Integration of Differential Equations, *Prikl. Mat. Meh.*, vol. 16, pp. 753–755, 1952 (in Russian).

46. E. J. Nyström, Über die numerische Integration von Differentialgleichungen, *Acta Soc. Sci. Fenn.*, vol. 50, no. 13, pp. 1–55, 1925.

47. E. Pflanz, Bemerkungen über die Methode von G. Duffing zur Integration von Differentialgleichungen, *Z. Angew. Math. Mech.*, vol. 28, pp. 167–172, 1948.

48. W. Quade, Numerische Integration von gewöhnlichen Differentialgleichungen durch Interpolation nach Hermite, *Z. Angew. Math. Mech.*, vol. 37, pp. 161–169, 1957.

49. W. Quade, Über die Stabilität numerischer Methoden zur Integration gewöhnlicher Differentialgleichungen erster Ordnung, *Z. Angew. Math. Mech.*, vol. 39, pp. 117–134, 1959.

50. H. Rademacher, On the Accumulation of Errors in Processes of Integration on High-speed Calculating Machines, *Ann. Comput. Lab. Harvard Univ.*, vol. 16, pp. 176–187, 1948.

51. L. F. Richardson, The Deferred Approach to the Limit, *Philos. Trans. Roy. Soc. London Ser. A*, vol. 226, pp. 299–349, 1927.

52. C. Runge, Über die numerische Auflösung von Differentialgleichungen, *Math. Ann.*, vol. 46, pp. 167–178, 1895.

53. C. Runge, Ueber die numerische Auflösung totaler Differentialgleichungen, *Nachr. K. Ges. Wiss. Göttingen*, pp. 252–257, 1905.

54. H. Rutishauser, Über die Instabilität von Methoden zur Integration gewöhnlicher Differentialgleichungen, *Z. Angew. Math. Phys.*, vol. 3, pp. 65–74, 1952.

55. H. Rutishauser, Bemerkungen zur numerischen Integration gewöhnlicher Differentialgleichungen *n*-ter Ordnung, *Z. Angew. Math. Phys.*, vol. 6, pp. 497–498, 1955.

56. H. E. Salzer, Osculatory Extrapolation and a New Method for the Numerical Integration of Differential Equations, *J. Franklin Inst.*, vol. 262, pp. 111–119, 1956.

57. T. H. Southard and E. C. Yowell, An Alternative "Predictor-Corrector" Process, *Math. Tables Aids Comput.*, vol. 6, pp. 253–254, 1952.

58. K. Stohler, Eine Vereinfachung bei der numerischen Integration gewöhnlicher Differentialgleichungen, *Z. Angew. Math. Mech.*, vol. 23, pp. 120–122, 1943.

59. M. R. Šura-Bura, Error Estimates in Numerical Integration of Ordinary Differential Equations, *Prikl. Mat. Meh.*, vol. 16, pp. 575–588, 1952 (in Russian).

60. J. Todd, Notes on Modern Numerical Analysis, I: Solution of Differential Equations by Recurrence Relations, *Math. Tables Aids Comput.*, vol. 4, pp. 39–44, 1950.

61. W. Tollmien, Über die Fehlerabschätzung beim Adamsschen Verfahren zur Integration gewöhnlicher Differentialgleichungen, *Z. Angew. Math. Mech.*, vol. 18, pp. 83–90, 1938.

62. W. Tollmien, Bemerkung zur Fehlerabschätzung beim Adamsschen Interpolationsverfahren, *Z. Angew. Math. Mech.*, vol. 33, pp. 151–155, 1953.

63. W. Uhlmann, Fehlerabschätzungen bei Anfangswertaufgaben gewöhnlicher Differentialgleichungssysteme 1. Ordnung, *Z. Angew. Math. Mech.*, vol. 37, pp. 88–99, 1957.

64. W. Uhlmann, Fehlerabschätzungen bei Anfangswertaufgaben einer gewöhnlichen Differentialgleichung höherer Ordnung, *Z. Angew. Math. Mech.*, vol. 37, pp. 99–111, 1957.

65. M. Urabe and T. Tsushima, On Numerical Integration of Ordinary Differential Equations, *J. Sci. Hiroshima Univ. Ser. A*, vol. 17, pp. 193–219, 1953.

66. O. Vejvoda, Error Estimate for the Runge-Kutta Formula, *Aplikace Mat.*, vol. 2, pp. 1–23, 1957 (in Czech).

67. J. Weissinger, Eine verschärfte Fehlerabschätzung zum Extrapolationsverfahren von Adams, *Z. Angew. Math. Mech.*, vol. 30, pp. 356–363, 1950.

68. J. Weissinger, Eine Fehlerabschätzung für die Verfahren von Adams und Störmer, *Z. Angew. Math. Mech.*, vol. 32, pp. 62–67, 1952.

69. H. S. Wilf, An Open Formula for the Numerical Integration of First Order Differential Equations, *Math. Tables Aids Comput.*, vol. 11, pp. 201–203, 1957; vol. 12, pp. 55–58, 1958.

70. H. S. Wilf, A Stability Criterion for Numerical Integration, *J. Assoc. Comput. Mach.*, vol. 6, pp. 363–365, 1959.

71. B. Zondek and J. W. Sheldon, On the Error Propagation in Adams' Extrapolation Method, *Math. Tables Aids Comput.*, vol. 13, pp. 52–55, 1959.

72. R. Zurmühl, Zur numerischen Integration gewöhnlicher Differentialgleichungen zweiter und höherer Ordnung, *Z. Angew. Math. Mech.*, vol. 20, pp. 104–116, 1940.

73. R. Zurmühl, Runge-Kutta-Verfahren zur numerischen Integration von Differentialgleichungen *n*-ter Ordnung, *Z. Angew. Math. Mech.*, vol. 28, pp. 173–182, 1948.

Added in proof. Since the preparation of this manuscript many results have appeared or come to our attention which have direct bearing on the subject matter treated here. While it has been impossible for us to include all of them in detail, we append a selected list of them for further reference:

1′. R. Alonso, A Starting Method for the Three-point Adams Predictor-Corrector Method, *J. Assoc. Comput. Mach.*, vol. 7, pp. 176–180, 1960.

2′. R. Ansorge and W. Töring, Zur Stabilität des Nyströmschen Verfahrens, *Z. Angew. Math. Mech.*, vol. 40, pp. 568–570, 1960.

3′. I. S. Berezin and N. P. Židkov, "Numerical Methods," vols. 1 and 2, Moscow, 1959 (in Russian).

4′. Ch. Blanc, Sur les formules d'intégration approchée d'équations différentielles, *Arch. Math.*, vol. 5, pp. 301–308, 1954.

5′. B. M. Budak and A. D. Gorbunov, Stability of Calculation Processes Involved in the Solution of the Cauchy Problem for the Equation $dy/dx = f(x,y)$ by Multi-point Difference Methods, *Dokl. Akad. Nauk SSSR*, vol. 124, pp. 1191–1194, 1959 (in Russian).

6′. F. Ceschino and J. Kuntzmann, Impossibilité d'un certain type de formule d'intégration approchée à pas liés, *Chiffres*, vol. 1, pp. 95–101, 1958.

7′. F. Ceschino and J. Kuntzmann, Faut il passer à la forme canonique dans les problèmes différentielles de conditions initiales?, pp. 33–36, in *Proc. Intern. Conf. on Information Processing*, *Paris*, 1959, UNESCO, Paris, 1960.

8′. S. C. R. Dennis, The Numerical Integration of Ordinary Differential Equations Possessing Exponential Type Solutions, *Proc. Cambridge Philos. Soc.*, vol. 56, pp. 240–246, 1960.

9′. E. Fehlberg, Neue genauere Runge-Kutta-Formeln für Differentialgleichungen zweiter Ordnung, *Z. Angew. Math. Mech.*, vol. 40, pp. 252–259, 1960.

10′. E. Fehlberg, Neue genauere Runge-Kutta-Formeln für Differentialgleichungen *n*-ter Ordnung, *Z. Angew. Math. Mech.*, vol. 40, pp. 449–455, 1960.

11′. E. Fehlberg, Numerisch stabile Interpolationsformeln mit günstiger Fehlerfortpflanzung für Differentialgleichungen erster und zweiter Ordnung, *Z. Angew. Math. Mech.*, vol. 41, pp. 101–110, 1961.

12′. C. V. D. Forrington, Extensions of the Predictor-Corrector Method for the Solution of Systems of Ordinary Differential Equations, *Comput. J.*, vol. 4, pp. 80–84, 1961.

13′. P. Henrici, "Discrete Variable Methods for Ordinary Differential Equations," John Wiley & Sons, Inc., New York (to appear).

14′. T. E. Hull and A. C. R. Newbery, Error Bounds for a Family of Three-point Integration Procedures, *J. Soc. Indust. Appl. Math.*, vol. 7, pp. 402–412, 1959.

15′. T. E. Hull and A. C. R. Newbery, Integration Procedures which Minimize Propagated Errors, *J. Soc. Indust. Appl. Math.*, vol. 9, pp. 31–47, 1961.

16′. V. I. Krylov, Convergence and Stability of the Numerical Solution of a Differential Equation of Second Order, *Dokl. Akad. Nauk BSSR*, vol. 4, pp. 187–189, 1960 (in Russian).

17′. W. E. Milne and R. R. Reynolds, Stability of a Numerical Solution of Differential Equations, *J. Assoc. Comput. Mach.*, vol. 6, pp. 196–203, 1959; vol. 7, pp. 46–56, 1960.

18′. National Physical Laboratory, "Modern Computing Methods," Notes on Applied Science, No. 16, 2d ed., H. M. Stationery Office, London, 1961.

19′. A. Ralston, Numerical Integration Methods for the Solution of Ordinary Differential Equations, pp. 95–109, in "Mathematical Methods for Digital Computers," A. Ralston and H. S. Wilf (eds.), John Wiley & Sons, Inc., New York, 1960.

20′. A. Ralston, Some Theoretical and Computational Matters Relating to Predictor-Corrector Methods of Numerical Integration, *Comput. J.*, C vol. 4, pp. 64–67, 1961.

21′. J. R. Rice, Split Runge-Kutta Method for Simultaneous Equations, *J. Res. Nat. Bur. Standards*, vol. 64B, pp. 151–170, 1960.

22′. W. Richter, Sur l'erreur commise dans la méthode d'intégration de Milne, *C. R. Acad. Sci. Paris*, vol. 233, pp. 1342–1344, 1951.

23′. W. Richter, Estimation de l'erreur commise dans la méthode de M. W. E. Milne pour l'intégration d'un système de *n* équations différentielles du premier ordre, Thèse, Université de Neuchâtel, 1952.

24′. M. J. Romanelli, Runge-Kutta Methods for the Solution of Ordinary Differential Equations, pp. 110–120, in "Mathematical Methods for Digital Computers," A. Ralston and H. S. Wilf (eds.), John Wiley & Sons, Inc., New York, 1960.

25′. H. Rutishauser, Bemerkungen zur numerischen Integration gewöhnlicher Differentialgleichungen *n*-ter Ordnung, *Numer. Math.*, vol. 2, pp. 263–279, 1960.

26′. E. Schechter, On the Error of Runge-Kutta's Method of Numerical Integration, *Acad. Repub. Pop. Romîne. Fil. Cluj. Stud. Cerc. Mat.*, vol. 8, pp. 115–124, 1957 (in Romanian).

27′. E. Schechter, Error Estimation in Some Methods of Numerical Integration of Differential Equations, *Acad. Repub. Pop. Romîne. Fil. Cluj. Stud. Cerc. Mat.*, vol. 9, pp. 343–350, 1958 (in Romanian).

28′. J. Schröder, Fehlerabschätzung mit Rechenanlagen bei gewöhnlichen Differentialgleichungen erster Ordnung, *Numer. Math.*, vol. 3, pp. 39–61, 1961.

29′. J. Schröder, Verbesserung einer Fehlerabschätzung für gewöhnliche Differentialgleichungen erster Ordnung, *Numer. Math.*, vol. 3, pp. 125–130, 1961.

30′. L. Stoller and D. Morrison, A Method for the Numerical Integration of Ordinary Differential Equations, *Math. Tables Aids Comput.*, vol. 12, pp. 269–272, 1958.

31′. S. S. Toktalaeva, Ordinate Formulas for Numerical Integration of Ordinary Differential Equations of First Order, *Vyčisl. Mat.*, vol. 5, pp. 3–57, 1959 (in Russian).

32′. M. Urabe and S. Mise, A Method of Numerical Integration of Analytic Differential Equations, *J. Sci. Hiroshima Univ. Ser. A*, vol. 19, pp. 307–320, 1955.

33′. M. Urabe, Theory of Errors in Numerical Integration of Ordinary Differential Equations, *J. Sci. Hiroshima Univ. Ser. A-I*, vol. 25, pp. 3–62, 1961.

34′. R. Zanovello, Sul metodo di Runge-Kutta per l'equazione differenziale $y' = f(x,y)$, *Rend. Sem. Mat. Univ. Padova*, vol. 30, pp. 349–360, 1960.

46.2. [56] “Comportement asymptotique des coefficients dans les formules d’intégration d’Adams, de Störmer et de Cowell”

[56] “Comportement asymptotique des coefficients dans les formules d’intégration d’Adams, de Störmer et de Cowell ”, *C. R. Acad. Sci. Paris Sér. A-B*, **283**, no. 10, Aiii, A787–A788 (1976).

ANALYSE NUMÉRIQUE. — *Comportement asymptotique des coefficients dans les formules d'intégration d'Adams, de Störmer et de Cowell.* Note (*) de M. **Walter Gautschi**, présentée par M. Jacques-Louis Lions.

On donne une démonstration brève d'une formule asymptotique de Steffensen pour les coefficients d'Adams-Moulton. Des résultats analogues subsistent pour les coefficients de Störmer et de Cowell.

1. Les coefficients dans les méthodes d'intégration d'Adams-Bashforth et d'Adams-Moulton sont donnés, respectivement, par

$$\gamma_n = \int_0^1 \binom{n+t-1}{n} dt, \quad \gamma_n^* = \int_{-1}^0 \binom{n+t-1}{n} dt, \qquad n = 0, 1, 2 \ldots$$

Pour grand n, on déduit les formules asymptotiques

$$\gamma_n \sim \frac{1}{\ln n}, \qquad n \to \infty \tag{1}$$

et

$$\gamma_n^* \sim -\frac{1}{n \ln^2 n}, \qquad n \to \infty. \tag{2}$$

La relation (1) a été posée comme problème par M. Spital ([2]), et a été démontrée, entre autres, par M. Trench ([3]). La relation (2) est due à M. Steffensen ([4]). Remarquons ici que la méthode de M. Trench peut être utilisée pour donner une brève démonstration de (2). Elle aussi fournit aisément les formules

$$\sigma_n \sim \frac{1}{\ln^2 n}, \quad \sigma_n^* \sim -\frac{2}{n \ln^3 n}, \qquad n \to \infty, \tag{3}$$

pour les coefficients de Störmer et de Cowell, définis, respectivement, par ([1]), ([5]) :

$$\sigma_n = -(n-1) \int_{-1}^0 \binom{n+t-1}{n} dt, \qquad \sigma_n^* = -\frac{1}{n} \int_{-1}^0 [(n-1)t+1] \binom{n+t-2}{n-1} dt.$$

2. A l'aide de la fonction gamma, on a

$$\gamma_n^* = \int_{-1}^0 \frac{\Gamma(n+t)}{\Gamma(n+1)} \frac{dt}{\Gamma(t)}.$$

Appliquons la formule classique

$$\frac{\Gamma(n+t)}{\Gamma(n+1)} = n^{t-1} \left[1 + 0\left(\frac{1}{n}\right) \right], \qquad n \to \infty,$$

valable uniformément pour t dans tout intervalle borné. On obtient :

$$(4) \qquad \gamma_n^* = \left\{\int_{-1}^{0} \frac{n^{t-1}}{\Gamma(t)}\, dt\right\}\left[1+0\left(\frac{1}{n}\right)\right], \qquad n \to \infty.$$

Soit

$$h(t) = \frac{1}{\Gamma(t)} = \frac{t(t+1)}{\Gamma(t+2)}, \qquad -1 \leqq t \leqq 0.$$

En intégrant deux fois par parties, on trouve

$$(5) \int_{-1}^{0} \frac{n^{t-1}}{\Gamma(t)}\, dt = \frac{1}{n \ln n}\left[h(0) - \frac{h(-1)}{n}\right] - \frac{1}{n \ln^2 n}\left[h'(0) - \frac{h'(-1)}{n}\right] + \frac{1}{\ln^2 n}\int_{-1}^{0} h''(t)\, n^{t-1}\, dt.$$

La dérivée seconde, $h''(t)$, étant bornée sur $[-1, 0]$, l'intégrale à droite dans (5) est de l'ordre $0(1/(n \ln n))$. En outre, $h(0) = h(-1) = 0$, et $h'(0) = -h'(-1) = 1$. On obtient donc :

$$(6) \qquad \int_{-1}^{0} \frac{n^{t-1}}{\Gamma(t)}\, dt = -\frac{1}{n \ln^2 n}\left[1+0\left(\frac{1}{\ln n}\right)\right], \qquad n \to \infty,$$

ce qui, tenant compte de (4), établit la relation (2).

La première formule dans (3) est conséquence immédiate de (2), puisque $\sigma_n = -(n-1)\,\gamma_n^*$. La seconde s'obtient de façon analogue à la formule (2), l'erreur relative étant également de l'ordre $0(1/\ln n)$.

[*] Séance du 14 juin 1976.

[1] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley and Sons, New York-London, 1962.

[2] Problem 5347, *Amer. Math. Monthly*, 72, 1965, p. 1136.

[3] Solution 5347, *ibid.*, 73, 1966, p. 1130-1131.

[4] J. F. Steffensen, *Skand. Aktuarietidskr.*, 7, 1924, p. 1-15.

[5] J. Weissinger, *Z. Angew. Math. Mech.*, 32, 1952, p. 62-67.

Department of Computer Sciences,
Purdue University,
Lafayette,
Indiana 47907,
U.S.A.

47

Papers on History

177 A historical note on Gauss–Kronrod quadrature, Numer. Math. 100, 483–484 (2005)
206 Interpolation before and after Lagrange, Rend. Semin. Mat. Univ. Politec. Torino 70, 347–368 (2012)

47.1. [177] "A historical note on Gauss–Kronrod quadrature"

[177] "A historical note on Gauss–Kronrod quadrature", *Numer. Math.* **100**, 483–484 (2005).

Numer. Math. (2005) 100: 483–484
Digital Object Identifier (DOI) 10.1007/s00211-005-0592-7

Numerische
Mathematik

A historical note on Gauss–Kronrod quadrature

Walter Gautschi

Department of Computer Sciences, Purdue University, West Lafayette,
IN 47907-2066, USA; E-mail: wxg@cs.purdue.edu

Received December 9, 2004
Published online: April 6, 2005 – © Springer-Verlag 2005

Summary. The idea of Gauss–Kronrod quadrature, in a germinal form, is traced back to an 1894 paper of R. Skutsch.

Mathematics Subject Classification (2000): 01–08, 65D32

The idea of inserting $n+1$ nodes into an n-point Gaussian quadrature rule and choosing them and the weights of the resulting $(2n+1)$-point quadrature rule in such a manner as to maximize the polynomial degree of exactness is generally attributed to A.S. Kronrod [2], [3]. This is entirely justified, given that Kronrod developed the underlying theory and produced extensive numerical tables. The same idea, nevertheless, can be traced back at least to 1894, when R. Skutsch [5] pointed out the possibility of obtaining in this way a $(2n+1)$-point formula of degree of exactness $3n+1$ (resp. $3n+2$ if n is odd). He also notes that the degree of exactness of the n-point Gauss formula cannot be improved by inserting fewer than $n+1$ points (*ibid.*, p. 81), a result proved later by Monegato [4, Lemma 1].

On p. 83, the paper also gives numerical results to 11 decimal digits for the integral $\int_{-1}^{1} \mathrm{d}x/(x+3) = \ln 2$. The 7-point extension of the 3-point Gauss formula is compared in this example with, among others, the 3-point and 7-point Gauss formulae. Only end results are stated, none of the respective quadrature formulae. They are all correct to 11 digits, except for the result for the 7-point extension, which is off in the last two digits.

Could it be that Stieltjes knew about Skutsch's paper? In his last letter to Hermite [1, Vol. 2, p. 439], dated November 8, 1894, Stieltjes considered orthogonal polynomials relative to a sign-variable weight function (a Legendre polynomial), which are relevant to the Kronrod extension of Gauss formulae, but makes no reference to mechanical quadrature.

References

1. Baillaud, B., Bourget, H.: Correspondance d'Hermite et de Stieltjes I, II, Gauthier-Villars, Paris, 1905
2. Kronrod, A.S.: Integration with control of accuracy (Russian), Dokl. Akad. Nauk SSSR **154**, 283–286 (1964)
3. Kronrod, A.S. 1964. Nodes and weights for quadrature formulae. Sixteen-place tables (Russian), Izdat. Nauka, Moscow. [English translation in: Consultants Bureau, New York, 1965]
4. Monegato, Giovanni: An overview of results and questions related to Kronrod schemes. In: Numerische Integration, G. Hämmerlin (ed.), Internat. Ser. Numer. Math. Birkhäuser, Basel, **45**, 231–240 (1979)
5. Skutsch, Rudolf: Ueber Formelpaare der mechanischen Quadratur. Arch. Math. Phys. **13**(2), 78–83 (1894)

47.2. [206] "Interpolation before and after Lagrange"

[206] "Interpolation before and after Lagrange", *Rend. Semin. Mat. Univ. Politec. Torino* **70**, 347–368 (2012).

Rend. Sem. Mat. Univ. Politec. Torino
Vol. 70, 4 (2012), 347 – 368

W. Gautschi

INTERPOLATION BEFORE AND AFTER LAGRANGE

Abstract. A brief history of interpolation is given and selected aspects thereof, centered around Lagrange's contribution to the subject.

1. Introduction

The idea and practice of interpolation has a long history going back to antiquity and extending to modern times. We will briefly sketch the early development of the subject in ancient times and the middle ages through the 17th century, culminating in the work of Newton. We next draw attention to a little-known paper of Waring predating Lagrange's interpolation formula by 16 years. The rest of the paper deals with a few selected contributions made after Lagrange till recent times. They include the computationally more attractive barycentric form of Lagrange's formula, the theory of error and convergence based on real-variable and complex-variable analyses, Hermite and Hermite-Fejér as well as nonpolynomial interpolation. Applications to numerical quadrature and the solution of ordinary and partial differential equations are briefly indicated.

As seems appropriate for this auspicious occasion, however, we begin with Lagrange himself.

2. The Lagrange interpolation formula

In the words of Lagrange [67, v. 7, p. 535]: "La méthode des interpolations est une des plus ingénieuses et des plus utiles que l'Astronomie possède," echoing what Newton wrote to Oldenburg in 1676 about the problem of interpolation [76, p. 137]: "...it ranks among the most beautiful of all that I could wish to solve." Both citations indicate the high regard in which the problem of interpolation was held by scholars of the 17th and 18th century, and the former the principal area — astronomy — in which it was applied. Of course, times have changed, but the utility aspect of interpolation is as valid today as it was three hundred years ago.

There are two memoirs of Lagrange, [47, 48], in which he deals with special questions of interpolation, polynomial as well as trigonometric, with equally spaced values of the independent variable. They involve differences of one sort or another, in the tradition of Newtonian mathematics. What today is called the "Lagrange interpolation formula" is not dealt with in any of Lagrange's memoirs, but appears in the fifth lecture of his "Leçons élémentaires sur les mathématiques" delivered in 1795 at the École Normale, which were published in two issues of the *Séances des Écoles Normales*, year III (1794–1795) and reprinted in 1812, at the request of Lagrange, in the *Journal de l'École Polytechnique*; see also [67, v. 7, 271–287]. In this fifth lecture

"Sur l'usage des courbes dans la solution des problèmes," Lagrange discusses two geometric problems and uses analysis to solve them. He then continues with: "En effet, tout se réduit à d'écrire ou faire passer une courbe par plusieurs points, soit que ces points soient donnés par le calcul ou par une construction, ou même par des observations ou des expériences isolées et indépendantes les unes des autres. Ce Problème est, à la vérité, indéterminé; car on peut, à la rigueur, faire passer par des points donnés une infinité de courbes différentes, régulières ou irrégulières, c'est-à-dire soumises à des équations, ou tracées arbitrairement à la main; mais il ne s'agit pas de trouver des solutions quelconques, mais les plus simples et les plus aisées à employer."

Taking for these "simplest solutions" polynomials of appropriate degree, Lagrange then goes on to derive Newton's form of the interpolation polynomial. Thus, he sets up a table of (what we now call) divided differences, which he writes as

$$\begin{array}{cccccc}
x & y & & & & \\
\hline
p & P & & & & \\
q & Q & Q_1 & & & \\
r & R & R_1 & R_2 & & \\
s & S & S_1 & S_2 & S_3 & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

where

$$Q_1 = \frac{Q-P}{q-p}, \quad R_1 = \frac{R-Q}{r-q}, \quad S_1 = \frac{S-R}{s-r}, \quad \ldots$$

and[1]

$$R_2 = \frac{R_1-Q_1}{r-p}, \quad S_2 = \frac{S_1-R_1}{s-q}, \quad S_3 = \frac{S_2-R_2}{s-p},$$

and so on. (Note Lagrange's elegant and uncomplicated notation; the quantities indexed by ν are the νth divided differences, a name not used by Newton but apparently first introduced by de Morgan [58, Ch.XVII, p. 550] in 1842; see [54, footnote 15 on p. 322].) From this table of divided differences, Lagrange obtains, "après les réductions", Newton's formula,

$$(1) \qquad y = P + Q_1(x-p) + R_2(x-p)(x-q) + S_3(x-p)(x-q)(x-r) + \cdots,$$

"qu'il est aisé de continuer aussi loin qu'on voudra." Thus, the diagonal entries in the table of divided differences serve as coefficients in the factorial expansion (1) of the interpolation polynomial.

Lagrange then continues with: "Mais on peut réduire cette solution à une plus grande simplicité par la considération suivante." And, using elementary algebraic manipulations, he arrives at the formula now bearing his name,

$$(2) \qquad y = AP + BQ + CR + DS + \cdots,$$

[1] The third equation, for S_3, added by the author, does not appear in Lagrange's text.

where

$$\begin{aligned} A &= \frac{(x-q)(x-r)(x-s)\cdots}{(p-q)(p-r)(p-s)\cdots}, \\ B &= \frac{(x-p)(x-r)(x-s)\cdots}{(q-p)(q-r)(q-s)\cdots}, \\ C &= \frac{(x-p)(x-q)(x-s)\cdots}{(r-p)(r-q)(r-s)\cdots}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \end{aligned}$$

"en prenant autant de facteurs, dans les numérateurs et dans les dénominateurs, qu'il y aura de points donnés de la courbe, moins un." Lagrange then says about his formula (2) that "elle est préférable par la simplicité de l'Analyse sur laquelle elle est fondée, et par sa forme même, qui est beaucoup plus commode pour le calcul." (Here, "calcul" is probably intended to mean analytic, not necessarily numerical, calculation.) Notice that the first function A has the value 1 at the first point p and vanishes at all the other points. Similarly for the other functions. This is a characteristic property of these coefficient functions, known as the elementary Lagrange interpolation polynomials.

Interestingly, Lagrange then suggests — and this is motivated by the two geometrical problems discussed previously — what is now called "inverse interpolation", i.e., reversing the roles of $p,q,r,s,\ldots$ and $P,Q,R,S,\ldots$ and finding an x for which y has a given value, specifically $y = 0$.

3. The period before Lagrange

3.1. The pre-Newtonian period

Instances of interpolation, especially linear interpolation, can be traced back to ancient Babylonian and Greek times, i.e., several centuries BC and the first centuries AD. They relate to astronomical observations and trigonometric tables used in astronomy. Evidences of higher-order interpolation appear in the early medieval ephemerides of China and India, and also in Arabic and Persian sources. In the Western countries, interpolation theory developed in the late 16th century quite independently from the much earlier work elsewhere, some of the early contributors being Harriot, Napier, Bürgi, and Briggs, of logarithms fame. See [54, Sec. II] for more details.

3.2. Newton

It was in the 17th century when in the Western world, especially England and France, the theory of interpolation began to flourish, driven not only by the exigencies of astronomy, but also by the need for preparing and using logarithmic and other tables. Initially, most attention was given to interpolation at equally spaced points, which gave rise to the calculus of finite differences. Notable contributors include Gregory, Newton, Stirling, and many others. For historical accounts, see, e.g., Joffe [42], Fraser [24], Turnbull [75], and Goldstine [35]. The culmination of this development, however, was

Newton's general formula for interpolation at arbitrary (distinct) points $x_0, x_1, x_2, \ldots$, which in modern notation can be written in the form

$$f(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1) \cdots (x-x_{n-1}) + R_n(f;x), \tag{3}$$

where

$$a_k = [x_0, x_1, \ldots, x_k]f, \quad k = 0,1,2,\ldots,n, \tag{4}$$

are the kth-order divided differences of f with respect to the interpolation abscissae $x_0, x_1, \ldots, x_k$ and $R_n(f;x)$ is the remainder term. Neither Newton nor Lagrange said much about the remainder. The formula (3) (without the remainder) is contained in Lemma V, Book III, of Newton's *Principia* [59, p. 499], occupying not much more than half a page. In a corollary at the end of the lemma, Newton proposes integrating the polynomial part in (3) to approximate a definite integral of f; cf. §4.5.

It may be amusing to record here the curious use of Newton's formula made by Euler in a letter of 1734 to Daniel Bernoulli. Euler sought to compute the common logarithm $\log x$ for some x between 1 and 10 by applying (3) with $x_k = 10^k$, $\log x_k = k$, $k = 0,1,2,\ldots$. (Only Euler would dare to try such a thing!) He calculates the divided differences to be

$$a_k = [x_0, x_1, \ldots, x_k]\log = \frac{(-1)^{k-1}}{10^{k(k-1)/2}(10^k - 1)}, \quad k = 1,2,3,\ldots,$$

and thus, from Newton's formula extended to infinity, obtains

$$S(x) = \sum_{k=1}^{\infty} a_k(x-1)(x-10)\cdots(x-10^{k-1}). \tag{5}$$

Amazingly, the series converges quite fast but, alas, not to $\log x$. Indeed, for $x = 9$, Euler finds $S(9) = .897778586588\ldots$ instead of the correct value $\log 9 = .954242509439\ldots$. In writing this to his friend Daniel Bernoulli, Euler presumably hoped to get from him some sensible explanation of this phenomenon. Unfortunately, Bernoulli's reply has not survived. It is unlikely, though, that he would have been able to shed some light on the matter. Analysis, particularly complex analysis, just hasn't yet sufficiently advanced to provide an answer. What happens, in a nutshell, is that the fast convergence in (5) renders $S(x)$ to be an entire function and therefore incapable of representing the logarithm; cf. [32].

Once a subject has caught Euler's fancy, he usually never abandoned it, but returned to it time and again. In this case, however, it took almost twenty years until (in [20]; see also [19, ser. 1, v. 14, 516–541]) he revisited the function $S(x)$, now using logarithms to an arbitrary base, and developed many interesting identites, among them two special cases of what today is known as the q-binomial theorem. The function studied by Euler has recently been suggested as a way to define a q-analogue of the logarithm [45].

3.3. Waring

Although this may not be the most opportune occasion, it must be said, in the interest of historical accuracy, that Waring [81], a Lucasian Professor at Cambridge University, predates Lagrange by 16 years.

In the cited paper, Waring starts out by saying that "Mr. Briggs was the first person, I believe, that invented a method of differences for interpolating logarithms at small intervals from each other: his principles were followed by Reginald[2] and Mouton in France. Sir Isaac Newton, from the same principles, discovered a general and elegant solution of the abovementioned problem: perhaps a still more elegant one on some accounts has been since discovered by Mess. Nichole and Stirling. In the following theorems the same problem is resolved and rendered somewhat more general, without having any recourse to finding the successive differences." He then formulates and proves a first theorem that reads as follows (in his original wording and notation): "Assume an equation $a+bx+cx^2+dx^3\ldots.x^{n-1}=y$, in which the coefficients a,b,c,d,e, &c. are invariable; let $\alpha,\beta,\gamma,\delta,\varepsilon$, &c. denote n values of the unknown quantity x, whose correspondent values of y let be represented by $\mathrm{s}^{\alpha},\mathrm{s}^{\beta},\mathrm{s}^{\gamma},\mathrm{s}^{\delta},\mathrm{s}^{\varepsilon}$, &c. Then will be the equation $a+bx+cx^2+dx^3+ex^4\ldots.x^{n-1}=y=$

$$\frac{\overline{x-\beta}\times\overline{x-\gamma}\times\overline{x-\delta}\times\overline{x-\varepsilon}\times\text{\&c.}}{\overline{\alpha-\beta}\times\overline{\alpha-\gamma}\times\overline{\alpha-\delta}\times\overline{\alpha-\varepsilon}\times\text{\&c.}}\times\mathrm{s}^{\alpha}+\frac{\overline{x-\alpha}\times\overline{x-\gamma}\times\overline{x-\delta}\times\overline{x-\varepsilon}\times\text{\&c.}}{\overline{\beta-\alpha}\times\overline{\beta-\gamma}\times\overline{\beta-\delta}\times\overline{\beta-\varepsilon}\times\text{\&c.}}\times\mathrm{s}^{\beta}$$

$$+\frac{\overline{x-\alpha}\times\overline{x-\beta}\times\overline{x-\delta}\times\overline{x-\varepsilon}\times\text{\&c.}}{\overline{\gamma-\alpha}\times\overline{\gamma-\beta}\times\overline{\gamma-\delta}\times\overline{\gamma-\varepsilon}\times\text{\&c.}}\times\mathrm{s}^{\gamma}+\frac{\overline{x-\alpha}\times\overline{x-\beta}\times\overline{x-\gamma}\times\overline{x-\varepsilon}\times\text{\&c.}}{\overline{\delta-\alpha}\times\overline{\delta-\beta}\times\overline{\delta-\gamma}\times\overline{\delta-\varepsilon}\times\text{\&c.}}\times\mathrm{s}^{\delta}$$

$$+\frac{\overline{x-\alpha}\times\overline{x-\beta}\times\overline{x-\gamma}\times\overline{x-\delta}\times\text{\&c.}}{\overline{\varepsilon-\alpha}\times\overline{\varepsilon-\beta}\times\overline{\varepsilon-\gamma}\times\overline{\varepsilon-\delta}\times\text{\&c.}}\times\mathrm{s}^{\varepsilon}+\text{\&c.''}$$

Clearly, this is the formula of Lagrange. Waring then states a second theorem that can be derived from the one just formulated.

4. The period after Lagrange

4.1. The barycentric form of Lagrange's formula

The original Lagrange interpolation formula, written in modern form, is

$$f(x)=p_n(f;x)+R_n(f;x),\quad p_n(f;x)=\sum_{k=0}^{n}f(x_k)\ell_k(x), \tag{6}$$

where

$$\ell_k(x)=\prod_{\substack{j=0\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j},\quad k=0,1,\ldots,n, \tag{7}$$

are the elementary Lagrange interpolation polynomials (cf. §2). It is attractive more for theoretical than for computational purposes. Only relatively recently, in the mid to

[2] Probably François Regnaud; see [48, p. 664] (author's note).

late 1940s ([74, 17]), it was given another form — the barycentric form — which, as Rutishauser [64, §6.2] put it, "is more palatable for numerical computation". Indeed, its many advantages and extraordinary robustness have been emphazised recently by Berrut and Trefethen [5] and Higham [40].

Having in mind the situation where one is given an infinite sequence $x_0, x_1, x_2, \dots$ of points and seeks to successively, for $n = 1, 2, 3, \dots$, interpolate at the first $n+1$ points $x_0, x_1, \dots, x_n$, we introduce a triangular array of auxiliary quantities defined by

$$\lambda_0^{(0)} = 1, \ \lambda_k^{(n)} = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{1}{x_k - x_j}, \quad k = 0, 1, \dots, n; \ n = 1, 2, 3, \dots .$$

Then (7) can be written in the form

$$\text{(8)} \qquad \ell_k(x) = \frac{\lambda_k^{(n)}}{x - x_k} w_n(x), \ k = 0, 1, \dots, n; \quad w_n(x) = \prod_{j=0}^{n} (x - x_j).$$

Dividing $p_n(f;x)$ in (6) through by $1 \equiv \sum_{k=0}^{n} \ell_k(x)$ and using (8), one finds

$$\text{(9)} \qquad p_n(f;x) = \frac{\displaystyle\sum_{k=0}^{n} \frac{\lambda_k^{(n)}}{x - x_k} f_k}{\displaystyle\sum_{k=0}^{n} \frac{\lambda_k^{(n)}}{x - x_k}}, \quad x \neq x_k \text{ for } k = 0, 1, \dots, n.$$

This expresses the interpolation polynomial as a "weighted" average of function values $f_k = f(x_k)$ and is therefore called the barycentric formula. Note, however, that the "weights" are not necessarily all positive.

Comparison with (6), incidentally, reveals that

$$\text{(10)} \qquad \ell_k(x) = \frac{\dfrac{\lambda_k^{(n)}}{x - x_k}}{\displaystyle\sum_{j=0}^{n} \frac{\lambda_j^{(n)}}{x - x_j}}, \quad k = 0, 1, \dots, n.$$

It may be noted that in both formulae, (9) and (10), any factor of $\lambda_k^{(n)}$ that does not depend on k can be ignored since it cancels out in the numerator and denominator of (9) resp. (10). This can be used to advantage, for example, when $x_k = \cos(k\pi/n)$, and $\lambda_k^{(n)} = (-1)^k 2^{n-1}/n$, if $0 < k < n$, and half this value otherwise [66], so that $\lambda_k^{(n)}$ can be replaced by $(-1)^k$ resp. $\frac{1}{2}(-1)^k$.

4.2. Error and convergence

With regard to his interpolation process, Lagrange writes [67, v. 7, p. 284]: "Or il est clair que, quelle que puisse être la courbe proposée [the given function (or curve)], la courbe parabolique [the polynomial curve] ainsi tracée en différera toujours d'autant moins que le nombre des points donnés sera plus grand, et leur distance moindre," suggesting that the nth-degree interpolation polynomial converges to the correct value of the given function as n goes to infinity and the largest distance between consecutive interpolation points goes to zero. Yet, convergence is by no means guaranteed, as will be shown further on by a remarkable example of Runge.

Real-variable analysis

A serious study of the error of interpolation has been taken up only about 25 years after Lagrange's death. Cauchy [11], in 1840, investigates divided differences (called "fonctions interpolaires", following Ampère), particularly the effect of confluences, $[\underbrace{x_0,x_0,\ldots,x_0}_{n+1 \text{ times}}]f = f^{(n)}(x_0)/n!$. He then obtains the mean-value formula $[x_0,x_1,...,x_n]f = f^{(n)}(\xi)/n!$ with $\xi \in \operatorname{span}(x_0,x_1,\ldots,x_n)$, and the remainder term in Newton's formula (3), first in the form $[x_0,x_1,\ldots,x_n,x]f \cdot \prod_{k=0}^{n}(x-x_k)$, and then, using his mean-value formula for divided differences, in the form generally known today,

$$R_n(f;x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n) \tag{11}$$

with $\xi \in \operatorname{span}(x_0,x_1,\ldots,x_n,x)$. We note that (11) reduces to Lagrange's form of the remainder term in Taylor's expansion [67, v. 9, p. 84], which is the special case of (3) when $x_0 = x_1 = \cdots = x_n$ (cf. Hermite interpolation in §4.3).

The representation (11) of the remainder is useful only if sufficient information is known about higher derivatives of f. If we know only, for example, that f is continuous on the interval $[a,b]$ (containing the points $x_0,x_1,\ldots,x_n$), then another representation must be used that involves the best (uniform) approximation

$$\mathcal{E}_n = \min_{p\in\mathbb{P}_n} \|f-p\|_\infty = \|f-\hat{p}_n(f;\cdot)\|_\infty$$

of f on $[a,b]$ by polynomials of degree $\leq n$. Writing $R_n(f;x) = f(x) - p_n(f;x) = f(x) - \hat{p}_n(f;x) + \hat{p}_n(f;x) - p_n(f;x)$ and noting that $\hat{p}_n - p_n$, being a polynomial of degree $\leq n$, can be represented by Lagrange's formula, and $p_n(f;x_k) = f(x_k)$, one gets

$$R_n(f;x) = f(x) - \hat{p}_n(f;x) + \sum_{k=0}^{n} \ell_k(x)[\hat{p}_n(x_k) - f(x_k)].$$

By the triangle inequality, therefore,

$$\|R_n(f;\cdot)\|_\infty \leq (1+\Lambda_n)\mathcal{E}_n, \tag{12}$$

where

$$\Lambda_n = \|\lambda_n\|_\infty, \quad \lambda_n(x) = \sum_{k=0}^{n} |\ell_k(x)|$$

are respectively the Lebesgue constant and the Lebesgue function of interpolation (so named by the analogy with the Lebesgue constant in Fourier analysis [49, §8], [12, Ch. 4, §6, p. 147]).

Although $\mathcal{E}_n \to 0$ as $n \to \infty$, by Weierstrass's theorem, one cannot conclude from (12) the same for $\|R_n(f;\cdot)\|_\infty$ since Λ_n is unbounded, at least logarithmically, as $n \to \infty$. Faber [21] indeed has shown in 1914 that no matter how the interpolation points are selected on $[a,b]$, there always exists a continuous function $f \in C[a,b]$ for which Lagrange interpolation diverges. From Jackson's theorem in the theory of best uniform aproximation it is known, however, that in the case of Chebyshev points (for these, see Hermite-Fejér interpolation in §4.3), $\mathcal{E}_n(f)\log n \to 0$ as $n \to \infty$ whenever $f \in C^1[-1,1]$, implying uniform convergence in this case. The same holds for functions that are continuous and of bounded variation in the case of Jacobi points (zeros of the Jacobi polynomial $P_n^{(\alpha,\beta)}$) when $-1 < \max(\alpha,\beta) < 1/2$ [78].

An interesting problem is the determination of optimal nodes minimizing Λ_n over all possible choices of $n+1$ distinct nodes. While characterizations for optimal nodes, originally conjectured by Bernstein and Erdős, have been established by Kilgore [43, 44] and de Boor and Pinkus [8], explicit representations are not known, though good approximations for them (on $[-1,1]$) are the stretched Chebyshev points (stretched linearly so that the smallest and largest of the Chebyshev points come to lie respectively on -1 and $+1$); see [9, §2.3]. Even better nodes exist which are in agreement with the known estimate [79, eq. (2.11)]

$$\Lambda_n = \frac{2}{\pi}\left(\log n + \gamma + \log\frac{4}{\pi}\right) + o(1) \quad \text{as } n \to \infty,$$

where $\gamma = .577215\ldots$ is Euler's constant. The Lebesgue constant Λ_n also measures the stability of Lagrange interpolation with respect to small changes in the data. For many other interesting properties of Λ_n see [79, Sec. 2].

For convergence in the mean, there are more positive results. For example, if one interpolates a function f on $[-1,1]$ at the n zeros $x_k^{(n)}$, $k = 1,2,\ldots,n$, of a polynomial π_n orthogonal (relative to a weight function w) on the interval $[-1,1]$, then one has the classical result of Erdős and Turán [18] that the process converges in the mean for any continuous function f, i.e.,

$$\|f - p_n(f;\cdot)\|_w \to 0 \quad \text{as } n \to \infty, \text{ for all } f \in C[-1,1], \tag{13}$$

where $\|u\|_w^2 = \int_{-1}^1 u^2(x)w(x)dx$. For a survey of results on mean convergence, see [72].

An intriguing and largely open problem (see, however, [53, §4.2.3.3]) is extended Lagrange interpolation, that is, interpolation by a polynomial of degree $\le 2n$ at the n zeros $x_k^{(n)}$ of π_n and $n+1$ additional points τ_j, $j = 1,2,\ldots,n+1$, suitably interlaced with the $x_k^{(n)}$. The most natural choice for the τ_j would be the zeros of the orthogonal polynomial π_{n+1}, but nothing seems to be known about convergence in the mean in this case. Another choice suggested by Bellen [2] are the zeros of the polynomial of degree $n+1$ orthogonal on $[-1,1]$ relative to the weight function $\pi_n^2 w$. Here sufficient conditions are known for convergence in the mean [2] and have been studied in [30]. For a related problem of "quadrature convergence", see also [34].

Complex-variable analysis

An analogue of (11) for analytic functions f has been obtained by Hermite [39] (or [62, pp. 432–443]) in 1878, even in the more general case of Hermite interpolation (cf. §4.3). Assuming f analytic in a domain $\mathcal{D}$ in the complex plane that contains all the points $x_0, x_1, \dots, x_n$ in its interior, and defining

$$w_n(z) = \prod_{k=0}^{n} (z - x_k), \quad z \in \mathcal{D},$$

there holds

$$R_n(f;z) = \frac{1}{2\pi i} \int_\Gamma \frac{w_n(z)}{w_n(\zeta)} \frac{f(\zeta)}{\zeta - z} d\zeta, \tag{14}$$

where Γ is a contour in $\mathcal{D}$ containing all the x_k and z.

For equidistant points $x_k^{(n)} = a + k(b-a)/n$, $k = 0, 1, \dots, n$, in $[a,b]$, Runge [63, p. 243] used (14) to obtain a family of concentric contours Γ_ρ, $\rho > 0$, and domains $\mathcal{D}_\rho$ bounded by them,

$$\Gamma_\rho = \{z \in \mathbb{C}:\ \sigma(z) = \rho\}, \quad \mathcal{D}_\rho = \{z \in \mathbb{C}:\ \sigma(z) \le \rho\}, \tag{15}$$

where

$$\sigma(z) = \exp\left\{ \frac{1}{b-a} \int_a^b \ln|z - t| dt \right\}, \tag{16}$$

with the property: if $f \in A(\mathcal{D}_\rho)$ is analytic in $\mathcal{D}_\rho$, then $R_n(f;z) \to 0$ as $n \to \infty$ for any z in the interior of $\mathcal{D}_\rho$, and if $\rho = \rho^*$ is the largest value of ρ for which $f \in A(\mathcal{D}_\rho)$, then $\sup |R_n(f;z)| = \infty$ for z outside of $\mathcal{D}_{\rho^*}$.

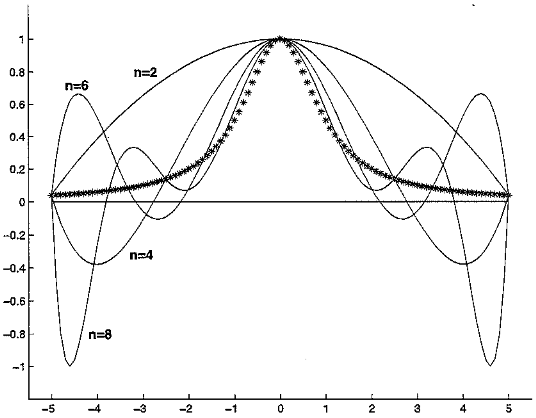


Figure 1: Runge's example

The example given by Runge is the function

$$f(x) = \frac{1}{1+x^2}, \quad -5 \le x \le 5,$$

and the $x_k^{(n)}$ equally spaced on $[-5,5]$,

$$x_k^{(n)} = -5 + k\frac{10}{n}, \quad k = 0,1,2,\dots,n.$$

What one finds, when one tries to interpolate f for $n = 2,4,6,\dots$ by polynomials p_n of degree n, is shown in Fig. 1. The approximation provided by p_n, as n increases, becomes better and better in the central part of the interval $[-5,5]$, but worse and worse in the lateral parts of the interval.

Runge, in fact, proves that for real x convergence takes place if $|x| < x^*$ and divergence if $|x| > x^*$, where $x^* = 3.633\dots$. One can show that $x^* = 3.633384302388339\dots$ is the solution of the transcendental equation

$$(5+x)\ln(5+x) + (5-x)\ln(5-x) = 10 + 5\int_0^1 \ln(1+25t^2)dt.$$

The domains $\mathcal{D}_\rho$ for this example are shown in Fig. 2 for several values of ρ determined in such a way that Γ_ρ intersects the positive real axis resp. at x^*, 5, 6–8. Note that the first of them, $\mathcal{D}_{\rho^*}$, must pass through the poles $\pm i$. It bounds the largest domain $\mathcal{D}_\rho$ inside of which f is analytic. The interpolation process therefore converges inside this domain but not outside, which explains the behavior shown in Fig. 1. The next domain, the smallest one containing the interval $[-5,5]$, is a critical domain, denoted here by $\mathcal{D}_{\rho_0}$. It has the property that if f is analytic in any domain $\mathcal{D}$ that contains $\mathcal{D}_{\rho_0}$ in

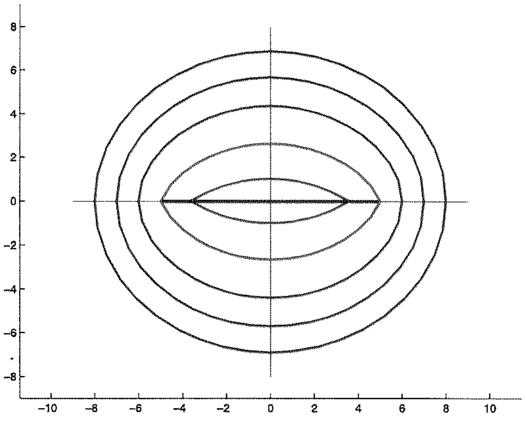


Figure 2: Runge's convergence domains $\mathcal{D}_\rho$

its interior, but can be arbitrarily close to $\mathcal{D}_{\rho^0}$, and if all points $x_k^{(n)}$ are in $\mathcal{D}$, then $p_n(f;z) \to f(z)$ for any $z \in \mathcal{D}$.

If the points $x_k^{(n)}$ are not necessarily equally distributed, but have a limit distribution $d\mu$ on a finite interval $[a,b]$, i.e. $\int_a^x d\mu(t)$, $a < x \le b$, is the ratio of the number of points $x_k^{(n)}$ in $[a,x]$ to the total number, $n+1$, of points, as $n \to \infty$, then Runge's theory remains valid if dt in (16) is replaced by $d\mu(t)$. (Cf., e.g., Krylov [46, Ch. 12, Sect. 2].) For example, in the case of the arcsin-distribution $d\mu(t) = \dfrac{1}{\pi}\dfrac{dt}{(1-t^2)^{1/2}}$ on $[-1,1]$, typical for points $x_k^{(n)}$ that are zeros of an orthogonal polynomial of degree n, one finds that $\mathcal{D}_{\rho^0} = [-1,1]$; cf. Fig. 3. Thus, in this case, Lagrange interpolation converges for any function f analytic on $[-1,1]$. Figure 3 shows the domains $\mathcal{D}_\rho$ that intersect the real axis at 2, 1.5, 1.1, 1.01 and correspond to the ρ-values 1.36602..., 1.14412..., .88268..., and .75887... .

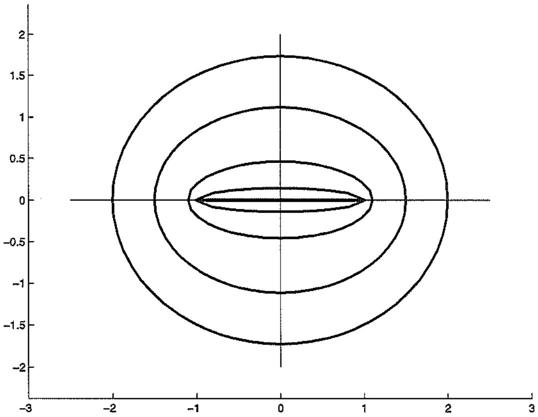


Figure 3: Convergence domains $\mathcal{D}_\rho$ for points with an arcsin-distribution

4.3. Hermite and Hermite–Fejér interpolation

Hermite interpolation

The Hermite interpolation problem consists in obtaining the polynomial of lowest degree that interpolates not only to the function values $f(x_k)$ at the (distinct) points x_k, but also to the successive derivatives $f^{(\mu)}(x_k), \mu = 1,2,\ldots,m_k-1, k = 1,2,\ldots,K$. Hermite in [39] expresses the polynomial (of degree $n = m_1 + m_2 + \cdots + m_K - 1$) as a sum of residues of a contour integral in the complex plane. In practice, it can be obtained as a limit case of Newton's formula, by setting up a table of divided differences in which each point x_k is listed m_k times and Cauchy's limit formulae for confluent divided differences (cf. Real-variable analysis in §4.2) are used, when $m_k > 1$, to initialize the table. All remaining divided differences are computed by the same rules as in Newton's table of divided differences. The remainder (cf. (11)), accordingly, will assume the form

$$R_n(f;x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=1}^{K} (x-x_k)^{m_k}. \tag{17}$$

For $K = 1$, the Hermite interpolation polynomial becomes Taylor's polynomial of degree $m_1 - 1$ and (17) Lagrange's form of the remainder.

A large body of literature is devoted to lacunary Hermite interpolation, also called Birkhoff interpolation, where the derivatives $f^{(\mu)}(x_k)$ involved are not necessarily for successive values of $\mu = 1,2,3,\ldots$, but for an arbitrary sequence of μ-values. For this, see, e.g., [50]. An important topic here is the study of existence and uniqueness, which is no longer guaranteed.

Hermite–Fejér interpolation

Soon after Faber's negative result (cf. Real-variable analysis in §4.2) was published, Fejér asked himself whether a polynomial interpolation process exists which would converge for any continuous function on a finite interval. This led him to consider the special case of Hermite interpolation in which $m_k = 2$ and $p'(x_k) = 0$, $k = 1,2,\ldots,n$, now called Hermite-Fejér interpolation. The corresponding polynomial p of degree $2n-1$ can be represented similarly as in the Lagrange formula (6) (with n replaced by $2n-1$), involving elementary Hermite-Fejér polynomials $h_k(x)$ of degree $2n-1$ in place of the $\ell_k(x)$. The case of Chebyshev points $x_k = \cos((2k-1)\pi/(2n))$ is again very favorable. Fejér [22] (or [77, pp. 25–48]), in 1916, indeed proved that, in contrast to Lagrange interpolation, the interpolation error tends to zero uniformly on $[-1,1]$ for any function continuous on $[-1,1]$.

The Hermite-Fejér interpolation process has since been studied from a number of different angles. One is to look at points other than Chebyshev points. Fejér himself already considered what he calls Gauss points (the zeros of the Legendre polynomial P_n). For $f \in C[-1,1]$ he proves uniform convergence to f on any compact subinterval of $[-1,1]$ and convergence to $\frac{1}{2}\int_{-1}^{1} f(x)dx$ at ± 1. Szegő considers more generally the zeros of the nth-degree Jacobi polynomial with parameters $\alpha,\beta > -1$. He obtains [73, Theorem 14.6], [71, Ch. V, §1.2] uniform convergence to f for arbitrary $f \in C[-1,1]$ whenever $-1 < \alpha,\beta < 0$ but not if $\max(\alpha,\beta) \geq 0$. Grünwald [37] proves the same for another class of points, called ρ-normal, when $\rho > 0$ (cf. [71, Ch. 5, §1.3]). Error estimates are discussed in [71, Ch. 5, §2], and convergence when the function f is suitably restricted, in [71, Ch. 5, §3]. Comparisons with Lagrange interpolation are made in [71, Ch. 6]. For a comprehensive review of the literature on Hermite–Fejér interpolation up to 1987, see [36].

One talks of Hermite–Fejér type interpolation when all derivatives up to some order m are required to vanish at the interpolation points. If these are again Chebyshev points, uniform convergence on $[-1,1]$ for all continuous functions f then holds whenever m is odd, but not for m even ([79, Sec. 4.5, Remark 6]). The convergence behavior for m odd and m even indeed is similar to respectively Hermite–Fejér and Lagrange interpolation [70].

4.4. Other interpolants and extensions

Functions other than polynomials can of course also serve as interpolants, for example rational functions, trigonometric polynomials, and sinc functions. In each of these cases there exists a barycentric form of the interpolant; for rational functions, see [6], for trigonometric polynomials, [38] when the points x_k are equally spaced and [65, 3] otherwise, and for cardinal sinc-interpolation [4]. Piecewise polynomial interpolants (spline functions, especially cubic splines) are popular in computer-aided geometric design and are treated extensively in [7]. In the area of signal and image processing, a number of techniques are in use, collectively named convolution-based interpolation, for which we refer to [54, Sec. IV].

An important extension is multivariate Lagrange interpolation. This also has a long history [25, 26] and is a topic currently receiving renewed attention [61]. For multivariate Hermite and Birkhoff interpolation, see [52, 51].

4.5. Applications

Quadrature

Newton–Cotes quadrature. The idea of approximating a definite integral of a function f by replacing f by an interpolating polynomial $p_n(f;\cdot)$ and integrating the polynomial p_n instead of f goes back to Newton (cf. §3.2) and has been implemented numerically by Roger Cotes, a protégé of Newton, who passed away at a young age. It provides here a first opportunity to apply Lagrange's interpolation formula.

Considering a general weighted integral $\int_{\mathbb{R}} f(x)d\lambda(x)$, where typically $d\lambda(x) = w(x)dx$ and w is a positive weight function supported on a finite or infinite interval, we integrate (6) (with n instead of $n+1$ points) to get

$$\int_{\mathbb{R}} f(x)d\lambda(x) = \sum_{k=1}^{n} \lambda_k f(x_k) + E_n(f), \tag{18}$$

where

$$\lambda_k = \lambda_k^{(n)} = \int_{\mathbb{R}} \ell_k(x)d\lambda(x), \quad \ell_k(x) = \prod_{\substack{j=1 \\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j}, \quad k=1,2,\ldots,n, \tag{19}$$

and $E_n(f) = \int_{\mathbb{R}} R_n(f;x)d\lambda(x)$ is the remainder term. This is called a weighted Newton–Cotes formula, the classical Newton–Cotes formula being the special case with $d\lambda(x) = dx$ on $[-1,1]$ and x_k equally spaced on $[-1,1]$ with $x_1 = 1$ and $x_n = -1$. In this case, Cotes around 1712 published the respective coefficients $\lambda_k^{(n)}$ for $n \leq 11$. For much larger values of n, these formulae become numerically unattractive because of the coefficients $\lambda_k^{(n)}$ becoming large and oscillatory in sign.

It is of interest, therefore, to know of integration measures $d\lambda$ and/or quadrature points x_k which give rise to weighted Newton–Cotes formulae (18) whose coefficients λ_k are all nonnegative. Here again, in the case $d\lambda(x) = dx$ on $[-1,1]$, the Chebyshev points (of the first kind) come to our aid. In this case, and also in the case of Chebyshev points of the second kind, giving rise to what are called Filippi rules, Fejér [23] has shown that for all n the coefficients are positive. The same is true for Clenshaw–Curtis rules [15] whose points x_k are the extreme points (including ± 1) of the Chebyshev polynomial T_{n-1}; the positivity of the quadrature weights in this case has been proved by Imhof [41].

To compute weighted Newton–Cotes formulae, it is convenient to use Gaussian quadrature relative to the measure $d\lambda$ (see below) in (19) combined with the barycentric form (10) of $\ell_k(x)$; see, e.g., Machine Assignment 3 in [33, Ch. 3, pp. 215–217] and its solution therein on pp. 232–247 for some relevant experimentation with x_k the zeros of a Jacobi polynomial and $d\lambda$ a Jacobi measure relative to a different pair of Jacobi parameters. In this case, quasi-positivity, i.e.,

$$\lim_{n\to\infty} \sum_{\substack{k=1 \\ \lambda_k<0}}^{n} |\lambda_k^{(n)}| = 0,$$

has also been studied in [80].

Gaussian quadrature. By construction, n-point Newton–Cotes formulae have polynomial degree of exactness $n-1$, i.e., are exact whenever f is a polynomial of degree $n-1$. As Gauss [27] discovered in 1816, in the case of $d\lambda(x) = dx$, the degree of exactness can be made as large as $2n-1$ (which is optimal when $d\lambda$ is a positive measure) by taking the nodes x_k to be the zeros of the nth-degree orthogonal polynomial π_n relative to the measure $d\lambda$, and the weights λ_k such as to make the formula "interpolatory". The resulting formula has been called the Gauss–Christoffel formula [28], since Christoffel [13] in 1877 generalized Gauss's original formula to arbitrary positive weight functions. What is nice about these formulae is that all weights λ_k are positive, as has been shown very elegantly by Stieltjes [69, pp. 384–385].

The standard way nowadays of computing the n-point Gauss–Christoffel formula is by eigenvalue/vector techniques based on the Jacobi matrix of order n [31, Sec. 3.1.1.1], wich is defined in terms of the recurrence coefficients of the respective orthogonal polynomials. If these are not known explicitly, there are various methods available to compute them numerically [31, Ch. 2].

If $d\lambda$ is supported on a finite interval $[a,b]$, minor extensions of the Gauss–Christoffel formula are the Gauss–Radau and Gauss–Lobatto formulae, having one or both of the end points of $[a,b]$ as nodes. A more substantial extension is the $(2n+1)$-point Gauss–Kronrod formula whose nodes x_k are the n Gauss–Christoffel nodes together with $n+1$ additional real nodes chosen so as to achieve maximum degree of exactness [31, Sec. 3.1.2]. This poses intriguing problems of existence and challenges with regard to constructive methods. For reviews of Gauss–Kronrod quadrature, see Monegato [55, 56, 57], Gautschi [29], and Notaris [60].

Multistep methods in ordinary differential equations

Quadrature, and therefore interpolation, is implicit also in the numerical solution of initial value problems for ordinary differential equations,

$$(20) \qquad y' = f(x,y), \quad y(0) = y_0,$$

especially when multistep methods are being used. Basically, one has information (given by f) about the derivative of a real- or vector-valued function y and wants to obtain from this some information about the function itself. In particular, one may want to know how to advance the solution y from some point x to another point $x+h$, where $h > 0$ is a small increment. An answer to this is provided by the fundamental theorem of calculus,

$$(21) \qquad y(x+h) = y(x) + \int_x^{x+h} y'(\xi)d\xi.$$

Now suppose we already know the values of the derivative at k points $x, x-h, x-2h, \dots, x-(k-1)h$, where $k > 1$. Then we may approximate the integral in (21) by passing through the points $y'_s = y'(x-sh)$, $s = 0, 1, \dots, k-1$, a polynomial of degree $k-1$ and then (following Newton!) integrate the polynomial instead of y'. Using the Lagrange interpolation formula, one gets

$$(22) \qquad y(x+h) \approx y(x) + h \sum_{s=0}^{k-1} \beta_{k,s} y'_s,$$

where

$$\beta_{k,s} = \int_0^1 \prod_{\substack{r=0 \\ r \neq s}}^{k-1} \left(\frac{t+r}{r-s} \right) dt, \quad s = 0, 1, \dots, k-1.$$

To develop this into a computational algorithm, we take a grid $x_n = nh$, $n = 0, 1, 2, \dots$, and denote approximations to $y(x_n)$ by u_n. Then formula (22) suggests the following method,

$$(23) \qquad u_{n+1} = u_n + h \sum_{s=0}^{k-1} \beta_{k,s} f_{n-s}, \quad n = k-1, k, k+1, \dots,$$

where

$$f_m = f(x_m, u_m), \quad m = 0, 1, 2, \dots,$$

and it is assumed that the first $k-1$ approximations $u_1, u_2, \dots, u_{k-1}$ are obtained in some other way. (Of course, $u_0 = y_0$.)

Had we used Newton's interpolation formula rather than Lagrange's, we would have obtained the difference form of the method,

$$(24) \qquad u_{n+1} = u_n + h \sum_{s=0}^{k-1} \gamma_s \nabla^s f_n, \quad n = k-1, k, k+1, \dots,$$

where

$$\gamma_s = \int_0^1 \binom{t+s-1}{s} dt, \quad s = 0,1,\ldots,k-1,$$

and $\nabla f_n = f_n - f_{n-1}$, $\nabla^2 f_n = \nabla(\nabla f_n), \ldots$ are the backward differences of f.

What we have obtained are the explicit k-step Adams–Bashforth method in the Lagrange resp. Newton form, published in 1883 by the astronomer John Couch Adams in the book [1]. There is a companion method—the Adams–Moulton method—which on the right-hand sides of (23) and (24) involves also f_{n+1}, and therefore is an implicit method requiring the solution of a nonlinear equation resp. a system of nonlinear equations. Its derivation, as well as the derivation of many similar multistep methods, uses the same techniques as outlined above.

Lagrangian bases in finite element methods

Classical finite element methods for approximating the solution of a boundary value problem use piecewise polynomial functions over a partition of the domain, with suitable inter-element continuity. For instance, a second-order self-adjoint problem is formulated in a variational (or weak) form, which after "integration by parts" involves only integrals of functions and their first derivatives. Then a corresponding discrete variational formulation is developed by selecting a finite-dimensional space of trial and test functions. Typically, these are polynomials within each element, glued together to form globally C^0 functions.

This paradigmatic scenario suggests the importance of having simple and efficient ways to "glue together" pieces of polynomials, usually having the same local degree. Here, the concept of Lagrange interpolation plays a natural role, leading to the construction of Lagrangian bases in the spaces of trial and test functions.

The simplest example, in one dimension, is given by piecewise linear interpolation on the partition of an interval $[a,b]$ defined by nodes $a = x_0 < x_1 < \cdots < x_\nu < \cdots < x_{n-1} < x_n = b$. Here the basis functions are the elementary Lagrange piecewise linear interpolation functions—the "hat" functions

$$u_\nu(x) = \begin{cases} \dfrac{x - x_{\nu-1}}{x_\nu - x_{\nu-1}} & \text{if } x_{\nu-1} \le x \le x_\nu\,, \\[2ex] \dfrac{x_{\nu+1} - x}{x_{\nu+1} - x_\nu} & \text{if } x_\nu \le x \le x_{\nu+1}\,, \\[2ex] 0 & \text{elsewhere}\,, \end{cases} \qquad \nu = 0,1,\ldots,n,$$

which like the polynomial counterpart $\ell_\nu(x)$ in (7) satisfy $u_\nu(x_\mu) = \delta_{\nu\mu} = 0$ if $\nu \neq \mu$ and $= 1$ otherwise (cf. Fig. 4). Note that inter-element continuity of the piecewise linear interpolant $p_n(f;\cdot)$ is trivially guaranteed by the fact that the interpolation nodes are the common points between consecutive elements $[x_{\nu-1}, x_\nu]$ and $[x_\nu, x_{\nu+1}]$.

Errors in finite element methods are naturally measured in Sobolev norms, such

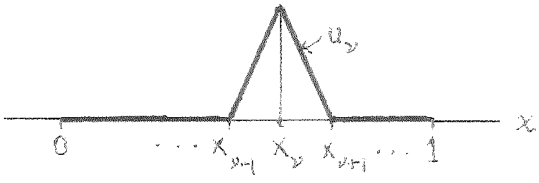


Figure 4: The hat function

as $|f|_m = \left(\int_a^b |f^{(m)}(x)|^2\,\mathrm{d}x\right)^{1/2}$, $m \geq 0$. If $|f|_2$ is finite, one easily proves that

(25) $$|f - p_n(f;\cdot)|_m \leq Ch^{2-m}|f|_2, \quad m = 0,1,$$

where $h = \max_\nu(x_\nu - x_{\nu-1})$. Higher-order approximations are achieved by increasing the polynomial degree in each element.

In two dimensions, the analogue of the previous example is obtained by introducing a triangulation on the domain and requiring that the interpolant restricted to each triangle is an affine function $a + bx + cy$. Inter-element C^0-continuity is naturally achieved by choosing for the freely variable entities (to be subject to linear conditions to uniquely identify the interpolant) the values at all vertices of the triangulation. Indeed, the restrictions of two affine functions to an edge shared by two triangles coincide if and only if they coincide at the end points of the edge. The Lagrangian basis functions u_ν are now "pyramidal" functions associated with the vertices of the triangulation, each

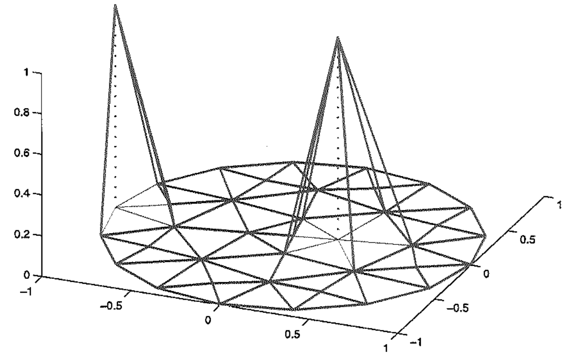


Figure 5: The pyramidal functions

of them vanishing identically outside the patch of triangles sharing the corresponding vertex (cf. Fig. 5). The element just described is called the Courant element, since its origin dates back to a 1943 paper by R. Courant [16], where it was used in the numerical treatment of a torsion problem.

In general, a Lagrange finite element method is such that all variable entities are values of the approximating functions at a selected set of nodes in the domain. The corresponding Lagrangian basis is obtained by setting to zero all but one of these values. If some of the entities involve certain derivatives of the aproximating functions, one speaks of a Hermite finite element method.

One of the major properties of Lagrangian bases is that they have minimum support (within the linear space of all continuous piecewise polynomial functions over the given domain mesh). Therefore, in the case of linear problems, for example, the final linear systems generated by the finite element methods have maximum degree of sparsity. Moreover, also the numerical evaluation of all the integrals needed to define these linear systems is greatly simplified when one uses such bases.

The mathematical analysis of the finite element method dates back to the late 1960s. The 1972 paper by P. G. Ciarlet and P.-A. Raviart [14] was one of the most influential in the field, since it provided error estimates similar to (25) in a very general setting and with a unifying approach.

Acknowledgements. The author is indebted to Claudio Canuto and Giovanni Monegato for contributing the section Lagrangian bases in finite element methods of §4.5, to Giovanni Monegato for Fig. 5, and to Peter Vértesi for helpful comments to the section Hermite-Fejér interpolation of §4.3.

References

[1] BASHFORTH, FRANCIS AND ADAMS, J. C. *An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops*, Cambridge University Press, Cambridge, 1883.

[2] BELLEN, ALFREDO. Alcuni problemi aperti sulla convergenza in media dell'interpolazione Lagrangiana estesa, *Rend. Istit. Mat. Univ. Trieste* 20 (1988), fasc. suppl., 1–9.

[3] BERRUT, JEAN-PAUL. Baryzentrische Formeln zur trigonometrischen Interpolation I., *Z. Angew. Math. Phys.* 35 (1984), 91–105; Baryzentrische Formeln zur trigonometrischen Interpolation II. Stabilität und Anwendung auf die Fourieranalyse bei ungleichabständigen Stützstellen, *ibid.* 193–205.

[4] BERRUT, JEAN-PAUL. Barycentric formulae for cardinal (SINC-)interpolants, *Numer. Math.* 54 (1989), 703–718. [Erratum: *ibid.* 55 (1989), 747.]

[5] BERRUT, JEAN-PAUL AND TREFETHEN, LLOYD N. Barycentric Lagrange interpolation, *SIAM Rev.* 46 (2004), 501–517.

[6] BERRUT, JEAN-PAUL; BALTENSPERGER, RICHARD; AND MITTELMANN, HANS D. Recent developments in barycentric rational interpolation. Trends and applications in constructive approximation, 27–51, *Internat. Ser. Numer. Math.* 151, Birkhäuser, Basel, 2005.

[7] DE BOOR, CARL. *A practical guide to splines*, rev. ed., Applied Mathematical Sciences 27, Springer, New York, 2001.

[8] DE BOOR, CARL AND PINKUS, ALLAN. Proof of the conjectures of Bernstein and Erdős concerning the optimal nodes for polynomial interpolation, *J. Approx. Theory* 24 (1978), 289–303.

[9] BRUTMAN, L. Lebesgue functions for polynomial interpolation—a survey, The heritage of P. L. Chebyshev: a Festschrift in honor of the 70th birthday of T. J. Rivlin, *Ann. Numer. Math.* 4 (1997), 111–127.

[10] CAUCHY, A. *Œuvres complètes d'Augustin Cauchy*, ser. 1, vols. 1–12 (1882–1900), ser. 2, vols. 1–15 (1905–1974), Académie des Sciences, Gauthier-Villars, Paris.

[11] CAUCHY, A. Sur les fonctions interpolaires, (1840), in [10, ser. 1, v. 5, 409–424].

[12] CHENEY, E. W. *Introduction to approximation theory*, Reprint of the second (1982) edition, AMS Chelsea Publishing, Providence, RI, 1998.

[13] CHRISTOFFEL, E. B. Sur une classe particulière de fonctions entières et de fractions continues, *Ann. Mat. Pura Appl.* (2)8(1877), 1–10. [Also in *Ges. Math. Abhandlungen* II, 42–50.]

[14] CIARLET, P. G. AND RAVIART, P.-A. General Lagrange and Hermite interpolation in R^n with applications to finite element methods, *Arch. Rational Mech. Anal.* 46 (1972), 177–199.

[15] CLENSHAW, C. W. AND CURTIS, A. R. A method for numerical integration on an automatic computer, *Numer. Math.* 2 (1960), 197–205.

[16] COURANT, R. Variational methods for the solution of problems of equilibrium and vibrations, *Bull. Amer. Math. Soc.* 49 (1943), 1–23.

[17] DUBUY, MICHEL. Le calcul numérique des fonctions par l'interpolation barycentrique, *C. R. Acad. Sci. Paris* 226 (1948), 158–159.

[18] ERDŐS, P. AND TURÁN. P. On interpolation.I. Quadrature- and mean-convergence in the Lagrange-interpolation, *Ann. of Math.* 38 (1937), 142–155.

[19] EULER, LEONHARD. *Opera omnia*, ser. 1, v. 1–29 (1911–1956), ser. 2, v. 1–30 (1912–1964), ser. 3, v. 1–12 (1926–1960), B. G. Teubner, Leipzig and Berlin, and Orell Füssli, Zürich.

[20] EULER, LEONHARD. Consideratio quarumdam serierum quae singularibus proprietatibus sunt praeditae, *Novi Comment. Acad. Sci. Petropolitanae* 3 (1750/51) 1753, 10–12, 86–108. [An English translation of this memoir can be downloaded from the E190 page of the Euler Archive at `http://www.math.dartmouth.edu/~euler`.]

[21] FABER, GEORG. Über die interpolatorische Darstellung stetiger Funktionen, Jahresber. Deutsch. Math.-Verein 23 (1914), 192–210.

[22] FEJÉR, L. Über Interpolation, *Göttinger Nachr.* (1916), 66–91.

[23] FEJÉR, L. Mechanische Quadraturen mit positiven Cotesschen Zahlen, *Math. Z.* 37 (1933), 287–309.

[24] FRASER, D. C. *Newton's interpolation formulas*, C. & E. Layton, London. 1927.

[25] GASCA, MARIANO AND SAUER, THOMAS. Polynomial interpolation in several variables. Multivariate polynomial interpolation, *Adv. Comput. Math.* 12 (2000), 377-410.

[26] GASÇA, MARIANO AND SAUER, THOMAS. On the history of multivariate polynomial interpolation. Numerical analysis 2000, Vol. II: Interpolation and extrapolaton, *J. Comput. Appl. Math.* 122 (2000), 23–35.

[27] GAUSS, C. F. Methodus nova integralium valores per approximationem inveniendi, *Commentationes Societatis Regiae Scientarium Gottingensis Recentiores* 3 (1816). [Also in *Werke* III, 163–196.]

[28] GAUTSCHI, WALTER. A survey of Gauss–Christoffel quadrature formulae, in *E. B. Christoffel—the influence of his work in mathematics and the physical sciences* (P. L. Butzer and F. Fehér, eds.), 72–147, Birkhäuser, Basel, 1981.

[29] GAUTSCHI, WALTER. Gauss–Kronrod quadrature—a survey, in *Numerical methods and approximation III (Niš, 1987)*, 39–66, Univ. Niš, Niš, 1988.

[30] GAUTSCHI, WALTER. On mean convergence of extended Lagrange interpolation, *J. Comput. Appl. Math.* 43 (1992), 19–35.

[31] GAUTSCHI, WALTER. *Orthogonal polynomials: computation and approximation*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2004.

[32] GAUTSCHI, WALTER. On Euler's attempt to compute logarithms by interpolation: a commentary to his letter of February 16, 1734 to Daniel Bernoulli, *J. Comput. Appl. Math.* 219 (2008), 408–415.

[33] GAUTSCHI, WALTER. *Numerical analysis*, 2d ed., Birkhäuser, New York, 2012.

[34] GAUTSCHI, WALTER AND LI, SHIKANG. On quadrature convergence of extended Lagrange interpolation, *Math. Comp.* 65 (1996), 1249–1256.

[35] GOLDSTINE, H. H. *A history of numerical analysis from the 16th through the 19th century*, Springer, Berlin, 1977.

[36] GONSKA, H. H. AND KNOOP, H.-B. On Hermite–Fejér interpolation: a bibliography (1914–1987), *Studia Sci. Math. Hungar.* 25 (1990), 147–198.

[37] GRÜNWALD, G. On the theory of interpolation, *Acta Math.* 75 (1942), 219–245.

[38] HENRICI, PETER. Barycentric formulas for interpolating trigonometric polynomials and their conjugates, *Numer. Math.* 33 (1979), 225–234.

[39] HERMITE, CH. Sur la formule d'interpolation de Lagrange, *J. Reine Angew. Math.* 84 (1878), 70–79.

[40] HIGHAM, NICHOLAS J. The numerical stability of barycentric Lagrange interpolation, *IMA J. Numer. Anal.* 24 (2004), 547–556.

[41] IMHOF, J. P. On the method for numerical integration of Clenshaw and Curtis, *Numer. Math.* 5 (1963), 138–141.

[42] JOFFE, S. A. Interpolation-formulae and central-difference notation, *Trans. Actuar. Soc. Amer.* 18 (1917), 72–98.

[43] KILGORE, T. A. Optimization of the norm of the Lagrange interpolation operator, *Bull. Amer. Math. Soc.* 83 (1977), 1069–1071.

[44] KILGORE, THEODORE A. A characterization of the Lagrange interpolating projection with minimal Tchebycheff norm, *J. Approx. Theory* 24 (1978), 273–288.

[45] KOELINK, ERIK AND VAN ASSCHE, WALTER. Leonhard Euler and a q-analogue of the logarithm, *Proc. Amer. Math. Soc.* 137 (2009), 1663–1676.

[46] KRYLOV, VLADIMIR IVANOVICH. *Approximate calculation of integrals* [translated from the Russian by Arthur H. Stroud], ACM Monograph Ser., Macmillan, New York, 1962.

[47] LAGRANGE, J. -L. Sur les interpolations, (1778), in [67, v. 7, 535–553].

[48] LAGRANGE, J. -L. La méthode d'interpolation, (1792/93), in [67, v. 5, 663–684].

[49] LEBESGUE, HENRI. Sur la représentation trigonométrique approchée des fonctions satisfaisant à une condition de Lipschitz, *Bull. Soc. Math. France* 38 (1910), 184–210. [Also in *Œuvres Scientifiques*, III, 363–389.]

[50] LORENTZ, GEORGE G., JETTER, KURT, AND RIEMENSCHNEIDER, SHERMAN D. *Birkhoff interpolation*, Encyclopedia of Mathematics and its Applications, 19, Addison Wesley Publ. Co., Reading, MA, 1983.

[51] LORENTZ, RUDOLPH A. Multivariate Birkhoff interpolation, *Lecture Notes in Mathematics* 1516, Springer, Berlin, 1992.

[52] LORENTZ, R. A. Multivariate Hermite interpolation by algebraic polynomials: a survey. Numerical analysis 2000, Vol. II: Interpolation and extrapolation, *J. Comput. Appl. Math.* 122 (2000),167–201.

[53] MASTROIANNI, GIUSEPPE AND MILOVANOVIĆ, GRADIMIR V. *Interpolation processes: basic theory and applications*, Springer Monographs in Mathematics, Springer, Berlin, 2008.

[54] MEIJERING, ERIK. A chronology of interpolation: from ancient astronomy to modern signal and image processing, *Proc. IEEE* 90(3)(2002), 319–342.

[55] MONEGATO, GIOVANNI. An overview of results and questions related to Kronrod schemes, in G. Hämmerlin, ed., *Numerische Integration, Internat. Ser. Numer. Math.* 45, 231–240, Birkhäuser Basel, 1979.

[56] MONEGATO, GIOVANNI. Stieltjes polynomials and related quadrature rules, *SIAM Rev.* 24 (1982), 137–158.

[57] MONEGATO, GIOVANNI. An overview of the computational aspects of Kronrod quadrature rules, *Numer. Algorithms* 26 (2001), 173–196.

[58] DE MORGAN, A. *The differential and integral calculus*, Baldwin and Cradock, London, 1842.

[59] NEWTON, I. *Philosophiæ naturalis principia mathematica* (1687). [English translation in *Sir Isaac Newton's mathematical principles of natural philosophy and his system of the world* (Florian Cajori, ed.), University of California Press, Berkeley, CA, 1946.]

[60] NOTARIS, SOTIRIOS E. An overview of results on the existence or nonexistence and the error term of Gauss–Kronrod quadrature formulae, in R. V. W. Zahar, ed., *Approximation and computation (West Lafayette, IN, 1993), Internat. Ser. Numer. Math.* 119, 485–496, Birkhäuser Boston, 1994.

[61] OLVER, PETER J. On multivariate interpolation, *Stud. Appl. Math.* 116 (2006), 201–240.

[62] PICARD, ÉMILE, ed. *Œuvres de Charles Hermite*, v. 3, Gauthier-Villars, Paris, 1912.

[63] RUNGE, CARL. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, *Z. Math. und Phys.* 46 (1901), 224–243.

[64] RUTISHAUSER, HEINZ. *Vorlesungen über numerische Mathematik*, vol. 1: Gleichungssysteme, Interpolation und Approximation, Birkhäuser, Basel, 1976. [Annotated English translation in *Lectures on numerical mathematics*, Birkhäuser, Boston, MA, 1990.]

[65] Salzer, Herbert E. Coefficients for facilitating trigonometric interpolation, *J. Math. Physics* 27 (1949), 274–278.

[66] Salzer, H. E. Lagrangian interpolation at the Chebyshev points $x_{n,\nu} \equiv \cos(\nu\pi/n)$, $\nu = 0(1)n$; some unnoted advantages, *Comput. J.* 15 (1972), 156–159.

[67] Serret, J. -A. , ed. *Œuvres de Lagrange*, vols. 1–14, Gauthier-Villars, Paris, 1867–1892.

[68] Smith, Simon J. Lebesgue constants in polynomial interpolation, *Ann. Math. Inform.* 33 (2006), 109–123.

[69] Stieltjes, T. J. Quelques recherches sur la théorie des quadratures dites mécaniques, *Ann. Sci. Éc. Norm. Paris*, Sér. 3, 1 (1884), 409–426. [Also in *Œuvres* I, 377–396.]

[70] Szabados, J. On the order of magnitude of fundamental polynomials of Hermite interpolation, *Acta Math. Hungar.* 61 (1993), 357–368.

[71] Szabados, J. and Vértesi, P. *Interpolation of functions*, World Scientific, Singapore, 1990.

[72] Szabados, J. and Vértesi, P. A survey on mean convergence of interpolatory processes, *J. Comput. Appl. Math.* 43 (1992), 3–18.

[73] Szegő, Gabor. *Orthogonal polynomials*, American Mathematical Society, Colloquium Publications 23, 4th ed., American Mathematical Society, Providence, RI, 1975.

[74] Taylor, William J. Method of Lagrangian curvilinear interpolation, *J. Research Nat. Bur. Standards* 35 (1945), 151–155.

[75] Turnbull, H. W. James Gregory: a study in the early history of interpolation, *Proc. Edinburgh Math. Soc.*, ser. 2, v. 3 (1932), 151–172.

[76] Turnbull, H. W. (ed.) *The correspondence of Isaac Newton*, v. 2, 1676–1687, Cambrige University Press, 1960.

[77] Turán, Pál, ed. *Leopold Fejér—Gesammelte Arbeiten*, vol. 2, Akadémiai Kiadó, Budapest, 1970.

[78] Vértesi, Péter. One sided convergence conditions for Lagrange interpolation, *Acta Sci. Math. (Szeged)* 45 (1983), 419–428.

[79] Vértesi, Péter. Classical (unweighted) and weighted interpolation, in *A panorama of Hungarian mathematics in the twentieth century*, 71–117, Bolyai Soc. Math. Stud. 14, Springer, Berlin, 2006.

[80] Vértesi, Péter. Some remarks on quadrature formulae, *Jaen J. Approx.* 2 (2010), 31–49.

[81] Waring, Edward. Problems concerning interpolations, *Philos. Trans. R. Soc. Lond.* 69 (1779), 59–67.

AMS Subject Classification: 33C47, 65D30

Walter GAUTSCHI
Department of Computer Sciences, Purdue University
West Lafayette, IN 47907-2066, USA
e-mail: `wgautschi@purdue.edu`

Lavoro pervenuto in redazione il 07.06.2013

48

Paper on Miscellanea

222 (with Ernst Hairer) On conjectures of Stenger in the theory of orthogonal polynomials, J. Inequal. Appl., Paper No. 159, 27 pp. (2019)

48.1. [222] (with Ernst Hairer) “On conjectures of Stenger in the theory of orthogonal polymomials”

[222] (with Ernst Hairer) “On conjectures of Stenger in the theory of orthogonal polynomials”, *J. Inequal. Appl.,* Paper No. 159, 27 pp. (2019).

Gautschi and Hairer *Journal of Inequalities and Applications* (2019) 2019:159
https://doi.org/10.1186/s13660-019-2107-6

Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

On conjectures of Stenger in the theory of orthogonal polynomials

Walter Gautschi[1*] and Ernst Hairer[2]

Dedicated to Gradimir V. Milovanović on his 70th birthday

*Correspondence:
wgautschi@purdue.edu
[1] Department of Computer Science, Purdue University, West Lafayette, USA
Full list of author information is available at the end of the article

Abstract

The conjectures in the title deal with the zeros x_j, $j = 1, 2, \dots, n$, of an orthogonal polynomial of degree $n > 1$ relative to a nonnegative weight function w on an interval $[a, b]$ and with the respective elementary Lagrange interpolation polynomials $\ell_k^{(n)}$ of degree $n - 1$ taking on the value 1 at the zero x_k and the value 0 at all the other zeros x_j. They involve matrices of order n whose elements are integrals of $\ell_k^{(n)}$, either over the interval $[a, x_j]$ or the interval $[x_j, b]$, possibly containing w as a weight function. The claim is that all eigenvalues of these matrices lie in the open right half of the complex plane. This is proven to be true for Legendre polynomials and a special Jacobi polynomial. Ample evidence for the validity of the claim is provided for a variety of other classical, and nonclassical, weight functions when the integrals are weighted, but not necessarily otherwise. Even in the case of weighted integrals, however, the conjecture is found by computation to be false for a piecewise constant positive weight function. Connections are mentioned with the theory of collocation Runge–Kutta methods in ordinary differential equations.

Keywords: Zeros of orthogonal polynomials; Lagrange interpolation; Matrix eigenvalues; Conjectured location of eigenvalues in the complex plane

1 Introduction

Let w be a nonnegative weight function on $[a, b]$, $-\infty \le a < b \le \infty$, and p_n be the orthonormal polynomial of degree n relative to the weight function w. Let $\{x_j\}_{j=1}^n$ be the zeros of p_n and

$$\ell_k^{(n)}(x) = \prod_{\substack{1 \le j \le n \\ j \ne k}} \frac{x - x_j}{x_k - x_j}, \qquad k = 1, 2, \dots, n, \tag{1}$$

the elementary Lagrange interpolation polynomial of degree $n - 1$ having the value 1 at x_k and 0 at all the other zeros x_j. The Stenger conjectures relate to the eigenvalues of matrices of order n whose elements are certain integrals involving the elementary Lagrange polynomials (1), the claim being that the real part of all eigenvalues is positive. We distinguish between the *restricted Stenger conjecture* [8, §2.3, Remark 2.2], in which the matrices are

$$\begin{aligned} U_n &= \bigl[u_{jk}^{(n)}\bigr], \quad u_{jk}^{(n)} = \int_a^{x_j} \ell_k^{(n)}(x)\,\mathrm{d}x, \\ V_n &= \bigl[v_{jk}^{(n)}\bigr], \quad v_{jk}^{(n)} = \int_{x_j}^{b} \ell_k^{(n)}(x)\,\mathrm{d}x, \end{aligned} \qquad j,k = 1,2,\ldots,n, \tag{2}$$

and the *extended Stenger conjecture* (called "new conjecture" in [8, §2.4]), in which the matrices are

$$\begin{aligned} U_n &= \bigl[u_{jk}^{(n)}\bigr], \quad u_{jk}^{(n)} = \int_a^{x_j} \ell_k^{(n)}(x)w(x)\,\mathrm{d}x, \\ V_n &= \bigl[v_{jk}^{(n)}\bigr], \quad v_{jk}^{(n)} = \int_{x_j}^{b} \ell_k^{(n)}(x)w(x)\,\mathrm{d}x, \end{aligned} \qquad j,k = 1,2,\ldots,n, \tag{3}$$

where w is assumed to be positive a.e. on $[a,b]$. (For the fact that this assumption is essential, see Sects. 7 and 8.) Thus, in the latter conjecture the elements of U_n, V_n depend on the weight function w not only through the polynomials $\ell_k^{(n)}$, but also by virtue of w being part of the integration process. Note that, unlike for the extended conjecture, the restricted conjecture requires $[a,b]$ to be a finite interval, at least for one of the two matrices U_n, V_n.

We also note that the order in which the x_j are arranged is immaterial since a permutation of $j = \{1,2,3,\ldots,n\}$ implies the same permutation of $k = \{1,2,3,\ldots,n\}$, which amounts to a similarity transformation of U_n resp. V_n, and therefore leaves the eigenvalues unchanged.

The weight function $w(x) = 1$ on $[-1.1]$ is special in the sense that the extended conjecture is the same as the restricted one and will be simply called the *Stenger conjecture*. Its proof will be given in Sect. 4. In Sect. 2 we will prove that the eigenvalues of U_n and V_n in the restricted as well as in the extended Stenger conjecture are the same if w is a symmetric weight function. In Sect. 3 we show that, both in the restricted and extended conjecture, the matrix $U_n^{(\alpha,\beta)}$ belonging to the Jacobi weight function $w(x) = (1-x)^\alpha(1+x)^\beta$ on $[-1,1]$ with parameters α, β is the same as the matrix $V_n^{(\beta,\alpha)}$ with the Jacobi parameters interchanged. Section 5, devoted to the restricted Stenger conjecture, shows, partly by numerical computation, that the conjecture may be true for large classes of weight functions, but can also be false for other classes of weight functions. In contrast, Sect. 6 provides ample computational support for the validity of the extended Stenger conjecture for a variety of classical and nonclassical weight functions. Discrete weight functions are considered in Sect. 7. In Sect. 8 the extended Stenger conjecture is challenged in the case of a piecewise constant positive weight function. Related work on collocation Runge–Kutta methods is mentioned in the Appendix.

2 Symmetric weight functions

We assume here the weight function $w(x)$ to be symmetric, i.e., $w(-x) = w(x)$ on $[-b,b]$, $0 < b \leq \infty$, and the zeros x_j of the corresponding orthonormal polynomial p_n ordered increasingly:

$$-b < x_1 < x_2 < \cdots < x_n < b.$$

We then have, by symmetry,

$$x_j + x_{n+1-j} = 0, \quad j = 1, 2, \dots, n. \tag{4}$$

Theorem 1 *If w is symmetric, the eigenvalues of V_n are the same as those of U_n, both in the case of the restricted* (where $b < \infty$) *and the extended Stenger conjecture.*

Proof We present the proof for the extended conjecture, the one for the restricted conjecture being the same (just drop the factor $w(t)$ in all integrals). From the definition of V_n in (3), we have

$$v_{jk} = \int_{x_j}^{b} \ell_k^{(n)}(x)w(x)\,\mathrm{d}x = \int_{-b}^{-x_j} \ell_k^{(n)}(-t)w(t)\,\mathrm{d}t,$$

and, therefore, by (4),

$$v_{jk} = \int_{-b}^{x_{n+1-j}} \ell_k^{(n)}(-t)w(t)\,\mathrm{d}t.$$

Since $\ell_k^{(n)}(-t) = 1$ if $-t = x_k$, that is, $t = -x_k = x_{n+1-k}$, and $\ell_k^{(n)}(-t) = 0$ if $t = x_j$, $j \neq n + 1 - k$, we get

$$v_{jk} = \int_{-b}^{x_{n+1-j}} \ell_{n+1-k}^{(n)}(x)w(x)\,\mathrm{d}x,$$

thus, by (3) (with $a = -b$),

$$v_{jk} = u_{n+1-j,n+1-k}.$$

In matrix form, this can be written as

$$V_n = \begin{bmatrix} & & & 1 \\ & & \iddots & \\ & 1 & & \\ 1 & & & \end{bmatrix} U_n \begin{bmatrix} & & & 1 \\ & & \iddots & \\ & 1 & & \\ 1 & & & \end{bmatrix},$$

which is a similarity transformation of U_n. Hence, V_n and U_n have the same eigenvalues. □

3 Jacobi weight functions

In this section we look at Jacobi weight functions

$$w^{(\alpha,\beta)}(x) = (1 - z)^{\alpha}(1 + x)^{\beta} \quad \text{on } [-1, 1], \tag{5}$$

where α, β are greater than -1.

Switching Jacobi parameters has the effect of turning a U-matrix into a V-matrix and vice versa. More precisely, we have the following.

Theorem 2 *Let $U_n^{(\alpha,\beta)}$ be the matrix U_n for Jacobi polynomials with parameters α, β, and $V_n^{(\beta,\alpha)}$ be the matrix V_n for Jacobi polynomials with parameters β, α. Then*

$$U_n^{(\alpha,\beta)} = V_n^{(\beta,\alpha)}, \tag{6}$$

both in the restricted and extended Stenger conjecture.

Proof We give the proof for the restricted Stenger conjecture. It is the same for the extended conjecture, using $w^{(\alpha,\beta)}(-x) = w^{(\beta,\alpha)}(x)$.

We denote quantities x related to Jacobi parameters α, β by x^* after interchange of the parameters. Since the Jacobi polynomial satisfies $P_n^{(\alpha,\beta)}(x) = (-1)^n P_n^{(\beta,\alpha)}(-x)$ (cf. [9, Eq. (4.1.3)]), we can take $x_j^* = x_j^{(\beta,\alpha)} = -x_j = -x_j^{(\alpha,\beta)}$ for the zeros of $P_n^{(\beta,\alpha)}$. Noting that

$$\ell_k^{(n)}(x;\alpha,\beta) = \prod_{j\neq k} \frac{x - x_j}{x_k - x_j} = -\prod_{j\neq k} \frac{x + x_j^*}{x_k^* - x_j^*} = \prod_{j\neq k} \frac{(-x) - x_j^*}{x_k^* - x_j^*} = \ell_k^{(n)}(-x;\beta,\alpha),$$

we get

$$u_{jk}^{(\alpha,\beta)} = \int_{-1}^{x_j} \ell_k^{(n)}(t;\alpha,\beta)\,\mathrm{d}t = \int_{-1}^{x_j} \ell_k^{(n)}(-t;\beta,\alpha)\,\mathrm{d}t = \int_{x_j^*}^{1} \ell_k^{(n)}(x;\beta,\alpha))\,\mathrm{d}x = v_{jk}^{(\beta,\alpha)}. \qquad \square$$

4 Proof of the Stenger conjecture for Legendre polynomials

By virtue of Theorem 1, it suffices to consider the matrix U_n.

Let $\lambda \in \mathbb{C}$ be an eigenvalue of U_n and $y = [y_1, y_2, \dots, y_n]^T \in \mathbb{C}^n$ be a corresponding eigenvector,

$$U_n y = \lambda y, \quad y \neq [0, 0, \dots, 0]^T, \tag{7}$$

so that

$$\int_{-1}^{x_i} \left(\sum_{j=1}^{n} \ell_j^{(n)}(x) y_j \right) \mathrm{d}x = \lambda y_i, \quad i = 1, 2, \dots, n. \tag{8}$$

Let $y(x) \in \mathbb{P}_{n-1}$ be the unique polynomial of degree $\leq n-1$ interpolating to y_j at x_j, $j = 1, 2, \dots, n$. By the Lagrange interpolation formula and (8), we then have

$$\int_{-1}^{x_i} y(t)\,\mathrm{d}t = \lambda y(x_i), \quad i = 1, 2, \dots, n. \tag{9}$$

With w_i, $i = 1, 2, \dots, n$, denoting the weights of the n-point Gauss–Legendre quadrature formula, multiply (9) by $w_i\overline{y(x_i)}$ and sum over i to get

$$\sum_{i=1}^{n} w_i \overline{y(x_i)} \int_{-1}^{x_i} y(t)\,\mathrm{d}t = \lambda \sum_{i=1}^{n} w_i \left|y(x_i)\right|^2.$$

Since $\overline{y(x)} \int_{-1}^{x} y(t)\,\mathrm{d}t$ is a polynomial of degree $2n-1$, and n-point Gauss quadrature is exact for any such polynomial, and since $|y(x)|^2$ is a polynomial of degree $2n-2$, we have

$$\int_{-1}^{1} \overline{y(x)} \left(\int_{-1}^{x} y(t)\,\mathrm{d}t \right) \mathrm{d}x = \lambda \int_{-1}^{1} \left|y(x)\right|^2 \mathrm{d}x. \tag{10}$$

Integration by parts on the left yields the identity

$$\int_{-1}^{1}\overline{y(x)}\left(\int_{-1}^{x}y(t)\,\mathrm{d}t\right)\mathrm{d}x+\int_{-1}^{1}y(x)\left(\int_{-1}^{x}\overline{y(t)}\,\mathrm{d}t\right)\mathrm{d}x=\left|\int_{-1}^{1}y(t)\,\mathrm{d}t\right|^{2}. \tag{11}$$

The real part of the left-hand side of (10) is

$$\frac{1}{2}\left[\int_{-1}^{1}\overline{y(x)}\left(\int_{-1}^{x}y(t)\,\mathrm{d}t\right)\mathrm{d}x+\int_{-1}^{1}y(x)\left(\int_{-1}^{x}\overline{y(t)}\,\mathrm{d}t\right)\mathrm{d}x\right],$$

which, by (11), equals $\frac{1}{2}|\int_{-1}^{1}y(t)\,\mathrm{d}t|^{2}$. Therefore, taking the real part on the right of (10) yields

$$\operatorname{Re}\lambda\int_{-1}^{1}\left|y(x)\right|^{2}\mathrm{d}x=\frac{1}{2}\left|\int_{-1}^{1}y(t)\,\mathrm{d}t\right|^{2}. \tag{12}$$

From this, it follows that $\operatorname{Re}\lambda\geq 0$.

To prove strict positivity of $\operatorname{Re}\lambda$, we have to show that the integral on the right of (12) does not vanish. To do this, we look at $\int_{-1}^{x}y(t)\,\mathrm{d}t-\lambda y(x)$, which is a polynomial of degree n vanishing at x_i, $i=1,2,\dots,n$, by (9). Therefore,

$$\int_{-1}^{x}y(t)\,\mathrm{d}t-\lambda y(x)=\mathrm{const}\,P_n(x), \tag{13}$$

where P_n is the Legendre polynomial of degree n. We now multiply (13) by $(1-x)^{k-1}$, $1\leq k\leq n$, and integrate over $[-1,1]$. Then, by orthogonality, we get

$$\int_{-1}^{1}(1-x)^{k-1}\left(\int_{-1}^{x}y(t)\,\mathrm{d}t\right)\mathrm{d}x=\lambda\int_{-1}^{1}(1-x)^{k-1}y(x)\,\mathrm{d}x.$$

On the left, integrating by parts, letting

$$u(x)=\int_{-1}^{x}y(t)\,\mathrm{d}t,\qquad v'(x)=(1-x)^{k-1},$$
$$u'(x)=y(x),\qquad v(x)=\int_{1}^{x}(1-t)^{k-1}\,\mathrm{d}t=-(1-x)^{k}/k,$$

and noting that $u(-1)=v(1)=0$, we get

$$\int_{-1}^{1}\frac{(1-x)^{k}}{k}y(x)\,\mathrm{d}x=\lambda\int_{-1}^{1}(1-x)^{k-1}y(x)\,\mathrm{d}x,\quad 1\leq k\leq n. \tag{14}$$

Now suppose that $\int_{-1}^{1}y(x)\,\mathrm{d}x=0$. Then (14) for $k=1$ implies that $y(x)$ is orthogonal to all linear functions. Putting $k=2$ in (14) then implies orthogonality of $y(x)$ to all quadratic functions. Proceeding in this manner up to $k=n-1$, we conclude that $y(x)$ is orthogonal to all polynomials of degree $n-1$, in particular to itself, so that $\int_{-1}^{1}y^{2}(x)\,\mathrm{d}x=0$, hence $y(x)\equiv 0$. This contradicts (7). Thus, by (12), $\operatorname{Re}\lambda>0$. □

5 The restricted Stenger conjecture

5.1 Proof of the restricted Stenger conjecture for a special Jacobi polynomial

Here we consider the weight function $w(x) = 1 - x$ on $[-1, 1]$, that is, the Jacobi weight function $(1 - x)^\alpha(1 + x)^\beta$ with parameters $\alpha = 1$, $\beta = 0$, and denote by x_i, $i = 1, 2, \ldots, n$, the zeros of the Jacobi polynomial $P_n^{(1,0)}$ and by U_n the matrix in (2) formed with these zeros x_i. As is well known, the x_i are the internal nodes of the $(n + 1)$-point Gauss–Radau quadrature formula

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{i=1}^{n} w_i f(x_i) + w_{n+1} f(x_{n+1}), \quad f \in \mathbb{P}_{2n}, \tag{15}$$

where $x_{n+1} = 1$.

Let again $\lambda \in \mathbb{C}$ be an eigenvalue of U_n and $y = [y_1, y_2, \ldots, y_n] \in \mathbb{C}^n$ be a corresponding eigenvector, and $y(x)$ be as defined in Sect. 4. Multiplying (9) now by $w_i(1 - x_i)\overline{y(x_i)}$ and summing over $i = 1, 2, \ldots, n + 1$, we obtain

$$\sum_{i=1}^{n+1} w_i(1 - x_i)\overline{y(x_i)} \int_{-1}^{x_i} y(t)\,\mathrm{d}t = \lambda \sum_{i=1}^{n+1} w_i(1 - x_i)\bigl|y(x_i)\bigr|^2.$$

(The last term in the sums on the left and right, of course, is zero.) Therefore, by (15), since $(1 - x)\overline{y(x)} \int_{-1}^{x} y(t)\,\mathrm{d}t$ is a polynomial of degree $\le 2n$ and $(1 - x)|y(x)|^2$ a polynomial of degree $\le 2n - 1$,

$$\int_{-1}^{1} (1 - x)\overline{y(x)} \left(\int_{-1}^{x} y(t)\,\mathrm{d}t \right) \mathrm{d}x = \lambda \int_{-1}^{1} (1 - x)\bigl|y(x)\bigr|^2\,\mathrm{d}x. \tag{16}$$

The real part of the left-hand side of (16) is

$$\begin{aligned} &\frac{1}{2}\left[\int_{-1}^{1} (1 - x)\overline{y(x)} \left(\int_{-1}^{x} y(t)\,\mathrm{d}t \right) \mathrm{d}x + \int_{-1}^{1} (1 - x) y(x) \left(\int_{-1}^{x} \overline{y(t)}\,\mathrm{d}t \right) \mathrm{d}x \right] \\ &\quad = \frac{1}{2} \int_{-1}^{1} (1 - x) \frac{\mathrm{d}}{\mathrm{d}x} \left| \int_{-1}^{x} y(t)\,\mathrm{d}t \right|^2 \mathrm{d}x, \end{aligned} \tag{17}$$

having used the product rule of differentiation on the right. Integration by parts then yields

$$\frac{1}{2} \int_{-1}^{1} \left| \int_{-1}^{x} y(t)\,\mathrm{d}t \right|^2 \mathrm{d}x = \operatorname{Re}\lambda \int_{-1}^{1} (1 - x)\bigl|y(x)\bigr|^2\,\mathrm{d}x.$$

Since the integral on the right is positive, and so is the integral on the left, there follows $\operatorname{Re}\lambda > 0$. □

It may be thought that the same kind of proof might work also for Jacobi weight functions with parameters $\alpha = 0$, $\beta = 1$, or $\alpha = \beta = 1$ using Gauss–Radau quadrature with fixed node -1 or Gauss–Lobatto quadrature, respectively. The last step in the proof (integration by parts of the integral on the right of (17)), however, fails to produce the desired conclusion, the first factor in that integral being $1 + x$, resp. $1 - x^2$.

5.2 A counterexample

The simplest counterexample we came across involves a Gegenbauer polynomial of small degree.

Counterexample

$$p_n(x) = C_n^{(\alpha)}(x), \quad n = 5, \alpha = 10, \tag{18}$$

where $C_n^{(\alpha)}$ is the Gegenbauer polynomial of degree n.

From [1, Eq. 22.3.4] one finds

$$C_5^{(\alpha)}(x) = \alpha(\alpha+1)(\alpha+2)x\left[\frac{4}{15}(\alpha+3)(\alpha+4)x^4 - \frac{4}{3}(\alpha+3)x^2 + 1\right].$$

One zero of $C_5^{(\alpha)}$, of course, is 0, while the other four are the zeros of the polynomial P in brackets. When $\alpha = 10$, one finds

$$P(x) = \frac{1}{3}\left(\frac{728}{5}x^4 - 52x^2 + 3\right).$$

This is a quadratic polynomial in x^2, the zeros of which could be found explicitly. However, we proceed computationally, using Matlab, since eventually, to obtain eigenvalues, one has to compute anyway.

The Matlab routine doing the computations is `counterex.m`.[a] It computes the elements of U_n in (2) (where $n = 5$) exactly by 3-point Gauss–Legendre quadrature of the last integral in

$$u_{jk} = \int_{-1}^{x_j} \ell_k^{(5)}(x)\,\mathrm{d}x = \frac{1}{2}(1+x_j)\int_{-1}^{1} \ell_k^{(5)}\left(\frac{1}{2}(1+x_j)t - \frac{1}{2}(1-x_j)\right)\mathrm{d}t \tag{19}$$

and uses a routine `lagrange.m` for calculating the elementary Lagrange interpolation polynomials as well as the `OPQ` routines `r_jacobi.m`, `gauss.m`. For the latter, see [4, pp. 301, 304].

The output, showing the five eigenvalues `d` of U_5, is

```
>> counterex

d =

  .431796388637445 + 0.000000000000000i
  .285123529721968 + .272861054932517i
  .285123529721968 - .272861054932517i
 -.001021724040688 + .286723270044925i
 -.001021724040688 - .286723270044925i

>>
```

The last pair of eigenvalues has negative real part, disproving, at least computationally, the restricted Stenger conjecture. The extended conjecture, however, seems to be valid for this example; see Sect. 6.2, Example 1.

5.3 Conjectures

The counterexample in Sect. 5.2 is symptomatic for more general counterexamples, not only regarding Gegenbauer, but also many other weight functions. They are formulated here as separate conjectures, all firmly rooted in computational evidence.

5.3.1 Gegenbauer polynomials

Conjecture 5.1 *The restricted Stenger conjecture for U_n* (and, by Theorem 1, also for V_n) *is true for all Gegenbauer polynomials $C_n^{(\alpha)}$ with $2 \le n \le 4$, but for $n \ge 5$ it is true only for $\alpha > -1$ up to some $\alpha_n > 1$.*

The routine Uconj_restr_jac.m evaluates the matrix U_n (for Jacobi polynomials) in Matlab double-precision arithmetic and its eigenvalues in 32-digit variable-precision arithmetic. Since the eigenvalues become more ill-conditioned as n increases, we first make sure that they are accurate to at least four significant decimal digits by running the routine entirely in 32-digit arithmetic for selected values of α (and also of β) in $(-1, 1]$ and selected values of n, using the routine sUconj_restr_jac.m, and comparing the results with those obtained in double precision.

Conjecture 5.1 has then been confirmed for all $\alpha = -0.9 : 0.1 : 10$, using the routine run_Uconj_restr_jac.m. Estimates of α_n have been obtained by a bisection-type procedure and are shown in Table 1. They are "estimates" in the sense that the conjecture is true for $\alpha \le \alpha_n$, but false for $\alpha = \alpha_n + 0.001$.

It appears that α_n converges monotonically down to 1 as $n \to \infty$.

5.3.2 Jacobi polynomials

Conjecture 5.2 *The restricted Stenger conjecture for U_n holds true in the case of Jacobi polynomials $P_n^{(\alpha,\beta)}$ for all $n > 1$ if $-1 < \alpha, \beta \le 1$, but not necessarily otherwise.*

The positive part of the conjecture has been confirmed for $[\alpha, \beta] = -0.9 : 0.1 : 1$, and in each case for $n = 2 : 40$, using the routine run_Uconj_restr_jac.m. The negative part follows from Conjecture 5.1, Table 1 (if true). By Theorem 2, the same conjecture can be made for the matrix V_n.

5.3.3 Algebraic/logarithmic weight functions

Here we first examine weight functions of the type

$$w_\alpha(x) = x^\alpha \log(1/x) \quad \text{on}[0, 1] \text{ with } \alpha > -1. \tag{20}$$

Conjecture 5.3 *For the matrix U_n, the restricted Stenger conjecture holds true in the case of the weight function* (20) *for all $n > 1$ if $-1 < \alpha \le \alpha_1$, where $1 < \alpha_1 < 2$, but not necessarily otherwise. For the matrix V_n, in contrast, the conjecture is true for all $\alpha > -1$.*

Table 1 Estimates of α_n, $n = 5 : 5 : 40$

n	α_n	n	α_n
5	9.000	25	1.025
10	1.264	30	1.017
15	1.081	35	1.012
20	1.041	40	1.009

In order to compute the zeros x_j of the required orthogonal polynomials (needed to obtain the Lagrange polynomials $\ell_k^{(n)}$) for degrees $2 \le n \le 40$ and arbitrary $\alpha > -1$, we need a routine that generates the respective recurrence coefficients for the orthogonal polynomials. This can be done by applying a multicomponent discretization procedure, using appropriate quadrature rules to discretize the integral $\int_0^1 f(x)x^\alpha \log(1/x)\,\mathrm{d}x$, where f is a polynomial of degree $\le 2n-1$. It was found to be helpful to split the integral in two integrals, one extended from 0 to ξ, and the other from ξ to 1, $0 < \xi < 1$, and use ξ to optimize the rate of convergence (that is, to minimize the parameter `Mcap` in the discretization routine `mcdis.m`). Using obvious changes of variables, one finds

$$\begin{aligned}\int_0^\xi f(x)x^\alpha \log(1/x)\,\mathrm{d}x = \xi^{\alpha+1}\Biggl[\log(1/\xi)\int_0^1 f(t\xi)t^\alpha\,\mathrm{d}t \\ + \frac{1}{(1+\alpha)^2}\int_0^\infty f\bigl(\xi \mathrm{e}^{-t/(1+\alpha)}\bigr)t\mathrm{e}^{-t}\,\mathrm{d}t\Biggr],\end{aligned} \tag{21}$$

$$\int_\xi^1 f(x)x^\alpha \log(1/x)\,\mathrm{d}x = (1-\xi)\int_0^1 f\bigl(x(t)\bigr)\bigl[x(t)\bigr]^\alpha \log\bigl(1/x(t)\bigr)\,dt, \tag{22}$$

where in (22), $x(t) = (1-\xi)t + \xi$ maps the interval $[0,1]$ onto $[\xi,1]$. In (21), the first integral on the right can be discretized (without error) by n-point Gauss–Jacobi quadrature on $[0,1]$ with Jacobi parameters 0 and α, and the second integral (with small error) by sufficiently high-order generalized Gauss–Laguerre quadrature with Laguerre parameter 1. The integral in (22) can be discretized by sufficiently high-order Gauss–Legendre quadrature on $[0,1]$. For the optimal ξ, one can use, as found empirically (using the routine `run_r_alglog1.m`),

$$\xi = \begin{cases}[1+10(\alpha+0.9)]/1000 & \text{if } -0.9 \le \alpha \le 1, \\ 0.02 & \text{if } \alpha > 1.\end{cases}$$

This is implemented in the routine `r_alglog1.m`.

The routine `sUconj_restr_log1.m`, run with `dig` = 32, generates the matrix U_n and its eigenvalues in 32-digit arithmetic. It relies on the global $n \times 2$ arrays `ab` and `ableg` containing the first n recurrence coefficients of the (monic) orthogonal polynomials relative to the weight functions w_α and 1, respectively (both supported on $[0,1]$). The array `ab`, when $\alpha = -1/2, 0, 1/2, 1, 2$ is available, partly in [5, 2.3.1,2.41,2.4.3], to 32 digits for n at least as large as 100, whereas `ableg` can easily be generated by the routine `sr_jacobi01.m`. For these five values of α, we can therefore produce reference values to high precision for the eigenvalues of U_n.

The Matlab double-precision routine `Uconj_restr_log1.m`, also run with `dig` = 32, generates the matrix U_n in double-precision arithmetic and the eigenvalues in 32-digit arithmetic for arbitrary values of $\alpha > -1$, its global array `ab` being produced by the routine `r_alglog1.m`. When the eigenvalues so obtained are compared with the reference values, for the above five values of α, it is found that for $n \le 40$ they all are accurate to at least four decimal digits (cf. `test_Uconj_restr_log1.m`). This provides us with some confidence that the routine `Uconj_restr_log1.m`, when $n \le 40$, will produce eigenvalues to the same accuracy, also when α is arbitrary in the range from $-1/2$ to 2.

The routine `run_Uconj_restr_log1.m` validates the restricted Stenger conjecture for the matrix U_n when $\alpha = -1/2, 0, 1/2, 1$, at least for all n between 2 and 40, but refutes it when $\alpha = 2$ and $n = 8$, producing a pair of eigenvalues with negative real part $-1.698\ldots(-3)$. This provides some indication that Conjecture 5.3 for the matrix U_n may be valid. We strengthen this expectation by running the routine for additional values of α, and at the same time try to estimate the value of α_1 in dependence of n by applying a bisection-type procedure. It is found that, when $n \leq 40$, Conjecture 5.3 for U_n is true with α_1 as shown in Table 2.

It appears that α_1 is monotonically decreasing. Since it is bounded below by 1, it would then have to converge to a limit value (perhaps = 1).

The routines dealing with the matrix V_n are `Vconj_restr_log1.m` and `run_Vconj_restr_log1.m`. They validate Conjecture 5.3 for the matrix V_n when $\alpha = -1/2, 0, 1/2, 1, 2, 5, 10$, in each case for $2 \leq n \leq 40$.

For illustration, the eigenvalues of U_n are shown in Fig. 1 for $\alpha = 0$ and $n = 10, 20, 40$, and those of V_n in Fig. 2 for the same α and n.

For the weight function

$$w(x) = x^\alpha \log^2(1/x) \quad \text{on } [0,1], \text{with } \alpha > -1, \tag{23}$$

our conjecture for U_n is the same as the one in Conjecture 5.3, but not so for V_n.

Table 2 The values of α_1 in Conjecture 5.2 in dependence of n

n	α_1
10	1.511
20	1.253
30	1.203
40	1.179

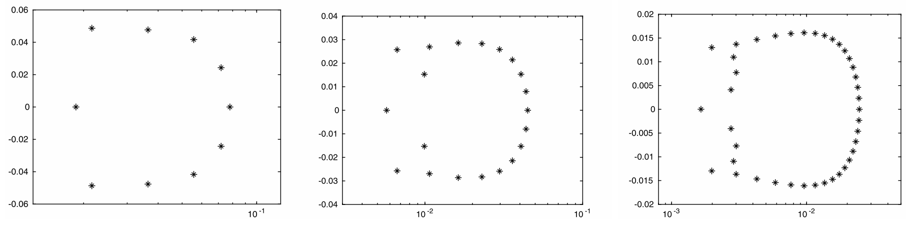


Figure 1 Eigenvalues of the matrix U_n for a logarithmic weight function and $n = 10, 20, 40$ (from left to right)

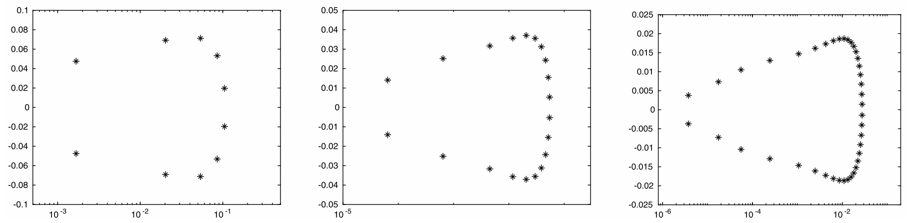


Figure 2 Eigenvalues of the matrix V_n for a logarithmic weight function and $n = 10, 20, 40$ (from left to right)

Table 3 The values of α_2 in Conjecture 5.4 in dependence of n

n	α_2
10	1.852
20	1.480
30	1.394
40	1.353

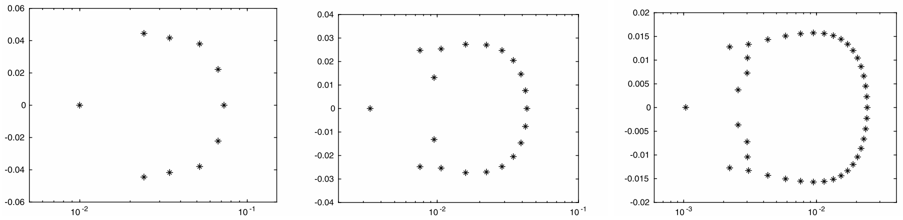


Figure 3 Eigenvalues of the matrix U_n for a square-logarithmic weight function and $n = 10, 20, 40$ (from left to right)

Conjecture 5.4 *For the matrix U_n, the restricted Stenger conjecture holds true in the case of the weight function* (23) *for all $n > 1$ if $-1 < \alpha < \alpha_2$, where α_2 is a number between* 1 *and* 2, *but not necessarily otherwise. For the matrix V_n, the conjecture is false for all $\alpha > -1$.*

The routines used to make this conjecture are the same as those used for Conjecture 5.3 but with "`log1`" replaced by "`log2`". The statements regarding the matrix U_n are arrived at in the same way as in Conjecture 5.3, the values of α_2 now being as shown in Table 3.

With regard to V_n, the conjecture is found to be false for $\alpha = -1/2, 0, 1/2, 1, 2, 5$ and $n = 7$ in each case, there being a single pair of conjugate complex eigenvalues with negative real part.

We illustrate by showing in Fig. 3 the eigenvalues of U_n for $\alpha = 0$ and $n = 10, 20, 40$.

5.3.4 Laguerre and generalized Laguerre weight functions

For generalized Laguerre weight functions

$$w(x) = x^{\alpha} e^{-x} \quad \text{on } [0, \infty], \alpha > -1, \tag{24}$$

it only makes sense to look at the U-conjecture.

Conjecture 5.5 *For the matrix U_n, the restricted Stenger conjecture is true in the case of the weight function* (24) *for all $n > 1$ if $-1 < \alpha \le \alpha_0$, where $1 < \alpha_0 < 2$, but not necessarily otherwise.*

The routines written for this conjecture are `Uconj_restr_lag.m` and `run_Uconj_restr_lag.m`. The latter, run for $\alpha = -0.9 : 0.1 : 2$, $n = 2 : 40$, confirms the conjecture up to, and including, $\alpha = 1.2$, but refutes it when $\alpha = 1.3$ and $n = 40$, producing a single pair of conjugate complex eigenvalues with negative real part. The case $\alpha = 1.3$ was checked by running the routine `run_sUconj_restr_lag.m` in 32-digit arithmetic, which produced eigenvalues agreeing with those obtained in double precision to at least 12 digits.

Table 4 The values of α_0 in Conjecture 5.5 in dependence of n

n	α_0
10	2.475
20	1.522
30	1.317
40	1.228

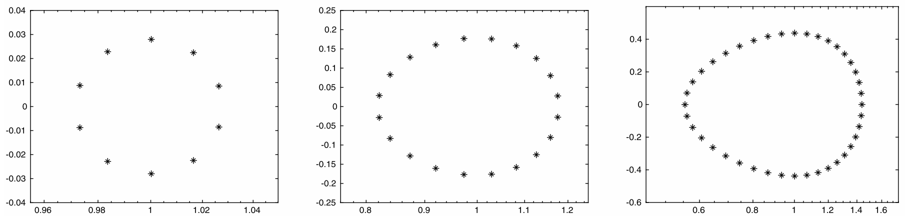


Figure 4 Eigenvalues of the matrix U_n for the Laguerre weight function and $n = 10, 20, 40$ (from left to right)

(This check may take as many as five hours to run.) A bisection-type procedure, run in double precision, yields the values of α_0 shown in Table 4 in dependence of n.

Figure 4 shows the eigenvalues of U_n when $\alpha = 0$ and $n = 10, 20, 40$.

6 The extended Stenger conjecture

To avoid extensive and time-consuming Matlab variable-precision computations, we restrict ourselves in Sects. 6.2–6.6 to values of n that are less than, or equal to, 30. Also note that in all figures of this section the horizontal axis carries a logarithmic scale.

6.1 Proof of a weak form of the extended Stenger conjecture for a special Jacobi polynomial

We consider here, as in Sect. 5.1, the Jacobi weight function $w(x) = (1-x)^\alpha(1+x)^\beta$ on $[-1,1]$, with $\alpha = 1$, $\beta = 0$, and continue using the same notations as in that section. In particular, we again use the $(n+1)$-point Gauss–Radau quadrature formula

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \sum_{i=1}^{n+1} w_i f(x_i) + R_n(f), \tag{25}$$

where $x_{n+1} = 1$, but this time we include the remainder term

$$R_n(f) = -\gamma_n \frac{f^{(2n+1)}(\xi)}{(2n+1)!}, \qquad \gamma_n = 2^{2n+1}\frac{(n+1)n!^4}{(2n+1)!^2} \tag{26}$$

(cf. [3, top of p. 158, where γ^b should read γ_n^b]). In place of (9), we now have

$$\int_{-1}^{x_i} y(t)(1-t)\,\mathrm{d}t = \lambda y(x_i), \qquad i = 1, 2, \ldots, n. \tag{27}$$

Multiplying this, as in Sect. 5.1, by $w_i(1-x_i)\overline{y(x_i)}$ and summing over $i = 1, 2, \ldots, n+1$, we obtain

$$\sum_{i=1}^{n+1} w_i(1-x_i)\overline{y(x_i)} \int_{-1}^{x_i} y(t)(1-t)\,\mathrm{d}t = \lambda \sum_{i=1}^{n+1} w_i(1-x_i)\left|y(x_i)\right|^2. \tag{28}$$

Since

$$f(x) := (1-x)\overline{y(x)}\int_{-1}^{x} y(t)(1-t)\,\mathrm{d}t \tag{29}$$

is a polynomial of degree $2n+1$ and the left-hand side of (28) is equal to the quadrature sum on the right of (25) with f as in (29), we get

$$\begin{aligned}&\sum_{i=1}^{n+1} w_i(1-x_i)\overline{y(x_i)}\int_{-1}^{x_i} y(t)(1-t)\,\mathrm{d}t\\ &\quad= \int_{-1}^{1}(1-x)\overline{y(x)}\left(\int_{-1}^{x} y(t)(1-t)\,\mathrm{d}t\right)\mathrm{d}x + \gamma_n\frac{f^{(2n+1)}(\xi)}{(2n+1)!},\end{aligned}$$

where $f^{(2n+1)}$ is a nonnegative constant, namely

$$f^{(2n+1)}(\xi) = \frac{(2n+1)!}{n+1}|a_{n-1}|^2,$$

with a_{n-1} the leading coefficient (of x^{n-1}) of the polynomial $y(x)$. Thus,

$$\begin{aligned}&\sum_{i=1}^{n+1} w_i(1-x_i)\overline{y(x_i)}\int_{-1}^{x_i} y(t)(1-t)\,\mathrm{d}t\\ &\quad= \int_{-1}^{1}(1-x)\overline{y(x)}\left(\int_{-1}^{x} y(t)(1-t)\,\mathrm{d}t\right)\mathrm{d}x + C_n,\end{aligned} \tag{30}$$

where

$$C_n = \frac{\gamma_n}{n+1}|a_{n-1}|^2.$$

Now the real part of the left-hand side of (28), by (30), is

$$\begin{aligned}&\frac{1}{2}\left[\int_{-1}^{1}(1-x)\overline{y(x)}\left(\int_{-1}^{x} y(t)(1-t)\,\mathrm{d}t\right)\mathrm{d}x\right.\\ &\qquad\left.+\int_{-1}^{1}(1-x)y(x)\left(\int_{-1}^{x}\overline{y(t)}(1-t)\,\mathrm{d}t\right)\mathrm{d}x\right] + C_n\\ &\quad= \frac{1}{2}\int_{-1}^{1}\frac{\mathrm{d}}{\mathrm{d}x}\left|\int_{-1}^{x} y(t)(1-t)\,\mathrm{d}t\right|^2\mathrm{d}x + C_n\\ &\quad= \frac{1}{2}\left|\int_{-1}^{1} y(t)(1-t)\,\mathrm{d}t\right|^2 + C_n,\end{aligned}$$

so that, by (28),

$$\frac{1}{2}\left|\int_{-1}^{1} y(t)(1-t)\,\mathrm{d}t\right|^2 + C_n = \operatorname{Re}\lambda\int_{-1}^{1}(1-x)\big|y(x)\big|^2\,\mathrm{d}x. \tag{31}$$

the integrand on the right being a polynomial of degree $2n-1$. From this, it follows that $\operatorname{Re}\lambda \geq 0$. □

Table 5 The minimum values min *int* of $|\int_{-1}^{1} y(t)(1-t)\,\mathrm{d}t|$ and $|a_{n-1}|$

n	min *int*	min $\vert a_{n-1}\vert$	n	min *int*	min $\vert a_{n-1}\vert$
5	2.273(–1)	4.425(–1)	25	4.906(–1)	1.088(–1)
10	4.228(–1)	2.062(–1)	30	4.809(–1)	1.126(–1)
15	4.200(–1)	1.358(–1)	35	4.595(–1)	1.162(–1)
20	4.966(–1)	1.193(–1)	40	4.435(–1)	1.295(–1)

Strict positivity of $\mathrm{Re}\,\lambda$ holds if $|a_{n-1}| > 0$, that is, if $y(x)$ is a polynomial of exact degree $n-1$, or if the integral on the left of (31) does not vanish. Computation, using the routines `check_pos.m` and `run_check_pos.m`, confirms that both are indeed the case, at least for $n \le 40$. Table 5 shows, for selected values of n, the minimum values of $|\int_{-1}^{1} y(t)(1-t)\,\mathrm{d}t|$ and $|a_{n-1}|$, the minimum being taken over all eigenvalues/vectors. For checking purposes, the computations have also been carried out entirely in 32-digit arithmetic.

6.2 Jacobi weight functions

The element $u_{jk}^{(n)}$ of the matrix U_n in (3) for the Jacobi weight function $w(x) = (1-x)^\alpha(1+x)^\beta$ on $[-1,1]$ is

$$u_{jk}^{(n)} = \int_{-1}^{x_j} \ell_k^{(n)}(x)w(x)\,\mathrm{d}x = \frac{1}{2}(1+x_j)\int_{-1}^{1} \ell_k^{(n)}\bigl(x(t)\bigr)w\bigl(x(t)\bigr)\,\mathrm{d}t,$$

where

$$x(t) = \frac{1}{2}(1+x_j)t - \frac{1}{2}(1-x_j)$$

maps $[-1,1]$ onto $[-1,x_j]$. An elementary computation yields

$$u_{jk}^{(n)} = \left(\frac{1+x_j}{2}\right)^{\alpha+\beta+1}\int_{-1}^{1} \ell_k^{(n)}\bigl(x(t)\bigr)\left[\frac{3-x_j}{1+x_j} - t\right]^\alpha (1+t)^\beta\,\mathrm{d}t. \tag{32}$$

Although the second factor in the integrand of (32) may be algebraically singular at a point close to, but larger than, 1 (when $x_j < 1$ is close to 1), we simply apply Gauss–Jacobi quadrature with Jacobi parameters 0 and β to the integral in (32) and choose the number of quadrature points large enough so as to produce eigenvalues of U_n accurate to at least four decimal places (which is good enough for plotting purposes). This is implemented by the Matlab function `Uconj_ext_jac.m` and can be run with the Matlab script `run_Uconj_ext_jac.m`.

Example 1 Gegenbauer weight function $w(x) = (1-x^2)^\alpha$ on $[-1,1]$ with $\alpha = 10$.

This is the weight function for which the restricted Stenger conjecture is false already for $n = 5$ (cf. Sect. 5.2). The extended conjecture, however, is found to be true for all $2 \le n \le 30$; see Fig. 5 for the cases $n = 5, 15, 30$.

Example 2 Jacobi weight function with parameters $(\alpha,\beta) = [-0.9 : 0.6 : 0.9, 1.7 : 0.7 : 3.8, 4.7 : 0.9 : 7.4]$.

We used the script `run_Uconj_ext_jac.m` to check the extended U-conjecture for all these Jacobi weight functions, separately for $n = 5, 15, 30$, and found in every case that the conjecture is valid. By Theorem 2, the same is true for the matrix V_n.

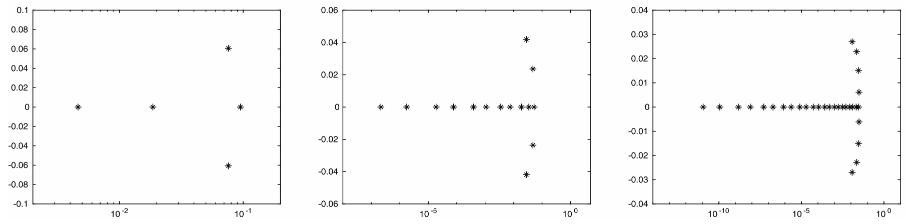


Figure 5 Eigenvalues of the matrix U_n for a special Gegenbauer polynomial of degrees $n = 5, 15, 30$ (from left to right)

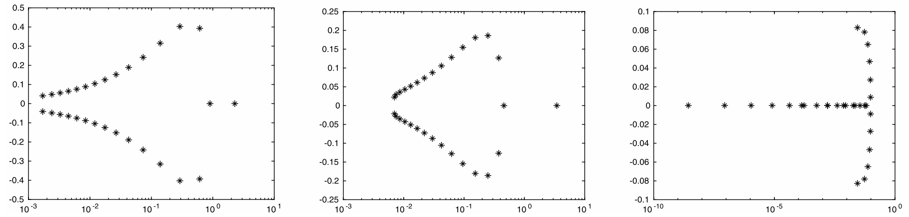


Figure 6 Eigenvalues of the matrix U_n, $n = 30$, for selected Jacobi polynomials

To illustrate, we show in Fig. 6 the eigenvalues of U_n for the three parameter choices $\alpha = \beta = -0.9$, $\alpha = -0.3$, $\beta = -0.9$, and $\alpha = 5.6$, $\beta = 1.7$, in each case with $n = 30$.

6.3 Algebraic/logarithmic weight functions

6.3.1 The weight function $w(x) = x^\alpha \log(1/x)$ *on* $[0, 1]$

Here, for the matrix U_n, we use the change of variables $x = x_j t$ in

$$u_{jk}^{(n)} = \int_0^{x_j} \ell_k^{(n)}(x) x^\alpha \log(1/x)\,\mathrm{d}x = x_j^{\alpha+1} \int_0^1 \ell_k^{(n)}(x_j t) t^\alpha \log\bigl(1/(x_j t)\bigr)\,\mathrm{d}t$$

to get

$$u_{jk}^{(n)} = x_j^{\alpha+1} \biggl[\log(1/x_j) \int_0^1 \ell_k^{(n)}(x_j t) t^\alpha\,\mathrm{d}t + \int_0^1 \ell_k^{(n)}(x_j t) t^\alpha \log(1/t)\,\mathrm{d}t\biggr]. \tag{33}$$

Both integrals can be evaluated exactly, the first by m-point Gauss–Jacobi quadrature on $[0, 1]$ with Jacobi parameters 0 and α, where $m = \lceil n/2 \rceil$, and the second by m-point Gauss quadrature relative to the weight function $w(t) = t^\alpha \log(1/t)$ on $[0, 1]$. For the latter, the recurrence coefficients for the relevant orthogonal polynomials (when $\alpha = 0, -1/2, 1/2, 1, 2, 5$) are available to 32 decimal digits, partly in [5, 2.3.1, 2.4.1, 2.4.3], which allow us to generate the Gaussian quadrature rule in a well-known manner (cf., e.g., [3, §3.1.1]) using the `OPQ` routine `gauss.m` (see [4, p. 304]). This is implemented by the Matlab function `Uconj_ext_log1.m` and can be run with the Matlab script `run_Uconj_ext_log1.m`.

Alternatively, when $n \leq 40$, we may compute the recurrence coefficients for arbitrary $\alpha > -1$ as described in Sect. 5.3.3. This is implemented by the routines `r_alglog1.m`, `Uconj_ext_log1.m`, and `run0_Uconj_ext_log1.m`.

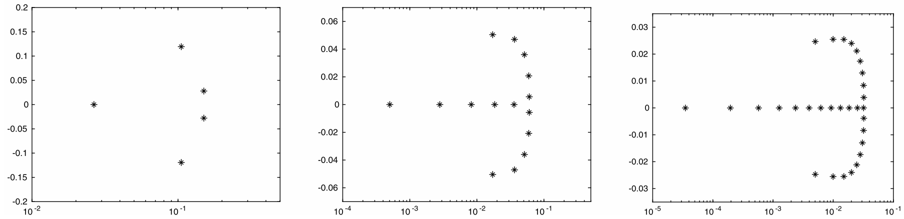


Figure 7 Eigenvalues of U_n in the case of a logarithmic weight function for $n = 5, 15, 30$ (from left to right)

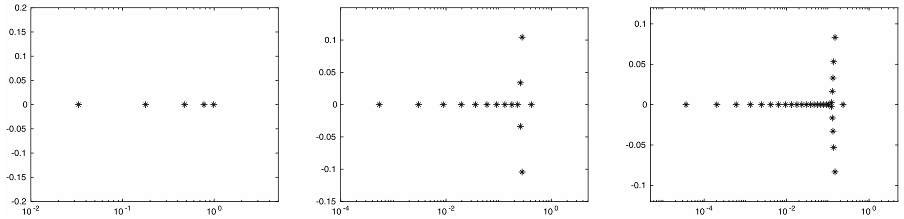


Figure 8 Eigenvalues of U_n in the case of an algebraic/logarithmic weight function with parameter $\alpha = -1/2$ for $n = 5, 15, 30$ (from left to right)

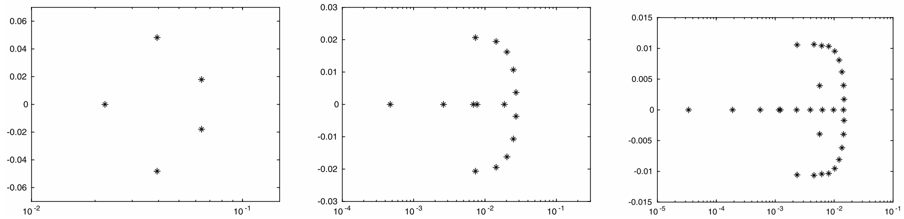


Figure 9 Eigenvalues of U_n in the case of an algebraic/logarithmic weight function with parameter $\alpha = 1/2$ for $n = 5, 15, 30$ (from left to right)

Example 3 Algebraic/logarithmic weight function $w(x) = x^\alpha \log(1/x)$ on $[0,1]$ with $\alpha = (-0.9:0.1:5)(5.2:0.2:7)(7.5:0.5:10)$.

Our routines validate the extended Stenger conjecture for all these values of α and $2 \le n \le 30$. The eigenvalues of U_n are shown in the case $\alpha = 0$ in Fig. 7, and in the cases $\alpha = -1/2, 1/2$ in Figs. 8 and 9, respectively, for $n = 5, 15, 30$. They are similar when $\alpha = 1, 2, 5$.

With regard to V_n, the conjecture has been similarly validated, using the routines `Vconj_ext_log1` and `run_Vconj_ext_log1.m`, for the same values of n and α as in Example 3. To compute the matrix V_n, we have used

$$v_{jk}^{(n)} = \int_0^1 \ell_k^{(n)}(x) x^\alpha \log(1/x)\,\mathrm{d}x - u_{jk}^{(n)} \tag{34}$$

with $u_{jk}^{(n)}$ as in (33) and the integral evaluated by $\lceil n/2 \rceil$-point Gaussian quadrature relative to the weight function $w(x)$. The eigenvalues of V_n are found to be similar to those for U_n shown in Figs. 7–9.

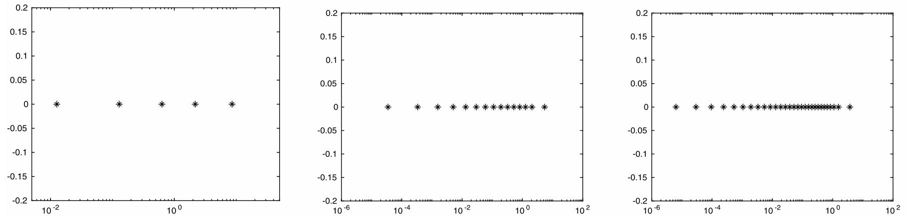


Figure 10 Eigenvalues of U_n in the case of an algebraic/square-logarithmic weight function, with exponent $\alpha = -1/2$, for $n = 5, 15, 30$ (from left to right)

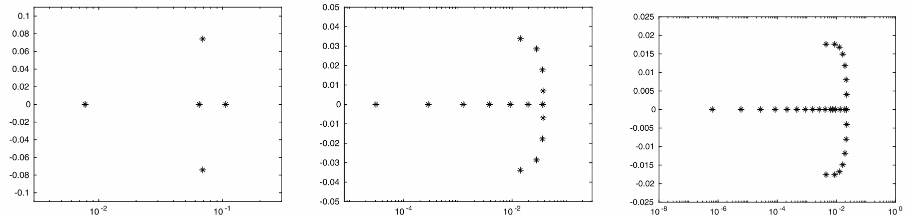


Figure 11 Eigenvalues of U_n in the case of an algebraic/square-logarithmic weight function, with exponent $\alpha = 1/2$, for $n = 5, 15, 30$ (from left to right)

6.3.2 Algebraic/square-logarithmic weight function $w(x) = x^\alpha \log^2(1/x)$ *on* $[0,1]$, $\alpha > -1$

Similarly as in Sect. 6.3.1, one finds

$$u_{jk}^{(n)} = x_j^{\alpha+1}\left[\log^2(1/x_j)\int_0^1 \ell_k^{(n)}(x_j t)t^\alpha\,\mathrm{d}t + 2\log(1/x_j)\int_0^1 \ell_k^{(n)}(x_j t)t^\alpha \log(1/t)\,\mathrm{d}t + \int_0^1 \ell_k^{(n)}(x_j t)t^\alpha \log^2(1/t)\,\mathrm{d}t\right], \tag{35}$$

where again the integrals can be evaluated exactly and some of the required recurrence coefficients taken from [5, 2.3.2], [5, 2.4.5], [5, 2.4.7]. This is implemented by the Matlab function `Uconj_ext_log2.m` and driver `run_Uconj_ext_log2.m`.

Example 4 Algebraic/square-logarithmic weight function $w(x) = x^\alpha \log^2(1/x)$ on $[0,1]$ with $\alpha = 0, -1/2, 1/2, 1, 2, 5$.

Our routines validate the extended Stenger conjecture for all these values of α and $2 \le n \le 30$. The eigenvalues of U_n in the case $\alpha = 0$ are found to be similar to those depicted in Fig. 7 for the weight function $\log(1/x)$. For the cases $\alpha = -1/2, 1/2, 5$, they are shown respectively in Figs. 10–12 for $n = 5, 15, 30$. Interestingly, all eigenvalues appear to be real when $\alpha - = -1/2$.

Similar results and validations, using the routines `Vconj_ext_log2.m` and `run_Vconj_ext_log2.m`, are obtained for the matrix V_n, which, as in (34), is computed exactly by

$$v_{jk}^{(n)} = \int_0^1 \ell_k^{(n)}(x)x^\alpha \log^2(1/x)\,\mathrm{d}x - u_{jk}^{(n)} \tag{36}$$

with $u_{jk}^{(n)}$ as in (35).

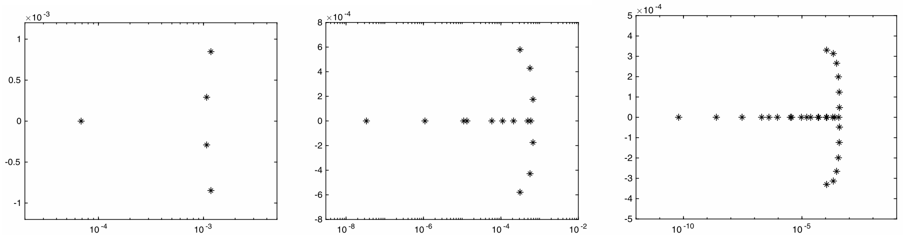


Figure 12 Eigenvalues of U_n in the case of an algebraic/square-logarithmic weight function, with exponent $\alpha = 5$, for $n = 5, 15, 30$ (from left to right)

6.4 Laguerre and generalized Laguerre weight functions

Here, the weight function is assumed to be $w(x) = x^\alpha \mathrm{e}^{-x}$ on $[0, \infty]$, where $\alpha > -1$. We write

$$u_{jk}^{(n)} = \int_0^\infty \ell_k^{(n)}(x) x^\alpha \mathrm{e}^{-x}\,\mathrm{d}x - \int_{x_j}^\infty \ell_k^{(n)}(x) x^\alpha \mathrm{e}^{-x}\,\mathrm{d}x$$

and, in the second integral, make the change of variables $x = x_j + t$ to get

$$u_{jk}^{(n)} = \int_0^\infty \ell_k^{(n)}(x) x^\alpha \mathrm{e}^{-x}\,\mathrm{d}x - \mathrm{e}^{-x_j} \int_0^\infty \ell_k^{(n)}(x_j + t)(x_j + t)^\alpha \mathrm{e}^{-t}\,\mathrm{d}t. \tag{37}$$

The first integral can be evaluated exactly by $\lceil n/2 \rceil$-point generalized Gauss–Laguerre quadrature. The second integral, similarly as in (32) for Jacobi weight functions, has an algebraic singularity close to, and to the left of, the origin when x_j is close to zero (and α not an integer). As in Sect. 6.2, we ignore this and simply apply Gauss–Laguerre quadrature of sufficiently high order so as to obtain plotting accuracy for all the eigenvalues of U_n. However, there is yet another complication: Around $n = 25$, the Gauss–Laguerre weights, in Matlab double precision, start becoming increasingly inaccurate (in terms of relative accuracy) and adversely affect the accuracy of the second integral in (37). For this reason, we use 32-digit variable-precision arithmetic to compute these weights and convert them to Matlab double precision, once computed. At the same time we lower the accuracy requirement from 4- to 3-digit accuracy.

Example 5 Generalized Laguerre weight function $w(x) = x^\alpha \mathrm{e}^{-x}\,\mathrm{d}x$ on $[0, \infty]$ for the same values of α and n as in Example 2.

The Matlab routines implementing this and validating the conjecture in each case are `Uconj_ext_lag.m` and `run_Uconj_ext_lag.m`. They may take several hours to run because of the extensive variable-precision work involved. The accuracy achieved for the eigenvalues is consistently of the order of 10^{-4} or better, but the necessary number of quadrature points is found to be as large as 440 (for $\alpha = -0.9$ and $n = 30$).

For illustration, we show in Fig. 13 the eigenvalues obtained in the case of the ordinary Laguerre weight function ($\alpha = 0$) and for $n = 5, 15, 30$. Notice the extremely small real eigenvalues when $n = 30$, the smallest being of the order 10^{-43}.

Using

$$v_{jk}^{(n)} = \int_0^\infty \ell_k^{(n)}(x) x^\alpha \mathrm{e}^{-x}\,\mathrm{d}x - u_{jk}^{(n)} \tag{38}$$

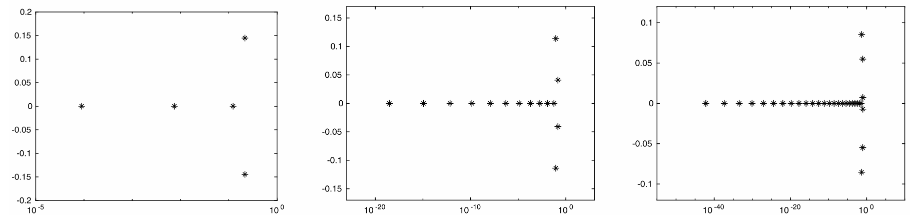


Figure 13 Eigenvalues of U_n in the case of the Laguerre weight function for $n = 5, 15, 30$ (from left to right)

with $u_{jk}^{(n)}$ as in (37), the conjecture has been similarly validated with the help of the routines `Vconj_ext_lag.m`, `run_Vconj_ext_lag.m`.

6.5 Hermite and generalized Hermite weight functions

These are the weight functions $w(x) = |x|^{2\mu} e^{-x^2}$ on $[-\infty, \infty]$, $\mu > -1/2$. Since they are symmetric, it suffices, by Theorem 1, to consider U_n. To simplify matters, we assume 2μ to be a nonnegative integer.

For the evaluation of $u_{jk}^{(n)}$, we distinguish the cases $x_j < 0$ and $x_j \geq 0$. In the former case, by the change of variables $x = x_j - t$, one gets

$$u_{jk}^{(n)} = \mathrm{e}^{-x_j^2} \int_0^\infty \ell_k^{(n)}(x_j - t)(t - x_j)^{2\mu} \mathrm{e}^{2x_j t} \mathrm{e}^{-t^2}\,\mathrm{d}t, \quad x_j < 0. \tag{39}$$

Here, half-range Gauss–Hermite quadrature (cf. [5, 2.9.1]) is expected to converge rapidly. When $x_j \geq 0$, breaking up the first integral in (3) (with $a = -\infty$) into two parts, one extended from $-\infty$ to 0 and the other from 0 to x_j, and making appropriate changes of variables in each yield

$$u_{jk}^{(n)} = \int_0^\infty \ell_k^{(n)}(-t) t^{2\mu} \mathrm{e}^{-t^2}\,\mathrm{d}t + x_j^{2\mu+1} \int_0^1 \ell_k^{(n)}(x_j t) \mathrm{e}^{-x_j^2 t^2} t^{2\mu}\,\mathrm{d}t, \quad x_j \geq 0. \tag{40}$$

The first integral can be evaluated exactly by $\lceil (n + 2\mu)/2 \rceil$-point half-range Gauss–Hermite quadrature. The second integral may be approximated by Gauss–Jacobi quadrature on $[0, 1]$ with Jacobi parameters 0 and 2μ. This, too, is expected to converge quickly.

Example 6 Generalized Hermite weight function $w(x) = |x|^{2\mu} \mathrm{e}^{-x^2}$ on $[-\infty, \infty]$, $\mu = 0 : 1/2 : 25$ and $n = 5, 15, 30$.

The conjecture has been validated in all cases using the routines `Uconj_ext_herm.m`, `run_Uconj_ext_herm.m`. For illustration, the eigenvalues of U_n are shown in Fig. 14 for the case $\mu = 0$.

6.6 A weight function supported on two disjoint intervals

We now consider a weight function which is not positive a.e.:

$$w(x) = \begin{cases} |x|(x^2 - \xi^2)^p (1 - x^2)^q & \text{if } x \in [-1, -\xi] \cup [\xi, 1], \\ 0 & \text{otherwise,} \end{cases} \tag{41}$$

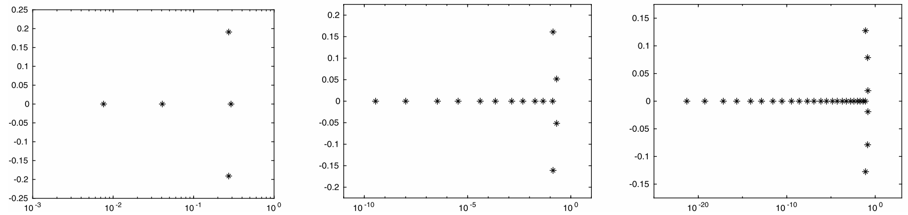


Figure 14 Eigenvalues of U_n in the case of the Hermite weight function for $n = 5, 15, 30$ (from left to right)

where $0 < \xi < 1$, $p > -1$, $q > -1$. This weight function, of interest in theoretical chemistry when $p = q = -1/2$, has been studied in [2]. In our present context, we assume, for simplicity, that p and q are nonnegative integers. Then only integrations of polynomials are required, which, as before, can be done exactly.

Since the weight function w is symmetric, it suffices, by Theorem 1, to look at the matrices U_n only.

Any polynomial π_n orthogonal with respect to w can have at most one zero in the interval $[-\xi, \xi]$ where w is zero [3, Theorem 1.20]. By symmetry, therefore, all zeros of π_n are located in the intervals $(-1, -\xi)$ or $(\xi, 1)$, except when n is odd, in which case there is a zero at the origin.

The recurrence coefficients α_k, β_k for the (monic) polynomials π_n are known explicitly [2, Eq. (4.1)]: All $\alpha_k = 0$, by symmetry, and

$$\beta_0 = \bigl(1-\xi^2\bigr)^{p+q+1}\Gamma(p+1)\Gamma(q+1)/\Gamma(p+q+2),$$

$$\beta_1 = \frac{1}{2}\bigl(1-\xi^2\bigr)\alpha_0^J + \frac{1}{2}\bigl(1+\xi^2\bigr),$$

$$\left.\begin{aligned} \beta_{2k} &= \bigl(\tfrac{1}{2}(1-\xi^2)\bigr)^2\beta_k^J/\beta_{2k-1} \\ \beta_{2k+1} &= \tfrac{1}{2}(1-\xi^2)\alpha_k^J + \tfrac{1}{2}(1+\xi^2) - \beta_{2k} \end{aligned}\right\} \quad k = 1, 2, 3, \ldots,$$

where α_k^J, β_k^J are the recurrence coefficients of the monic Jacobi polynomials with parameters $\alpha = q$, $\beta = p$. Therefore, the zeros of π_n are easily computed by the `OPQ` routine `gauss.m` (see [4, p. 304]).

The computation of $u_{jk}^{(n)}$ is different, depending on where the zero x_j is located. In fact,

$$u_{jk}^{(n)} = -\frac{1+x_j}{2}\int_{-1}^{1}\ell_k^{(n)}\bigl(x_1(t)\bigr)x_1(t)\bigl(x_1^2(t)-\xi^2\bigr)^p\bigl(1-x_1^2(t)\bigr)^q\,\mathrm{d}t \quad \text{if } x_j < -\xi,$$

where $x_1(t) = \frac{1+x_j}{2}t + \frac{x_j-1}{2}$ maps $[-1, 1]$ onto $[-1, x_j]$;

$$u_{jk}^{(n)} = -\frac{1-\xi}{2}\int_{-1}^{1}\ell_k^{(n)}\bigl(x_2(t)\bigr)x_2(t)\bigl(x_2^2(t)-\xi^2\bigr)^p\bigl(1-x_2^2(t)\bigr)^q\,\mathrm{d}t \quad \text{if } x_j = 0,$$

where $x_2(t) = \frac{1-\xi}{2}t - \frac{1+\xi}{2}$ maps $[-1, 1]$ onto $[-1, -\xi]$; and

$$u_{jk}^{(n)} = \bigl(u_{jk}^{(n)}\bigr)_{x_j=0} + \frac{x_j-\xi}{2}\int_{-1}^{1}\ell_k\bigl(x_3(t)\bigr)x_3(t)\bigl(x_3^2(t)-\xi^2\bigr)^p\bigl(1-x_3^2(t)\bigr)^q\,\mathrm{d}t \quad \text{if } x_j > \xi,$$

where $x_3(t) = \frac{x_j-\xi}{2}t + \frac{x_j+\xi}{2}$ maps $[-1, 1]$ onto $[\xi, x_j]$.

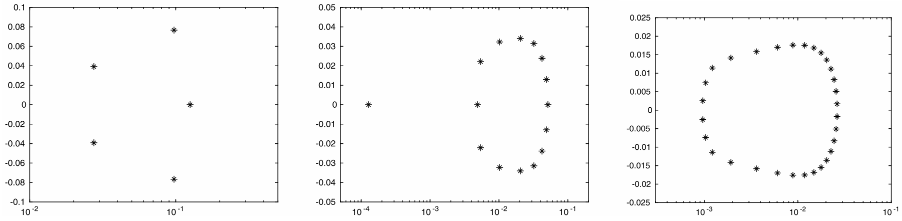


Figure 15 Eigenvalues of U_n in the case of a two-interval weight function for $n = 5, 15, 30$ (from left to right)

All integrals can be computed exactly by $(\lceil (n+1)/2 \rceil + p + q)$-point Gauss–Legendre quadrature.

Example 7 The weight function (41) with $\xi = 0.1 : 0.2 : 0.9$ and $p, q = 0 : 5$ for $n = 5, 15, 30$.

The routines `Uconj_ext_twoint.m`, `run_Uconj_ext_twoint.m` can be used to validate the conjecture in all cases, even though the weight function is not in the class of weight functions assumed in the conjecture. (For another such example, see Example 9 with $N = 1$.)

To illustrate, we show in Fig. 15 the eigenvalues of U_n, $n = 5, 15, 30$, in the case $\xi = 1/2$, $p = q = 0$, i.e., for the weight function $w(x)$ on $[-1, 1]$ equal to $|x|$ outside of $[-1/2, 1/2]$ and 0 inside.

7 Discrete weight functions

To demonstrate that an assumption about the weight function like the one made for the extended Stenger conjecture is called for, we now consider a discrete measure $d\lambda_{N+1}$ supported on $N + 1$ points $0, 1, 2, \dots, N$ with jumps $w_k > 0$ at the points k, $k = 0, 1, \dots, N$. The corresponding orthogonal polynomials, now $N + 1$ in number, are again denoted by p_n, $n = 0, 1, \dots, N$. If $w_0 = w_1 = \cdots = w_N = 1$, we are dealing with the classical discrete orthogonal polynomials attributed to Chebyshev [3, Example 1.15]). They are the special case $\alpha = \beta = 0$ of Hahn polynomials with parameters α, β (cf. [3, last entry of Table 1.2]). Both the weight function and the zeros of p_n are symmetric about the midpoint $N/2$. In particular, when N is even and n odd, one of the zeros is equal to $N/2$, hence an integer.

For the elements of U_n, we have

$$u_{j,k}^{(n)} = \sum_{i=0}^{i_j} w_i \ell_k^{(n)}(i), \quad i_j = \lfloor x_j \rfloor, \tag{42}$$

where x_j are the zeros of p_n (assumed in increasing order). These can be generated by the functions `r_hahn.m` and `gauss.m`.

Example 8 The measure $d\lambda_{N+1}$, $N \ge 2$, with $w_0 = w_1 = \cdots = w_N = 1$, and p_n with $2 \le n \le N$.

It is important to note that when the zeros of p_n are computed by the routine `gauss.m`, and when N is even and n odd, the integer zero $x_j = N/2$ may end up becoming slightly less than $N/2$, in which case $\lfloor x_j \rfloor$ in (42) will yield an incorrect result. Similarly, the smallest zero, when computed, may turn out to become negative, or the largest zero equal to N. To

Table 6 The presence of delinquent eigenvalues of U_n in the case of a discrete weight function

N	n	N	n
11	11	21	18–21
12	12	22	18–22
13	13	23	19–22 23*
14	13 14	24	16 19–23 24*
15	14 15	25	16 18 20–24 25*
16	15 16	26	18 20–25 26*
17	15–17	27	17 21*22*23* 24–26 27*
18	16–18	28	20–25 26*27*28*
19	16–19	29	18 19 22* 23 24 25*26* 27 28*29*
20	17–20	30	20 22–27 28* 29* 30*

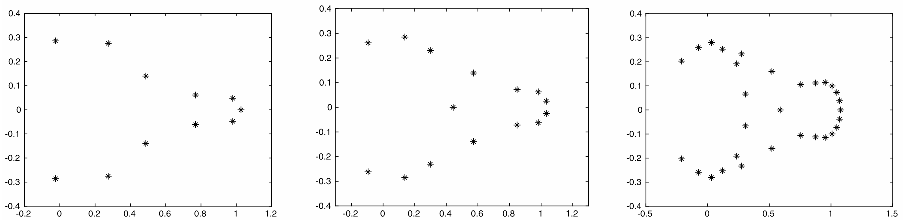


Figure 16 Eigenvalues of U_n in the case of discrete weight functions for $n = N = 11, 15, 30$ (from left to right)

avoid these pitfalls, we overwrite the zero, once computed, by $N/2$ or reset $\lfloor x_j \rfloor$, $j = 1, n$, by 0 resp. $N - 1$.

On running the script `run_Uconj_ext_hahn.m`, using `Uconj_ext_hahn.m`, to compute U_n and its eigenvalues, we found that the extended Stenger conjecture is still true for all $N \leq 10$ and all $2 \leq n \leq N$, but no longer when $N > 10$. The values of N and n for which eigenvalues with negative real parts appear are shown in Table 6 for $11 \leq N \leq 30$.

Asterisks indicate the presence of two pairs of delinquent complex conjugate eigenvalues rather than the usual single pair. (48-digit arithmetic was used for the last two entries in Table 6.)

Since the weight function is symmetric (with respect to the midpoint $N/2$), by Theorem 1 the same pattern of validity and nonvalidity holds also for the V-conjecture.

We illustrate by showing in Fig. 16 the eigenvalues of U_n, $n = N$, for $N = 11, 15, 30$.

Since there are no approximations involved, the results obtained should be quite accurate. In fact, we reran Example 8 in 48-digit arithmetic and found the double-precision eigenvalues accurate to 13, 12, and 10 digits for, resp., $n = 11, 15, 30$.

With regard to the restricted Stenger conjecture, the routines used are `run_Uconj_restr_hahn.m` and `Uconj_restr_hahn.m`. They, too, confirm the validity of the conjecture for $N \leq 10$ and $2 \leq n \leq N$. But for $N > 11$, there are now more values of n than shown in Table 6 for which there are eigenvalues with negative real parts, and there can be as many as four pairs of delinquent eigenvalues.

8 Block-discrete and ε-block-discrete weight functions

It may be interesting to see whether the eigenvalues of U_n behave similarly as in Example 8 when the weight function is not $(N + 1)$-discrete, but $(N + 1)$-block-discrete, that is, of the form

$$w(x;N+1)=\begin{cases} w_\nu & \text{if } 2\nu \le x \le 2\nu+1, \nu=0,1,\dots,N, \\ 0 & \text{otherwise,} \end{cases} \tag{43}$$

where $w_0, w_1, \dots, w_N$, $N \ge 1$, are positive numbers. Thus, the weight function is made up of $N+1$ "blocks" with base 1 and heights w_ν, $\nu = 0,1,\dots,N$, any two consecutive blocks being separated by a zero-block. More generally, we may consider $(N+1)$-ε-block-discrete weight functions, where the separating zero-blocks are replaced by ε-blocks, that is,

$$w(x;N+1,\varepsilon)=\begin{cases} w_\nu & \text{if } 2\nu \le x < 2\nu+1, \nu=0,1,\dots,N, \\ \varepsilon & \text{if } 2\nu-1 \le x < 2\nu, \nu=1,2,\dots N, \\ 0 & \text{otherwise.} \end{cases} \tag{44}$$

The orthogonal polynomials p_n associated with the weight function $w(x;N+1,\varepsilon)$ can be generated from their three-term recurrence relation, which in turn can be computed (exactly) by a $(2N+1)$-component discretization procedure (cf. [3, §2.2.4]) using $\lceil n/2 \rceil$-point Gauss–Legendre quadrature on $[0,1]$. This is implemented in Matlab double and variable precision by the routines `ab_blockhahn.m`, `sab_blockhahn.m`. (For checking purposes, the same recurrence relation was also computed by a moment-based routine in sufficiently high precision.)

The elements u_{jk} of the matrix U_n

$$u_{jk}=\int_0^{x_j} \ell_k^{(n)}(x) w(x;N+1,\varepsilon)\,\mathrm{d}x,$$

where x_j are the zeros of p_n, can be computed (exactly) as follows. Let $m = \lfloor x_j \rfloor$.

If $m=0$,

$$u_{jk}^{(n)} = w_0 \int_0^{x_j} \ell_k^{(n)}(x)\,\mathrm{d}x = w_0 x_j \int_0^1 \ell_k^{(n)}(x_j t)\,\mathrm{d}t;$$

if $m=1$,

$$\begin{aligned} u_{jk} &= w_0 \int_0^1 \ell_k^{(n)}(x)\,\mathrm{d}x + \varepsilon \int_1^{x_j} \ell_k^{(n)}(x)\,\mathrm{d}x \\ &= \int_0^1 \bigl[w_0 \ell_k^{(n)}(t) + \varepsilon (x_j-1) \ell_k^{(n)}\bigl((x_j-1)t+1\bigr)\bigr]; \end{aligned}$$

if $m>0$ is even,

$$\begin{aligned} u_{jk} &= \sum_{\nu=0}^{(m-2)/2} w_\nu \int_{2\nu}^{2\nu+1} \ell_k^{(n)}(x)\,\mathrm{d}x + w_{m/2} \int_m^{x_j} \ell_k^{(n)}(x)\,\mathrm{d}x + \varepsilon \sum_{\nu=1}^{m/2} \int_{2\nu-1}^{2\nu} \ell_k^{(n)}(x)\,\mathrm{d}x \\ &= \int_0^1 \left(\sum_{\nu=0}^{(m-2)/2} w_\nu \ell_k^{(n)}(2\nu+t) + w_{m/2}(x_j-m)\ell_k^{(n)}\bigl((x_j-m)t+m\bigr) \right. \\ &\quad \left. + \varepsilon \sum_{\nu=1}^{m/2} \ell_k^{(n)}(2\nu-1+t) \right) \mathrm{d}t; \end{aligned}$$

Table 7 The presence of delinquent eigenvalues of U_n in the case of a block-discrete weight function

N	n
2	9 17 18 20–22 25 28–30
3	11 14 21 22 24 27 28 30
4	13 14 16 18 20–22 24* 25* 29* 30
5	14 17 18 20 22–26 27* 28* 30
6	10 15–17 20–25 26* 27 29 30
7	11 17 20 21 23–28
8	14 18 20 23* 24 25* 27 28* 29* 30
9	15 19 21 23 24 26 27* 29* 30
10	13 14 16 18 20 22 26* 28* 29*

if $m > 1$ is odd,

$$\begin{aligned} u_{jk} &= \sum_{\nu=0}^{(m-1)/2} w_\nu \int_{2\nu}^{2\nu+1} \ell_k^{(n)}(x)\,\mathrm{d}x + \varepsilon \sum_{\nu=1}^{(m-1)/2} \int_{2\nu-1}^{2\nu} \ell_k^{(n)}(x)\,\mathrm{d}x + \varepsilon \int_m^{x_j} \ell_k^{(n)}(x)\,\mathrm{d}x \\ &= \int_0^1 \Biggl(w_0 \ell_k^{(n)}(t) + \sum_{\nu=1}^{(m-1)/2} \bigl[w_\nu \ell_k^{(n)}(2\nu + t) + \varepsilon \ell_k^{(n)}(2\nu - 1 + t) \bigr] \\ &\quad + \varepsilon (x_j - m) \ell_k^{(n)}\bigl((x_j - m)t + m\bigr) \Biggr)\,\mathrm{d}t. \end{aligned}$$

All integrals on the far right of these equations can be computed exactly by $\lceil n/2 \rceil$-point Gauss–Legendre quadrature on $[0, 1]$. The first pitfall mentioned in Example 8, associated with computing the floor of x_j, is no longer an issue since the midpoint is now $N + 1/2$, a half-integer, not an integer.

Example 9 The $(N + 1)$-block-discrete Hahn weight function with parameters $\alpha = \beta = 0$ and p_n with $2 \le n \le N$.

This is the weight function (43) with $w_0 = w_1 = \cdots = w_N = 1$. To check the behavior of the eigenvalues in this case, we have run the script `run_Uconj_ext_blockhahn.m` using the function `Uconj_ext_blockhahn.m` and `epsilon` = 0 for $N = 1 : 10$ and $2 \le n \le 30$ for each N. It was found that the extended Stenger conjecture is still true for $2 \le n \le 30$ (and probably for all $n \ge 2$) when $N = 1$, i.e., for a 2-block-discrete Hahn weight function. When $N > 1$, however, eigenvalues with negative real parts again show up, starting from some $n \ge 9$, and frequently, but not always, thereafter. The values of N and n, for which this occurs, are shown in Table 7. There is usually one pair of delinquent complex conjugate eigenvalues, but in some cases there are two such pairs. These are identified by an asterisk in Table 7.

The validity of the extended Stenger conjecture for $N = 1$ is interesting. It may well be for the same (unknown) reason that validates the conjecture in the case of the two-interval weight function of Sect. 6.6; cf. Example 7.

To illustrate, we show in Fig. 17 the eigenvalues in the cases $(N, n) = (2, 30), (5, 28), (10, 26)$.

The restricted Stenger conjecture, in this example, fares much better, though failing also in a few cases. Using the routines `run_Uconj_restr_blockhahn.m` and `Uconj_restr_blockhahn.m` for $N = 1 : 10$, $2 \le n \le 30$, we found the conjecture

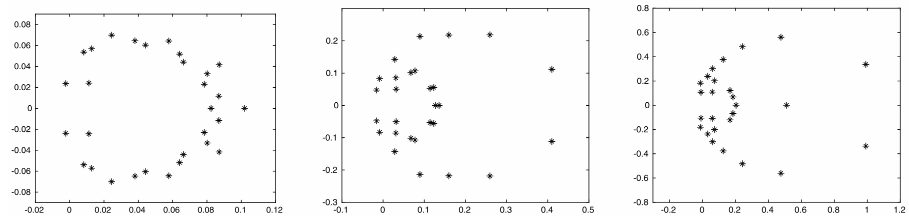


Figure 17 Eigenvalues of U_n in the case of the $(N+1)$-block-discrete Hahn weight functions with $(N,n) = (2,30), (5,28), (10,26)$ (from left to right)

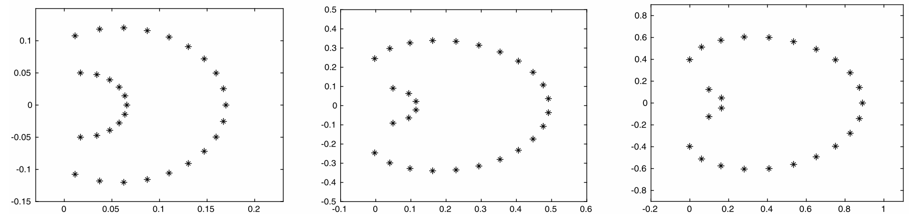


Figure 18 Eigenvalues of U_n in the case of the $(N+1)$-block-discrete Hahn weight functions with $(N,n) = (2,30), (6,28), (10,25)$ (from left to right)

to be true for $N = [1,2,3,4,9]$, $2 \le n \le 30$, and false in only the five cases: $(N,n) = (5,30),(6,28),(7,30),(8,28),(10,25)$. To rule out the presence of severe numerical instabilities as a cause for this unexpected behavior, all cases have been rerun, and confirmed, in 32-digit arithmetic. The double-precision eigenvalues were compared with those obtained in 32-digit precision and found to agree to 5–15 digits, the delinquent ones always to at least 11 digits.

For illustration, we show in Fig. 18 the eigenvalues in the cases $(N,n) = (2,30),(6,28),(10,25)$, the last two containing a pair of eigenvalues with negative real part.

The presence of delinquent eigenvalues in this example, strictly speaking, does not invalidate the extended Stenger conjecture, since the weight function (43) does not satisfy the positivity a.e. condition imposed by Stenger. However, the matrix U_n associated with the weight function (44), depending on the positive parameter ε, by a continuity argument will have the same pattern of delinquent eigenvalues as the matrix U_n associated with the weight function (43) when ε is sufficiently small. This then shows that the extended Stenger conjecture cannot be valid for all admissible weight functions. We illustrate this with the final example,

Example 10 The $(N+1)$-ε-block-discrete weight function (44) for $N = 2$, $\varepsilon = 1/100$, and $n = 9$.

This relates to the first item in Table 7. The routine `run_Uconj_ext_epsilon_blockhahn_N2_n9.m`, using `r_blockhahn` to generate the required recurrence coefficients by an $(N+1)$-component discretization procedure $(N = 2)$ implemented by the routines `mcdis.m` and `quad_blockhahn.m`, computes the eigenvalues of U_n for $n = 9$. They are shown in Table 8.

Recomputing them in 32-digit arithmetic proves them correct to all digits shown.

Table 8 The eigenvalues λ_k of U_n, $n = 9$, for the weight function of Example 10

k	λ_k	k	λ_k
1	0.269543881598 + 0.100451106056i	6	0.113959909084 + 0.146440180631i
2	0.269543881598 – 0.100451106056i	7	0.113959909084 – 0.146440180631i
3	0.257834699637	8	–0.000421036373 + 0.156050111474i
4	0.242999895873 + 0.190318565957i	9	–0.000421036373 – 0.156050111474i
5	0.242999895873 – 0.190318565957i		

Appendix: Relation to Runge–Kutta methods

Let $x_1, x_2, \dots, x_n$ be distinct real numbers (typically in the interval $[0,1]$). The corresponding (collocation) Runge–Kutta method (see [6, Theorem II. 7.7] is then given by the coefficients

$$a_{jk} = \int_0^{x_j} \ell_k(x)\,\mathrm{d}x, \qquad b_k = \int_0^1 \ell_k(x)\,\mathrm{d}x, \tag{45}$$

where $\ell_k(x)$ is the kth elementary Lagrange interpolation polynomial of degree $n-1$. We collect the coefficients in the $n \times n$ matrix $A = (a_{jk})_{j,k=1}^n$, in the column vector $b = (b_k)_{k=1}^n$, and we denote the column vector with all elements equal to 1 by $\mathbb{1}$.

An application of the Runge–Kutta method with step size h to the Dahlquist test equation $\dot{y} = \lambda y$ yields (with $z = h\lambda$)

$$y_1 = R(z)y_0, \qquad R(z) = 1 + zb^{\mathrm{T}}(I - zA)^{-1}\mathbb{1}, \tag{46}$$

where $R(z)$ is the stability function of the method. Note that for an invertible matrix A, its eigenvalues are the reciprocal of the poles of the rational function $R(z)$.

The adjoint method of (45) is given by the coefficients (cf. [6, Theorem II. 8.3])

$$a^*_{n+1-j,n+1-k} = b_k - a_{jk} = \int_{x_j}^1 \ell_k(x)\,\mathrm{d}x, \qquad b^*_{n+1-k} = b_k. \tag{47}$$

Its stability function is related to that of (45) by $R^*(z) = 1/R(-z)$.

Connection to the Stenger conjecture. The $n \times n$ matrix with coefficients a_{jk} of (45) is equal to the matrix U_n (with $a = 0$) of (2) in Sect. 1, and the matrix with coefficients a^*_{jk} of (47) is equal to V_n (with $b = 1$). Since the nonzero eigenvalues of A are the reciprocal of the poles of the stability function (46), there is a close connection between the Stenger conjecture and A-stability of a Runge–Kutta method.

The (shifted) Legendre polynomials are orthogonal with respect to the constant weight function $w(x) = 1$ on $[0,1]$. The corresponding collocation Runge–Kutta method is the so-called Gauss method of order $2n$, which is A-stable (see [7, Section IV.5]). Its stability function is the diagonal Padé approximation $R_{n,n}(x)$, for which all poles are in the right half of the complex plane. This provides another proof of the Stenger conjecture for Legendre polynomials.

Acknowledgements
The authors thank Martin J. Gander for having alerted them to the sensitivity, when n is large, of the eigenvalues of the matrices U_n, V_n to small changes in their elements, and they acknowledge helpful correspondence with Frank Stenger.

Funding
Not applicable.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The authors have equal contributions. All authors read and approved the final manuscript.

Author details

[1]Department of Computer Science, Purdue University, West Lafayette, USA. [2]Section de mathématiques, Université de Genève, Genève, Switzerland.

Endnote

[a] All Matlab routines referenced in this paper, and all textfiles used, can be accessed at CONJS of the website `https://www.cs.purdue.edu/archives/2002/wxg/codes`.

Publisher's Note

Received: 25 March 2019 Accepted: 20 May 2019 Published online: 04 June 2019

References

1. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions. Appl. Math. Ser., vol. 55 National Bureau of Standards, Washington (1964)
2. Gautschi, W.: On some orthogonal polynomials of interest in theoretical chemistry. BIT Numer. Math. **24**, 473–483 (1984) [Also in *Selected Works*, v. 2, 101–111.]
3. Gautschi, W.: Orthogonal Polynomials: Computation and Approximation, Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
4. Gautschi, W. Orthogonal Polynomials in MATLAB: Exercises and Solutions. SIAM, Philadelphia (2016)
5. Gautschi, W.: A Software Repository for Orthogonal Polynomials. SIAM, Philadelphia (2018)
6. Hairer, E., Nørsett, P., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems, 2nd rev. edn. Springer Series in Computational Mathematics, vol. 8. Springer, Berlin (1993)
7. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, 2nd rev. edn. Springer Series in Computational Mathematics, vol. 14. Springer, Berlin (1996)
8. Stenger, F., Baumann, G., Koures, V.G.: Computational methods for chemistry and physics, and Schrödinger in 3 + 1. In: Sabin, J.R., Cabrera-Trujillo, R. (eds.) Advances in Quantum Chemistry, pp. 265–298. Academic Press, San Diego (2015) Ch. 11
9. Szegö, G.: Orthogonal Polynomials, 4th edn. Colloquium Publications, vol. 23. Am. Math. Soc., Providence (1975)