Preface

This document contains a large number of datasets, each being devoted to a particular orthogonal polynomial or family of orthogonal polynomials if the weight function depends on one or more parameters. The principal objective is to provide access to the three-term recurrence relation satisfied by these polynomials, that is, to the coefficients α_k, β_k in the recurrence relation

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

where π_k is the (monic) orthogonal polynomial of degree k.

With regard to this recurrence relation, one may distinguish between classical, quasi-classical, and nonclassical orthogonal polynomials.[1] Classical orthogonal polynomials have a long history going back to the 19th or early-to-mid 20th century, and are named after individuals credited for having first used them. Quasi-classical orthogonal polynomials, on the other hand, may not have such a long history, but share with classical polynomials the fact that the recurrence coefficients are explicitly known. Nonclassical orthogonal polynomials, often occurring in specific applications (physics, chemistry, statistics, etc.), are such that the recurrence coefficients are not known in closed form and therefore require special procedures to compute them. This document is concerned foremost with such nonclassical orthogonal polynomials.

For classical and quasi-classical orthogonal polynomials, our datasets provide variable-precision Matlab routines for generating their recurrence coefficients. In the case of nonclassical orthogonal polynomials, each dataset, with a few exceptions, contains a file whose name starts with "`coeff`", which provides a table of the first $N = 100$ (occasionally fewer or more) recurrence coefficients α_k, β_k, $k = 0, 1, 2, \dots, N-1$, to an accuracy of 32 decimal digits, where β_0 is the integral of the weight function extended over its support. In addition, it contains a few variable-precision Matlab routines that allow the user to generate an arbitrary number N of recurrence coefficients to any desired accuracy, and for any admissible parameter possibly present.

There are two principal routines implementing this effort. The first, whose name begins with "`dig_`", determines the number `dig` of digits needed to obtain the first N recurrence coefficients to a prescribed accuracy. This number `dig` may well be considerably larger than the number of digits desired, depending on the procedure used to compute the recurrence coefficients. This is particularly so when moment-based methods are used, which are known to be highly susceptible to rounding errors. The way this first routine is meant to work is that the user provides a positive integer `dd` (a common choice is `dd` $= 4$) and an estimated number `dig0` of digits needed. The routine then increases (and prints) this number successively by `dd` units until the required accuracy

[1] This distinction, admittedly, is somewhat artificial, as it is not based on intrinsic structural properties of the orthogonal polynomials; it may even change with time. But, for the purpose at hand, we find it convenient.

is achieved. If this happens already after the first increase, the user is expected to lower the estimate `dig0` until at least two successive increases have occurred. The last number printed can then be taken as the value of `dig`. The second routine, whose name starts with "`sr_`", then computes in `dig`-digit arithmetic the first N recurrence coefficients and outputs them to the desired number of digits in an $N \times 2$ array `ab`.

The 32-digit recurrence coefficients tabulated in the respective dataset can be loaded into the Matlab command window by the routine `loadvpa.m` contained in the dataset

Loading variable-precision recurrence coefficients.
Purdue University Research Repository, `https://dx.doi.org/10.4231/R7T151VZ`,

where this is illustrated for a Jacobi polynomial with Jacobi parameters $\alpha = -1/2$, $\beta = 3/4$.

Each dataset can be accessed via the URL

$$\texttt{https : //dx.doi.org/10.4231/R7}\ldots,$$

where "…" has to be specified according to the `doi` information provided in the dataset.

It is important to note that the $N \times 2$ variable-precision array `ab` of recurrence coefficients furnished by the routine `sr_...` allows the user to also obtain any n-point, $n \leq N$, Gaussian quadrature formula associated with the weight function at hand, and thus, in particular, the n zeros of the nth-degree orthogonal polynomial. For this, one needs the variable-precision routine `sgauss.m` contained in the dataset

Gauss quadrature rules.
Purdue University Research Repository, `https://dx.doi.org/10.4231/R72805KQ`,

where

$$\texttt{xw} = \texttt{sgauss}(\texttt{dig}, \texttt{n}, \texttt{ab})$$

produces in `dig`-digit arithmetic the $n \times 2$ array `xw` containing in its first column the n Gaussian nodes (in increasing order) and the corresponding Gaussian weights in its second column.

The datasets displayed in Chapters 1–2 are organized in groups (subsections) according to the type of weight function w. The latter is specified by its equation and, for nonclassical weight functions, is followed by three plots: the first depicting the weight function and the other two the N resp. $N-1$ (usually $N = 100$) recurrence coefficients α_k and β_k, the former (in red) for $k = 0 : N-1$, the latter (in blue) for $k = 1 : N-1$. For classical and quasi-classical weight functions, the recurrence coefficients are displayed by their equations. In both cases, references are provided when appropriate.

The author is grateful to Michael Witt for his support and encouragement, and to Stanislav Pejša for his tireless efforts to get the datasets published. He is also receptive to suggestions for additional datasets that could be included in this repository. Please send them to `wgautschi@purdue.edu`.

October, 2017 Walter Gautschi