As demonstrated by Slepian et. al. in a sequence of classical papers (see [32], [33], [14], [34], [35]), prolate spheroidal wave functions (PSWFs) provide a natural and efficient tool for computing with bandlimited functions defined on an interval. Recently, PSWFs have been becoming increasingly popular in various areas in which such functions occur - this includes physics (e.g. wave phenomena, fluid dynamics), engineering (signal processing, filter design), etc.

To use PSWFs as a computational tool, one needs fast and accurate numerical algorithms for the evaluation of PSWFs and related quantities, as well as for the construction of corresponding quadrature rules, interpolation formulas, etc. During the last 15 years, substantial progress has been made in the design of such algorithms - see, for example, [37] (see also [3], [33], [14], [34] for some classical results).

The complexity of many of the existing algorithms, however, is at least quadratic in the band limit c. For example, the evaluation of the nth eigenvalue of the prolate integral operator requires at least $O(c^2)$ operations (see e.g. [37]); the construction of accurate quadrature rules for the integration (and associated interpolation) of bandlimited functions with band limit c requires $O(c^3)$ operations (see e.g. [4]). Therefore, while the existing algorithms are satisfactory for moderate values of c (e.g. $c \leq 10^3$), they tend to be relatively slow when c is large (e.g. $c \geq 10^4$).

In this paper, we describe several numerical algorithms for the evaluation of PSWFs and related quantities, and design a class of PSWF-based quadratures for the integration of bandlimited functions. While the analysis is somewhat involved and will be published separately (currently, it can be found in [25], [26]), the resulting numerical algorithms are quite simple and efficient in practice. For example, the evaluation of the nth eigenvalue of the prolate integral operator requires $O(n + c \cdot \log c)$ operations; the construction of accurate quadrature rules for the integration (and associated interpolation) of bandlimited functions with band limit c requires $O(c)$ operations. All algorithms described in this paper produce results essentially to machine precision. Our results are illustrated via several numerical experiments.

On the evaluation of prolate spheroidal wave functions and associated quadrature rules

Andrei Osipov[†], Vladimir Rokhlin[‡⋆]
Research Report YALEU/DCS/TR-1471
Yale University
January 8, 2013

[†] This author's research was supported in part by the AFOSR grant #FA9550-09-1-0241.
[‡] This author's research was supported in part by the ONR grants #N00014-10-1-0570, #N00014-11-1-0718, the AFOSR grant #FA9550-09-1-0241, and the ONR grant #N00014-10-C-0176 .
[⋆] This author has a significant financial interest in the Fast Mathematical Algorithms and Hardware corporation (FMAHc) of Connecticut.

Approved for public release: distribution is unlimited.
Keywords: *bandlimited functions, prolate spheroidal wave functions, quadratures, interpolation*

Contents

1 Introduction

The principal purpose of this paper is to describe several numerical algorithms associated with bandlimited functions. While these algorithms are quite simple and efficient in practice, the analysis is somewhat involved, and will be published separately (currently the proofs and additional details can be found in [25], [26], [27], [28]).

A function $f : \mathbb{R} \to \mathbb{R}$ is said to be bandlimited with band limit $c > 0$ if there exists a function $\sigma \in L^2[-1, 1]$ such that

$$f(x) = \int_{-1}^{1} \sigma(t) e^{icxt}\, dt. \tag{1}$$

In other words, the Fourier transform of a bandlimited function is compactly supported. While (1) defines f for all real x, one is often interested in bandlimited functions whose argument is confined to an interval, e.g. $-1 \le x \le 1$. Such functions are encountered in physics (wave phenomena, fluid dynamics), engineering (signal processing), etc. (see e.g. [32], [7], [29]).

About 50 years ago it was observed that the eigenfunctions of the integral operator $F_c : L^2[-1, 1] \to L^2[-1, 1]$, defined via the formula

$$F_c[\varphi](x) = \int_{-1}^{1} \varphi(t) e^{icxt}\, dt, \tag{2}$$

provide a natural tool for dealing with bandlimited functions defined on the interval $[-1, 1]$. Moreover, it was observed (see [33], [14], [34]) that the eigenfunctions of F_c are precisely the prolate spheroidal wave functions (PSWFs), well known from the mathematical physics (see, for example, [20], [7]).

Obviously, to use PSWFs as a computational tool, one needs fast and accurate numerical algorithms for the evaluation of PSWFs and related quantities, as well as for the construction of quadratures, interpolation formulas, etc. For the last 15 years, substantial progress has been made in the design of such algorithms - see, for example, [37] (see also [3], [33], [14], [34] for some classical results).

The complexity of many of the existing algorithms, however, is at least quadratic in the band limit c. For example, the evaluation of the nth eigenvalue of the prolate integral operator requires $O(c^2 + n^2)$ operations (see e.g. [37]); also, the construction of accurate quadrature rules for the integration (and associated interpolation) of bandlimited functions with band limit c requires $O(c^3)$ operations (see e.g. [4]). Therefore, while the existing algorithms are satisfactory for moderate values of c (e.g. $c \le 10^3$), they tend to be relatively slow when c is large (e.g. $c \ge 10^4$).

In this paper, we describe several numerical algorithms for the evaluation of PSWFs and related quantities, and design a class of PSWF-based quadratures for the integration of bandlimited functions. While the analysis is somewhat involved and will be published separately (currently, it can be found in [25], [26]), the resulting numerical algorithms are quite simple and efficient in practice. For example, the evaluation of the nth eigenvalue of the prolate integral operator requires $O(n + c \log c)$ operations; also, the construction of accurate quadrature rules for the integration of bandlimited functions with band limit c requires $O(c)$ operations. In addition, the evaluation of the nth PSWF is done in two steps. First, we carry out a certain precomputation, that requires $O(n + c \log c)$ operations. Then, each subsequent evaluation of this PSWF at a point in $[-1, 1]$ requires $O(1)$ operations.

This paper is organized as follows. Section 2 contains a brief overview. Section 3 contains mathematical and numerical preliminaries to be used in the rest of the paper. Section 4 contains the summary of the principal analytical results of the paper. Section 5 contains the

description and analysis of the numerical algorithms for the evaluation of the quadrature rules and some related quantities. In Section 6, we report some numerical results. In Section 7, we illustrate the analysis via several numerical experiments.

2 Overview

In this section, we provide an overview of the paper. More specifically, Section 2.1 is dedicated to the numerical evaluation of PSWFs and related quantities. In Section 2.2, we discuss several existing quadrature rules for the integration of bandlimited functions. In Section 2.3, we introduce a new class of PSWFs-based quadrature rules and describe the underlying ideas. In Section 2.4, we outline the analysis (further details can be found in [25], [26]).

2.1 Numerical Evaluation of PSWFs

For any real $c > 0$ and integer $n \geq 0$, the corresponding PSWF ψ_n can be expanded into an infinite series of Legendre polynomials (see Section 3.2). The coefficients of such expansions decay superalgebraically (see e.g [37]); in particular, relatively few terms of the Legendre series are required to evaluate $\psi_n(x)$ to essentially the machine precision, for any $-1 \leq x \leq 1$. The use of this observation for the numerical evaluation of PSWFs goes back at least to the classical Bouwkamp algorithm [3] (see also Section 3.2, in particular Theorem 10 and Remark 9, and [37] for more details).

Thus, the evaluation of PSWFs reduces to the evaluation of the corresponding Legendre coefficients. For any integer $n \geq 0$, the Legendre coefficients of *all* the first n PSWFs $\psi_0, \psi_1, \ldots, \psi_{n-1}$ can be obtained via the solution of a certain symmetric tridiagonal eigenproblem roughly of order $\max\{n, c\}$ (see Theorem 10 and Remark 9 in Section 3.2, and also [37] for more details about this algorithm). The corresponding eigenvalues $\chi_0, \chi_1, \ldots, \chi_{n-1}$ of the prolate differential operator (see Theorem 3 in Section 3.1) are obtained as a by-product of this procedure. On the other hand, additional computations are required to evaluate the corresponding eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{n-1}$ of the integral operator F_c (see (2) in Section 1). In practice, it is sometimes desirable to evaluate extremely small λ_j's (e.g. 1E-50), which presents a numerical challenge (see Section 3.1). To overcome this obstacle, the algorithm of [37] evaluates $\lambda_0, \lambda_1, \ldots, \lambda_{n-1}$ by computing the ratios λ_j/λ_{j+1}, which turns out to be a well-conditioned numerical procedure (see [37] for more details).

Suppose, on the other hand, that one is interested in a single PSWF ψ_n only (as opposed to all the first n PSWFs). Obviously, one can use the algorithm of [37]; however, its cost is at least $O(n^2)$ operations (see Remark 9). Moreover, the cost of evaluating the corresponding eigenvalue λ_n of the prolate integral operator F_c (see (2)) via the algorithm of [37] is at least $O(n^2)$ operations, with a large proportionality constant.

In this paper, we describe more efficient algorithms for the numerical evaluation of ψ_n and associated quantities. In particular, the cost of the evaluation of the Legendre coefficients of ψ_n via this algorithm is $O(n + c \log c)$ operations (see Section 5.1). In addition, the cost of the evaluation of the eigenvalue λ_n is also $O(n + c \log c)$ operations (see Section 5.2). On the other hand, this algorithm has the same accuracy as that of [37]; in other words, all of the quantities are evaluated to essentially the machine precision (see Section 5 for more

details). Since λ_n can be extremely small, the fact that it can be evaluated to high *relative* accuracy (without computing the preceding λ_j's) is, perhaps, surprising (the related analysis is somewhat subtle, and will be published separately; see [27], [28] for some preliminary results).

2.2 Quadrature Rules for Bandlimited Functions

One of principal goals of this paper is a class of quadrature rules designed for the integration of bandlimited functions with a specified band limit $c > 0$ over the interval $[-1, 1]$. Suppose that $n > 0$ is an integer; a quadrature rule of order n is a pair $\left(t_1^{(n)}, \ldots, t_n^{(n)}, W_1^{(n)}, \ldots W_n^{(n)}\right)$ of finite sequences of length n, where

$$-1 < t_1^{(n)} < \cdots < t_n^{(n)} < 1 \tag{3}$$

are referred to as "the quadrature nodes", and

$$W_1^{(n)}, \ldots, W_n^{(n)} \tag{4}$$

are referred to as "the quadrature weights". If $f : [-1, 1] \rightarrow \mathbb{R}$ is a bandlimited function (see (1) in Section 1), we use the quadrature rule to approximate the integral of f over the interval $[-1, 1]$ by a finite sum; more specifically,

$$\int_{-1}^{1} f(t)\, dt \approx \sum_{j=1}^{n} W_j^{(n)} f\left(t_j^{(n)}\right). \tag{5}$$

The PSWFs constitute a natural basis for the bandlimited functions with band limit $c > 0$ over the interval $[-1, 1]$ (see Section 1 above). Therefore, when designing a quadrature rule for the integration of such functions, it is reasonable to require that this quadrature rule integrate several first PSWFs with band limit c to high accuracy. To describe this property in a more precise manner, we introduce the following definition.

Definition 1. *Suppose that $c > 0$ is a real number, and that $n > 0$ is an integer. Suppose also that a quadrature rule for the integration of bandlimited functions with band limit c over $[-1, 1]$ is specified via its n nodes and weights, as in (3), (4). Suppose furthermore that $\varepsilon > 0$ is a real number, and that this quadrature rule integrates the first n PSWFs of band limit c to precision ε, in other words,*

$$\left| \int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} W_j^{(n)} \psi_m\left(t_j^{(n)}\right) \right| \leq \varepsilon, \tag{6}$$

for every integer $m = 0, 1, \ldots, n-1$, where $\psi_m : [-1, 1] \rightarrow \mathbb{R}$ is the mth PSWF corresponding to band limit c. We refer to such quadrature rules as "quadrature rules of order n to precision ε (corresponding to band limit c)". We omit the reference to c whenever the band limit is clear from the context.

Remark 1. *Obviously, if ε is the machine precision (e.g. $\varepsilon \approx$ 1D-16 in double precision calculations), then quadrature rules of order n to precision ε (in the sense of Definition 1) integrate the first n PSWFs exactly, for all practical purposes.*

Remark 2. *In practice, for a quadrature rule of order n to precision ε to be of any use for the integration of bandlimited functions with band limit c, not only ε should be "small", but also n has to be at least equal to $2c/\pi$. See Section 3.1 and [37] for more details.*

Quadrature rules for the integration of bandlimited functions have already been discussed in the literature, for example:

Generalized Gaussian Quadrature Rules. Suppose that $n > 0$ is an integer, and that $f_1, f_2, \ldots, f_{2n}$ are $2n$ linearly independent functions defined on an interval. Under very mild conditions on $f_1, \ldots, f_{2n}$, there exists a quadrature rule of order n that integrates these $2n$ functions exactly; moreover, its weights are usually positive. Such quadrature rules are referred to as "generalized Gaussian quadrature rules", and their existence was first observed more than 100 years ago (see, for example, [12], [13], [17], [18]). Perhaps surprisingly, numerical algorithms for the design of generalized Gaussian quadrature rules were constructed only recently (see, for example, [4], [16], [38]). These algorithms tend to be rather expensive (they require $O(n^3)$ operations with a large proportionality constant). Thus, the evaluation of the nodes and weights of a PSWF-based generalized Gaussian quadrature rule for accurate integration of bandlimited functions with band limit c requires $O(c^3)$ operations (see Remark 2 above, and also [37] for more details).

Remark 3. *We observe that a PSWF-based generalized Gaussian quadrature rule of order n integrates the first $2n$ PSWFs exactly; in other words,* (6) *holds for every integer m between 0 and $2n-1$ with $\varepsilon = 0$.*

Quadrature Rules from [37]. Suppose now that $n > 0$ is an integer, and that ψ_n is the nth PSWF corresponding to band limit c. Suppose also that $t_1, \ldots, t_n$ are the roots of ψ_n in the interval $(-1, 1)$ (see Theorem 1 in Section 3.1). Suppose furthermore that $W_1, \ldots, W_n$ are real numbers, and that

$$\sum_{i=1}^{n} \psi_m(t_i) \cdot W_i = \int_{-1}^{1} \psi_m(t)\, dt, \tag{7}$$

for every $m = 0, \ldots, n-1$. Obviously, due to (7), the quadrature rule with nodes $t_1, \ldots, t_n$ and weights $W_1, \ldots, W_n$ integrates the first n PSWFs exactly (i.e. (6) holds for every $m = 0, \ldots, n-1$ with $\varepsilon = 0$). While this quadrature rule is clearly "sub-optimal" compared to the generalized Gaussian quadrature rule of order n (the latter integrates the first $2n$ PSWFs exactly), it is somewhat less expensive to evaluate. More specifically, the cost of evaluating the roots $t_1, \ldots, t_n$ of ψ_n in $(-1, 1)$ and the weights $W_1, \ldots, W_n$, defined via (7), is dominated by the cost of solving the dense n by n linear system (7) for the unknowns $W_1, \ldots, W_n$ (see [37] for more details about the numerical aspects of this procedure). Thus, due to Remark 2 above, the cost of evaluating the nodes and weights of this quadrature rule for accurate integration of bandlimited functions with band limit c requires $O(c^3)$ operations.

Remark 4. *The cost of the evaluation of the quadrature rule, defined via* (7)*, is $O(c^3)$ operations. The cost of the evaluation of the generalized Gaussian quadrature rule is also $O(c^3)$ operations, but tends to have a larger proportionality constant.*

Remark 5. *The quadrature rule defined via* (7) *is based on the PSWFs corresponding to band limit* c*. It turns out, however, that this quadrature rule will also integrate bandlimited functions with band limit* $2c$ *to high accuracy. The reason for this is that the classical Euclid algorithm for polynomial division can be generalized to the PSWFs; the reader is referred to [37] for further details.*

In this paper, we describe another class of quadrature rules whose nodes are the n roots of ψ_n in $(-1, 1)$. However, their weights differ slightly from those defined via (7). In particular, strictly speaking, these quadrature rules do not integrate the first n PSWFs exactly, as opposed to the generalized Gaussian quadrature rules and those defined via (7) above. Nevertheless, for any $\varepsilon > 0$, they *do* integrate the first n PSWFs to precision ε, provided that

$$n > \frac{2c}{\pi} + 10 + \frac{2}{\pi^2} \cdot (\log c) \cdot \log \frac{1}{\varepsilon} \tag{8}$$

(see Theorem 15 from Section 4.2 and Conjectures 3, 4 from Section 7 for more precise statements, and Experiment 3 in Section 7.1 for some numerical results).

Thus, provided that ε is the machine precision and that (8) holds, the quadrature rules of this paper are, for all practical purposes, as accurate as those defined via (7) above. Also, their nodes and weights can be used as starting points for an iterative scheme that computes the generalized Gaussian quadrature rule (see, for example, [4], [16], [38] for more details). Last but not least, the quadrature rules of this paper are much faster to evaluate than those described above: $O(c)$ operations are required (see Sections 5.3, 5.4).

2.3 Intuition Behind Quadrature Weights

In this section, we describe the quadrature rules of this paper, and discuss the intuition behind them.

We start with a classical interpolation problem. Suppose that $t_1, \ldots, t_n$ are n distinct points on the interval $(-1, 1)$. We need to find the real numbers $W_1, \ldots, W_n$ such that

$$\int_{-1}^{1} p(t)\, dt = \sum_{i=1}^{n} W_i \cdot p(t_i), \tag{9}$$

for all polynomials p of degree at most $n - 1$. In other words, the quadrature rule with nodes $t_1, \ldots, t_n$ and weights $W_1, \ldots, W_n$ integrates all polynomials of degree up to $n - 1$ exactly (see (3), (4), (5)).

To this end, one constructs n polynomials $l_1, \ldots, l_n$ of degree $n - 1$ with the property

$$l_j(t_i) = \begin{cases} 0 & i \neq j, \\ 1 & i = j \end{cases} \tag{10}$$

for every integer $i, j = 1, \ldots, n$ (see, for example, [11]). It is easy to verify that, for every $j = 1, \ldots, n$, the polynomial l_j is defined via the formula

$$l_j(t) = \frac{w_n(t)}{w_n'(t_j) \cdot (t - t_j)}, \tag{11}$$

for all real $-1 \leq t \leq 1$, where w_n is defined via the formula

$$w_n(t) = (t - t_1) \cdot (t - t_2) \cdot \cdots \cdot (t - t_n), \tag{12}$$

for all real $-1 \leq t \leq 1$ (in other words, w_n is the polynomial of degree n whose roots are precisely $t_1, \ldots, t_n$). The weights $W_1, \ldots, W_n$ are defined via the formula

$$W_j = \int_{-1}^{1} l_j(t)\, dt = \frac{1}{w_n'(t_j)} \int_{-1}^{1} \frac{w_n(t)\, dt}{t - t_j}, \tag{13}$$

for every integer $j = 1, \ldots, n$.

In our case, the basis functions are the PSWFs rather than polynomials. We will consider the quadrature rule $(t_1, \ldots, t_n, W_1, \ldots, W_n)$, with $t_1, \ldots, t_n$ the roots of ψ_n on the interval $(-1, 1)$, and $W_1, \ldots, W_n$ to be determined. If we choose the weights $W_1, \ldots, W_n$ such that the resulting quadrature rule integrates the first n PSWFs exactly, this will lead to the linear system (7) from Section 2.2 (and hence to the corresponding quadrature rule). Instead, we define the weights using ψ_n in the same way we used w_n in (13). More specifically, for every integer $j = 1, \ldots, n$, we define the function $\varphi_j : [-1, 1] \to \mathbb{R}$ via the formula

$$\varphi_j(t) = \frac{\psi_n(t)}{\psi_n'(t_j) \cdot (t - t_j)}, \tag{14}$$

with ψ_n the obvious analogue of w_n in (11). We observe that, for every integer $i, j = 1, \ldots, n$,

$$\varphi_j(t_i) = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases} \tag{15}$$

analogous to (10). Viewed as a function on the whole real line, each φ_j is bandlimited with the same band limit c (see, for example, [25], [26], or Theorem 19.3 in [31]). We define the weights $W_1, \ldots, W_n$ via the formula

$$W_j = \int_{-1}^{1} \varphi_j(t)\, dt, \tag{16}$$

for every $j = 1, 2, \ldots, n$ (note the analogy with (13)). The weights $W_1, \ldots, W_n$, defined via (16), are different from the solution of the linear system (7). Nevertheless, the resulting quadrature rule turns out to satisfy (6), provided that ε is of order $|\lambda_n|$ (see Theorem 14 in Section 4.1 for a more precise statement).

The analysis of this issue is somewhat long and involved; the reader is referred to [25], [26] for details and proofs. On the other hand, the underlying ideas are relatively simple: Section 2.4 below contains a short overview of this analysis.

2.4 Overview of the Analysis

The following observation lies at the heart of the analysis: for any band limit $c > 0$ and any integer $n > 0$, the reciprocal of ψ_n can be approximated by a rational function with n poles in $(-1, 1)$ up to an error of order $|\lambda_n|$, where λ_n is the nth eigenvalue of the integral operator

F_c (see (2) in Section 1). In other words, the reciprocal of ψ_n resembles the reciprocal of a polynomial of order n, in the following sense.

If P is a polynomial with n simple roots $z_1, \ldots, z_n$ in $(-1, 1)$, then the function $z \to (P(z))^{-1}$ is meromorphic in the complex plane; moreover,

$$\frac{1}{P(z)} = \sum_{j=1}^{n} \frac{1}{P'(z_j) \cdot (z - z_j)}, \tag{17}$$

for all complex z different from $z_1, \ldots, z_n$ (this is a special case of the well known Cauchy's integral formula: see, for example, [31]). Similarly, the function $z \to (\psi_n(z))^{-1}$ is meromorphic; however, it has infinitely many poles, all of which are real and simple (see Remark 6 in Section 3.1), and exactly n of which lie in $(-1, 1)$ (see Theorem 1 in Section 3.1). Suppose that the roots of ψ_n in $(-1, 1)$ are denoted by $t_1 < \cdots < t_n$. It turns out that

$$\frac{1}{\psi_n(t)} = \sum_{j=1}^{n} \frac{1}{\psi_n'(t_j) \cdot (t - t_j)} + O(|\lambda_n|), \tag{18}$$

for all real $-1 \leq t \leq 1$ (note the similarity between (17) and (18)). In other words, (18) means that the reciprocal of ψ_n differs from a certain rational function with n poles by a function whose magnitude in the interval $[-1, 1]$ is of order $|\lambda_n|$. A rigorous version of (18) is provided by Theorem 9 in Section 3.1 (its proof is somewhat involved; see [25], [26] for details). More specifically, according to this theorem,

$$\left| \frac{1}{\psi_n(t)} - \sum_{j=1}^{n} \frac{1}{\psi_n'(t_j) \cdot (t - t_j)} \right| \leq |\lambda_n| \left(24 \cdot \log \left(\frac{1}{|\lambda_n|} \right) + 130 \cdot (\chi_n)^{1/4} \right), \tag{19}$$

for all real $-1 \leq t \leq 1$, where χ_n is the nth eigenvalue of the prolate differential operator (see Theorem 3 in Section 3.1).

The identity (18) is related to the quadrature, discussed in Section 2.3 above, in the following way. Multiplying both sides of (18) by $\psi_n(t)$ and using (14), we obtain

$$1 = \varphi_1(t) + \cdots + \varphi_n(t) + \psi_n(t) \cdot O\left(|\lambda_n|\right) \tag{20}$$

In other words, $\varphi_1, \ldots, \varphi_n$ constitute a partition of unity on the interval $[-1, 1]$, up to an error of order $|\lambda_n|$. We integrate both sides of (20) over $[-1, 1]$ and use Theorem 1 in Section 3.1 and (16) in Section 2.3 to obtain

$$W_1 + \cdots + W_n = 2 + O\left(|\lambda_n|\right), \tag{21}$$

where $W_1, \ldots, W_n$ are the weights of the quadrature rule (see Section 4.3 for more details).

Suppose now that $m \neq n$ is an integer. We multiply both sides of (20) by ψ_m to obtain

$$\psi_m(t) = \left(\sum_{j=1}^{n} \psi_m(t) \cdot \varphi_j(t) \right) + \psi_m(t) \cdot \psi_n(t) \cdot O\left(|\lambda_n|\right). \tag{22}$$

A detailed analysis of a combination of (19) and (22) leads to the conclusion that, for all integer $0 \leq m < n$,

$$\left| \int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \leq |\lambda_n| \cdot \left(24 \cdot \log \frac{1}{|\lambda_n|} + 6 \cdot \chi_n \right) \tag{23}$$

(see Theorem 14 in Section 4.1, and also [25], [26] for more details).

According to (23), the quadrature rule of order n integrates the first n PSWFs to precision of order $|\lambda_n|$ (see also (6) in Section 2.2). It remains to establish for what values of n this error is smaller than a predetermined $\varepsilon > 0$. Theorem 16 from Section 4.2 provides an answer to this question: namely, if

$$n > \frac{2c}{\pi} + \left(10 + \frac{3}{2} \cdot \log(c) + \frac{1}{2} \cdot \log \frac{1}{\varepsilon} \right) \cdot \log \left(\frac{c}{2} \right), \tag{24}$$

then

$$\left| \int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \leq \varepsilon, \tag{25}$$

for all integer $0 \leq m < n$.

Numerical experiments seem to indicate that the situation is even better in practice: namely, to achieve the accuracy ε it suffices to pick the minimal n such that $|\lambda_n| < \varepsilon$, which occurs for $n \approx 2c/\pi + 2(\log c) \cdot (-\log \varepsilon)/\pi^2$ (see Section 7, in particular, Conjectures 3, 4 and Experiment 3 in Section 7.1).

3 Mathematical and Numerical Preliminaries

In this section, we introduce notation and summarize several facts to be used in the rest of the paper.

3.1 Prolate Spheroidal Wave Functions

In this subsection, we summarize several facts about the PSWFs. Unless stated otherwise, all these facts can be found in [37], [30], [15], [33], [14], [21], [22].

Given a real number $c > 0$, we define the operator $F_c : L^2[-1,1] \to L^2[-1,1]$ via the formula

$$F_c[\varphi](x) = \int_{-1}^{1} \varphi(t) e^{icxt}\, dt. \tag{26}$$

Obviously, F_c is compact. We denote its eigenvalues by $\lambda_0, \lambda_1, \ldots, \lambda_n, \ldots$ and assume that they are ordered such that $|\lambda_n| \geq |\lambda_{n+1}|$ for all natural $n \geq 0$. We denote by ψ_n the eigenfunction corresponding to λ_n. In other words,

$$\lambda_n \psi_n(x) = \int_{-1}^{1} \psi_n(t) e^{icxt}\, dt, \tag{27}$$

for all integer $n \geq 0$ and all real $-1 \leq x \leq 1$. We adopt the convention[1] that $\|\psi_n\|_{L^2[-1,1]} = 1$. The following theorem describes the eigenvalues and eigenfunctions of F_c.

Theorem 1. *Suppose that $c > 0$ is a real number, and that the operator F_c is defined via (26) above. Then, the eigenfunctions $\psi_0, \psi_1, \ldots$ of F_c are purely real, are orthonormal and are complete in $L^2[-1,1]$. The even-numbered functions are even, the odd-numbered ones are odd. Each function ψ_n has exactly n simple roots in $(-1,1)$. All eigenvalues λ_n of F_c are non-zero and simple; the even-numbered ones are purely real and the odd-numbered ones are purely imaginary; in particular, $\lambda_n = i^n|\lambda_n|$, for every integer $n \geq 0$.*

We define the self-adjoint operator $Q_c : L^2[-1,1] \to L^2[-1,1]$ via the formula

$$Q_c[\varphi](x) = \frac{1}{\pi}\int_{-1}^{1} \frac{\sin(c(x-t))}{x-t}\,\varphi(t)\,dt. \tag{28}$$

Clearly,

$$Q_c[\varphi](x) = \chi_{[-1,1]}(x)\cdot \mathcal{F}^{-1}\left[\chi_{[-c,c]}(\xi)\cdot \mathcal{F}[\varphi](\xi)\right](x), \tag{29}$$

where $\mathcal{F} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ is the Fourier transform, and $\chi_{[-a,a]} : \mathbb{R} \to \mathbb{R}$ is the characteristic function of the interval $[-a,a]$, defined via the formula

$$\chi_{[-a,a]}(x) = \begin{cases} 1 & -a \leq x \leq a, \\ 0 & \text{otherwise}, \end{cases} \tag{30}$$

for all real x. In other words, Q_c represents low-passing followed by time-limiting. Q_c relates to F_c, defined via (26), by

$$Q_c = \frac{c}{2\pi}\cdot F_c^* \cdot F_c, \tag{31}$$

and the eigenvalues μ_n of Q_n satisfy the identity

$$\mu_n = \frac{c}{2\pi}\cdot |\lambda_n|^2, \tag{32}$$

for all integer $n \geq 0$. Obviously,

$$\mu_n < 1, \tag{33}$$

for all integer $n \geq 0$, due to (29). Moreover, Q_c has the same eigenfunctions ψ_n as F_c. In other words,

$$\mu_n\psi_n(x) = \frac{1}{\pi}\int_{-1}^{1} \frac{\sin(c(x-t))}{x-t}\,\psi_n(t)\,dt, \tag{34}$$

for all integer $n \geq 0$ and all $-1 \leq x \leq 1$. Also, Q_c is closely related to the operator $P_c : L^2(\mathbb{R}) \to L^2(\mathbb{R})$, defined via the formula

$$P_c[\varphi](x) = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\sin(c(x-t))}{x-t}\,\varphi(t)\,dt, \tag{35}$$

which is a widely known orthogonal projection onto the space of functions of band limit $c > 0$ on the real line $\mathbb{R}$.

The following theorem can be traced back to [15]:

[1] This convention agrees with that of [37], [30] and differs from that of [33].

Theorem 2. *Suppose that $c > 0$ and $0 < \alpha < 1$ are positive real numbers, and that the operator $Q_c : L^2[-1,1] \to L^2[-1,1]$ is defined via (28) above. Suppose also that the integer $N(c,\alpha)$ is the number of the eigenvalues μ_n of Q_c that are greater than α. In other words,*

$$N(c,\alpha) = \max\{k = 1, 2, \dots \,:\, \mu_{k-1} > \alpha\}. \tag{36}$$

Then,

$$N(c,\alpha) = \frac{2c}{\pi} + \left(\frac{1}{\pi^2}\log\frac{1-\alpha}{\alpha}\right)\log c + O(\log c). \tag{37}$$

According to (37), there are about $2c/\pi$ eigenvalues whose absolute value is close to one, order $\log c$ eigenvalues that decay rapidly, and the rest of them are very close to zero.

The eigenfunctions ψ_n of Q_c turn out to be the PSWFs, well known from classical mathematical physics [20]. The following theorem, proved in a more general form in [34], formalizes this statement.

Theorem 3. *For any $c > 0$, there exists a strictly increasing unbounded sequence of positive numbers $\chi_0 < \chi_1 < \dots$ such that, for each integer $n \geq 0$, the differential equation*

$$(1-x^2)\cdot\psi''(x) - 2x\cdot\psi'(x) + (\chi_n - c^2x^2)\cdot\psi(x) = 0 \tag{38}$$

has a solution that is continuous on $[-1,1]$. Moreover, all such solutions are constant multiples of the eigenfunction ψ_n of F_c, defined via (26) above.

Remark 6. *For all real $c > 0$ and all integer $n \geq 0$, (27) defines an analytic continuation of ψ_n onto the entire complex plane. All the roots of ψ_n are simple, real, and symmetric about the origin. Moreover, ψ_n has infinitely many roots in $(1,\infty)$. In addition, the ODE (38) is satisfied for all complex x.*

Many properties of the PSWF ψ_n depend on whether the eigenvalue χ_n of the ODE (38) is greater than or less than c^2. In the following theorem from [21], [22], we describe a simple relationship between c, n and χ_n.

Theorem 4. *Suppose that $n \geq 2$ is a non-negative integer.*

- *If $n \leq (2c/\pi) - 1$, then $\chi_n < c^2$.*
- *If $n \geq (2c/\pi)$, then $\chi_n > c^2$.*
- *If $(2c/\pi) - 1 < n < (2c/\pi)$, then either inequality is possible.*

In the following theorem, upper and lower bounds on χ_n in terms of c and n are provided.

Theorem 5. *Suppose that $c > 0$ is a real number, and $n \geq 0$ is an integer. Then,*

$$n(n+1) < \chi_n < n(n+1) + c^2. \tag{39}$$

It turns out that, for the purposes of this paper, the inequality (39) is insufficiently sharp. Tighter bounds on χ_n are described in the following theorem (see [21], [22]).

Theorem 6. *Suppose that $n \geq 2$ is an integer, and that $\chi_n > c^2$. Then,*

$$n < \frac{2}{\pi} \int_0^1 \sqrt{\frac{\chi_n - c^2 t^2}{1 - t^2}} \, dt < n + 3. \tag{40}$$

In the following theorem from [23], [24], we provide an upper bound on $|\lambda_n|$ in terms of n and c.

Theorem 7. *Suppose that $c > 0$ is a real number, and that*

$$c > 22. \tag{41}$$

Suppose also that $\delta > 0$ is a real number, and that

$$3 < \delta < \frac{\pi c}{16}. \tag{42}$$

Suppose, in addition, that n is a positive integer, and that

$$n > \frac{2c}{\pi} + \frac{2}{\pi^2} \cdot \delta \cdot \log\left(\frac{4e\pi c}{\delta}\right). \tag{43}$$

Suppose furthermore that the real number $\xi(n, c)$ is defined via the formula

$$\xi(n, c) = 7056 \cdot c \cdot \exp\left[-\delta\left(1 - \frac{\delta}{2\pi c}\right)\right]. \tag{44}$$

Then,

$$|\lambda_n| < \xi(n, c). \tag{45}$$

In the following theorem, we provide a recurrence relation between the derivatives of ψ_n of arbitrary order (see Lemma 9.1 in [37]).

Theorem 8. *Suppose that $c > 0$ is a real number, and that $n \geq 0$ is an integer. Then,*

$$\left(1 - t^2\right) \psi_n'''(t) - 4t\psi_n''(t) + \left(\chi_n - c^2 t^2 - 2\right) \psi_n'(t) - 2c^2 t \psi_n(t) = 0 \tag{46}$$

for all real t. Moreover, for all integer $k \geq 2$ and all real t,

$$\begin{aligned} &\left(1 - t^2\right) \psi_n^{(k+2)}(t) - 2\left(k + 1\right) t\psi_n^{(k+1)}(t) + \left(\chi_n - k\left(k + 1\right) - c^2 t^2\right) \psi_n^{(k)}(t) \\ &\quad - c^2 k t \psi_n^{(k-1)}(t) - c^2 k \left(k - 1\right) \psi_n^{(k-2)}(t) = 0. \end{aligned} \tag{47}$$

The following theorem asserts that, on the interval $[-1,1]$, the difference between the reciprocal of ψ_n and a certain rational function with n poles is of order $|\lambda_n|$. Its proof can be found in [25], [26].

Theorem 9. *Suppose that $c > 30$ is a real number, that n is a positive integer, and that*

$$n > \frac{2c}{\pi} + 7. \tag{48}$$

Suppose furthermore that $-1 < t_1 < \cdots < t_n < 1$ are the roots of ψ_n in $(-1,1)$, and that the function $\delta : [-1,1] \to \mathbb{R}$ is defined via the formula

$$\delta(t) = \frac{1}{\psi_n(t)} - \sum_{k=1}^{n} \frac{1}{\psi_n'(t_j) \cdot (t - t_j)}, \tag{49}$$

for all real $-1 \leq t \leq 1$. Then,

$$|\delta(t)| \leq |\lambda_n| \cdot \left(24 \cdot \log\left(\frac{1}{|\lambda_n|} \right) + 130 \cdot (\chi_n)^{1/4} \right), \tag{50}$$

for all real $-1 \leq t \leq 1$.

Remark 7. *Suppose that the function $\delta : [-1,1] \to \mathbb{R}$ is defined via (49). If n is even, then δ is an even function. If n is odd, then δ is an odd function.*

3.2 Legendre Polynomials and PSWFs

In this subsection, we list several well known facts about Legendre polynomials and the relationship between Legendre polynomials and PSWFs. All of these facts can be found, for example, in [9], [37], [1].

The Legendre polynomials $P_0, P_1, P_2, \ldots$ are defined via the formulae

$$\begin{aligned} P_0(t) &= 1, \\ P_1(t) &= t, \end{aligned} \tag{51}$$

and the recurrence relation

$$(k+1)\, P_{k+1}(t) = (2k+1)\, t P_k(t) - k P_{k-1}(t), \tag{52}$$

for all $k = 1, 2, \ldots$. Even Legendre polynomials are even functions, and odd Legendre polynomials are odd. The Legendre polynomials $\{P_k\}_{k=0}^{\infty}$ constitute a complete orthogonal system in $L^2[-1,1]$. The normalized Legendre polynomials are defined via the formula

$$\overline{P_k}(t) = P_k(t) \cdot \sqrt{k + 1/2}, \tag{53}$$

for all $k = 0, 1, 2, \ldots$. The $L^2[-1,1]$-norm of each normalized Legendre polynomial equals to one, i.e.

$$\int_{-1}^{1} \left(\overline{P_k}(t) \right)^2 \, dt = 1. \tag{54}$$

Therefore, the normalized Legendre polynomials constitute an orthonormal basis for $L^2[-1,1]$. In particular, for every real $c > 0$ and every integer $n \geq 0$, the prolate spheroidal wave function ψ_n, corresponding to the band limit c, can be expanded into the series

$$\psi_n(x) = \sum_{k=0}^{\infty} \beta_k^{(n)} \cdot \overline{P_k}(x) = \sum_{k=0}^{\infty} \alpha_k^{(n)} \cdot P_k(x), \tag{55}$$

for all $-1 \leq x \leq 1$, where $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ are defined via the formula

$$\beta_k^{(n)} = \int_{-1}^{1} \psi_n(x) \cdot \overline{P_k}(x)\, dx, \tag{56}$$

and $\alpha_0^{(n)}, \alpha_1^{(n)}, \ldots$ are defined via the formula

$$\alpha_k^{(n)} = \beta_k^{(n)} \cdot \sqrt{k+1/2} = (k+1/2) \cdot \int_{-1}^{1} \psi_n(x) \cdot P_k(x)\, dx, \tag{57}$$

for all $k = 0, 1, 2, \ldots$. Due to the combination of Theorem 1 in Section 3.1 with (54), (55), (56),

$$\left(\beta_0^{(n)}\right)^2 + \left(\beta_1^{(n)}\right)^2 + \left(\beta_2^{(n)}\right)^2 + \cdots = 1. \tag{58}$$

For any integer $n \geq 0$, the sequence $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ satisfies the recurrence relation

$$\begin{aligned} A_{0,0} \cdot \beta_0^{(n)} + A_{0,2} \cdot \beta_2^{(n)} &= \chi_n \cdot \beta_0^{(n)}, \\ A_{1,1} \cdot \beta_1^{(n)} + A_{1,3} \cdot \beta_3^{(n)} &= \chi_n \cdot \beta_1^{(n)}, \\ A_{k,k-2} \cdot \beta_{k-2}^{(n)} + A_{k,k} \cdot \beta_k^{(n)} + A_{k,k+2} \cdot \beta_{k+2}^{(n)} &= \chi_n \cdot \beta_k^{(n)}, \end{aligned} \tag{59}$$

for all $k = 2, 3, \ldots$, where $A_{k,k}$, $A_{k+2,k}$, $A_{k,k+2}$ are defined via the formulae

$$\begin{aligned} A_{k,k} &= k(k+1) + \frac{2k(k+1)-1}{(2k+3)(2k-1)} \cdot c^2, \\ A_{k,k+2} = A_{k+2,k} &= \frac{(k+2)(k+1)}{(2k+3)\sqrt{(2k+1)(2k+5)}} \cdot c^2, \end{aligned} \tag{60}$$

for all $k = 0, 1, 2, \ldots$. In other words, the infinite vector $\left(\beta_0^{(n)}, \beta_1^{(n)}, \ldots\right)$ satisfies the identity

$$(A - \chi_n I) \cdot \left(\beta_0^{(n)}, \beta_1^{(n)}, \ldots\right)^T = 0, \tag{61}$$

where I is the infinite identity matrix, and the non-zero entries of the infinite symmetric matrix A are given via (60).

The matrix A naturally splits into two infinite symmetric tridiagonal matrices, A^{even} and A^{odd}, the former consisting of the elements of A with even-indexed rows and columns, and

the latter consisting of the elements of A with odd-indexed rows and columns. Moreover, for every pair of integers $n, k \geq 0$,

$$\beta_k^{(n)} = 0, \quad \text{if } k+n \text{ is odd}, \tag{62}$$

due to the combination of Theorem 1 in Section 3.1 and (56). In the following theorem (that appears in [37] in a slightly different form), we summarize certain implications of these observations, that lead to numerical algorithms for the evaluation of PSWFs.

Theorem 10. *Suppose that $c > 0$ is a real number, and that the infinite tridiagonal symmetric matrices A^{even} and A^{odd} are defined, respectively, via*

$$A^{even} = \begin{pmatrix} A_{0,0} & A_{0,2} & & & \\ A_{2,0} & A_{2,2} & A_{2,4} & & \\ & A_{4,2} & A_{4,4} & A_{4,6} & \\ & & \ddots & \ddots & \ddots \end{pmatrix} \tag{63}$$

and

$$A^{odd} = \begin{pmatrix} A_{1,1} & A_{1,3} & & & \\ A_{3,1} & A_{3,3} & A_{3,5} & & \\ & A_{5,3} & A_{5,5} & A_{5,7} & \\ & & \ddots & \ddots & \ddots \end{pmatrix}, \tag{64}$$

where the entries $A_{k,j}$ are defined via (60)*. Suppose also that the infinite vectors $\beta_{even}^{(n)} \in l^2$ and $\beta_{odd}^{(n)} \in l^2$ are defined, respectively, via the formulae*

$$\beta_{even}^{(n)} = \left(\beta_0^{(n)}, \beta_2^{(n)}, \dots\right)^T, \quad \beta_{odd}^{(n)} = \left(\beta_1^{(n)}, \beta_3^{(n)}, \dots\right)^T, \tag{65}$$

where $\beta_0^{(n)}, \beta_1^{(n)}, \dots$ are defined via (56)*. If n is even, then*

$$A^{even} \cdot \beta_{even}^{(n)} = \chi_n \cdot \beta_{even}^{(n)}. \tag{66}$$

If n is odd, then

$$A^{odd} \cdot \beta_{odd}^{(n)} = \chi_n \cdot \beta_{odd}^{(n)}. \tag{67}$$

Remark 8. *We define the infinite vector $\beta^{(n)} \in l^2$ to be equal to $\beta_{even}^{(n)}$, if n is even, or to $\beta_{odd}^{(n)}$, if n is odd. In this notation, $\beta^{(0)}, \beta^{(2)}, \dots$ are the eigenvectors of A^{even}, and $\beta^{(1)}, \beta^{(3)}, \dots$ are the eigenvectors of A^{odd}.*

Remark 9. *While the matrices A^{even} and A^{odd} are infinite, and their entries do not decay with increasing row or column number, the coordinates of each eigenvector $\beta^{(n)}$ decay superexponentially fast (see e.g. [37] for estimates of this decay). In particular, suppose that we need to evaluate the first $n+1$ eigenvalues $\chi_0, \dots, \chi_n$ and the corresponding eigenvectors*

$\beta^{(0)}, \ldots, \beta^{(n)}$ numerically. Then, we can replace the matrices A^{even}, A^{odd} in (66), (67)*, respectively, with their $N \times N$ upper left square submatrices, where N is of order $\max\{n, c\}$, and solve the resulting symmetric tridiagonal eigenproblem by any standard technique (see, for example, [36], [5]; see also [37] for more details about this numerical algorithm). The CPU cost of this procedure is $O(n^2)$ operations.*

The Legendre functions of the second kind $Q_0, Q_1, Q_2, \ldots$ are defined via the formulae

$$\begin{aligned} Q_0(t) &= \frac{1}{2} \log \frac{1+t}{1-t}, \\ Q_1(t) &= \frac{t}{2} \log \frac{1+t}{1-t} - 1, \end{aligned} \tag{68}$$

and the recurrence relation

$$(k+1)\, Q_{k+1}(t) = (2k+1)\, tQ_k(t) - kQ_{k-1}(t), \tag{69}$$

for all $k = 1, 2, \ldots$. We observe that the recurrence relation (69) is the same as the recurrence relation (52), satisfied by the Legendre polynomials. In addition, for every integer $k = 0, 1, 2, \ldots$, the kth Legendre polynomial P_k and the kth Legendre function of the second kind Q_k are two independent solutions of the second order differential equation

$$(1-t^2) \cdot y''(t) - 2t \cdot y'(t) + k(k+1) \cdot y(t) = 0. \tag{70}$$

Remark 10. *Suppose that $-1 \leq x \leq 1$ is a real number, and that $n \geq 0$ is an integer. Combining* (51), (52), (68), (69) *gives a numerical procedure for the evaluation of $P_0(x), \ldots, P_n(x)$ and $Q_0(x), \ldots, Q_n(x)$ to high precision. This procedure is stable, and requires $O(n)$ operations (see, for example, [5] for more details).*

3.3 Prüfer Transformations

The classical Prüfer transformation of a second-order ODE is a well known analytical tool for the study of the oscillatory properties of its solutions (see, for example, [19],[6]). Recently, a minor modification of Prüfer transformation was demonstrated to be also a convenient numerical tool (see [8]). In the following theorem, we summarize several properties of this transformation, applied to the prolate ODE (38) (see [8], [21], [22] for details).

Theorem 11. *Suppose that $n \geq 2$ is an integer, and that $\chi_n > c^2$. Suppose also that the functions $f, v : (-1, 1) \to \mathbb{R}$ are defined, respectively, via the formulae*

$$f(t) = \sqrt{\frac{\chi_n - c^2 t^2}{1-t^2}} \tag{71}$$

and

$$v(t) = \frac{1}{2} \left(\frac{t}{1-t^2} + \frac{c^2 t}{\chi_n - c^2 t^2} \right), \tag{72}$$

for all real $-1 < t < 1$. Suppose furthermore that t_1 is the minimal root of ψ_n in $(-1,1)$, and that the function $\theta : (-1,1) \to \mathbb{R}$ is the solution of the differential equation

$$\theta'(t) = f(t) + v(t) \cdot \sin(2\theta(t)) \tag{73}$$

with the initial condition

$$\theta(t_1) = \frac{\pi}{2}. \tag{74}$$

Then, θ has the following properties:

- *θ extends continuously to the interval $[-1,1]$, and, moreover,*

$$\theta(-1) = 0, \tag{75}$$

$$\theta(0) = \frac{\pi n}{2}, \tag{76}$$

$$\theta(1) = \pi n. \tag{77}$$

- *For any real $-1 < t < 1$ such that $\psi_n(t) \neq 0$,*

$$\theta(t) = atan\left(-\sqrt{\frac{1-t^2}{\chi_n - c^2 t^2}} \cdot \frac{\psi_n'(t)}{\psi_n(t)}\right) + m(t) \cdot \pi, \tag{78}$$

  where $m(t)$ is the number of the roots of ψ_n in the interval $(-1,t)$.

- *For each integer $i = 1, \ldots, n$,*

$$\theta(t_i) = \left(i - \frac{1}{2}\right) \cdot \pi, \tag{79}$$

  where $t_1, \ldots, t_n$ are the roots of ψ_n in $(-1,1)$.

- *For all real $-1 < t < 1$,*

$$\theta'(t) > 0. \tag{80}$$

  In other words, θ is monotonically increasing.

The following theorem is closely related to Theorem 11 (see [21], [22] for more details).

Theorem 12. *Suppose that the function $\theta : [t_1, t_n] \to \mathbb{R}$ that of Theorem 11. Suppose also that the function $s : [\pi/2, \pi \cdot (n - 1/2)] \to [t_1, t_n]$ is the inverse of θ. Then, s is well defined, monotonically increasing and continuously differentiable. Moreover, for all real $\pi/2 < \eta < \pi \cdot (n - 1/2)$,*

$$s'(\eta) = \frac{1}{f\left(s(\eta)\right) + v\left(s(\eta)\right) \cdot \sin(2\eta)}, \tag{81}$$

where the functions f, v are defined, respectively, via (71), (72). In addition, for every integer $i = 1, \ldots, n$,

$$s\left(\left(i - \frac{1}{2}\right) \cdot \pi\right) = t_i, \tag{82}$$

and also

$$s\left(\frac{\pi n}{2}\right) = 0. \tag{83}$$

3.4 Numerical Tools

In this subsection, we summarize several numerical techniques to be used in this paper.

3.4.1 Newton's Method

Newton's method solves the equation $f(x) = 0$ iteratively given an initial approximation x_0 to the root $\tilde{x}$. The nth iteration is defined by

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}. \tag{84}$$

The convergence is quadratic provided that $\tilde{x}$ is a simple root and x_0 is sufficiently close to $\tilde{x}$. More details can be found e.g. in [5].

3.4.2 The Taylor Series Method for the Solution of ODEs

The Taylor series method for the solution of a linear second order differential equation is based on the Taylor formula

$$u(x+h) = \sum_{j=0}^{k} \frac{u^{(j)}(x)}{j!} h^j + O(h^{k+1}). \tag{85}$$

This method evaluates $u(x+h)$ and $u'(x+h)$ by using (85) and depends on the ability to compute $u^{(j)}(x)$ for $j = 0, \ldots, k$. When the latter satisfy a simple recurrence relation such as (47) and hence can be computed in $O(k)$ operations, this method is particularly useful. The reader is referred to [8] for further details.

3.4.3 A Second Order Runge-Kutta Method

A standard second order Runge-Kutta Method (see, for example, [5]) solves the initial value problem

$$y(t_0) = y_0, \quad y'(t) = f(t, y) \tag{86}$$

on the interval $t_0 \le t \le t_0 + L$ via the formulae

$$\begin{aligned} t_{i+1} &= t_i + h, \\ k_{i+1} &= hf\left(t_{i+1}, y_i + k_i\right), \\ y_{i+1} &= y_i + \left(k_i + k_{i+1}\right)/2 \end{aligned} \tag{87}$$

with $i = 0, \ldots, n$, where h and k_0 are defined via the formulae

$$h = \frac{L}{n}, \quad k_0 = f(t_0, y_0). \tag{88}$$

This procedure requires exactly $n+1$ evaluations of f. The global truncation error is $O(h^2)$.

3.4.4 Shifted Inverse Power Method

Suppose that $n \geq 0$ is an integer, and that A is an n by n real symmetric matrix. Suppose also that $\sigma_1 < \sigma_2 < \cdots < \sigma_n$ are the eigenvalues of A. The Shifted Inverse Power Method iteratively finds the eigenvalue σ_k and the corresponding eigenvector $v_k \in \mathbb{R}^n$, provided that an approximation λ to σ_k is given, and that

$$|\lambda - \sigma_k| < \max\{|\lambda - \sigma_j| \ : \ j \neq k\}. \tag{89}$$

Each Shifted Inverse Power iteration solves the linear system

$$(A - \lambda_j I) \cdot x = w_j \tag{90}$$

in the unknown $x \in \mathbb{R}^n$, where λ_j and $w_j \in \mathbb{R}^n$ are the approximations to σ_k and v_k, respectively, after j iterations; the number λ_j is usually referred to as "shift". The approximations λ_{j+1} and $w_{j+1} \in \mathbb{R}^n$ (to σ_k and v_k, respectively) are evaluated from x via the formulae

$$w_{j+1} = \frac{x}{\|x\|}, \quad \lambda_{j+1} = w_{j+1}^T \cdot A \cdot w_{j+1} \tag{91}$$

(see, for example, [5], [36] for more details).

Remark 11. *For symmetric matrices, the Shifted Inverse Power Method converges cubically in the vicinity of the solution. In particular, if the matrix A is tridiagonal, and the initial approximation λ is sufficiently close to σ_k, the Shifted Inverse Power Method evaluates σ_k and v_k essentially to machine precision ε in $O(-\log\log\varepsilon)$ iterations, and each iteration requires $O(n)$ operations (see e.g [36], [5]).*

3.4.5 Sturm Bisection

In this subsection, we describe a well known algorithm for the evaluation of a single eigenvalue of a real symmetric tridiagonal matrix. This algorithm is based on the following theorem that can be found, for example, in [36], [2].

Theorem 13 (Sturm sequence)**.** *Suppose that $n > 0$ is an integer, that*

$$C = \begin{pmatrix} a_1 & b_2 & 0 & \cdots & \cdots & 0 \\ b_2 & a_2 & b_3 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{n-1} & a_{n-1} & b_n \\ 0 & \cdots & \cdots & 0 & b_n & a_n \end{pmatrix} \tag{92}$$

is an n by n symmetric tridiagonal matrix, and that none of numbers $b_2, \ldots, b_n$ is equal to zero. Suppose also that the polynomials $p_{-1}, p_0, \ldots, p_n$ are defined via the formulae

$$p_{-1}(x) = 0, \quad p_0(x) = 1 \tag{93}$$

and

$$p_k(x) = (a_k - x)\, p_{k-1}(x) - b_k^2 p_{k-2}(x), \tag{94}$$

for all real x and every integer $k = 2, \ldots, n$. Suppose furthermore that σ is a real number, and that the integer $A(\sigma)$ is defined as the number of positive elements in the finite sequence

$$p_0(\sigma)p_1(\sigma),\ p_1(\sigma)p_2(\sigma),\ \ldots,\ p_{n-1}(\sigma)p_n(\sigma). \tag{95}$$

Then, the number of eigenvalues of C that are strictly larger than σ is precisely $A(\sigma)$.

Remark 12. *Suppose now that $n > 0$ is an integer, and C is an $n \times n$ real symmetric tridiagonal matrix, such as (92). Theorem 13 yields a numerical scheme for the evaluation of the kth smallest eigenvalue σ_k of C. This scheme is known in the literature as "Sturm Bisection". Provided that two real numbers x_0 and y_0 are given such that*

$$x_0 < \sigma_k < y_0, \tag{96}$$

Sturm Bisection requires

$$O\left(n \cdot \log_2\left(\frac{y_0 - x_0}{|\sigma_k|}\right)\right) \tag{97}$$

operations to evaluate σ_k to machine precision (see, for example, [36], [2] for more details).

4 Analytical Apparatus

The purpose of this section is to provide the analytical apparatus to be used in the rest of the paper. More specifically, we define a PSWF-based quadrature rule and list several of its properties.

The principal result of this section is Theorem 16. The reader is referred to [25], [26] for the detailed analysis of all the tools listed in this section.

Throughout this section, the band limit $c > 0$ is assumed to be a positive real number. Also, for any integer $n \geq 0$, we denote by ψ_n the nth PSWF corresponding to the band limit c (see Section 3.1).

Definition 2. *Suppose that $n > 0$ is an integer, and that*

$$-1 < t_1 < t_2 < \cdots < t_n < 1 \tag{98}$$

are the roots of ψ_n in the interval $(-1, 1)$. For each integer $j = 1, \ldots, n$, we define the function $\varphi_j : [-1, 1] \to \mathbb{R}$ via the formula

$$\varphi_j(t) = \frac{\psi_n(t)}{\psi_n'(t_j)\,(t - t_j)}. \tag{99}$$

In addition, for each integer $j = 1, \ldots, n$, we define the real number W_j via the formula

$$W_j = \int_{-1}^{1} \varphi_j(s)\, ds = \frac{1}{\psi_n'(t_j)} \int_{-1}^{1} \frac{\psi_n(s)\, ds}{s - t_j}. \tag{100}$$

We refer to the pair of finite sequences

$$S_n = (t_1, \ldots, t_n, W_1, \ldots, W_n) \tag{101}$$

as the "PSWF-based quadrature rule of order n". The points $t_1, \ldots, t_n$ are referred to as the quadrature nodes, and the numbers $W_1, \ldots, W_n$ are referred to as the quadrature weights (see (3), (4) *in Section 2.2). We use S_n to approximate the integral of a bandlimited function f over the interval $[-1, 1]$ by a finite sum; more specifically,*

$$\int_{-1}^{1} f(t)\, dt \approx \sum_{j=1}^{n} W_j \cdot f(t_j). \tag{102}$$

We refer to the number $\delta_n(f)$ defined via the formula

$$\delta_n(f) = \left| \int_{-1}^{1} f(t)\, dt - \sum_{j=1}^{n} W_j \cdot f(t_j) \right| \tag{103}$$

as the "quadrature error".

4.1 Quadrature Error and its Relation to $|\lambda_n|$

Suppose now that n is a positive integer, and that $f : [-1, 1] \to \mathbb{C}$ is an arbitrary bandlimited function (with band limit c). Suppose also that S_n is the PSWF-based quadrature rule of order n (see (101) in Definition 2). One of the principal goals of this paper is to investigate the quadrature error $\delta_n(f)$ defined via (103). The reader is referred to Section 7 for the results of several related numerical experiments.

The following theorem, illustrated in Table 1, provides an upper bound on $\delta_n(\psi_m)$, for any integer $m = 0, \ldots, n-1$. This theorem is illustrated in Table 3 and in Figure 5 (see Experiment 2 in Section 7.1); see also Conjecture 3 and Remark 37 in Section 7.1.

Theorem 14. *Suppose that c is a positive real number, and that*

$$c > 30. \tag{104}$$

Suppose also that $n > 0$ and $0 \leq m \leq n-1$ are integers, and that

$$n > \frac{2c}{\pi} + 5. \tag{105}$$

Suppose further that $\delta_n(\psi_m)$ is defined via (103). *Then,*

$$\delta_n(\psi_m) = \left| \int_{-1}^{1} \psi_m(s)\, ds - \sum_{j=1}^{n} W_j \cdot \psi_m(t_j) \right| \leq |\lambda_n| \cdot \left(24 \cdot \log\left(\frac{1}{|\lambda_n|} \right) + 6 \cdot \chi_n \right), \tag{106}$$

where λ_n, χ_n are those of (27), (38) *in Section 3.1, respectively.*

4.2 Quadrature Error and its Relation to n and c

In Theorem 14, we established an upper bound on the quadrature error $\delta_n(\psi_m)$ (see (103) and (106) in Theorem 14). However, this bound depends on χ_n and λ_n. In particular, it is not obvious how large n should be to make sure that the quadrature error does not exceed a prescribed $\varepsilon > 0$. In this subsection, we eliminate this inconvenience.

The following theorem is illustrated in Table 4 (see Experiment 3 in Section 7.1).

Theorem 15. *Suppose that c, ε are positive real numbers such that*

$$c > 30 \tag{107}$$

and

$$0 < \log\frac{1}{\varepsilon} < \frac{5\cdot\pi}{4\sqrt{6}}\cdot c - 3\cdot\log(c) - \log(6^5\cdot 14340). \tag{108}$$

Suppose also that the real numbers $\alpha, \nu(\alpha)$ are defined via the formulae

$$\alpha = \frac{4\sqrt{6}}{\pi}\cdot\left(\log\frac{1}{\varepsilon} + 3\cdot\log(c) + \log(6^5\cdot 14340)\right) \tag{109}$$

and

$$\nu(\alpha) = \frac{2c}{\pi} + \frac{\alpha}{2\pi}\cdot\log\left(\frac{16ec}{\alpha}\right), \tag{110}$$

respectively. Suppose furthermore that $n > 0$ and $0 \le m \le n-1$ are integers such that

$$n > \nu(\alpha), \tag{111}$$

and that $\delta_n(\psi_m)$ is defined via (103)*. Then,*

$$\delta_n(\psi_m) = \left|\int_{-1}^{1}\psi_m(s)\,ds - \sum_{j=1}^{n}\psi_m(t_j)W_j\right| < \varepsilon. \tag{112}$$

The following theorem is a direct consequence of Theorem 15. This theorem is one of the principal results of the paper. It is illustrated in Table 4 (see Experiment 3 in Section 7.1). See also Conjecture 3 in Section 7.1.

Theorem 16. *Suppose that c, ε are positive real numbers such that*

$$c > 60 \tag{113}$$

and

$$0 < \varepsilon < 1. \tag{114}$$

Suppose also that $n > 0$ and $0 \le m < n$ are integers, and that

$$n > \frac{2c}{\pi} + \left(10 + \frac{3}{2}\cdot\log(c) + \frac{1}{2}\cdot\log\frac{1}{\varepsilon}\right)\cdot\log\left(\frac{c}{2}\right). \tag{115}$$

Suppose furthermore that $\delta_n(\psi_m)$ is defined via (103) *in Definition 2. Then,*

$$\delta_n(\psi_m) = \left|\int_{-1}^{1}\psi_m(s)\,ds - \sum_{j=1}^{n}\psi_m(t_j)W_j\right| < \varepsilon. \tag{116}$$

4.3 Quadrature Weights

In this subsection, we analyze the weights of the quadrature rule S_n (see (100), (101) in Section 4). This analysis has two principal purposes. On the one hand, it provides the basis for a fast algorithm for the evaluation of the weights. On the other hand, it provides an explanation of some empirically observed properties of the weights.

The results of this subsection are illustrated in Table 5 and in Figure 6 (see Experiment 4 in Section 7.2).

The following theorem is instrumental for the evaluation of the quadrature weights $W_1, \ldots, W_n$ (see (100) in Definition 2).

Theorem 17. *Suppose that $n \geq 0$ is an integer, and that the function $\tilde{\Phi}_n : (-1, 1) \to \mathbb{R}$ is defined via the formula*

$$\tilde{\Phi}_n(t) = \sum_{k=0}^{\infty} \alpha_k^{(n)} \cdot Q_k(t), \tag{117}$$

where $Q_k(t)$ and $\alpha_k^{(n)}$ are defined, respectively, via (68), (69) and (57) in Section 3.2 (compare to (55) in Section 3.2). Then, for every integer $j = 1, \ldots, n$,

$$W_j = -\frac{2}{\psi_n'(t_j)} \sum_{k=0}^{\infty} \alpha_k^{(n)} \cdot Q_k(t_j) = -2 \cdot \frac{\tilde{\Phi}_n(t_j)}{\psi_n'(t_j)}, \tag{118}$$

where $t_1, \ldots, t_n$ and $W_1, \ldots, W_n$ are, respectively, the nodes and weights of the quadrature rule S_n in Definition 2.

Theorem 17 is illustrated in Table 5. We observe that Theorem 17 describes a connection between the weights $W_1, \ldots, W_n$ and the values of $\tilde{\Phi}_n$ at $t_1, \ldots, t_n$, where the function $\tilde{\Phi}_n$ is defined via (117).

The following theorem states that $\tilde{\Phi}_n$ satisfies a certain second-order non-homogeneous ODE, closely related to the prolate ODE (38) in Section 3.1. In particular, a recurrence relation between the derivatives of $\tilde{\Phi}_n$ of arbitrary order is established (compare to Theorem 8 in Section 3.1).

Theorem 18. *Suppose that $n \geq 0$ is an integer, and that the function $\tilde{\Phi}_n : (-1, 1) \to \mathbb{R}$ is defined via (117). Suppose also that the real numbers $\alpha_0^{(n)}, \alpha_1^{(n)}$ are defined via (57) in Section 3.2. Then,*

$$(1 - t^2) \cdot \tilde{\Phi}_n''(t) - 2t \cdot \tilde{\Phi}_n'(t) + (\chi_n - c^2 t^2) \cdot \tilde{\Phi}_n(t) = -c^2 \left(\alpha_0^{(n)} t + \alpha_1^{(n)} / 3 \right), \tag{119}$$

for all real $-1 < t < 1$. Also,

$$\left(1 - t^2\right) \cdot \tilde{\Phi}_n'''(t) - 4t \cdot \tilde{\Phi}_n''(t) + \left(\chi_n - c^2 t^2 - 2\right) \cdot \tilde{\Phi}_n'(t) - 2c^2 t \cdot \tilde{\Phi}_n(t) = -c^2 \alpha_0^{(n)}, \tag{120}$$

for all real $-1 < t < 1$. Finally,

$$\begin{aligned} &\left(1 - t^2\right) \tilde{\Phi}_n^{(k+2)}(t) - 2\left(k + 1\right) t \tilde{\Phi}_n^{(k+1)}(t) + \left(\chi_n - k\left(k + 1\right) - c^2 t^2\right) \tilde{\Phi}_n^{(k)}(t) \\ &\quad - c^2 k t \tilde{\Phi}_n^{(k-1)}(t) - c^2 k \left(k - 1\right) \tilde{\Phi}_n^{(k-2)}(t) = 0, \end{aligned} \tag{121}$$

for every integer $k \geq 2$ and all real $-1 < t < 1$ (compare to (47) in Section 3.1).

In the following theorem, we establish the positivity of the weights of the quadrature rule S_n in Definition 2.

Theorem 19. *Suppose that c is a positive real number, and that*

$$c > 30. \tag{122}$$

Suppose also that n is a positive integer, and that

$$n > \frac{2c}{\pi} + 5 \cdot \log(c) \cdot \log\left(\frac{c}{2}\right). \tag{123}$$

Suppose further that $W_1, \ldots, W_n$ are defined via (100)*. Then, for all integer $j = 1, \ldots, n$,*

$$W_j > 0. \tag{124}$$

Remark 13. *Extensive numerical experiments (see e.g. Table 5 and Figure 6) seem to indicate that the assumption* (123) *is unnecessary. In other words, the weights $W_1, \ldots, W_n$ are always positive, even for small values of n (at the present time we do not have the proof of this fact).*

Remark 14. *It was observed in [25], [26] that, if $1 \leq j, k \leq n$ are integers, then*

$$\left(\psi_n'(t_j)\right)^2 \cdot (1 - t_j^2) \cdot W_j = \left(\psi_n'(t_k)\right)^2 \cdot (1 - t_k^2) \cdot W_k + O\left(|\lambda_n|\right) \tag{125}$$

(see also Experiment 4 in Section 7.2). We observe that as $c \to 0$ the quadrature rule in Definition 2 converges to the well known Gaussian quadrature rule, whose nodes are the roots $t_1, \ldots, t_n$ of the Legendre polynomial P_n (see Section 3.2), and whose weights are defined via the formula

$$W_j = \frac{2}{\left(P_n'(t_j)\right)^2 \cdot \left(1 - t_j^2\right)}, \tag{126}$$

for every $j = 1, \ldots, n$ (see e.g. [1], Section 25.4). Thus, (125) *is not surprising.*

5 Numerical Algorithms

In this section, we describe several numerical algorithms for the evaluation of the PSWFs, certain related quantities, and the quadrature rules defined in Section 4. Throughout this section, the band limit $c > 0$ is a real number, and the prolate index $n \geq 0$ is a non-negative integer.

5.1 Evaluation of χ_n and $\psi_n(x)$, $\psi_n'(x)$ for $-1 \leq x \leq 1$

The use of the expansion of ψ_n into a Legendre series (see (55) in Section 3.2) for the evaluation of ψ_n in the interval $[-1, 1]$ goes back at least to the classical Bouwkamp algorithm (see [3]). More specifically, the coefficients $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ of the Legendre expansion are precomputed first (see (56), (57) in Section 3.2). These coefficients decay superalgebraically; in particular, relatively few terms of the infinite sum (55) are required to evaluate ψ_n to essentially machine precision (see Section 3.2, in particular Theorem 10 and Remark 9, and also [37] for more details).

5.1.1 Evaluation of χ_n and $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$

Suppose now that $n \geq 0$, and one is interested in evaluating the coefficients $\beta_0^{(m)}, \beta_1^{(m)}, \ldots$ in (55), for every integer $0 \leq m \leq n$. This can be achieved by solving two $N \times N$ symmetric tridiagonal eigenproblems, where N is of order n (see Theorem 10 and Remark 9 in Section 3.2, and also [37] for more details about this algorithm). In addition, this algorithm evaluates $\chi_0, \ldots, \chi_n$. Once this precomputation is done, for every integer $0 \leq m \leq n$ and for every real $-1 \leq x \leq 1$ one can evaluate $\psi_m(x)$ in $O(n)$ operations, by computing the sum (55) (see, however, Remark 21 below).

Suppose, on the other hand, that we are interested in a single PSWF only (as opposed to all the first n PSWFs). Obviously, we can use the algorithm above; however, its cost is $O(n^2)$ operations (see Remark 9 in Section 3.2). In the rest of this subsection, we describe a procedure for the evaluation of $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ and χ_n, whose cost is $O(n+c\log(c))$ operations.

This algorithm is also based on Theorem 10 in Section 3.2. It consists of two principal steps. First, we compute a low-accuracy approximation $\tilde{\chi}_n$ of χ_n, by means of Sturm Bisection (see Section 3.4.5, (66), (67) and Remark 9 in Section 3.2, and also [2]). Second, we compute χ_n and $\beta^{(n)}$ (see (65) and Remark 8 in Section 3.2) by means of the Shifted Inverse Power Method (see Section 3.4.4, and also [36], [5]). The Shifted Inverse Power Method requires an initial approximation to the eigenvalue; for this purpose we use $\tilde{\chi}_n$.

Below is a more detailed description of these two steps.

Step 1 (initial approximation $\tilde{\chi}_n$ of χ_n). Suppose that the infinite symmetric tridiagonal matrices A^{even} and A^{odd} are defined, respectively, via (63), (64) in Section 3.2. Suppose also that $A^{(n)}$ is the $N \times N$ upper left square submatrix of A^{even}, if n is even, or of A^{odd}, if n is odd.

Comment. N is an integer of order n (see Remark 9 in Section 3.2). The choice

$$N = 1.1 \cdot c + n + 1000 \tag{127}$$

is sufficient for all practical purposes.

- use Theorems 4, 5 and 6 in Section 3.1 to choose real numbers $x_0 < y_0$ such that

  $$x_0 < \chi_n < y_0. \tag{128}$$

  Comment. For a more detailed discussion of lower and upper bounds on χ_n, see, for example, [21], [22]. See also Remark 16 below.

- use Sturm Bisection (see Section 3.4.5) with initial values x_0, y_0 to compute $\tilde{\chi}_n$. On each step of Sturm Bisection, the Sturm sequence (see (95) in Theorem 13) is computed based on the matrix $A^{(n)}$ (see above).
  Comment. In principle, Sturm Bisection can be used to evaluate χ_n to machine precision. However, the convergence rate of Sturm Bisection is linear, and each iteration requires order n operations (see Remark 12 in Section 3.4.5). On the other hand, the convergence rate of the Shifted Inverse Power Method is cubic in the vicinity of the solution, while each iteration requires also order n operations (see Remark 11 in Section 3.4.4). Thus, we use Sturm Bisection to compute a low-order approximation

$\tilde{\chi_n}$ to χ_n, and then refine it by the Shifted Inverse Power Method to obtain χ_n to machine precision.

Remark 15. *The use of Sturm Bisection as a tool to compute the eigenvalues of a symmetric tridiagonal matrix goes back at least to [2]; in the context of PSWFs, it appears in [10].*

The cost analysis of Step 1 relies on the following observation based on Theorems 3, 4, 5, 6 in Section 3.1.

Observation 1. Suppose that $n \geq 0$ is an integer.

If $0 \leq n < 2c/\pi$, then

$$\chi_{n+1} - \chi_n = O(c). \tag{129}$$

If $n > 2c/\pi$, then

$$\chi_{n+1} - \chi_n = O(n). \tag{130}$$

Remark 16. *Due to Theorems 4, 5 in Section 3.1, the inequality*

$$n \cdot (n+1) < \chi_n < c^2 \tag{131}$$

holds for any real $c > 0$ and all integer $0 \leq n < 2c/\pi$. In this case, Step 1 requires $O(c \cdot \log(c))$ operations, due to the combination of (129), (131) *and Remark 12 in Section 3.4.5. On the other hand, if $n > 2c/\pi$, then the cost of Step 1 is $O(n)$ operations, due to the combination of Theorems 4, 6, Remark 12 in Section 3.4.5 and* (130).

Step 2 (evaluation of χ_n and $\beta^{(n)}$). Suppose now that $\tilde{\chi}_n$ is an approximation to χ_n evaluated in Step 1. Suppose also that the integer N is defined via (127) above (see also Remark 9 in Section 3.2).

- generate a pseudorandom vector $\tilde{\beta} \in \mathbb{R}^N$ of unit length.
  Comment. We use $\tilde{\chi}_n$ and $\tilde{\beta}$ as initial approximations to the eigenvalue χ_n and the corresponding eigenvector, respectively, for the Shifted Inverse Power Method (see Section 3.4.4).

- conduct Shifted Inverse Power Method iterations until χ_n is evaluated to machine precision. The corresponding eigenvector of unit length is denoted by $\hat{\beta}^{(n)}$.
  Comment. Each Shifted Inverse Power iteration costs $O(N)$ operations, and essentially $O(1)$ iterations are required (see Remark 11 in Section 3.4.4 for more details). In practice, in double precision calculations the number of iterations is usually between three and five.

Remark 17. *Clearly, the cost of Step 2 is $O(n)$ operations (see Remark 9 in Section 3.2 and Remark 11 in Section 3.4.4).*

Remark 18. *Suppose that the coordinates of the vector $\beta^{(n)} \in \mathbb{R}^N$ are defined via* (65) *(see also Remark 8 in Section 3.2). Then, $\hat{\beta}^{(n)}$ (evaluated in Step 2 above) approximates $\beta^{(n)}$ to*

essentially machine precision (this is a well known property of the Inverse Power Method; see Section 3.4.4, and also [36], [5] for more details). In other words,

$$\|\hat{\beta}^{(n)} - \beta^{(n)}\| \leq \varepsilon \cdot \|\beta^{(n)}\| = \varepsilon, \tag{132}$$

where ε is the machine accuracy (e.g. $\varepsilon \approx$ 1D-16 for double precision calculations). In addition, the eigenvalue χ_n is also evaluated to relative accuracy ε.

5.1.2 Evaluation of $\psi_n(x)$, $\psi_n'(x)$ for $-1 \leq x \leq 1$, given χ_n and $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$

Suppose now that χ_n and the coefficients $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ defined via (56) have already been evaluated. Suppose also that the integer N is defined via (127) above.

For any real $-1 \leq x \leq 1$, we evaluate $\psi_n(x)$ via the formula

$$\psi_n(x) = \sum_{k=0}^{2N} P_k(x) \cdot \alpha_k^{(n)} = \sum_{k=0}^{2N} P_k(x) \cdot \beta_k^{(n)} \cdot \sqrt{k + 1/2} \tag{133}$$

(compare to (55) in Section 3.2). Also, we evaluate $\psi_n'(x)$ via the formula

$$\psi_n'(x) = \sum_{k=1}^{2N} P_k'(x) \cdot \alpha_k^{(n)} = \sum_{k=0}^{2N} P_k'(x) \cdot \beta_k^{(n)} \cdot \sqrt{k + 1/2}. \tag{134}$$

Remark 19. *Due to the combination of Remark 9 in Section 3.2 and Remark 18 above, both $\psi_n(x)$ and $\psi_n'(x)$ are evaluated via* (133), (134) *essentially to machine precision, for any real $-1 \leq x \leq 1$ (also see [37] for more details).*

Remark 20. *Due to Remarks 16, 17 above, the cost of the evaluation of χ_n and $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ via Steps 1,2 is $O(n + c \log c)$ operations. Once this precomputation has been performed, the cost of each subsequent evaluation of $\psi_n(x)$, $\psi_n'(x)$, for any real $-1 \leq x \leq 1$, is $O(n)$ operations, according to* (133), (134) *and Remark 10 in Section 3.2.*

Remark 21. *Once χ_n and $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ have been evaluated, one does not have to use* (133), (134), *to compute $\psi_n(x)$, $\psi_n'(x)$ at an arbitrary point x in $[-1, 1]$. Instead, the cost of evaluating, say, $\psi_n(x)$ can be brought down from $O(n)$ to $O(1)$ (see Remark 29 in Section 5.3).*

5.2 Evaluation of λ_n

Suppose now that $n \geq 0$ is an integer, and that one needs to evaluate the eigenvalue λ_n of the integral operator F_c (see (26) in Section 3.1). Due to the combination of (26) and Theorem 1 in Section 3.1, if n is even, then $\psi_n(0) \neq 0$, and

$$\lambda_n = \frac{1}{\psi_n(0)} \int_{-1}^{1} \psi_n(t)\, dt; \tag{135}$$

for odd n,

$$\lambda_n = \frac{ic}{\psi_n'(0)} \int_{-1}^{1} t \cdot \psi_n(t)\, dt. \tag{136}$$

The formulae (135) and (136) provide an obvious way to calculate λ_n for even and odd n, respectively, via numerical integration. In fact, when $|\lambda_n|$ is relatively large, such procedure is quite satisfactory. More specifically, if $n < 2c/\pi$, then $|\lambda_n| \approx \sqrt{2\pi/c}$, and λ_n can be calculated via (135), (136) to high relative precision (see Theorems 2, 7 in Section 3.1 and Remark 19 in Section 5.1; see also [37] for more details). On the other hand, we observe that $\|\psi_n\|_{L^2[-1,1]} = 1$, due to Theorem 1 in Section 3.1. As a result, when $|\lambda_n|$ is small, the formulae (135), (136) are unsuitable for the evaluation of λ_n via numerical integration, due to catastrophic cancellation. For example, if $|\lambda_n| < \varepsilon$, where ε is the machine precision, the formulae (135), (136) produce no correct digits at all.

The standard way to overcome this obstacle for numerical evaluation of small λ_n's is to calculate all the ratios $\lambda_0/\lambda_1, \ldots, \lambda_n/\lambda_{n-1}$ (see, for example, [14], [33], [34]); this turns out to be a well-conditioned numerical procedure (see [37] for more details). Then, λ_0 is evaluated via (135) above, and the eigenvalues $\lambda_1, \ldots, \lambda_n$ are evaluated via the formula

$$\lambda_m = \lambda_0 \cdot \frac{\lambda_1}{\lambda_0} \cdot \ldots \cdot \frac{\lambda_m}{\lambda_{m-1}}, \tag{137}$$

for every integer $m = 1, \ldots, n$.

Suppose, on the other hand, that one is interested in a single λ_n only (as opposed to all the first n eigenvalues). Obviously, λ_n can be evaluated via (137) from the ratios λ_{j+1}/λ_j, as described above; however, it requires at least $O(n^2)$ operations (see [37]).

Unexpectedly, it turns out that λ_n can be obtained to high relative accuracy in $O(1)$ operations as a by-product of the algorithm described in Section 5.1. More specifically, suppose that the coefficients $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ are defined via (56). We combine (135), (136) above with (27), (51), (53), (56), (57) to make the following observation.

Observation 1. If n is even, then $\psi_n(0) \neq 0$, and

$$\lambda_n = \frac{1}{\psi_n(0)} \int_{-1}^{1} \psi_n(t)\, dt = \frac{\beta_0^{(n)}\sqrt{2}}{\psi_n(0)}. \tag{138}$$

If n is odd, then $\psi_n'(0) \neq 0$, and

$$\lambda_n = \frac{ic}{\psi_n'(0)} \int_{-1}^{1} t \cdot \psi_n(t)\, dt = \sqrt{\frac{2}{3}} \cdot \frac{ic\beta_1^{(n)}}{\psi_n'(0)}. \tag{139}$$

Remark 22. *Obviously, the cost of evaluating λ_n from $\psi_n(0), \beta_0^{(n)}$ via* (138) *(for even n) or from $\psi_n'(0), \beta_1^{(n)}$ via* (139) *(for odd n) is $O(1)$ operations.*

Remark 23. *Due to Remarks 20, 22 and* (138), (139)*, a single λ_n can be evaluated as a by-product of the procedure described in Section 5.1, at the total cost of $O\left(n + c\log(c)\right)$ operations.*

Remarks 22, 23 describe the *cost* of the evaluation of λ_n via (138), (139). To describe the *accuracy* of this procedure, we start with the following observation.

Observation 2. Due to Remark 19, λ_n is evaluated to the same relative accuracy as $\beta_0^{(n)}$ (for even n) or as $\beta_1^{(n)}$ (for odd n). According to (132) in Remark 18, the algorithm of Section 5.1 evaluates the vector $\beta^{(n)}$ to relative accuracy ε, where ε is the machine precision.

However, this means that a single **coordinate** of $\beta^{(n)}$ is only guaranteed to be evaluated to **absolute** accuracy ε. More specifically, the inequality

$$\left| \frac{\beta_k^{(n)} - \hat{\beta}_k^{(n)}}{\beta_k^{(n)}} \right| \leq \frac{\varepsilon}{\left| \beta_k^{(n)} \right|} \tag{140}$$

holds for every integer $k = 0, \ldots, N$, where N is defined via (127) in Section 5.1, and $\hat{\beta}_k^{(n)}$ is the numerical approximation to $\beta_k^{(n)}$. In general, the inequality (140) can be rather tight; as a result, if, for example, $|\beta_0^{(n)}| \leq \varepsilon/10$, then, apriori, we cannot expect $\hat{\beta}_0^{(n)}$ to approximate $\beta_0^{(n)}$ to any digit at all!

In practical computations, it is sometimes desirable to evaluate extremely small λ_n's (e.g. $|\lambda_n| \approx$ 1D-50). Observation 2 seems to suggest that, in such cases, the evaluation of λ_n via the procedure described above is futile due to disastrous loss of accuracy.

Fortunately, it turns out that the algorithm described in Section 5.1 **always** evaluates $\beta_0^{(n)}, \beta_1^{(n)}$ to high relative accuracy, regardless of how small they are. This is a consequence of a more general (and somewhat surprising!) phenomenon studied in detail in [27], [28]. We summarize the corresponding results in the following theorem.

Theorem 20. *For a certain class of real symmetric tridiagonal matrices, the coordinates of their eigenvectors are defined to high relative precision. Moreover, the matrices A^{even}, A^{odd} defined, respectively, via* (63), (64) *in Section 3.2, belong to this class.*

In the following theorem, we summarize implications of Theorem 20 for the evaluation of $\beta_0^{(n)}, \beta_1^{(n)}$ via the algorithm in Section 5.1 (the proof of a slightly modified version of this theorem appears in [27], [28]).

Theorem 21. *Suppose that $c > 0$ is a real number, that $n \geq 0$ is an integer, and that $\beta_0^{(n)}, \beta_1^{(n)}$ are defined via* (56) *in Section 3.2. Then, the algorithm described in Section 5.1 evaluates $\beta_0^{(n)}, \beta_1^{(n)}$ to high relative accuracy. More specifically,*

$$\left| \frac{\beta_0^{(n)} - \hat{\beta}_0^{(n)}}{\beta_0^{(n)}} \right| \leq 10 \cdot \varepsilon \cdot c \tag{141}$$

for even n, and

$$\left| \frac{\beta_1^{(n)} - \hat{\beta}_1^{(n)}}{\beta_1^{(n)}} \right| \leq 10 \cdot \varepsilon \cdot c \tag{142}$$

for odd n, where $\hat{\beta}_0^{(n)}, \hat{\beta}_1^{(n)}$ are the numerical approximation to $\beta_0^{(n)}, \beta_1^{(n)}$, respectively, and ε is the machine accuracy (e.g. $\varepsilon \approx$ 1D-16 for double precision calculations).

Remark 24. *The algorithm described in Section 5.1 evaluates the eigenvectors $\beta^{(n)}$ by the Shifted Inverse Power Method (see Section 3.4.4). It turns out that the choice of method is important in this situation: if, for example, these eigenvectors are evaluated via the standard and well known Jacobi Rotations (rather than Inverse Power), the small coordinates exhibit the loss of accuracy expected from* (140) *(see [27], [28] for more details about this and related issues).*

Remark 25. *Due to the combination of Remark 19 in Section 5.1, Observation 2 above and Theorem 21, the algorithm of this section evaluates λ_n to high relative accuracy. More specifically, at most $1 + \log_{10}(c)$ decimal digits are lost in the evaluation of λ_n.*

5.3 Evaluation of the Quadrature Nodes

Suppose that $n > 0$ is an integer, and that the quadrature rule S_n is defined via (101) in Section 4. According to (98), the nodes of S_n are precisely the n roots $t_1, \ldots, t_n$ of ψ_n in the interval $(-1, 1)$.

In this section, we describe a numerical procedure for the evaluation of these quadrature nodes. This procedure is based on the fast algorithm for the calculation of the roots of special functions described in [8]. It combines Prüfer's transformation (see Section 3.3), Runge-Kutta method (see Section 3.4.3) and Taylor's method (see Section 3.4.2). This algorithm also evaluates $\psi_n'(t_1), \ldots, \psi_n'(t_n)$. It requires $O(n)$ operations to compute all roots of ψ_n in $(-1, 1)$ as well as the derivative of ψ_n at these roots.

A short outline of the principal steps of the algorithm is provided below. For a more detailed description of the algorithm and its properties, the reader is referred to [8].

Suppose that $t_{\min}$ is the minimal root of ψ_n in $[0, 1)$.

Step 1 (evaluation of $t_{\min}$). If n is odd, then

$$t_{\min} = t_{(n+1)/2} = 0, \tag{143}$$

due to Theorem 1 in Section 3.1. On the other hand, if n is even, then

$$t_{\min} = t_{(n+2)/2} > 0. \tag{144}$$

To compute $t_{\min}$ in the case of even n, we numerically solve the ODE (81) with the initial condition (83) in the interval $[\pi n/2, \pi \cdot (n+1)/2]$, by using 20 steps of Runge-Kutta method described in Section 3.4.3. The rightmost value $\tilde{t}_{\min}$ of the solution is a low-order approximation of $t_{\min} = t_{(n+2)/2}$ (see (82), (144)). Then, we evaluate $t_{\min}$ to machine precision via Newton's method (see Section 3.4.1), using $\tilde{t}_{\min}$ as an initial approximation to $t_{\min}$. On each Newton iteration, we evaluate ψ_n and ψ_n' by using the algorithm of Section 5.1 (see (133), (134)).

Observation 1. The point $\tilde{t}_{\min}$ approximates $t_{\min}$ to at least three decimal digits (see Section 3.4.3). Since Newton's method converges quadratically in the vicinity of the solution, only several Newton iterations are required to obtain $t_{\min}$ from $\tilde{t}_{\min}$ to essentially machine precision (see [8] for more details). In our experience, the number of Newton iterations in this step never exceeds four in double precision calculations (and never exceeds six in extended precision calculations). We combine this observation with Remark 20 in Section 5.1 to conclude that the total cost of Step 1 is $O(n)$ operations.

Step 2 (evaluation of $\psi_n'(t_{\min})$). We evaluate $\psi_n'(t_{\min})$ to machine precision via (134) in Section 5.1.

Observation 2. Due to Remark 20 in Section 5.1, the cost of Step 2 is $O(n)$ operations.

The remaining roots of ψ_n in $(t_{\min}, 1)$ are computed one by one, as follows. Suppose that $n/2 < j < n$ is an integer, and both t_j and $\psi_n'(t_j)$ have already been evaluated.

Step 3 (evaluation of t_{j+1} and $\psi_n'(t_{j+1})$, given t_j and $\psi_n'(t_j)$).

- evaluate $\psi_n^{(2)}(t_j), \ldots, \psi_n^{(M)}(t_j)$ via the recurrence relation (47) in Section 3.1 (in double precision calculations, $M = 30$; in extended precision calculations, $M = 60$).
- use 20 steps of Runge-Kutta method (see Section 3.4.3), to solve the ODE (81) with the initial condition

$$s\left(\pi \cdot \left(j - \frac{1}{2}\right)\right) = t_j \tag{145}$$

  in the interval $[\pi \cdot (j - 1/2), \pi \cdot (j + 1/2)]$ (see (82)). The rightmost value $\tilde{t}_{j+1}$ of the solution is a low-order approximation of t_{j+1}.
- compute t_{j+1} via Newton's method (see Section 3.4.1), using $\tilde{t}_{j+1}$ as the initial approximation to t_{j+1}. On each Newton iteration, we evaluate ψ_n and ψ_n' via Taylor's method (see Section 3.4.2). The Taylor expansion of appropriate order M about t_j is used, i.e.

$$\psi_n(t) = \sum_{k=0}^{M} \frac{\psi_n^{(k)}(t_j)}{k!} \cdot (t - t_j)^k + O\left((t - t_j)^{M+1}\right). \tag{146}$$

- evaluate $\psi_n'(t_{j+1})$ via Taylor's method. The Taylor expansion of order $M - 1$ is used, i.e.

$$\psi_n'(t_{j+1}) = \sum_{k=0}^{M-1} \frac{\psi_n^{(k+1)}(t_j)}{k!} \cdot (t_{j+1} - t_j)^k + O\left((t_{j+1} - t_j)^M\right). \tag{147}$$

  In both (146) and (147), we set $M = 30$ for double precision calculations, and $M = 60$ for extended precision calculations.

Observation 3. The point $\tilde{t}_{j+1}$ approximates t_{j+1} to at least three decimal digits (see Section 3.4.3). Subsequently, only several Newton iterations are required to obtain t_{j+1} to essentially machine precision (see Observation 1 above, and also [8] for more details). Thus the cost of Step 3 is $O(1)$ operations, for every integer $n/2 < j < n$.

Remark 26. *Obviously, on each Newton iteration one can evaluate ψ_n and ψ_n' via* (133), (134) *in Section 5.1 rather than via* (146), (147). *However, this would increase the cost of each such evaluation from $O(1)$ to $O(n)$, and the total cost of the procedure from $O(n)$ to $O(n^2)$ (see Remark 20 in Section 5.1).*

Step 4 (evaluation of t_j and $\psi_n'(t_j)$ for all $j \le n/2$). Step 3 is repeated for every integer $n/2 < j < n$. To evaluate t_j and $\psi_n'(t_j)$ for $-1 < t_j < 0$, we use the symmetry of ψ_n about zero (see Theorem 1 in Section 3.1). More specifically, for every integer $1 \le j \le n/2$, we compute t_j and $\psi_n'(t_j)$, respectively, via the formulae

$$t_j = t_{n+1-j} \tag{148}$$

and

$$\psi_n'(t_j) = (-1)^{n+1} \cdot \psi_n'(t_{n+1-j}). \tag{149}$$

Summary (evaluation of t_j and $\psi_n'(t_j)$, for all $j = 1, \ldots, n$). To summarize, the procedure for the evaluation of all roots of ψ_n in $(-1, 1)$ (as well as the derivative of ψ_n at these roots) is as follows:

- Evaluate $t_{\min}$ defined via (143), (144) (see Step 1). Cost: $O(n)$ operations.
- Evaluate $\psi_n'(t_{\min})$ (see Step 2). Cost: $O(n)$ operations.
- For every integer $n/2 < j < n$, evaluate t_{j+1} and $\psi_n'(t_{j+1})$ (see Step 3). Cost: $O(n)$ operations.
- For every integer $1 \leq j \leq n/2$, evaluate t_j and $\psi_n'(t_j)$ (see Step 4). Cost: $O(n)$ operations.

Remark 27. *We observe that the algorithm described in this section not only computes the roots $t_1, \ldots, t_n$ of ψ_n in $(-1, 1)$, but also evaluates ψ_n' at all these roots. The total cost of this algorithm is $O(n)$ operations, and all the quantities are evaluated essentially to machine precision (see Observations 1,2,3 above).*

Remark 28. *The algorithm described in this section uses the quantities χ_n and $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ computed via the procedure of Section 5.1. If $n < 2c/\pi$, then these quantities are obtained at the cost of $O(n + c\log(c))$ operations; if $n > 2c/\pi$, then these quantities are obtained at the cost of $O(n)$ operations (see Remarks 16, 20 in Section 5.1).*

Remark 29. *As a by-product of the algorithm described in this section, we obtain a table of all the derivatives of ψ_n up to order M at all roots of ψ_n in $(-1, 1)$ (here $M = 30$ in double precision calculation, and $M = 60$ in extended precision calculations). In other words, $\psi_n^{(k)}(t_j)$ are calculated for every $k = 1, \ldots, M$ and every $j = 1, \ldots, n$ (see Step 3 above). This table can be used to evaluate $\psi_n(x), \psi_n'(x)$ at an arbitrary point $t_1 \leq x \leq t_n$ to essentially machine precision in $O(1)$ operations via interpolation, using the formulae* (146), (147) *(see also Remark 21 in Section 5.1).*

5.4 Evaluation of the Quadrature Weights

Suppose now that $n > 0$ is an integer, and that the quadrature rule S_n is defined via (101) in Section 4. In this subsection, we describe an algorithm for the evaluation of the weights $W_1, \ldots, W_n$ of this quadrature rule (see (100) in Section 4). The results of this subsection are illustrated in Table 5 and in Figure 6 (see Experiment 4 in Section 7.2).

In the description of the algorithms below, we assume that the coefficients $\beta_0^{(n)}, \beta_1^{(n)}, \ldots$ (defined via (56) in Section 3.2) have already been evaluated (for example, by the algorithm in Section 5.1). In addition, we assume that the quadrature nodes $t_1, \ldots, t_n$ as well as $\psi_n'(t_1), \ldots, \psi_n'(t_n)$ have also been computed (for example, by the algorithm of Section 5.3).

An obvious way to compute $W_1, \ldots, W_n$ is to evaluate (100) numerically. However, due to (99), the integrand φ_j in (100) has $n - 1$ roots in $(-1, 1)$, for every $j = 1, \ldots, n$. In particular, such approach is unlikely to require less that $O(n^2)$ operations.

Rather than computing (100) directly, we evaluate $W_1, \ldots, W_n$ by using the results of Section 4.3. In the rest of this subsection, we describe two such algorithms; both evaluate $W_1, \ldots, W_n$ essentially to machine precision. One of these algorithms (based on Theorem 17)

is fairly straightforward; however, its cost is $O(n^2)$ operations. The other algorithm (based on Theorem 18), while still rather simple, is also computationally efficient: its cost is $O(n)$ operations.

Algorithm 1: evaluation of $W_1, \ldots, W_n$ in $O(n^2)$ operations. Suppose that the integer N is defined via (127) in Section 5.1. For every integer $j = 1, \ldots, n$, we compute an approximation $\widetilde{W}_j$ to W_j via the formula

$$\widetilde{W}_j = -\frac{2}{\psi_n'(t_j)} \sum_{k=0}^{2N} \alpha_k^{(n)} \cdot Q_k(t_j) = -\frac{2}{\psi_n'(t_j)} \sum_{k=0}^{2N} \beta_k^{(n)} \cdot Q_k(t_j) \cdot \sqrt{k+1/2}, \tag{150}$$

where $Q_k(t)$ and $\alpha_k^{(n)}$ are defined, respectively, via (68), (69) and (57) in Section 3.2. We observe that (150) is obtained from the identity (118) in Theorem 17 in Section 4.3 by truncating the infinite series at $2N$ terms.

Remark 30. *Due to the combination of Remarks 9, 10 in Section 3.2, Remark 18 in Section 5.1,* (127) *and Theorem 17, each weight W_j is evaluated via* (150) *essentially to machine precision (see also Experiment 4 in Section 7.2).*

Remark 31. *Due to the combination of Remark 10 in Section 3.2 and* (127) *in Section 5.1, the overall cost of computing $W_1, \ldots, W_n$ via* (150) *is $O(n^2)$ operations.*

Algorithm 2: evaluation of $W_1, \ldots, W_n$ in $O(n)$ operations. This algorithm is somewhat similar to the procedure for the evaluation of the roots of ψ_n in $(-1, 1)$ described in Section 5.3.

Suppose first that $t_{\min}$ is the minimal root of ψ_n in $[0, 1)$. In other words,

$$t_{\min} = \begin{cases} t_{(n+1)/2} = 0 & \text{if } n \text{ is odd,} \\ t_{(n+2)/2} > 0 & \text{if } n \text{ is even} \end{cases} \tag{151}$$

(see (143), (144) in Section 5.3). Suppose also that the function $\tilde{\Phi}_n : (-1, 1) \to \mathbb{R}$ is defined via (117) in Theorem 17 in Section 4.3.

Step 1 (evaluation of $\tilde{\Phi}_n(t_{\min})$ and $\tilde{\Phi}_n'(t_{\min})$). We evaluate $\tilde{\Phi}_n(t_{\min})$ and $\tilde{\Phi}_n'(t_{\min})$ via the formulae

$$\tilde{\Phi}_n(t_{\min}) = \sum_{k=0}^{2N} \alpha_k^{(n)} \cdot Q_k(t_{\min}) = \sum_{k=0}^{2N} \beta_k^{(n)} \cdot Q_k(t_{\min}) \cdot \sqrt{k+1/2} \tag{152}$$

and

$$\tilde{\Phi}_n'(t_{\min}) = \sum_{k=0}^{2N} \alpha_k^{(n)} \cdot Q_k'(t_{\min}) = \sum_{k=0}^{2N} \beta_k^{(n)} \cdot Q_k'(t_{\min}) \cdot \sqrt{k+1/2}, \tag{153}$$

respectively (see (150) in the description of Algorithm 1 above). Observe that (152), (153) are obtained from the infinite expansion (117) in Theorem 17 by truncation.

Remark 32. *Due to Remarks 30, 31, the cost of Step 1 is $O(n)$ operations; moreover, $\tilde{\Phi}_n(t_{\min})$ and $\tilde{\Phi}'_n(t_{\min})$ are evaluated via* (152), (153) *essentially to machine precision.*

We evaluate $\tilde{\Phi}_n$ at all but the last four remaining roots of ψ_n in $[0,1)$ as follows. Suppose that $n/2 < j < n$ is an integer, and both $\tilde{\Phi}_n(t_j)$ and $\tilde{\Phi}'_n(t_j)$ have already been evaluated.

Step 2 (evaluation of $\tilde{\Phi}_n(t_{j+1})$ and $\tilde{\Phi}'_n(t_{j+1})$, given $\tilde{\Phi}_n(t_j)$ and $\tilde{\Phi}'_n(t_j)$).

- use the recurrence relation (120), (121) (see Theorem 18 in Section 4.3) to evaluate $\tilde{\Phi}_n^{(2)}(t_j), \ldots, \tilde{\Phi}_n^{(M)}(t_j)$ (here $M = 60$ in double precision calculations, and $M = 120$ in extended precision calculations).

- evaluate $\tilde{\Phi}_n(t_{j+1})$ via Taylor's method (see Section 3.4.2). The Taylor expansion of appropriate order M is used, i.e.

$$\tilde{\Phi}_n(t_{j+1}) = \sum_{k=0}^{M} \frac{\tilde{\Phi}_n^{(k)}(t_j)}{k!} \cdot (t_{j+1} - t_j)^k + O\left((t_{j+1} - t_j)^{M+1}\right) \tag{154}$$

  (compare to (146) in Section 5.3).

- evaluate $\tilde{\Phi}'_n(t_{j+1})$ via Taylor's method. The Taylor expansion of order $M-1$ is used, i.e.

$$\tilde{\Phi}'_n(t_{j+1}) = \sum_{k=0}^{M-1} \frac{\tilde{\Phi}_n^{(k+1)}(t_j)}{k!} \cdot (t_{j+1} - t_j)^k + O\left((t_{j+1} - t_j)^{M}\right) \tag{155}$$

  (compare to (147) in Section 5.3). In both (154) and (155), we set $M = 60$ for double precision calculations and $M = 120$ for extended precision calculations.

Remark 33. *For each j, the cost of Step 2 is $O(1)$ operations (i.e. does not depend on n). Also, it turns out that $\tilde{\Phi}_n(t_j)$ and $\tilde{\Phi}'_n(t_j)$ are evaluated via* (154), (155) *respectively, essentially to machine precision (compare to* (146), (147) *in Section 5.3). For a detailed discussion of the accuracy and stability of this step, the reader is referred to [8].*

Step 3 (evaluation of $\tilde{\Phi}_n(t_j)$ for $n-3 \le j \le n$). For $j = n-3, n-2, n-1, n$, we evaluate $\tilde{\Phi}_n(t_j)$ via the formula

$$\tilde{\Phi}_n(t_j) = \sum_{k=0}^{2N} \alpha_k^{(n)} \cdot Q_k(t_j) = \sum_{k=0}^{2N} \beta_k^{(n)} \cdot Q_k(t_j) \cdot \sqrt{k+1/2} \tag{156}$$

(as in (152) in Step 1; see also (150) in the description of Algorithm 1 above).

Remark 34. *We compute $\tilde{\Phi}_n$ at the last four nodes via* (156) *rather than* (154)*, since the accuracy of the latter deteriorates when t_j is too close to* 1 *(interestingly, the evaluation of $\psi_n(t_j)$ via* (146) *in Section 5.3 for any $j = 1, \ldots, n$ does not have this unpleasant feature). Since this approach works in practice, is cheap in terms of the number of operations and eliminates the accuracy problem, there was no need in a detailed analysis of the issue (see, however, [8] for more details).*

Step 4 (evaluation of $\tilde{\Phi}_n(t_j)$ for $1 \leq j \leq n/2$). Due to the combination of Theorem 17 in Section 4.3 and (69) in Section 3.2, the function $\tilde{\Phi}_n$ is symmetric about the origin. We use this observation to evaluate $\tilde{\Phi}_n(t_j)$ via the formula

$$\tilde{\Phi}_n(t_j) = (-1)^{n+1} \cdot \tilde{\Phi}_n(t_{n+1-j}), \tag{157}$$

for every $j = 1, 2, \ldots, n/2$.

Step 5 (evaluation of $W_1, \ldots, W_n$). For every $j = 1, \ldots, n$, we compute an approximation $\widehat{W}_j$ to W_j from $\tilde{\Phi}_n(t_j)$ and $\psi_n'(t_j)$ via the formula

$$\widehat{W}_j = -2 \cdot \frac{\tilde{\Phi}_n(t_j)}{\psi_n'(t_j)} \tag{158}$$

(see (118) in Theorem 17 in Section 4.3).

Remark 35. *Due to the combination of Remarks 32, 33, 34, Algorithm 2 evaluates all $W_1, \ldots, W_n$ essentially to machine precision. This algorithms requires $O(n)$ operations (compare to Remark 31).*

Remark 36. *Algorithm 2 described in this section uses some of the quantities evaluated by the procedures of Sections 5.1, 5.3. If $n < 2c/\pi$, then the cost of obtaining these quantities is $O(n + c\log(c))$ operations; if $n > 2c/\pi$, then the cost of obtaining these quantities is $O(n)$ operations (see Remarks 27, 28 in Section 5.3).*

6 Numerical Results

In this section, we demonstrate the performance of the quadrature rules from Section 4. All the calculations were implemented in FORTRAN (the Lahey 95 LINUX version), and carried out in double precision. Extended precision calculations were used for comparison and verification (in extended precision, the floating point numbers are 128 bits long, as opposed to 64 bits in double precision).

Experiment 1. Here we demonstrate the performance of the quadrature rule S_n (see (101) in Section 4) on exponential functions. We proceed as follows. We choose, more or less arbitrarily, the band limit c and the prolate index n. Next, we evaluate the quadrature nodes $t_1, \ldots, t_n$ and the quadrature weights $W_1, \ldots, W_n$ via the algorithms of Sections 5.3, 5.4, respectively. Also, we evaluate $|\lambda_n|$ via the algorithm in Section 5.2. Then, we choose a real number $0 \leq a \leq 2$, and evaluate the integral of e^{icax} over $-1 \leq x \leq 1$ via the formula

$$\int_{-1}^{1} e^{iacx}\, dx = \int_{-1}^{1} \cos(acx)\, dx = \frac{2\sin(ac)}{ac}. \tag{159}$$

Also, we use S_n to approximate (159) via the formula

$$\int_{-1}^{1} e^{iacx}\, dx \approx \sum_{j=1}^{n} e^{icat_j} \cdot W_j \tag{160}$$

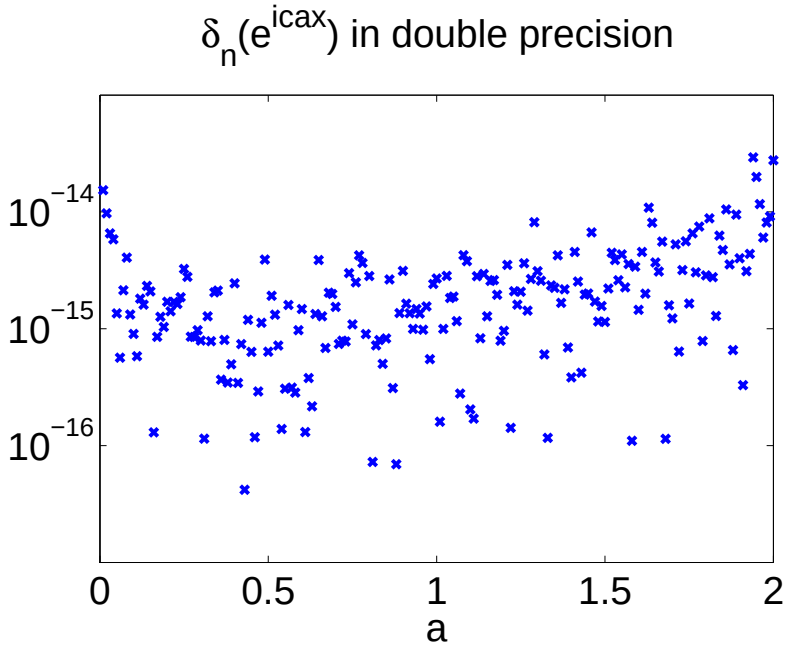

Figure 1: *The quadrature error vs* $|\lambda_n|$, *with* $c = 1000$ *and* $n = 682$. *Here* λ_n = -.60352E-15.

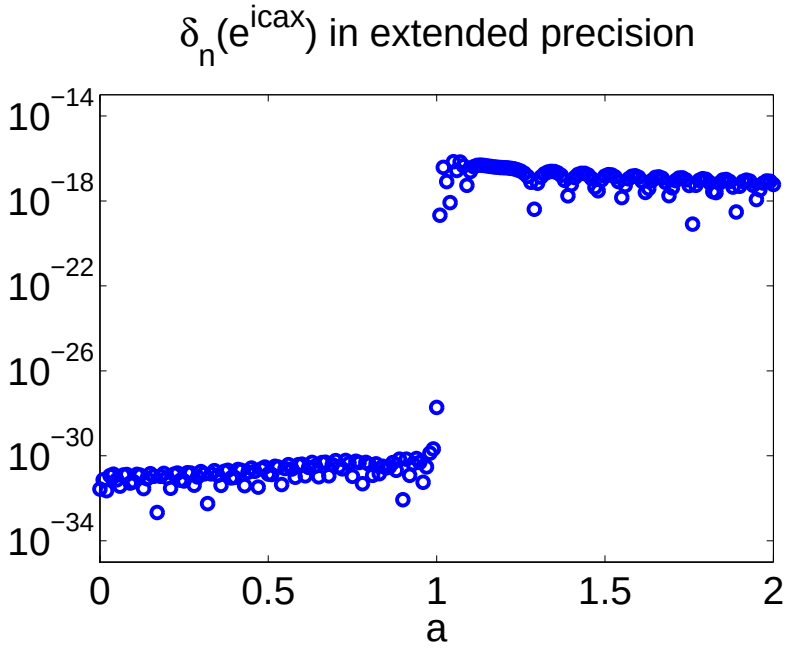

Figure 2: *The quadrature error vs* $|\lambda_n|$, *with* $c = 1000$ *and* $n = 682$. *Here* λ_n = -.60352E-15.

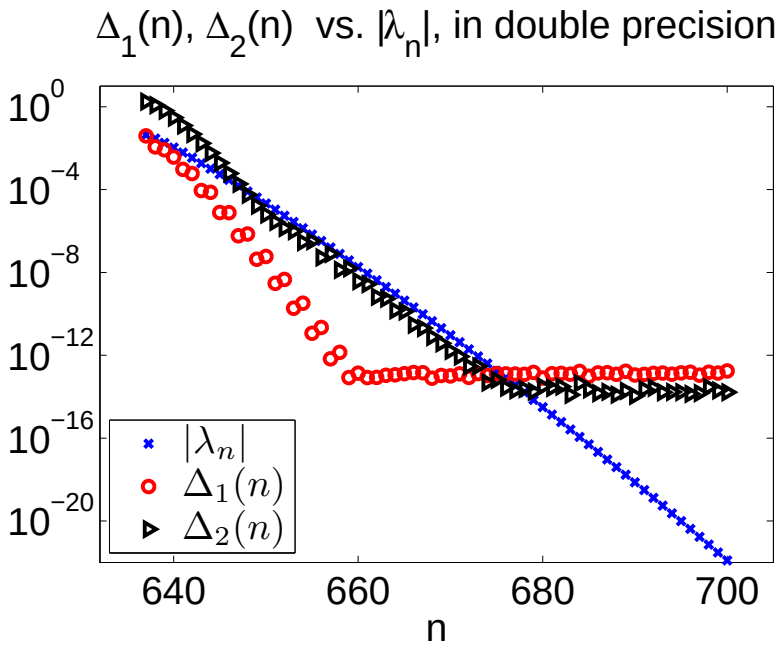

Figure 3: *The maximal quadrature errors* $\Delta_1(n), \Delta_2(n)$ *vs* $|\lambda_n|$, *with* $c = 1000$.

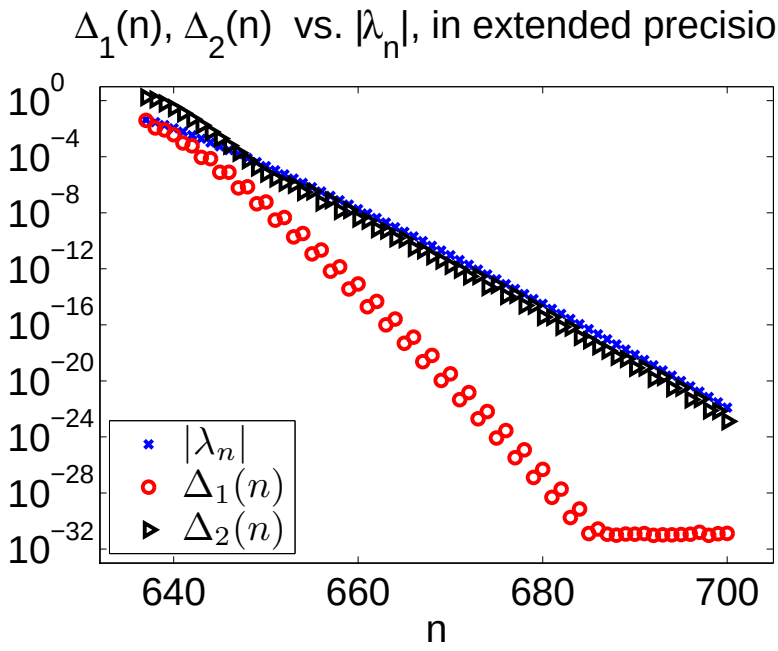

Figure 4: *The maximal quadrature errors* $\Delta_1(n), \Delta_2(n)$ *vs* $|\lambda_n|$, *with* $c = 1000$.

(see (102) in Section 4). Finally, we evaluate the quadrature error $\delta_n(e^{iacx})$ via the formula

$$\delta_n(e^{iacx}) = \left| \frac{2\sin(ac)}{ac} - \sum_{j=1}^{n} e^{icat_j} \cdot W_j \right| \tag{161}$$

(see (103) in Section 4).

In Figure 1, we display the results of this experiment. The band limit and the prolate index were chosen to be, respectively, $c = 1000$ and $n = 682$. For this choice of parameters, λ_n = -.60352E-15. In this figure, we plot the quadrature error (161) as a function of the real parameter a, for $0 \le a \le 2$, on the logarithmic scale. The calculations are carried out in double precision.

We make the following observations from Figure 1. The quadrature error is essentially zero up to machine precision ε, for all real $0 \le a \le 2$. In other words, for this choice of parameters, the quadrature rule S_n integrates the functions of the form $f(x) = e^{icax}$ with $0 \le a \le 1$ exactly, for all practical purposes. It is perhaps surprising, however, that such functions are integrated exactly via S_n even when $1 < a \le 2$. In other words, the quadrature rule S_n (corresponding to band limit c and $|\lambda_n| \approx \varepsilon$) integrates exactly the exponential functions with the band limit up to $2c$.

To get a clearer picture, we repeat this experiment in extended precision. In Figure 2, we plot the quadrature error (161) as a function of the real parameter a, for $0 \le a \le 2$, on the logarithmic scale. In other words, Figure 2 is a version of Figure 1 in extended precision.

We make the following observations from Figure 2. If $0 \le a \le 1$, then the quadrature rule S_n integrates the functions of the form $f(x) = e^{icax}$ up to the error of order $|\lambda_n|^2$ (in Figure 1 we used double precision calculations and thus did not have enough digits to see this phenomenon). On the other hand, for $1 < a \le 2$ the quadrature rule S_n integrates such functions up to the error roughly $|\lambda_n|$. In other words, the quadrature rule S_n (corresponding to band limit c and $|\lambda_n| \approx \varepsilon$) integrates the functions of band limit up to c up to ε^2 (rather than ε); on the other hand, the functions of band limit between c and $2c$ are integrated up to ε.

Explanation. These observations admit the following (somewhat imprecise) explanation (see [25], [26] for more details). Suppose that $a \ge 0$ is a real number. Due to (27) and Theorem 1 in Section 3.1,

$$e^{iacx} = \sum_{m=0}^{\infty} \lambda_m \psi_m(a) \psi_m(x), \tag{162}$$

for all real $-1 \le x \le 1$. Moreover,

$$\int_{-1}^{1} e^{iacx}\, dx = \frac{2\sin(ac)}{ac} = \sum_{m=0}^{\infty} \lambda_m^2 \psi_m(a) \psi_m(0). \tag{163}$$

We combine (161), (162), (163) to obtain

$$\frac{2\sin(ac)}{ac} - \sum_{j=1}^{n} W_j \cdot e^{icat_j} = \sum_{m=0}^{\infty} \lambda_m \psi_m(a) \left(\lambda_m \psi_m(0) - \sum_{j=1}^{n} W_j \psi_m(t_j) \right). \tag{164}$$

Obviously, the quadrature error $\delta_n(\psi_m)$ (see (173)) is zero for odd m. Also, $\delta_n(\psi_m)$ rapidly increases as a function of even $0 \leq m < n$; moreover, $\delta_n(\psi_m)$ is of order $|\lambda_n|$ when $m < n$ is an even integer close to n (see Conjectures 2, 3 in Section 7.1 and Theorem 14 in Section 4.1). Therefore, roughly speaking,

$$\sum_{m=0}^{n-1} \lambda_m \psi_m(a) \left(\lambda_m \psi_m(0) - \sum_{j=1}^{n} W_j \psi_m(t_j) \right) \approx |\lambda_n|^2 \cdot \psi_{n-1}(a). \tag{165}$$

On the other hand, due to the fast decay of $|\lambda_m|$ (see Theorems 2, 7 in Section 3.1),

$$\sum_{m=n}^{\infty} \lambda_m \psi_m(a) \left(\lambda_m \psi_m(0) - \sum_{j=1}^{n} W_j \psi_m(t_j) \right) \approx |\lambda_n|^2. \tag{166}$$

Finally, the following approximate formula appears in [25], [26], in a slightly different form: suppose that $n > 0$ is an integer, that $\chi_n > c^2$, and that $0 \leq a \leq 2$ is a real number. Then,

$$|\psi_n(a)| = \begin{cases} O\left(\sqrt{n}\right), & 0 \leq a \leq 1, \\ O\left(|\lambda_n|^{-1}\right), & 1 < a \leq 2. \end{cases} \tag{167}$$

It follows from the combination of (165), (166), (167) that the quadrature error (161) is expected to be of the order $|\lambda_n|^2 \cdot \sqrt{n}$, if $0 \leq a \leq 1$. On the other hand, the quadrature error (161) is expected to be of the order $|\lambda_n|$, if $1 < a \leq 2$. Figures 1, 2, 3, 4 support these somewhat vague conclusions.

We summarize this crude analysis, supported by the observations above, in the following conjecture about the quadrature error (161) for $0 \leq a \leq 2$.

Conjecture 1. *Suppose that $c > 0$ and $a \geq 0$ are real numbers, and that $n > 2c/\pi$ is an integer. Suppose also that $\delta_n(e^{icax})$ is defined via* (103) *in Definition 2 in Section 4. If $0 \leq a \leq 1$, then*

$$\delta_n\left(e^{icax}\right) = \left| \int_{-1}^{1} e^{icax}\, dx - \sum_{j=1}^{n} e^{icat_j} \cdot W_j \right| \approx |\lambda_n|^2 \cdot \sqrt{n}, \tag{168}$$

where λ_n is that of (27) *in Section 3.1. If, on the other hand, $1 < a \leq 2$, then*

$$\delta_n\left(e^{icax}\right) = \left| \int_{-1}^{1} e^{icax}\, dx - \sum_{j=1}^{n} e^{icat_j} \cdot W_j \right| \approx |\lambda_n|. \tag{169}$$

We repeat the above experiment with various values of n, and plot the results in Figure 3. This figure also corresponds to band limit $c = 1000$. We plot the following three quantities as functions of the prolate index n that varies between $637 \approx 2c/\pi$ and 700. First, we plot $|\lambda_n|$. Second, we plot the maximal quadrature error $\Delta_1(n)$ defined via the formula

$$\Delta_1(n) = \max_{0 \leq a \leq 1} \delta_n(e^{icax}) = \max_{0 \leq a \leq 1} \left| \frac{2\sin(ac)}{ac} - \sum_{j=1}^{n} e^{icat_j^{(n)}} \cdot W_j^{(n)} \right|, \tag{170}$$

where $t_1^{(n)}, \ldots, t_n^{(n)}$ and $W_1^{(n)}, \ldots, W_n^{(n)}$ are, respectively, the notes and weights of the quadrature rule S_n (see (101) in Section 4). Finally, we plot the maximal quadrature error $\Delta_2(n)$ defined via the formula

$$\Delta_2(n) = \max_{1<a\leq 2} \delta_n(e^{icax}) = \max_{1<a\leq 2} \left| \frac{2\sin(ac)}{ac} - \sum_{j=1}^{n} e^{icat_j^{(n)}} \cdot W_j^{(n)} \right|. \tag{171}$$

We observe that in (170) the parameter a varies between 0 and 1, and in (171) the parameter a varies between 1 and 2. In other words, $\Delta_1(n)$ is the maximal quadrature errors of S_n for the exponential functions of band limits up to c, and $\Delta_2(n)$ is the maximal quadrature error of S_n for the exponential functions of band limit between c and $2c$.

We make the following observations from Figure 3. As long as $|\lambda_n|$ is less than roughly $10^{-7} \approx \sqrt{\varepsilon}$ (with ε the machine precision), $\Delta_1(n)$ is roughly equal to $|\lambda_n|^2$. On the other hand, $\Delta_1(n)$ is zero up to machine precision once $|\lambda_n| > 10^{-7}$. These observations are in agreement with Conjecture 1 above.

We also observe that $\Delta_2(n)$ is roughly of order $|\lambda_n|$, as long as $|\lambda_n| > \varepsilon$. On the other hand, when λ_n is zero to machine precision, so is $\Delta_2(n)$ (see Conjecture 1).

We repeat this experiment in extended precision, and plot the results in Figure 4. In other words, Figure 4 is a version of Figure 3 in extended precision. We observe the same phenomenon: $\Delta_1(n)$ is of order $|\lambda_n|^2$, and $\Delta_2(n)$ is of order $|\lambda_n|$ (as long as we do not run out of digits to see it; if, for example, $|\lambda_n|$ is below the machine zero so are both $\Delta_1(n)$ and $\Delta_2(n)$). In other words, the quadrature error of S_n for exponential functions with band limit up to c is of order $|\lambda_n|^2$, and the quadrature error of S_n for exponential functions with band limit between c and $2c$ is of order $|\lambda_n|$, which supports Conjecture 1.

7 Numerical Illustration of Analysis in Section 4

In this section, we illustrate the analytical results from Section 4 and the performance of the algorithms described in Section 5. All the calculations were implemented in FORTRAN (the Lahey 95 LINUX version), and carried out in double precision. Extended precision calculations were used for comparison and verification (in extended precision, the floating point numbers are 128 bits long, as opposed to 64 bits in double precision).

7.1 Quadrature Error and its Relation to $|\lambda_n|$

In this section, we describe several numerical experiments that illustrate the quadrature error (see (101), (103) in Section 4) and its relation to $|\lambda_n|$.

Experiment 2. Here we illustrate Theorem 14 in Section 4.1. We choose, more or less arbitrarily, band limit c and prolate index n. We evaluate χ_n, λ_n and the quadrature rule S_n defined via (101) in Section 4 via the algorithms of Sections 5.1, 5.2, 5.3, 5.4, respectively. Then, we choose an even integer $0 \leq m < n$, and evaluate λ_m, $\psi_m(0)$, and $\psi_m(t_j)$ for all $j = 1, \ldots, n$, via the algorithms of Sections 5.1, 5.2. All the calculations are carried out in double precision.

m	$\lambda_m \psi_m(0)$	$\delta_n(\psi_m)$, double precision	$\delta_n(\psi_m)$, extended precision
0	0.70669E+00	0.44409E-15	0.33258E-26
2	0.49581E+00	0.16653E-15	0.22426E-25
4	0.42581E+00	0.13323E-14	0.26756E-23
6	0.38527E+00	0.21649E-14	0.19692E-21
8	0.35695E+00	0.22760E-14	0.91546E-20
10	0.33516E+00	0.16653E-14	0.29148E-18
12	0.31730E+00	0.23870E-14	0.88165E-17
14	0.30201E+00	0.24980E-14	0.21007E-15
16	0.28844E+00	0.11102E-14	0.35574E-14
18	0.27604E+00	0.59230E-13	0.57028E-13
20	0.26435E+00	0.83716E-12	0.83954E-12
22	0.25299E+00	0.89038E-11	0.89011E-11
24	0.24150E+00	0.76862E-10	0.76864E-10
26	0.22919E+00	0.65870E-09	0.65870E-09
28	0.21377E+00	0.45239E-08	0.45239E-08
30	0.18075E+00	0.19826E-07	0.19826E-07
32	0.10038E+00	0.68548E-07	0.68548E-07
34	0.27988E-01	0.33810E-06	0.33810E-06
36	0.49822E-02	0.27232E-05	0.27232E-05
38	0.70503E-03	0.22754E-04	0.22754E-04

Table 1: *Illustration of Theorem 14 with* $c = 50$ *and* $n = 40$. *For these parameters,* $\lambda_n = 0.12915\text{E-}03$. *See Experiment 2.*

We display the results of this experiment in Table 1. The data in this table correspond to $c = 50$ and $n = 40$. Table 1 has the following structure. The first column contains the even integer m, that varies between 0 and $n-2$. The second column contains $\lambda_m \psi_m(0)$ (we observe that

$$\lambda_m \psi_m(0) = \int_{-1}^{1} \psi_m(t)\, dt, \tag{172}$$

due to (27) in Section 3.1). The third column contains the quadrature error

$$\delta_n(\psi_m) = \left| \lambda_m \psi_m(0) - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \tag{173}$$

(see (103) in Section 4), computed in double precision.

Then, we repeat all the calculations in extended precision; the last column of Table 1 contains $\delta_n(\psi_m)$ defined via (173) (the same quantity as in the third column evaluated in extended precision).

We make the following observations from Table 1. We note that $\lambda_m \psi_m(0)$ is always positive and monotonically decreases with m. We also note that $\delta_n(\psi_m)$ (evaluated in double precision) is close to the machine accuracy for small m, and grows up to $\approx 2 \cdot 10^{-5}$ for $m = 38$. Also, $\delta_n(\psi_m)$ is bounded by $|\lambda_n|$, for all values of m in Table 1 (in this case, $|\lambda_n| = 0.12915\text{E-}03$). Finally, $\delta_n(\psi_m)$ (evaluated in extended precision) is a monotonically increasing function of even $0 \le m < n$ (obviously, $\delta_n(\psi_m) = 0$ for odd m).

We summarize these observations in the following conjecture. We have not fully investigated the phenomenon described in this conjecture; see, however, Theorem 14 in Section 4.1, Conjecture 3 below, Figure 5 and Table 3 (see also [25], [26] for additional details and analysis).

Conjecture 2. *Suppose that $c > 1$ is a real number, that $n > 2c/\pi$ is an integer, and that the quadrature rule S_n is defined via* (101) *in Section 4. Then, the quadrature error $\delta_n(\psi_m)$ defined via* (173) *above is a monotonically increasing function of even $0 \le m < n$. Moreover, in double precision calculations $\delta_n(\psi_m)$ is zero up to machine precision for all $0 \le m < 2c/\pi$.*

In (106) in Theorem 14, we provide an upper bound on $\delta_n(\psi_m)$. This bound does not depend on m; more specifically, for every $m = 0, \ldots, n-1$,

$$\delta_n(\psi_m) = \left| \lambda_m \psi_m(0) - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \le |\lambda_n| \cdot \left(24 \cdot \log\left(\frac{1}{|\lambda_n|} \right) + 6 \cdot \chi_n \right). \tag{174}$$

On the other hand, according to Table 1 the quadrature error $\delta_n(\psi_m)$ is bounded by $|\lambda_n|$ alone, for all even $0 \le m < n$ (obviously, $\delta_n(\psi_m) = 0$ for all odd m).

In Figure 5, we display the results of the same experiment with different choice of parameters c and n. Namely, we choose $c = 10000$ and plot $\lambda_m \psi_m(0)$ as a function of even $0 \le m < 6425$, on the logarithmic scale (solid line). In addition, we plot the quadrature error $\delta_n(\psi_m)$ as a function of m, for four different values of n: $n = 6393$ (dashed line),

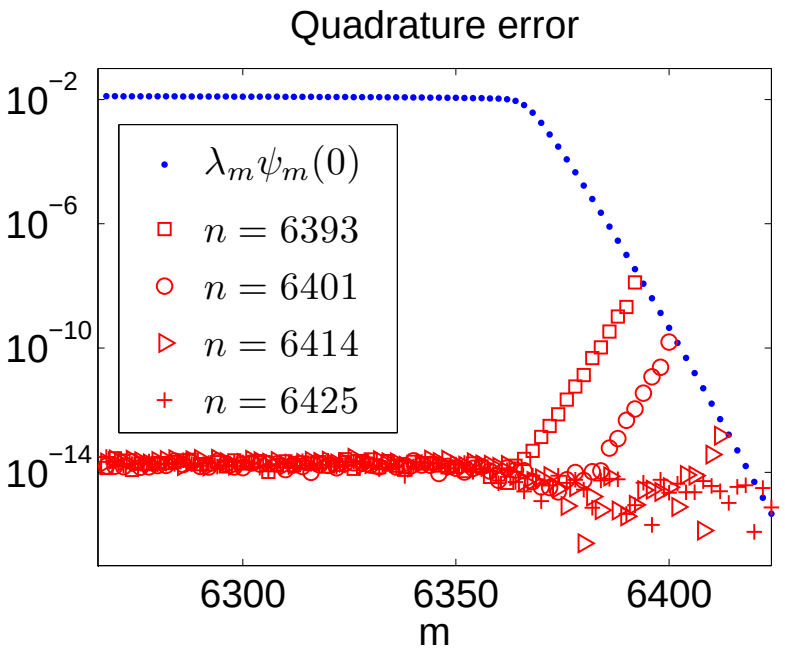

Figure 5: *The quadrature error* $\delta_n(\psi_m) = \left| \int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right|$ *as a function of even* $m < n$*, for four different values of* n *and* $c = 10000$*, vs.* $\lambda_m\psi_m(0)$*. See Experiment 2.*

n	6393	6401	6414	6425
$\lvert\lambda_n\rvert$	0.43299E-07	0.54119E-09	0.33602E-12	0.52616E-15

Table 2: *Values of* $|\lambda_n|$ *for* $c = 10000$ *and different choices of* n*.*

$n = 6401$ (circles), $n = 6414$ (triangles), and $n = 6425$ (pluses). The corresponding values of $|\lambda_n|$ are displayed in Table 2.

We make the following observations from Figure 5. First, the quantities $\lambda_m \psi_m(0)$ are of the same order of magnitude for all $m < 2c/\pi$, and decay rapidly with m for $m > 2c/\pi$. Also, for each value of n, the quadrature error $\delta_n(\psi_m)$ is essentially zero for all $m < 2c/\pi$ and increases rapidly with m for $m > 2c/\pi$. Nevertheless, $\delta_n(\psi_m)$ is always bounded from above by $|\lambda_n|$, for each n and each $m < n$. See also Tables 1, 3 and Conjecture 3 below.

c	n	m	$\lambda_m \psi_m(0)$	$\int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j$	$\lvert\lambda_n\rvert$
250	179	178	0.28699E-07	-.52496E-08	0.18854E-07
250	184	182	0.68573E-09	-.38341E-10	0.16130E-09
250	188	186	0.14108E-10	-.68758E-12	0.30500E-11
500	339	338	0.52368E-07	-.13473E-07	0.40938E-07
500	345	344	0.37412E-09	-.86136E-10	0.27418E-09
500	350	348	0.12148E-10	-.99816E-12	0.35537E-11
1000	659	658	0.42709E-07	-.14354E-07	0.38241E-07
1000	665	664	0.51665E-09	-.15924E-09	0.43991E-09
1000	671	670	0.52494E-11	-.15024E-11	0.42815E-11
2000	1297	1296	0.41418E-07	-.17547E-07	0.41740E-07
2000	1304	1302	0.77185E-09	-.15036E-09	0.37721E-09
2000	1311	1310	0.31078E-11	-.11386E-11	0.28754E-11
4000	2572	2570	0.54840E-07	-.15493E-07	0.33682E-07
4000	2579	2578	0.43032E-09	-.20771E-09	0.46141E-09
4000	2587	2586	0.28193E-11	-.12805E-11	0.29164E-11
8000	5119	5118	0.43268E-07	-.26751E-07	0.52899E-07
8000	5128	5126	0.50230E-09	-.16395E-09	0.33442E-09
8000	5136	5134	0.50508E-11	-.15448E-11	0.32132E-11
16000	10213	10212	0.42725E-07	-.30880E-07	0.56568E-07
16000	10222	10220	0.69663E-09	-.28201E-09	0.52821E-09
16000	10231	10230	0.34472E-11	-.22162E-11	0.42902E-11

Table 3: *Relation between the quadrature error and* $|\lambda_n|$. *See Experiment 2.*

We repeat the experiment above with several other values of band limit c and prolate index n. The results are displayed in Table 3. This table has the following structure. The first and second column contain, respectively, the band limit c and the prolate index n. The third column contains the even integer $0 \leq m < n$ (the values of m were chosen to be close to n). The fourth column contains $\lambda_m \psi_m(0)$. The fifth column contains (173). The last column contains $|\lambda_n|$.

We make the following observations from Table 3. First, for each of the seven values of c, the three indices n were chosen in such a way that $|\lambda_n|$ is between 10^{-12} and 10^{-7}. The values of the band limit c vary between 250 (the first three rows) and 16000 (the last three rows). For each n, the value of m is chosen to be the largest even integer below n. This

choice of m yields the smallest $\lambda_m \psi_m(0)$ and the largest quadrature error $\delta_n(\psi_m)$ among all $m < n$ (see also Table 1 and Figure 5). Obviously, for this choice of m the eigenvalues λ_m and λ_n are roughly of the same order of magnitude. We also observe that for all the values of c, n, m, the quadrature error $\delta_n(\psi_m)$ is bounded from above by $|\lambda_n|$ (and is roughly equal to $|\lambda_n|/2$). In other words, the upper bound on $\delta_n(\psi_m)$ provided by Theorem 14 (see (174)) is somewhat overcautious.

We summarize these observations in the following conjecture.

Conjecture 3. *Suppose that $c > 0$ is a positive real number, and that $n > 2c/\pi$ is an integer. Suppose also that $0 \leq m < n$ is an integer. Suppose furthermore that $\delta_n(\psi_m)$ is defined via* (103) *in Definition 2 in Section 4. Then,*

$$\delta_n(\psi_m) = \left| \int_{-1}^{1} \psi_m(s)\, ds - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \leq |\lambda_n|, \tag{175}$$

where λ_n is that of (27) *in Section 3.1.*

Remark 37. *The inequality* (175) *in Conjecture 3 is stronger than the inequality* (106) *in Theorem 14. On the other hand, as opposed to Theorem 14, Conjecture 3 has been only supported by numerical evidence.*

Experiment 3. Here we illustrate Theorems 15, 16 in Section 4.2. We proceed as follows. We choose, more or less arbitrarily, the band limit $c > 0$ and the accuracy parameter $\varepsilon > 0$. Then, we use the algorithm of Section 5.2 to find the minimal integer m such that $|\lambda_m| < \varepsilon$. In other words, we define the integer $n_1(\varepsilon)$ via the formula

$$n_1(\varepsilon) = \min \{ m \geq 0 \ : \ |\lambda_m| < \varepsilon \}. \tag{176}$$

Also, we find the minimal integer such that the right-hand side of (106) in Theorem 14 in Section 4.1 is less that ε. In other words, we define the integer $n_2(\varepsilon)$ via the formula

$$n_2(\varepsilon) = \min \left\{ m \geq 0 \ : \ |\lambda_m| \cdot \left(24 \cdot \log \left(\frac{1}{|\lambda_m|} \right) + 6 \cdot \chi_m \right) < \varepsilon \right\}. \tag{177}$$

Next, we evaluate the integer $n_3(\varepsilon)$ via the formula (110) in Theorem 15. In other words,

$$n_3(\varepsilon) = \text{floor} \left(\frac{2c}{\pi} + \frac{\alpha(\varepsilon)}{2\pi} \cdot \log \left(\frac{16ec}{\alpha(\varepsilon)}, \right) \right) \tag{178}$$

where $\alpha(\varepsilon)$ is defined via (109) in Theorem 15. Finally, we evaluate the integer $n_4(\varepsilon)$ via the right-hand side of (115) in Theorem 16. In other words,

$$n_4(\varepsilon) = \text{floor} \left(\frac{2c}{\pi} + \left(10 + \frac{3}{2} \cdot \log(c) + \frac{1}{2} \cdot \log \frac{1}{\varepsilon} \right) \cdot \log \left(\frac{c}{2} \right) \right). \tag{179}$$

In both (178) and (179), floor(a) denotes the integer part of a real number a.

We display the results of this experiment in Table 4. This table has the following structure. The first column contains the band limit c. The second column contains the

c	ε	$n_1(\varepsilon)$	$n_2(\varepsilon)$	$n_3(\varepsilon)$	$n_4(\varepsilon)$	$\lvert\lambda_{n_1(\varepsilon)}\rvert$	$\lvert\lambda_{n_2(\varepsilon)}\rvert$
250	10^{-10}	184	198	277	303	0.60576E-10	0.86791E-16
250	10^{-25}	216	227	326	386	0.31798E-25	0.14863E-30
250	10^{-50}	260	270	393	525	0.28910E-50	0.75155E-56
500	10^{-10}	346	362	460	488	0.49076E-10	0.60092E-16
500	10^{-25}	382	397	520	583	0.54529E-25	0.19622E-31
500	10^{-50}	433	446	607	742	0.82391E-50	0.38217E-56
1000	10^{-10}	666	687	803	834	0.95582E-10	0.92947E-17
1000	10^{-25}	707	725	875	942	0.97844E-25	0.14241E-31
1000	10^{-50}	767	783	981	1120	0.39772E-50	0.56698E-57
2000	10^{-10}	1305	1330	1467	1500	0.95177E-10	0.25349E-17
2000	10^{-25}	1351	1373	1550	1619	0.86694E-25	0.27321E-32
2000	10^{-50}	1418	1438	1675	1818	0.88841E-50	0.22795E-57
4000	10^{-10}	2581	2610	2768	2804	0.70386E-10	0.64396E-18
4000	10^{-25}	2632	2658	2862	2935	0.57213E-25	0.53827E-33
4000	10^{-50}	2707	2730	3007	3154	0.56712E-50	0.88819E-58
8000	10^{-10}	5130	5163	5344	5383	0.59447E-10	0.22821E-18
8000	10^{-25}	5185	5216	5450	5526	0.87242E-25	0.16237E-33
8000	10^{-50}	5268	5296	5614	5765	0.95784E-50	0.23927E-58
16000	10^{-10}	10225	10264	10468	10509	0.63183E-10	0.37516E-19
16000	10^{-25}	10285	10321	10585	10664	0.85910E-25	0.41416E-34
16000	10^{-50}	10377	10409	10769	10923	0.51912E-50	0.56250E-59
32000	10^{-10}	20413	20457	20686	20730	0.62113E-10	0.12818E-19
32000	10^{-25}	20478	20519	20815	20897	0.78699E-25	0.12197E-34
32000	10^{-50}	20577	20615	21018	21176	0.96802E-50	0.15816E-59
64000	10^{-10}	40786	40837	41092	41139	0.89344E-10	0.28169E-20
64000	10^{-25}	40857	40903	41232	41318	0.66605E-25	0.39212E-35
64000	10^{-50}	40964	41008	41454	41616	0.85451E-50	0.28036E-60
10^6	10^{-10}	636669	636747	637115	637174	0.79326E-10	0.13385E-22
10^6	10^{-25}	636759	636832	637301	637400	0.77413E-25	0.15758E-37
10^6	10^{-50}	636899	636968	637600	637778	0.69235E-50	0.15801E-62

Table 4: *Illustration of Theorems 15, 16. See Experiment 3.*

accuracy parameter ε. The third column contains $n_1(\varepsilon)$ defined via (176). The fourth column contains $n_2(\varepsilon)$ defined via (177). The fifth column contains $n_3(\varepsilon)$ defined via (178). The sixth column contains $n_4(\varepsilon)$ defined via (179). The seventh column contains $|\lambda_{n_1(\varepsilon)}|$. The last column contains $|\lambda_{n_2(\varepsilon)}|$.

Suppose that $c > 0$ is a band limit, and $n > 0$ is an integer. We define the real number $Q(c, n)$ via the formula

$$Q(c,n) = \max \left\{ \delta_n(\psi_m) = \left| \int_{-1}^{1} \psi_m(t)\, dt - \sum_{j=1}^{n} \psi_m(t_j) \cdot W_j \right| \ : \ 0 \leq m \leq n-1 \right\}, \quad (180)$$

where $t_1, \ldots, t_n$ and $W_1, \ldots, W_n$ are defined, respectively, via (98), (100) in Definition 2 in Section 4. In other words, $Q(c, n)$ is the maximal error to which the quadrature rule S_n defined via (101) integrates the first n PSWFs.

We make the following observations from Table 4. We observe that $Q(c, n_1(\varepsilon)) < \varepsilon$, due to the combination of Conjecture 3 in Section 7.1 and (176), (180). In other words, numerical evidence suggests that the quadrature rule $S_{n_1(\varepsilon)}$ integrates the first $n_1(\varepsilon)$ PSWFs up to an error less than ε (see Remark 37). On the other hand, we combine Theorem 14 in Section 4.1 with (177), (180), to conclude that the quadrature rule $S_{n_2(\varepsilon)}$ has been *rigorously proven* to integrate the first $n_2(\varepsilon)$ PSWFs up to an error less than ε. In both Theorem 14 and Conjecture 3, we establish upper bounds on $Q(c, n)$ in terms of $|\lambda_n|$. The ratio of $|\lambda_{n_1(\varepsilon)}|$ to $|\lambda_{n_2(\varepsilon)}|$ is quite large: from about 10^6 for $c = 250$ and $\varepsilon = 10^{-10}, 10^{-25}, 10^{-50}$ (see the first three rows in Table 4), to about 10^{10} for $c = 64000$ and $\varepsilon = 10^{-10}, 10^{-25}, 10^{-50}$, to about $5 \cdot 10^{12}$ for $c = 10^6$ and $\varepsilon = 10^{-10}, 10^{-25}, 10^{-50}$, (see the last six rows in Table 4). On the other hand, the difference between $n_2(\varepsilon)$ and $n_1(\varepsilon)$ is fairly small; for example, for $\varepsilon = 10^{-50}$, this difference varies from 10 for $c = 250$ to 23 for $c = 4000$, to merely 44 for $c = 64000$ and 69 for as large c as $c = 10^6$.

As opposed to $n_1(\varepsilon)$ and $n_2(\varepsilon)$, the integer $n_3(\varepsilon)$ is computed via the explicit formula (178) that depends only on c and ε (rather than on $|\lambda_n|$ and χ_n, that need to be evaluated numerically); this formula appears in Theorem 15. The convenience of (178) vs. (176), (177) comes at a price: for example, for $\varepsilon = 10^{-50}$, the difference between $n_3(\varepsilon)$ and $n_2(\varepsilon)$ is equal to 123 for $c = 250$, to 446 for $c = 64000$, and to 632 for $c = 10^6$. However, the difference $n_3(\varepsilon) - n_2(\varepsilon)$ is rather small compared to c: for example, for $\varepsilon = 10^{-50}$, this difference is roughly $4 \cdot (\log(c))^2$, for all values of c in Table 4.

Furthermore, we observe that $n_4(\varepsilon)$ is computed via the explicit formula (179) that depends only on c and ε. This formula can be viewed as a simplified version of (178) (see Theorems 15, 16); in particular, $n_4(\varepsilon)$ is greater than $n_3(\varepsilon)$, for all c and ε.

We summarize these observations as follows. Suppose that the band limit $c > 0$ and the accuracy parameter $\varepsilon > 0$ are given. According to Theorem 15, for any $n \geq n_3(\varepsilon)$ the quadrature rule S_n defined via (101) in Section 4 is *guaranteed* to integrate the first n PSWFs to precision ε (see Definition 1 in Section 2.1). On the other hand, numerical evidence (see Experiment 2) suggests that the choice $n \geq n_3(\varepsilon)$ is overly cautious for this purpose; more specifically, S_n integrates the first n PSWFs to precision ε for every n between $n_1(\varepsilon)$ and $n_3(\varepsilon)$ as well. In this experiment, we observed that the difference between the "theoretical" bound $n_3(\varepsilon)$ and "empirical" bound $n_1(\varepsilon)$ is of order $(\log(c))^2$,

and, in particular, is relatively small compared to both $n_1(\varepsilon)$ and $n_3(\varepsilon)$ (which are of order c).

Finally, we observe that

$$n_1(\varepsilon) < \frac{2c}{\pi} + \frac{2}{\pi^2} \cdot (\log c) \cdot \log \frac{1}{\varepsilon}, \tag{181}$$

for all the values of c and ε in Table 4. Combined with some additional numerical experiments by the authors, this observation leads to the following conjecture (see also Theorem 7 in Section 3.1 for a rigorously proven and more precise statement).

Conjecture 4. *Suppose that $c > 1$ and $0 < \varepsilon < 1$ are real numbers. Suppose also that $n > 0$ is an integer, and that*

$$n > \frac{2c}{\pi} + 10 + \frac{2}{\pi^2} \cdot (\log c) \cdot \log \frac{1}{\varepsilon}. \tag{182}$$

Then,

$$|\lambda_n| < \varepsilon, \tag{183}$$

where λ_n is that of (27) *in Section 3.1.*

7.2 Quadrature Weights

In this section, we illustrate the results of Section 4.3 and the algorithms of Section 5.4.

Experiment 4. In this experiment, we choose, more or less arbitrarily, band limit c and prolate index n. Then, we compute $t_1, \ldots, t_n$ (see (98)) and $\psi_n'(t_1), \ldots, \psi_n'(t_n)$ via the algorithm of Section 5.3. Also, we evaluate $\psi_n(0), \psi_n'(0)$ via the algorithm of Section 5.1. Next, compute approximations $\widetilde{W}_1, \ldots, \widetilde{W}_n$ to $W_1, \ldots, W_n$ via Algorithm 1 in Section 5.4 (in particular, $\widetilde{W}_j$ is evaluated via (150) for every $j = 1, \ldots, n$). Also, we compute approximations $\widehat{W}_1, \ldots, \widehat{W}_n$ to $W_1, \ldots, W_n$ via Algorithm 2 in Section 5.4. All the calculations are carried out in double precision.

We display the results of this experiment in Table 5. The data in this table correspond to $c = 40$ and $n = 41$. Table 5 has the following structure. The first column contains the weight index j, that varies between 1 and $(n+1)/2 = 21$. The second column contains $\widehat{W}_j$ (an approximation to W_j evaluated by Algorithm 2 in Section 5.4). The third column contains the difference between $\widehat{W}_j$ and $\widetilde{W}_j$ (evaluated via (150) by Algorithm 1). The last column contains the difference

$$\widehat{W}_j - \frac{\widehat{W}_{(n+1)/2} \cdot (\psi_n'(0))^2}{(\psi_n'(t_j))^2 \cdot \left(1 - t_j^2\right)} \tag{184}$$

(see (125) in Remark 14).

In Figure 6, we plot the weights W_j as a function of $j = 1, \ldots, n$. Each W_j is plotted as a circle above the corresponding node t_j.

j	$\widehat{W}_j$	$\widehat{W}_j - \widetilde{W}_j$	$\widehat{W}_j - \frac{\widehat{W}_{(n+1)/2}(\psi_n'(0))^2}{(\psi_n'(t_j))^2 \cdot (1-t_j^2)}$
1	0.7602931556894E-02	0.00000E+00	-.55796E-11
2	0.1716167229714E-01	0.00000E+00	-.55504E-10
3	0.2563684665002E-01	0.00000E+00	-.21825E-12
4	0.3278512460580E-01	0.00000E+00	-.11959E-09
5	0.3863462966166E-01	0.16653E-15	0.82238E-11
6	0.4334940472363E-01	0.22204E-15	-.16247E-09
7	0.4713107235981E-01	0.22204E-15	0.11270E-10
8	0.5016785516291E-01	0.19429E-15	-.18720E-09
9	0.5261660773966E-01	0.26368E-15	0.10495E-10
10	0.5460119701692E-01	0.29837E-15	-.20097E-09
11	0.5621699326080E-01	0.17347E-15	0.81464E-11
12	0.5753664411864E-01	0.12490E-15	-.20866E-09
13	0.5861531690539E-01	0.10408E-15	0.55098E-11
14	0.5949490764741E-01	0.23592E-15	-.21301E-09
15	0.6020725336886E-01	0.13184E-15	0.31869E-11
16	0.6077650804037E-01	0.18041E-15	-.21545E-09
17	0.6122088420703E-01	0.48572E-16	0.14361E-11
18	0.6155390478472E-01	0.83267E-16	-.21675E-09
19	0.6178529976346E-01	0.11102E-15	0.36146E-12
20	0.6192162112196E-01	0.48572E-16	-.21732E-09
21	0.6196665001384E-01	0.00000E+00	0.00000E+00

Table 5: *Quadrature weights* (100) *with* $c = 40$, $n = 41$. $\lambda_n =$ i0.69857E-08. *See Experiment 4.*

We make the following observations from Table 5. First, due to the combination of Theorems 17, 18 in Section 4.3, the value in the third column would be zero in exact arithmetic. We observe that, indeed, this value is zero up to the machine precision, which confirms Remarks 30, 35 in Section 5.4. (We note that, for $j = 1, 2, 3, 4$ and $j = 21$, both $\widehat{W}_j$ and $\widetilde{W}_j$ are evaluated via (150), and hence the value in the corresponding rows is exactly zero). In particular, either of the two approximations $\widetilde{W}_j, \widehat{W}_j$ can be used to evaluate W_j to essentially machine precision.

We also observe that all of the weights $W_1, \ldots, W_n$ are positive (see Theorem 19 and Remark 13). Moreover, W_j grow monotonically as j increases to $(n+1)/2$. Finally, we observe that, for all $j = 1, \ldots, n$, the value (184) in the last column is of the order $|\lambda_n|$ (see Remark 14).

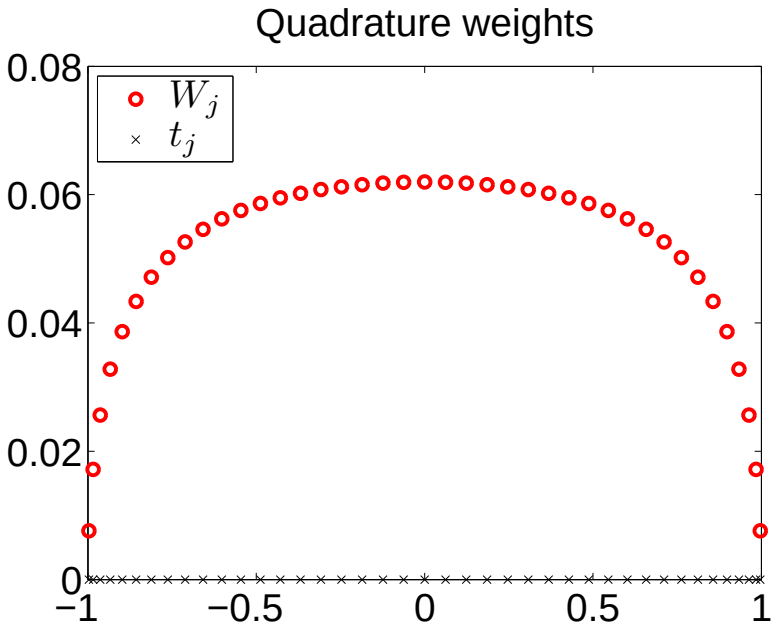

Figure 6: *The quadrature weights $W_1, \ldots, W_n$ with $c = 40$, $n = 41$. See Experiment 4.*

References

[1] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, Dover Publications, 1964.

[2] W. Barth, R. S. Martin, J. H. Wilkinson, *Calculation of the Eigenvalues of a Symmetric Tridiagonal Matrix by the Method of Bisection*, Numerische Mathematik 9, 386-393, 1967.

[3] C. J. Bouwkamp, *On spheroidal wave functions of order zero*, J. Math. Phys. 26, 79-92, 1947.

[4] H. Cheng, N. Yarvin, V. Rokhlin, *Non-linear Optimiuzation, Quadrature and Interpolation*, SIAM J. Optim. 9, 901-23, 1999.

[5] G. Dahlquist, A. Björk, *Numerical Methods*, Prentice-Hall Inc., 1974.

[6] M.V. Fedoryuk, *Asymptotic Analysis of Linear Ordinary Differential Equations*, Springer-Verlag, Berlin (1993).

[7] C. Flammer, *Spheroidal Wave Functions*, Stanford, CA: Stanford University Press, 1956.

[8] Andreas Glaser, Xiangtao Liu, Vladimir Rokhlin, *A fast algorithm for the calculation of the roots of special functions*, SIAM J. Sci. Comput., 29(4):1420-1438 (electronic), 2007.

[9] I.S. Gradshteyn, I.M. Ryzhik, *Table of Integrals, Series, and Products*, Seventh Edition, Elsevier Inc., 2007.

[10] D. B. Hodge, *Eigenvalues and Eigenfunctions of the Spheroidal Wave Equation*, J. Math. Phys. 11, 2308-2312, 1970.

[11] E. Isaacson, H. B. Keller, *Analysis of Numerical Methods*, New York: Wiley, 1966.

[12] S. Karlin, W. J. Studden, *Tchebycheff Systems with Applications in Analysis and Statistics*, Wiley-Interscience, New York, 1966.

[13] M. G. Krein, *The Ideas of P. L. Chevyshev and A. A. Markov in the THeory of Limiting Values of Integrals*, AM. Math. Soc. Trans., 12, 1-122, 1959.

[14] H. J. Landau, H. O. Pollak, *Prolate spheroidal wave functions, Fourier analysis, and uncertainty - II*, Bell Syst. Tech. J. January 65-94, 1961.

[15] H. J. Landau, H. Widom, *Eigenvalue distribution of time and frequency limiting*, J. Math. Anal. Appl. 77, 469-81, 1980.

[16] J. Ma, V. Rokhlin, S. Wandzura, *Generalized Gaussian Quadratures for Systems of Arbitrary Functions*, SIAM J. Numer. Anal. 33, 971-96, 1996.

[17] A. A. Markov, *On the Limiting Values of Integrals in Connection with Interpolation*, Zap. Imp. Akad. Nauk. Fiz.-Mat. Otd. (8) 6, no 5 (in Russian), 1898.

[18] A. A. Markov, *Selected Papers on Continued Fractions and the Theory of Functions Deviating Least From Zero*, OGIZ: Moscow (in Russian), 1948.

[19] Richard K. Miller, Anthony N. Michel, *Ordinary Differential Equations*, Dover Publications, Inc., 1982.

[20] P. M. Morse, H. Feshbach, *Methods of Theoretical Physics*, New York McGraw-Hill, 1953.

[21] A. Osipov, *Certain inequalities involving prolate spheroidal wave functions and associated quantities*, Appl. Comput. Harmon. Anal. (2012), http://dx.doi.org/10.1016/j.acha.2012.10.002 (in press).

[22] A. Osipov, *Certain inequalities involving prolate spheroidal wave functions and associated quantities*, arXiv:1206.4056v1, 2012.

[23] A. Osipov, *Explicit upper bounds on the eigenvalues associated with prolate spheroidal wave functions*, Yale CS Technical Report #1450, 2012.

[24] A. Osipov, *Certain upper bounds on the eigenvalues associated with prolate spheroidal wave functions*, arXiv:1206.4541v1, 2012.

[25] A. Osipov, V. Rokhlin, *Detailed analysis of prolate quadratures and interpolation formulas*, Yale CS Technical Report #1458, 2012.

[26] A. Osipov, V. Rokhlin, *Detailed analysis of prolate quadratures and interpolation formulas*, arXiv:1208.4816v1, 2012.

[27] A. Osipov, *Evaluation of small elements of the eigenvectors of certain symmetric tridiagonal matrices with high relative accuracy*, Yale CS Technical Report #1460, 2012.

[28] A. Osipov, *Evaluation of small elements of the eigenvectors of certain symmetric tridiagonal matrices with high relative accuracy*, arXiv:1208.4906v1, 2012.

[29] A. Papoulis, *Signal Analysis*, Mc-Graw Hill, Inc., 1977.

[30] Vladimir Rokhlin, Hong Xiao, *Approximate Formulae for Certain Prolate Spheroidal Wave Functions Valid for Large Value of Both Order and Band Limit.*

[31] W. Rudin, *Real and Complex Analysis*, Mc-Graw Hill Inc., 1970.

[32] D. Slepian, *Some comments on Fourier analysis, uncertainty, and modeling*, SIAM Rev.(3) 379-93, 1983.

[33] D. Slepian, H. O. Pollak, *Prolate spheroidal wave functions, Fourier analysis, and uncertainty - I*, Bell Syst. Tech. J. January 43-63, 1961.

[34] D. Slepian, H. O. Pollak, *Prolate spheroidal wave functions, Fourier analysis, and uncertainty - IV: extensions to many dimensions, generalized prolate spheroidal wave functions*, Bell Syst. Tech. J. November 3009-57, 1964.

[35] D. Slepian, *Some asymptotic expansions for prolate spheroidal wave functions*, J. Math. Phys. 44 99-140, 1965.

[36] J. H. Wilkinson, *Algebraic Eigenvalue Problem*, Oxford University Press, New York, 1965.

[37] H. Xiao, V. Rokhlin, N. Yarvin, *Prolate spheroidal wavefunctions, quadrature and interpolation*, Inverse Problems, 17(4):805-828, 2001.

[38] N. Yarvin, V. Rokhlin, *Generalized Gaussian Quadratures and Singular Value Decompositions of Integral Operators*, SIAM J. Sci. Comput. 20, 699-718, 1998.