Some New Results on and Applications of Interpolation in Numerical Computation

Anthony P. Austin

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term, 2016

To my family.

Abstract

This thesis discusses several topics related to interpolation and how it is used in numerical analysis. It begins with an overview of the aspects of interpolation theory that are relevant to the discussion at hand before presenting three new contributions to the field.

The first new result is a detailed error analysis of the barycentric formula for trigonometric interpolation in equally-spaced points. We show that, unlike the barycentric formula for polynomial interpolation in Chebyshev points (and contrary to the main view in the literature), this formula is not always stable. We demonstrate how to correct this instability via a rewriting of the formula and establish the forward stability of the resulting algorithm.

Second, we consider the problem of trigonometric interpolation in grids that are perturbations of equally-spaced grids in which each point is allowed to move by at most a fixed fraction of the grid spacing. We prove that the Lebesgue constant for these grids grows at a rate that is at most algebraic in the number of points, thus answering questions put forth by Trefethen and Weideman [135] about the robustness of numerical methods based on trigonometric interpolation in points that are uniformly distributed but not equally-spaced. We use this bound to derive theorems about the convergence rate of trigonometric interpolation in these grids and also discuss the related question of quadrature. Specifically, we prove that if a function has $\nu \geq 1$ derivatives, the νth of which is Hölder continuous (with a Hölder exponent that depends on the size of the maximum allowable perturbation), then the interpolants converge uniformly to the function at an algebraic rate; larger values of ν lead to more rapid convergence. A similar statement holds for the corresponding quadrature rule. We also consider what analogue, if any, there is for trigonometric interpolation of the famous $1/4$ theorem of Kadec from sampling theory that restricts the size of the perturbations one can make to the integers and still be guaranteed to have a set of stable sampling for the Paley–Wiener space. We present numerical evidence suggesting that in the discrete case, the $1/4$ threshold takes the form of a threshold for the boundedness of a "2-norm Lebesgue constant" and does not appear to have much significance in practice.

We believe that these are the first results regarding this problem to appear in the literature. While we do not believe the results we establish are the best possible quantitatively, they do (rigorously) capture the main features of trigonometric interpolation in perturbations of equally-spaced grids. We make several conjectures as to what the optimal results may be, backed by extensive numerical results.

Finally, we consider a new application of interpolation to numerical linear algebra. We show that recently developed methods for computing the eigenvalues of a matrix by discretizing contour integrals of its resolvent are equivalent to computing a rational interpolant to the resolvent and finding its poles. Using this observation as the foundation, we develop a method for computing the eigenvalues of real symmetric matrices that enjoys the same advantages as contour integral methods with respect to parallelism but employs only real arithmetic, thereby cutting the computational cost and storage requirements in half.

Acknowledgements

First, I would like to thank my doctoral supervisor, Nick Trefethen, for his guidance over the last four years. His abilities to see both the big and small pictures simultaneously and to lucidly explain sophisticated concepts are traits to which I can only aspire. It has been an honor to have been allowed to carry out this work under his tutelage. Obviously, his view of numerical analysis has greatly influenced my own, and I am sure the reader will see that influence on every page.

Second, I thank my examiners, Stefan Güttel and Andrew Thompson, for their most careful reading of this thesis. Their comments resulted in many improvements to the text, and the presentation has benefited at all levels from their insight.

While in Oxford, I have had the privilege and pleasure of working with many truly excellent colleagues. Among these, I acknowledge in particular Kuan Xu, together in collaboration with whom part of this work was completed. I thank my officemates Hrothgar, Mohsin Javed, Hadrien Montanelli, and Alex Townsend for countless hours of interesting discussions on topics both mathematical and otherwise. I also thank the rest of my fellow Chebfun team members from over the years—Jared Aurentz, Ásgeir Birkisson, Toby Driscoll, Nick Hale, Behnam Hashemi, Georges Klein, Olivier Séte, and Richard (Mikael) Slevinsky—for providing a highly stimulating intellectual environment. It has been truly wonderful to work with such great people. I can only hope that I have been able to be as good a colleague to them as they have been to me.

Other individuals from Oxford I would like to thank include my checkpoint examiners Coralia Cartis, Patrick Farrell, Ian Sobey, and Andy Wathen, who provided feedback on portions of this work in its early stages, and my college advisor, Frances Kirwan. I also thank the administrator for the numerical analysis group, Helen Taylor, and her predecessor, Lotti Ekert, for the many times they have assisted me over the years.

There are many individuals from outside Oxford with whom I have had the pleasure of interacting during my time here and who have helped shape my development as a researcher. Among these, I thank in particular Peter Kravanja, Karl Meerbergen, Yuji Nakatsukasa, Françoise Tisseur, Tetsuya Sakurai, André Weideman, and Grady Wright. I especially thank Mark Embree, who has been an important mentor to me since my days as an undergraduate. It was on his advice that I applied to study at Oxford and at Cambridge prior to that. Were it not for his influence, my path through graduate school would have been a very different one indeed.

The bulk of this work was funded by the European Research Council, and I thank them for making it possible. I also owe an enormous debt of gratitude to the Marshall Aid Commemoration Commission—and, by extension, the people of the United Kingdom—for giving me the chance to study on this side of the Atlantic in the first place. It is a debt I will never be able to truly repay.

I thank my good friend Patrick Wedgeworth and my friend and long-time mentor Peter Billingham from Jesuit Dallas for supporting me from home. Many times, they have made the roughly 4,750 mile distance that has separated us seem not quite so far.

Finally, I thank my family—my mother, Suzanne, my father, Anthony R., and my brother, James—for being an unconditional source of strength and encouragement for me not only for the five years that I have been overseas but throughout my life to date. I never would have made it this far without their love and support.

Anthony P. Austin
August 15, 2016

Contents

Chapter 1

Basic Interpolation Theory and Practice

This thesis is about interpolation and its uses in numerical computation. Before presenting our new contributions to the field in Chapters 2–5, here we set the stage for our discussion by reviewing the basic notions from interpolation theory and practice that we will need throughout. None of the material in this chapter is new; much of it can be found in standard textbooks. Our views and presentation have been particularly strongly influenced by [133] and the article [6].

1.1 Introduction

Interpolation is the name given to the general task of finding a function of a given form such that it and/or one or more of its derivatives assumes given values at a prescribed set of points. Classically developed as a technique for "filling in the blanks" in tables of numerical data, it is perhaps the most basic tool for approximating functions in all of mathematics. Yet it is so subtle and intricate—and its applications so wide-ranging—that it has spawned many thousands of pages of mathematical research, pure and applied, in the roughly four centuries that it has been a subject of study.[1]

As the imprecision of the definition just given might suggest, there are many problems that one can consider that fall under the heading of interpolation. Problems are typically classified first according to the functional form that the interpolant is required to take. Thus, one speaks of polynomial interpolation when the function to be found is a polynomial, piecewise linear interpolation when it is piecewise linear, and so on. We will not make any attempt to provide a unified treatment that handles all of the possibilities. Instead, we will immediately narrow our focus to the cases with which we will be most concerned in the later chapters, namely, polynomial, trigonometric, and rational interpolation in a single (complex) variable. As these themselves are each deep subjects

[1] Of course, we do not claim that interpolation was *invented* 400 years ago. To the contrary, basic interpolation schemes were known even to the ancients. For a brief account of the early history of this subject, we refer the reader to [82] and to the papers cited therein.

that possess book-length treatments of their own, we will only hit the highlights, leaving the reader to look up the details elsewhere as he or she desires.

1.2 Polynomial Interpolation

Polynomial interpolation is the most basic type of interpolation that one can consider. We denote the space of all polynomials of degree at most N by $\mathcal{P}_N$. The usual form of the polynomial interpolation problem reads:[2]

> Given K distinct interpolation points $z_0, \ldots, z_{K-1} \in \mathbb{C}$ and K corresponding values $f_0, \ldots, f_{K-1} \in \mathbb{C}$, find a polynomial $p \in \mathcal{P}_{K-1}$ such that $p(z_k) = f_k$ for each k.

A word about notation: throughout this thesis, the variable K will almost exclusively be reserved for the number of points involved in the interpolation problem under consideration, while N will similarly almost always represent the degree of the interpolant.[3]

1.2.1 Existence and Uniqueness

The polynomial p always exists and is unique. The classic way to prove this is to represent p in the monomial basis, writing $p(z) = \sum_{k=0}^{K-1} c_k z^k$, and then observe that the interpolation conditions yield the following linear system for the coefficients $c_0, \ldots, c_{K-1}$:

$$\begin{bmatrix} 1 & z_0 & \cdots & z_0^{K-1} \\ 1 & z_1 & \cdots & z_1^{K-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_{K-1} & \cdots & z_{K-1}^{K-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{K-1} \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{K-1} \end{bmatrix}. \tag{1.1}$$

The matrix on the left-hand side of this equation is called the *Vandermonde matrix* for the points $z_0, \ldots, z_{K-1}$. Expanding by minors down the last column and using an induction argument, one can show that the determinant of this matrix is $\prod_{i<j}(z_j - z_i)$. Since the z_k are distinct, the determinant is nonzero, and the result follows.

One can use (1.1) as the basis for a numerical algorithm for finding p: simply solve the Vandermonde system for the coefficients c_k. This turns out to be a bad idea, as Vandermonde matrices are notoriously poorly conditioned for general sets of interpolation points.[4] Thus, interpolation algorithms that rely on them will typically be numerically unstable.

[2] One can also consider versions of the problem in which one or more of the derivatives of the polynomial to be found are specified at the interpolation points. If the values of both the polynomial and its derivative are given at all the points, one obtains a *Hermite interpolation* problem. If the values of higher-order derivatives are given or if the number of derivatives specified varies from point to point (possibly with some intermediate derivatives omitted, e.g., the values of the first and third derivatives are specified at a point but not that of the second), the problem is called a *Hermite–Birkhoff interpolation* problem. As these variants will not figure into our later discussions, we will not consider them here.

[3] Thus, for the polynomial interpolation problem, N and K are related by $K = N + 1$, while for the trigonometric interpolation problem (see Section 1.3), we have $K = 2N + 1$. The major exception to this convention occurs in Chapter 5, where N is used for the dimension of a matrix.

[4] An important exception to this is the case of Vandermonde matrices for points that are equally spaced on the unit circle; see Section 1.2.3.

1.2.2 Lagrange Form, Barycentric Representation

A different way to represent p that is much better numerically and often convenient theoretically is as follows. For $0 \leq k \leq K-1$, define the polynomial $\ell_k(z)$ by

$$\ell_k(z) = \prod_{\substack{j=0 \\ j \neq k}}^{K-1} \frac{z - z_j}{z_k - z_j}. \tag{1.2}$$

It is easily checked that $\ell_k \in \mathcal{P}_{K-1}$, that $\ell_k(z_j) = 0$ if $j \neq k$, and that $\ell_k(z_k) = 1$. Thus, we have

$$p(z) = \sum_{k=0}^{K-1} f_k \ell_k(z). \tag{1.3}$$

This representation of p is known as the *Lagrange form*[5] of p, and the polynomials ℓ_k are termed the *Lagrange basis polynomials* (or, sometimes, the *cardinal polynomials*) for the points z_k.

By representing p in the Lagrange basis instead of the monomial basis, we avoid the pitfalls of Vandermonde matrices mentioned above; however, this new representation comes with a cost. Evaluating the monomial representation of p requires $O(K)$ operations. Evaluating (1.3), on the other hand, takes $O(K^2)$ operations, since $O(K)$ operations are needed to evaluate each of the K basis functions ℓ_k.

We can get around this problem by rewriting (1.3) in a clever way. Define $\ell(z) = (z - z_0) \cdots (z - z_{K-1})$. This polynomial is called the *node polynomial* for the points z_k. Straightforward calculations show that

$$\ell_k(z) = \frac{\ell(z)}{\ell'(z_k)(z - z_k)}.$$

Letting $\nu_k = 1/\ell'(z_k)$ for each k, it follows that (1.3) is equivalent to

$$p(z) = \ell(z) \sum_{k=0}^{K-1} \frac{\nu_k}{z - z_k} f_k, \tag{1.4}$$

and this requires only $O(K)$ operations to evaluate for each z once the values ν_k have been computed. The formula (1.4) is known as the *first barycentric formula* for p. The values ν_k are called the *barycentric weights* for the points z_k and are given by the formula

$$\nu_k^{-1} = \prod_{\substack{j=0 \\ j \neq k}}^{K-1} (z_k - z_j). \tag{1.5}$$

For an arbitrary grid, the computation of the ν_k requires $O(K^2)$ operations; however, for certain special and commonly used grids, there are closed-form expressions for the ν_k that require little or no work to evaluate.

One disadvantage of (1.4) is that it is prone to overflow and underflow when K is large. The culprits are the factor $\ell(z)$ that appears in front and the weights ν_k, both of which can grow or decay

[5] In spite of this nomenclature, Lagrange was not the first to discover this representation of the interpolating polynomial; it appears earlier in a paper by Waring [145].

exponentially as $K \to \infty$. Implementations of (1.4) must be carefully written to guard against this issue.

Alternatively, we can rewrite p in yet another form that circumvents this problem. Taking $f_0 = \cdots = f_{K-1} = 1$ in (1.4) yields the identity

$$1 = \ell(z) \sum_{k=0}^{K-1} \frac{\nu_k}{z - z_k}.$$

Dividing (1.4) by this identity on both sides and cancelling the common $\ell(z)$ factors then gives

$$p(z) = \frac{\displaystyle\sum_{k=0}^{K-1} \frac{\nu_k}{z - z_k} f_k}{\displaystyle\sum_{k=0}^{K-1} \frac{\nu_k}{z - z_k}}. \tag{1.6}$$

This formula is called the *second barycentric formula* for p, and it lacks the troublesome $\ell(z)$ factor. Moreover, one can pull out a constant factor common to all the ν_k and cancel it out of the quotient as well, counteracting any common exponential dependence of the ν_k on K. Thus, (1.6) is much less susceptible to overflow and underflow than (1.4).

The last remaining element needed to ensure the suitability of (1.4) and (1.6) for computation is an assessment of their numerical stability. At first glance, the situation appears dangerous due to the poles in the summands: when z is close to z_k, $1/(z - z_k)$ will be large, and any rounding error incurred in the subtraction $z - z_k$ will be amplified. It is reasonable to expect that this will result in a loss of accuracy when evaluating p near the interpolation points.

In fact, this does not happen. Intuitively, for (1.6), the reason is that while the errors in computing $1/(z - z_k)$ can indeed be large, the error made is the *same* in the numerator and the denominator and therefore cancels out when the quotient is taken. Something similar happens in (1.4) when one multiplies by $\ell(z)$. Higham provided rigorous justification for these observations in [58],[6] where he proved that, ignoring overflow and underflow, (1.4) is unconditionally backward stable and (1.6) is forward stable provided that the Lebesgue constant (see Section 1.2.8) for the interpolation problem is not too large. More recently, Mascarenhas has shown that when the interpolation points are second-kind Chebyshev points (see Section 1.2.4) and the evaluation point lies in $[-1, 1]$, (1.6) is actually also backward stable, provided that additional care is taken to evaluate the sums in a way that avoids cancellation [79].[7]

The name "barycentric formula" seemingly originates with [33] and comes from the fact that (1.6) resembles the formula from physics for the center of mass of a system of particles. The "first"

[6] There is also an earlier analysis of (1.4) due to Rack and Reimer [106] that reaches weaker conclusions than Higham's.

[7] If the evaluation point lies elsewhere, then (1.6) is not backward stable, though the forward error analysis given by Higham still applies. If the evaluation point is far from $[-1, 1]$, the forward error bound can be large. For further discussions, see [146].

and "second" terminology used to distinguish (1.4) from (1.6) is due to Rutishauser [116]. In the theoretical literature, the first formula (1.4) has been known for a long time; a version of it can be found in the early works of Jacobi [65]. The application of these formulas to numerical work is somewhat more recent; the earliest reference appears to be [131]. A particularly important figure in the development of these formulas is Salzer [124], who studied both them and their counterparts for trigonometric interpolation (see Section 1.3.2) with computation in mind. These formulas have enjoyed a boost in popularity following the publication of the article [14] by Berrut and Trefethen, to which we refer the reader for further discussions and historical notes.

1.2.3 Interpolation on the Unit Circle, Connection to Cauchy Integrals

Traditionally, polynomial interpolation is studied in the context of the unit interval $[-1, 1]$. While we will consider $[-1, 1]$ further below, it is our opinion that the unit circle $\mathbb{T}$ is in some sense even more fundamental as an interpolation domain, so we will discuss it first.

When interpolating on the circle, it is natural to take the points z_k to be equally spaced, and the most basic choice for such points is $z_k = \omega_k$, where $\omega_0, \ldots, \omega_{K-1}$ are the Kth roots of unity:

$$\omega_k = e^{2\pi i \frac{k}{K}}, \qquad 0 \le k \le K-1.$$

The node polynomial for these points is $\ell(z) = z^K - 1$, and it is easy to show that the barycentric weight for ω_k is $\nu_k = \omega_k / K$. Thus, (1.4) becomes

$$p(z) = \frac{z^K - 1}{K} \sum_{k=0}^{K-1} \frac{\omega_k}{z - \omega_k} f_k, \tag{1.7}$$

while (1.6) simplifies to

$$p(z) = \frac{\displaystyle\sum_{k=0}^{K-1} \frac{\omega_k}{z - \omega_k} f_k}{\displaystyle\sum_{k=0}^{K-1} \frac{\omega_k}{z - \omega_k}}.$$

Alternatively, interpolation in the roots of unity is one of the rare cases in which representing p in the monomial basis as suggested in Section 1.2.1 is a good idea. With $z_k = \omega_k$, the Vandermonde matrix in (1.1) has orthogonal columns, all of which have the same norm of $\sqrt{K}$. It is therefore a scaled unitary matrix and hence perfectly well-conditioned. Moreover, it can be efficiently inverted in $O(K \log K)$ operations using the fast Fourier transform.

There is a remarkable connection between polynomial interpolation in the roots of unity and discretized Cauchy integrals, which we now outline. Denote the open unit disc in $\mathbb{C}$ by $\mathbb{D}$ and its closure by $\overline{\mathbb{D}}$. If f is holomorphic in $\overline{\mathbb{D}}$, then Cauchy's integral formula [1] tells us that

$$f(z) = \frac{1}{2\pi i} \int_{\mathbb{T}} \frac{f(\zeta)}{\zeta - z} \, d\zeta \tag{1.8}$$

whenever $z \in \mathbb{D}$. One can therefore approximate f on $\mathbb{D}$ by discretizing the integral on the right-hand side of (1.8) using a quadrature rule. The natural quadrature rule to use in this case is the trapezoid rule, which has as its nodes the Kth roots of unity ω_k with corresponding quadrature weights[8] ω_k/K. Since f is holomorphic, this rule converges geometrically as $K \to \infty$ [135], which makes it a good choice for approximating (1.8) numerically. Using this rule, we obtain

$$f(z) \approx \frac{1}{K} \sum_{k=0}^{K-1} \frac{\omega_k}{\omega_k - z} f(\omega_k), \tag{1.9}$$

which is the same as (1.7) with $f_k = f(\omega_k)$, divided by $1 - z^K$. For $z = 0$, (1.7) and (1.9) are *exactly* the same.

Thus, approximating f on the unit disc by its polynomial interpolant in the roots of unity and by the trapezoid rule discretization of (1.8) are two very closely related operations. The difference between them arises because the integral in (1.8) vanishes for z outside of $\mathbb{D}$, and the discretization (1.9) has to reflect this. It accomplishes this by multiplying the polynomial interpolant by a "cage" of poles at the roots of unity—the factor $1/(1 - z^K)$—that acts to isolate the unit disc from the rest of the complex plane, a behavior that may or may not be desirable depending on one's application.

These concepts will figure prominently in Chapter 5 when we discuss an application of rational interpolation to finding the eigenvalues of large matrices. Further discussion of the relationship between polynomial interpolation and discretized Cauchy integrals with a particular emphasis on algorithms founded on each can be found in [6].

1.2.4 Interpolation on the Unit Interval, Chebyshev Points

For the remainder of our discussion of polynomial interpolation, we will mostly focus on the important case of polynomial interpolation in points in the unit interval $[-1, 1]$. To emphasize that we are working on a real interval, we will use the variables x and x_k in place of z and z_k for the evaluation point and interpolation points, respectively.

First, we observe that we can associate with every polynomial interpolation problem on $[-1, 1]$ a certain polynomial interpolation problem on the unit circle. This is easiest to see by representing the interpolant on the interval in terms of the *Chebyshev polynomials of the first kind*, denoted T_k, which are defined by $T_0(x) = 1$, $T_1(x) = x$, and

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \qquad k \geq 1. \tag{1.10}$$

Alternatively, one has the following explicit representation, valid for $x \in [-1, 1]$:

$$T_k(x) = \cos\bigl(k \arccos(x)\bigr). \tag{1.11}$$

[8] These quadrature weights contain the $1/(2\pi i)$ constant factor that appears in front of the integral, i.e., they give a method for approximating $\frac{1}{2\pi i} \int_{\mathbb{T}} g(z)\, dz$, not $\int_{\mathbb{T}} g(z)\, dz$.

For more about these polynomials and their many applications, see [81] and [112].

Since T_k is a polynomial of degree exactly k for each k, the set $\{T_0, \ldots, T_{K-1}\}$ is a basis for $\mathcal{P}_{K-1}$, and we can write our interpolant in the form

$$p(x) = \sum_{k=0}^{K-1} c_k T_k(x) \tag{1.12}$$

for some coefficients $c_0, \ldots, c_{K-1} \in \mathbb{C}$. The crucial fact that we need to connect $[-1, 1]$ to the unit circle is that for $x \in [-1, 1]$, $T_k(x) = \operatorname{Re} z^k$, where $z = x \pm i\sqrt{1 - x^2} \in \mathbb{T}$. This is easy to see for $k = 0$ and $k = 1$. For larger k, it follows from the fact that $\overline{z} = 1/z$ for $z \in \mathbb{T}$ together with the easily verified identity

$$\frac{z^{k+1} + z^{-(k+1)}}{2} = 2\frac{z + z^{-1}}{2}\frac{z^k + z^{-k}}{2} - \frac{z^{k-1} + z^{-(k-1)}}{2},$$

which shows that, as a function of x, $\operatorname{Re}(z^k)$ satisfies the same recurrence (1.10) as $T_k(x)$. Hence, in terms of the variable z, we can write

$$p(x) = \sum_{k=0}^{K-1} c_k \frac{z^k + z^{-k}}{2}.$$

Rewriting this to better emphasize its dependence on z, let

$$P(z) = \sum_{k=-(K-1)}^{K-1} a_k z^k,$$

where $a_0 = c_0$ and $a_k = c_{|k|}/2$ for $k \neq 0$. The function P is a *Laurent polynomial* in z of degree $K-1$ (i.e., a linear combination of $z^{-(K-1)}, \ldots, z^{K-1}$), and $P(z) = p(x)$, where $z \in \mathbb{T}$ and $x = \operatorname{Re} z$. Thus, if p interpolates the data $f_0, \ldots, f_{K-1}$ at the points $x_0, \ldots, x_{K-1} \in [-1, 1]$, we have that P interpolates f_k at the points z_k and $\overline{z}_k$, where $z_k = x_k + i\sqrt{1 - x_k^2}$ for each k. Hence, the polynomial $q(z) = z^{K-1}P(z) \in \mathcal{P}_{2K-2}$ interpolates the data $z_k^{K-1} f_k$ and $\overline{z}_k^{K-1} f_k$ at the points z_k and $\overline{z}_k$, respectively.

Thus, we have shown that polynomial interpolation in the K points x_k in $[-1, 1]$ is equivalent to finding a polynomial $q \in \mathcal{P}_{2K-2}$ such that $q(z_k) = z_k^{K-1} f_k$ and $q(\overline{z}_k) = \overline{z}_k^{K-1} f_k$ for each k. The interpolation points for this problem are conjugate-symmetric, and their real parts are the interpolation points x_k. Note that this is not quite a polynomial interpolation problem of the general form laid out at the beginning of this chapter. The reason is that the degree of q is $2K - 2$, while the number of interpolation conditions varies depending on whether neither, one, or both of 1 and -1 occur among the x_k, since if $x_k = \pm 1$, then $z_k = x_k = \overline{z}_k$. We can handle this issue as follows:

- If $x_k \neq \pm 1$ for each k, there are $2K$ interpolation conditions on q. The standard interpolation problem would choose q from $\mathcal{P}_{2K-1}$ to satisfy them; the given interpolation problem takes q from $\mathcal{P}_{2K-2}$. The preceding developments show that the solution when q is drawn from the latter exists. The uniqueness of polynomial interpolants (see Section 1.2.1) then shows that it must also be the solution when q is drawn from the former, so in fact, the interpolation problem can be interpreted as the standard one after all.

- If exactly one of the x_k is equal to 1 or -1, then there are $2K - 1$ conditions, and the interpolation problem for q is the standard one.
- If both 1 and -1 occur among the x_k, there are $2K - 2$ conditions, so q has one more degree of freedom than is necessary to satisfy them. A unique solution can be specified by additionally requiring that the leading-order and zeroth-order (constant) coefficients of q be the same. This will lead to a Laurent polynomial P that has symmetric coefficients (i.e., the z^k and z^{-k} terms have the same coefficient), as we had above.

In the previous section, we observed that the roots of unity are natural points for polynomial interpolation on the unit circle. Since the roots of unity are conjugate-symmetric, it is reasonable in light of the observations just made to ask how to choose points in $[-1, 1]$ so that the points for the corresponding interpolation problem on $\mathbb{T}$ are those roots. The most important case occurs when the number of roots of unity is even, in which the answer to this question is given by the following:

$$x_k^{(2)} = \cos\left(\frac{k\pi}{K-1}\right), \qquad 0 \le k \le K-1. \tag{1.13}$$

(For $K = 1$, we set $x_0^{(2)} = 0$.) Since both 1 and -1 belong to this grid, the interpolation problem on the unit circle will have $2K - 2$ points, and it is easily checked that the points $x_k^{(2)}$ are the real parts of the $2(K-1)$th roots of unity. These points are known as the *Chebyshev points of the second kind* in $[-1, 1]$ or sometimes the *Chebyshev extreme points* or *Chebyshev–Lobatto points*, as they are the points at which $T_{K-1}(x)$ assumes its extreme values on $[-1, 1]$. For $K \ge 2$, they are also the roots of $(1 - x^2)U_{K-2}(x)$, where U_k is the kth *Chebyshev polynomial of the second kind.* As we will not need the polynomials U_k later, we will omit their definition here.

Another important case occurs when the interpolation points on $\mathbb{T}$ are the $(2K)$th roots of unity, shifted along the circle counterclockwise by an angle of $\pi/(2K)$ so that the resulting points are still conjugate-symmetric and equispaced on the circle but neither 1 nor -1 is among them.[9] In this case, the corresponding points on $[-1, 1]$ are the *Chebyshev points of the first kind*

$$x_k^{(1)} = \cos\left(\frac{(2k+1)\pi}{2K}\right), \qquad 0 \le k \le K-1. \tag{1.14}$$

These points are the zeros of $T_K(x)$, as is readily verified using (1.11). They are also, curiously, the real parts of the Kth roots of i, a fact that will play a minor role later in Section 5.5.5.

One can also derive point sets in $[-1, 1]$ corresponding to the $(2K - 1)$th roots of unity (a conjugate-symmetric set of equispaced points on $\mathbb{T}$ that includes 1 but not -1) or the reflection

[9] In terms of quadrature, these points are the nodes for the midpoint rule on $\mathbb{T}$, whereas the roots of unity are those for the trapezoid rule.

of those roots across the imaginary axis (a set that includes -1 but not 1). These points are the zeros of $(1+x)V_{K-1}(x)$ and $(1-x)W_{K-1}(x)$, where V_k and W_k are what Mason and Handscomb call *Chebyshev polynomials of the third and fourth kinds*, respectively [81]. As these grids are less important than (1.13) and (1.14) in practice, and as we shall not need them later, we will not say anything further.

The grids (1.13) and (1.14) have a number of properties that make them well-suited to numerical computation. Much more will be said about this in the sections that follow. For now, we just note that their barycentric weights are known in closed form: for (1.13), they are

$$\nu_k^{(2)} = (-1)^k \frac{2^{K-2}}{K-1}, \qquad 1 \le k \le K-2,$$

and half this value for $k=0$ and $k=K-1$, while for (1.14), they are

$$\nu_k^{(1)} = \frac{2^{K-1}}{K} T_{K-1}\big(x_k^{(1)}\big) = (-1)^k \frac{2^{K-1}}{K} \sin\left(\frac{(2k+1)\pi}{2K}\right), \quad 0 \le k \le K-1. \tag{1.15}$$

Moreover, using the connections between interpolation in these sets of points and interpolation in equispaced points on the unit circle just outlined, one can develop simple and efficient methods based on the fast Fourier transform to compute the coefficients of the interpolant in the T_k basis. That is, one can pass between the representation of the interpolant in terms of its values on either of the grids (1.13) and (1.14) and its representation as a series of the form (1.12) using only $O(K \log K)$ operations instead of the $O(K^2)$ operations that would be required to solve the underlying linear system by standard methods.

As the second-kind points (1.13) are used more often in practice, we will focus on them for the remainder of this chapter, though the first-kind points (1.14) will enter into some of our work in Chapter 5. By and large, there is little difference between the two grids from either a theoretical or a practical standpoint. Perhaps the most significant difference is that the second-kind points include the endpoints ± 1, while those of the first kind do not. This has some implications for practical work, e.g., when the grids are used as the foundation for spectral collocation methods for solving boundary value problems [28]. For a recent study focusing on the first-kind points that discusses some of these issues in detail, see [151].

1.2.5 Interpolation and Approximation

One does not typically pursue polynomial interpolation as an end in itself. Rather, one usually constructs an interpolant as a means of approximating or modeling some other function. The hope is that the interpolant will be easier to analyze and/or compute with than the original function. If the interpolant approximates the original function sufficiently well, it can be used in place of the original function wherever the latter is required. This is the principle underlying the Chebfun software package for numerical computing with functions [10, 29], which inspired much of the work in this thesis and which we will use, often without comment, for many of our numerical illustrations. For a brief description of Chebfun, see Section 1.5.

Thus, it is important for us to understand the extent to which polynomials and, in particular, polynomial interpolants can be used to approximate more general functions. We will devote the next few sections to this task. The classic setting for this discussion is uniform approximation in the space $C([-1,1])$ of continuous, complex-valued functions on $[-1,1]$, and we will mostly restrict ourselves to this case. We denote the uniform (supremum) norm on $C([-1,1])$ by $\|\cdot\|_\infty$.

Perhaps the most basic result concerning polynomial approximation of functions is the celebrated *Weierstrass approximation theorem*,[10] [133, Ch. 6, p. 43], [147] which states that polynomials are dense in $C([-1,1])$:

Theorem 1.1 (Weierstrass approximation theorem). *Let $f \in C([-1,1])$. For all $\varepsilon > 0$, there is a polynomial p such that $\|f - p\|_\infty < \varepsilon$.*

In words, any continuous function on $[-1,1]$ can be uniformly approximated arbitrarily well by polynomials. This is not the end of the story, however, for while there are constructive proofs of the Weierstrass theorem, the methods they describe are generally not favorable from a computational standpoint. Nevertheless, the theorem gives one hope that the enterprise of approximating functions by polynomials is at least possible.

Perhaps the next natural question to ask is how well one can approximate a function using a polynomial of a given complexity as measured by its degree. That is, given $f \in C([-1,1])$ and $N \geq 0$, how small can we make the error $\|f - p\|_\infty$ if we are allowed to choose p from $\mathcal{P}_N$? One can show that there is a unique $p \in \mathcal{P}_N$ that minimizes this error; we call this polynomial the *best approximation* to f from $\mathcal{P}_N$. If f is real-valued, it can be characterized by the property that $f - p$ assumes its maximum magnitude on $[-1,1]$ in at least $N+2$ points, alternating in sign from one point to the next, a result known as the *equioscillation theorem* [133, Ch. 10].

The error in best approximations was studied by Jackson in the early 20th century [63, 64], and there are several results bearing his name that bound it in terms of the regularity of the function being approximated.[11] The version of his results that we present can be found in [20, Ch. 4, §6, p. 147]. In the first part of this theorem, regularity is measured using the *modulus of continuity*, which is defined for $f \in C([-1,1])$ and $\delta \geq 0$ by

$$\omega_f(\delta) = \sup_{\substack{x,y \in [-1,1] \\ |x-y| \leq \delta}} |f(x) - f(y)|.$$

That is, $\omega_f(\delta)$ measures the maximum amount by which f can change over an interval of width δ. Note that as f is uniformly continuous on $[-1,1]$, $\omega_f(\delta) \to 0$ as $\delta \to 0$. We recall that $f \in C([-1,1])$

[10] For an extended discussion of this theorem and its many proofs and generalizations, see [103].

[11] More generally, any theorem that does this is often referred to as a theorem of *Jackson type*.

is said to be *Hölder continuous* with exponent α, $0 < \alpha \leq 1$ if there is a constant L such that $|f(x) - f(y)| \leq L|x-y|^{\alpha}$. If f is Hölder continuous with exponent $\alpha = 1$, we say that f is *Lipschitz continuous*, and the associated constant is called the *Lipschitz constant.* We denote by $C^k([-1,1])$ the space of all real-valued functions with k continuous derivatives on $[-1,1]$.

Theorem 1.2 (Jackson's theorems)**.** *Let $f \in C([-1,1])$, and suppose that p_N is the best approximation to f from $\mathcal{P}_N$.*

(i) $\|f - p_N\|_\infty \leq \omega_f\big(\pi/(N+1)\big)$.

(ii) If f is Hölder continuous with constant L and exponent α, $0 < \alpha \leq 1$, then $\|f - p_N\|_\infty \leq L\pi^\alpha/(N+1)^\alpha$.[12]

(iii) If f is Lipschitz continuous with constant L, then $\|f - p_N\|_\infty \leq (L\pi/2)/(N+1)$.

(iv) If $f \in C^k([-1,1])$ and $N \geq k$, then $\|f-p_N\|_\infty \leq (\pi/2)^k \|f^{(k)}\|_\infty / \big((N+1)(N)\cdots(N-k+2)\big)$.

Thus, the smoother the function f, the more rapidly we can expect its best approximations to converge as the degree N tends to infinity. Note that convergence is already guaranteed by Theorem 1.1; the utility of Theorem 1.2 is that it enables us to estimate the *rate* at which it happens for a given function.

In Chapter 3, we will need a Jackson-type theorem for functions with a derivative that is Hölder continuous. Such a result can be easily deduced from those already given together with the following theorem, which bounds the error in the best degree-N approximation to a function f by that in the best degree-$(N-k)$ approximation to its kth derivative.

Theorem 1.3. *Let $f \in C^k([-1,1])$, $k \geq 0$, and suppose that $N \geq k$. If p_N and q_{N-k} are the best approximations to f and $f^{(k)}$ from $\mathcal{P}_N$ and $\mathcal{P}_{N-k}$, respectively, then*

$$\|f - p_N\|_\infty \leq \frac{(\pi/2)^k}{(N+1)(N)\cdots(N-k+2)} \|f^{(k)} - q_{N-k}\|_\infty.$$

Proof. The proof of the result for $k = 1$ is given in [20, p. 148]. The result for general k follows by induction. □

The result we need for our later work is:

Theorem 1.4. *Let $f \in C^k([-1,1])$, $k \geq 0$, and suppose that $f^{(k)}$ is Hölder continuous with exponent α, $0 < \alpha \leq 1$, and constant L. Let $N \geq k$. If p_N is the best approximation to f from $\mathcal{P}_N$, then*

$$\|f - p_N\|_\infty \leq \frac{L(\pi/2)^k \pi^\alpha}{(N+1)(N)\cdots(N-k+2)(N-k+1)^\alpha}.$$

Proof. Combine part (ii) of Theorem 1.2 with Theorem 1.3. □

[12] This theorem is not stated directly in [20, Ch. 4, §6]; however it is implicit in [20, p. 149, Problem 12] and can be easily deduced from part (i) of the theorem.

Best approximations can be computed numerically using the *Remez algorithm* [108, 109], which iteratively refines an initial guess for the solution until it satisfies the equioscillation characterization mentioned above. Unfortunately, this algorithm is relatively expensive, and implementing it requires some attention to detail for it to work robustly at high degrees.[13] Polynomial interpolants are much easier to compute. It would be nice if their approximation properties were powerful enough to provide an acceptable alternative to best approximations for practical applications.

Thus, we turn to the study of the approximation properties specifically of polynomial interpolants. The first result we present is due to Marcinkiewicz [78]. Plainly, it says that for any real-valued[14] $f \in C([-1,1])$, one can devise a scheme for interpolating it by polynomials of increasing degree such that the interpolants converge uniformly to f:

Theorem 1.5 (Marcinkiewicz's theorem). *Let $f \in C([-1,1])$ be a real-valued function. There is a sequence $\{\{x_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ of sets of (distinct) interpolation points in $[-1,1]$ such that p_K converges to f uniformly on $[-1,1]$ as $K \to \infty$, where $p_K \in \mathcal{P}_{K-1}$ is the polynomial interpolant to f in the points $\{x_{k,K}\}_{k=0}^{K-1}$.*

One can think of this theorem as a "strengthening" of the Weierstrass approximation theorem in that it tells us that we can always take the approximating polynomials to be interpolants, at least in the real-valued case.

Marcinkiewicz's theorem is excellent news, but it comes with the major drawback in that the interpolation points whose existence it posits vary with the function $f \in C([-1,1])$ being approximated.[15] It would be much more convenient if there were a single, universal interpolation scheme that worked for all f. Unfortunately, no such scheme exists. This fact is the content of the following theorem, proved by Faber[16] in 1914 [39]:

Theorem 1.6 (Faber's theorem). *Let $\{\{x_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ be any sequence of sets of (distinct) interpolation points in $[-1,1]$. There exists $f \in C([-1,1])$ such that the interpolants $p_K \in \mathcal{P}_{K-1}$ to f in the points $\{x_{k,K}\}_{k=0}^{K-1}$ do not converge uniformly to f as $K \to \infty$.*

In fact, the situation is even worse than this. The following considerable strengthening of Faber's theorem, due to Erdös and Vértesi [37, 38], shows that we can choose the function being interpolated so that the divergence takes place on a subset of $[-1,1]$ that is both of full (Lebesgue) measure and large in the sense of Baire category:[17]

[13] As a point of reference, the implementation of the Remez algorithm in Chebfun, due to Pachón [96, 98], is robust for degrees up to several thousand.

[14] The reason f must be real-valued is that the proof of the result proceeds by using the equioscillation theorem to conclude that every best approximation of degree N is also an interpolant in at least $N+1$ points. It chooses the points $\{x_{k,K}\}_{k=0}^{K-1}$ to be K points at which the best approximation of degree $K-1$ interpolates f. The author is unaware of any extension of Marcinkiewicz's result for complex-valued continuous functions.

[15] See the previous footnote.

[16] Though we have presented them in the other order, Faber's theorem predates Marcinkiewicz's historically.

[17] Baire's concept of category gives a way to assess the "size" of a set that is based purely on topological notions as opposed to the more "analytical" concept of measure. A set is "small" in the sense of Baire category if it can be

Theorem 1.7. *Let $\{\{x_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ be any sequence of sets of (distinct) interpolation points in $[-1,1]$. There exists $f \in C([-1,1])$ such that the interpolants $p_K \in \mathcal{P}_{K-1}$ to f in the points $\{x_{k,K}\}_{k=0}^{K-1}$ satisfy*

$$\limsup_{K\to\infty} |p_K(x)| = \infty$$

for almost every $x \in [-1,1]$. Moreover, the set of divergence is of the second category in $[-1,1]$.

This theorem and other results related to the divergence of polynomial interpolation can be found in [128, Ch. IV].

1.2.6 Convergence of Chebyshev Interpolation

The last two theorems make the picture for interpolation seem pretty bleak; however, further study shows that they actually reflect more the bad behavior of arbitrary continuous functions than they do the power or lack thereof of polynomial interpolation. Most functions that arise in practice are not merely continuous. It turns out that by requiring just a little additional regularity beyond continuity in the functions being approximated, one *can* devise polynomial interpolation schemes that work for all of them at once. Perhaps the most important of these schemes—and certainly one that is extremely useful in numerical computation—is *Chebyshev interpolation*, or interpolation in Chebyshev points. In keeping with our statements at the end of Section 1.2.4, we will consider only interpolation in the second-kind Chebyshev points (1.13). Similar results hold for the first-kind points (1.14).

First, we have the following theorem, which gives an extremely general criterion under which Chebyshev interpolation is guaranteed to converge:

Theorem 1.8 (Dini–Lipschitz test). *If $f \in C([-1,1])$ satisfies $\lim_{\delta\to 0} \omega_f(\delta)\log\delta = 0$, then $p_K \to f$ uniformly on $[-1,1]$ as $K \to \infty$, where $p_K \in \mathcal{P}_{K-1}$ is the interpolant to f in the K Chebyshev points* (1.13).

This theorem is a consequence of part (i) of Theorem 1.2 and Theorems 1.13 and 1.15, below.[18] We emphasize that the criterion established in this theorem is extremely weak and that virtually every continuous function that one encounters in practical applications will satisfy it. For instance any function that is Hölder continuous meets the hypotheses of the theorem, and Hölder continuity is

expressed as a countable union of sets whose closures have empty interior; such sets are said to be of the *first category.* Sets that do not have this property are of the *second category.*

In a way, the notions of "size" expressed by measure and category are unrelated: one can construct examples of sets of full measure that are of the first category as well as sets that have measure zero but are of the second category. Nevertheless, this is not the end of the story. For further details on the relationship between the two, we refer the reader to [95].

[18] See also [20, pp. 129, 146]. The theorem stated in this reference is for Chebyshev projection—approximations derived by truncations of expansions in Chebyshev series (see footnote 29)—not Chebyshev interpolation, but the interpolation version is easily deduced from the arguments given by using the Lebesgue constant for interpolation (see Section 1.2.8) in place of that for projection.

already a rather weak requirement. Thus, in spite of Theorems 1.6 and 1.7, Chebyshev interpolation succeeds for an enormous class of interesting functions.

With more information about the regularity of f, we can say more about the rate at which the convergence takes place. If f is differentiable, convergence occurs at an algebraic rate governed by the number of derivatives that f possesses. Specifically, we have the following result, which can be found in [133, Ch. 7, p. 53] and should be compared with part (iv) of Theorem 1.2. Recall that a function f is *absolutely continuous* on an interval $[a, b]$ if it is the indefinite (Lebesgue) integral of a function $g \in L^1([a, b])$; in this case, f' exists and is equal to g almost everywhere. The *total variation* of a function f on $[-1, 1]$ is the quantity $V(f) = \sup \sum_{j=1}^{k} |f(t_j) - f(t_{j-1})|$, where the supremum is taken over all partitions $-1 = t_0 < t_1 < \cdots < t_k = 1$ of $[-1, 1]$. We say that f is of *bounded variation* on $[-1, 1]$ if $V(f)$ is finite.

Theorem 1.9. *Suppose that f and its derivatives through $f^{(\nu-1)}$ are absolutely continuous on $[-1, 1]$ for some integer $\nu \geq 0$. Suppose further that the νth derivative $f^{(\nu)}$, which exists almost everywhere on $[-1, 1]$, is of bounded variation. Then, for any $K > \nu + 1$,*

$$\|f - p_K\|_\infty \leq \frac{4V(f^{(\nu)})}{\pi\nu(K - 1 - \nu)^\nu},$$

where $p_K \in \mathcal{P}_{K-1}$ is the interpolant to f in the K points (1.13).

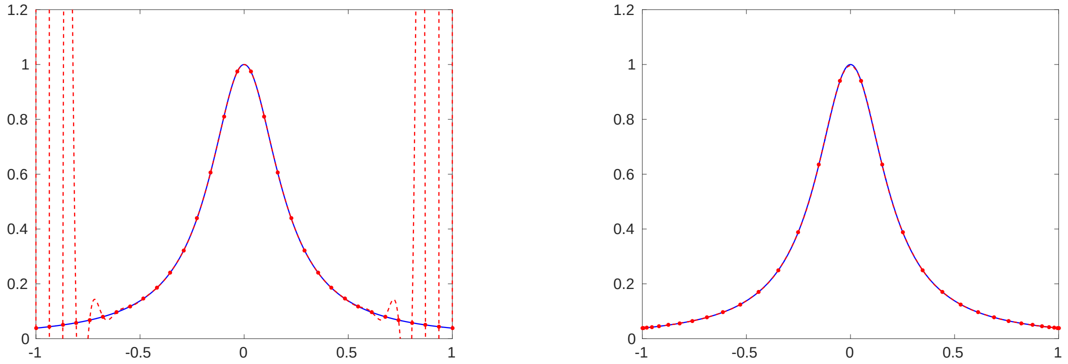
If the function f is not merely differentiable but holomorphic on $[-1, 1]$, the rate of convergence is geometric. To state the relevant theorem, we need to introduce the notion of a Bernstein ellipse. For $\rho > 1$, the *Bernstein ellipse* with parameter ρ, denoted E_ρ, is the image of the open disc of radius ρ centered at 0 in the complex plane under the Joukowski map $J(z) = (z + z^{-1})/2$. It is easily checked that E_ρ is an ellipse with foci at ± 1 and that the lengths of its major and minor semiaxes sum to ρ. As $\rho \to 1$ from above, E_ρ "collapses" to the interval $[-1, 1]$. A compactness argument can be used to show that any function that is holomorphic on $[-1, 1]$ is holomorphic in E_ρ for some $\rho > 1$. The following result can be found in [133, Ch. 8, p. 57].

Theorem 1.10. *Suppose that f is holomorphic in E_ρ for some $\rho > 1$ and that $|f(z)| \leq M$ for all $z \in E_\rho$. Then, for each $K \geq 1$,*

$$\|f - p_K\|_\infty \leq \frac{4M\rho^{-(K-1)}}{\rho - 1},$$

where $p_K \in \mathcal{P}_{K-1}$ is the interpolant to f in the K points (1.13).

We summarize the content of these theorems concisely as follows. If f has ν derivatives, with $f^{(\nu)}$ being of bounded variation, the interpolants p_K to f in the points (1.13) converge to f uniformly as $K \to \infty$ at a rate of $O(K^{-\nu})$. If f is holomorphic on $[-1, 1]$, then p_K converges to f uniformly at a rate of $O(\rho^{-K})$, where ρ is the parameter that determines the largest Bernstein ellipse in which f is holomorphic. These fundamental theorems form the foundation for the success of Chebyshev interpolation in practical applications.



(a) (b)

Figure 1.1: The Runge phenomenon. Interpolation of the function $f(x) = 1/(1 + 25x^2)$ on $[-1, 1]$ in $K = 32$ (a) equispaced points and (b) second-kind Chebyshev-points. The solid blue line is the graph of f. The dashed red lines are the graphs of the interpolants, and the red dots mark the interpolation points. The equispaced interpolant oscillates wildly near the interval endpoints, while the Chebyshev interpolant is well-behaved.

1.2.7 Distribution of Interpolation Points, The Runge Phenomenon

It is natural to wonder if the results of the previous section apply, perhaps with minor adjustments, to schemes other than Chebyshev interpolation. Is lack of regularity the only barrier to getting polynomial interpolation to succeed? In other words, is polynomial interpolation in *any* set of points guaranteed to converge in theory if the function is only a little smooth, e.g., if it satisfies the criterion of Theorem 1.8 or another one that is similarly weak? What about in practice?

The answer to all of these questions is decidedly negative. In this section and the next, we will explain why this is the case. At the same time, we will gain some insight into why Chebyshev interpolation works so well both in theory and in practice. To do this, we will contrast it with a scheme that famously does *not* work well: interpolation in equispaced points in $[-1, 1]$.

Equispaced interpolation is so poorly behaved that is not guaranteed to converge even if the function being interpolated is holomorphic on $[-1, 1]$. This fact is termed the *Runge phenomenon* after Carl Runge, who developed a general theory of convergence and illustrated it by the example of equispaced interpolation of the function $f(x) = 1/(1 + 25x^2)$ [115]. Chebyshev interpolation of this function, on the other hand, is perfectly well-behaved. This is illustrated in Figure 1.1, which shows the polynomial interpolants to this function in $K = 32$ equispaced points and second-kind Chebyshev points (1.13). The Chebyshev interpolant matches f closely over the entire interval. The equispaced interpolant, on the other hand, oscillates wildly near the interval endpoints. As $K \to \infty$, the endpoint oscillations in the equispaced interpolant become increasingly severe while the Chebyshev interpolant converges uniformly to f at a geometric rate in accordance with Theorem 1.10.

What makes Chebyshev interpolation succeed for this function, while equispaced interpolation fails? The fundamental reason has to do with how the two sets of interpolation points are asymptotically distributed. Equispaced points are distributed uniformly over $[-1, 1]$, while the Chebyshev points (of both kinds) cluster quadratically near the interval endpoints. The asymptotic distribution of the points turns out to be the key factor in determining the convergence or lack thereof of polynomial interpolation for functions that are holomorphic on $[-1, 1]$: convergence for all such functions is guaranteed if and only if the interpolation points asymptotically follow the Chebyshev distribution. Other distributions can still yield convergence on $[-1, 1]$ for *some* holomorphic functions, but the functions will need to be holomorphic in a larger region of the complex plane rather than just on $[-1, 1]$ itself, the precise shape and size of the region depending on the distribution.

One way to get a handle on these concepts is via the *Hermite integral formula* [133, Ch. 11, p. 82], which expresses the error in a polynomial interpolant using a contour integral; indeed, this was the technique used by Runge in his investigations. Returning momentarily to the setting of polynomial interpolation in general points in $\mathbb{C}$, we have:

Theorem 1.11 (Hermite integral formula)**.** *Suppose that f is holomorphic in a region $E \subset \mathbb{C}$ and that $z_0, \ldots, z_{K-1}$ are K distinct*[19] *points in E. Let p be the polynomial interpolant of degree $K - 1$ to f in these points. If Γ is any positively oriented contour in E that encloses these points and the point z, then*

$$f(z) - p(z) = \frac{1}{2\pi i} \int_\Gamma \frac{\ell(z)}{\ell(\zeta)} \frac{f(\zeta)}{\zeta - z} \, d\zeta, \tag{1.16}$$

where $\ell(z) = (z - z_0) \cdots (z - z_{K-1})$ is the node polynomial for the grid.

The key observation to make from (1.16) is that the interpolation error depends on the behavior of the ratio $\ell(z)/\ell(\zeta)$, where ζ lies on the contour Γ, on and inside of which f is assumed to be holomorphic. The more rapidly this ratio decays as K increases, the more rapidly $p(z)$ converges to $f(z)$. On the other hand, if the ratio grows with K, we will see divergence.

In more detail, if we are working on $[-1, 1]$, select a contour Γ that encloses that interval. Defining $\gamma_K(x, \zeta) = \left|\ell(\zeta)/\ell(x)\right|^{1/K}$, where $x \in [-1, 1]$ and $\zeta \in \Gamma$, we see that if $\gamma_K(x, \zeta) \geq C(x) > 1$ for all $\zeta \in \Gamma$ and large enough K, we will obtain convergence of $p(x)$ to $f(x)$ at a rate of $O\big(C(x)^{-K}\big)$. If C can be chosen independently of x, we will get uniform convergence of p to f on $[-1, 1]$ at a rate of $O(C^{-K})$.

These observations suggest that we study the asymptotic level curves of $\ell(x)^{1/K}$ for an interpolation scheme in the complex plane to better understand how it behaves. From the observations of the preceding paragraph, we expect the worst-case behavior to be governed by the ratio of the magnitude of the smallest value of $|\ell(\zeta)|^{1/K}$ on Γ to that of the largest value of $|\ell(x)|^{1/K}$ on $[-1, 1]$; we need this ratio to be larger than 1 to ensure convergence on all of $[-1, 1]$. Noting that ℓ grows in magnitude the farther away we are from the interval, we see that we need to be able to take Γ

[19] Our assumption that the points are distinct is merely for consistency with our setting thus far. The theorem still holds if some of the points coincide and the interpolation problem is interpreted in the Hermite sense; see footnote 2.

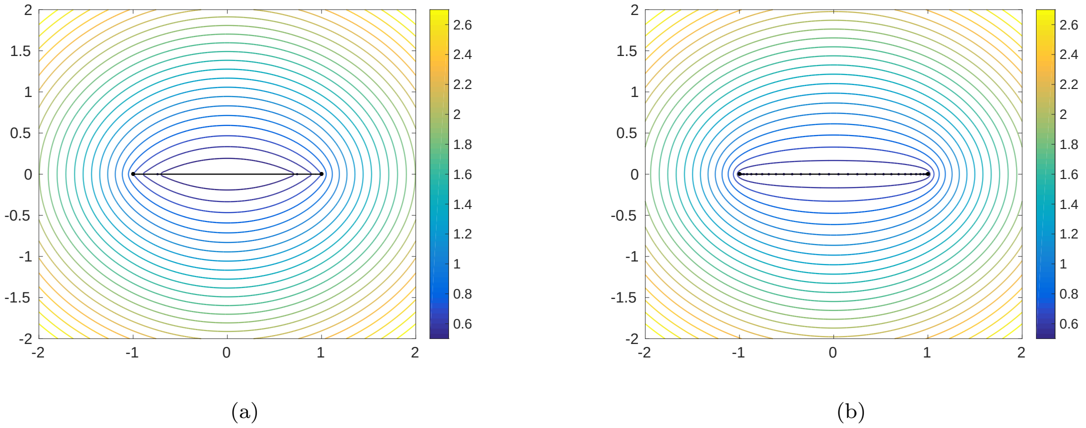


Figure 1.2: Level curves of $\ell(x)^{1/K}$ for $K = 32$ (a) equispaced points and (b) second-kind Chebyshev points on $[-1, 1]$. Note that each of the level curves in (b) surrounds $[-1, 1]$, while several of those in (a) do not.

far enough out into the complex plane that it encloses the first (asymptotic) level curve of $|\ell(x)|^{1/K}$ that encloses $[-1, 1]$. Of course, we will need f to be holomorphic in the region of the complex plane enclosed by this level curve in order to make this possible.

The level curves of $\ell(x)^{1/K}$ for $K = 32$ equispaced points and second-kind Chebyshev points are displayed in Figure 1.2. In the equispaced case, the first level curve that encloses $[-1, 1]$ is a football-shaped curve[20] that extends a modest distance out into the complex plane. To reiterate, for equispaced interpolation to converge on all of $[-1, 1]$, we will need f to be holomorphic within the region bounded by this curve. This explains why we saw divergence for the function $f(x) = 1/(1 + 25x^2)$, which has poles inside this region at $\pm i/5$. Instead, we are only able to obtain convergence on the middle part of $[-1, 1]$ that lies within the first level curve that does not include these poles.

For Chebyshev interpolation, the situation is completely different. The level curves, which approach Bernstein ellipses, tend not to pass through $[-1, 1]$; rather, $[-1, 1]$ is almost itself a level curve! This behavior indicates that if f is holomorphic on *any* region containing $[-1, 1]$, no matter how small, we are guaranteed convergence of the interpolants to f on all of $[-1, 1]$ as $K \to \infty$.

We emphasize that these differences in behavior are due to the asymptotic distribution of the points, since the asymptotic level curves are determined ultimately by these distributions and not the points themselves. Thus, any interpolation scheme that distributes its points in the same way as Chebyshev interpolation is also guaranteed to converge for all holomorphic f. This includes polynomial interpolation in the roots of other orthogonal polynomial systems, such as Legendre and, more generally, Jacobi polynomials [129]. One can also consider interpolation in conformally-mapped Chebyshev grids that have the same asymptotic distribution but which individually place

[20] More precisely, it is an *American* football-shaped curve.

more points in the center of the interval, giving a slightly faster rate of convergence in some cases [54], [133, Ch. 22].

Our discussion here has been informal; however, one can make all of these notions precise using tools from a branch of mathematics known as *potential theory.* In particular, the "asymptotic level curves" of which we have spoken are the level curves of a logarithmic *potential function* associated with the interpolation scheme. As presenting this subject carefully would take several pages and as we shall not need any of the details in our later work, we will refrain from doing so. Instead, we will content ourselves with stating the following very general theorem due to Fejér [41], Kalmár [67] (cited in [22]), and Walsh [143] that confirms what we have written and extends it to interpolation on more general sets in the complex plane. The version of it that we present can be found in [44, Ch. 2, p. 65]; see also Theorem 1.1 of [22].

To state the theorem, we need to introduce some additional notions; we will not give details. Let E be a compact, simply connected subset of $\mathbb{C}$. By the Riemann mapping theorem [1], there is a unique function ψ that is holomorphic on the exterior of the unit disc (i.e., the set $\{z \in \mathbb{C} : |z| > 1\}$) and that maps it one-to-one and conformally onto $\mathbb{C} \setminus E$, normalized so that $\psi(\infty) = \infty$ and $\psi'(\infty) > 0$. Expanding ψ in a power series for $|z| > 1$, we have

$$\psi(z) = cz + c_0 + \frac{c_1}{z} + \frac{c_2}{z} + \cdots ,$$

and our choice of normalization means that $c > 0$. The number c is called the *logarithmic capacity* of the boundary ∂K of K. We say that a sequence $\{\{z_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ of sets of points in E is *equidistributed*[21] *on* E if

$$\lim_{K\to\infty} M_K^{1/K} = c, \tag{1.17}$$

where $M_K = \|\ell_K(z)\|_{\infty,E}$, $\ell_K(z) = (z - z_{0,K}) \cdots (z - z_{K-1,K})$ is the node polynomial for the Kth grid, and $\|\cdot\|_{\infty,E}$ denotes the supremum norm on the space of continuous functions on E. For $R > 1$, let Γ_R denote the level curve of ψ defined by $|\psi(z)| = R$. If f is holomorphic on E, then it is holomorphic in the interior of Γ_R for some $R > 1$. Let ρ, $1 < \rho \leq \infty$, be the largest number such that f is holomorphic in the interior of Γ_ρ. A sequence $\{f_n\}_{n=1}^{\infty}$ of functions on E is said to *converge maximally* to f on E if

$$\lim_{n\to\infty} \|f - f_n\|_{\infty,E}^{1/n} = \rho^{-1}.$$

Theorem 1.12 (Fejér–Kalmár–Walsh theorem)**.** *Let E and ψ be as above, and let $\{\{z_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ be a sequence of sets of points in E. Then, $p_{f,K} \to f$ pointwise on K for all functions f holomorphic on E if and only if $\{\{z_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ is equidistributed on E, where $p_{f,K}$ is the polynomial interpolant*

[21] The term *uniformly distributed* is also frequently used; however, to avoid confusion, we prefer to reserve that term for sequences that asymptotically follow a uniform distribution on K.

of degree $K-1$ to f in the points $z_{0,K}, \dots, z_{K-1,K}$.[22] *If $\{\{z_{k,K}\}_{k=0}^{K-1}\}_{K=1}^{\infty}$ is equidistributed on E, then $p_{f,K} \to f$ maximally on E for each f holomorphic on E.*

For further information on and extensions of the concepts discussed in this section, we refer the reader to [44, Ch. 2], [76, 107, 118], [133, Ch. 11–13], and [144, Ch. VII].

1.2.8 Lebesgue Constants

The last concept that we will discuss in our tour of polynomial interpolation theory is that of the Lebesgue constant. Given a grid of K distinct points $x_0, \dots, x_{K-1} \in [-1, 1]$, the map that takes a function $f \in C([-1, 1])$ to its polynomial interpolant p_f in those points is a linear operator on $C([-1, 1])$, in fact, a linear projection onto $\mathcal{P}_{K-1}$. The *Lebesgue constant* Λ for the grid is the norm of this projection:

$$\Lambda = \sup_{\|f\|_\infty \le 1} \|p_f\|_\infty, \tag{1.18}$$

Since the values of f on the grid are the only ones that affect the behavior of the interpolant, we can equivalently write (1.18) as

$$\Lambda = \sup_{\substack{|f(x_k)| \le 1 \\ 0 \le k \le K-1}} \|p_f\|_\infty.$$

When working with a system of interpolation grids parameterized by the grid length, we will frequently write Λ_K for the Lebesgue constant for the grid of length K.

The traditional way to study the Lebesgue constant for a grid is via the properties of the corresponding *Lebesgue function*

$$L(x) = \sum_{k=0}^{K-1} |\ell_k(x)|, \tag{1.19}$$

where ℓ_k is the kth Lagrange basis polynomial defined by (1.2). One can show that

$$\Lambda = \sup_{x \in [-1,1]} L(x), \tag{1.20}$$

i.e., the Lebesgue constant is the maximum value of the Lebesgue function on $[-1, 1]$.

Informally, the Lebesgue constant tells us how much bigger we can expect a polynomial interpolant to be than the function it interpolates or, in other words, the amount by which the interpolation process magnifies the function. In this sense, it gives us an absolute condition number for the interpolation problem: the norm of the difference between the interpolant of a function f and that of a perturbation $f + g$ of f is $\|p_{f+g} - p_f\|_\infty = \|p_g\|_\infty \le \Lambda \|g\|_\infty$. Thus, when passing to the interpolant, a perturbation in the function can be amplified by a factor of at most the Lebesgue constant. As we will see momentarily, this fact can be very important in determining the success or failure of a polynomial interpolation scheme in the presence of rounding error.

[22] Here, the points $z_{0,K}, \dots, z_{K-1,K}$ do not need to be distinct. If some of the points coincide, the interpolation needs to be performed in the Hermite sense; see footnote 2.

Another reason Lebesgue constants are important is that they allow us to bound the difference between a polynomial interpolant of a function and the best polynomial approximation to that function of the same degree [133, Ch. 13, p. 108]:

Theorem 1.13. *Let $f \in C([-1,1])$. Let $x_0, \ldots, x_{K-1}$ be K (distinct) points in $[-1,1]$, and let $p \in \mathcal{P}_{K-1}$ be the polynomial interpolant to f in these points. Let p^* be the best polynomial approximation to f of degree $K-1$ in $C([-1,1])$. Then,*

$$\|f - p\|_\infty \leq (1 + \Lambda)\|f - p^*\|_\infty,$$

where Λ is the Lebesgue constant for the points x_k.

The upshot of this theorem is that if the Lebesgue constant for the grid is not too large, then polynomial interpolants in that grid are almost best approximations. This is tremendously useful, since, as we mentioned in Section 1.2.5, true best approximations are expensive to compute, while polynomial interpolants are simple. Since the number of applications in which a truly optimal solution is required (as opposed to one which is merely near-optimal) is small, this theorem tells us that as long as we can find a satisfactory grid, we can work with interpolants in the majority of cases and still get good results.

Thus, it is clear that for an interpolation scheme to be successful, it should have a small Lebesgue constant. The following theorem, due to Erdös [36] and Brutman [18] (with weaker forms known even earlier), establishes a bound on just how small it can be for an arbitrary grid of size K:

Theorem 1.14. *The Lebesgue constant for polynomial interpolation in any set of K distinct points in $[-1,1]$ satisfies*

$$\Lambda \geq \frac{2}{\pi} \log K + C,$$

where $C = (2/\pi)\big(\gamma + \log(4/\pi)\big) = 0.52125\ldots$, and γ is the Euler–Mascheroni constant.

It follows that the Lebesgue constant for *any* interpolation scheme must grow[23] at least as fast as $O(\log K)$ as $K \to \infty$. The next result shows that the Lebesgue constant for Chebyshev interpolation attains the optimal asymptotic growth rate.[24] In this sense, it is as well-behaved as one could ever ask.

[23] This is the key ingredient in the proof of Faber's theorem (Theorem 1.6) on the divergence of polynomial interpolation. If there were a scheme that yielded convergence for all $f \in C([-1,1])$, then the uniform boundedness principle of functional analysis would imply that its Lebesgue constants would remain bounded as $K \to \infty$, a contradiction.

[24] It is not true, however, that the Chebyshev points (of either kind) are optimal in the sense that a Chebyshev grid minimizes the Lebesgue constant over all grids of the same length. Points that satisfy the latter condition are termed "optimal points"; they are known to exist and to be unique, and their Lebesgue functions can be characterized via an equioscillation property [25]. Finding an explicit representation for the optimal points remains an open problem. For further discussion, see [128, Ch. III].

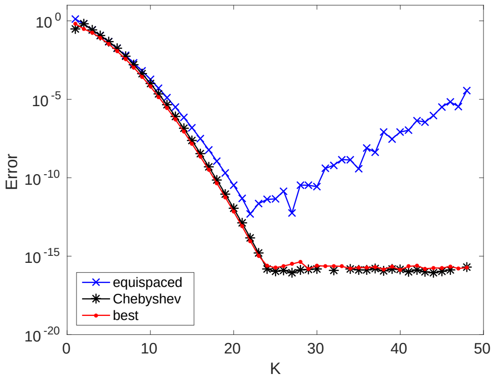


Figure 1.3: Numerically computed infinity-norm errors in the interpolants to $f(x) = e^{5(x-1)}$ in K equispaced points (blue crosses) and (second-kind) Chebyshev points (red dots) for various values of K, together with the errors in the best approximations (black stars) of the corresponding degrees. In spite of the fact that equispaced interpolation should converge for this function in theory, it diverges numerically owing to the poor conditioning of the interpolation problem, which is reflected in the exponentially growing Lebesgue constant.

Theorem 1.15. *The Lebesgue constant for polynomial interpolation on* $[-1, 1]$ *in a grid of* K *second-kind Chebyshev points* (1.13) *satisfies*

$$\Lambda_K \leq \frac{2}{\pi} \log K + 1,$$

and hence $\Lambda_K \sim (2/\pi) \log K$ *as* $K \to \infty$*.*

In contrast, the Lebesgue constant for equispaced points is very large, growing at an exponential rate:

Theorem 1.16. *As* $K \to \infty$*, the Lebesgue constant for polynomial interpolation on* $[-1, 1]$ *in a grid of* K *equispaced points satisfies*

$$\Lambda_K \sim \frac{2^K}{eK \log K}.$$

Theorem 1.15 is due to Ehlich and Zeller [35]. Theorem 1.16 was discovered independently by Turetskii [136] and Schönhage [125]. All three of these results are presented in [133, Ch. 15, p. 109], to which we refer the reader for further discussion and historical information.

Recalling our discussion from above, the explosive growth in the Lebesgue constant for equispaced points means that equispaced interpolation becomes very badly conditioned as the grid size increases. This causes it to behave poorly in the presence of rounding errors even when it should theoretically succeed. This fact is illustrated in Figure 1.3, which displays the infinity-norm errors on $[-1, 1]$ of the interpolants in equispaced and (second-kind) Chebyshev points to the function $f(x) = e^{5(x-1)}$ for several grid lengths K as well as those of the best polynomial approximations to f of the corresponding degrees, all computed in standard IEEE double precision arithmetic using Chebfun.

The function f is entire (i.e., holomorphic in the entire complex plane), so there is no issue of the Runge phenomenon; interpolants in equispaced points will converge to it in theory. Numerically, however, we observe that the error in the equispaced interpolants decays only for grids up to length $K = 22$, after which it increases steadily due to the amplification of rounding errors in the interpolation process. In contrast, the Chebyshev interpolants converge nicely with the error reaching the level of machine precision for grids of size $K = 24$ and larger. Their errors are only slightly larger than those of the best approximations, in keeping with Theorems 1.13 and 1.15, thanks to the slow growth of $\log K$ as $K \to \infty$.

1.3 Trigonometric Interpolation

The next form of interpolation that we consider is *trigonometric interpolation*, in which data is fit using trigonometric polynomials instead of algebraic ones. Recall that a *trigonometric polynomial* is a function of the form

$$t(x) = \sum_{k=-N}^{N} c_k e^{ikx}. \tag{1.21}$$

The integer N is the *degree* of the trigonometric polynomial. We denote the space of all degree-N trigonometric polynomials by $\mathcal{T}_N$.

Trigonometric interpolation is an appropriate tool to use when the function to be approximated is 2π-periodic; it is perhaps the simplest method available for approximating such functions. The precise interpolation problem we consider is the following:

> Given $K = 2N+1$ distinct interpolation points $x_{-N}, \ldots, x_N \in [0, 2\pi)$ and K corresponding values $f_{-N}, \ldots, f_N \in \mathbb{C}$, find a trigonometric polynomial $t \in \mathcal{T}_N$ such that $t(x_k) = f_k$ for each k.

The solution to this problem always exists and is unique; this can be established via an argument similar to that given for the polynomial interpolation problem in Section 1.2.1.

The standard reference on all matters related to trigonometric polynomials and series is Zygmund's classic monograph [155]. Another useful reference on these subjects is the paper [150], which highlights the basics of the theory in addition to describing the recently developed trigonometric interpolation capabilities of Chebfun.

1.3.1 Relation to Polynomial Interpolation

Trigonometric and polynomial interpolation are closely connected. Indeed, upon setting $z = e^{ix}$, the trigonometric polynomial (1.21) becomes the Laurent polynomial

$$P(z) = \sum_{k=-N}^{N} c_k z^k,$$

and, hence, setting $z_k = e^{ix_k}$, $-N \leq k \leq N$, our problem becomes one of Laurent polynomial interpolation of the data f_k in the points z_k on the unit circle. This can be converted into an ordinary polynomial interpolation problem of degree $2N$ by multiplying P by z^N and the data f_k by z_k^N. In more detail, let p be the polynomial of degree $2N$ such that $p(z_k) = e^{iNx_k} f_k$ for each k. Write $p(z) = \sum_{k=0}^{2N} c_{k-N} z^k$ and define $t(x) = e^{-iNx} p(e^{ix})$. It is easily checked that t has the form (1.21) and that $t(x_k) = f_k$ for each k. Thus, every trigonometric interpolation problem is exactly equivalent to a polynomial interpolation problem in which the interpolation points lie on the unit circle.

More generally, each theorem about polynomial interpolation or approximation that we stated in the previous section has an analogue for trigonometric interpolation or approximation. In fact, our presentation is backward in the sense that the trigonometric versions of the results are often used to prove the polynomial ones! The key observation to make is that interpolation or approximation of a function $f \in C([-1,1])$ by polynomials is equivalent to interpolating or approximating the function $F(z) = f(\operatorname{Re} z)$ on the unit circle using Laurent polynomials. We have already seen this principle at work in Section 1.2.4, in which it was shown that every polynomial interpolation problem on $[-1,1]$ is equivalent to a Laurent polynomial interpolation problem on the unit circle in points that are conjugate symmetric.

Since this correspondence allows us to transform most results on either polynomial or trigonometric interpolation into the other with relatively little effort,[25] rather than laboriously repeat every result of the previous section in full with the necessary slight adjustments, we will simply refer to the theorems already stated as needed, pointing out the differences when they are important. Some of the main points to keep in mind are:

- For trigonometric interpolation, equispaced points are the natural interpolation points, whereas for polynomial interpolation on $[-1,1]$, that role is played by the Chebyshev points. On the unit circle, the natural points for polynomial or Laurent polynomial interpolation are the roots of unity. By "natural", we mean that these point sets have simple, explicit representations, have the proper distributions in the potential-theoretic sense that are required to ensure convergence of interpolation for all holomorphic functions (see Section 1.2.7), and have Lebesgue constants with the optimal asymptotic growth rate (see Sections 1.2.8 and 1.3.3).

- When doing trigonometric interpolation of periodic functions that are holomorphic on $\mathbb{R}$, the natural domain for holomorphic functions is a strip centered on the real axis with half-width ρ. On the unit circle, it is an annulus $\rho^{-1} \leq |z| \leq \rho$. For polynomial interpolation, it is a Bernstein ellipse E_ρ. By "natural", we mean that if a function is holomorphic in the appropriate region,

[25] Often, this additional effort is trivial and can be done on sight with a change of variable. At other times, a little more thought may be required, cf., the deduction of the trigonometric version of the Weierstrass approximation theorem from its ordinary counterpart in [103].

trigonometric	polynomial (circle)	polynomial (interval)
$[0, 2\pi]$ or $[-\pi, \pi]$	$\mathbb{T}$	$[-1, 1]$
e^{ikx}	z^k	$T_k(x)$
equispaced points	roots of unity	Chebyshev points
strip	annulus	Bernstein ellipse

Table 1.1: Analogies between trigonometric interpolation, polynomial and Laurent polynomial interpolation on the circle, and polynomial interpolation on $[-1, 1]$.

the corresponding type of interpolation in points that follow the appropriate distribution as described in the previous item will converge at a rate of $O(\rho^{-K})$ as the number of points K tends to infinity.

- Trigonometric interpolants are naturally expressed in the complex exponential basis e^{ikx}. Polynomial and Laurent polynomial interpolants on the unit circle are naturally expressed using the monomial basis z^k. Polynomial interpolants on the interval are naturally expressed using the Chebyshev basis $T_k(x)$. These bases are favorable numerically as well as theoretically.

A brief summary of these analogies is presented in Table 1.1.

1.3.2 Barycentric Representation

All mathematicians are familiar with the Lagrange form (1.3) for a polynomial interpolant. Less widely appreciated is the fact that trigonometric interpolants, too, have Lagrange-style representations and even analogues of the barycentric formulas (1.4) and (1.6). As these formulas will be the subject of study in Chapter 2, we will spend some time discussing them carefully here.

One way to arrive at the Lagrange form of a trigonometric interpolant is via the connection to polynomial interpolation just described. Letting $z_k = e^{ix_k}$, $-N \le k \le N$ as before, let $\ell(z) = (z - z_{-N}) \cdots (z - z_N)$ be the node polynomial for the points $z_{-N}, \ldots, z_N$, and let $w_{-N}, \ldots, w_N$ be the associated barycentric weights. Then, by (1.4) and the discussion of first paragraph in the previous section, we can represent the trigonometric interpolant t in the form

$$t(x) = e^{-iNx} \ell(e^{ix}) \sum_{k=-N}^{N} \frac{w_k e^{iNx_k}}{e^{ix} - e^{ix_k}} f_k. \tag{1.22}$$

This is already a trigonometric version of (1.4), but with a little more work, we can put it into a form that is even more convenient. Using the identity

$$e^{i\alpha} - e^{i\beta} = 2ie^{i\frac{\alpha+\beta}{2}} \sin\left(\frac{\alpha - \beta}{2}\right),$$

we can write

$$\begin{aligned}\ell(e^{ix}) &= (2i)^{2N+1}e^{i\left(N+\frac{1}{2}\right)x}\prod_{k=-N}^{N} e^{i\frac{x_k}{2}}\sin\left(\frac{x-x_k}{2}\right), \\ w_k^{-1} &= (2i)^{2N}e^{iNx_k}\prod_{\substack{j=-N\\ j\neq k}}^{N} e^{i\frac{x_j}{2}}\sin\left(\frac{x-x_j}{2}\right), \\ e^{ix}-e^{ix_k} &= 2ie^{i\frac{x+x_k}{2}}\sin\left(\frac{x-x_k}{2}\right),\end{aligned}$$

and upon inserting these expressions into (1.22), we arrive at

$$t(x) = \left(\prod_{k=-N}^{N}\sin\left(\frac{x-x_k}{2}\right)\right)\sum_{k=-N}^{N}\frac{\nu_k}{\sin\left(\frac{x-x_k}{2}\right)}f_k, \tag{1.23}$$

where we have defined the trigonometric barycentric weights ν_k by

$$\nu_k^{-1} = \prod_{\substack{j=-N\\ j\neq k}}^{N}\sin\left(\frac{x_k-x_j}{2}\right).$$

This is the trigonometric analogue of (1.4) that we seek.

To derive an analogue of (1.6), we use the observation that, by the uniqueness of trigonometric interpolants,

$$1 = \left(\prod_{k=-N}^{N}\sin\left(\frac{x-x_k}{2}\right)\right)\sum_{k=-N}^{N}\frac{\nu_k}{\sin\left(\frac{x-x_k}{2}\right)}.$$

Dividing (1.23) through by this identity on both sides, we obtain the representation

$$t(x) = \frac{\displaystyle\sum_{k=-N}^{N}\frac{\nu_k}{\sin\left(\frac{x-x_k}{2}\right)}f_k}{\displaystyle\sum_{k=-N}^{N}\frac{\nu_k}{\sin\left(\frac{x-x_k}{2}\right)}}, \tag{1.24}$$

as desired. In analogy with the polynomial case, we refer to (1.23) and (1.24) as the *first* and *second trigonometric barycentric formulas*, respectively. Using these, we can derive a formula for t analogous to (1.3) by setting

$$\ell_k(x) = \prod_{\substack{j=-N\\ j\neq k}}^{N}\frac{\sin\left(\frac{x-x_j}{2}\right)}{\sin\left(\frac{x_k-x_j}{2}\right)} \tag{1.25}$$

for $-N \leq k \leq N$. We then have

$$t(x) = \sum_{k=-N}^{N} f_k\ell_k(x) \tag{1.26}$$

by (1.23).

In the important case in which the interpolation points are equally-spaced points in $[0, 2\pi)$, (1.23) and (1.24) take on particularly simple forms. In writing these, it will be convenient to index the points from 0 to $2N = K - 1$ rather than from $-N$ to N as we have done thus far. Specifically, suppose that $x_k = (k + \alpha)h$, $0 \le k \le K - 1$, where $h = 2\pi/K$ is the grid spacing and $0 \le \alpha \le 1$ is a parameter that determines the grid shift, i.e., the deviation of x_0 from 0. Then, we have

$$\prod_{k=0}^{K-1} \sin\left(\frac{x - x_k}{2}\right) = 2^{-(K-1)} \sin\left(\frac{K(x - \alpha h)}{2}\right)$$

and

$$\nu_k = \frac{2^{K-1}}{K}(-1)^k.$$

Thus, (1.23) simplifies to

$$t(x) = \frac{1}{K} \sin\left(\frac{K(x - \alpha h)}{2}\right) \sum_{k=0}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)} f_k, \tag{1.27}$$

while (1.24) becomes

$$t(x) = \frac{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)} f_k}{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)}}. \tag{1.28}$$

We observe that this latter formula is independent of α.

The Lagrange form expressed in (1.25) and (1.26) and of which (1.23) is a simple rewriting was known to Gauss [45]. The equispaced formula (1.27) appears for $\alpha = 0$ in the works of de la Vallée Poussin [26] and Henrici [55]. The second formula (1.24) seems to have been first introduced by Salzer in [122], and Henrici [55] appears to be the first to have written down its equispaced version (1.28). Berrut [12] has provided special variants of (1.24) and (1.28) for cases in which the interpolation data f_k possess odd or even symmetry. Just as for their polynomial counterparts, the primary advantage of these formulas is that they offer a way to evaluate trigonometric interpolants in just $O(K)$ operations.

Unlike their polynomial counterparts, however, the issue of the numerical stability of the trigonometric barycentric formulas is not quite as straightforward, even for the equispaced formulas (1.27) and (1.28). These matters will be the focus of our results in Chapter 2.

1.3.3 Lebesgue Constants

Since trigonometric interpolation in $K = 2N + 1$ points is a linear projection from the space $C_{2\pi}$ of continuous, real-valued, 2π-periodic functions on $\mathbb{R}$ (equipped, as usual, with the supremum norm $\|\cdot\|_\infty$) to $\mathcal{T}_N$, we can define a Lebesgue constant Λ for it as well in exactly the same manner as we did for polynomial interpolation in Section 1.2.8, and it has exactly the same interpretation. In analogy to (1.19) and (1.20), it can be shown that

$$\Lambda = \sup_{x \in [0, 2\pi)} L(x),$$

where L is the trigonometric Lebesgue function

$$L(x) = \sum_{k=-N}^{N} |\ell_k(x)|,$$

and $\ell_k(x)$ is defined by (1.25).

For equispaced points, the Lebesgue constant for trigonometric interpolation grows at the optimal rate of $(2/\pi)\log K$ as $K \to \infty$ [21]:

Theorem 1.17. *The Lebesgue constant for trigonometric interpolation in $K = 2N + 1$ equispaced points satisfies*

$$\Lambda_K \leq \frac{2}{\pi}\log K + 2,$$

and $\Lambda_K \sim (2/\pi)\log K$ as $K \to \infty$.

In fact, more is true: equispaced points actually minimize Λ_K for any fixed K [25]. That is, they constitute an *optimal* grid for trigonometric interpolation.

In Chapters 3 and 4, we will consider what happens to Λ_K when the interpolation points are not equispaced. In this case, Λ_K can in general be made arbitrarily large for any fixed $K \geq 3$ by taking two of the grid points to be close to one another. We will show, however, that if we exclude this possibility by requiring the grid to be a perturbation of an equispaced grid (in a sense that we will define precisely), the Lebesgue constant grows at a rate that is at most algebraic with a modest exponent. This shows that trigonometric interpolation in nearly equispaced points is not much worse than interpolation in exactly equispaced points, a fact that is potentially important in practice, as it happens not infrequently that applications force one to consider nonuniform grids.

1.3.4 Even-Length Interpolants

Thus far, we have only spoken of trigonometric interpolants in an odd number of points. The reason for this that a general trigonometric polynomial (1.21) has an odd number of terms/coefficients and hence an odd number of degrees of freedom associated with it. With an even number of points, the interpolation problem is either overspecified, in which case it generally has no solution, or underspecified, in which case the solution is not unique. As a simple example, consider interpolation of the data $f_0 = 1$ and $f_1 = -1$ at the points $x_0 = 0$ and $x_1 = \pi$ in $[0, 2\pi)$. In order to satisfy these two conditions, the degree N of the interpolant must be at least 1. It is easy to check that the trigonometric polynomial

$$t(x) = \left(\frac{1}{2} - \frac{\beta}{2i}\right)e^{-ix} + \left(\frac{1}{2} + \frac{\beta}{2i}\right)e^{ix} = \cos(x) + \beta\sin(x) \tag{1.29}$$

satisfies them for any $\beta \in \mathbb{C}$.

When the interpolation points are the $K = 2N$ equispaced points $x_k = 2\pi k/K$, $0 \leq k \leq K - 1$, the usual way to deal with this issue is to take the interpolant from $\mathcal{T}_N$ but require that the coefficients

c_{-N} and c_N of the highest-order terms e^{-iNx} and e^{iNx} be equal.[26] This amounts to the requirement that the interpolant to the "sawtooth" data $f_k = (-1)^k$ be a pure cosine. This is reflected in the 2-point example just given, since the coefficients of e^{-ix} and e^{ix} in (1.29) are equal if and only if $\beta = 0$, and in this case $t(x) = \cos(x)$.

One can write down barycentric formulas for even-length trigonometric interpolants similar to those for odd-length ones given above, though they are a little more complicated for general grids. For equispaced grids, however, they are simple to write down: just replace the sine in (1.27) and (1.28) with the tangent! Specifically, in the special case of an equispaced grid with shift $\alpha \in [0, 1]$, we have

$$t(x) = \frac{1}{K} \sin\left(\frac{K(x - \alpha h)}{2}\right) \sum_{k=0}^{K-1} \frac{(-1)^k}{\tan\left(\frac{x - x_k}{2}\right)} f_k, \tag{1.30}$$

instead of (1.27) and

$$t(x) = \frac{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k}{\tan\left(\frac{x-x_k}{2}\right)} f_k}{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k}{\tan\left(\frac{x-x_k}{2}\right)}}. \tag{1.31}$$

in place of (1.28).

The first formula (1.30) is, like (1.27), a special case of a general Lagrange form known to Gauss [45] and appears (with $\alpha = 0$) in [26]. A version of it also appears in the work of of M. Riesz [110, 111]. The second formula (1.31) seems to have been first written down by Henrici [55]. It is a special case of the more general second barycentric formula for even-length trigonometric interpolation introduced by Salzer [123]. Just as in the odd-length case, Berrut [12] has developed special versions for the cases in which the data possess symmetry.

Even-length trigonometric interpolants in equispaced points are encountered frequently in practice through the fast Fourier transform (FFT); indeed, the natural length for the FFT is a power of 2. For non-equispaced grids, they are much less natural than odd-length ones, and accordingly, we will not say much about this case here.

1.4 Rational Interpolation

The last scheme we consider is interpolation by rational functions, which can have significant advantages over polynomials for approximating functions with singularities.[27] The most basic form of the rational interpolation problem is the *Cauchy interpolation problem*, which reads:

[26] Compare this with what was done in Section 1.2.4 when relating interpolation in a grid on $[-1, 1]$ that includes both endpoints to an interpolation problem on the unit circle.

[27] The prototypical example of this is Newman's 1964 result showing that the best (uniform norm) type-(N, N) rational approximations to $|x|$ on $[-1, 1]$ converge to the function at a rate of $O\big(\exp(-\sqrt{N})\big)$ [92]. The best degree-N polynomial approximations, on the other hand, converge only at a rate of $O(N^{-1})$ [11]. See [133, Ch. 23, 25] for further discussion.

> Given integers $M, N \geq 0$, $K = M + N + 1$ distinct interpolation points $z_0, \ldots, z_{K-1} \in \mathbb{C}$ and K corresponding values $f_0, \ldots, f_{K-1} \in \mathbb{C}$, find polynomials $p \in \mathcal{P}_M$ and $q \in \mathcal{P}_N$ such that $r(z) = p(z)/q(z)$ satisfies $r(z_k) = f_k$ for each k.

We say that the rational interpolant is of *type* (M, N). The expression for the number of points K comes from the fact that one needs $M + 1$ and $N + 1$ conditions to specify $p \in \mathcal{P}_M$ and $q \in \mathcal{P}_N$, respectively, and that because it is their quotient that matters, p and q can only be uniquely determined up to a common constant factor. Thus, the total number of degrees of freedom present in the problem is $(M + 1) + (N + 1) - 1 = M + N + 1$, and the number of interpolation conditions is selected to match this.

1.4.1 The Linearized Problem

Unlike the polynomial and trigonometric interpolation problems, the solution to the Cauchy interpolation problem may not always exist. As a simple example, take $M = 0$ and $N = 1$ with $K = 2$ interpolation points $z_0 = 0$ and $z_1 = 1$ and corresponding data $f_0 = 0$ and $f_1 = 1$. Writing $r(z) = a_0/(b_1 z + b_0)$ for some coefficients $a_0, b_1, b_0 \in \mathbb{C}$, the only way to enforce the condition $r(0) = 0$ is to require $a_0 = 0$, but then, r is identically 0 and cannot satisfy $r(1) = 1$.

A way around this problem that also leads to an algorithm for computing rational interpolants is to linearize the problem by multiplying through by the denominator polynomial q. Rather than enforcing the condition that $p(z_k)/q(z_k) = f_k$ for each k, we require that $p(z_k) = f_k q(z_k)$. As we will see shortly, we can always find p and q that satisfy the linearized conditions; moreover, their quotient will satisfy the Cauchy interpolation conditions at each point z_k such that $q(z_k) \neq 0$. If $q(z_k) = 0$ then the Cauchy condition might be satisfied at z_k, or it might not. If not, we say that z_k is an *unattainable point* for the Cauchy interpolation problem.

Continuing with our example from above, $p(z) = 0$ and $q(z) = z - 1$ provide a solution to the linearized equations for this problem. The quotient $r(z)$ satisfies $r(0) = 0$, but $r(1) \neq 1$, so 1 is an unattainable point. As an example of a problem for which one can have $q(z_k) = 0$ and still satisfy the Cauchy conditions, consider finding a type $(1, 1)$ interpolant in the points $z_0 = -1$, $z_1 = 0$, $z_2 = 1$ to the data $f_0 = f_1 = f_2 = 1$. Taking $p(z) = q(z) = z - 1$ solves the linearized problem, and $q(1) = 0$, but $r(1) = 1$ all the same.[28]

1.4.2 Solving the Linearized Problem

There are several ways to solve the linearized problem. The approach we will take is the one described by Pachón, Gonnet, and Van Deun in [97]. As we will see in the next section, this method lends itself naturally to a technique for dealing with some of the practical problems that can arise when computing rational interpolants.

[28] Of course, $p(z) = q(z) = 1$ is a simpler solution to the same problem.

There are two key ideas underlying this approach. The first is to represent p and q with respect to a basis of polynomials $\varphi_0, \varphi_1, \ldots, \varphi_{K-1}$ which have the properties that the degree of each φ_k is exactly k and that the φ_k are orthogonal with respect to the inner product

$$\langle f, g \rangle = \sum_{k=0}^{K-1} f(z_k)\overline{g(z_k)}$$

on $\mathcal{P}_{K-1}$. By "orthogonal", we mean that

$$\langle \varphi_j, \varphi_k \rangle = \begin{cases} c_j > 0 & \text{if } j = k \\ 0 & \text{if } j \neq k. \end{cases} \tag{1.32}$$

We write p and q in this basis as follows:

$$p(z) = \sum_{k=0}^{K-1} a_k \varphi_k(z) \qquad \text{and} \qquad q(z) = \sum_{k=0}^{K-1} b_k \varphi_k(z), \tag{1.33}$$

where here we have taken $M = N = K - 1$. We will return to the issue of getting p and q to have the correct degrees shortly.

The second key idea is to consider the map that takes the coefficients of q to the coefficients of p required to satisfy the interpolation conditions, assuming that the coefficients of q have already been found. Let

$$a = \begin{bmatrix} a_0 \\ \vdots \\ a_{K-1} \end{bmatrix} \qquad \text{and} \qquad b = \begin{bmatrix} b_0 \\ \vdots \\ b_{K-1} \end{bmatrix}$$

be the vectors of coefficients in the expansions (1.33), let

$$v_k = \begin{bmatrix} \varphi_k(z_0) \\ \vdots \\ \varphi_k(z_{K-1}) \end{bmatrix}$$

for $0 \leq k \leq K-1$ be the vector of values that the basis polynomial φ_k assumes on the interpolation grid, and let

$$f = \begin{bmatrix} f_0 \\ \vdots \\ f_{K-1} \end{bmatrix}$$

be the vector of interpolation data. In [97], it is shown that

$$c \cdot a = Zb,$$

where $\cdot$ represents elementwise (or "Hadamard") multiplication, c is the vector of the values $c_j = \langle \varphi_j, \varphi_j \rangle$,

$$c = \begin{bmatrix} c_0 \\ \vdots \\ c_{K-1} \end{bmatrix},$$

and Z is the matrix

$$Z = \begin{bmatrix} v_0^* \\ \vdots \\ v_{K-1}^* \end{bmatrix} \begin{bmatrix} f_0 & & \\ & \ddots & \\ & & f_{K-1} \end{bmatrix} \begin{bmatrix} v_0 & \cdots & v_{K-1} \end{bmatrix}.$$

Briefly, the rightmost matrix in the product for Z is a Vandermonde-like matrix for the basis $\varphi_0, \ldots, \varphi_{K-1}$ over the nodes $z_0, \ldots, z_{K-1}$. Multiplying b by it on the left therefore yields a vector with the values of the polynomial q on the interpolation grid. The diagonal matrix in the middle then multiplies each of these values by the corresponding interpolation data, yielding the values of p on the grid according to the interpolation conditions. Finally, the leftmost matrix, which by (1.32) is "almost" the inverse of the rightmost one, converts the vector of values of p to the coefficient vector a up to some scalings caused by the fact that the polynomials φ_j are not normalized.

We now have a way to pass from b to a, but we are not done: we need a way to get b, and we must restrict p and q to have the desired degrees. Let

$$\widehat{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_M \end{bmatrix} \qquad \text{and} \qquad \widehat{b} = \begin{bmatrix} b_0 \\ \vdots \\ b_N \end{bmatrix}$$

be the vectors of the "true" coefficients (i.e., those that are not automatically zero per our requirements). Supposing that we have found $\widehat{b}$ already, then by the same reasoning used above, we can get a corresponding vector of numerator coefficients a via

$$c \cdot a = \begin{bmatrix} v_0^* \\ \vdots \\ v_{K-1}^* \end{bmatrix} \begin{bmatrix} f_0 & & \\ & \ddots & \\ & & f_{K-1} \end{bmatrix} \begin{bmatrix} v_0 & \cdots & v_N \end{bmatrix} \widehat{b}.$$

In general, the entries $M+1, \ldots, K-1$ of a will be nonzero, so that this multiplication really does give us a and not $\widehat{a}$; however, if we can choose $\widehat{b}$ so that these entries are zero, then we will be able to compute

$$\widehat{c} \cdot \widehat{a} = \widehat{Z}\widehat{b},$$

where

$$\widehat{Z} = \begin{bmatrix} v_0^* \\ \vdots \\ v_M^* \end{bmatrix} \begin{bmatrix} f_0 & & \\ & \ddots & \\ & & f_{K-1} \end{bmatrix} \begin{bmatrix} v_0 & \cdots & v_N \end{bmatrix},$$

and

$$\widehat{c} = \begin{bmatrix} c_0 \\ \vdots \\ c_M \end{bmatrix}.$$

Choosing $\widehat{b}$ to force $a_{M+1} = \cdots = a_{K-1} = 0$ amounts to the requirement

$$\widetilde{Z}\widehat{b} = 0,$$

where

$$\widetilde{Z} = \begin{bmatrix} v_{M+1}^* \\ \vdots \\ v_{K-1}^* \end{bmatrix} \begin{bmatrix} f_0 & & \\ & \ddots & \\ & & f_{K-1} \end{bmatrix} \begin{bmatrix} v_0 & \cdots & v_N \end{bmatrix}. \tag{1.34}$$

The matrix $\widetilde{Z}$ is a product of three matrices of dimensions $(K-M-1)\times K$, $K\times K$, and $K\times(N+1)$, respectively. Recalling that $K = M + N + 1$, we have that $\widetilde{Z}$ has dimension $N \times (N + 1)$. Being rectangular, $\widetilde{Z}$ is guaranteed to have a nontrivial null vector, so finding such a $\widehat{b}$ is always possible; this proves that the linearized rational interpolation problem always has a solution.

We thus have our procedure for computing $\widehat{a}$ and $\widehat{b}$: compute a null vector of $\widetilde{Z}$, e.g., via the singular value decomposition (SVD), to find the denominator coefficients and then apply $\widehat{Z}$ to this vector to obtain the numerator coefficients. This completes our description of the method from [97].

1.4.3 Spurious Poles, Robust Rational Interpolation

While rational approximations (and interpolants in particular) are powerful tools, they also have a reputation for being fragile to compute numerically. The essence of the matter is that rounding errors can introduce spurious poles into the approximation that destroy its accuracy in certain regions. This problem is especially severe when the degrees of the numerator and denominator are large; however, it can occur even when they are small.

As an example, consider computing a type $(2, 3)$ rational interpolant to $f(x) = 1/(x-3/2)$. The function f is a rational function of type $(1, 1)$, so the interpolant should match the function exactly. An implementation of the algorithm described in the previous section is available in MATLAB via the Chebfun (see Section 1.5) code `ratinterp`, which by default takes its interpolation points to be Chebyshev points of the second kind in $[-1, 1]$. Using this code, we compute the interpolant as follows:

```
>> [p, q] = ratinterp(@(x) 1./(x - 3/2), 2, 3, [], [], 0);
```

The outputs `p` and `q` are chebfun objects representing the numerator and denominator polynomials of the interpolant, respectively. (The purpose of the “0” final argument to `ratinterp` will be explained later.) The poles of the interpolant are just the roots of `q`:

```
>> roots(q, ’all’)
ans =
  -0.926192390512845
   0.248684045509895
   1.499999999999999
```

We see that `q` has a root at $3/2$, matching the pole in f, but it also has roots at two points in $[-1, 1]$, which will be poles of the interpolant unless they are exactly cancelled by corresponding roots in `p`. This cancellation would happen in exact arithmetic; however, it does not in the presence of rounding error. Instead, the roots of `p` only approximately cancel these roots of `q`, and the interpolant winds up with a spurious pole-zero pair, sometimes called a *Froissart doublet.* As a result, evaluations of the interpolant near these points will suffer from loss of accuracy, e.g.:

```
>> p(0.248684045509900)/q(0.248684045509900)
ans =
  -0.822916666666667
>> 1./(0.248684045509900 - 3/2)
ans =
  -0.799158674842831
```

This evaluation is accurate to only one digit.

The problem in this example is that the interpolant has more degrees of freedom than necessary to capture the behavior of the function being interpolated, so there are many choices of the numerator and denominator polynomials p and q that satisfy the linearized interpolation conditions. In terms of the quantities introduced in the previous section, this manifests itself in the form of the matrix $\widetilde{Z}$ defined by (1.34) having a nullspace of dimension greater than 1.

To get a more robust procedure, the authors of [50] modified the algorithm of the previous section to choose p and q to be of minimal degree. The basic idea is that if the nullspace of $\widetilde{Z}$ has dimension d, then we can always choose $\widehat{b}$ so that its last $d-1$ components—those corresponding to the highest degree terms in q—are zero. Thus, we can take q from $\mathcal{P}_{N-(d-1)}$ instead of $\mathcal{P}_N$. For general interpolation problems, the dimension of the nullspace of $\widetilde{Z}$ will never be more than 1; however $\widetilde{Z}$ may have singular values that are small enough that they can be taken to be effectively zero. Mathematically, this is the same as replacing the interpolation problem with a least-squares problem.

This improved scheme is also implemented in `ratinterp`, which by default uses a threshold of 10^{-14} to decide when a singular value of $\widetilde{Z}$ is small enough to be negligible. Indeed, to recover the behavior of the non-robust algorithm, we had to explicitly disable robustness by setting the threshold to zero when we called `ratinterp` earlier. To see that this technique fixes things for our simple example, we recompute the rational interpolant with these robustness enabled via

```
>> [p, q] = ratinterp(@(x) 1./(x - 1.5), 2, 3);
```

Now, `q` is a linear polynomial with its sole root at the true pole:

```
>> roots(q, 'all')
ans =
   1.499999999999999
```

and evaluation at the previously troublesome point is now accurate to essentially full machine precision:

```
>> p(0.248684045509900)/q(0.248684045509900)
ans =
  -0.799158674842830
```

For further details, see [50] and [133, Ch. 26]. Note that a similar technique to the one just described can be applied to construct a robust algorithm for *Padé approximation*, in which one seeks a rational function of prescribed numerator and denominator degrees whose Taylor series coincides with that of the function being approximated up to maximal order at a specified point. For more information, see [49] and [133, Ch. 27].

1.5 Chebfun

The ideas discussed in the preceding sections form the mathematical foundation for the Chebfun software package for numerical computing with functions [29], which consists of a set of MATLAB classes and subroutines for working with what the Chebfun development team calls "Chebyshev technology." While Chebfun itself is not the focus of this thesis, it has been a stimulus for much of the work herein, and we will make use of it often for numerical illustrations. We close this introductory chapter with a brief description of how it works.

The key idea underlying Chebfun is that high-degree polynomial interpolants can act as proxies for the functions that they interpolate. These polynomials are represented using MATLAB objects called chebfuns. Users construct chebfuns to represent the functions in which they are interested and then operate using these chebfuns as if they were the functions themselves. The software takes care of all of the operations on the underlying polynomial representations, shielding the user from the fact that they are working with polynomials instead of the original functions. The result is a computing experience that feels symbolic but which is actually numeric.

A chebfun is constructed from a given function by interpolating on finer and finer Chebyshev grids until the interpolant approximates the function uniformly to machine precision on the interval of interpolation. The system knows that it has accomplished this when the high-degree coefficients of the polynomial in its Chebyshev basis representation (1.12) are sufficiently small.[29]

For instance, running

```
>> f = chebfun(@(x) x.*tanh(x + 1/5).*sin(10*(x - 1/2)))
f =
   chebfun column (1 smooth piece)
       interval       length     endpoint values
[      -1,       1]       42     -0.43     -0.8
vertical scale = 0.81
```

creates a chebfun object representing the function $f(x) = x\tanh(x + 1/5)\sin\bigl(10(x - 1/2)\bigr)$ on the interval $[-1, 1]$. The "length" of 44 means that an interpolant in 44 Chebyshev points was necessary

[29] The basis for this criterion is the fact that every function $f \in C([-1, 1])$ that is just a little smooth (e.g., satisfies the hypothesis of Theorem 1.8) can be expanded in a uniformly convergent *Chebyshev series* $f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$. If this series is absolutely convergent (which requires a little more than the hypothesis of Theorem 1.8; Lipschitz continuity is sufficient), the coefficients c_k of the Chebyshev interpolants to f in the representation (1.12) can be seen to be sums of certain subsequences of the expansion coefficients a_k. This happens because the Chebyshev polynomials of degree K and higher *alias* to Chebyshev polynomials of a lower degree on the length-K Chebyshev grid. For further details, see [133, Ch. 4].

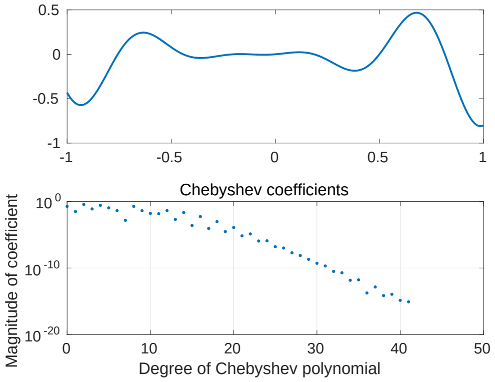


Figure 1.4: Plot of the chebfun representing $f(x) = x\tanh(x-1/5)\sin\big(10(x-1/2)\big)$ on $[-1,1]$ along with its Chebyshev coefficients. Note that the coefficients asymptotically decay at a geometric rate, reflecting the geometric convergence of the Chebyshev interpolants to this holomorphic function predicted by Theorem 1.10.

to resolve this function to machine precision. Figure 1.4 shows a plot of the interpolant together with its Chebyshev coefficients, which decay steadily in magnitude down to about 10^{-16}. Note that this function is holomorphic on $[-1,1]$, so its Chebyshev interpolants converge at a geometric rate by Theorem 1.10. This is reflected in the asymptotic geometric decay rate of the coefficients observed in the plot.

From a practical standpoint, there is essentially no difference between a function f and a polynomial p that approximates f to machine precision. Polynomials, however, are simpler to work with than general functions. It is much easier to extract information from a polynomial than it is from an arbitrary function. For instance, we can ask for the integral of f, which is calculated by integrating the Chebyshev interpolant; this is known as *Clenshaw–Curtis quadrature* [133, Ch. 19]:

```
>> sum(f)
ans =
  -0.092888511319387
```

This operation is called `sum` after the built-in MATLAB operation of the same name that sums the elements of a discrete numeric vector. The idea is that a chebfun behaves as a sort of “continuous” MATLAB vector, and the continuous analogue of summation is integration. Many other operations on chebfuns are derived from operations on discrete vectors using similar reasoning. Since the interpolant represents f to machine precision, this result will be accurate in perhaps all but its last digit or two.

We could also ask for the zeros of f, which can be computed by solving an eigenvalue problem involving what is known as a *colleague matrix*, in which the Chebyshev coefficients appear in the final row [51], [133, Ch. 18]:

```
>> roots(f)
ans =
  -0.756637061435917
  -0.442477796076938
  -0.200000000000000
  -0.128318530717964
   0.000000000000004
   0.185840734641019
   0.499999999999999
   0.814159265358979
```

These, too, will be accurate to essentially full machine precision.

As a last example, we compute the maximum value of $\sin\bigl(f(x)^2\bigr)$:

```
>> max(sin(f.^2))
ans =
   0.608576905765540
```

This runs the adaptive sampling process twice: first to build a chebfun for $f(x)^2$ by evaluating the one for $f(x)$ and then to build one for $\sin\bigl(f(x)^2\bigr)$ by evaluating the one for $f(x)^2$ just constructed. It then computes the roots of the derivative of this chebfun to find all of the local extrema and checks each to see which yields the maximum value.

Chebfun can do much more than we have described here. Further details can be found in the Chebfun users' guide [29].

Chapter 2

Numerical Stability of the Barycentric Formula for Trigonometric Interpolation[1]

In this chapter, we present the first new contribution of this thesis: a study of the numerical stability of the barycentric formulas for trigonometric interpolation presented in Section 1.3.2. We restrict our attention to the case in which the points are equispaced. Additionally, we will be concerned primarily with the formula for interpolation in an odd number of points, though we will make a few remarks about what happens for the even-length formula towards the end.

2.1 Introduction

We begin by recalling our notational setup from Section 1.3.2. Let $K \geq 1$ be an odd integer, and let X be a set of K equispaced points

$$x_k = (k + \alpha)h, \qquad 0 \leq k \leq K - 1, \tag{2.1}$$

in $[0, 2\pi]$, where $h = 2\pi/K$ is the grid spacing and $\alpha \in [0, 1]$ is a parameter that determines the grid shift (i.e., the deviation of x_0 from 0). Let $f_0, \ldots, f_{K-1}$ be arbitrary real numbers, which we take as elements of a vector f. The unique trigonometric polynomial $t_{f,X}$ of degree $N = (K-1)/2$ that interpolates the value f_k at the points x_k for each k can be expressed using either the first barycentric formula (1.27) or the second formula (1.28), which are analogues for trigonometric interpolation of the polynomial barycentric formulas (1.4) and (1.6).

Recall from Section 1.2.2 that the polynomial barycentric formulas enjoy favorable properties with regard to numerical stability: the first formula (1.4) is backward stable, and the second formula (1.6) is forward stable if the Lebesgue constant for the grid is not too large. In particular, the second

[1] The content in this chapter is adapted from the paper [8] by the author and collaborator Kuan Xu. Xu raised the original question of whether the formula (1.28) is stable and worked jointly with the author in investigating the matter numerically. The author proposed the suggested method for stabilizing the formula, worked out most of the theoretical analysis with Xu carefully checking the details, and wrote the text of the paper.

formula is stable when the interpolation points are Chebyshev points in $[-1, 1]$. On the basis that the trigonometric formulas (1.27) and (1.28) are so similar in structure to their polynomial counterparts, it is reasonable to guess that they should possess similar properties.

Unfortunately, this is not the case. Henrici [55] notes that the first formula (1.27) suffers from instability as K grows due to the inability to evaluate the factor $\sin\bigl(K(x-\alpha h)/2\bigr)$ in front of the sum to high relative accuracy for large K. Even for small K, both Henrici [55] and Berrut [12] indicate that this factor causes instability when evaluating (1.27) when x is close to one of the interpolation points x_k. No such problems occur with the polynomial formula (1.4).

For (1.28), the situation appears at first to be better. With the problematic leading factor from (1.27) out of the picture, the only remaining issue to settle is what happens for evaluations near interpolation points, the same concern that was expressed about (1.6). It would seem that the same informal argument about potentially large errors cancelling out in the quotient that is used to explain the stability of (1.6) (see Section 1.2.2) should apply here as well, and Henrici [55] does indeed make this argument. Moreover, both Henrici [55] and Berrut [13] provide numerical examples illustrating the apparent stability of (1.28).

It turns out, however, that this is not quite true. While (1.28) produces good results in the majority of cases, it does, in fact, possess a subtle instability that seems to have been overlooked in the investigations of Henrici [55] and Berrut [12, 13]. We illustrate and explain the origin of this instability in Section 2.2. Fortunately, it is possible to correct the instability via a rewriting of (1.28), as we show in Section 2.3. Combining the original and rewritten formulas, we obtain an algorithm that is forward stable, and we prove this rigorously in Section 2.4 by adapting the analysis of Higham [58] for the polynomial formulas to our setting. Finally, in Sections 2.5 and 2.6 we discuss interpolation on intervals other than $[0, 2\pi]$ and make a few remarks on what happens when K is even instead of odd.

2.2 Instability of the Second Formula

We can demonstrate the instability in (1.28) by a simple numerical example. Take $\alpha = 1$, $K = 3$, and $f_k = \sin(x_k)$ for each k. We evaluate (1.28) with these parameters at several points x whose distances from 0 range from 1 to 10^{-15}. We perform the evaluation twice: once in double precision and once in 256-bit (approximately 75-digit) precision using the arbitrary precision arithmetic features of the Julia programming language [16], which are based on the GNU MPFR library [42]. We take the high precision results as “exact” and use them to measure the relative error in the results obtained in double precision.

The results are displayed in Figure 2.1. The error increases steadily as the evaluation point x moves closer to 0. On the other hand, the product of the relative condition number $\kappa(x, X, f)$ for

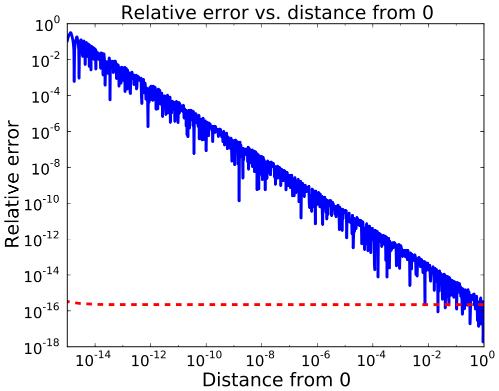


Figure 2.1: Illustration of instability in (1.28) in the case $\alpha = 1$. The solid blue line depicts the relative error in the evaluations for the example described in Section 2.2. The dashed red line shows the product of the condition number of the evaluations (computed using the formula given by Lemma 2.2 below) and the unit roundoff $u = 2^{-52}$. As the distance between the evaluation point and 0 decreases, the relative error rises even though the evaluation remains well-conditioned, indicating numerical instability.

evaluating $t_{f,X}(x)$ (see Section 2.4.1) and the unit roundoff $u = 2^{-52}$ is at the level of u for all evaluation points x considered. We conclude that (1.28) is indeed unstable under these circumstances.[2]

After a little thought, the origin of the instability can be identified. For our choice of α, $x_{K-1} = 2\pi$, so when x is near 0, we evaluate the sine function at a point close to π when computing the terms at $k = K-1$ in the numerator and denominator of (1.28). The sine function is poorly conditioned near π,[3] so the rounding errors incurred when forming $(x - x_{K-1})/2$ get magnified into large relative errors in the computed value of $\sin\bigl((x - x_{K-1})/2\bigr)$.

For many uses of (1.28), these errors do not cause any problems, since they cancel out in the final quotient as described in Section 2.1. The mechanism driving the cancellation in this case is the dominance of the $k = K-1$ terms in the numerator and denominator of (1.28): for α near 1 and x near 0, these terms will typically be much larger than the terms for $k < K-1$, since $\sin\bigl((x - x_{K-1})/2\bigr)$ is nearly 0. Hence, any relative error in the $k = K-1$ terms, even a large one, will divide out neatly when taking the quotient. In our example, however, f_{K-1}, which has a magnitude on the order of

[2] Note that in this example, we measure the *relative* error in the evaluation near a root of the interpolant. This may seem strange initially, as one can usually only expect the error to be small in an *absolute* sense at such points. Nevertheless, the small relative condition number tells us that, in principle, we should be able to evaluate the interpolant with a low relative error here. The reason for this is that while we are evaluating near a root, that root is near an interpolation point, and the value of the interpolant at the interpolation point is known exactly.

[3] By this, we mean that for values of x near π, a change in x that is small in a relative sense can yield changes in $\sin(x)$ that are large in a relative sense. More quantitatively, the *condition number* of a differentiable function f at a point x is defined to be

$$\kappa_f(x) = \left| \lim_{\Delta x \to 0} \frac{\frac{f(x+\Delta x) - f(x)}{f(x)}}{\frac{\Delta x}{x}} \right| = \left| x \frac{f'(x)}{f(x)} \right|.$$

If $\kappa_f(x)$ is large, then small relative changes in x can yield large relative changes in $f(x)$. For $f(x) = \sin(x)$, we have $\kappa_f(x) = |x \cot(x)|$, and this number grows without bound as x approaches π. For further details on these concepts, see [57, §1.6], [94, Ch. 12], and [134, Lecture 12].

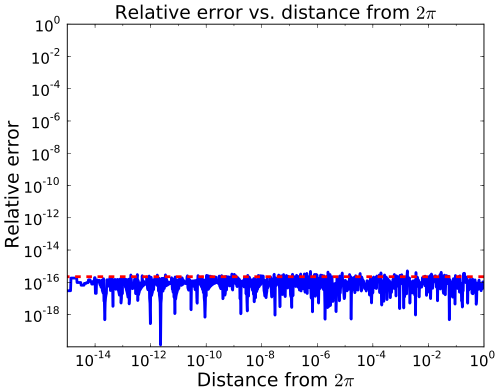


Figure 2.2: Same as Figure 2.1 but with $\alpha = 0$ and with the evaluation points located near 2π instead of 0. The relative error is at the level of machine precision due to the special circumstances enjoyed by this case.

10^{-16}, is much smaller than f_k for $k < K-1$, all of which have magnitudes on the order of 1. This poor scaling of the function values relative to each other offsets the dominance of the $k = K-1$ terms, resulting in imperfect cancellation.

Something interesting occurs if we repeat the experiment but take $\alpha = 0$ instead of $\alpha = 1$. In this case, we anticipate instability when evaluating at points near 2π instead of 0, with the problematic terms occurring at $k = 0$ instead of $k = K-1$. The results are displayed in Figure 2.2. While the plot of the product of the condition number and the unit roundoff is unaltered, surprisingly, the relative error is at the level of machine precision for all evaluation points. The reason this happens is that when $\alpha = 0$, $x_0 = 0$. Thus, $(x - x_0)/2$ is evaluated *exactly* for all $x \in [0, 2\pi]$, since subtraction of 0 and division by 2 incur no errors in standard IEEE floating-point arithmetic. There is therefore no rounding error made whose effect can be amplified by the ill conditioning of the sine function, so the instability cannot be excited in this special but very common case. If α is taken to be only near 0 (say, 10^{-15}) instead of exactly 0, the instability appears as expected.

We speculate that this behavior is one of the reasons the instability described in this section has evaded notice in the literature, as the $\alpha = 0$ grid is perhaps the most frequently employed grid of equispaced points in $[0, 2\pi]$; indeed, [55] works exclusively with this grid. The instability becomes noticeable when considering other grids, such as the $\alpha = 1$ grid and when working with the analogue of the $\alpha = 0$ grid on $[-\pi, \pi]$, for which the first point is $-\pi$, a number which is undistinguished in floating-point arithmetic.

2.3 A Stable Algorithm

The instability just described only arises when the interpolation data are poorly scaled in the sense that either f_0 or f_{K-1} is much smaller than the other f_k when α is near 0 or 1, respectively. If this is not the case, i.e., if $\max_{1 \le k \le K-1} |f_k/f_0|$ (for α near 0) or $\max_{0 \le k \le K-2} |f_k/f_{K-1}|$ (for α near 1) is not too large,[4] then (1.28) will be stable for all evaluation points in $[0, 2\pi]$. Interpolation data that are as wildly poorly-scaled as those of the examples shown in the previous section are relatively uncommon in practice. Even when the data are poorly scaled, most evaluations of the trigonometric interpolant are done in the interior of the interval, where the sine evaluations are well-conditioned, and (1.28) will be stable in this case as well. Thus, (1.28) is stable in most cases of practical interest.

Nevertheless, knowing how to fix the instability is valuable so that it can be done when needed. This can be accomplished by rewriting (1.28) to avoid evaluating the expression $\sin\bigl((x - x_k)/2\bigr)$ near points where it is poorly conditioned. There are two situations in which a bad evaluation can occur: when α is near 0 and the evaluation point x is near 2π and when α is near 1 and x is near 0.

The remedy we propose is to use periodicity to adjust the location of the interpolation point furthest from x in these cases so that the distance between it and x can never get too close to 2π. Consider the case where α is near 0. For x near 2π, the interpolation point furthest from x is x_0, so we modify (1.28) by replacing x_0 by its periodic image $x_0 + 2\pi$ and changing the signs of the $k = 0$ terms in both sums. The resulting formula, which amounts to using (1.28) to compute an interpolant in the points $x_1, \ldots, x_{K-1}, x_0 + 2\pi$ instead of $x_0, x_1, \ldots, x_{K-1}$, is exactly equal to (1.28) mathematically but not in floating-point arithmetic. Similar comments apply to the case where α is near 1 and x is near 0, for which we replace x_{K-1} by $x_{K-1} - 2\pi$ and change the signs of the $k = K - 1$ terms. For explicit formulas, see (2.2) and (2.3), below.

We are not done yet, however, as all we have actually done is rewrite the poorly conditioned terms in (1.28) in a different way. The modified terms are still poorly conditioned, as a problem's conditioning is independent of how it is written down or represented. What has changed is the *source* of the poor conditioning. Instead of through the sine function itself, it now enters via the potential for cancellation error in the computation of the argument to the sine function. The second key idea needed to stabilize (1.28) is the realization that we can avoid these problems by computing the argument in a particular way, as we now describe.

First, we must group the terms of the argument appropriately. Consider the case where α is near 0, so that the argument to the sine function in the modified term is $(x - x_0 - 2\pi)/2$. Ignoring the division by 2, which has no potential for cancellation, if we evaluate the rest in floating-point arithmetic from left to right as $(x - x_0) - \mathrm{fl}(2\pi)$, where $\mathrm{fl}(2\pi)$ is the nearest floating-point number to 2π (see Section 2.4.2), then the second subtraction will involve two nearby quantities whenever x is near 2π and x_0 is near 0. Even if the second subtraction is performed without rounding error, accuracy will be lost if the magnitude of the rounding error made in the first subtraction is significant compared to the magnitude of the final result.

[4] As a rule of thumb, one can expect to lose roughly one digit of accuracy in evaluations near the "bad" endpoint for each order of magnitude in these quantities. For instance, if α is near 0 and $\max_{1 \le k \le K-1} |f_k/f_0|$ is on the order of 10^8, then a loss of about 8 digits in evaluations near 2π would be typical.

We can fix this by grouping the terms as $\big(x - \mathrm{fl}(2\pi)\big) - x_0$ instead. While the subtraction $x - \mathrm{fl}(2\pi)$ still incurs cancellation, it is of a benign sort, as neither x nor $\mathrm{fl}(2\pi)$ has been contaminated by rounding errors from previous computations (but see the next paragraph). Moreover, since $x \le \mathrm{fl}(2\pi)$ and $x_0 \ge 0$, the second subtraction involves two quantities of opposite sign, and hence no further cancellation can occur. The final result will therefore be a high relative accuracy approximation to the exact value (i.e., computed without rounding error) of $x - x_0 - \mathrm{fl}(2\pi)$.

This is almost what we want but not quite: we really want a high relative accuracy approximation to $x - x_0 - 2\pi$. Evaluating $\big(x - \mathrm{fl}(2\pi)\big) - x_0$ in floating-point arithmetic will not generally deliver this because of the rounding error in the approximation $\mathrm{fl}(2\pi) \approx 2\pi$. While the cancellation in $x - \mathrm{fl}(2\pi)$ is benign when this subtraction is viewed simply as a difference between two floating-point numbers, it is catastrophic from the perspective of computing an approximation to $x - 2\pi$.

The fix for this is to subtract off an additional correction term to compensate for the error in the approximation $\mathrm{fl}(2\pi) \approx 2\pi$. More precisely, let c be the nearest floating-point number to $2\pi - \mathrm{fl}(2\pi)$. Then, in exact arithmetic, $\mathrm{fl}(2\pi) + c$ is an approximation to 2π with relative error on the order of the square of the unit roundoff (see Section 2.4.2). We cannot form $\mathrm{fl}(2\pi) + c$ directly in floating-point arithmetic because c is insignificant compared to $\mathrm{fl}(2\pi)$ and would be rounded off; however, if adding or subtracting $\mathrm{fl}(2\pi)$ to or from something results in a quantity small enough that c is significant in comparison, we can expect to obtain a higher accuracy result if we subsequently add or subtract c as appropriate.

In this discussion, we have considered only the case where α is close to 0 for definiteness; similar remarks apply to the modified version of (1.28) for α close to 1. Rigorous justification for all of these statements will be given in the analysis of Section 2.4. The value c can be easily computed using any software package that supports arbitrary precision arithmetic or even by hand with aid of a table that lists the value of π to many places. For IEEE floating-point arithmetic, $c = 2.4492935982947064 \times 10^{-16}$ in double-precision, and $c = -1.7484555 \times 10^{-7}$ in single-precision.

The only remaining matter is to decide precisely when to use the modified formulas instead of (1.28), i.e., to give a criterion for determining when x is "too close" to 0 or 2π. For reasons that we will justify in Section 2.4, we switch to the modified formulas whenever x is within $\pi|1 - 2\alpha|/K$ of the relevant endpoint. Note that for $\alpha = 1/2$, this quantity is zero, so (1.28) is used without modification for all $x \in [0, 2\pi]$.

To summarize, the exact procedure we propose is the following:

- If $\alpha \in [0, 1/2)$, use (1.28) for $x \in [0, 2\pi - \pi(1 - 2\alpha)/K]$. For other values of x, use

$$t_{f,X}(x) = \frac{\displaystyle\sum_{k=1}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)} f_k - \frac{1}{\sin\left(\frac{x - x_0 - 2\pi}{2}\right)} f_0}{\displaystyle\sum_{k=1}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)} - \frac{1}{\sin\left(\frac{x - x_0 - 2\pi}{2}\right)}}, \tag{2.2}$$

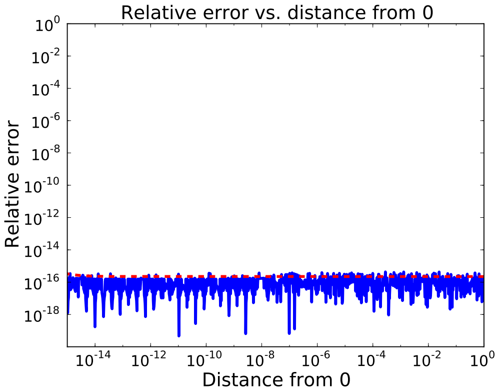


Figure 2.3: Same as Figure 2.1 but using the formula (2.3) (with $x - x_0 - 2\pi$ computed as prescribed in Section 2.3) instead of (1.28) to do the evaluations. All errors are now at the level of machine precision.

with $x - x_0 - 2\pi$ computed as $\big((x - \mathrm{fl}(2\pi)) - c\big) - x_0$.

- If $\alpha = 1/2$, use (1.28) for all $x \in [0, 2\pi]$.
- If $\alpha \in (1/2, 1]$, use (1.28) for $x \in [\pi(2\alpha - 1)/K, 2\pi]$. For other values of x, use

$$t_{f,X}(x) = \frac{\displaystyle\sum_{k=0}^{K-2} \frac{(-1)^k}{\sin\left(\frac{x-x_k}{2}\right)} f_k - \frac{1}{\sin\left(\frac{x-x_{K-1}+2\pi}{2}\right)} f_{K-1}}{\displaystyle\sum_{k=0}^{K-2} \frac{(-1)^k}{\sin\left(\frac{x-x_k}{2}\right)} - \frac{1}{\sin\left(\frac{x-x_{K-1}+2\pi}{2}\right)}}, \tag{2.3}$$

with $x - x_{K-1} + 2\pi$ computed as $x - \big((x_{K-1} - \mathrm{fl}(2\pi)) - c\big)$.

This scheme has the disadvantage that an implementation must make decisions based on the location of the evaluation point. It is therefore less computationally efficient than (1.28); however, this is the price that must be paid to guarantee stability for all evaluation points x and all possible interpolation data f_k.

To verify that the scheme we have described works, we repeat our experiment from Section 2.2 with $\alpha = 1$ using this algorithm. All of the evaluation points x considered lie in $[0, \pi(2\alpha - 1)/K)$, so we use (2.3) for all of them. The relative error, depicted in Figure 2.3, is now at the level of machine precision, even for evaluation points that are very close to 0.

2.4 Analysis of the Proposed Algorithm

We complete our investigation by putting the observed stability of the algorithm given in the previous section on a rigorous basis with a formal proof. The notation and framework we use for our analysis are borrowed directly from Higham's paper [58]. To keep our discussion self-contained, we repeat the relevant definitions here.

2.4.1 Condition Number

Our bounds will be stated in terms of the condition number given in following definition. We denote by $|f|$ the vector whose components are the absolute values of the corresponding components of the vector f. Inequalities between vectors are understood to hold componentwise.

Definition 2.1. *For $t_{f,X}(x) \neq 0$, the* relative condition number *of $t_{f,X}$ at x with respect to perturbations in f is*

$$\kappa(x, X, f) = \lim_{\varepsilon \to 0} \sup \left\{ \left| \frac{t_{f,X}(x) - t_{f+\Delta f,X}(x)}{\varepsilon t_{f,X}(x)} \right| \ : \ |\Delta f| \leq \varepsilon |f| \right\}.$$

A trivial rearranging of (1.27) yields the following Lagrange form for $t_{f,X}(x)$ (cf. (1.25) and (1.26)):

$$t_{f,X}(x) = \sum_{k=0}^{K-1} \ell_k(x) f_k, \qquad \ell_k(x) = \frac{(-1)^k}{K} \frac{\sin\left(\frac{K(x-\alpha h)}{2}\right)}{\sin\left(\frac{x-x_k}{2}\right)}.$$

The following lemma gives an explicit expression for $\kappa(x, X, f)$ and shows how it can be used to bound the relative difference between $t_{f,X}(x)$ and $t_{f+\Delta f,X}(x)$ for a given perturbation Δf. It directly parallels Lemma 2.2 of [58] and can be proved in exactly the same way.

Lemma 2.2. *We have*

$$\kappa(x, X, f) = \frac{\sum_{k=0}^{K-1} |\ell_k(x) f_k|}{|t_{f,X}(x)|} \geq 1,$$

and for any vector Δf with $|\Delta f| \leq \varepsilon |f|$,

$$\left| \frac{t_{f,X}(x) - t_{f+\Delta f,X}(x)}{t_{f,X}(x)} \right| \leq \varepsilon \kappa(x, X, f).$$

2.4.2 Floating-Point Model

We denote floating-point approximations to quantities by $\mathrm{fl}(\cdot)$. The standard model of floating-point arithmetic [57, Ch. 2] posits that whenever x and y are floating-point numbers and $\circledast$ is one of the four basic arithmetic operations $+$, $-$, $\times$, or $\div$, we have

$$\mathrm{fl}(x \circledast y) = (x \circledast y)(1+\delta)^{\pm 1}, \qquad |\delta| \leq u, \tag{2.4}$$

where u is the unit roundoff. We use this model with one modification: we assume additionally that whenever x is a floating-point number

$$\mathrm{fl}\big(\sin(x)\big) = \sin(x)(1+\delta)^{\pm 1}, \qquad |\delta| \leq u. \tag{2.5}$$

This assumption is not guaranteed to hold by any floating-point standard; however, it is possible to accomplish this and similarly for the other common transcendental functions with high-quality implementations [83]. Moreover, the latest revision of the IEEE floating-point standard recommends (but does not mandate) that languages supporting floating-point operations also provide correctly rounded implementations for all such basic functions[5] [60]. This suggests that our additional assumption is, at the very least, reasonable. In fact, we will not require its full force: for our purposes, it is sufficient for it to hold when $x \in [-\pi, \pi]$.

The symbol $\langle n \rangle$ denotes the accumulation of n relative errors accrued during a floating-point computation:

$$\langle n \rangle = \prod_{i=1}^{n} (1 + \delta_i)^{\rho_i}, \qquad \rho_i = \pm 1, \qquad |\delta_i| \leq u.$$

When necessary, we write $\langle n \rangle_k$ to indicate that the relative errors depend on an index k.

Throughout our analysis, we will at times need to assume that $nu \leq 1$, where n is a small positive integer. These assumptions will hold for any floating-point system that is used in practice.

2.4.3 Technical Lemmas

Before we proceed, we first pause to establish a pair of minor technical results that we will need in our analysis. Recall the formula for the points x_k given in (2.1).

Lemma 2.3. *If $\alpha \in [0, 1/2]$, then for $1 \leq k \leq K - 1$ and all $x \in [0, 2\pi]$,*

$$\left| \frac{x - x_k}{2} \cot\left(\frac{x - x_k}{2}\right) \right| \leq 2K - 1.$$

If $k = 0$, then the same holds for $x \in [0, 2\pi - \pi(1 - 2\alpha)/K]$.

Proof. We use the following inequality, valid for $t \in [-\pi, \pi]$, whose proof we omit:

$$|\cot(t)| \leq \max\left(\frac{1}{|t|}, \frac{1}{\pi - |t|}\right). \tag{2.6}$$

Thus, we have

$$\left| \frac{x - x_k}{2} \cot\left(\frac{x - x_k}{2}\right) \right| \leq \max\left(1, \frac{|x - x_k|}{2\pi - |x - x_k|}\right),$$

for $0 \leq k \leq K - 1$, since $(x - x_k)/2 \in [-\pi, \pi]$. The second argument to max is maximized when $|x - x_k|$ is as close to 2π as possible. If $1 \leq k \leq K - 1$, then since $\alpha \in [0, 1/2]$, this happens when $x = 0$ and $k = K - 1$, giving

$$\frac{|x - x_k|}{2\pi - |x - x_k|} \leq \frac{(K - 1 + \alpha)\frac{2\pi}{K}}{2\pi - (K - 1 + \alpha)\frac{2\pi}{K}} = \frac{K}{1 - \alpha} - 1 \leq 2K - 1.$$

On the other hand, if $k = 0$ and x is restricted to $[0, 2\pi - \pi(1 - 2\alpha)/K]$, then $|x - x_0|$ is closest to 2π when x is at the right endpoint of that interval, so

$$\frac{|x - x_0|}{2\pi - |x - x_0|} \leq \frac{2\pi - \pi\frac{1-2\alpha}{K} - \alpha\frac{2\pi}{K}}{2\pi - \left(2\pi - \pi\frac{1-2\alpha}{K} - \alpha\frac{2\pi}{K}\right)} = 2K - 1$$

as well. As $2K - 1 \geq 1$, this completes the proof. □

[5] For instance, the implementation for sine contributed by IBM to glibc (v. 2.21 at the time of this writing) claims to do this.

Lemma 2.4. *If $\alpha \in [0, 1/2]$, then for $x \in (2\pi - \pi(1-2\alpha)/K, 2\pi]$,*

$$\left| \frac{x - x_0 - 2\pi}{2} \cot\left(\frac{x - x_0 - 2\pi}{2}\right) \right| \leq 1.$$

Proof. Since $|x - x_0| \leq 2\pi$, we have $|x - x_0 - 2\pi| = 2\pi - (x - x_0)$, and for x in the given interval, we have $x \geq x_0$, so $|x - x_0| = x - x_0$. Thus, by (2.6),

$$\begin{aligned} \left| \frac{x - x_0 - 2\pi}{2} \cot\left(\frac{x - x_0 - 2\pi}{2}\right) \right| &= \left| \frac{x - x_0 - 2\pi}{2} \cot\left(\frac{x - x_0}{2}\right) \right| \\ &\leq \max\left(\frac{|x - x_0 - 2\pi|}{|x - x_0|}, \frac{|x - x_0 - 2\pi|}{2\pi - |x - x_0|} \right) \\ &= \max\left(\frac{2\pi - (x - x_0)}{x - x_0}, 1 \right). \end{aligned}$$

The first argument to max in the final line is maximized when $x - x_0$ is as small as possible. Given the restrictions on x, this happens when $x = 2\pi - \pi(1-2\alpha)/K$, so we have

$$\frac{2\pi - (x - x_0)}{x - x_0} \leq \frac{2\pi - 2\pi + \pi\frac{1-2\alpha}{K} + \alpha\frac{2\pi}{K}}{2\pi - \pi\frac{1-2\alpha}{K} - \alpha\frac{2\pi}{K}} = \frac{1}{2K - 1}.$$

The result follows, since $1/(2K - 1) \leq 1$. □

2.4.4 Stability Analysis

We are now ready to carry out our analysis. For the remainder of this section, x is taken to be a fixed value in $[0, 2\pi]$, and we assume that x, x_k, and f_k are all floating-point numbers.[6] We ignore all issues of overflow and underflow. Our goal is to obtain a bound on the relative error $|t_{f,X}(x) - \widehat{t}_{f,X}(x)|/|t_{f,X}(x)|$, where $\widehat{t}_{f,X}(x)$ is the approximation to $t_{f,X}(x)$ obtained by evaluating (1.28), (2.2), or (2.3) in floating-point arithmetic as prescribed in Section 2.3. Specifically, we will prove the following theorem:

Theorem 2.5. *In the absence of overflow and underflow, the relative error in evaluating $t_{f,X}(x)$ in floating-point arithmetic using the algorithm of Section 2.3 satisfies*

$$\left| \frac{t_{f,X}(x) - \widehat{t}_{f,X}(x)}{t_{f,X}(x)} \right| \leq (5K + 7)u\kappa(x, X, f) + (5K + 6)\left(\frac{2}{\pi}\log(K) + 2\right)u + O(u^2) \tag{2.7}$$

for all $\alpha \in [0, 1]$.

Thus, the procedure outlined in Section 2.3 gives a forward stable method for evaluating trigonometric interpolants in equispaced points.

[6] Of course, it is not possible that the x_k are simultaneously exactly equispaced in $[0, 2\pi]$ and also floating-point numbers. With approximately equispaced x_k, the formulas (1.28), (2.2), and (2.3) only approximate the trigonometric interpolant instead of computing it exactly. This does not matter for our investigation, however, as we are only concerned with the numerical stability of these formulas. Mascarenhas and de Camargo [80] have given an analysis of the effects of rounding errors in the interpolation points in the polynomial case.

Proof. We will establish the bound for $\alpha \in [0, 1/2]$; the argument for $\alpha \in (1/2, 1]$ is similar. Our argument is identical in structure to the one given by [58] for the polynomial case.

First, we develop an expression for $\widehat{t}_{f,X}(x)$ in the case where (1.28) is used for the evaluation. By (2.4), we have, for some $\delta_{k,1}$ and $\delta_{k,2}$ with $|\delta_{k,1}| \leq u$ and $|\delta_{k,2}| \leq u$,[7]

$$\mathrm{fl}\left(\frac{x - x_k}{2}\right) = \frac{x - x_k}{2}(1 + \delta_{k,1})(1 + \delta_{k,2}). \tag{2.8}$$

Hence, by (2.5) and the fact that $\sin\bigl(x(1 + \varepsilon)\bigr) = \sin(x)\bigl(1 + \varepsilon x \cot(x) + O(\varepsilon^2)\bigr)$ for small ε, we have

$$\mathrm{fl}\left(\sin\left(\frac{x - x_k}{2}\right)\right) = \sin\left(\frac{x - x_k}{2}(1 + \delta_{k,1})(1 + \delta_{k,2})\right) \langle 1 \rangle_k = \sin\left(\frac{x - x_k}{2}\right) \left(1 + \eta_k + O(u^2)\right) \langle 1 \rangle_k,$$

where

$$\eta_k = (\delta_{k,1} + \delta_{k,2})\frac{x - x_k}{2} \cot\left(\frac{x - x_k}{2}\right).$$

Therefore, our floating-point approximation to the numerator of (1.28) is given by

$$\begin{aligned}\mathrm{fl}\left(\sum_{k=0}^{K-1} \frac{(-1)^k f_k}{\sin\left(\frac{x - x_k}{2}\right)}\right) &= \sum_{k=0}^{K-1} \frac{(-1)^k f_k}{\sin\left(\frac{x - x_k}{2}\right)} \frac{\langle 2 \rangle_k \langle K - 1 \rangle_k}{1 + \eta_k + O(u^2)} \\ &= \sum_{k=0}^{K-1} \frac{(-1)^k f_k}{\sin\left(\frac{x - x_k}{2}\right)} \langle K + 1 \rangle_k \bigl(1 - \eta_k + O(u^2)\bigr),\end{aligned}$$

where we have picked up one rounding error from the division in each term and $K - 1$ rounding errors from the $K - 1$ additions in the sum[8] and have used the expansion $1/(1 + \varepsilon) = 1 - \varepsilon + O(\varepsilon^2)$. The denominator of (1.28) may be handled similarly. Adding one more rounding error to account for the final division, we arrive at

$$\widehat{t}_{f,X}(x) = \frac{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k f_k}{\sin\left(\frac{x - x_k}{2}\right)} \langle K + 2 \rangle_k \bigl(1 - \eta_k + O(u^2)\bigr)}{\displaystyle\sum_{k=0}^{K-1} \frac{(-1)^k}{\sin\left(\frac{x - x_k}{2}\right)} \langle K + 1 \rangle_k \bigl(1 - \eta_k + O(u^2)\bigr)}.$$

This expression is similar in form to the corresponding one obtained in [58] in the polynomial case, the key difference being the presence of the $1 - \eta_k + O(u^2)$ factors, which represent the error due to the conditioning of the sine evaluations.

[7] If one assumes the use of a binary floating-point system (like IEEE floating-point arithmetic), the division by 2 will be performed exactly, so one can take $\delta_{k,2} = 0$. Doing this will yield a bound that is tighter than the one we establish but only very slightly so.

[8] The order in which the terms are summed does not matter here; see [57, Ch. 4].

Next, just as in the proof of Theorem 4.1 of [58], we have

$$
\begin{aligned}
\left|\frac{t_{f,X}(x)-\widehat{t}_{f,X}(x)}{t_{f,X}(x)}\right| &\leq \left(K+2+2\max_{0\leq k\leq K-1}\left|\frac{x-x_k}{2}\cot\left(\frac{x-x_k}{2}\right)\right|\right)u\frac{\sum_{k=0}^{K-1}\left|\frac{f_k}{\sin\left(\frac{x-x_k}{2}\right)}\right|}{\left|\sum_{k=0}^{K-1}\frac{(-1)^k f_k}{\sin\left(\frac{x-x_k}{2}\right)}\right|} \\
&\quad+\left(K+1+2\max_{0\leq k\leq K-1}\left|\frac{x-x_k}{2}\cot\left(\frac{x-x_k}{2}\right)\right|\right)u\frac{\sum_{k=0}^{K-1}\left|\frac{1}{\sin\left(\frac{x-x_k}{2}\right)}\right|}{\left|\sum_{k=0}^{K-1}\frac{(-1)^k}{\sin\left(\frac{x-x_k}{2}\right)}\right|}+O(u^2) \\
&=\left(K+2+2\max_{0\leq k\leq K-1}\left|\frac{x-x_k}{2}\cot\left(\frac{x-x_k}{2}\right)\right|\right)u\kappa(x,X,f) \\
&\quad+\left(K+1+2\max_{0\leq k\leq K-1}\left|\frac{x-x_k}{2}\cot\left(\frac{x-x_k}{2}\right)\right|\right)u\kappa(x,X,1)+O(u^2),
\end{aligned}
\tag{2.9}
$$

where the second step follows from Lemma 2.2. (Here, the 1 in $\kappa(x,X,1)$ refers to a vector of interpolation data whose entries are all 1.)

The only potential problem with this bound is in the terms involving the cotangent function, which can be large if $|x-x_k|/2$ is close to π for some k, reflecting the poor conditioning of the sine function near $\pm\pi$. Most dramatically, if $\alpha=0$, then $x_0=0$, and $\cot\bigl((x-x_0)/2\bigr)$ becomes unbounded as x gets close to 2π. Additionally, in such cases, the error term represented by the $O(u^2)$ symbol may not be negligible, since the implied constant contains terms with $\cot\bigl((x-x_k)/2\bigr)$ as a factor for each k.

These remarks do not apply to the algorithm of Section 2.3, however, because its rules prevent (1.28) from being used in these problematic cases. Since we are assuming $\alpha\in[0,1/2]$, it will only be used if $x\in[0,2\pi-\pi(1-2\alpha)/K]$. The reason for this particular choice of restriction is given by Lemma 2.3, which gives a very simple bound for the cotangent terms in (2.9). Applying this result to (2.9), for $x\in[0,2\pi-\pi(1-2\alpha)/K]$, we obtain

$$
\left|\frac{t_{f,X}(x)-\widehat{t}_{f,X}(x)}{t_{f,X}(x)}\right|\leq 5Ku\kappa(x,X,f)+(5K-1)u\kappa(x,X,1)+O(u^2). \tag{2.10}
$$

On the other hand, if $x\in(2\pi-\pi(1-2\alpha)/K,2\pi]$, we use (2.2) instead of (1.28). We handle the argument to the sine function in the modified terms as follows. Write $\mathrm{fl}(2\pi)=2\pi(1+\delta_{2\pi})$, where $|\delta_{2\pi}|\leq u$. Then,

$$
\mathrm{fl}\bigl(x-\mathrm{fl}(2\pi)\bigr)=\bigl(x-2\pi(1+\delta_{2\pi})\bigr)(1+\delta_{0,1})=\bigl(x-2\pi\bigr)\left(1-\frac{2\pi}{x-2\pi}\delta_{2\pi}\right)(1+\delta_{0,1}),
$$

where $|\delta_{0,1}|\leq u$. Next, we subtract the correction term c to adjust for the error in the approximation $\mathrm{fl}(2\pi)\approx 2\pi$ as explained in Section 2.3. As c is by definition the nearest floating-point number to

$2\pi - \mathrm{fl}(2\pi) = -2\pi\delta_{2\pi}$, we have $c = -2\pi\delta_{2\pi}(1+\delta_c)$ with $|\delta_c| \leq u$. Therefore,

$$\begin{aligned}\mathrm{fl}\big(\big(x - \mathrm{fl}(2\pi)\big) - c\big) &= \big(\mathrm{fl}\left(x - \mathrm{fl}(2\pi)\right) - c\big)(1+\delta_{0,2}) \\ &= \left((x-2\pi)\left(1 - \frac{2\pi}{x-2\pi}\delta_{2\pi}\right)(1+\delta_{0,1}) + 2\pi\delta_{2\pi} + 2\pi\delta_{2\pi}\delta_c\right)(1+\delta_{0,2}) \\ &= (x-2\pi)\left(1 + \frac{2\pi\delta_{2\pi}(\delta_c - \delta_{0,1})}{x-2\pi} + \delta_{0,1}\right)(1+\delta_{0,2}),\end{aligned}$$

where $|\delta_{0,2}| \leq u$. Since x is a floating-point number, and since $\mathrm{fl}(2\pi)$ is the nearest floating-point number to 2π, we have $|x - 2\pi| \geq |\,\mathrm{fl}(2\pi) - 2\pi| = 2\pi|\delta_{2\pi}|$. Thus,

$$\left|\frac{2\pi\delta_{2\pi}(\delta_c - \delta_{0,1})}{x-2\pi}\right| \leq |\delta_c| + |\delta_{0,1}| \leq 2u,$$

and so we may write $\mathrm{fl}\big(\big(x - \mathrm{fl}(2\pi)\big) - c\big) = (x-2\pi)(1+\widehat{\xi}_{0,1})(1+\delta_{0,2})$, where $|\widehat{\xi}_{0,1}| \leq 3u$. Multiplying out the error terms and making the reasonable assumption that $|\widehat{\xi}_{0,1}\delta_{0,2}| \leq u$, which will hold if $3u \leq 1$, we can simplify this to $\mathrm{fl}\big(\big(x - \mathrm{fl}(2\pi)\big) - c\big) = (x-2\pi)(1+\widetilde{\xi}_{0,1})$, where $|\widetilde{\xi}_{0,1}| \leq 5u$.

These developments allow us to write, in analogy to (2.8),

$$\begin{aligned}\mathrm{fl}\left(\frac{\big(\big(x - \mathrm{fl}(2\pi)\big) - c\big) - x_0}{2}\right) &= \frac{\big((x-2\pi)(1+\widetilde{\xi}_{0,1}) - x_0\big)(1+\delta_{0,3})}{2}(1+\delta_{0,4}) \\ &= \frac{x - x_0 - 2\pi}{2}(1+\xi_{0,1})(1+\delta_{0,3})(1+\delta_{0,4}), \qquad (2.11)\end{aligned}$$

where $\widetilde{\xi}_{0,1}$ is as above, $|\delta_{0,3}|$ and $|\delta_{0,4}|$ are both at most u, and $\xi_{0,1} = \widetilde{\xi}_{0,1}\big((x-2\pi)/(x-x_0-2\pi)\big)$. Since $x - 2\pi$ and $-x_0$ have the same sign, $|(x-2\pi)/(x-2\pi-x_0)| \leq 1$, and so $|\xi_{0,1}| \leq |\widetilde{\xi}_{0,1}| \leq 5u$. Note that this is a consequence of our having grouped the terms as prescribed in Section 2.3. From here, we work similarly to before and arrive at

$$\left|\frac{t_{f,X}(x) - \widehat{t}_{f,X}(x)}{t_{f,X}(x)}\right| \leq (K+2+C)u\kappa(x,X,f) + (K+1+C)u\kappa(x,X,1) + O(u^2),$$

where

$$C = \max\left(2\left(\max_{1\leq k\leq K-1}\left|\frac{x-x_k}{2}\cot\left(\frac{x-x_k}{2}\right)\right|\right), 7\left|\frac{x-x_0-2\pi}{2}\cot\left(\frac{x-x_0-2\pi}{2}\right)\right|\right).$$

The factor of 7 in the second argument to the outer instance of max comes from the fact there are three rounding error terms in (2.11) that add up to at most $7u$ compared with the two in (2.8) that add up to at most $2u$. To bound C, we use Lemma 2.4. Combining this with Lemma 2.3, we have

$$C \leq \max(4K-2, 7) \leq 4K+5,$$

and so

$$\left|\frac{t_{f,X}(x) - \widehat{t}_{f,X}(x)}{t_{f,X}(x)}\right| \leq (5K+7)u\kappa(x,X,f) + (5K+6)u\kappa(x,X,1) + O(u^2) \qquad (2.12)$$

for $x \in (2\pi - \pi(1-2\alpha)/K, 2\pi]$. In fact, noting that (2.12) is slightly weaker than (2.10), we see that (2.12) actually holds for $x \in [0, 2\pi]$.

To finish, we note, again following [58], that $\kappa(x, X, 1)$ is bounded above by the Lebesgue constant for the interpolation problem, which is at most $(2/\pi)\log(K) + 2$ by Theorem 1.17. Combining this with (2.12) yields (2.7). This completes the proof of Theorem 2.5. □

Note carefully that this analysis depends strongly on the evaluation point x and the interpolation points x_k being in $[0, 2\pi]$. In particular, for the $\alpha < 1/2$ case that we described in detail, it does *not* apply to evaluations at $x = \mathrm{fl}(2\pi)$ if $\mathrm{fl}(2\pi) > 2\pi$.[9] The reason is that, under these circumstances, $x - 2\pi$ and x_0 may have the same sign, and so the factor multiplying $\widetilde{\xi}_{0,1}$ to define $\xi_{0,1}$ in (2.11) can be large if x_0 is chosen carefully. In IEEE double-precision arithmetic, $\mathrm{fl}(2\pi) < 2\pi$, so this is not a problem; however, in IEEE single-precision arithmetic, $\mathrm{fl}(2\pi) > 2\pi$, and it is not difficult to construct a numerical example in which the algorithm of Section 2.3 is unstable for $x = \mathrm{fl}(2\pi)$.

If $\mathrm{fl}(2\pi) > 2\pi$ and a stable evaluation at $x = \mathrm{fl}(2\pi)$ is desired, it can be accomplished with the aid of a second correction term. To see this, note first that since $x = \mathrm{fl}(2\pi)$, when we compute $x - x_0 - 2\pi$ as prescribed in Section 2.3, the subtraction $x - \mathrm{fl}(2\pi)$ evaluates exactly to zero. Thus, we are left to evaluate

$$\mathrm{fl}(-c - x_0) = (2\pi\delta_{2\pi} + 2\pi\delta_{2\pi}\delta_c - x_0)(1 + \delta_1) = (x - x_0 - 2\pi)\left(1 + \frac{2\pi\delta_{2\pi}\delta_c}{x - x_0 - 2\pi}\right)(1 + \delta_1), \quad (2.13)$$

where $|\delta_1| \leq u$ and we have used the fact that $x = \mathrm{fl}(2\pi) = 2\pi(1 + \delta_{2\pi})$. Since c is by definition the nearest floating-point number to $-2\pi\delta_{2\pi}$, and since $-x_0$ is a floating-point number, we have $|x - x_0 - 2\pi| = |-x_0 - (-2\pi\delta_{2\pi})| \geq |c - (-2\pi\delta_{2\pi})| = 2\pi|\delta_{2\pi}\delta_c|$. Thus, $|2\pi\delta_{2\pi}\delta_c/(x - x_0 - 2\pi)|$ could be as large as 1, and if this is the case, we will not have computed $x - x_0 - 2\pi$ accurately.

This calculation highlights that the problem is due to the fact that $|x - x_0 - 2\pi|$ can be as small as $2\pi|\delta_{2\pi}\delta_c|$, which is $O(u^2)$, while we have only corrected for the error in $\mathrm{fl}(2\pi) \approx 2\pi$ down to $O(u)$. This naturally suggests a fix of subtracting an additional term that corrects the error down to $O(u^2)$.

To this end, let c_2 be the nearest floating-point number to $2\pi\delta_{2\pi}\delta_c$. We have $c_2 = 2\pi\delta_{2\pi}\delta_c(1+\delta_{c_2})$, where $|\delta_{c_2}| \leq u$. In IEEE double precision, $c_2 = -5.989539619436679 \times 10^{-33}$, and in single precision, $c_2 = -6.860498 \times 10^{-15}$. Subtracting c_2 from the result of (2.13), we obtain

$$\begin{aligned}\mathrm{fl}\big((-c - x_0) - c_2\big) &= \left((x - x_0 - 2\pi)\left(1 + \frac{2\pi\delta_{2\pi}\delta_c}{x - x_0 - 2\pi}\right)(1 + \delta_1) - 2\pi\delta_{2\pi}\delta_c - 2\pi\delta_{2\pi}\delta_c\delta_{c_2}\right)(1 + \delta_2) \\ &= (x - x_0 - 2\pi)\left(1 + \delta_1 + 2\pi\delta_{2\pi}\delta_c\frac{\delta_1 - \delta_{c_2}}{x - x_0 - 2\pi}\right)(1 + \delta_2),\end{aligned}$$

where $|\delta_2| \leq u$. Using the lower bound on $|x - x_0 - 2\pi|$ just derived and collecting the error terms, we find that $\mathrm{fl}\big((-c - x_0) - c_2\big) = (x - x_0 - 2\pi)(1 + \xi)$ with $|\xi| \leq 5u$ (assuming that $3u \leq 1$). This is certainly accurate enough for our purposes.

Similar remarks apply in the $\alpha > 1/2$ case if x_{K-1} is taken to be $\mathrm{fl}(2\pi)$.

[9] Note that $\mathrm{fl}(2\pi)$ is the only floating-point number x in $[0, \mathrm{fl}(2\pi)]$ for which one can have $x > 2\pi$, for if $2\pi < x < \mathrm{fl}(2\pi)$, then x would be a closer floating-point number to 2π than $\mathrm{fl}(2\pi)$.

2.5 Interpolation on Intervals Other than $[0, 2\pi]$

With the main result of this chapter now established, in the remaining two sections, we examine what happens to our discussions in some settings beyond the one we have considered up to this point.

Thus far, we have confined our discussion to interpolation on the interval $[0, 2\pi]$. At first glance, it would seem that much of what we have said translates directly to other intervals with little additional work, since (1.28) holds for x and x_k drawn from *any* given interval of length 2π, a consequence of its depending only on the values of $x - x_k$ for each k and not on the values of x and x_k individually. In fact, the issue is more subtle, as we now explain.

The instability in (1.28) that we presented in Section 2.2 arises when poor conditioning of the sine function amplifies rounding errors in the computation of $(x - x_k)/2$ into large relative errors in the computed value of $\sin\bigl((x - x_k)/2\bigr)$ for some k. If it happens that $(x - x_k)/2$ is computed exactly for all k for which the corresponding evaluation of the sine function is ill-conditioned, this cannot occur, and (1.28) will perform the evaluation stably. We observed this behavior empirically in the numerical experiments of Section 2.2 involving the $\alpha = 0$ grid on $[0, 2\pi]$. In terms of the analysis of Section 2.4, this corresponds to having $\delta_{k,1} = \delta_{k,2} = 0$ in (2.8) for the relevant values of k so that the bound (2.10) for the relative error in using (1.28) to evaluate $t_{f,X}(x)$ in floating-point arithmetic holds for all $x \in [0, 2\pi]$ instead of just for x in the restricted interval given there.

In IEEE floating-point arithmetic, which uses a binary floating-point system, multiplication and division by 2 are always exact, barring overflow and underflow. Thus, whether $(x - x_k)/2$ is computed exactly boils down to whether the subtraction $x - x_k$ is done exactly. When working on intervals other than $[0, 2\pi]$, especially those away from the origin, this can happen with a far greater frequency than one might initially expect, owing to the following theorem of Sterbenz [57, Ch. 2]:

Theorem 2.6 (Sterbenz's theorem)**.** *If s and t are floating-point numbers such that $t/2 \leq s \leq 2t$, then $\mathrm{fl}(s - t) = s - t$ in the absence of underflow.*

Note that the hypotheses of the theorem imply that s and t are both nonnegative; an analogous result can be stated when s and t are both negative. It is easy to check that for $a \geq 2\pi$, the condition $t/2 \leq s \leq 2t$ is satisfied for all $s, t \in [a, a + 2\pi]$. Hence, all of the subtractions $x - x_k$ that occur when using (1.28) to interpolate on such an interval will be done exactly, and it follows that (1.28) is stable! Similarly, (1.28) is stable for interpolation on all intervals of the form $[b - 2\pi, b]$, for $b \leq -2\pi$. Thus, there is no need to modify (1.28) in these circumstances.

For other intervals, i.e., length-2π subintervals $[a, b]$ of $(-4\pi, 4\pi)$, we can interpolate stably in the vast majority of cases using a modified version of the algorithm of Section 2.3 under some additional assumptions that are given in the discussion below. The formulas (2.2) and (2.3) are still applicable; we just need to change how we compute the arguments to the sine function in the modified terms.

If we can show that these can be computed to high relative accuracy, then the rest of the analysis in Section 2.4 can be applied with only very minor modifications to conclude that the resulting algorithm is stable. As before, there are two issues that must be handled:[10] how to group the terms and how to correct for the fact that 2π cannot be represented exactly in floating-point arithmetic.

For the former, the appropriate generalization in the case where $\alpha < 1/2$ is to compute $x - x_0 - 2\pi$ as $(x - b) - (x_0 - a)$, while for $\alpha > 1/2$, we compute $x - x_{K-1} + 2\pi$ as $(x - a) - (x_{K-1} - b)$. These arrangements have the same previously identified crucial property that the terms in the final subtraction have opposite signs so that the only cancellation that occurs is the benign cancellation in each of the individual subtractions $x - b$ and $x_0 - a$.

The latter issue is more delicate, since the approximation $\mathrm{fl}(2\pi) \approx 2\pi$ does not enter into the computation directly. Instead, what we must correct for is the deviation of $b - a$ from 2π that we get when a and b are floating-point numbers. Ideally, we would have $b - a = \mathrm{fl}(2\pi)$ so that we could correct the error using the quantity c introduced previously, but this is not guaranteed. In general, the most we can say is that $b - a = \mathrm{fl}(2\pi) + \gamma$ for some γ that is hopefully not too large.

This leads us to the two key assumptions we make for the remainder of this section. First, we assume that $|b - a - 2\pi| \leq |x - x_0 - 2\pi|$ in the case where $\alpha < 1/2$; for $\alpha > 1/2$, we assume $|a - b + 2\pi| \leq |x - x_{K-1} + 2\pi|$. These inequalities would hold if $b - a$ were exactly 2π, but they can fail when b and a are floating-point numbers whose difference merely approximates 2π.

Second, we assume that γ itself is a floating-point number. At first glance, this seems rather restrictive, but it actually holds quite often as we now explain. Suppose for the moment that $\pi \leq b \leq 4\pi$. Then, by Theorem 2.6, $b - \mathrm{fl}(2\pi)$ is exactly a floating-point number. If a has been chosen well, then a and $b - \mathrm{fl}(2\pi)$ will not be far from each other. If they are close enough to each other that the subtraction $\gamma = \bigl(b - \mathrm{fl}(2\pi)\bigr) - a$ can be done exactly in floating-point arithmetic, then we are done. Looking to Theorem 2.6 once again, this is guaranteed if $a/2 \leq b - \mathrm{fl}(2\pi) \leq 2a$. Similarly, if $-4\pi \leq a \leq -\pi$, then $a + \mathrm{fl}(2\pi)$ is exactly a floating-point number, and if $b/2 \leq a + \mathrm{fl}(2\pi) \leq 2b$, then $\gamma = b - \bigl(a + \mathrm{fl}(2\pi)\bigr)$ will be a floating-point number as well.

Since any length-2π subinterval $[a, b]$ of $[-4\pi, 4\pi]$ has either $-4\pi \leq a \leq -\pi$ or $\pi \leq b \leq 4\pi$, and since the conditions imposed by Theorem 2.6 are rather mild, γ will be exactly a floating-point number in virtually every case of practical interest. In particular, this is true for interpolation on $[-\pi, \pi]$ with $a = -\mathrm{fl}(\pi)$, $b = \mathrm{fl}(\pi)$ (in fact, $\gamma = 0$ in that case), which is arguably the most important interval for trigonometric interpolation aside from $[0, 2\pi]$. Note that the discussion of the preceding paragraph also gives a way to compute γ in floating-point arithmetic for a given a and b.

To convert γ into an approximation of $b - a - 2\pi$, it remains to correct for the error in the approximation $\mathrm{fl}(2\pi) \approx 2\pi$. For reasons similar to those given in the remarks following the proof of Theorem 2.5, using c alone will not suffice. We must additionally correct for the rounding error in the

[10] We remark that similar issues—with similar resolutions—arise in the investigations of [80] into the effects of rounding errors in the interpolation points on the performance of the barycentric formulas for polynomial interpolation.

approximation $c \approx -2\pi\delta_{2\pi}$ using the constant c_2 defined previously. We compute, in floating-point arithmetic,

$$\begin{aligned}
\mathrm{fl}\big((\gamma - c) - c_2\big) &= \big(\big(b - a - \mathrm{fl}(2\pi) - c\big)(1+\delta_1) - c_2\big)(1+\delta_2) \\
&= \big(\big(b - a - 2\pi + 2\pi\delta_{2\pi}\delta_c\big)(1+\delta_1) - 2\pi\delta_{2\pi}\delta_c - 2\pi\delta_{2\pi}\delta_c\delta_{c_2}\big)(1+\delta_2) \\
&= (b - a - 2\pi)\left(1 + \delta_1 + 2\pi\delta_{2\pi}\delta_c\frac{\delta_1 - \delta_{c_2}}{b - a - 2\pi}\right)(1+\delta_2), \qquad (2.14)
\end{aligned}$$

where $|\delta_1|$ and $|\delta_2|$ are at most u.

Our assumption that γ is a floating-point number has the consequence that we can bound $|b - a - 2\pi|$ from below, for $|b - a - 2\pi| = |b - a - \mathrm{fl}(2\pi) + 2\pi\delta_{2\pi}| = |\gamma - (-2\pi\delta_{2\pi})|$. Since $-\gamma$ is a floating-point number, and since c is the closest floating-point number to $2\pi\delta_{2\pi}$, the right-hand side can be no smaller than $|c - (-2\pi\delta_{2\pi})| = 2\pi|\delta_{2\pi}\delta_c|$. It follows that

$$\left|2\pi\delta_{2\pi}\delta_c\frac{\delta_1 - \delta_{c_2}}{b - a - 2\pi}\right| \le |\delta_1| + |\delta_{c_2}| \le 2u,$$

and hence, multiplying out the error terms in (2.14), we find that we may write $\mathrm{fl}\big((\gamma - c) - c_2\big) = (b - a - 2\pi)(1 + \xi_\gamma)$, where $|\xi_\gamma| \le 5u$ and where we have implicitly made the reasonable assumption that $4u \le 1$. Thus, $\mathrm{fl}\big((\gamma - c) - c_2\big)$ is a high relative accuracy approximation to $b - a - 2\pi$.

At last, we can show how to compute the argument to the sine function in the modified terms of (2.2) and (2.3) to the needed accuracy. As usual, we consider the $\alpha < 1/2$ case, the $\alpha > 1/2$ case being similar. First, we compute, using the grouping of the terms prescribed earlier in this section

$$\begin{aligned}
\mathrm{fl}\big((x - b) - (x_0 - a)\big) &= \big((x - b)(1+\delta_1) - (x_0 - a)(1+\delta_2)\big)(1+\delta_3) \\
&= \big(x - x_0 - (b - a)\big)\left(1 + \frac{\delta_1(x - b)}{(x - b) - (x_0 - a)} + \frac{\delta_2(x_0 - a)}{(x - b) - (x_0 - a)}\right)(1+\delta_3),
\end{aligned}$$

where $|\delta_1|$, $|\delta_2|$, and $|\delta_3|$ are all at most u. Since $x - b$ and $-(x_0 - a)$ have the same sign, the factors multiplying δ_1 and δ_2 in the second bracketed factor are at most 1 in absolute value, and it follows that we have $\mathrm{fl}\big((x - b) - (x_0 - a)\big) = \big(x - x_0 - (b - a)\big)(1 + \xi_1)$, where $|\xi_1| \le 4u$, assuming that $2u \le 1$. Writing $c' = \mathrm{fl}\big((\gamma - c) - c_2\big)$ for brevity, we now apply the correction we just computed to obtain

$$\begin{aligned}
\mathrm{fl}\big(\big((x - b) - (x_0 - a)\big) - c'\big) &= \big((x - x_0 - (b - a))(1 + \xi_1) - (b - a - 2\pi)(1 + \xi_\gamma)\big)(1 + \delta_1) \\
&= (x - x_0 - 2\pi)\left(1 + \xi_1\frac{x - x_0 - (b - a)}{x - x_0 - 2\pi} + \xi_\gamma\frac{b - a - 2\pi}{x - x_0 - 2\pi}\right)(1 + \delta_4),
\end{aligned}$$

where $|\delta_4| \le u$. By our assumption that $|b - a - 2\pi| \le |x - x_0 - 2\pi|$, the factors multiplying ξ_1 and ξ_γ in this equation are bounded in magnitude by 2 and 1, respectively. Thus, we have $\mathrm{fl}\big(\big((x - b) - (x_0 - a)\big) - c'\big) = (x - x_0 - 2\pi)(1 + \xi)$ with $|\xi| \le 15u$, assuming that $13u \le 1$, as desired.

Applying similar arguments for the case where $\alpha > 1/2$, we can summarize our findings in this section as follows:

- If $a \geq 2\pi$ or $b \leq -2\pi$, we can interpolate stably using (1.28) directly.

- If $[a, b] \subset (-4\pi, 4\pi)$ with $b - a = \mathrm{fl}(2\pi) + \gamma$, we can interpolate stably if γ is exactly a floating-point number and if $|b-a-2\pi| \leq |x-x_0-2\pi|$ when $\alpha \in [0, 1/2)$ or $|a-b+2\pi| \leq |x-x_{K-1}+2\pi|$ when $\alpha \in (1/2, 1]$. These assumptions are not always valid, but they hold in many cases.

- If both assumptions in the last item hold, the interpolant can be computed by calculating γ via

  - $\big(b - \mathrm{fl}(2\pi)\big) - a$ if $\pi \leq b < 4\pi$ or
  - $b - \big(\mathrm{fl}(2\pi) + a\big)$ if $-4\pi < a \leq -\pi$

  and then computing the correction factor $c' = (\gamma - c) - c_2$. Then, there are three cases:

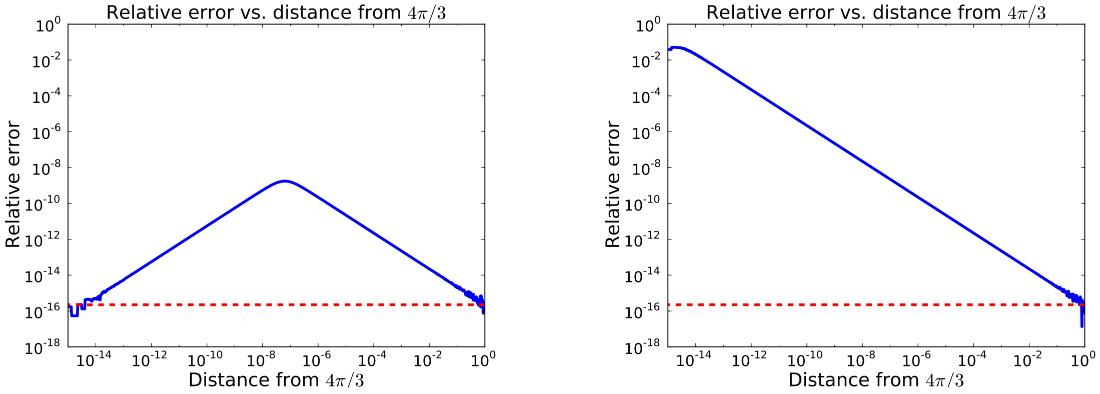
  - If $\alpha \in [0, 1/2)$, use (1.28) for $x \in [a, b-\pi(1-2\alpha)/K]$. Otherwise, use (2.2) with $x-x_0-2\pi$ computed as $\big((x - b) - (x_0 - a)\big) - c'$.
  - If $\alpha = 1/2$, use (1.28) for all $x \in [a, b]$.
  - If $\alpha \in (1/2, 1]$, use (1.28) for $x \in [a + \pi(2\alpha - 1)/K, b]$. Otherwise, use (2.3) with $x - x_{K-1} + 2\pi$ computed as $\big((x - a) - (x_{K-1} - b)\big) - c'$.

2.6 Interpolation in an Even Number of Points

Our analysis in this chapter has focused exclusively on the version of the second formula applicable to an odd number K of equispaced points. We close with a few words about the case of even K.

The even-K counterpart of (1.28) was given in (1.31) in Section 1.3.4. It is identical to (1.28) except that the sine function is replaced by the tangent. The instability in (1.28) emerged from the poor conditioning of the sine function near $\pm\pi$. The tangent function is also poorly conditioned near $\pm\pi$, so one would expect (1.31) to suffer from a similar instability. Indeed, this is the case, and it can be corrected analogously.

This is not the end of the story, however, as the tangent function additionally suffers from poor conditioning near $\pm\pi/2$, suggesting the possibility of an instability in (1.31) when $|x - x_k|$ is close to π for some k, an instability that (1.28) does not possess. At first glance, it seems that this is not an issue, since if $|x - x_k|$ is close to π, then $\tan\big((x - x_k)/2\big)$ is large. If the interpolation data in the vector f are all comparable in magnitude to one another, this means that the kth terms in the sums in (1.31) will be small relative to the rest so that while they may have been evaluated inaccurately due to the poor conditioning, their contribution to the final result will be negligible. If the datum f_k is much larger than the others, however, it will offset the growth in $\tan\big((x-x_k)/2\big)$, the kth term in the numerator will not be relatively small, and the instability will manifest itself.



(a)

(b)

Figure 2.4: Illustration of the instability in (1.31) when $|x - x_k|$ is close to π for some k. The solid blue lines show the relative error in the evaluations for the experiment described in Section 2.6 with (a) $f_1 = 10^{15}$ and (b) $f_1 = 10^{30}$. The dashed red lines depict the product of the condition number (computed using the even-K analogue of Lemma 2.2) and the unit roundoff $u = 2^{-52}$.

We can illustrate these effects with the following numerical example. Let $\alpha = 0$ and $K = 6$, so that the grid points are $x_k = k\pi/3$, $0 \le k \le 5$, and let the interpolation data be $f_0 = f_2 = f_3 = f_4 = f_5 = 1$ and $f_1 = 10^{15}$. We evaluate the interpolant using (1.31) at several points x near $4\pi/3$ so that $|x - x_1|$ is close to π. Just as in the experiments of Sections 2.2 and 2.3, we perform the evaluation once using double-precision arithmetic and once using higher-precision arithmetic and then compute the relative error in the former, taking the latter as "exact." Since f_1 is considerably larger than the other interpolation data, we expect to see evidence of instability.

The results are displayed in Figure 2.4a, which plots the relative error as a function of the distance of x from $4\pi/3$. As predicted, the formula does indeed exhibit instability. The "pyramid" shape of the error curve is due to the fact that $4\pi/3$ is a grid point, x_4. As x gets closer to $4\pi/3$, $\tan\bigl((x - x_4)/2\bigr)$ shrinks, causing the $k = 4$ term in the sum in the numerator of (1.31) to become larger. At the same time, the $k = 1$ term that suffers from the ill conditioning of the tangent function becomes smaller, as explained previously; it only retains its significance because of the large magnitude of f_1. For the value of f_1 that we have chosen, the $k = 1$ term is the dominant term in the sum until the distance from $4\pi/3$ has decreased to about 10^{-7}, after which the $k = 4$ term takes over. Since the evaluations of the tangent function in the $k = 4$ term are all well-conditioned for x in the chosen range, it is computed to high relative accuracy. Hence, we expect the error in the overall evaluation to improve as it takes more and more control from the $k = 1$ term.

We can verify the correctness of this explanation by making the datum f_1 so large that the $k = 1$ term always makes a significant contribution to the sum, even when the distance between x and $4\pi/3$ is very small. In this case, we expect to see the error rise steadily as the distance shrinks. These expectations are confirmed by Figure 2.4b, which shows the results of running the same experiment with $f_1 = 10^{30}$.

Regrettably, this new instability is not as easily corrected as the one described in Section 2.2. The technique from Section 2.3 of using periodicity to change the interpolation grid is not applicable here: if $|x - x_k|$ is close to π, then $|x - (x_k + 2n\pi)|$ will be close to an odd multiple of π for any integer n, so the evaluation of $\tan\bigl((x - (x_k + 2n\pi))/2\bigr)$ associated with the adjusted point in the resulting modified formula is still poorly conditioned. Moreover, there are more ways to excite this new instability than there are for the previous one, since for a given interpolation grid, there are several choices of x and x_k such that $|x - x_k|$ is close to π, while there is only one such that that $|x - x_k|$ is close to 2π. A method for stabilizing (1.31), if one exists, will likely require several modified formulas, one for each possible case, in addition to the even-K analogues of (2.2) and (2.3).

Chapter 3

Trigonometric Interpolation in Non-Equispaced Points I[1]

In the last chapter, we focused exclusively on the particular case of trigonometric interpolation in equispaced points. In this chapter and the next, we consider the problem of trigonometric interpolation in non-equispaced points and study the approximation properties of trigonometric interpolation in what we call perturbed equispaced grids. We also consider the related problem of quadrature. Note that while we discuss only trigonometric interpolation, thanks to the correspondence between trigonometric and polynomial interpolation outlined in Section 1.3.1, all of our results possess analogues for the polynomial case as well.

3.1 Introduction

As mentioned in Section 1.3.3, equispaced points constitute an optimal grid for trigonometric interpolation. When one is free to choose the points that are used, there is usually no reason to consider anything else. In applications, however, this is not always the case. For instance, practical considerations in biomedical [141] and satellite imaging [3, 40] make the use of irregular sampling grids appealing or even mandatory. Further examples may be found in the review article [2] and the references therein. We are thus led to consider the problem of interpolation in non-equispaced points.

Just as with polynomial interpolation, trigonometric interpolation in arbitrary points is, in general, badly behaved. If the points are not properly—that is, uniformly[2]—distributed, one can get divergence even for holomorphic functions via the trigonometric version of the Runge phenomenon

[1] The content of this chapter and the next contains work carried out by the author in collaboration with his doctoral supervisor Lloyd N. Trefethen. Trefethen suggested the possibility of proving theorems for approximation and quadrature via trigonometric interpolation in perturbed equispaced points by studying the Lebesgue constant. The author performed the extensive numerical investigations presented in the text and worked out the proof for the bound on the Lebesgue constant given in Theorem 3.6.

[2] By "uniformly distributed", we mean that the fraction of points lying in a subinterval $[a, b]$ of $[-\pi, \pi]$ tends to $(b - a)/(2\pi)$ as the number of points tends to infinity. This definition dates back to Weyl [148]. In contrast to polynomial interpolation, for trigonometric interpolation, uniformly distributed points are equidistributed according to (the trigonometric analogue of) the definition given in Section 1.2.7.

(see Section 1.2.7). Even when the points *are* properly distributed, one can still run into problems when computing interpolants numerically if the Lebesgue constants (see Sections 1.2.8 and 1.3.3) are large, as can happen, e.g., if two of the interpolation points are very close together. The latter statement can be rephrased as saying that the convergence guaranteed in theory by the Fejér–Kalmár–Walsh theorem (Theorem 1.12), which makes suppositions only about the asymptotic distribution of the points and not the locations of the points themselves, is not robust numerically.

Perhaps even more fundamentally, the Fejér–Kalmár–Walsh theorem does not apply at all to functions that are not holomorphic. It gives no information about the convergence or divergence of trigonometric interpolation for functions that are merely C^∞, much less those that possess only a finite number of continuous derivatives. While interpolation in equispaced points can be shown to converge for such functions by other means, interpolation in non-equispaced grids, even those that are uniformly distributed, is not guaranteed even in theory without further assumptions on the locations of the points. In these matters, as in those discussed above, the size of the Lebesgue constants again plays an important role.

These observations were made by Trefethen and Weideman in [135], who examined the stability of the exponential convergence of the trapezoid rule for integrating periodic functions under perturbations of the quadrature nodes. Recall that the trapezoid rule is an interpolatory rule: it approximates the integral of a function by that of a trigonometric interpolant to the function in a given number of equispaced points. It converges exponentially fast as the number of points increases when the integrand is periodic and holomorphic; we noted this fact during our brief discussion of Cauchy integrals in Section 1.2.3. Trefethen and Weideman observe that if the equispaced points are replaced by non-equispaced ones (but the procedure—interpolate, then integrate—is kept the same), the convergence rate degrades[3] but remains exponential. Nevertheless, they remark that the practical utility of this fact is limited if the quadrature nodes can be arbitrarily close together, causing the underlying interpolation problem to become poorly conditioned. They also note that preventing the nodes from coalescing is necessary if the quadrature rule is to converge for functions that are not holomorphic.

Similar issues arise in the related field of sampling theory, in which one takes a class of functions defined on a given domain and seeks to reconstruct members of that class from samples of their values on a "thin" subset of the domain. This field includes polynomial and trigonometric interpolation as special cases but also incorporates much wider classes of problems in which the number of samples taken need not be finite. Perhaps the most basic result of this type is the statement that band-limited functions—functions that have limited frequency content in the sense that the supports of their Fourier transforms are contained within $[-\pi, \pi]$ (see Section 3.2.3)—can be recovered from knowledge

[3] The degradation comes from the fact that the trapezoid rule in equispaced points is not just an interpolatory rule but a Gauss-type rule: the trapezoid rule in $K = 2N + 1$ equispaced points exactly integrates trigonometric polynomials of degree $2N$, not N. When non-optimal (i.e., non-equispaced) nodes are employed, this "Gauss-ness" is lost, and the degree of accuracy is reduced to N.

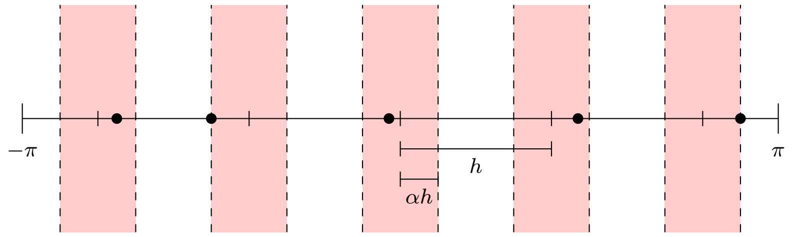


Figure 3.1: A perturbed equispaced grid in $[-\pi, \pi]$. The black dots show the perturbed points $\widetilde{x}_k$, and the tick marks show the underlying equispaced grid x_k. The shaded areas mark the subintervals in which the perturbed points are permitted to lie for a given choice of the parameter α.

of their values at the integers: this is the famous *Shannon–Whittaker–Kotel'nikov sampling theorem* [71, 126, 149], and it has many applications in signal processing and communication theory. The reconstruction guaranteed by this theorem is stable[4] in the sense that small perturbations to the samples yield only small changes in the reconstructed function. This is an important feature for practical applications, as samples obtained from real measurements are almost always contaminated by noise.

A natural question to ask is to what extent the Shannon–Whittaker–Kotel'nikov result can be extended to more general, non-uniform sampling sets. The simplest place to begin investigating this question is with sets that are perturbations of the integers. If $\{\lambda_n\}_{n=\infty}^{\infty}$ is a sequence of real numbers such that $|\lambda_n - n| \leq \alpha$ for some α, what can we say about our ability to recover a band-limited function F stably from knowledge of the values $F(\lambda_n)$? An often-quoted theorem of Kadec (see Theorem 3.1) states that such a reconstruction is possible if $\alpha < 1/4$, and, moreover, an example due to Levinson (see (3.14)) shows that this result is sharp: if α is taken equal to $1/4$, there is a choice for the points λ_n such that stable reconstruction is not possible. Thus, in this setting, even ensuring that the sample points are separated is insufficient; they must further be "well-separated" according to Kadec's criterion to guarantee success.

Returning to the setting of trigonometric interpolation, these observations lead us to ask several questions. What can we say about trigonometric interpolation in grids that are uniformly distributed but non-equispaced and that also satisfy a condition requiring the points to remain separated to ensure numerical robustness? What approximation theorems can we prove, and what about the corresponding quadrature scheme? Is mere separation of the points sufficient for robustness, or is something further akin to Kadec's $1/4$ condition required as well?

In this chapter and the next, we investigate these questions for trigonometric interpolation in perturbed equispaced points. Specifically, let $K = 2N + 1$ be an odd integer, $K \geq 1$, let $h = 2\pi/K$, and let

$$x_k = kh, \qquad -N \leq k \leq N \tag{3.1}$$

[4] A numerical analyst would probably prefer the term "well-conditioned" instead of "stable," since we are describing the reconstruction problem itself, not an algorithm for solving it. Nevertheless, we will, throughout this chapter, speak of "stable reconstructions," and "sets of stable sampling," as this is the terminology used in the literature.

be a zero-centered grid of K equispaced points in $[-\pi, \pi]$. We consider points $\widetilde{x}_k$, $-N \leq k \leq N$, that are perturbations of the points x_k, where the perturbation is at most a fraction α, $0 \leq \alpha < 1/2$,[5] of the grid spacing h:

$$\widetilde{x}_k = x_k + t_k h, \qquad |t_k| \leq \alpha. \tag{3.2}$$

It is easy to verify that the points $\widetilde{x}_k$ are uniformly distributed in $[-\pi, \pi]$ as $K \to \infty$. The condition $\alpha < 1/2$ ensures that they remain a minimum distance apart. For an illustration, see Figure 3.1.

Our discussion is organized as follows. After reviewing some theoretical concepts in Section 3.2 that are needed to establish context for our later remarks, in Section 3.3 we take up the task of studying trigonometric interpolation and approximation in the perturbed equispaced grids just described using a mixture of theory and numerical computations. Our main result is Theorem 3.6, which provides a bound on the Lebesgue constant for these grids as the number of points increases. The proof of this result, which is lengthy, is presented in the next chapter. We use this result to derive theorems about approximation by trigonometric interpolants in these grids. In Section 3.4, we numerically study the problem from the perspective of the 2-norm instead of the more usual uniform norm and conjecture a version of Kadec's result that applies to trigonometric interpolation, though its practical implications seem to be minimal. Finally, in Section 3.5, we discuss Trefethen and Weideman's original question about trigonometric interpolatory quadrature.

Our main findings can be summarized as follows:

1. We show that the Lebesgue constant for grids of the form (3.2) grows at an algebraic rate as $K \to \infty$ and make a conjecture as to the true growth rate in the worst case (Theorem 3.6 and Conjecture 3.5).

2. Using this result, we prove that for all α, $0 < \alpha < 1/2$, if f has $\nu \geq 1$ derivatives with $f^{(\nu)}$ Hölder continuous with an appropriate exponent (depending on α), then its trigonometric interpolants in a sequence of grids of the form (3.2) converge uniformly as $K \to \infty$ (Theorem 3.8). Larger values of ν yield more rapid convergence. For $\alpha < 1/4$, f need not be differentiable; Hölder continuity of f (again, with a suitable exponent that depends on α) suffices for convergence. We conjecture that this is actually true for all α, $0 < \alpha < 1/2$ (Conjecture 3.7).

3. We prove a similar result and make a similar conjecture for convergence of the corresponding quadrature rule (Theorem 3.12 and Conjecture 3.11). We provide numerical evidence in support of a further conjecture that the quadrature rule converges if f is merely continuous (Conjecture 3.14).

[5] Note that in this chapter and the next, the variable α has a meaning different from and completely unrelated to the one it had in Chapter 2.

We believe that these are the first results for this problem to appear in the literature. Nevertheless, we do not pretend that our work is the final word on these matters. To the contrary, as the reader will see, there are many interesting questions that remain open for future study.

3.2 Theoretical Background

In this section, we set the stage for our discussions by recalling some concepts from sampling theory, the theory of bases for Hilbert spaces, and the theory of non-harmonic Fourier series. We will see that questions about the existence of bases for $L^2([-\pi, \pi])$ consisting of non-harmonically related (and thus generally non-orthogonal) complex exponentials are equivalent to questions about the existence of non-equispaced interpolation schemes for a certain class of entire functions. For these schemes to be stable, it will turn out that not just a basis but a *Riesz* basis is required. Our exposition has been heavily influenced by the excellent introductory treatise of Young [154], to which we refer the reader for proofs and further information. Other useful references include [56], [113, Ch. 19], and [153].

3.2.1 Bases in Hilbert Space

Let X be an infinite-dimensional Hilbert space equipped with an inner product $\langle \cdot, \cdot \rangle$ (semilinear in the second factor) with accompanying norm $\| \cdot \|$. A sequence $\{x_n\}_{n=1}^{\infty}$ of vectors in X is said to be a *(Schauder) basis*[6] for X if for every $x \in X$, there is a unique sequence $\{c_n\}_{n=1}^{\infty}$ of scalars such that the expansion

$$x = \sum_{n=1}^{\infty} c_n x_n \tag{3.3}$$

is valid in the topology of X. Clearly, for X to possess a basis, it must be separable. Conversely, every separable Hilbert space possesses a basis; in fact, it possesses a basis that is *orthonormal*, i.e., that satisfies $\|x_n\| = 1$ and $\langle x_m, x_n \rangle = 0$ for $m \neq n$. It is an immediate consequence of the definition that a basis for X is *complete*[7] in X, i.e., its linear span is dense in X; equivalently, the only vector $x \in X$ that satisfies $\langle x, x_n \rangle = 0$ for all n is $x = 0$. A basis is also *exact*: removing any one element from a basis yields a set that is incomplete. If $\{x_n\}_{n=1}^{\infty}$ is an orthonormal sequence in X, then it is a basis if and only if it is complete.

Orthonormal bases are by far the most important bases, and they enjoy a number of convenient properties. If $\{e_n\}_{n=1}^{\infty}$ is an orthonormal basis for X, then the expansion (3.3) takes the form

$$x = \sum_{n=1}^{\infty} \langle x, e_n \rangle \, e_n. \tag{3.4}$$

[6] This is in contrast to a *Hamel basis* for X, which is a subset V of X such that every $x \in X$ can be written uniquely as a finite linear combination of elements of V. Using the axiom of choice, one can show that every infinite-dimensional vector space possesses a Hamel basis (in fact, this statement is *equivalent* to the axiom of choice), but these bases are of limited use even theoretically, much less practically.

[7] This is not to be confused with the more usual definition that a subset V of X is complete if every Cauchy sequence in V converges in V. In the literature (especially older literature), the word "closed", itself not free from definitional clash, is sometimes used instead of "complete."

We also have the following representation for the inner product:

$$\langle x, y \rangle = \sum_{n=1}^{\infty} \langle x, e_n \rangle \, \overline{\langle y, e_n \rangle}, \tag{3.5}$$

and upon taking $y = x$ in this equation, we obtain the following representation for the norm:

$$\|x\|^2 = \sum_{n=1}^{\infty} | \langle x, e_n \rangle |^2. \tag{3.6}$$

This last equation is known as *Parseval's identity.*

Another important class of bases consists of those that can be obtained from orthonormal bases via a linear transformation. We say that a basis $\{f_n\}_{n=1}^{\infty}$ is a *Riesz basis* for X if there is an orthonormal basis $\{e_n\}_{n=1}^{\infty}$ for X and a bounded invertible[8] linear operator T on X such that $f_n = Te_n$ for each n. Every orthonormal basis is trivially a Riesz basis. The properties for orthonormal bases given in the previous paragraph do not hold for general Riesz bases; however, in place of (3.4), we have

$$x = \sum_{n=1}^{\infty} \langle x, g_n \rangle \, f_n \tag{3.7}$$

for all $x \in X$, where $g_n = (T^*)^{-1} e_n$ for each n and T^* denotes the adjoint of T. The sequence $\{g_n\}_{n=1}^{\infty}$ is the unique sequence in X that is *biorthogonal* to $\{f_n\}_{n=1}^{\infty}$, i.e., $\langle f_m, g_n \rangle = 1$ if $m = n$ and 0 if $m \neq n$.[9] Note that $\{g_n\}_{n=1}^{\infty}$ is itself a Riesz basis and that by exchanging the roles of $\{f_n\}_{n=1}^{\infty}$ and $\{g_n\}_{n=1}^{\infty}$, we also have the expansion

$$x = \sum_{n=1}^{\infty} \langle x, f_n \rangle \, g_n. \tag{3.8}$$

Combining (3.7) and (3.8), we have the following replacement for (3.5):

$$\langle x, y \rangle = \sum_{n=1}^{\infty} \langle x, g_n \rangle \, \overline{\langle y, f_n \rangle}. \tag{3.9}$$

Formulas (3.7)-(3.9) actually hold if $\{f_n\}_{n=1}^{\infty}$ is only a basis and not a Riesz basis (though the formula given for the biorthogonal sequence $\{g_n\}_{n=1}^{\infty}$ is no longer valid; one must treat the problem more abstractly). What sets Riesz bases apart is the following property. Multiplying (3.8) through on the left by T^* and taking norms, we find

$$\frac{1}{\|(T^*)^{-1}\|} \|x\| \leq \left\| \sum_{n=1}^{\infty} \langle x, f_n \rangle \, e_n \right\| \leq \|T^*\| \|x\|.$$

Using the fact that $\|T^*\| = \|T\|$ and applying (3.6), we thus obtain

$$\frac{1}{\|T^{-1}\|^2} \|x\|^2 \leq \sum_{n=1}^{\infty} | \langle x, f_n \rangle |^2 \leq \|T\|^2 \|x\|^2, \tag{3.10}$$

[8] Recall that the inverse of a bounded invertible linear operator on a Hilbert space is also bounded by the open mapping theorem.

[9] Proof: $\langle f_m, g_n \rangle = \langle Te_m, (T^*)^{-1} e_n \rangle = \langle T^{-1} Te_m, e_n \rangle = \langle e_m, e_n \rangle$. That $\{g_n\}_{n=1}^{\infty}$ is the only sequence in X biorthogonal to $\{f_n\}_{n=1}^{\infty}$ follows from the fact that $\{f_n\}_{n=1}^{\infty}$ is complete: if $\{g_n'\}_{n=1}^{\infty}$ is another such sequence, then $\langle f_m, g_n - g_n' \rangle = 0$ for all m for each n, so g_n must be equal to g_n'.

a relaxed form of Parseval's identity. The inequalities (3.10) show that every Riesz basis constitutes a *frame* in X [32], [154, Ch. 4, §7]. More generally, a frame in X is any complete sequence $\{x_n\}_{n=1}^{\infty}$ of vectors in X for which there exist constants $A, B > 0$, the *frame bounds*, such that

$$A\|x\|^2 \leq \sum_{n=1}^{\infty} |\langle x, x_n \rangle|^2 \leq B\|x\|^2.$$

for all $x \in X$. Of these two inequalities, the first is more important. We can interpret it as a statement about the conditioning of the problem of reconstructing x from the knowledge of its *moment sequence* $\{\langle x, x_n\rangle\}_{n=1}^{\infty}$ with respect to $\{x_n\}_{n=1}^{\infty}$.[10] In effect, it says that $1/A$ is a Lebesgue constant of sorts for the solution process (see Section 1.2.8).[11] Similarly, the upper frame bound B provides a quantitative measure of how much a perturbation in x can affect the size of its moments.

It turns out that a frame is a Riesz basis if and only if it is exact. Thus, a basis is a Riesz basis if and only if it is also a frame. We can therefore think of Riesz bases informally as bases with additional constraints that prevent them from being infinitely badly conditioned.

Before moving on, we pause to remark that the theory just presented is not trivial in that not all bases are Riesz bases. It is easy to construct examples of bases that are not Riesz bases using the fact that any Riesz basis $\{f_n\}_{n=1}^{\infty}$ for a Hilbert space is *bounded*, i.e.,

$$0 < \inf_{n\geq 1} \|f_n\| \leq \sup_{n\geq 1} \|f_n\| < \infty.$$

This is easy to prove: if $\{e_n\}_{n=1}^{\infty}$ is an orthonormal basis and T is a bounded invertible linear operator such that $Te_n = f_n$ for each n, then $1/\|T^{-1}\| \leq \|f_n\| \leq \|T\|$. As a concrete example, let $\{e_n\}_{n=1}^{\infty}$ be the standard orthonormal sequence in $\ell^2(\mathbb{C})$. The sequences $\{e_n/n\}_{n=1}^{\infty}$ and $\{ne_n\}_{n=1}^{\infty}$ are readily seen to be bases for $\ell^2(\mathbb{C})$, but they are not bounded and hence are not Riesz bases. It is true that there are even bounded bases that are not Riesz bases, but these are considerably more difficult to construct. Babenko has provided an explicit example of such a basis for $L^2([-\pi,\pi])$ [9].

3.2.2 Non-Harmonic Fourier Series and Kadec's 1/4 Theorem

Let us now consider the material of the preceding section in the familiar context of Fourier series in $L^2([-\pi,\pi])$. Taking the inner product to be the normalized one,

$$\langle f, g\rangle_{L^2([-\pi,\pi])} = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}\, dx, \tag{3.11}$$

we recall that the sequence $\{e^{inx}\}_{n=-\infty}^{\infty}$ of harmonically-related complex exponentials is an orthonormal basis for this space. Thus, for every $f \in L^2([-\pi,\pi])$, we have the Fourier expansion

$$f(x) = \sum_{n=-\infty}^{\infty} \left\langle f, e^{inx}\right\rangle e^{inx}, \tag{3.12}$$

[10] If $\{x_n\}_{n=1}^{\infty}$ is a Riesz basis, then the solution to this problem is given by the biorthogonal expansions (3.7) and (3.8). For general frames, it can be accomplished via similar expansions using the *dual frame* [56, §3.3].

[11] More precisely, $1/A$ is an *upper bound* on a Lebesgue constant, since A can be taken arbitrarily small.

the series converging in $L^2([-\pi, \pi])$.[12] In (3.12), we have dropped the subscript from the inner product to reduce clutter; we will continue to do this in what follows except where necessary for clarity. Note also that we have made the customary abuse of notation, writing $\langle f, e^{inx} \rangle$ to mean $\langle f, e_n \rangle$, where $e_n(x) = e^{inx}$. That is, in the expression $\langle f, e^{inx} \rangle$, the variable x is to be interpreted as a dummy variable that merely indicates the independent variable in the function being considered, not as a parameter to which a fixed value can be assigned.

It is an intriguing question to ask what happens if we consider sequences of more general complex exponentials, i.e., if we replace $\{e^{inx}\}_{n=-\infty}^{\infty}$ with $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ for some sequence $\{\lambda_n\}_{n=\infty}^{\infty}$ of real numbers. The new sequence will not generally be orthonormal, but it may still be a basis, in which case each $f \in L^2([-\pi, \pi])$ will have a non-harmonic Fourier expansion

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\lambda_n x} \tag{3.13}$$

in $L^2([-\pi, \pi])$ for some choice of the coefficients c_n uniquely determined by f. Even better, we would like $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ to be a *Riesz* basis so that the expansion is not too badly conditioned in the sense outlined in the previous section.

It is clear that not just any choice of the λ_n will do. For instance, if $\lambda_n = 2n$, then $\langle e^{imx}, e^{i\lambda_n x} \rangle = 0$ for all odd integers m, and so $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is not complete in $L^2([-\pi, \pi])$. A natural place to start investigating these matters is to restrict attention to sequences $\{\lambda_n\}_{n=-\infty}^{\infty}$ that are not "too far" from the integers in that $|\lambda_n - n| \leq \alpha$ for some $\alpha \geq 0$. How big can we take α and still be guaranteed that $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is a Riesz basis?

In their foundational work on non-harmonic Fourier series from 1934, Paley and Wiener showed that $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is a Riesz basis[13] whenever $|\lambda_n - n| \leq \alpha < 1/\pi^2$ and posed the question of whether the constant $1/\pi^2$ could be improved [99]. In 1936, Levinson proved that the best constant could be no larger than 1/4 by showing that the sequence $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ obtained by setting

$$\lambda_n = \begin{cases} n + \frac{1}{4} & n < 0 \\ 0 & n = 0 \\ n - \frac{1}{4} & n > 0 \end{cases} \tag{3.14}$$

is not a Riesz basis [77]; in fact, it is neither a basis nor a frame. The constant was improved several times by different authors, notably Duffin and Eachus, who showed in 1942 that $|\lambda_n - n| \leq \alpha < \log(2)/\pi$ is sufficient and moreover showed that $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ forms a Riesz basis under these conditions even if the λ_n are taken to be complex [31]. The question was finally settled by Kadec in 1964, who showed that Levinson's upper bound of 1/4 is, in fact, optimal [66]:

[12] A result of Carleson [19] shows that the expansion also holds pointwise almost everywhere in $[-\pi, \pi]$, but we will not need this fact.

[13] The term "Riesz basis" did not exist in Paley and Wiener's time and therefore does not appear in their writing; however, their results have been reinterpreted in terms of Riesz bases by later scholars. (For instance, Kadec does this in the opening paragraph of [66].)

Theorem 3.1 (Kadec's 1/4 theorem). *If $\{\lambda_n\}_{n=-\infty}^{\infty}$ is a sequence of real numbers such that $|\lambda_n - n| \leq \alpha < 1/4$ for each n, then $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is a Riesz basis for $L^2([-\pi, \pi])$.*

3.2.3 The Paley–Wiener Theorem

The preceding discussions may seem to have taken us far from our original topic of interpolation, but we will see shortly that they are actually highly relevant. First, we observe that if $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is a basis for $L^2([-\pi, \pi])$, it must, at a minimum, be complete. Thus, it must be true that if $f \in L^2([-\pi, \pi])$, and if

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-i\lambda_n x}\, dx = 0, \qquad n \in \mathbb{Z}, \tag{3.15}$$

then $f = 0$ almost everywhere.

The key observation is the following. If $f \in L^2([-\pi, \pi])$, then the function $F : \mathbb{C} \to \mathbb{C}$ defined by

$$F(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-izx}\, dx \tag{3.16}$$

is an entire function[14] and satisfies a bound of the form

$$|F(z)| \leq Ce^{\pi|z|} \tag{3.17}$$

for some constant $C > 0$ that depends on F but not z; we say that F is of *exponential type at most* π. It is a fundamental fact that the rate of growth of an entire function and the distribution of its zeros are intimately connected; a restriction on one places a corresponding restriction on the other. As the conditions (3.15) are exactly the requirement that $F(\lambda_n) = 0$ for each n, we see that questions about the completeness of sequences of complex exponentials in $L^2([-\pi, \pi])$ are equivalent to ones about which sequences in $\mathbb{C}$ are admissible as zero sets of entire functions satisfying the growth bound (3.17). To be more explicit, if $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is *not* complete, then by taking a function $f \in L^2([-\pi, \pi])$ that satisfies (3.15) but that is not almost everywhere zero and using (3.16), we obtain an entire function of exponential type at most π that is not identically zero and that has zeros at each of the points λ_n.

Even more is true, however, because (3.16) defines F by taking the Fourier transform of a function in $L^2(\mathbb{R})$, the support of which is contained in $[-\pi, \pi]$. By Plancherel's theorem [113, p. 186], $F \in L^2(\mathbb{R})$, and $\|f\|_{L^2([-\pi,\pi])} = \|F\|_{L^2(\mathbb{R})}$, where the inner product on $L^2(\mathbb{R})$ is defined as usual by

$$\langle f, g \rangle_{L^2(\mathbb{R})} = \int_{-\infty}^{\infty} f(x)\overline{g(x)}\, dx.$$

The correspondence $f \mapsto F$ given by (3.16) thus defines an isometry[15] of $L^2([-\pi, \pi])$ into the subspace of $L^2(\mathbb{R})$ consisting of all entire functions in $L^2(\mathbb{R})$ that are of exponential type at most π. The crucial insight of Paley and Wiener is that this map is actually onto: every function belonging to this subspace of $L^2(\mathbb{R})$ can be obtained via the Fourier transform in this way [99].

[14] Proof: Apply Morera's theorem [113, p. 208].

[15] Note that this statement depends on our use of the normalized inner product (3.11) on $L^2([-\pi, \pi])$.

Theorem 3.2 (Paley–Wiener theorem)**.** *If F is an entire function of exponential type at most π belonging to $L^2(\mathbb{R})$, then there is a function $f \in L^2([-\pi, \pi])$ such that F and f are related by* (3.16)*.*

The space of entire functions of exponential type at most π that belong to $L^2(\mathbb{R})$ is called the *Paley–Wiener space* and will be denoted by PW_π. The Paley–Wiener theorem can be concisely summarized as saying that the Fourier transform defines an isometric isomorphism between $L^2([-\pi, \pi])$ and PW_π. In particular, note that it implies that the support of the Fourier transform of every function in PW_π is contained in $[-\pi, \pi]$. If we view PW_π as the "time domain" and $L^2([-\pi, \pi])$ as the "frequency domain", this says that a function in PW_π has no frequency content beyond $\pm\pi$. Because of this, in the engineering literature, PW_π is often referred to as the space of functions in $L^2(\mathbb{R})$ that are *band-limited to π*.

3.2.4 Interpolation in PW_π

We now have all we need to establish a link between the theory of non-harmonic Fourier series and interpolation. The Paley–Wiener theorem allows us to take questions and results about $L^2([-\pi, \pi])$ and turn them into ones about PW_π. In particular, any basis expansion in $L^2([-\pi, \pi])$ yields an equivalent one in PW_π. Consider the classic Fourier expansion (3.12). If F is the Fourier transform of f obtained via (3.16), then since $F(n) = \langle f, e^{inx} \rangle$, and since the Fourier transform of e^{inx} is

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{inx} e^{-izx} \, dx = \frac{\sin\bigl(\pi(z-n)\bigr)}{\pi(z-n)},$$

we immediately obtain the equivalent expansion

$$F(z) = \sum_{n=-\infty}^{\infty} F(n) \frac{\sin\bigl(\pi(z-n)\bigr)}{\pi(z-n)} \tag{3.18}$$

in PW_π regarded as a subspace of $L^2(\mathbb{R})$.[16] Thus, we see that any $F \in \mathrm{PW}_\pi$ can be reconstructed from its values at the integers via the formula (3.18), which we view as a Lagrange-type interpolation formula; this result is the Shannon–Whittaker–Kotel'nikov theorem mentioned in Section 3.1. The series (3.18) is sometimes called the *cardinal series* for $F \in \mathrm{PW}_\pi$. Observe that the basis element $\sin\bigl(\pi(z-n)\bigr)/\bigl(\pi(z-n)\bigr)$ behaves like a Lagrange basis function for the integer n in that it takes on the value 1 at $z = n$ and vanishes at every other integer.

What about the more general expansion (3.13)? Taking its Fourier transform will not yield anything immediately recognizable since the coefficients in the expansion are not the inner products of f with the complex exponentials $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$. If, however, we let $\{g_n(x)\}_{n=-\infty}^{\infty}$ be the sequence in $L^2([-\pi, \pi])$ that is biorthogonal to $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$, then we have the dual expansion

$$f(x) = \sum_{n=-\infty}^{\infty} \langle f, e^{i\lambda_n x} \rangle \, g_n(x),$$

[16] It is easy to show further via a simple estimate that the series converges uniformly on $\mathbb{R}$ [154, p. 90].

in $L^2([-\pi, \pi])$, and we can take the Fourier transform of this to obtain

$$F(z) = \sum_{n=-\infty}^{\infty} F(\lambda_n) G_n(z),$$

in PW_π, where G_n is the Fourier transform of g_n. This, too, is a Lagrange-type interpolation formula, but the interpolation points are no longer necessarily uniformly spaced. Notice that $G_n(\lambda_m) = \left\langle g_n, e^{i\lambda_m x} \right\rangle$, so G_n takes on the value 1 at λ_n and is 0 at λ_m for $m \neq n$.

If $\{e^{i\lambda_n x}\}_{n=-\infty}^{\infty}$ is not just a basis but a *Riesz* basis, we can say something further. By (3.10), there are constants $A, B > 0$ such that

$$A\|f\|_{L^2([-\pi,\pi])}^2 \leq \sum_{n=1}^{\infty} |\left\langle f, e^{i\lambda_n x} \right\rangle|^2 \leq B\|f\|_{L^2([-\pi,\pi])}^2$$

for all $f \in L^2([-\pi, \pi])$. Taking Fourier transforms, this becomes

$$A\|F\|_{L^2(\mathbb{R})}^2 \leq \sum_{n=1}^{\infty} |F(\lambda_n)|^2 \leq B\|F\|_{L^2(\mathbb{R})}^2 \tag{3.19}$$

for all $F \in \mathrm{PW}_\pi$. Informally, the first of the inequalities in (3.19) says that there cannot be much more "energy" in the samples of F at the points λ_n than there is in F itself, a condition which is usually interpreted in the engineering literature as saying that small errors in the samples cannot be magnified into large errors in the reconstructed function. Because of this, a sequence $\{\lambda_n\}_{n=-\infty}^{\infty}$ that satisfies such an inequality is often called a *set of stable sampling* for PW_π [56, Ch. 10] [152].

Kadec's 1/4 theorem (Theorem 3.1) gives a simple condition sufficient to ensure that a given sequence $\{\lambda_n\}_{n=-\infty}^{\infty}$ is a set of stable sampling for PW_π: require that $|\lambda_n - n| \leq \alpha < 1/4$ for some $\alpha \geq 0$. It is clear that this condition is merely sufficient, not necessary: setting $\lambda_n = n + 1$, for instance, we have $\{\lambda_n\}_{n=-\infty}^{\infty} = \mathbb{Z}$. Nevertheless, Levinson's example (3.14) shows that if Kadec's condition is violated, there is no guarantee that $\{\lambda_n\}_{n=-\infty}^{\infty}$ will be a set of stable sampling in general.

The problem of characterizing those sequences that yield sets of stable sampling for PW_π attracted the attention of several mathematicians in the second half of the 20th century. In 1979, Pavlov made an important advance by establishing a necessary and sufficient condition for a set of complex exponentials to form a Riesz basis for $L^2([-\pi, \pi])$ [102]. The problem was finally solved in 2002 by Ortega-Cerdà and Seip, who derived a necessary and sufficient condition for a set of complex exponentials to form a frame in $L^2([-\pi, \pi])$ [93].

3.2.5 Relationship with Trigonometric Interpolation

We conclude this introductory section with a brief discussion of the parallels between the continuous (infinite-dimensional) problem of interpolation in PW_π just discussed and the discrete (finite-dimensional) problem of trigonometric interpolation that is the focus of our work. These relationships are well-studied, and discussions can be found in most textbooks on signal processing.

Let $\mathcal{T}_N([-\pi,\pi])$ be the space of 2π-periodic trigonometric polynomials of degree N, regarded as a finite-dimensional subspace of $L^2([-\pi,\pi])$. That is, we endow it with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{T}_N([-\pi,\pi])}$ formed by restricting to $\mathcal{T}_N([-\pi,\pi])\times\mathcal{T}_N([-\pi,\pi])$ the inner product on $L^2([-\pi,\pi])$ defined by (3.11). It can be shown that in addition to (3.11), $\langle\cdot,\cdot\rangle_{\mathcal{T}_N([-\pi,\pi])}$ can be computed via the sum[17]

$$\langle f,g\rangle_{\mathcal{T}_N([-\pi,\pi])} = \frac{1}{K}\sum_{k=-N}^{N} f\left(k\frac{2\pi}{K}\right)\overline{g\left(k\frac{2\pi}{K}\right)}. \tag{3.20}$$

Moreover, for $f\in\mathcal{T}_N([-\pi,\pi])$, the expansion (3.12) takes the form

$$f(x) = \sum_{k=-N}^{N} \langle f, e^{ikx}\rangle_{\mathcal{T}_N([-\pi,\pi])}\, e^{ikx}. \tag{3.21}$$

The appropriate analogue of the Fourier transform (3.16) in this setting is

$$F(z) = \frac{1}{K}\sum_{k=-N}^{N} f\left(k\frac{2\pi}{K}\right) e^{-ik\frac{2\pi}{K}z} \tag{3.22}$$

for $f\in\mathcal{T}_N([-\pi,\pi])$. The transformed function F is a trigonometric polynomial of degree at most N but of period K instead of 2π. We denote the space of such trigonometric polynomials by $\mathcal{T}_N([-K/2,K/2])$ and regard it as a subspace of $L^2([-K/2,K/2])$, where the inner product on the latter is taken to be

$$\langle F,G\rangle_{L^2([-K/2,K/2])} = \int_{-K/2}^{K/2} F(x)\overline{G(x)}\,dx.$$

This definition for the inner product makes the map $f\mapsto F$ from $\mathcal{T}_N([-\pi,\pi])$ to $\mathcal{T}_N([-K/2,K/2])$ defined by (3.22) an isometry. Denoting the restriction of this inner product to $\mathcal{T}_N([-K/2,K/2])\times\mathcal{T}_N([-K/2,K/2])$ by $\langle\cdot,\cdot\rangle_{\mathcal{T}_N([-K/2,K/2])}$, we have, in analogy to (3.20),

$$\langle F,G\rangle_{\mathcal{T}_N([-K/2,K/2])} = \sum_{k=-N}^{N} F(k)\overline{G(k)}.$$

Taking the transform of e^{inx}, we have

$$\frac{1}{K}\sum_{k=-N}^{N} e^{ink\frac{2\pi}{K}} e^{-ik\frac{2\pi}{K}z} = \frac{\sin\bigl(\pi(n-z)\bigr)}{K\sin\left(\frac{\pi}{K}(n-z)\right)}.$$

Therefore, taking the transform of (3.21), we obtain

$$F(z) = \sum_{k=-N}^{N} F(k)\frac{\sin\bigl(\pi(z-k)\bigr)}{K\sin\left(\frac{\pi}{K}(z-k)\right)},$$

which is a discrete version of (3.18) and a Lagrange-type formula for $\mathcal{T}_N([-K/2,K/2])$; this result should be compared with the barycentric formula (1.27).

For easy comparison, these formulas are given alongside their infinite-dimensional counterparts in Figure 3.2.

[17] This sum is just the trapezoid rule in the points (3.1) (i.e., the midpoint rule), which is exact for all trigonometric polynomials of degree at most $2N$.

$$\begin{tikzcd}[column sep=large]
\begin{array}{c} f(x) = \displaystyle\sum_{k=-\infty}^{\infty} \left\langle f, e^{ikx} \right\rangle e^{ikx} \\ \text{Fourier series} \end{array} \arrow[rr, rightsquigarrow, "\begin{array}{c} F(z) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x) e^{-ixz}\, dx \\ \text{Fourier transform} \end{array}"'] & & \begin{array}{c} F(z) = \displaystyle\sum_{k=-\infty}^{\infty} F(k) \frac{\sin\bigl(\pi(z-k)\bigr)}{\pi(z-k)} \\ \text{cardinal series} \end{array} \\
\langle f, g \rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}\, dx \arrow[rr, rightsquigarrow] & & \langle F, G \rangle = \int_{-\infty}^{\infty} F(x)\overline{G(x)}\, dx \\
L^2([-\pi,\pi]) & \text{continuous} & \mathrm{PW}_\pi \\
\mathcal{T}_N([-\pi,\pi]) & \text{discrete} & \mathcal{T}_N([-K/2, K/2]) \\
\begin{array}{c} f(x) = \displaystyle\sum_{k=-N}^{N} \left\langle f, e^{ikx} \right\rangle e^{ikx} \\ \text{trigonometric polynomial} \\ \text{(2π-periodic, Fourier form)} \end{array} \arrow[rr, rightsquigarrow, "\begin{array}{c} F(z) = \frac{1}{K}\sum_{k=-N}^{N} f\left(k\frac{2\pi}{K}\right) e^{-ik\frac{2\pi}{K}z} \\ \text{discrete Fourier transform} \end{array}"'] & & \begin{array}{c} F(z) = \displaystyle\sum_{k=-N}^{N} F(k) \frac{\sin\bigl(\pi(z-k)\bigr)}{K\sin\left(\frac{\pi}{K}(z-k)\right)} \\ \text{trigonometric polynomial} \\ \text{(K-periodic, Lagrange form)} \end{array} \\
\langle f, g \rangle = \frac{1}{K}\sum_{k=-N}^{N} f\left(k\frac{2\pi}{K}\right)\overline{g\left(k\frac{2\pi}{K}\right)} \arrow[rr, rightsquigarrow] & & \langle F, G \rangle = \sum_{k=-N}^{N} F(k)\overline{G(k)}
\end{tikzcd}$$

Figure 3.2: Summary of the relationships between the problem of interpolation in PW_π (continuous/infinite-dimensional; see Section 3.2.4) and trigonometric interpolation (discrete/finite-dimensional). The subscripts on the inner products have been suppressed to reduce clutter.

3.3 Trigonometric Interpolation and Approximation on Perturbed Equispaced Grids

We now turn to our main objective: a study of the properties of trigonometric interpolation in the perturbed points (3.2). As mentioned in Section 3.1, our goal is to obtain an understanding of the behavior of the Lebesgue constant, which we will denote by $\widetilde{\Lambda}_K$, as the number of points K increases. In particular, we would like to find an upper bound for $\widetilde{\Lambda}_K$, as this will allow us to bound the interpolation error in the worst case.

We begin by investigating $\widetilde{\Lambda}_K$ numerically. Our work will lead to several conjectures for which the evidence is compelling (Conjectures 3.3, 3.4, and 3.5). Although we have not proved these statements, we are able to prove a slightly weaker result (Theorem 3.6) that suffices to establish rigorously the main characteristics of trigonometric interpolation in grids of the form (3.2).

Recall from Sections 1.2.8 and 1.3.3 that $\widetilde{\Lambda}_K$ is given by

$$\widetilde{\Lambda}_K = \sup_{x\in[-\pi,\pi]} \widetilde{L}(x), \tag{3.23}$$

where $\widetilde{L}(x)$ is the Lebesgue function

$$\widetilde{L}(x) = \sum_{k=-N}^{N} |\widetilde{\ell}_k(x)|, \tag{3.24}$$

and the $\widetilde{\ell}_k$ are the Lagrange basis functions for the perturbed grid defined by (1.25):

$$\widetilde{\ell}_k(x) = \prod_{\substack{j=-N \\ j\neq k}}^{N} \frac{\sin\left(\frac{x-\widetilde{x}_j}{2}\right)}{\sin\left(\frac{\widetilde{x}_k-\widetilde{x}_j}{2}\right)}. \tag{3.25}$$

We can approximate $\widetilde{\Lambda}_K$ for a particular choice of the points $\widetilde{x}_j$ easily by using Chebfun (see Section 1.5) to construct a high-accuracy numerical approximation to $\widetilde{L}$ and computing the maximum of this approximation. In fact, Chebfun contains a function `lebesgue` for doing just this.[18]

3.3.1 The "Worst" Grid

Initially, exploring this problem seems intimidating because of the large size of the parameter space. For $0 < \alpha < 1/2$, each point $\widetilde{x}_j$ can take on a continuum of possible values, and it is not at all obvious which configuration yields the largest value for $\widetilde{\Lambda}_K$. Our first task is to simplify things by trying to determine this value numerically.

We therefore perform the following simple experiment. Fix values for α and K. Assuming that $\widetilde{\Lambda}_K$ will be maximized for a configuration in which all points have been maximally perturbed (i.e., perturbed by an amount αh in one direction or the other), we compute $\widetilde{\Lambda}_K$ numerically using Chebfun for each such grid and see which yields the largest value. The validity of this assumption is not self-evident; however, numerical tests (the results of which we do not report here) strongly suggest that it is true. Because the number of grids grows exponentially in the number of points, such an exhaustive search is feasible only for very small K. Nevertheless, we can improve the situation a bit by taking advantage of symmetry[19] to reduce the number of cases we must consider.

The results of this experiment for $K = 3$, 5, 7, and 9 are presented in Table 3.1. The first column of the table shows the perturbation pattern of the points for the grid being considered; a `-1` means that the point was perturbed to the left by αh, and a `+1` means that it was perturbed to the right. The pattern `-1 -1 +1` corresponds to the grid of length $K = 3$ with $\widetilde{x}_{-1} = x_1 - \alpha h$, $\widetilde{x}_0 = x_0 - \alpha h$, and $\widetilde{x}_1 = x_1 + \alpha h$, and similarly for the rest. The remaining columns show the computed value of the Lebesgue constant for the indicated grid for seven different values of α between 0 and 1/2. Bold values denote the value of $\widetilde{\Lambda}_K$ that is largest for each combination of K and α.

Examining these results, we observe that one particular choice of grid consistently yields the largest value of $\widetilde{\Lambda}_K$ for all choices of K and α: the grid that shifts all points x_k with $k \leq 0$ to the right and all those with $k > 0$ to the left; see Figure 3.3 for an illustration. Running the experiment further for all odd values of K up to 17 and for $\alpha = 1/32, 2/32, \ldots, 15/32$, we observe

[18] Readers familiar with Chebfun may recall that the `lebesgue` function is usually used for computing Lebesgue functions and constants for polynomial interpolation. As of Chebfun v5.5.0, it can do so trigonometric interpolation as well.

[19] It is clear that given a choice of the points $\widetilde{x}_j$, the Lebesgue constant $\widetilde{\Lambda}_K$ does not change if the points are reflected across the origin or circularly shifted by a fixed amount.

Pattern	$\alpha = 1/16$	$\alpha = 1/8$	$\alpha = 3/16$	$\alpha = 1/4$	$\alpha = 5/16$	$\alpha = 3/8$	$\alpha = 7/16$
`+1 +1 +1`	1.66667	1.66667	1.66667	1.66667	1.66667	1.66667	1.66667
`+1 +1 -1`	**1.89848**	**2.21034**	**2.64922**	**3.30940**	**4.41122**	**6.61612**	**13.23192**
`+1 +1 +1 +1 +1`	1.98885	1.98885	1.98885	1.98885	1.98885	1.98885	1.98885
`+1 +1 +1 +1 -1`	2.30507	2.71576	3.27366	4.08501	5.39856	7.96187	15.51585
`+1 +1 +1 -1 -1`	**2.36286**	**2.87899**	**3.62286**	**4.76655**	**6.71269**	**10.67105**	**22.69486**
`+1 +1 -1 +1 -1`	2.31056	2.74335	3.35316	4.27223	5.80942	8.89183	18.15595
`+1 +1 +1 +1 +1 +1 +1`	2.20221	2.20221	2.20221	2.20221	2.20221	2.20221	2.20221
`+1 +1 +1 +1 +1 +1 -1`	2.57381	3.04668	3.67480	4.56728	5.98030	8.68510	16.54551
`+1 +1 +1 +1 +1 -1 -1`	2.65752	3.28219	4.17555	5.53656	7.82965	12.44838	26.36087
`+1 +1 +1 +1 -1 +1 -1`	2.60334	3.13796	3.88265	4.99082	6.82065	10.44669	21.24155
`+1 +1 +1 +1 -1 -1 -1`	**2.68500**	**3.36477**	**4.36361**	**5.92744**	**8.63280**	**14.21426**	**31.35700**
`+1 +1 +1 -1 +1 +1 -1`	2.57980	3.07703	3.76268	4.77482	6.43607	9.71468	19.45258
`+1 +1 +1 -1 +1 -1 -1`	2.66035	3.29742	4.22262	5.65547	8.10999	13.13118	28.45159
`+1 +1 +1 -1 -1 +1 -1`	2.60672	3.15539	3.93423	5.11569	7.10270	11.10484	23.17187
`+1 +1 -1 +1 +1 -1 -1`	2.63636	3.23361	4.09258	5.41136	7.65337	12.21095	26.05076
`+1 +1 -1 +1 -1 +1 -1`	2.58367	3.09651	3.81912	4.90855	6.73170	10.38981	21.39036
`+1 +1 +1 +1 +1 +1 +1 +1 +1`	2.36186	2.36186	2.36186	2.36186	2.36186	2.36186	2.36186
`+1 +1 +1 +1 +1 +1 +1 +1 -1`	2.77491	3.29361	3.97212	4.92039	6.39717	9.18265	17.19031
`+1 +1 +1 +1 +1 +1 +1 -1 -1`	2.87655	3.57775	4.57144	6.06898	8.56250	13.52639	28.32782
`+1 +1 +1 +1 +1 +1 -1 +1 -1`	2.81671	3.41728	4.24427	5.45905	7.43807	11.31004	22.71693
`+1 +1 +1 +1 +1 +1 -1 -1 -1`	2.91807	3.70286	4.85661	6.66134	9.77693	16.18615	35.81050
`+1 +1 +1 +1 +1 -1 +1 +1 -1`	2.79413	3.35740	4.12404	5.23926	7.04229	10.55111	20.85600
`+1 +1 +1 +1 +1 -1 +1 -1 -1`	2.89450	3.63689	4.71564	6.38467	9.23746	15.05612	32.75500
`+1 +1 +1 +1 +1 -1 -1 +1 -1`	2.83456	3.47497	4.38126	5.74998	8.03988	12.62722	26.39301
`+1 +1 +1 +1 +1 -1 -1 -1 -1`	**2.93448**	**3.75412**	**4.97804**	**6.92398**	**10.33865**	**17.47222**	**39.60129**
`+1 +1 +1 +1 -1 +1 +1 +1 -1`	2.78059	3.32246	4.05582	5.11814	6.83088	10.15919	19.93035
`+1 +1 +1 +1 -1 +1 +1 -1 -1`	2.88016	3.59739	4.63261	6.22455	8.93107	14.42708	31.09009
`+1 +1 +1 +1 -1 +1 -1 +1 -1`	2.82069	3.43806	4.30644	5.61103	7.78419	12.12316	25.11405
`+1 +1 +1 +1 -1 +1 -1 -1 -1`	2.91982	3.71264	4.88810	6.74433	9.98110	16.70541	37.47177
`+1 +1 +1 +1 -1 -1 +1 +1 -1`	2.84422	3.50252	4.44133	5.87042	8.28010	13.14268	27.82298
`+1 +1 +1 +1 -1 -1 +1 -1 -1`	2.89644	3.64756	4.74929	6.47152	9.44673	15.57729	34.38758
`+1 +1 +1 +1 -1 -1 -1 +1 -1`	2.83692	3.48742	4.41886	5.84295	8.25447	13.13928	27.93004
`+1 +1 +1 -1 +1 +1 +1 -1 -1`	2.86622	3.56004	4.55618	6.08107	8.66380	13.89300	29.71489
`+1 +1 +1 -1 +1 +1 -1 +1 -1`	2.80729	3.40354	4.23864	5.48900	7.56655	11.70748	24.09314
`+1 +1 +1 -1 +1 +1 -1 -1 -1`	2.90539	3.67244	4.80234	6.57575	9.65092	16.00858	35.56768
`+1 +1 +1 -1 +1 -1 +1 +1 -1`	2.78618	3.35066	4.13754	5.31169	7.25825	11.13321	22.71763
`+1 +1 +1 -1 +1 -1 +1 -1 -1`	2.88226	3.60876	4.66800	6.31467	9.14531	14.95344	32.71652
`+1 +1 +1 -1 +1 -1 -1 +1 -1`	2.82328	3.45153	4.34659	5.70901	8.00742	12.64883	26.67104
`+1 +1 +1 -1 -1 +1 +1 -1 -1`	2.86813	3.57026	4.58775	6.16074	8.85138	14.34931	31.11039
`+1 +1 +1 -1 -1 +1 -1 +1 -1`	2.80961	3.41553	4.27416	5.57505	7.76107	12.16182	25.42726
`+1 +1 +1 -1 -1 +1 -1 -1 -1`	2.90539	3.67244	4.80234	6.57575	9.65092	16.00858	35.56768
`+1 +1 -1 +1 +1 -1 +1 +1 -1`	2.76437	3.29441	4.02699	5.11261	6.90293	10.45351	21.04307
`+1 +1 -1 +1 +1 -1 +1 -1 -1`	2.85976	3.54848	4.54420	6.08041	8.70345	14.05523	30.35210
`+1 +1 -1 +1 +1 -1 -1 +1 -1`	2.80169	3.39602	4.23723	5.51050	7.64844	11.94959	24.90841
`+1 +1 -1 +1 -1 +1 +1 -1 -1`	2.84587	3.51128	4.46801	5.93677	8.43398	13.51113	28.93132
`+1 +1 -1 +1 -1 +1 -1 +1 -1`	2.78825	3.36123	4.16842	5.38542	7.42241	11.51058	23.80737

Table 3.1: Results of the experiment of Section 3.3.1 for identifying a likely candidate for the choice of points $\widetilde{x}_k$ that yields the largest value of $\widetilde{\Lambda}_K$ for a given K and α. The first column displays the perturbation pattern; a `-1` (respectively, `+1`) means that the point was perturbed to the left (respectively, right) by the maximum amount of αh. The other columns show the numerically computed value of $\widetilde{\Lambda}_K$ for several values of α between 0 and 1/2. Bold values indicate the largest value of $\widetilde{\Lambda}_K$ for a given choice of K and α. Observe that the grid (3.26) yields the largest value of $\widetilde{\Lambda}_K$ in all cases.

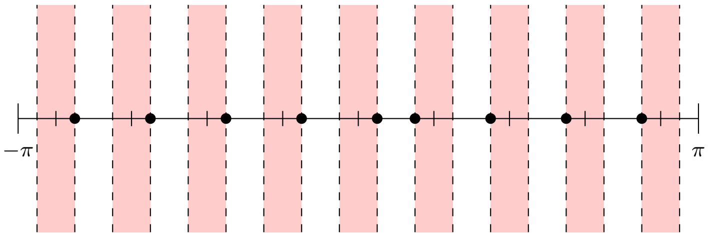


Figure 3.3: Illustration of the “worst” grid, a configuration of points that seems particularly likely to maximize the Lebesgue constant $\widetilde{\Lambda}_K$ over all choices of $\widetilde{x}_k$ for given values of α and K.

the same results. Moreover, we observe that this grid pattern is the unique one (up to the symmetries identified in the footnote) that gives this maximum.

On the basis of this evidence, we make the following conjecture:

Conjecture 3.3. *For given values of α and K, $0 < \alpha < 1/2$, the choice of the points $\widetilde{x}_k$ that maximizes $\widetilde{\Lambda}_K$ is*

$$\widetilde{x}_k = \begin{cases} x_k + \alpha h & -N \leq k \leq -1, \\ x_k - \alpha h & 0 \leq k \leq N, \end{cases} \tag{3.26}$$

uniquely, up to symmetry.

At present, we are unable to prove this statement; nevertheless, for convenience, we will refer to the grid defined by (3.26) as the “worst” grid from this point forward. We note that this grid is similar in structure to the grid used to construct Levinson’s example (3.14). In our setting, an appropriate analogue of Levinson’s example is the grid

$$\widetilde{x}_k = \begin{cases} x_k + \alpha h & -N \leq k \leq -1, \\ 0 & k = 0, \\ x_k - \alpha h & 1 \leq k \leq N, \end{cases} \tag{3.27}$$

which is the same as (3.26) but with the point at 0 left unperturbed.

3.3.2 Asymptotic Behavior of the Lebesgue Constant

With a good candidate for the grid that yields the largest value of $\widetilde{\Lambda}_K$ in hand, we next examine the asymptotic behavior of this hypothetically worst-case Lebesgue constant as $K \to \infty$. We are particularly interested in understanding how the asymptotic behavior of $\widetilde{\Lambda}_K$ varies with α, as this will tell us how the non-uniformity of the grid impacts our ability to use interpolants over it to approximate functions.

In order to see a clear trend, we will need to compute $\widetilde{\Lambda}_K$ for values of K up to at least several thousand. Unfortunately, the approach used in the computations of the last section based on Chebfun does not scale well to large values of K. We work around this by using the following empirical observation. Consider Figure 3.4, which plots $\widetilde{L}(x)$ for the “worst” grid (3.26) for a given value of K and several values of α. Notice that $\widetilde{L}$ assumes its maximum value on $[-\pi, \pi]$ at the endpoints

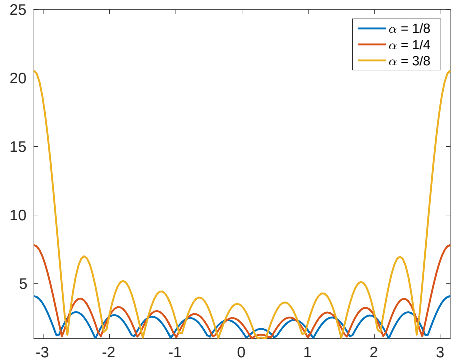


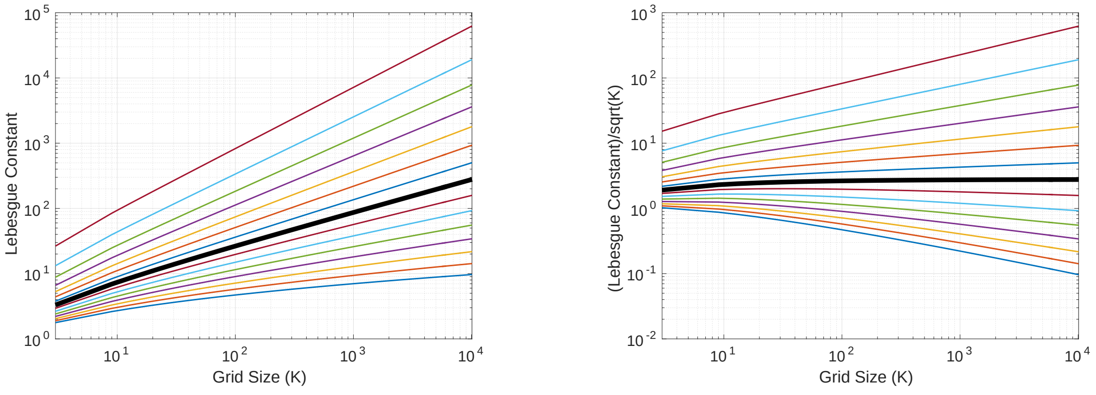
Figure 3.4: Plot of the Lebesgue function (3.24) on $[-\pi, \pi]$ for the grid (3.26) for $K = 11$ and $\alpha = 1/8$, $1/4$, and $3/8$. Note that the maximum value, the Lebesgue constant $\widetilde{\Lambda}_K$, is attained at the interval endpoints $\pm\pi$ in all three cases.

$\pm\pi$ in each of the three cases. Using Chebfun, we confirm numerically that this happens not just for the examples shown but for all odd values of K with $3 \leq K \leq 201$ and $\alpha = 1/64, 2/64, \ldots, 31/64$. From this, it is reasonable to guess that this happens for all choices of α and K, and we are thus led to make the following conjecture:

Conjecture 3.4. *For $0 < \alpha < 1/2$, and any odd $K \geq 1$, the Lebesgue function for the grid* (3.26) *assumes its maximum value on $[-\pi, \pi]$ at exactly the points $\pm\pi$.*

If this conjecture holds, then evaluating $\widetilde{\Lambda}_K$ for a particular instance of (3.26) is easy, since all we have to do is compute $\widetilde{L}(\pm\pi)$. Presently, we are not able to prove this statement; however, assuming that it is true, we obtain the plots of $\widetilde{\Lambda}_K$ for (3.26) shown in Figure 3.5. Looking at Figure 3.5a, we see that, for a fixed value of α, the Lebesgue constant for this grid appears to grow with K at a rate that is at most algebraic in K. Moreover, the asymptotic rate appears to increase as α gets closer to $1/2$. Estimating roughly by eye, it appears that the growth rate is approximately $O(K^{1/2})$ for $\alpha = 1/4$ (the thick black line) and that it approaches $O(K)$ as α approaches $1/2$. The trend for $\alpha = 1/4$ is confirmed in Figure 3.5b, which plots $\widetilde{\Lambda}_K/\sqrt{K}$ against K. Observe that on this plot, the line for $\alpha = 1/4$ appears to remain bounded as K becomes large.

We further observe that in Figure 3.5a, not only does the asymptotic growth rate vary with α but so does the spacing between the lines: for a fixed value of K, the amount of vertical space between the lines corresponding to successive values of α increases as α increases towards $1/2$. This trend is expected, since as α approaches $1/2$, the points $\widetilde{x}_0$ and $\widetilde{x}_1$ in (3.26) become arbitrarily close together, so $\widetilde{\Lambda}_K$ becomes unbounded. To quantify the rate at which this occurs, we plot $\widetilde{\Lambda}_K$ for (3.26) against $1/2 - \alpha$ for several fixed values of K. The results are shown in Figure 3.6, which makes it clear that $\widetilde{\Lambda}_K$ blows up at a rate of $O\big(1/(1/2 - \alpha)\big)$ as $\alpha \to 1/2$.



(a)

(b)

Figure 3.5: (a) Numerically computed values of $\widetilde{\Lambda}_K$ for the grid (3.26) as a function of K for several values of K between 3 and 10,103 and for $\alpha = 1/32, 2/32, \ldots, 15/32$. Lower lines correspond to lower values of α. The thick black line corresponds to $\alpha = 1/4$. (b) Same as (a) but plots $\widetilde{\Lambda}_K/\sqrt{K}$ against K. Notice that the line for $\widetilde{\Lambda}_K/\sqrt{K}$ for $\alpha = 1/4$ appears to remain bounded as K becomes large, suggesting that $\widetilde{\Lambda}_K = O(K^{1/2})$ for that value of α. These plots were made by evaluating $\widetilde{L}(-\pi)$ and thus are only valid assuming that Conjecture 3.4 holds. If that conjecture is false, the plots show lower bounds for $\widetilde{\Lambda}_K$ instead.

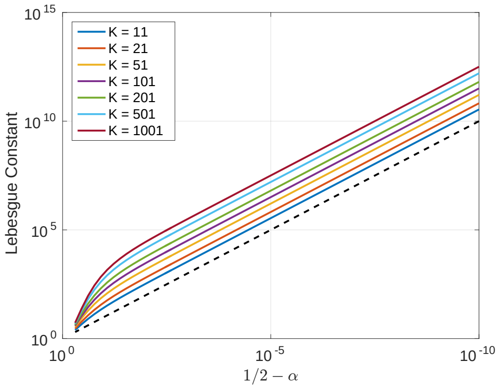


Figure 3.6: Numerically computed values of $\widetilde{\Lambda}_K$ for the grid (3.26) as a function of $1/2 - \alpha$ for several fixed values of K. Note that the orientation of the horizontal axis has been reversed. The dashed black line depicts $1/(1/2 - \alpha)$. As with the previous figure, these computations were performed assuming that Conjecture 3.4 is true. If it is false, the plots show lower bounds for $\widetilde{\Lambda}_K$ instead.

Taken together with Conjecture 3.3, this evidence leads us to make the following conjecture about the Lebesgue constant $\widetilde{\Lambda}_K$ for any grid of the general type (3.2) that we have been considering:

Conjecture 3.5. *There are absolute constants A and B such that for $0 < \alpha < 1/2$ and sufficiently large K, the Lebesgue constant $\widetilde{\Lambda}_K$ for any grid of the form (3.2) satisfies*

$$\widetilde{\Lambda}_K \leq \frac{1}{1-2\alpha}\left(A\frac{K^{2\alpha}-1}{\alpha}+B\right).$$

Thus,

$$\widetilde{\Lambda}_K = O\left(\frac{K^{2\alpha}-1}{\alpha(1-2\alpha)}\right)_{K\to\infty},$$

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$. When the $\widetilde{x}_k$ are chosen according to (3.26), this bound is attained in the sense that the big-O symbol cannot be replaced by a small-o symbol.

Note that this statement captures the hypothesized $O(K^{1/2})$ behavior for $\alpha = 1/4$ and that it also tends to $O(K)$ as $\alpha \to 1/2$. Moreover, the multiplying constant blows up as $\alpha \to 1/2$ in line with the trend observed in Figure 3.6. The reason for the subtracting 1 in the numerator and adding a factor of α in the denominator is that these adjustments cause the bound to exhibit the correct $O(\log K)$ behavior in the limit as $\alpha \to 0$, since

$$\lim_{\alpha\to 0}\frac{K^{2\alpha}-1}{\alpha} = \lim_{\alpha\to 0}\frac{2K^{2\alpha}\log K}{1} = 2\log K$$

by l'Hopital's rule.[20] Ignoring these considerations and treating α as fixed and nonzero (as we assume), we may of course write the bound expressed by the second statement in the conjecture as

$$\widetilde{\Lambda}_K = O\left(\frac{K^{2\alpha}}{\alpha(1-2\alpha)}\right)_{K\to\infty}$$

with the understanding that the reason the right-hand side blows up as $\alpha \to 0$ is because if it did not, then continuity would give $\widetilde{\Lambda}_K = O(1)$ as $K \to \infty$ as $\alpha \to 0$, contradicting the true limiting behavior of $O(\log K)$.

Unfortunately, we are not able to prove Conjecture 3.5 at this time. What we can prove is the following result, for which the exponent is off by only a factor of 2, and the dependence on α in the denominator is unchanged:

Theorem 3.6. *There are absolute constants A and B such that for $0 < \alpha < 1/2$ and sufficiently large K, the Lebesgue constant $\widetilde{\Lambda}_K$ for any grid of the form (3.2) satisfies*

$$\widetilde{\Lambda}_K \leq \frac{1}{1-2\alpha}\left(A\frac{K^{4\alpha}-1}{\alpha}+B\right).$$

Thus,

$$\widetilde{\Lambda}_K = O\left(\frac{K^{4\alpha}-1}{\alpha(1-2\alpha)}\right)_{K\to\infty},$$

[20] The author is indebted to Andrew Thompson for this observation.

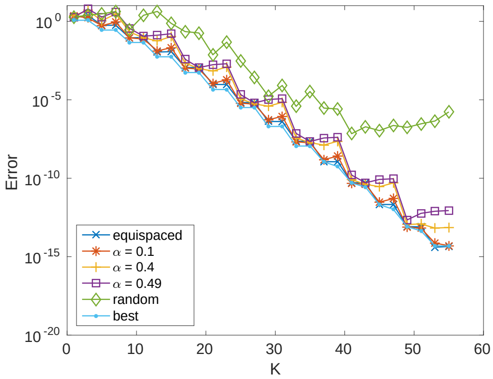


Figure 3.7: Numerically computed infinity-norm errors in the trigonometric interpolants to $f(x) = e^{\sin(2x)}$ in five different grids of length K for various values of K: equispaced points (blue crosses), the "worst" grid (3.26) with $\alpha = 0.1$ (orange stars), $\alpha = 0.4$ (gold pluses), $\alpha = 0.49$ (purple squares), and random points drawn from a uniform distribution on $[-\pi, \pi]$ (green diamonds). Also plotted are the errors in the best (trigonometric) approximations (light blue dots) of the corresponding degrees.

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$.

The proof of this theorem is the subject of the next chapter.

The important observation to make about the bounds on $\widetilde{\Lambda}_K$ given by Conjecture 3.5 and Theorem 3.6 is that the growth they predict is modest. We conclude that interpolation in these grids will converge (in exact arithmetic) for functions that are merely smooth, not holomorphic. Precise statements about the level of smoothness that is required are given in Conjecture 3.7 and Theorem 3.8 below.

Moreover, the interpolation scheme will be well-behaved in the presence of rounding errors, provided that α is not so close to $1/2$ that the constant in the bounds becomes a problem. This is illustrated in Figure 3.7, which examines the convergence as K increases of the trigonometric interpolants to the function $f(x) = e^{\sin(2x)}$ in five different grids in $[-\pi, \pi]$: equispaced points, the "worst" grid (3.26) for $\alpha = 0.1$, 0.4, and 0.49, and a grid of random points drawn from a uniform distribution on $[-\pi, \pi]$. In addition to the infinity-norm errors in the interpolants, we plot the infinity-norm error in the best approximation to f from $\mathcal{T}_N$ for each K. (Recall that the number of points K and the degree N of the interpolant are related by $K = 2N + 1$.) All approximations were computed numerically using Chebfun.

True to its name, the best approximation gives the lowest values for the error, but the interpolants in equispaced points and in (3.26) with $\alpha = 0.1$ are virtually indistinguishable from it. The interpolants for $\alpha = 0.4$ and $\alpha = 0.49$ are not much worse. The error in the interpolants in random points initially decays similarly to the others, albeit less steadily, but eventually bottoms out at around $K = 41$, after which it begins to increase. The random points are uniformly distributed, and

f is entire. In theory, the interpolants in these points should converge without trouble; however, the large Lebesgue constants (for the random grid of length $K = 41$ in the example, the Lebesgue constant is on the order of 10^8) prevent this convergence from taking place in practice.

3.3.3 Approximation Theorems

Having a bound on the Lebesgue constant for interpolation in grids of the form (3.2) enables us to make precise statements about the circumstances under which interpolation in a system of such grids will converge as the number of points increases. To the best of our knowledge, these are the first results to have been established for this problem.

When it comes to interpolation of holomorphic functions, Conjecture 3.5 and Theorem 3.6 do not tell us anything new about convergence, which is already assured for our perturbed equispaced grids by the Fejér–Kalmár–Walsh theorem (Theorem 1.12). What they do tell us is that this convergence is robust against rounding error, which is not true for more general uniformly distributed grids, as we have already remarked. The example considered at the end of the previous section attests to this robustness.

From a theoretical point of view, the value of Conjecture 3.5 and Theorem 3.6 is that they enable us to make assertions about the convergence of the interpolants for functions that are less smooth. This is accomplished by combining them with the trigonometric versions of Jackson's theorems (Theorem 1.2) and Theorem 1.13. In more detail, if f is a continuous periodic function on $[-\pi, \pi]$, and if $t_N \in \mathcal{T}_N$ is the trigonometric interpolant to f of degree N in the $K = 2N + 1$ points (3.2), then the trigonometric analogue of Theorem 1.13 tells us that

$$\|f - t_N\|_\infty \leq (1 + \widetilde{\Lambda}_K)\|f - t_N^*\|_\infty,$$

where t_N^* is the best approximation to f from $\mathcal{T}_N$. (Here, $\|\cdot\|_\infty$ denotes the supremum norm on the space of continuous periodic functions on $[-\pi, \pi]$.) Since we can bound $\|f - t_N^*\|_\infty$ under some additional assumptions on the regularity of f using the trigonometric versions of Jackson's theorems, a bound on $\widetilde{\Lambda}_K$ automatically implies a bound on $\|f - t_N\|_\infty$. If $\widetilde{\Lambda}_K$ grows more slowly as $K \to \infty$ than the rate at which $\|f - t_N^*\|_\infty$ decays to zero[21] then we will get uniform convergence of t_N to f as $K \to \infty$.

Assuming Conjecture 3.5 holds, the following result is an immediate consequence of the trigonometric version of Theorem 1.4:

Conjecture 3.7. *Let $0 < \alpha < 1/2$, and suppose that f is a 2π-periodic, ν-times continuously differentiable function, $\nu \geq 0$, such that $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent γ,*

[21] Note that $\|f - t_N^*\|$ decays to zero as $K \to \infty$ for any continuous f by the trigonometric version of the Weierstrass approximation theorem. (See Theorem 1.1 and the discussion of the analogies between trigonometric and polynomial approximation in Section 1.3.1. See also [69, p. 16] and [147].)

$2\alpha < \gamma \leq 1$. If t_N is the trigonometric interpolant to f of degree N in the $K = 2N+1$ perturbed points (3.2)*, then t_N converges to f uniformly on $[-\pi, \pi]$ at the rate*

$$\|f - t_N\|_\infty = O\left(\frac{K^{2\alpha} - 1}{\alpha(1-2\alpha)} K^{-\nu-\gamma}\right)_{K\to\infty},$$

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$.

Combining Theorems 3.6 and 1.4 instead, we obtain the following result, which is weaker but which we can prove rather than merely conjecture:

Theorem 3.8. *Let $0 < \alpha < 1/2$, and suppose that f is a 2π-periodic, ν times continuously differentiable function on $[-\pi, \pi]$, $\nu \geq 0$. Let t_N be the trigonometric interpolant to f of degree N in the $K = 2N+1$ perturbed points* (3.2)*. If either*

(i) $0 < \alpha < 1/4$ and $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent γ, $4\alpha < \gamma \leq 1$ or

(ii) $1/4 \leq \alpha < 1/2$, $\nu \geq 1$, and $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent $4\alpha - 1 < \gamma \leq 1$,

then t_N converges to f uniformly on $[-\pi, \pi]$ at the rate

$$\|f - t_N\|_\infty = O\left(\frac{K^{4\alpha} - 1}{\alpha(1-2\alpha)} K^{-\nu-\gamma}\right)_{K\to\infty},$$

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$.

3.4 2-Norm Lebesgue Constants and Kadec's 1/4 Theorem

Recalling a question we asked in Section 3.1, we observe that the experiments of the previous section suggest that, unlike the infinite-dimensional problem (see Section 3.2.4), there does not appear to be any sort of "1/4 threshold" for the problem of trigonometric interpolation. Indeed, nothing in the plots of Figure 3.5 indicates that $\alpha = 1/4$ (or any other value of α, $0 < \alpha < 1/2$, for that matter) occupies a distinguished position, nor do Conjecture 3.5 and Theorem 3.6 single it out as special.

Nevertheless, we cannot yet close the case on this matter because there is a gap between the analysis of the previous section and the theory presented in Section 3.2: they use different norms. The Lebesgue constant $\widetilde{\Lambda}_K$ measures the size of the interpolant using the uniform norm. This is the traditional norm to use from the viewpoint of approximation theory; however, the sampling theory presented in Section 3.2 makes all of its measurements using the 2-norm.

3.4.1 The 2-Norm Lebesgue Constant

In order to get a more direct comparison, we repeat our experiments of the previous section but instead measure the "2-norm Lebesgue constant" $\widetilde{\Lambda}_K^{(2)}$ defined for a given grid of the form (3.2) by

$$\widetilde{\Lambda}_K^{(2)} = \sup_{\|f\|_2 \leq 1} \|t_f\|_{L^2([-\pi,\pi])},$$

where $f = (f_{-N}, \ldots, f_N)$ is a vector in $\mathbb{C}^K$, and t_f is the unique trigonometric polynomial in $\mathcal{T}_N$ that satisfies $t_f(\widetilde{x}_k) = f_k$ for each k. The norm $\|\cdot\|_{L^2([-\pi,\pi])}$ is the one induced by the inner product (3.11) on $L^2([-\pi,\pi])$, while $\|\cdot\|_2$ denotes the norm on $\mathbb{C}^K$ induced by the normalized Euclidean inner product

$$\langle f, g \rangle_2 = \frac{1}{K} \sum_{k=-N}^{N} f_k \overline{g}_k. \tag{3.28}$$

Another way to look at $\widetilde{\Lambda}_K^{(2)}$ is that it is the lower frame bound A from (3.19) but for trigonometric interpolation instead of interpolation in PW_π.

To be able to calculate $\widetilde{\Lambda}_K^{(2)}$, we need a way to characterize it that is amenable to numerical computation. Remarkably, this is easy to do. If $\widetilde{\ell}_k$ denotes the k^{th} Lagrange basis function for our interpolation scheme, then for a given $f \in \mathbb{C}^K$,

$$\|t_f\|^2_{L^2([-\pi,\pi])} = \left\langle \sum_{k=-N}^{N} f_k \widetilde{\ell}_k, \sum_{k'=-N}^{N} f_{k'} \widetilde{\ell}_{k'} \right\rangle_{L^2([-\pi,\pi])} = \sum_{k,k'=-N}^{N} f_k \overline{f}_{k'} \left\langle \widetilde{\ell}_k, \widetilde{\ell}_{k'} \right\rangle_{L^2([-\pi,\pi])} = f^* G f,$$

where

$$G = \begin{bmatrix} \left\langle \widetilde{\ell}_{-N}, \widetilde{\ell}_{-N} \right\rangle_{L^2([-\pi,\pi])} & \cdots & \left\langle \widetilde{\ell}_{-N}, \widetilde{\ell}_{N} \right\rangle_{L^2([-\pi,\pi])} \\ \vdots & & \vdots \\ \left\langle \widetilde{\ell}_{N}, \widetilde{\ell}_{-N} \right\rangle_{L^2([-\pi,\pi])} & \cdots & \left\langle \widetilde{\ell}_{N}, \widetilde{\ell}_{N} \right\rangle_{L^2([-\pi,\pi])} \end{bmatrix},$$

the Gram matrix for $\widetilde{\ell}_{-N}, \ldots, \widetilde{\ell}_N$, viewed as members of $L^2([-\pi,\pi])$. Note that G is Hermitian, and since the $\widetilde{\ell}_k$ are linearly independent, it is also positive definite. Since

$$\sup_{\|f\|_2 \leq 1} f^* G f = K \lambda_{\max}(G),$$

where $\lambda_{\max}(G)$ is the largest eigenvalue of G,[22] we have

$$\widetilde{\Lambda}_K^{(2)} = \sqrt{K \lambda_{\max}(G)},$$

and this is easy to evaluate numerically. The inner products needed to form G can be computed simply and quickly using Chebfun.

[22] This follows from the variational characterization of eigenvalues using Rayleigh quotients. Note that the factor of K comes from the choice of normalization in (3.28).

3.4.2 The 2-Norm "Worst" Grid

Just as in Section 3.3.1, we begin our study by searching for the "worst" grid for a given K and α, this time in terms of which choice of perturbations yields the largest value of $\widetilde{\Lambda}_K^{(2)}$ instead of $\widetilde{\Lambda}_K$. Again, it seems reasonable to expect that the largest value will occur for a grid that has all of its points maximally perturbed. We therefore employ the same strategy of enumerating all such grids up to symmetry and computing $\widetilde{\Lambda}_K^{(2)}$ for each.

The results are given in Table 3.2, which is structured identically to Table 3.1; it presents the results of the search for $K = 3$, 5, 7, and 9 and for seven values of α between 0 and 1/2. We observe the interesting but perhaps not unexpected result that the same grid (3.26) that gave rise to the largest traditional Lebesgue constant gives rise to the largest 2-norm Lebesgue constant as well. Moreover, it appears to be the unique grid that maximizes $\widetilde{\Lambda}_K^{(2)}$, up to symmetry. Checking further for $K = 9, 11, \ldots, 17$ and $\alpha = 1/32, 2/32, \ldots 15/32$, we find that the same results holds in all cases. Thus, we conjecture:

Conjecture 3.9. *For given values of α and K, $0 < \alpha < 1/2$, the choice of the points $\widetilde{x}_k$ that maximizes $\widetilde{\Lambda}_K^{(2)}$ is given by* (3.26)*, uniquely, up to symmetry.*

As before, we cannot presently prove this statement; however, we will continue to refer to (3.26) as the "worst" grid in this context as well.

3.4.3 Asymptotic Behavior of the 2-Norm Lebesgue Constant

We now carry out similar experiments to those of Section 3.3.2 and study the behavior of $\widetilde{\Lambda}_K^{(2)}$ for the "worst" grid (3.26) as K becomes large for α fixed. The results are shown in Figure 3.8a. At first glance, this plot does not look substantially different from the one for $\widetilde{\Lambda}_K$ in Figure 3.5a. We see that $\widetilde{\Lambda}_K^{(2)}$ does not grow as quickly as $\widetilde{\Lambda}_K$ at least for smaller values of α, but otherwise, the two plots are qualitatively fairly similar. In particular, the growth of $\widetilde{\Lambda}_K^{(2)}$ also appears to be at most algebraic in K, the line for $\alpha = 1/4$ looks no more distinguished in Figure 3.8a than it does in Figure 3.5a.

Figure 3.8b, which plots $\widetilde{\Lambda}_K^{(2)}/\log K$ instead, shows that this apparent similarity is deceiving. This figure strongly suggests that if $\widetilde{\Lambda}_K^{(2)}$ does grow as $K \to \infty$ for $\alpha < 1/4$, then it does so more slowly than $\log K$. Looking at Figure 3.8a again, it seems likely that $\widetilde{\Lambda}_K^{(2)}$ may actually remain bounded as $K \to \infty$ for these values of α. For $1/4 < \alpha < 1/2$, Figure 3.8b shows that $\widetilde{\Lambda}_K^{(2)}$ grows more rapidly than $\log K$. This is in keeping with Figure 3.8a, which shows clear algebraic growth for these values of α. Estimating the rates by fitting least-squares lines to the last five data values of each suggests growth at a rate of approximately $O(K^{4\alpha-1})$ for α in this range. The trend for $\alpha = 1/4$ is less clear. The figure suggests the possibility that $\widetilde{\Lambda}_K^{(2)}$ grows more slowly than $\log(K)$ when $\alpha = 1/4$, but it is also possible that the asymptotic trend has not yet fully set in over the range

Pattern	$\alpha = 1/16$	$\alpha = 1/8$	$\alpha = 3/16$	$\alpha = 1/4$	$\alpha = 5/16$	$\alpha = 3/8$	$\alpha = 7/16$
+1 +1 +1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
+1 +1 -1	**1.12647**	**1.30533**	**1.56870**	**1.98168**	**2.69782**	**4.18154**	**8.76690**
+1 +1 +1 +1 +1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
+1 +1 +1 +1 -1	1.13228	1.32160	1.60364	2.05113	2.83646	4.48308	9.62844
+1 +1 +1 -1 -1	**1.14914**	**1.37095**	**1.71624**	**2.29301**	**3.36711**	**5.76887**	**13.74639**
+1 +1 -1 +1 -1	1.12815	1.31000	1.57860	2.00103	2.73567	4.26183	8.98963
+1 +1 +1 +1 +1 +1 +1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
+1 +1 +1 +1 +1 +1 -1	1.13386	1.32605	1.61326	2.07039	2.87523	4.56818	9.87408
+1 +1 +1 +1 +1 -1 -1	1.15411	1.38583	1.75119	2.37078	3.54527	6.22295	15.28286
+1 +1 +1 +1 -1 +1 -1	1.13584	1.33167	1.62555	2.09535	2.92634	4.68269	10.21248
+1 +1 +1 +1 -1 -1 -1	**1.16093**	**1.40677**	**1.80193**	**2.48847**	**3.82886**	**6.98788**	**18.02460**
+1 +1 +1 -1 +1 +1 -1	1.13097	1.31792	1.59568	2.03517	2.80431	4.41240	9.42406
+1 +1 +1 -1 +1 -1 -1	1.14649	1.36289	1.69694	2.24890	3.26301	5.49557	12.79738
+1 +1 +1 -1 -1 +1 -1	1.14649	1.36289	1.69694	2.24890	3.26301	5.49557	12.79738
+1 +1 -1 +1 +1 -1 -1	1.14430	1.35659	1.68296	2.22019	3.20372	5.36230	12.40551
+1 +1 -1 +1 -1 +1 -1	1.12867	1.31144	1.58165	2.00698	2.74728	4.28640	9.05752
+1 +1 +1 +1 +1 +1 +1 +1 +1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
+1 +1 +1 +1 +1 +1 +1 +1 -1	1.13450	1.32787	1.61721	2.07833	2.89125	4.60346	9.97622
+1 +1 +1 +1 +1 +1 +1 -1 -1	1.15602	1.39164	1.76498	2.40195	3.61799	6.41219	15.93727
+1 +1 +1 +1 +1 +1 -1 +1 -1	1.13829	1.33868	1.64097	2.12687	2.99133	4.82946	10.65027
+1 +1 +1 +1 +1 +1 -1 -1 -1	1.16497	1.41934	1.83295	2.56210	4.01153	7.49779	19.91994
+1 +1 +1 +1 +1 -1 +1 +1 -1	1.13450	1.32787	1.61721	2.07833	2.89125	4.60346	9.97622
+1 +1 +1 +1 +1 -1 +1 -1 -1	1.15181	1.37872	1.73365	2.32926	3.44321	5.94345	14.27336
+1 +1 +1 +1 +1 -1 -1 +1 -1	1.15181	1.37872	1.73365	2.32926	3.44321	5.94345	14.27336
+1 +1 +1 +1 +1 -1 -1 -1 -1	**1.16857**	**1.43072**	**1.86160**	**2.63203**	**4.19115**	**8.01901**	**21.93411**
+1 +1 +1 +1 -1 +1 +1 +1 -1	1.13250	1.32222	1.60497	2.05377	2.84173	4.49452	9.66105
+1 +1 +1 +1 -1 +1 +1 -1 -1	1.15032	1.37442	1.72410	2.30972	3.40323	5.85488	14.01684
+1 +1 +1 +1 -1 +1 -1 +1 -1	1.13717	1.33547	1.63387	2.11229	2.96108	4.76063	10.44319
+1 +1 +1 +1 -1 +1 -1 -1 -1	1.15726	1.39528	1.77320	2.41910	3.65367	6.49143	16.15980
+1 +1 +1 +1 -1 -1 +1 +1 -1	1.15032	1.37442	1.72410	2.30972	3.40323	5.85488	14.01684
+1 +1 +1 +1 -1 -1 +1 -1 -1	1.15582	1.39104	1.76359	2.39897	3.61178	6.39929	15.91081
+1 +1 +1 +1 -1 -1 -1 +1 -1	1.15726	1.39528	1.77320	2.41910	3.65367	6.49143	16.15980
+1 +1 +1 -1 +1 +1 +1 -1 -1	1.15029	1.37435	1.72412	2.31030	3.40601	5.86601	14.06810
+1 +1 +1 -1 +1 +1 -1 +1 -1	1.13282	1.32311	1.60686	2.05747	2.84896	4.50981	9.70326
+1 +1 +1 -1 +1 +1 -1 -1 -1	1.15613	1.39201	1.76599	2.40464	3.62554	6.43572	16.03363
+1 +1 +1 -1 +1 -1 +1 +1 -1	1.13282	1.32311	1.60686	2.05747	2.84896	4.50981	9.70326
+1 +1 +1 -1 +1 -1 +1 -1 -1	1.14558	1.36020	1.69062	2.23486	3.23098	5.41481	12.52904
+1 +1 +1 -1 +1 -1 -1 +1 -1	1.14292	1.35226	1.67200	2.19348	3.13662	5.17656	11.73422
+1 +1 +1 -1 -1 +1 +1 -1 -1	1.14975	1.37276	1.72042	2.30215	3.38760	5.81985	13.91443
+1 +1 +1 -1 -1 +1 -1 +1 -1	1.14558	1.36020	1.69062	2.23486	3.23098	5.41481	12.52904
+1 +1 +1 -1 -1 +1 -1 -1 -1	1.15613	1.39201	1.76599	2.40464	3.62554	6.43572	16.03363
+1 +1 -1 +1 +1 -1 +1 +1 -1	1.12647	1.30533	1.56870	1.98168	2.69782	4.18154	8.76690
+1 +1 -1 +1 +1 -1 +1 -1 -1	1.14194	1.34956	1.66648	2.18351	3.11983	5.14946	11.69084
+1 +1 -1 +1 +1 -1 -1 +1 -1	1.14194	1.34956	1.66648	2.18351	3.11983	5.14946	11.69084
+1 +1 -1 +1 -1 +1 +1 -1 -1	1.14327	1.35355	1.67597	2.20502	3.17001	5.27942	12.13588
+1 +1 -1 +1 -1 +1 -1 +1 -1	1.12889	1.31205	1.58295	2.00953	2.75225	4.29688	9.08646

Table 3.2: Results of the experiment of Section 3.4.2 for identifying a likely candidate for the choice of points $\widetilde{x}_k$ that yields the largest value of $\widetilde{\Lambda}_K^{(2)}$ for a given K and α. As in Table 3.1, the first column displays the perturbation pattern, and the other columns show the numerically computed value of $\widetilde{\Lambda}_K^{(2)}$ for several values of α between 0 and 1/2. The largest value of $\widetilde{\Lambda}_K^{(2)}$ for a given choice of K and α is printed in bold. We see that the same grid (3.26) that yielded the largest value of $\widetilde{\Lambda}_K$ also yields the largest value of $\widetilde{\Lambda}_K^{(2)}$ in all cases.

of K considered. Letting $\alpha \to 1/4$ in the $O(K^{4\alpha-1})$ behavior that we have observed for $\alpha > 1/4$, it seems likely that the behavior for $\alpha = 1/4$ is $O(1)$, i.e., $\widetilde{\Lambda}_K^{(2)}$ continues to remain bounded as $K \to \infty$ in this case.

The behavior of $\widetilde{\Lambda}_K^{(2)}$ as $\alpha \to 1/2$ for fixed K is illustrated in Figure 3.9. In stark contrast the behavior of $\widetilde{\Lambda}_K$ depicted in Figure 3.6, $\widetilde{\Lambda}_K^{(2)}$ appears to remain bounded as $\alpha \to 1/2$, though the limiting value (predictably) increases with K.

Together with Conjecture 3.9, these results lead us to make the following conjecture, which could be regarded as a discrete version of Kadec's 1/4 theorem (Theorem 3.1):

Conjecture 3.10. *The* 2*-norm Lebesgue constant* $\widetilde{\Lambda}_K^{(2)}$ *for any grid of the form* (3.2) *satisfies*

$$\widetilde{\Lambda}_K^{(2)} = \begin{cases} O(1) & 0 < \alpha \le 1/4 \\ O(K^{4\alpha-1}) & 1/4 < \alpha < 1/2 \end{cases}$$

as $K \to \infty$ with α fixed. The implied constants in the big-O symbols are independent both of α and of the choice of the perturbed points $\widetilde{x}_k$. This bound is attained when the $\widetilde{x}_k$ are chosen according to (3.26)*.*

That is, we conjecture that $\widetilde{\Lambda}_K^{(2)}$ remains bounded as $K \to \infty$ for all $\alpha \le 1/4$, while for $1/4 < \alpha < 1/2$, it may grow without bound. As with the other conjectures we have made, we are unable to prove this one at this time. The claim about which we feel least certain is the behavior for $\alpha = 1/4$. If $\widetilde{\Lambda}_K^{(2)}$ does not remain bounded as $K \to \infty$ for $\alpha = 1/4$, Figure 3.8b suggests strongly that it grows at a rate that is at most $O(\log K)$.

We emphasize that even though this conjecture predicts that $\widetilde{\Lambda}_K^{(2)}$ grows as $K \to \infty$ for $\alpha > 1/4$, the growth is modest. In the worst case when α is near 1/2, the rate is at most $O(K)$. This suggests that in spite of the emphasis on the importance of the 1/4 condition present in the sampling theory literature—which focuses mainly on the infinite-dimensional problem—its impact is negligible for the finite-dimensional approximations to the infinite-dimensional problem that get solved in practice. Together, Conjectures 3.5 and 3.10 (and Theorem 3.6) indicate that there is little to fear from taking $\alpha > 1/4$ in the finite-dimensional problem. The point to avoid, rather, is $\alpha = 1/2$.

3.5 Quadrature via Trigonometric Interpolation

We finish this chapter by revisiting Trefethen and Weideman's original problem of approximating the integrals of continuous periodic functions by integrating trigonometric interpolants. As we indicated in Section 3.1, Trefethen and Weideman noted that the integrals of the trigonometric interpolants to a periodic holomorphic function will converge in the absence of rounding error to the integral of the function at a geometric rate as long as the interpolation points are uniformly distributed, even if some of them coalesce.

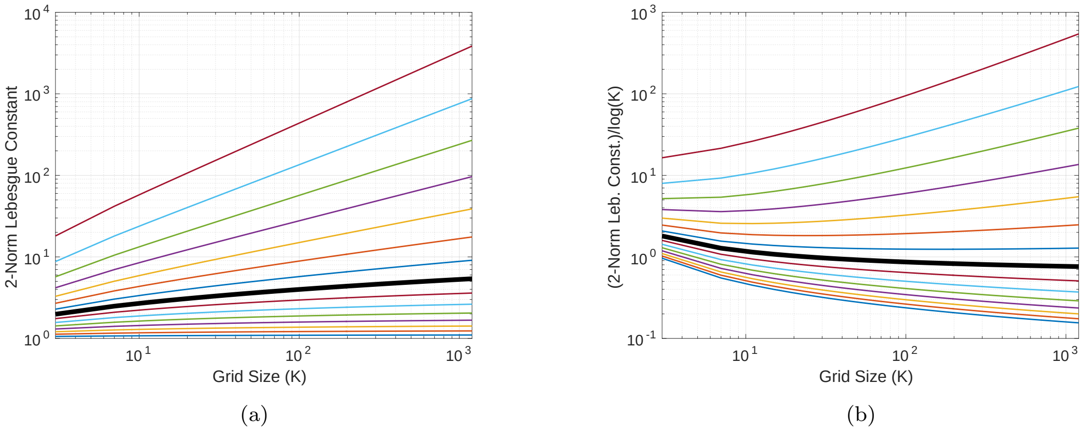


Figure 3.8: (a) Numerically computed values of $\widetilde{\Lambda}_K^{(2)}$ for the grid (3.26) as a function of K for several values of K between 3 and 1,201 and for $\alpha = 1/32, 2/32, \ldots, 15/32$. Lower lines correspond to lower values of α. The thick black line corresponds to $\alpha = 1/4$. (b) Same as (a) but plots $\widetilde{\Lambda}_K^{(2)}/\log(K)$ against K. The plots suggest that $\widetilde{\Lambda}_K^{(2)}$ remains bounded—or at least grows more slowly than $\log(K)$—as $K \to \infty$ when $\alpha < 1/4$ and grows at an algebraic rate that is at most $O(K)$ for $1/4 < \alpha < 1/2$ For $\alpha = 1/4$, the trend is not clear from the plot; we conjecture that $\widetilde{\Lambda}_K^{(2)}$ continues to remain bounded as $K \to \infty$ in this case.

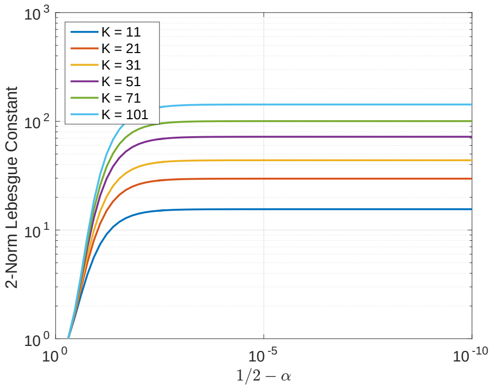


Figure 3.9: Numerically computed values of $\widetilde{\Lambda}_K^{(2)}$ for the grid (3.26) as a function of $1/2-\alpha$ for several fixed values of K. Note that the orientation of the horizontal axis has been reversed. Observe that $\widetilde{\Lambda}_K^{(2)}$ appears to remain bounded as $\alpha \to 1/2$ for any fixed K.

3.5.1 Results Derived from Approximation Theorems

This result is ultimately based on the trigonometric version of the Fejér–Kalmár–Walsh theorem (Theorem 1.12; see also the work of Hlawka [59] and Kis [70]) and is not numerically robust, nor does it apply to functions that are not holomorphic. Using the results from Section 3.3.3, we can state theorems for quadrature based on trigonometric interpolation in grids of the form (3.2) that both apply to non-holomorphic functions and are meaningful numerically. Denoting the integral of a function f from $-\pi$ to π by $I(f)$, all that is necessary is to relate the approximation error to the quadrature error using the elementary inequality

$$|I(f) - I(g)| \le I(|f - g|) \le 2\pi \|f - g\|_\infty.$$

From Conjecture 3.7, we obtain

Conjecture 3.11. *Let $0 < \alpha < 1/2$, and suppose that f is a 2π-periodic, ν-times continuously differentiable function, $\nu \ge 0$, such that $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent γ, $2\alpha < \gamma \le 1$. If t_N is the trigonometric interpolant to f of degree N in the $K = 2N + 1$ perturbed points (3.2), then $I(t_N)$ converges to $I(f)$ on $[-\pi, \pi]$ at the rate*

$$|I(f) - I(t_N)| = O\left(\frac{K^{2\alpha} - 1}{\alpha(1 - 2\alpha)} K^{-\nu-\gamma}\right)_{K\to\infty},$$

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$.

Similarly, Theorem 3.8 yields

Theorem 3.12. *Let $0 < \alpha < 1/2$, and suppose that f is a 2π-periodic, ν times continuously differentiable function on $[-\pi, \pi]$, $\nu \ge 0$. Let t_N be the trigonometric interpolant to f of degree N in the $K = 2N + 1$ perturbed points (3.2). If either*

(i) $0 < \alpha < 1/4$ and $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent γ, $4\alpha < \gamma \le 1$ or

(ii) $1/4 \le \alpha < 1/2$, $\nu \ge 1$, and $f^{(\nu)}$ is Hölder continuous on $[-\pi, \pi]$ with exponent $4\alpha - 1 < \gamma \le 1$,

then $I(t_N)$ converges to $I(f)$ on $[-\pi, \pi]$ at the rate

$$|I(f) - I(t_N)| = O\left(\frac{K^{4\alpha} - 1}{\alpha(1 - 2\alpha)} K^{-\nu-\gamma}\right)_{K\to\infty},$$

with the implied constant in the big-O symbol independent both of α and of the choice of the perturbed points $\widetilde{x}_k$.

3.5.2 Sums of Quadrature Weights

In this final section, we give some experimental evidence that, in fact, the situation may be even better than the theorems just stated suggest. A very general theorem due to Pólya asserts that a quadrature rule converges for all continuous functions on $[-1, 1]$ if and only if it converges for all polynomials and the sums of the absolute values of the quadrature weights remain bounded as the number of points increases [23, p. 130], [105]. Taking the trigonometric analogue, if we can prove that the sum of the absolute values of the weights for trigonometric interpolatory quadrature in grids of the form (3.2) remains bounded for large K, then we will have shown that such quadrature schemes are convergent for all continuous 2π-periodic functions without any further assumptions on the functions' regularity.

At the moment, we cannot prove any such statement; however, we can investigate the situation numerically. The quadrature weights for a given grid are easy to compute: the weight w_k at the kth node is just the integral of the corresponding Lagrange basis function (3.25). This can be computed via

$$w_k = \int_{-\pi}^{\pi} \widetilde{\ell}_k(x)\,dx = \frac{2\pi}{K}\sum_{j=-N}^{N} \widetilde{\ell}_k(x_j),$$

i.e., by computing the mean value of $\widetilde{\ell}_k$ on the equispaced grid (3.1) and multiplying by 2π.[23]

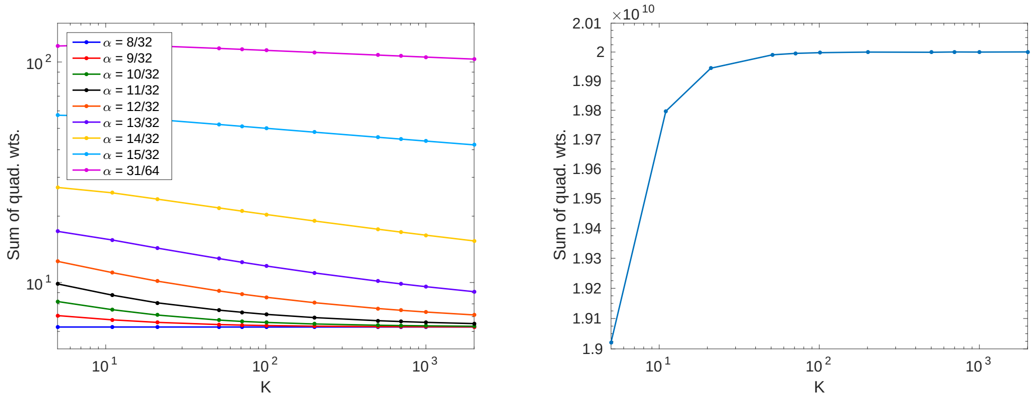
We begin by considering the sums of the absolute values of the weights for the "worst" grid (3.26). We compute these sums for the grids with $\alpha = 1/64, 2/64, \ldots, 31/64$ for several values of K up to $K = 2001$. For $\alpha \leq 1/4$, it turns out that the quadrature weights are all nonnegative and therefore sum to 2π for all values of K. For $1/4 < \alpha < 1/2$, this is no longer true, and Figure 3.10a displays how the sums vary with K for several values of α in this range. Not only do the sums appear to remain bounded as $K \to \infty$ for all α, $0 < \alpha < 1/2$, they actually appear to decay toward the value of 2π. Further testing shows that this behavior appears to continue to hold even when α is very close to $1/2$. Figure 3.10b shows the behavior of the sum of the quadrature weights for the same values of K as Figure 3.10a but with $\alpha = 1/2 - 10^{-10}$. Though the sum is very large (as expected), it appears to at least level off as K grows.

On the basis of this evidence, we make the following conjecture:

Conjecture 3.13. *The weights for quadrature by trigonometric interpolation in the nodes* (3.26) *are all nonnegative for all* K *for* $0 < \alpha \leq 1/4$. *For* $1/4 < \alpha < 1/2$, *they are no longer nonnegative, but the sum of their absolute values remains uniformly bounded as* $K \to \infty$.

For more general grids, the situation is much less clear. It is not true that the grid (3.26) leads to the largest sum of the absolute values of the quadrature weights for a given K and α. Exhaustively

[23] This computes the coefficient for the constant term in the representation of $\widetilde{\ell}_k$ with respect to the usual (Fourier) basis $\{e^{ijx}\}_{j=-N}^{N}$, which is the only term that does not integrate to zero.



(a) (b)

Figure 3.10: (a) Behavior of the sum of the absolute values of the quadrature weights for the "worst" grid (3.26) for several values of α as K becomes large (up to $K = 2001$). (b) Same but plotting for $\alpha = 1/2 - 10^{-10}$ only. Note in (a) that the sums appear to decay toward 2π as $K \to \infty$, while in (b), the sum levels off.

	$K = 7$		$K = 9$	
α	Sum	Pattern	Sum	Pattern
3/16	6.419300	-1 -1 +1 -1 +1 -1 +1	6.573631	-1 -1 +1 -1 +1 -1 +1 -1 +1
7/32	6.931198	-1 -1 +1 -1 +1 -1 +1	7.020432	-1 -1 +1 -1 +1 -1 +1 -1 +1
1/4	7.552545	-1 -1 +1 -1 +1 -1 +1	7.562472	-1 -1 +1 -1 +1 -1 +1 -1 +1
9/32	8.330386	-1 -1 +1 -1 +1 -1 +1	8.240611	-1 -1 +1 -1 +1 -1 +1 -1 +1
5/16	9.343019	-1 -1 +1 -1 +1 -1 +1	9.122849	-1 -1 +1 -1 +1 -1 +1 -1 +1
11/32	10.731242	-1 -1 +1 -1 +1 -1 +1	10.888812	-1 -1 +1 -1 +1 +1 -1 0 +1
3/8	13.356012	-1 -1 -1 +1 -1 +1 +1	13.565832	-1 -1 +1 -1 +1 +1 -1 0 +1
13/32	18.024649	-1 -1 -1 +1 -1 +1 +1	18.237170	-1 -1 +1 -1 -1 +1 -1 -1 +1 -1 -1 +1 -1 -1 +1 +1 -1 +1
7/16	27.324807	-1 -1 -1 +1 -1 +1 +1	27.806386	-1 -1 -1 -1 +1 -1 +1 +1 +1
15/32	57.506019	-1 -1 -1 -1 +1 +1 +1	57.296953	-1 -1 -1 -1 +1 -1 +1 +1 +1

Table 3.3: Perturbation patterns that yield the largest sum of the absolute values of the quadrature weights for $K = 7, 9$ and several values of α. As in Table 3.1, a -1 represents a point that was perturbed to the left by αh, and a +1 represents a points that was perturbed to the right. A 0 indicates a point that was left unperturbed.

enumerating the grids in the same way as in Section 3.3.1 and looking at the sums does not turn up a clear candidate for a "worst" grid in terms of quadrature weights.

Expanding our search to include grids that leave some of the points unperturbed, we find that it is not even true that the largest sum always occurs for a choice of the grid with all points maximally perturbed. Table 3.3 shows the perturbation patterns obtained from this expanded search that yield the largest value of the sum of the absolute values of the quadrature weights for $K = 7$ and 9 and several values of α. For $K = 9$, the grid that yielded the largest sum for $\alpha = 11/32$ and $\alpha = 3/8$ leaves one point unperturbed. For $K = 9$ and $\alpha = 13/32$, there are two inequivalent (in the sense of symmetry) grids that yield the same sum. The table also shows that, unlike (3.26), general grids can have negative values for some of the quadrature weights even for $\alpha \leq 1/4$.

This lack of a clear pattern makes it difficult to see how to proceed; however, we also observe

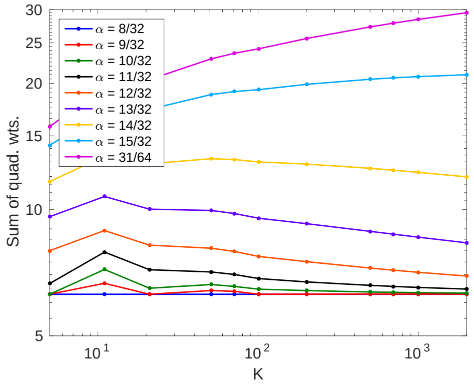


Figure 3.11: Same as Figure 3.10a but for the Levinson-type grid (3.27).

from Table 3.3 that the maximal values over the grids that we searched do not increase much as K increases from 7 to 9; indeed, for some values of α, we actually see a *decrease* in the maximal value from $K = 7$ to $K = 9$. Looking at some larger values of K, we find that this trend continues for $K = 11$, for which the largest sum we found is 59.119096, and for $K = 13$, for which the largest sum is 59.620563. This suggests that if the sums ultimately do grow without bound, then they do so at a very slow rate.

We can gain some additional insight by examining the sums of the absolute values of the quadrature weights for the Levinson-type grid (3.27), which are depicted in Figure 3.11. As was the case for the "worst" grid (3.26), the weights for $\alpha \leq 1/4$ turn out to be nonnegative for all K. We observe that for most values of α, $1/4 < \alpha < 1/2$, the sums appear to remain bounded and even decay toward 2π, just as was observed for (3.26). For $\alpha = 15/32$ and $\alpha = 31/64$, we appear to observe growth, but it is not clear if this is genuine or if we simply have not taken K large enough to determine the true trend. If the sums do grow for these values of α, they appear to do so at a sub-algebraic rate.

The outcomes of these experiments lead us to make the following conjecture:

Conjecture 3.14. *The sums of the absolute values of the weights for quadrature by trigonometric interpolation in any system of grids of the form* (3.2) *for a fixed value of* α, $0 < \alpha < 1/2$ *remain uniformly bounded as* $K \to \infty$.

We are not as confident about this conjecture as Conjecture 3.13; however, it seems likely to be true. Resolving this and Conjecture 3.13 is a topic for future research.

Chapter 4

Trigonometric Interpolation in Non-Equispaced Points II

The sole aim of this chapter is to provide a proof of the bound on the Lebesgue constant $\widetilde{\Lambda}_K$ for trigonometric interpolation in the perturbed equispaced points (3.2) that we claimed in Theorem 3.6. For ease of reference, we repeat our basic notational setup from Chapter 3 here. If $K = 2N + 1$ is an odd integer, the zero-centered equispaced grid of length K in $[-\pi, \pi]$ consists of the points

$$x_k = kh, \qquad -N \leq k \leq N, \tag{4.1}$$

where $h = 2\pi/K$ is the grid spacing. We consider the perturbed grid

$$\widetilde{x}_k = x_k + t_k h, \qquad |t_k| \leq \alpha, \tag{4.2}$$

where the parameter α is a fixed value in the range $0 \leq \alpha < 1/2$. The kth trigonometric Lagrange basis function associated with the perturbed grid is denoted by $\widetilde{\ell}_k$; we have $\widetilde{\ell}_k(x_j) = 1$ if $j = k$ and 0 if $j \neq k$. From (3.23) and (3.24), we have

$$\widetilde{\Lambda}_K = \sup_{x \in [-\pi,\pi]} \sum_{k=-N}^{N} |\widetilde{\ell}_k(x)|. \tag{4.3}$$

Our argument can be loosely outlined as follows. The bulk of the work is devoted to bounding $|\widetilde{\ell}_0(x)|$, which takes several steps to accomplish. Taking x as fixed, we determine the choice of the points $\widetilde{x}_j$ that maximizes $|\widetilde{\ell}_0(x)|$ and then bound the maximum using integrals. Since the resulting bound is independent of $\widetilde{x}_j$, we can exploit symmetry to obtain bounds on $|\widetilde{\ell}_k(x)|$ for $k \neq 0$. We then sum these bounds over k to obtain a bound on $\widetilde{\Lambda}_K$.

We begin with the following result, which shows that to bound $|\widetilde{\ell}_0(x)|$ we need only consider grids in which all the points, possibly aside from $\widetilde{x}_0$, are perturbed by the maximum amount of αh.

Lemma 4.1. *For all $x \in [-\pi, \pi]$ and $-N \leq j \leq N$, $j \neq 0$,*

$$\left| \frac{\sin\left(\frac{x-\widetilde{x}_j}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_j}{2}\right)} \right| \leq \max\left(\left| \frac{\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j-\alpha)h}{2}\right)} \right|, \left| \frac{\sin\left(\frac{x-(j+\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j+\alpha)h}{2}\right)} \right| \right). \tag{4.4}$$

Proof. The statement is trivially true if $x = \widetilde{x}_0$. If $x \neq \widetilde{x}_0$, then from

$$\frac{d}{dt}\frac{\sin\left(\frac{x-t}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-t}{2}\right)} = \frac{1}{2}\frac{\sin\left(\frac{x-\widetilde{x}_0}{2}\right)}{\left[\sin\left(\frac{\widetilde{x}_0-t}{2}\right)\right]^2},$$

we see that $t \mapsto \sin\big((x-t)/2\big)/\sin\big((\widetilde{x}_0-t)/2\big)$ has no critical points in $[-\pi,\pi]$ apart from $t = \widetilde{x}_0$, where it is singular. In particular, it has no critical points in any of the intervals $[(j-\alpha)h, (j+\alpha)h]$ for $-N \le j \le N$, $j \neq 0$ and therefore must assume its extreme values on these intervals at the endpoints. Since $\widetilde{x}_j \in [(j-\alpha)h, (j+\alpha)h]$ for each j, we are done. □

Which of the two arguments to the maximum function on the right-hand side of (4.4) is larger depends on both x and j. We need to understand the exact conditions under which each one takes over. Our first step in this direction is the following lemma, which tells us when the two are equal.

Lemma 4.2. *For $0 < \alpha < 1/2$ and $-N \le j \le N$, $j \neq 0$, the equation*

$$\left|\frac{\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j-\alpha)h}{2}\right)}\right| = \left|\frac{\sin\left(\frac{x-(j+\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j+\alpha)h}{2}\right)}\right| \tag{4.5}$$

has exactly two solutions in $[-\pi,\pi]$: $x = \widetilde{x}_0$ and $x = x_j^*$, where[1]

$$x_j^* = 2\arctan\left(\frac{\cos(jh) - \cos(\alpha h) + \tan\big(\widetilde{x}_0/2\big)\sin(jh)}{\tan\big(\widetilde{x}_0/2\big)\big(\cos(jh)+\cos(\alpha h)\big) - \sin(jh)}\right).$$

Proof. Multiplying through by the denominators of both sides and applying some trigonometric identities, we find that (4.5) can be reduced to

$$\left|\cos\left(\frac{\widetilde{x}_0 - x}{2} + \alpha h\right) - \cos\left(\frac{\widetilde{x}_0 + x}{2} - jh\right)\right| = \left|\cos\left(\frac{\widetilde{x}_0 - x}{2} - \alpha h\right) - \cos\left(\frac{\widetilde{x}_0 + x}{2} - jh\right)\right|. \tag{4.6}$$

If the expressions within the absolute value signs on either side of (4.6) are equal, then we have

$$\cos\left(\frac{\widetilde{x}_0 - x}{2} + \alpha h\right) = \cos\left(\frac{\widetilde{x}_0 - x}{2} - \alpha h\right).$$

To solve this equation, there are two cases to consider.

<u>*Case 1*: $(\widetilde{x}_0 - x)/2 + \alpha h = (\widetilde{x}_0 - x)/2 - \alpha h + 2n\pi$ for some integer n.</u> Rearranging gives $\alpha h = n\pi$, and substituting for h, we arrive at $\alpha = Kn/2$. Since $\alpha < 1/2$, this can only hold if $n = 0$, in which case $\alpha = 0$, but this is disallowed by our hypotheses.

<u>*Case 2*: $(\widetilde{x}_0 - x)/2 + \alpha h = \alpha h - (\widetilde{x}_0 - x)/2 + 2n\pi$ for some integer n.</u> If this holds, then $\widetilde{x}_0 - x = 4n\pi$, but this can only happen if $n = 0$, since $\widetilde{x}_0 - x \in [-\pi - \alpha h, \pi + \alpha h]$, and this interval is contained in $[-2\pi, 2\pi]$ because $\alpha h \le \pi$. Thus, $x = \widetilde{x}_0$.

We conclude that $x = \widetilde{x}_0$ is the only solution when the expressions within the absolute value signs on either side of (4.6) are equal. On the other hand, if they are equal but of opposite sign, we get

$$2\cos\left(\frac{\widetilde{x}_0 + x}{2} - jh\right) = \cos\left(\frac{\widetilde{x}_0 - x}{2} + \alpha h\right) + \cos\left(\frac{\widetilde{x}_0 - x}{2} - \alpha h\right).$$

[1] Here and throughout this thesis, arctan denotes the principal branch of the inverse tangent function.

Simplifying the right-hand side to $2\cos\bigl((\widetilde{x}_0 - x)/2\bigr)\cos(\alpha h)$ and then expanding both sides out completely using trigonometric identities, we find that

$$\cos\left(\frac{\widetilde{x}_0}{2}\right)\cos\left(\frac{x}{2}\right)\cos(jh) - \sin\left(\frac{\widetilde{x}_0}{2}\right)\sin\left(\frac{x}{2}\right)\cos(jh) + \sin\left(\frac{\widetilde{x}_0}{2}\right)\cos\left(\frac{x}{2}\right)\sin(jh)$$
$$+ \cos\left(\frac{\widetilde{x}_0}{2}\right)\sin\left(\frac{x}{2}\right)\sin(jh) = \cos\left(\frac{\widetilde{x}_0}{2}\right)\cos\left(\frac{x}{2}\right)\cos(\alpha h) + \sin\left(\frac{\widetilde{x}_0}{2}\right)\sin\left(\frac{x}{2}\right)\cos(\alpha h).$$

Dividing both sides of this through by $\cos(\widetilde{x}_0/2)\cos(x/2)$ and rearranging, we obtain

$$\tan\left(\frac{x}{2}\right) = \frac{\cos(jh) - \cos(\alpha h) + \tan\bigl(\widetilde{x}_0/2\bigr)\sin(jh)}{\tan\bigl(\widetilde{x}_0/2\bigr)\bigl(\cos(jh) + \cos(\alpha h)\bigr) - \sin(jh)}.$$

Taking the inverse tangent of both sides and multiplying by 2, we arrive at $x = x_j^*$. □

To move forward, we need a better understanding of the locations of the points x_j^*. The requisite inequalities are simple to state and are given in Lemma 4.4, but first we pause to establish a minor fact that we will need in their proof.

Lemma 4.3. *For $|t| \le \alpha$ and $-N \le j \le N$, $j \ne 0$,*

$$\bigl|\sin\bigl((j+t)h\bigr)\bigr| > \sin\bigl((\alpha - |t|)h\bigr).$$

Proof. This is a consequence of the following chain of inequalities:

$$0 \le (\alpha - |t|)h < (1 - |t|)h \le (|j| - |t|)h$$
$$\le (|j| + |t|)h \le (N + |t|)h < \left(N + |t| + \frac{1}{2} - \alpha\right)h = \pi - (\alpha - |t|)h \le \pi.$$

□

Lemma 4.4. *For $-N \le j \le N$, $j \ne 0$, and $0 < \alpha < 1/2$,*

$$(j - \alpha)h < x_j^* < (j + \alpha)h.$$

Proof. Let

$$f(t) = \frac{\cos(jh) - \cos(\alpha h) + t\sin(jh)}{t\bigl(\cos(jh) + \cos(\alpha h)\bigr) - \sin(jh)}.$$

Note that $f\bigl(\tan(\widetilde{x}_0/2)\bigr) = \tan(x_j^*/2)$. A straightforward computation using the quotient rule and some trigonometric identities shows that

$$f'(t) = -\left(\frac{\sin(\alpha h)}{t\bigl(\cos(jh) + \cos(\alpha h)\bigr) - \sin(jh)}\right)^2,$$

which is always negative wherever it is defined. By Lemma 4.3, we have $|\sin(jh)| > \sin(\alpha h)$ for each j. Furthermore, note that $j \ne 0$ implies $N \ge 1$, so that $\alpha h < \pi/3$, and so $\cos(\alpha h) > 0$. Therefore, $|\cos(jh) + \cos(\alpha h)| \le 1 + \cos(\alpha h)$ for each j. It follows that

$$\left|\frac{\sin(jh)}{\cos(jh) + \cos(\alpha h)}\right| > \frac{\sin(\alpha h)}{1 + \cos(\alpha h)} = \tan\left(\frac{\alpha h}{2}\right).$$

Hence, the singularity in f is outside the interval $[\tan(-\alpha h/2), \tan(\alpha h/2)]$, and we conclude that

$$f\big(\tan(\alpha h/2)\big) \leq f\big(\tan(\widetilde{x}_0/2)\big) \leq f\big(\tan(-\alpha h/2)\big).$$

Next, consider the function g_+ and the number M_+ defined by

$$g_+(t) = \frac{\cos(jh) - t + \tan(-\alpha h/2)\sin(jh)}{\tan(-\alpha h/2)\big(\cos(jh) + t\big) - \sin(jh)}, \qquad M_+ = \frac{\cos(jh) - 1 + \tan(-\alpha h/2)\sin(jh)}{\tan(-\alpha h/2)\big(\cos(jh) - 1\big) - \sin(jh)}.$$

Note that $g_+\big(\cos(\alpha h)\big) = f\big(\tan(-\alpha h/2)\big)$ and that

$$M_+ = \frac{\frac{\cos(jh)-1}{\sin(jh)} + \tan(-\alpha h/2)}{\tan(-\alpha h/2)\frac{\cos(jh)-1}{\sin(jh)} - 1} = \frac{\tan(jh/2) - \tan(-\alpha h/2)}{1 + \tan(jh/2)\tan(-\alpha h/2)} = \tan\left(\frac{(j+\alpha)h}{2}\right).$$

Therefore, if we can show that $g_+\big(\cos(\alpha h)\big) < M_+$, we will have shown that $\tan(x_j^*/2) < \tan\big((j+\alpha)h/2\big)$, which implies that $x_j^* < (j+\alpha)h$, as desired. The remainder of the proof will be devoted to establishing this fact. The lower bound on x_j^* can be derived by considering the function g_- and the number M_- defined by

$$g_-(t) = \frac{\cos(jh) - t + \tan(\alpha h/2)\sin(jh)}{\tan(\alpha h/2)\big(\cos(jh) + t\big) - \sin(jh)}, \qquad M_- = \frac{\cos(jh) - 1 + \tan(\alpha h/2)\sin(jh)}{\tan(\alpha h/2)\big(\cos(jh) - 1\big) - \sin(jh)}$$

and arguing similarly. We omit the details.

To show that $g_+\big(\cos(\alpha h)\big) < M_+$, we begin by noting that by multiplying the numerator and denominator of both $g_+(t)$ and M_+ by $\cos(-\alpha h/2)$ and applying some trigonometric identities, they can be rewritten as

$$g_+(t) = \frac{\cos(\alpha h/2)t - \cos\big((j+\alpha/2)h\big)}{\sin(\alpha h/2)t + \sin\big((j+\alpha/2)h\big)}, \qquad M_+ = -\frac{\cos(\alpha h/2) - \cos\big((j+\alpha/2)h\big)}{\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big)}.$$

Consider the affine function φ obtained by multiplying together the denominators in these new expressions for g_+ and M_+, where that of the latter is taken to include the leading minus sign:

$$\begin{aligned}\varphi(t) = -\sin(\alpha h/2)\Big(\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big)\Big)t \\ - \sin\big((j+\alpha/2)h\big)\Big(\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big)\Big).\end{aligned}$$

We will show that $\varphi\big(\cos(\alpha h)\big) > 0$. First, note that $\varphi(t) = 0$ at $t = t_0 = -\sin\big((j+\alpha/2)h\big)/\sin(\alpha h/2)$ and that by Lemma 4.3, this point lies outside of the interval $[-1, 1]$. Next, observe that $\sin(\alpha h/2) > 0$, that $\sin\big((j+\alpha/2)h\big)$ has the same sign as j, and that

$$\varphi'(t) = -\sin(\alpha h/2)\Big(\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big)\Big).$$

If $j < 0$, then $\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big) > 0$ trivially, so $\varphi'(t) < 0$. Thus, $\varphi(t) > 0$ for $t < t_0$. Inspecting the formula for t_0, we find that $t_0 > 0$ in this case. Since, t_0 cannot lie in the interval $[-1, 1]$, it must further be true that $t_0 > 1$. As $\cos(\alpha h) \leq 1$, we have that $\cos(\alpha h) < t_0$, as desired. On the other hand, if $j > 0$, then $\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big) < 0$ by Lemma 4.3, and we have that

$\varphi'(t) > 0$, so that $\varphi(t) > 0$ for $t > t_0$. But $t_0 < 0$ in this case, and since $\cos(\alpha h) > 0$, we have $\cos(\alpha h) > t_0$, and we are done.

It follows that $g_+\big(\cos(\alpha h)\big) < M_+$ is equivalent to the inequality

$$\begin{aligned}-\Big(\cos(\alpha h/2)\cos(\alpha h) - \cos\big((j+\alpha/2)h\big)\Big)\Big(\sin(\alpha h/2) - \sin\big((j+\alpha/2)h\big)\Big)&\\ < \Big(\sin(\alpha h/2)\cos(\alpha h) + \sin\big((j+\alpha/2)h\big)\Big)\Big(\cos(\alpha h/2) - \cos\big((j+\alpha/2)h\big)\Big).&\end{aligned}$$

Expanding out the products, moving all terms involving $\cos(\alpha h)$ to the left and those not involving it to the right, and using some trigonometric identities to simplify the result, we find that this in turn is equivalent to

$$\Big(\sin\big((j+\alpha)h\big) - \sin(\alpha h)\Big)\cos(\alpha h) < \sin(jh).$$

Next, we expand $\sin\big((j+\alpha)h\big)$ and move all terms involving $\sin(jh)$ to the right, leaving us with

$$\big(\cos(jh) - 1\big)\sin(\alpha h)\cos(\alpha h) < \sin(jh)\Big(1 - \big(\cos(\alpha h)\big)^2\Big).$$

Finally, using the identities $1 - \cos(jh) = \sin(jh)\tan(jh/2)$ and $1 - \big(\cos(\alpha h)\big)^2 = \sin\big(\alpha h\big)^2$, we can rearrange this one more time to find that our original inequality is equivalent to

$$\sin(jh)\big(\tan(\alpha h) + \tan(jh/2)\big) > 0.$$

If $j > 0$, then since $\sin(jh) > 0$, this is equivalent to $-\tan(jh/2) < \tan(\alpha h)$, which holds trivially, as the left-hand side is negative, while the right-hand side is positive. If $j < 0$, then $\sin(jh) < 0$, and the inequality is equivalent to $-\tan(jh/2) > \tan(\alpha h)$. Taking inverse tangents, we see that this is equivalent to $\alpha < -j/2$, and this inequality holds, since $-j \geq 1$ and $\alpha < 1/2$. This completes the proof. □

Assembling these results, we can prove the following statement about the right-hand side of (4.4).

Lemma 4.5. *We have*

$$\max\left(\left|\frac{\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j-\alpha)h}{2}\right)}\right|, \left|\frac{\sin\left(\frac{x-(j+\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j+\alpha)h}{2}\right)}\right|\right) = \begin{cases} \left|\dfrac{\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j-\alpha)h}{2}\right)}\right| & \begin{array}{l} x \in [-\pi, \widetilde{x}_0] \cup [x_j^*, \pi] \\ \quad \textit{for } 1 \leq j \leq N \\ x \in [x_j^*, \widetilde{x}_0] \\ \quad \textit{for } -N \leq j \leq -1 \end{array} \\[2ex] \left|\dfrac{\sin\left(\frac{x-(j+\alpha)h}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-(j+\alpha)h}{2}\right)}\right| & \begin{array}{l} x \in [\widetilde{x}_0, x_j^*] \\ \quad \textit{for } 1 \leq j \leq N \\ x \in [-\pi, x_j^*] \cup [\widetilde{x}_0, \pi] \\ \quad \textit{for } -N \leq j \leq -1. \end{array} \end{cases}$$

Proof. We will give the proof assuming $1 \leq j \leq N$; the proof for $-N \leq j \leq -1$ is similar. When $\alpha = 0$, there is nothing to prove, so we may assume $\alpha > 0$. By Lemma 4.2, the two arguments of the maximum function are equal only at $x = \widetilde{x}_0$ and $x = x_j^*$, and by Lemma 4.4, we have

$-\pi < \widetilde{x}_0 < (j-\alpha)h < x_j^* < (j+\alpha)h < \pi$. Evaluating both arguments of the maximum function at $x = (j-\alpha)h$, we see that the first is zero, while the second is nonzero. Thus, the second must be the larger on $[\widetilde{x}_0, x_j^*]$. Evaluating at $x = (j+\alpha)h$, the situation is reversed, and by periodicity, we find that the first must be the larger on $[-\pi, \widetilde{x}_0] \cup [x_j^*, \pi]$. □

This lemma is all we need for maximizing the factors in $|\widetilde{\ell}_0(x)|$ with respect to the $\widetilde{x}_j$ for $j \neq 0$. We would like to do something similar for $\widetilde{x}_0$. Unfortunately, the dependence on $\widetilde{x}_0$ of the various cases in this result tells us that we cannot go further and maximize any one factor over $\widetilde{x}_0$ independently of x. The next result shows that we can get around this by pairing up the factors at $\pm j$ for $1 \le j \le N$ instead of considering them in isolation.

Note that we state the result only for $x \in [-\pi, 0]$. This is because, by symmetry, any bound we obtain on $|\widetilde{\ell}_0(x)|$ for $x \in [-\pi, 0]$ that is independent of x must also hold for $x \in [0, \pi]$. We will therefore ignore the case of $x \in [0, \pi]$ until we reach the end of our argument, at which point we will see that it has been taken care of for free. Alternatively, one could write out an analogous argument that assumes $x \in [0, \pi]$ instead.

Lemma 4.6. *For $x \in [-\pi, 0]$ and $1 \le j \le N$,*

$$\left|\frac{\sin\left(\frac{x-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{x-\widetilde{x}_j}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_j}{2}\right)}\right| \le \begin{cases} \left|\dfrac{\sin\left(\frac{x+(j-\alpha)h}{2}\right)\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{jh}{2}\right)\sin\left(\frac{(2\alpha-j)h}{2}\right)}\right| & -\pi \le x \le x_{-j}^* \\ \left|\dfrac{\sin\left(\frac{x+(j+\alpha)h}{2}\right)\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{(2\alpha+j)h}{2}\right)\sin\left(\frac{(2\alpha-j)h}{2}\right)}\right| & x_{-j}^* \le x \le 0. \end{cases}$$

Proof. Fix x, and define the functions f_1, f_2, and f_3 by

$$f_1(t) = \frac{\sin\left(\frac{x+(j-\alpha)h}{2}\right)\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{t+(j-\alpha)h}{2}\right)\sin\left(\frac{t-(j-\alpha)h}{2}\right)} = \frac{\cos\bigl((j-\alpha)h\bigr)-\cos(x)}{\cos\bigl((j-\alpha)h\bigr)-\cos(t)}$$

$$f_2(t) = \frac{\sin\left(\frac{x+(j+\alpha)h}{2}\right)\sin\left(\frac{x-(j-\alpha)h}{2}\right)}{\sin\left(\frac{t+(j+\alpha)h}{2}\right)\sin\left(\frac{t-(j-\alpha)h}{2}\right)} = \frac{\cos(jh)-\cos(x+\alpha h)}{\cos(jh)-\cos(t+\alpha h)}$$

$$f_3(t) = \frac{\sin\left(\frac{x+(j-\alpha)h}{2}\right)\sin\left(\frac{x-(j+\alpha)h}{2}\right)}{\sin\left(\frac{t+(j-\alpha)h}{2}\right)\sin\left(\frac{t-(j+\alpha)h}{2}\right)} = \frac{\cos(jh)-\cos(x-\alpha h)}{\cos(jh)-\cos(t-\alpha h)}.$$

Note that only the denominators of these functions vary with t; the numerators are constant. By Lemma 4.5, we have

$$\left|\frac{\sin\left(\frac{x-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{x-\widetilde{x}_j}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_j}{2}\right)}\right| \le \begin{cases} |f_1(\widetilde{x}_0)| & -\pi \le x \le x_{-j}^* \\ |f_2(\widetilde{x}_0)| & x_{-j}^* \le x \le \widetilde{x}_0 \\ |f_3(\widetilde{x}_0)| & \widetilde{x}_0 \le x \le x_j^*. \end{cases} \tag{4.7}$$

Recalling that $\widetilde{x}_0 \in [-\alpha h, \alpha h]$, by maximizing $|f_1(t)|$, $|f_2(t)|$, and $|f_3(t)|$ over $t \in [-\alpha h, \alpha h]$ under the appropriate conditions on x, we will show that this inequality may be replaced by

$$\left|\frac{\sin\left(\frac{x-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{x-\widetilde{x}_j}{2}\right)}{\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_{-j}}{2}\right)\sin\left(\frac{\widetilde{x}_0-\widetilde{x}_j}{2}\right)}\right| \leq \begin{cases} |f_1(\alpha h)| & -\pi \leq x \leq x_{-j}^* \\ |f_2(\alpha h)| & x_{-j}^* \leq x \leq 0, \end{cases}$$

and this is the inequality we are trying to establish. There are three cases to consider.

Case 1: $-\pi \leq x \leq x_{-j}^*$. In this case, the right-hand side of (4.7) is governed by f_1. The denominator of f_1 has a critical point in $[-\alpha h, \alpha h]$ at $t = 0$, and it takes on identical values at the endpoints $\pm\alpha h$. Since

$$0 < \alpha h < (1-\alpha)h \leq (j-\alpha)h \leq (N-\alpha)h < \left(N + \frac{1}{2}\right)h = \pi,$$

we have $\cos\bigl((j-\alpha)h\bigr) \leq \cos(\alpha h) \leq 1$, and so $\bigl|\cos\bigl((j-\alpha)h\bigr) - \cos(\alpha h)\bigr| \leq \bigl|\cos\bigl((j-\alpha)h\bigr) - 1\bigr|$. Thus, the denominator is smallest in magnitude at $t = \pm\alpha h$. Since the numerator of f_1 does not vary with t, we are done.

Case 2: $x_{-j}^* \leq x \leq -\alpha h$. Here, the behavior of (4.7) is determined by f_2. The only critical point of the denominator f_2 in $[-\alpha h, \alpha h]$ is at the left endpoint, where it takes the value $\cos(jh) - 1$. At the right endpoint, the denominator is $\cos(jh) - \cos(2\alpha h)$. From

$$0 < 2\alpha h < h \leq jh \leq Nh < \left(N + \frac{1}{2}\right)h = \pi,$$

we see that $\cos(jh) \leq \cos(2\alpha h) \leq 1$, and so we have $|\cos(jh) - \cos(\alpha h)| \leq |\cos(jh) - 1|$. Thus, the denominator is smallest in magnitude at $t = \alpha h$, and we are done, as in the previous case.

Case 3: $-\alpha h \leq x \leq 0$. In this case, for $-\alpha h \leq \widetilde{x}_0 \leq x$, the right-hand side of (4.7) is governed by f_3, while for $x \leq \widetilde{x}_0 \leq \alpha h$, it is governed by f_2. From the previous case, we know that the maximum absolute value of $f_2(t)$ for $t \in [-\alpha h, \alpha h]$ occurs at $t = \alpha h$, and a virtually identical argument shows that the maximum absolute value of $f_3(t)$ over the same range occurs at $t = -\alpha h$. We are thus left to compare $|f_3(-\alpha h)|$ and $|f_2(\alpha h)|$. Since these two quantities have the same denominator, we need only compare their numerators. The conditions on x imply that

$$0 \leq \alpha h + x \leq \alpha h - x \leq 2\alpha h \leq jh < \pi,$$

the later inequalities following as in the developments of the previous case. Therefore, $\cos(jh) \leq \cos(x - \alpha h) \leq \cos(x + \alpha h)$, which implies that $|\cos(jh) - \cos(x-\alpha h)| \leq |\cos(jh) - \cos(x + \alpha h)|$. It follows that $|f_2(\alpha h)| \geq |f_3(-\alpha h)|$, as desired. □

At last, we can prove the following result, which gives a bound on $|\widetilde{\ell}_0(x)|$ for $x \in [-\pi, 0]$ that is independent of the points $\widetilde{x}_j$. First, we introduce some additional notation that we will need for the remainder of our argument. Define $x_0^* = 0$ and $x_{-N-1}^* = -\pi$. For $0 \leq k \leq N$, let $R_k^* = [x_{-k-1}^*, x_{-k}^*]$ and $R_k = [(-k-1-\alpha)h, (-k+\alpha)h]$. Observe that $\bigcup_{k=0}^N R_k^* = [-\pi, 0]$. Again for $0 \leq k \leq N$, let

$$P_k(x) = \left(\prod_{j=1}^N \left|\sin\left(\frac{x-(j-\alpha)h}{2}\right)\right|\right)\left(\prod_{j=1}^k \left|\sin\left(\frac{x+(j-\alpha)h}{2}\right)\right|\right)\left(\prod_{j=k+1}^N \left|\sin\left(\frac{x+(j+\alpha)h}{2}\right)\right|\right),$$

and let

$$Q_k = \left(\prod_{j=1}^{N} \left| \sin\left(\frac{(2\alpha - j)h}{2} \right) \right| \right) \left(\prod_{j=1}^{k} \left| \sin\left(\frac{jh}{2} \right) \right| \right) \left(\prod_{j=k+1}^{N} \left| \sin\left(\frac{(2\alpha + j)h}{2} \right) \right| \right).$$

Finally, define

$$M_k = \max_{x \in [-\pi, 0] \cap R_k} \frac{P_k(x)}{Q_k},$$

and note that M_k does not depend on the points $\widetilde{x}_j$.

Lemma 4.7. *For $0 \le k \le N$ and $x \in R_k^*$, we have $|\widetilde{\ell}_0(x)| \le M_k$.*

Proof. Just multiply together the inequalities derived in Lemma 4.6 for $1 \le j \le N$ and note that $R_k^* \subset R_k$ by Lemma 4.4. □

Next, we turn to bounding M_k. Our strategy will be to reduce the products $P_k(x)$ and Q_k to sums by taking logarithms and then bounding the sums using integrals. We begin with $P_k(x)$, which requires more work than Q_k because of its dependence on x. The bound that we need is given by Lemma 4.14, but before presenting it, we first establish several minor technical results that we will need in its proof.

Lemma 4.8. *For $0 \le k \le N$ and $x \in R_k$,*

$$\left| \sin\left(\frac{x + (k - \alpha)h}{2} \right) \sin\left(\frac{x + (k + 1 + \alpha)h}{2} \right) \right| \le \left| \sin\left(\frac{(\alpha + 1/2)h}{2} \right) \right|^2.$$

Proof. The derivative of the expression inside the absolute value signs on the left-hand side of this inequality is $(1/2)\sin\bigl(x + (k + 1/2)h\bigr)$, which vanishes inside R_k only at $x = -(k + 1/2)h$. The maximum absolute value of the expression must occur at this point, since it is zero at the endpoints of R_k. Substituting this value in for x in the left-hand side, we arrive at the right-hand side. □

Lemma 4.9. *For $1 \le k \le N$ and $x \in R_k$,*

$$\left| \sin\left(\frac{x - (1 - \alpha)h}{2} \right) \sin\left(\frac{x + (1 - \alpha)h}{2} \right) \right| \ge \left| \sin\left(\frac{(k + 1 - 2\alpha)h}{2} \right) \sin\left(\frac{(k - 1)h}{2} \right) \right|.$$

Proof. Let $f(x)$ be the expression inside the absolute value signs on the left-hand side of this inequality. Applying some trigonometric identities, we find that $f(x) = \cos\bigl((1 - \alpha)h\bigr)/2 - \cos(x)/2$. If $1 \le k \le N - 1$, then since

$$0 \le (1 - \alpha)h \le (k - \alpha)h \le -x \le (k + 1 + \alpha)h \le (N + \alpha)h < \pi,$$

we have $\cos(x) \le \cos\bigl((1 - \alpha)h\bigr)$, and so $f(x) \ge 0$ for $x \in R_k$. The same string of inequalities shows that $f'(x) = \sin(x)/2$ is negative on R_k, so f is decreasing on R_k. Therefore, the smallest absolute value of f is obtained by evaluating at the right endpoint $x = (-k + \alpha)h$, and this produces the expression on the right-hand side of the inequality to be established.

For the $k = N$ case, we note that f has a critical point in R_N at the midpoint $x = -\pi$. Since $f''(x) = \cos(x)/2$, we have $f''(-\pi) = -1/2$, and so this point is a local maximum. Thus, the minimum must occur at one of the two endpoints. Noting that f is even about π, the value of f must be the same at both endpoints, so we may as well pick the right endpoint $x = (-N + \alpha)h$. Since $0 \leq (1 - \alpha)h \leq (N - \alpha)h \leq \pi$, the value of f at this endpoint is nonnegative, completing the proof. □

Lemma 4.10. *For $K \geq 3$ and $x \in R_0$,*

$$\left|\sin\left(\frac{x - (1-\alpha)h}{2}\right)\sin\left(\frac{x + (1+\alpha)h}{2}\right)\right| \leq \left|\sin\left(\frac{h}{2}\right)\right|^2$$

Proof. As in the previous argument, let $f(x)$ be the expression inside the absolute value signs on the left-hand side of the inequality, and note that $f(x) = \cos(h)/2 - \cos(x + \alpha h)/2$. Since

$$-\pi < -h \leq x + \alpha h \leq 2\alpha h \leq h < \pi,$$

for $x \in R_0$, we have $\cos(h) \leq \cos(x + \alpha h)$ for $x \in R_0$, and it follows that f is negative on R_0. Since $\cos(x + \alpha h) \leq 1$, we have $0 \geq f(x) \geq \cos(h)/2 - 1/2$. This lower bound is attained for $x \in R_0$ at $x = -\alpha h$. Thus, f attains its maximum absolute value on R_0 at $x = -\alpha h$, and substituting this value into the original expression for f yields the claimed inequality. □

Lemma 4.11. *For $K \geq 3$ and $x \in R_1$, the following inequalities hold:*

$$\begin{aligned}\left|\sin\left(\frac{x-(1-\alpha)h}{2}\right)\right| &\geq \left|\sin((1-\alpha)h)\right|,\\ \left|\sin\left(\frac{x+(1-\alpha)h}{2}\right)\right| &\leq \left|\sin\left(\frac{(1+2\alpha)h}{2}\right)\right|,\\ \left|\sin\left(\frac{x+(2+\alpha)h}{2}\right)\right| &\leq \left|\sin\left(\frac{(1+2\alpha)h}{2}\right)\right|.\end{aligned}$$

Proof. The first inequality follows from

$$-\pi \leq -\frac{3}{2}h \leq \frac{x-(1-\alpha)h}{2} \leq (\alpha - 1)h \leq 0,$$

the second from

$$-\pi \leq -\frac{(1+2\alpha)h}{2} \leq \frac{x+(1-\alpha)h}{2} \leq 0,$$

and the third from

$$0 \leq \frac{x+(2+\alpha)h}{2} \leq \frac{(1+2\alpha)h}{2} \leq \pi.$$

□

Lemma 4.12. *For $0 \leq k \leq N$ and $x \in R_k$,*

$$\begin{aligned}&[x + (k-\alpha)h]\log\left(-\frac{x+(k-\alpha)h}{2}\right) - [x + (k+1+\alpha)h]\log\left(\frac{x+(k+1+\alpha)h}{2}\right)\\ &\qquad \leq -(1+2\alpha)h\log\left(\frac{(\alpha+1/2)h}{2}\right).\end{aligned}$$

Proof. Let $f(x)$ be the expression on the left-hand side of this inequality. The derivative of f is

$$f'(x) = \log\left(-\frac{x+(k-\alpha)h}{x+(k+1+\alpha)h}\right),$$

and this vanishes in R_k only at the point $x = -(k+1/2)h$. Since $f(x)$ tends to $-\infty$ as x approaches the endpoints of R_k, f must assume its maximum value on R_k at this point. Evaluating f at this point yields the right-hand side of the claimed inequality. □

Lemma 4.13. *For $x \in R_0$,*

$$\big(x-(1-\alpha)h\big)\log\left(-\frac{x-(1-\alpha)h}{2}\right) - \big(x+(1+\alpha)h\big)\log\left(\frac{x+(1+\alpha)h}{2}\right) \le -2h\log\left(\frac{h}{2}\right).$$

Proof. As in the previous argument, let $f(x)$ be the expression on the left-hand side of the inequality. We have

$$f'(x) = \log\left(-\frac{x+(\alpha-1)h}{x+(\alpha+1)h}\right),$$

and this vanishes in R_0 only at the point $x = -\alpha h$. Moreover,

$$f''(x) = \frac{2h}{(x+\alpha h)^2 - h^2}.$$

The denominator of this function is a quadratic polynomial with positive leading coefficient and zeroes at $(-1-\alpha)h$ and $(1-\alpha)h$. Since $x \in R_0$, we have $(-1-\alpha)h \le x \le \alpha h < (1-\alpha)h$, and it follows that f'' is negative everywhere on R_0. This implies that f has a global maximum on R_0 at the critical point at $-\alpha h$ that we just found. Evaluating $f(-\alpha h)$ produces the right-hand side of the inequality to be established. □

Lemma 4.14. *For sufficiently large K and $x \in R_k$, $0 \le k \le N$, we have*

$$P_k(x) \le 5 \cdot 2^{-K} K$$

for $k = 0, 1$ and

$$P_k(x) \le 3 \cdot 2^{-K} K^{2\alpha} \left|\sin\left(\frac{(k+1-2\alpha)h}{2}\right)\sin\left(\frac{(k-1)h}{2}\right)\right|^{\alpha-1/2}$$

for $2 \le k \le N$.

Proof. Let $S_k(x) = \log P_k(x)$. For $1 \le j \le N$, define $a_j(x)$, $b_j(x)$, and $c_j(x)$ by

$$a_j(x) = \log\left|\sin\left(\frac{x-(j-\alpha)h}{2}\right)\right|,$$

$$b_j(x) = \log\left|\sin\left(\frac{x+(j-\alpha)h}{2}\right)\right|,$$

$$c_j(x) = \log\left|\sin\left(\frac{x+(j+\alpha)h}{2}\right)\right|.$$

For brevity, we will typically suppress the argument when referring to these quantities, writing a_j in place of $a_j(x)$, etc. Let

$$A_k(x) = \sum_{j=1}^{N-1} \frac{1}{2} h(a_j + a_{j+1}), \qquad B_k(x) = \sum_{j=1}^{k-1} \frac{1}{2} h(b_j + b_{j+1}), \qquad C_k(x) = \sum_{j=k+1}^{N-1} \frac{1}{2} h(c_j + c_{j+1}),$$

and note that

$$hS_k(x) = A_k(x) + B_k(x) + C_k(x) + \frac{1}{2} h(a_1 + a_N + b_1 + b_k + c_{k+1} + c_N).$$

The sums $A_k(x)$, $B_k(x)$, and $C_k(x)$ are composite trapezoid rule approximations to the integral of $\log\left|\sin\big((x+t)/2\big)\right|$ (with respect to t) over certain subintervals of $[-\pi, \pi]$. Since this function is concave-down everywhere on $[-\pi, \pi]$, these approximations will yield lower bounds on the corresponding integrals [23, p. 54]. More precisely, we have

$$\begin{aligned} A_k(x) &\leq \int_{-(N-\alpha)h}^{-(1-\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt, \\ B_k(x) &\leq \int_{(1-\alpha)h}^{(k-\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt, \\ C_k(x) &\leq \int_{(k+1+\alpha)h}^{(N+\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt, \end{aligned}$$

where the inequality for $B_k(x)$ holds for $1 \leq k \leq N$, and the inequality for $C_k(x)$ holds for $0 \leq k \leq N-1$. There are four cases to consider. To aid the reader's comprehension, diagrams illustrating the different cases and how they are handled are presented in Figure 4.1.

Case 1: $2 \leq k \leq N-1$. In this case, the preceding developments yield

$$\begin{aligned} hS_k(x) \leq \int_{-\pi}^{\pi} - \int_{-\pi}^{-(N-\alpha)h} - \int_{-(1-\alpha)h}^{(1-\alpha)h} - \int_{(k-\alpha)h}^{(k+1+\alpha)h} - \int_{(N+\alpha)h}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt \\ + \frac{1}{2} h(a_1 + a_N + b_1 + b_k + c_{k+1} + c_N). \end{aligned}$$

Now we just need to bound the integrals and loose terms on the right-hand side of this inequality. It turns out that the first integral can be evaluated explicitly [52, 4.384-7]:

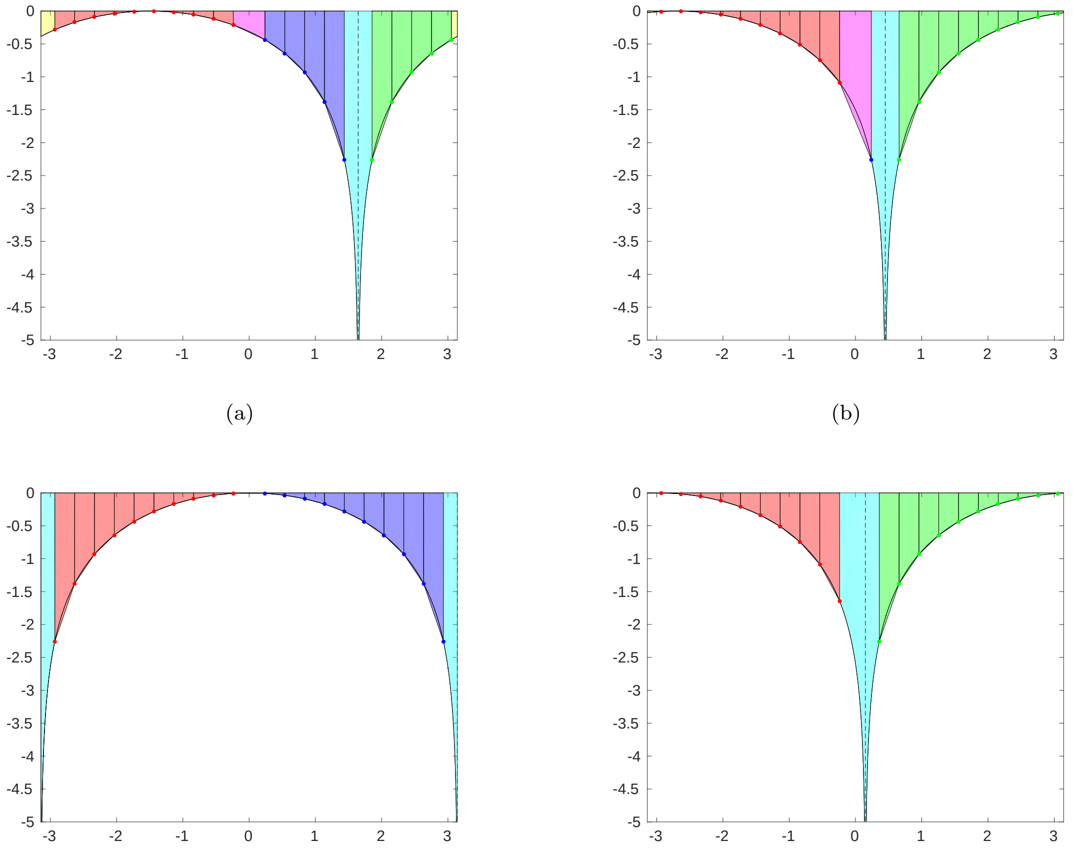
$$\int_{-\pi}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = -\pi \log(4). \tag{4.8}$$

For the second and fifth integrals, we have the following bound, which can be derived by applying the trapezoid rule to the integral from $(N+\alpha)h$ to $2\pi - (N-\alpha)h$ and using the periodicity of the integrand:

$$-\int_{-\pi}^{-(N-\alpha)h} - \int_{(N+\alpha)h}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt \leq -\frac{1}{2} h(a_N + c_N). \tag{4.9}$$

The fourth integral requires some care, since it has a singularity in the interval of integration at the point $t = -x$. (Recall our assumption that $x \in R_k = [(-k-1-\alpha)h, (-k+\alpha)h]$.) We therefore split the integral into two parts at that point. Noting the expansion

$$\log\big(\sin(t)\big) = \log(t) - \frac{1}{6} t^2 - \frac{1}{180} t^4 + O(t^6)_{t \to 0^+}, \tag{4.10}$$



(a)

(b)

(c)

(d)

Figure 4.1: Diagrams illustrating how to handle the four different cases in the proof of Lemma 4.14. Here, $K = 21$, and $\alpha = 0.2$. (a) Case 1 ($2 \leq k \leq N-1$; the plot shows $k = 5$). (b) Case 2 ($k = 1$). (c) Case 3 ($k = N$). (d) Case 4 ($k = 0$). Each plot displays the integrand $\sin\big((x+t)/2\big)$ as a function of t for x equal to a chosen point in R_k. The integrand is divided into several regions based on the trapezoid scheme used to bound it. The regions and evaluation points are colored according to the trapezoid sum in which they are included. Red regions and points correspond to the sum $A_k(x)$ and the points a_j. Blue regions and points correspond to the sum $B_k(x)$ and the points b_j. Green regions and points correspond to the sum $C_k(x)$ and the points c_j. The regions that must be handled differently due to the singularity in the integrand are colored in cyan; the singularity is indicated by a dashed black line (except in (c), in which it occurs at the interval boundaries). The magenta regions correspond to the interval of width $2(1-\alpha)h$ between a_1 and b_1. The yellow regions correspond to $[-\pi, a_N]$ and $[c_N, \pi]$. Using periodicity, we treat these as a single interval and bound the integral over it with a single trapezoid sum. Note that a magenta region is present in (c) and that yellow regions are present in (b) and (d), though they are difficult to see in the diagrams due to the integrand being small.

we have

$$-\int_{(k-\alpha)h}^{-x} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(x+(k-\alpha)h\big)\left[\log\left(-\frac{x+(k-\alpha)h}{2}\right)-1\right]+O(h^3)$$

and

$$-\int_{-x}^{(k+1+\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(x+(k+1+\alpha)h\big)\left[1-\log\left(\frac{x+(k+1+\alpha)h}{2}\right)\right]+O(h^3).$$

Adding these expressions together and applying Lemma 4.12, we obtain

$$-\int_{(k-\alpha)h}^{(k+1+\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt \le (1+2\alpha)h-(1+2\alpha)h\log\left(\frac{(\alpha+1/2)h}{2}\right)+O(h^3).$$

For the third integral, we use another trapezoid rule bound and combine the result with the loose terms $(1/2)h(a_1+b_1)$ to yield

$$\begin{aligned}-\int_{-(1-\alpha)h}^{(1-\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt + \frac{1}{2}h(a_1+b_1) &\le \left(\alpha-\frac{1}{2}\right)h(a_1+b_1)\\ &\le \left(\alpha-\frac{1}{2}\right)h\log\left|\sin\left(\frac{(k+1-2\alpha)h}{2}\right)\sin\left(\frac{(k-1)h}{2}\right)\right|, \quad (4.11)\end{aligned}$$

where the second inequality follows from Lemma 4.9 and the fact that $\alpha < 1/2$. Finally, by Lemma 4.8 and (4.10),

$$\frac{1}{2}h(b_k+c_{k+1}) \le h\log\left(\frac{(\alpha+1/2)h}{2}\right)+O(h^3). \quad (4.12)$$

Putting all of these results together, we conclude that

$$\begin{aligned}hS_k(x) \le -\pi\log(4)+\left(\alpha-\frac{1}{2}\right)h\log\left|\sin\left(\frac{(k+1-2\alpha)h}{2}\right)\sin\left(\frac{(k-1)h}{2}\right)\right|&\\ -2\alpha h\log\left(\frac{(\alpha+1/2)h}{2}\right)+(1+2\alpha)h+O(h^3).& \quad (4.13)\end{aligned}$$

Dividing through by h, exponentiating, and suitably relaxing the constants that emerge now yields the claimed bound in this case.

Case 2: $k = 1$. This case is very similar to the previous one. In particular, all of the same integral bounds apply except that the second inequality in (4.11) is meaningless because the argument to the logarithm function vanishes. We replace (4.11) and (4.12) with

$$\begin{aligned}-\int_{-(1-\alpha)h}^{(1-\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt + \frac{1}{2}h(a_1+2b_1+c_2) &\le \left(\alpha-\frac{1}{2}\right)ha_1+\frac{1}{2}hc_2+\alpha hb_1\\ &\le \left(\alpha-\frac{1}{2}\right)h\log\big((1-\alpha)h\big)+\left(\alpha+\frac{1}{2}\right)h\log\left(\frac{(1+2\alpha)h}{2}\right)+O(h^3),\end{aligned}$$

where the second inequality follows from Lemma 4.11 and (4.10). Combining this with the other results just established, we obtain

$$\begin{aligned}hS_1(x) \le -\pi\log(4)+\left(\alpha-\frac{1}{2}\right)h\log\big((1-\alpha)h\big)+\left(\alpha+\frac{1}{2}\right)h\log\left(\frac{(1+2\alpha)h}{2}\right)&\\ -(1+2\alpha)h\log\left(\frac{(\alpha+1/2)h}{2}\right)+(1+2\alpha)h+O(h^3),&\end{aligned}$$

and this implies the claimed bound for this case.

Case 3: $k = N$. Since $C_N(x)$ has no terms, we have, in this case,

$$hS_N(x) \le \int_{-\pi}^{\pi} - \int_{-\pi}^{-(N-\alpha)h} - \int_{-(1-\alpha)h}^{(1-\alpha)h} - \int_{(N-\alpha)h}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt + \frac{1}{2}h(a_1 + a_N + b_1 + b_N).$$

We can bound the third integral and the loose terms $(1/2)h(a_1+b_1)$ using (4.11); however, we cannot use (4.9) to bound the second and fourth integrals. Instead, noting that there is a singularity at $-x$ (or a periodic image thereof) within the domain of integration, we use (4.10) to find that

$$-\int_{(N-\alpha)h}^{-x} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(x + (N-\alpha)h\big)\left[\log\left(-\frac{x+(N-\alpha)h}{2}\right) - 1\right] + O(h^3)$$

and

$$-\int_{-x}^{2\pi-(N-\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(2\pi + x - (N-\alpha)h\big)\left[1 - \log\left(\frac{2\pi + x - (N-\alpha)h}{2}\right)\right] + O(h^3).$$

Noting that $2\pi + x - (N-\alpha)h = x + (N+1+\alpha)h$, we can add these together and use periodicity and Lemma 4.12 to obtain

$$-\int_{-\pi}^{-(N-\alpha)h} - \int_{(N-\alpha)h}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt \le (1+2\alpha)h - (1+2\alpha)h\log\left(\frac{(\alpha+1/2)h}{2}\right) + O(h^3).$$

Finally, by the same identity, Lemma 4.8, and (4.10), we have

$$\frac{1}{2}h(a_N + b_N) \le h\log\left(\frac{(\alpha+1/2)h}{2}\right) + O(h^3).$$

Putting everything together, we arrive once again at (4.13), which finishes the argument in this case.

Case 4: $k = 0$. As $B_0(x)$ has no terms, we have

$$hS_0(x) \le \int_{-\pi}^{\pi} - \int_{-\pi}^{-(N-\alpha)h} - \int_{-(1-\alpha)h}^{(1+\alpha)h} - \int_{(N+\alpha)h}^{\pi} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt + \frac{1}{2}h(a_1 + a_N + c_1 + c_N).$$

We can take care of the second and fourth integrals and the loose terms $(1/2)h(a_N+c_N)$ using (4.9). For the third integral, noting that $-x$ lies in the interval of integration, we use (4.10) one more time to conclude that

$$-\int_{-(1-\alpha)h}^{-x} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(x - (1-\alpha)h\big)\left[\log\left(-\frac{x-(1-\alpha)h}{2}\right) - 1\right] + O(h^3)$$

and

$$-\int_{-x}^{(1+\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt = \big(x + (1+\alpha)h\big)\left[1 - \log\left(\frac{x+(1+\alpha)h}{2}\right)\right] + O(h^3).$$

Adding these together and using Lemma 4.13 gives

$$-\int_{-(1-\alpha)h}^{(1+\alpha)h} \log\left|\sin\left(\frac{x+t}{2}\right)\right| \, dt \le -2h\log\left(\frac{h}{2}\right) + 2h + O(h^3).$$

Finally, by Lemma 4.10 and (4.10), we have

$$\frac{1}{2}h(a_1 + c_1) \le h\log\left(\frac{h}{2}\right) + O(h^3).$$

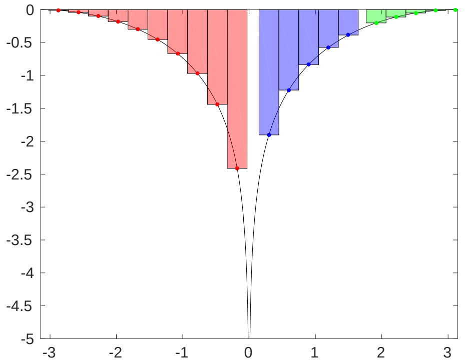


Figure 4.2: Diagram illustrating the argument for the proof of Lemma 4.15. Here, $K = 21$, $\alpha = 0.2$, and $k = 5$. The red, blue, and green rectangles correspond to the midpoint rule bounds associated with the first, second, and third sums in (4.14), respectively. The integral over singularity-containing gap between the rightmost red rectangle and leftmost blue one is accounted for in the second integral on the right-hand side of (4.15), while the integral over the gap between the rightmost blue rectangle and the leftmost green one is accounted for in the third.

Assembling all of these facts, we find that

$$hS_0(x) \leq -\pi \log(4) - h \log\left(\frac{h}{2}\right) + 2h + O(h^3),$$

and upon dividing through by h, exponentiating, and adjusting the constant factors that arise, we obtain the desired result.

All cases have now been handled. The proof is complete. □

Next, we bound Q_k. The result we need is the following:

Lemma 4.15. *For sufficiently large K,*

$$Q_k \geq (1 - 2\alpha)2^{-K} K^{1-2\alpha} \left|\sin\left(\frac{(k + 1/2 + \alpha)h}{2}\right)\right|^{-2\alpha}.$$

Proof. The proof is similar in structure to that of Lemma 4.14. Let $S_k = \log(Q_k)$, so that

$$S_k = \sum_{j=1}^{N} \log\left|\sin\left(\frac{(2\alpha - j)h}{2}\right)\right| + \sum_{j=1}^{k} \log\left|\sin\left(\frac{jh}{2}\right)\right| + \sum_{j=k+1}^{N} \log\left|\sin\left(\frac{(2\alpha + j)h}{2}\right)\right|. \tag{4.14}$$

We will bound S_k using integrals of $\log\left|\sin(t/2)\right|$, just as before, but this time, since we seek a lower bound, we use the midpoint rule instead of the trapezoid rule [23, p. 54]. A diagram illustrating the argument is given in Figure 4.2. Assuming $0 \leq \alpha \leq 1/4$, we have

$$hS_k \geq \int_{-\pi}^{\pi} - \int_{(2\alpha-1/2)h}^{h/2} - \int_{(k+1/2)h}^{(k+2\alpha+1/2)h} \log\left|\sin\left(\frac{t}{2}\right)\right| \, dt. \tag{4.15}$$

We evaluated the first integral in (4.8), above. We bound the third integral using the midpoint rule:

$$-\int_{(k+1/2)h}^{(k+2\alpha+1/2)h} \log\left|\sin\left(\frac{t}{2}\right)\right| \, dt \geq -2\alpha h \log\left|\sin\left(\frac{(k + 1/2 + \alpha)h}{2}\right)\right|.$$

For the second integral, we split the interval of integration at the singularity at 0 and use (4.10) to compute

$$-\int_{(2\alpha-1/2)h}^{h/2} \log\left|\sin\left(\frac{t}{2}\right)\right| \, dt = \left(2\alpha - \frac{1}{2}\right) h \log\left(\frac{(1/2-2\alpha)h}{2}\right) - \frac{h}{2}\log\left(\frac{h}{4}\right) + (1-2\alpha)h + O(h^3).$$

From these results, it follows that

$$\begin{aligned} hS_k \geq -\pi\log(4) - 2\alpha h \log\left|\sin\left(\frac{(k+1/2+\alpha)h}{2}\right)\right| \\ + \left(2\alpha - \frac{1}{2}\right) h \log\left(\frac{(1/2-2\alpha)h}{2}\right) - \frac{h}{2}\log\left(\frac{h}{4}\right) + (1-2\alpha)h + O(h^3). \end{aligned}$$

Dividing through by h, exponentiating, and suitably adjusting the constant factors that arise, we obtain the claimed result.

If $1/4 < \alpha < 1/2$, the argument is similar except that we have to track the $j = 1$ term in the first sum in the definition of S_k independently. We write

$$hS_k \geq \int_{-\pi}^{\pi} - \int_{(2\alpha-3/2)h}^{h/2} - \int_{(k+1/2)h}^{(k+2\alpha+1/2)h} \log\left|\sin\left(\frac{t}{2}\right)\right| \, dt + h\log\left|\sin\left(\frac{(2\alpha-1)h}{2}\right)\right|.$$

Using (4.10) one last time, we compute

$$h\log\left|\sin\left(\frac{(2\alpha-1)h}{2}\right)\right| = h\log\left(\frac{(1-2\alpha)h}{2}\right) + O(h^3).$$

and

$$-\int_{(2\alpha-3/2)h}^{h/2} \log\left|\sin\left(\frac{t}{2}\right)\right| \, dt = \left(2\alpha - \frac{3}{2}\right) \log\left(\frac{(3/2-2\alpha)h}{2}\right) - \frac{h}{2}\log\left(\frac{h}{4}\right) + 2(1-\alpha)h + O(h^3).$$

Therefore,

$$\begin{aligned} hS_k \geq -\pi\log(4) - 2\alpha h \log\left|\sin\left(\frac{(k+1/2+\alpha)h}{2}\right)\right| + h\log\left(\frac{(1-2\alpha)h}{2}\right) \\ + \left(2\alpha - \frac{3}{2}\right) \log\left(\frac{(3/2-2\alpha)h}{2}\right) - \frac{h}{2}\log\left(\frac{h}{4}\right) + 2(1-\alpha)h + O(h^3). \end{aligned}$$

and this implies the claimed bound in the usual way. □

Note that the proof of this lemma shows that the constant $1-2\alpha$ can be dropped from the bound when $0 \leq \alpha \leq 1/4$. We have chosen for simplicity to include it in this case anyway because omitting it will at best improve our final results by a small constant factor.

At this point, we have all that we need to bound $|\widetilde{\ell}_0(x)|$ uniformly for $x \in [-\pi, \pi]$ and independent of the points $\widetilde{x}_j$: evaluate the bounds on M_k for $x \in [-\pi, 0] \cap R_k$ given by Lemmas 4.14 and 4.15 and take the maximum over k. Lemma 4.7 shows that the result bounds $|\widetilde{\ell}_0(x)|$ for $x \in [-\pi, 0]$. By symmetry, the same bound must hold for $x \in [0, \pi]$ as well. Even further, by considering circular rotations of the points $\widetilde{x}_j$, the bound can be seen to apply to $|\widetilde{\ell}_k(x)|$ for $k \neq 0$. Therefore, by (4.3), we could bound $\widetilde{\Lambda}_K$ by multiplying the bound on $|\widetilde{\ell}_0(x)|$ by K.

We can do better than this, however, because Lemmas 4.7 and 4.14 retain some information about how $|\widetilde{\ell}_0(x)|$ varies with x through the hypothesis that $x \in R_k$. We can use this information to get a better bound on $|\widetilde{\ell}_k(x)|$ for $k \neq 0$ than the one just described. The result we need is given by the following lemma, which we could have proved earlier but have delayed until now.

Lemma 4.16. *If $x \in R_p^*$, $0 \le p \le N$, then for $-N \le k \le N$,*

$$|\widetilde{\ell}_k(x)| \le \begin{cases} \max(M_{-(p+k)}, M_{-(p+k+1)}, M_{-(p+k+2)}) & -N \le p+k \le -2 \\ \max(M_0, M_1) & p+k = -1, 0 \\ \max(M_{p+k-1}, M_{p+k}, M_{p+k+1}) & 1 \le p+k \le N-1 \\ \max(M_{N-1}, M_N) & p+k = N \\ \max(M_{K-(p+k)}, M_{K-(p+k+1)}, M_{K-(p+k+2)}) & N+1 \le p+k \le 2N-1 \\ \max(M_0, M_1) & p+k = 2N. \end{cases} \tag{4.16}$$

Proof. For $k = 0$, the result follows from Lemma 4.7, which actually gives a stronger bound. The proof for $k \neq 0$ is ultimately just a matter of reducing it to the $k = 0$ case by exploiting circular and reflectional symmetry; however, there are some subtleties, so we will spell out the details to make things clear. Note that $x \in R_p^*$ implies $x \in R_p$ by Lemma 4.4.

First, suppose that $1 \le k \le N$. Then, $1 \le p+k \le 2N$, so only the last four cases in (4.16) are relevant. Let

$$\widehat{x}_j = \begin{cases} \widetilde{x}_{j+k} - kh & -N \le j \le N-k \\ \widetilde{x}_{j+k-K} + 2\pi - kh & N-k+1 \le j \le N. \end{cases}$$

These points are just a circular shift in $[-\pi, \pi]$ of the points $\widetilde{x}_j$ by kh. It follows that $\widetilde{\ell}_k(x) = \widehat{\ell}_0(x - kh)$, where $\widehat{\ell}_0$ is the (trigonometric) Lagrange basis function for the points $\widehat{x}_j$ that takes on the value 1 at $\widehat{x}_0$. It is easy to check that

$$\widehat{x}_j = \begin{cases} x_j + t_{j+k}h & -N \le j \le N-k \\ x_j + t_{j+k-K}h & N-k+1 \le j \le N, \end{cases}$$

where the x_j are the equispaced points (4.1), and the t_j are defined by (4.2). Thus, the points $\widehat{x}_j$ constitute a set of perturbed equispaced points of the sort that we have been considering. In particular, we can use Lemma 4.7 to bound $\widehat{\ell}_0(x - kh)$ and hence $\widetilde{\ell}_k(x)$. There are several cases to consider.

Case 1: $1 \le p+k \le N-1$. Since $x \in R_p$, it follows that $x - kh \in R_{p+k}$, which means that $x - kh$ must belong to one of R_{p+k-1}^*, R_{p+k}^*, and R_{p+k+1}^*, again by Lemma 4.4. By Lemma 4.7, $|\widehat{\ell}_0(x - kh)| \le \max(M_{p+k-1}, M_{p+k}, M_{p+k+1})$.

Case 2: $p+k = N$ and $(-p-1/2)h \le x \le (-p+\alpha)h$. We have $x - kh \in R_N$. Moreover, $x - kh \ge (-p-k-1/2)h = (-N-1/2)h = -\pi$, so $x - kh \in [-\pi, 0] \cap R_N$. Thus, $x - kh$ belongs to either R_N^* or R_{N-1}^* by Lemma 4.4, and so, Lemma 4.7 gives $|\widehat{\ell}_0(x - kh)| \le \max(M_{N-1}, M_N)$.

Case 3: $p+k = N$ and $(-p-1-\alpha)h \le x < (-p-1/2)h$. Again, we have $x - kh \in R_N$, but this time, $x - kh < \pi$. Nevertheless, $\widehat{\ell}_0(x - kh) = \widehat{\ell}_0(x - kh + 2\pi)$, and $x - kh + 2\pi \in [0, \pi] \cap -R_N$.

By reflecting the problem about 0 (i.e., replacing $\widehat{x}_j$ with $-\widehat{x}_j$ for each j and $x - kh + 2\pi$ by $-(x-kh+2\pi) \in [-\pi, 0] \cap R_N$), and applying Lemma 4.7, we obtain $|\widehat{\ell}_0(x-kh)| \leq \max(M_{N-1}, M_N)$ as in the previous case.

Case 4: $N+1 \leq p+k \leq 2N-1$. Just as in the previous case, we will look not at $\widehat{\ell}_0(x-kh)$ but at $\widehat{\ell}_0(x-kh+2\pi)$. Noting that $2\pi = Kh$, we see that $x-kh+2\pi \in -R_{K-(p+k+1)}$. Since $x \geq -\pi$ and $k \leq N$, we have $x-kh+2\pi \geq -\pi + (K-N)h = h/2 > 0$. Thus, $x-kh+2\pi \in [0,\pi] \cap -R_{K-(p+k+1)}$. Reflecting about 0 as was done in the previous case and noting that $-(x-kh+2\pi)$ must belong to one of $R^*_{K-(p+k)}$, $R^*_{K-(p+k+1)}$, and $R^*_{K-(p+k+2)}$ by Lemma 4.4, we may apply Lemma 4.7 to conclude that $|\widehat{\ell}_0(x-kh)| \leq \max(M_{K-(p+k)}, M_{K-(p+k+1)}, M_{K-(p+k+2)})$.

Case 5: $p+k = 2N$. This is handled exactly the same as the previous case except that since $x - kh + 2\pi \in [0,\pi] \cap -R_0$, we have that $-(x-kh+2\pi)$ can only belong to one of R^*_0 and R^*_1. Therefore, $|\widehat{\ell}_0(x-kh)| \leq \max(M_0, M_1)$.

For $-N \leq k \leq -1$, the argument is similar. In this case, the circularly shifted points $\widehat{x}_j$ are

$$\widehat{x}_j = \begin{cases} \widetilde{x}_{j+k} - kh & -N-k \leq j \leq N \\ \widetilde{x}_{j+k+K} - 2\pi - kh & -N \leq j \leq -N-k-1, \end{cases}$$

so that

$$\widehat{x}_j = \begin{cases} x_j + t_{j+k}h & -N-k \leq j \leq N \\ x_j + t_{j+k+K}h & -N \leq j \leq -N-k-1. \end{cases}$$

Just as before, we have $\widetilde{\ell}_k(x) = \widehat{\ell}_0(x-kh)$. Noting that $-N \leq p+k \leq N-1$, the proof again breaks into cases as follows.

Case 1: $1 \leq p+k \leq N-1$. Just as in the previous Case 1, we have $x - kh \in R_{p+k}$, and the result follows in exactly the same way.

Case 2: $p+k = 0$ and $(-p-1-\alpha)h \leq x \leq -ph$. Here, $x - kh \in R_0$, and the restriction on x forces $x - kh \leq 0$, so in fact, $x - kh \in [-\pi, 0] \cap R_0$. Therefore, $x - kh$ belongs to one of R^*_0 and R^*_1 by Lemma 4.4, and so by Lemma 4.7, we have $|\widehat{\ell}_0(x-kh)| \leq \max(M_0, M_1)$.

Case 3: $p+k = 0$ and $-ph < x \leq (-p+\alpha)h$. Now $x-kh \in R_0$, but $0 < x-kh \leq \alpha h$. To bound $\widehat{\ell}_0(x-kh)$ in this case, we reflect the problem about 0 as we did in some of the cases for positive k above. Since $[-\alpha h, \alpha h] \subset R_0$, we have $-(x-kh) \in [-\pi, 0] \cap R_0$, and so Lemma 4.7 tells us that $|\widehat{\ell}_0(x-kh)| \leq \max(M_0, M_1)$ once again.

Case 4: $p+k = -1$ and $(-p-1-\alpha)h \leq x \leq (-p-1)h$. In this case, $x - kh \in [-\alpha h, 0]$ and hence belongs to $[-\pi, 0] \cap R_0$. Applying Lemma 4.7, we have $|\widehat{\ell}_0(x-kh)| \leq \max(M_0, M_1)$ just as in the previous two cases.

Case 5: $p+k = -1$ and $(-p-1)h < x \leq (-p+\alpha)h$. Now, $x - kh \in [0,\pi] \cap -R_0$. Reflecting in 0 and using Lemma 4.7 yet again gives $|\widehat{\ell}_0(x-kh)| \leq \max(M_0, M_1)$.

Case 6: $-N \leq p+k \leq -2$. We have $x-kh \in -R_{-(p+k+1)}$. Since $-R_{-(p+k+1)} \subset [0,\pi]$, we reflect in 0 and observe that, by Lemma 4.4, $-(x-kh)$ belongs to one of $R^*_{-(p+k)}$, $R^*_{-(p+k+1)}$, and $R^*_{-(p+k+2)}$. Applying Lemma 4.7 one last time, we obtain $|\widehat{\ell}_0(x-kh)| \leq \max(M_{-(p+k)}, M_{-(p+k+1)}, M_{-(p+k+2)})$.

All cases have been handled. The proof is finished. □

The point of Lemma 4.16 is that it allows us to bound $\widetilde{\Lambda}_K$ by summing the bounds of Lemma 4.7 over k instead of maximizing them over k and multiplying by K as described previously.

Lemma 4.17. *We have*

$$\widetilde{\Lambda}_K \leq 9 \sum_{k=0}^{N} M_k. \tag{4.17}$$

Proof. Suppose that $x \in [-\pi, 0] \cap R_p^*$, $0 \leq p \leq N$. We can use Lemma 4.16 to bound the sum in (4.3) for this value of x by summing the right-hand side of (4.16) over $-N \leq k \leq N$. This is equivalent to summing it over the values of $p + k$ such that $-N + p \leq p + k \leq N + p$, and this is certainly bounded above by the sum over the larger range $-N \leq p + k \leq 2N$. Writing j in place of $p + k$, it follows that

$$\begin{aligned}\sum_{k=-N}^{N} |\widetilde{\ell}_k(x)| \leq{}& \sum_{j=-N}^{-2} \max(M_{-j}, M_{-(j+1)}, M_{-(j+2)}) + \sum_{j=1}^{N-1} \max(M_{j-1}, M_j, M_{j+1}) \\ &+ \sum_{j=N+1}^{2N-1} \max(M_{K-j}, M_{K-(j+1)}, M_{K-(j+2)}) + 3\max(M_0, M_1) + \max(M_{N-1}, M_N).\end{aligned}$$

Since $\max(a, b) \leq a + b$ when $a, b \geq 0$, we can convert the maxima into sums to obtain

$$\begin{aligned}\sum_{k=-N}^{N} |\widetilde{\ell}_k(x)| \leq{}& \sum_{j=2}^{N} M_j + \sum_{j=1}^{N-1} M_j + \sum_{j=0}^{N-2} M_j + \sum_{j=0}^{N-2} M_j + \sum_{j=1}^{N-1} M_j + \sum_{j=2}^{N} M_j \\ &+ \sum_{j=2}^{N} M_j + \sum_{j=1}^{N-1} M_j + \sum_{j=0}^{N-2} M_j + 3M_0 + 3M_1 + M_{N-1} + M_N\end{aligned}$$

after simplifying the indices of summation. We immediately obtain

$$\sum_{k=-N}^{N} |\widetilde{\ell}_k(x)| \leq 9 \sum_{j=0}^{N} M_j.$$

Since the right-hand side of this inequality is independent of p, this bound actually holds for all $x \in [-\pi, 0]$. Even further, since the M_j are independent of both x and the points (4.2), by symmetry, it holds for all $x \in [-\pi, \pi]$. The result now follows from (4.3). □

At last, we can prove Theorem 3.6.

Proof of Theorem 3.6. We use Lemmas 4.14 and 4.15 to bound the right-hand side of (4.17). For K sufficiently large and $k = 0, 1$, we have

$$M_k \leq \frac{5}{1-2\alpha} K^{2\alpha} \left| \sin\left(\frac{(k + 1/2 + \alpha)\pi}{K} \right) \right|^{2\alpha} \leq \frac{5}{1-2\alpha} K^{2\alpha} \left| \frac{(k + 1/2 + \alpha)\pi}{K} \right|^{2\alpha} \leq \frac{10\pi}{1-2\alpha}, \tag{4.18}$$

while for $2 \leq k \leq N$,

$$M_k \leq \frac{3}{1-2\alpha} K^{4\alpha-1} \frac{\left|\sin\left(\frac{(k+1/2+\alpha)\pi}{K}\right)\right|^{2\alpha}}{\left|\sin\left(\frac{(k+1-2\alpha)\pi}{K}\right)\sin\left(\frac{(k-1)\pi}{K}\right)\right|^{1/2-\alpha}} \leq \frac{3}{1-2\alpha} K^{4\alpha-1} \frac{\left|\sin\left(\frac{(k+1/2+\alpha)\pi}{K}\right)\right|^{2\alpha}}{\left|\sin\left(\frac{(k-1)\pi}{K}\right)\right|^{1-2\alpha}}.$$

In deriving the last expression, we have used the inequality

$$\left|\sin\left(\frac{(k+1-2\alpha)\pi}{K}\right)\right| \geq \left|\sin\left(\frac{(k-1)\pi}{K}\right)\right|,$$

which clearly holds for $2 \leq k \leq N-1$ and for $k = N$ with $1/4 \leq \alpha < 1/2$, since in those cases, $(k+1-2\alpha)\pi/K \in [0, \pi/2]$, and $k+1-2\alpha \geq k > k-1$. To see that it holds for $k = N$ with $0 < \alpha < 1/4$ as well, note that in this case

$$\sin\left(\frac{(N+1-2\alpha)\pi}{K}\right) = \sin\left(\frac{(N+2\alpha)\pi}{K}\right)$$

by the symmetry of sine about $\pi/2$. Since $N + 2\alpha \in [0, \pi/2]$ and $N + 2\alpha > N - 1$, the inequality follows.

Using the inequalities $|\sin(x)| \leq |x|$ for $x \in \mathbb{R}$ and $|\sin(x)| \geq (2/\pi)|x|$ for $|x| \leq \pi/2$, we can simplify the bound on M_k for $2 \leq k \leq N$ even further to

$$M_k \leq \frac{3}{1-2\alpha} K^{4\alpha-1} \frac{\left|\frac{(k+1/2+\alpha)\pi}{K}\right|^{2\alpha}}{\left|\frac{2(k-1)}{K}\right|^{1-2\alpha}} \leq \frac{3\pi}{1-2\alpha} \frac{(k+1)^{2\alpha}}{(k-1)^{1-2\alpha}}. \tag{4.19}$$

The result now follows from summing the bounds on the M_k established in (4.18) and (4.19) and bounding the sum by interpreting it as a midpoint rule approximation[2] to the integral of a function that is concave-up (note that $N + 1/2 = K/2$):

$$\begin{aligned}\sum_{k=2}^{N} \frac{(k+1)^{2\alpha}}{(k-1)^{1-2\alpha}} &\leq \int_{3/2}^{N+1/2} \frac{(x+1)^{2\alpha}}{(x-1)^{1-2\alpha}}\,dx \leq (K/2+1)^{2\alpha} \int_{3/2}^{K/2} \frac{dx}{(x-1)^{1-2\alpha}} \\ &= \frac{(K^2/4-1)^{2\alpha} - (K/4+1/2)^{2\alpha}}{2\alpha} \leq \frac{(K^2/4)^{2\alpha} - (K/4)^{2\alpha}}{2\alpha} = \frac{K^{4\alpha} - K^{2\alpha}}{4^{2\alpha} 2\alpha} \leq \frac{K^{4\alpha} - 1}{2\alpha}.\end{aligned}$$

□

We close this chapter with a word about why our argument falls short of establishing the stronger bound on $\widetilde{\Lambda}_K$ predicted by Conjecture 3.5. As summarized in the opening paragraphs of this chapter, our argument proceeds by choosing the perturbed points $\widetilde{x}_j$ to maximize $|\widetilde{\ell}_k|$ for a fixed value of k, bounding the maximum, and then summing the bounds. This is a different (and easier) problem than choosing the points to maximize the sum $\sum_{k=-N}^{N} |\widetilde{\ell}_k|$ and bounding that maximum instead.

In symbols, our argument bounds $\widetilde{\Lambda}_K$ by bounding the rightmost expression in the following chain of inequalities:

$$\widetilde{\Lambda}_K \leq \max_{\widetilde{x}_{-N},\ldots,\widetilde{x}_N} \max_{x \in [-\pi,\pi]} \sum_{k=-N}^{N} |\widetilde{\ell}_k(x)| \leq \max_{x \in [-\pi,\pi]} \sum_{k=-N}^{N} \max_{\widetilde{x}_{-N},\ldots,\widetilde{x}_N} |\widetilde{\ell}_k(x)|.$$

The loss enters in the passage to the rightmost expression from the one in the middle. It seems likely that any attempt to prove Conjecture 3.5 will need to consider the $|\widetilde{\ell}_k|$ all together at once in the sum instead of individually as we have done here.

[2] The author thanks Andrew Thompson for suggesting the use of the midpoint rule instead of a simpler Riemann sum. The latter yields a bound that does not have $O(\log K)$ behavior in the limit as $\alpha \to 0$.

Chapter 5

Rational Interpolation and Eigenvalue Computation[1]

In this final chapter, we consider an application of rational interpolation to computing the eigenvalues of large matrices. We show that the contour integral methods for such problems that have enjoyed some popularity in recent years are actually equivalent to computing the poles of a rational interpolant to the matrix resolvent. We then exploit this observation to devise a new algorithm that improves on these methods in the case where the matrix is real and symmetric by enabling one to use only real arithmetic.

5.1 Introduction

Let $A \in \mathbb{C}^{N \times N}$ be a large square matrix, and consider the problem of computing the eigenvalues of A that lie within a given region of the complex plane. Some of the most successful techniques for attacking this problem are based on projecting A onto an approximately invariant subspace associated with the eigenvalues of interest and computing the eigenvalues of the projection. Of these, perhaps the best known are the Krylov subspace techniques such as the implicitly restarted Arnoldi method [127], which is implemented in the widely used software package ARPACK [75].

Recently, a new class of algorithms has been proposed which derive their projections from complex contour integrals. Though early traces of these ideas can be found in the work of Goedecker on linear-scaling methods for electronic structure calculation [47, 48] and in that of Labreuche on numerical methods for nonlinear eigenvalue problems that arise in the study of acoustic resonance [74, p. 192–196], the best-known algorithms of this type are the Sakurai–Sugiura (SS) method [120] and the FEAST algorithm, due to Polizzi [104]. A major computational advantage offered by these algorithms is that they are very easily parallelizable.

[1] The content in this chapter is adapted from the paper [7] by the author and his doctoral supervisor Lloyd N. Trefethen. Trefethen posed the original question of whether rational interpolation in Chebyshev points could be useful for eigenvalue computation. The author worked out the details of how to accomplish this, including establishing the connection between rational interpolation and the Kravanja–Van Barel method for zerofinding and identifying the importance of the Rayleigh–Ritz approach based on rational filters. The author also wrote the text of the paper.

Some of the most important eigenvalue problems involve matrices A that are real and symmetric. For large such problems, it is natural to want to take advantage of the parallelism offered by contour integral methods; however, by their very nature, these methods require complex arithmetic even though the eigenvalues sought are real. Aside from the fact that the need to use complex arithmetic to solve a real problem is conceptually jarring, this means that methods based on contour integrals suffer roughly a factor of two penalty in both time and storage costs, compared with approaches that rely only on real arithmetic.

In this chapter, we present a technique that addresses this deficiency. Our approach is motivated by the connection between the SS method and rational interpolation established in [6] and can be succinctly described as a projection method that uses a rational filter with only real poles. Projection methods based on rational filters have been examined in the Japanese literature by Murakami [85, 86], whose work we discuss in some detail later on.

The remainder of our discussion is organized as follows. In Section 5.2, we review the connection between the SS method and rational interpolation, introduce the idea of using the latter to find eigenvalues, and show how this formulation yields a method with SS-like parallelism that uses only real arithmetic. Unfortunately, as we will see in Section 5.4, methods based directly on rational interpolation are numerically unstable, but in Section 5.5, we will show how they can be reformulated to avoid this difficulty by using a Rayleigh–Ritz procedure and rational filters. Section 5.6 contains information pertinent to developing practical realizations of our method. In Section 5.7, we give a summary of the proposed algorithm, and finally, in Section 5.8, we illustrate its performance on a numerical example.

While our main application is to real symmetric matrices A, much of our discussion does not depend directly on this structure. Accordingly, we will assume most of the time that A is arbitrary and specialize to the real symmetric or Hermitian case when appropriate.

5.2 Finding Poles of the Resolvent

The methods we consider are rooted in the fact that the eigenvalues of a matrix A are the poles of its resolvent $(A - zI)^{-1}$. Given vectors $u, v \in \mathbb{C}^N$, we consider the function

$$f(z) = u^*(A - zI)^{-1}v, \tag{5.1}$$

a "scalarized" version of $(A - zI)^{-1}$. (If A is Hermitian, it is common to take $u = v$.) If A is diagonalizable (an assumption we will make throughout the chapter), we can write it in an eigenvalue decomposition as

$$A = \sum_{h=1}^{N'} \lambda_h P_h,$$

where $\lambda_1, \ldots, \lambda_{N'}$ are the distinct eigenvalues of A and $P_1, \ldots, P_{N'}$ are the corresponding spectral projectors. Then, f takes the form

$$f(z) = \sum_{h=1}^{N'} \frac{u^* P_h v}{\lambda_h - z}.$$

Thus, f is a rational function, and if v and u are generic in the sense that v (respectively, u) is not orthogonal to any of the right (respectively, left) eigenspaces of A, then f will have a simple pole at each of the points λ_h. Our goal will be to compute the poles of this function that lie within a given region of interest.

5.2.1 The Sakurai–Sugiura method

The original method proposed by Sakurai and Sugiura in [120] computes the poles of f within a region $\Omega \subset \mathbb{C}$ bounded by a simple, closed, piecewise smooth curve γ by applying the derivative-free variant of the pole-finding algorithm of Kravanja and Van Barel [6, 72, 73]. If A has $s \leq N'$ distinct eigenvalues within Ω (a number which, for the time being, we will assume is known) this algorithm works by computing the moment integrals

$$\mu_j = \frac{1}{2\pi i} \int_\gamma z^j f(z)\, dz, \qquad j = 0, \ldots, 2s-1, \tag{5.2}$$

and using them to form the $s \times s$ Hankel matrices

$$H_s = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{s-1} \\ \mu_1 & \mu_2 & \cdots & \mu_s \\ \vdots & \vdots & & \vdots \\ \mu_{s-1} & \mu_s & \cdots & \mu_{2s-2} \end{bmatrix}, \qquad H_s^< = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_s \\ \mu_2 & \mu_3 & \cdots & \mu_{s+1} \\ \vdots & \vdots & & \vdots \\ \mu_s & \mu_{s+1} & \cdots & \mu_{2s-1} \end{bmatrix}.$$

The poles of f within γ are then given by the eigenvalues of the generalized eigenvalue problem for the pencil $H_s^< - \lambda H_s$. Because it makes use of these Hankel matrices, we will refer to this method as the SS-H method.

In practice, the integrals (5.2) cannot be computed exactly and must be approximated using a quadrature rule. If this rule is defined by nodes $z_0, \ldots, z_{K-1} \in \mathbb{C}$ and corresponding weights $w_0, \ldots, w_{K-1} \in \mathbb{C}$, then we obtain the approximation

$$\mu_j \approx \sum_{k=0}^{K-1} w_k z_k^j u^* (A - z_k I)^{-1} v. \tag{5.3}$$

The dominant contribution to the computational cost of applying this method comes from solving the K linear systems involving shifts of A at each quadrature node required to compute (5.3). If A is large, these linear solves are potentially expensive; however, as each system is independent of the others, they can be solved in parallel.

5.3 Rational Interpolation

An alternative method for computing the poles of f is to form a rational approximation to f and compute the poles of the approximation. The simplest type of rational approximation is a rational interpolant. Given K points $z_0, \ldots, z_{K-1} \in \mathbb{C}$, we seek polynomials p and q of appropriately chosen maximum degrees m and n such that

$$\frac{p(z_k)}{q(z_k)} = f(z_k), \qquad k = 0, \ldots, K-1.$$

As was described in Section 1.4.1, in practice, it is easier to multiply through by the denominator and work with the linearized conditions

$$p(z_k) = f(z_k)q(z_k), \qquad k = 0, \ldots, K-1. \tag{5.4}$$

Recall from Section 1.4 that we must choose m and n such that $m + n + 1 = K$. Obviously, the degree of q should be at least as large as the number of eigenvalues sought. As we are assuming for now that this number is known, we will take n to be exactly equal to it. In practice, n will need to be selected based on an estimate of this number, and we note in particular that it may be advantageous to make n larger, even if the number is known exactly; see Section 5.6.2. Once we have computed the interpolant, finding the poles is simply a matter of finding the roots of q, which can be accomplished by solving an eigenvalue problem, the structure of which depends on the basis chosen to represent q.

In terms of computational structure, rational interpolation is very similar to the SS-H method. The dominant cost involved comes from solving the K linear systems required to evaluate f at each of the interpolation nodes, and, just as in the SS-H method, these systems can be solved in parallel. Also as in the SS-H method, obtaining the approximations to the eigenvalues of A boils down to solving a small, dense eigenvalue problem.

In fact, it turns out that the SS-H method and rational interpolation are mathematically equivalent (in exact arithmetic) in the most basic case where (1) the eigenvalues sought are those in the unit disc, (2) the contour integrals in the SS-H method are discretized using the trapezoidal rule in the roots of unity, and (3) those same roots are used for the interpolation nodes when constructing the rational interpolant. This was shown in [6], where the observation takes the form of a theorem asserting the equivalence of rational interpolation and the derivative-free Kravanja–Van Barel method. This equivalence, combined with the success enjoyed by contour integral methods in practice, is one of the main reasons we have been motivated to consider approaches to eigenvalue computation based on rational interpolation.

5.3.1 Rational Interpolation on a Real Interval

Though the SS-H method and rational interpolation are very similar, as just observed, they provide different perspectives on how to approach eigenvalue computation. In particular, rational interpolation schemes are not constrained by the need to worry about contours, regions, or quadrature rules; there are only interpolation nodes. This simpler structure naturally leads one to consider schemes that are not so easily conceived in a framework based strictly on contour integrals.

Suppose that A is a Hermitian matrix, so that the eigenvalues of A are all real, and suppose that we seek the eigenvalues that lie in a given interval $I \subset \mathbb{R}$. For concreteness, let us suppose that $I = [-1, 1]$. Applying a contour integral method requires the selection of a contour in the complex plane that encloses I. The unit circle is an obvious choice and is often used, as are long, narrow ellipses that enclose I. Both of these choices of contour can be used as the basis for a successful algorithm.

If, in addition, A is real, however, this method has an unfortunate defect. As any reasonable choice of quadrature rule for approximating the contour integrals will have nodes that do not lie on the real axis, we must use complex arithmetic to solve a real eigenvalue problem. Complex operations take roughly twice as much work to perform as real operations, and storing complex matrices takes twice as much memory as storing real ones.

Rational interpolation affords us a way out of this problem. Unlike in a contour integral framework, we are not forced by the demands of a quadrature rule to take some evaluation points off the real axis. Instead, we are free to take the interpolation nodes all to lie in I. For the reasons discussed in Chapter 1, a natural candidate for a good set of nodes is a set of Chebyshev points in I.[2] In this chapter, we will work with the K Chebyshev points of the first kind $x_k^{(1)}$ given by (1.14). To keep our notation uncluttered, we will drop the superscript and refer to these points simply as x_k.

Thus, we arrive at the following simple algorithm: find p and q that satisfy (5.4) with $z_k = x_k$ and then compute the roots of q. Since all the points x_k are real, only linear systems involving real shifts are solved to evaluate the scalarized resolvent f. If A is real, only real arithmetic is employed.

All that remains is to decide how to solve the linearized rational interpolation problem (5.4). As discussed in Section 1.4.3, rational interpolation has a propensity for rounding errors to cause spurious pole-zero pairs to appear when the interpolant is computed in the obvious way. To combat this, we use the algorithm presented in [50], which combines the earlier algorithm of [97] with a regularization technique based on the SVD to detect and remove these spurious pairs; for a short description, see Sections 1.4.2 and 1.4.3. This algorithm is implemented in the MATLAB code `ratinterp` within the freely available Chebfun software package (see Section 1.5).

[2] The discussion in Chapter 1 is for polynomial, not rational, interpolation; however, the choice of Chebyshev points still makes sense as we are solving the linearized rational interpolation problem, which computes the numerator and denominator polynomials individually. Moreover, the algorithm for solving this problem described in Section 1.4.2 can be implemented efficiently using the fast Fourier transform when the Chebyshev points are used; see [97].

Exact	Computed	Error
0.0	0.00000240	2.40×10^{-6}
0.1	0.09992181	7.82×10^{-5}
0.2	0.20039592	3.96×10^{-4}
0.3	0.27060498	2.94×10^{-2}
0.4	0.40129619	1.30×10^{-3}
0.5	0.48977909	1.02×10^{-2}
0.6	0.59984376	1.56×10^{-4}
0.7	0.70009663	9.66×10^{-5}
0.8	0.79959982	4.00×10^{-4}
0.9	0.89999995	4.51×10^{-8}

Table 5.1: Eigenvalues and absolute errors illustrating instability of eigenvalue computation via rational interpolation for the test problem considered in the text.

5.4 Instability of Rational Interpolation for Finding Eigenvalues

Unfortunately, the simple algorithm just described suffers from numerical instability. The difficulty is that the eigenvalues of A in $[-1,1]$ may be distributed in such a manner that the polynomial rootfinding problem set up by the interpolation process is poorly conditioned. This problem can occur even if the number of eigenvalues of A in $[-1,1]$ is small and the eigenvalue problem itself is well-conditioned (which it always is if A is Hermitian), as we now illustrate.[3]

Suppose that A is a 12×12 diagonal matrix with diagonal entries -10, 10, and $0, 0.1, 0.2, \ldots, 0.9$ and that we wish to compute the 10 eigenvalues of A inside $[-1,1]$. Using the `ratinterp` code just mentioned, we can implement the algorithm of the previous section in just a few lines of MATLAB and Chebfun as follows:

```
A = diag([0:0.1:0.9, -10, 10]); I = eye(12); v = randn(12, 1);
K = 32; xk = chebpts(K, 1); fk = zeros(K, 1);
for k = 1:K, fk(k) = v’*((A - xk(k)*I) \ v); end
[p, q, r, mu, nu, pol] = ratinterp(fk, K - 11, 10, K, ’type1’, 0);
```

The first line of this code creates the matrix A and generates the random vector v that we will use to scalarize the resolvent. The second line uses the Chebfun code `chebpts` to create a vector of $K = 32$ Chebyshev points (1.14) in $[-1,1]$. The third line computes the values of the scalarized resolvent function at each interpolation point, storing the results in a vector `fk`. Finally, the fourth line computes a linearized rational interpolant to these values with a denominator degree of 10. The `’type1’` argument to `ratinterp` specifies that we are working with data from a first-kind Chebyshev grid, and the final `0` argument disables the aforementioned SVD-based robustness techniques, which are not necessary for this demonstration. The poles that form our computed approximations to the eigenvalues are stored in the output variable `pol`.

Approximations to the eigenvalues produced by a typical run of this code are displayed in Table 5.1. The eigenvalues at 0 and 0.9 are computed to only about five and seven digits of accuracy,

[3] This example was suggested by Grady Wright.

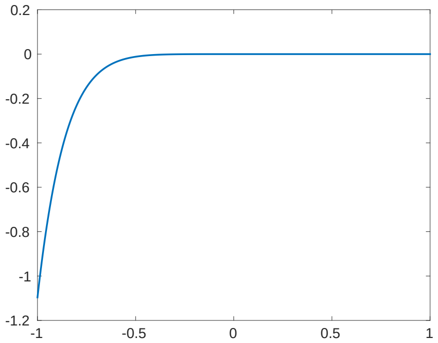


Figure 5.1: Plot of the denominator of the rational interpolant whose roots were computed to produce the eigenvalue approximations in Table 5.1.

respectively, and the accuracy is even worse for the eigenvalues in the middle of the spectrum. Increasing the value of K does not improve the accuracy, nor does enabling the SVD-based robustness techniques in `ratinterp`. The computation has been spoiled by rounding error.

The reason for this behavior becomes apparent when we look at the denominator polynomial of the rational interpolant, which is depicted in Figure 5.1. Observe how the polynomial is relatively large on the left half of the interval, while on the right half, where the spectrum of A is concentrated, it is nearly zero. (In fact, the values on the right half fluctuate near 10^{-8}.) This large scaling of the polynomial over the entire interval relative to its size on the half of the interval containing the spectrum causes its roots to be badly conditioned, resulting in the poor approximations to the eigenvalues we see in the table. In the polynomial rootfinding literature, this is sometimes referred to as a "dynamic range" problem [17].

The example just given is admittedly manufactured; however, it is easy to imagine that similar situations could arise in actual applications. Unfortunately, there seems to be little that can be done to prevent this without some a priori knowledge of the spectrum of A. A better solution is to avoid polynomial rootfinding altogether. Thus, we now modify the approach based on rational interpolation by turning to a method based on Rayleigh–Ritz procedures and rational filters.

5.5 Rayleigh–Ritz Reformulation Using Rational Filters

5.5.1 The SS-RR Method

Sakurai and coworkers noticed similar instabilities to those just described in the SS-H algorithm of [120] and devised an alternative version of their method based on a Rayleigh–Ritz procedure to correct this [61, 121]. When applied to Hermitian matrices, their algorithm works as follows. Given a vector v that is generic in the sense defined in Section 5.2, the method computes a basis $\{v_0, \ldots, v_{s-1}\}$ for the invariant subspace corresponding to the eigenvalues of A within γ by computing the projections of the vectors $A^j v$ onto this eigenspace via the integrals

$$v_j = \frac{1}{2\pi i} \int_\gamma z^j (A - zI)^{-1} v \, dz, \qquad j = 0, \ldots, s-1. \tag{5.5}$$

The vectors v_j are then orthonormalized, and the resulting vectors are gathered as columns into a matrix Q. The desired eigenvalues and eigenvectors are then obtained by solving the $s \times s$ eigenvalue problem for the matrix $Q^* A Q$. Because of its use of a Rayleigh–Ritz procedure, we will refer to this method as the SS-RR method.

5.5.2 Contour Integrals and Filter Functions

The key mechanism underlying the SS-H and SS-RR methods—and, indeed, all contour integral methods—is that the contour integrals (5.2) and (5.5) compute a projection of a vector onto the eigenspace of interest, since

$$P = -\frac{1}{2\pi i} \int_\gamma (A - zI)^{-1} \, dz \tag{5.6}$$

is the spectral projector associated with the eigenspace corresponding to the eigenvalues of A contained within γ [68]. When we discretize (5.6) using a quadrature rule defined by K distinct nodes $z_0, \ldots, z_{K-1}$ and corresponding weights $w_0, \ldots, w_{K-1}$, we obtain an approximate projector

$$\widehat{P} = \sum_{k=0}^{K-1} w_k (A - z_k I)^{-1}. \tag{5.7}$$

Written another way, we have $\widehat{P} = H(A)$, where H is the rational function

$$H(z) = \sum_{k=0}^{K-1} \frac{w_k}{z - z_k}. \tag{5.8}$$

We call H the *filter function* associated with the method because it describes how (5.7) acts to filter out undesired eigenvectors while retaining the rest. The points z_k are the *poles* of the filter, and the w_k are the corresponding *residues*. If λ is an eigenvalue of A for which $H(\lambda)$ is small, then when $\widehat{P}$ acts on a vector v, it will reduce the components of v in the directions of eigenvectors of A corresponding to λ.

Contour integral methods have been discussed from the viewpoint of rational filters in several places in the literature. Sakurai and coworkers do this for the SS-H method in [62] and for SS-RR in [61]. Tang and Polizzi do the same for FEAST in [130].

5.5.3 Filters Derived from Rational Interpolation

We will now show that the process of rational interpolation described in Section 5.3 for our scalarized resolvent function also acts to filter the spectrum of A by a rational function. This is clear in the case where the result from [6] mentioned at the end of Section 5.3 is applicable, for under those conditions, rational interpolation is equivalent to the SS-H method based on discretized contour integrals, and we have just shown that all discretized contour integrals have a rational filter behind them. The easiest way to see that this is true for other choices of the interpolation nodes is to extend the result from [6] to a more general setting.

Specifically, let $f : \mathbb{C} \to \mathbb{C} \cup \{\infty\}$ be a meromorphic function, let $z_0, \ldots, z_{K-1}$ be K distinct points in $\mathbb{C}$ that are not poles of f, and let $w_0, \ldots, w_{K-1} \in \mathbb{C}$ be nonzero. Let $n \geq 1$, and consider the following two computational procedures:

- Procedure (K):

  1. Compute the quantities
  $$\mu_j = \sum_{k=0}^{K-1} w_k z_k^j f(z_k), \qquad j = 0, \ldots, 2n-1.$$
  2. Form the Hankel matrices
  $$H_n = \begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{n-1} \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & & \vdots \\ \mu_{n-1} & \mu_n & \cdots & \mu_{2n-2} \end{bmatrix}, \qquad H_n^< = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \vdots & \vdots & & \vdots \\ \mu_n & \mu_{n+1} & \cdots & \mu_{2n-1} \end{bmatrix}.$$
  3. Compute the eigenvalues, counted according to multiplicity, of the matrix pencil $H_n^< - \lambda H_n$.

- Procedure (R):

  1. Compute the $J \leq K-1$ zeros $\eta_0, \ldots, \eta_{J-1}$ of the rational function
  $$H(z) = \sum_{k=0}^{K-1} \frac{w_k}{z - z_k}.$$
  2. Compute a linearized rational interpolant with maximum denominator degree n and maximum numerator degree $m = K - n - 1$ in the points $z_0, \ldots, z_{K-1}$ to the function
  $$g(z) = \left(\prod_{j=0}^{J-1} (z - \eta_j) \right) f(z).$$
  3. Calculate the zeros, counted according to multiplicity, of the denominator polynomial of the interpolant computed in step 2.

Procedure (K) is essentially a statement of the derivative-free Kravanja–Van Barel method after discretization, applied to compute n poles. In its original formulation, z_k and w_k are, respectively, the nodes and weights of some quadrature rule, but there is nothing in the statement of the result that constrains them to be chosen in this way. Procedure (K) gives a way to apply the derivative-free Kravanja–Van Barel method—and, hence, the SS-H method—with *any* rational filter of the form we are considering. In Procedure (R), we compute the roots of the denominator polynomial of a linearized rational interpolant to a modified version of f that incorporates the zeros of the filter. The result we now prove asserts that these two methods are equivalent under one additional assumption.

Theorem 5.1. *The matrix H_n of Procedure* (K) *is nonsingular if and only if the denominator polynomial computed in Procedure* (R) *has degree exactly n. If these equivalent conditions hold, then Procedure* (K) *and Procedure* (R) *yield identical results in exact arithmetic in step 3: the eigenvalues computed in Procedure* (K) *are the same as the roots computed in Procedure* (R)*.*

The nonsingularity requirement precludes degenerate situations in which the pencil $H_n^< - \lambda H_n$ of Procedure (K) has infinite eigenvalues or the denominator of the interpolant in Procedure (R) has fewer than n roots.

Proof of Theorem 5.1. Let $\ell(z) = (z - z_0)\cdots(z - z_{K-1})$ be the node polynomial for the points $z_0, \ldots, z_K$ and $\nu_0, \ldots, \nu_{K-1}$ be the corresponding barycentric weights (see Section 1.2.2). Consider the Hankel matrices

$$\widehat{H}_n = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \cdots & h_n \\ \vdots & \vdots & & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix}, \qquad \widehat{H}_n^< = \begin{bmatrix} h_1 & h_2 & \cdots & h_n \\ h_2 & h_3 & \cdots & h_{n-1} \\ \vdots & \vdots & & \vdots \\ h_n & h_{n-1} & \cdots & h_{2n-1} \end{bmatrix},$$

where

$$h_j = \sum_{k=0}^{K-1} \nu_k z_k^j g(z_k), \qquad j = 0, \ldots, n-1.$$

One can show [34, 46], [73, Theorems 1.2.2 and 2.3.4] that the denominator polynomial computed in Procedure (R) has degree exactly n if and only if $\widehat{H}_n$ is nonsingular and that if this is so, the roots of this polynomial are given by solving the generalized eigenvalue problem for the pencil $\widehat{H}_n^< - \lambda \widehat{H}_n$.

Since H has poles at exactly the points z_k and J zeros, we have $H(z) = p(z)/\ell(z)$ for some polynomial p of degree J. Using the barycentric formula (1.4) to represent p in terms of its values at the points z_k and dividing through by $\ell(z)$, we find that

$$H(z) = \sum_{k=0}^{K-1} \frac{\nu_k p(z_k)}{z - z_k}.$$

It follows from the definition of H and the uniqueness of partial fraction representations that $w_k = \nu_k p(z_k)$.

As the zeros of p are exactly the zeros of H, we can factor p to obtain $p(z) = \alpha(z - \eta_0)\cdots(z - \eta_{J-1})$ for some nonzero constant α. As $g(z) = (p(z)/\alpha)f(z)$, it follows that $h_j = \mu_j/\alpha$ for each j. Thus, $\widehat{H}_n = (1/\alpha)H_n$, so $\widehat{H}_n$ is nonsingular if and only if H_n is, establishing the equivalence of the conditions stated at the end of the theorem. Moreover, $\widehat{H}_n^< - \lambda \widehat{H}_n = (1/\alpha)(H_n^< - \lambda H_n)$, and so Procedure (R) reduces to the same generalized eigenvalue problem as Procedure (K), establishing the claim that the two yield identical results in exact arithmetic. □

The preceding discussion casts Theorem 5.1 as a way to associate a rational interpolation problem with the use of a given rational filter, but we can also use it "in reverse" to determine the rational filter

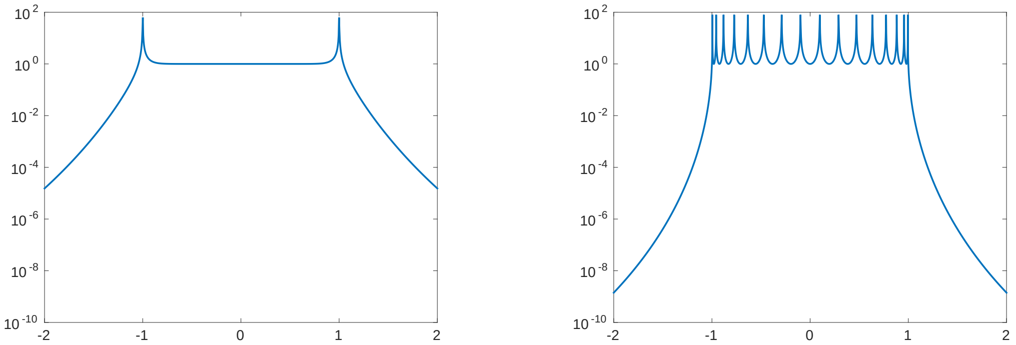


Figure 5.2: Magnitudes on $[-2, 2]$ of filters derived from rational interpolation in the Kth roots of unity (left) and K first-kind Chebyshev points on $[-1, 1]$ (right) for $K = 16$.

that underlies a given rational interpolation problem. In particular, we see that finding eigenvalues of A by computing the poles of a rational interpolant to $f(z) = u^*(A - zI)^{-1}v$, unmodified, in distinct points, is equivalent to applying the SS-H method with a rational filter that has poles at those same points *and that has no zeros*.

Thus, if the interpolation nodes are $z_0, \ldots, z_{K-1}$, it follows that the rational filter implicitly applied by rational interpolation is $H(z) = \alpha/\ell(z)$, where $\ell(z) = (z - z_0) \cdots (z - z_{K-1})$ is the node polynomial for the interpolation points and α is a nonzero constant. Expressing H in pole-residue form, we have

$$H(z) = \alpha \sum_{k=0}^{K-1} \frac{\nu_k}{z - z_k}, \tag{5.9}$$

where $\nu_0, \ldots, \nu_{K-1}$ are the barycentric weights corresponding to the interpolation nodes z_k. This can be seen, e.g., by taking p to be the constant polynomial $p(z) = \alpha$ in (1.4). Rational filters of this form have the property that they achieve the maximum possible asymptotic decay rate as $|z| \to \infty$ among all rational filters (5.8) with poles at the same points, an immediate consequence of the fact that the numerator polynomial is a constant. When computing eigenvalues, this can be a desirable property, as it ensures that the corresponding approximate spectral projector strongly attenuates components of unwanted eigenvectors that are far from the region of interest.

As an example, if the points z_k are the Kth roots of unity, then the corresponding node polynomial is $\ell(z) = z^K - 1$, and the resulting filter is $H(z) = 1/(z^K - 1)$, up to an arbitrary scaling factor. For the K first-kind Chebyshev points (1.14) in $[-1, 1]$, the node polynomial is $\ell(z) = T_K(z)/2^K$, where T_K is the Kth degree Chebyshev polynomial of the first kind, defined at the beginning of Section 1.2.4. Rescaling to eliminate the 2^K factor, the filter is $H(z) = 1/T_K(z)$.

Graphs of the absolute values of these filters on $[-2, 2]$ for $K = 16$ are shown in Figure 5.2. Note that while both of these filters ultimately decay as $O(z^{-K})$ as $|z| \to \infty$, the filter associated with the Chebyshev points decays much more rapidly immediately outside of $[-1, 1]$. This means that approximate projectors based on it will do a much better job than the filter based on roots of unity at suppressing components in the direction of eigenvectors corresponding to eigenvalues that lie outside but close to $[-1, 1]$. This advantage is another reason to consider using methods based on this filter instead of methods derived from discretized contour integrals.

5.5.4 Rayleigh–Ritz for the Chebyshev Interpolation Filter

Now that we have determined the filters that underlie methods based on rational interpolation, we can remove the instabilities observed in Section 5.4 by replacing rational interpolation with a Rayleigh–Ritz procedure based on the same filter. We proceed exactly as in the SS-RR method described in Section 5.5.1, but instead of discretizing (5.5) to project v onto the eigenspace of interest, we use the filter (5.9).

For the first-kind Chebyshev grid on $[-1, 1]$ of length K defined by (1.14), the barycentric weight ν_k corresponding to the point x_k is given by (1.15). After rescaling to eliminate the 2^{K-1} factor, the filter for rational interpolation on this grid may be written in pole-residue form as follows:

$$H(z) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{T_{K-1}(x_k)}{z - x_k}. \tag{5.10}$$

Thus, in the notation of Section 5.5.1, to calculate the vectors v_j that form the basis for the subspace we use when applying the Rayleigh–Ritz procedure, we compute

$$v_j = \frac{1}{K} \sum_{k=0}^{K-1} T_{K-1}(x_k) x_k^j (A - x_k I)^{-1} v, \qquad j = 0, \ldots, s - 1. \tag{5.11}$$

Note that the computation of these vectors requires the solution of exactly the same linear systems as the algorithm based directly on rational interpolation presented previously. This reformulation therefore does not require any significant additional work compared to the original method.

To show that this allows us to get around the instabilities described in Section 5.4, we rerun the same example from that section using the new procedure. This can be accomplished with the following MATLAB code:

```
A = diag([0:0.1:0.9, -10, 10]); I = eye(12); v = randn(12, 1);
K = 32; xk = chebpts(K, 1); wk = cos((K - 1)*acos(xk))/K;
V = zeros(12, 10); Y = zeros(12, K); e = ones(12, 1);
for k = 1:K, Y(:, k) = (A - xk(k)*I) \ v; end
for j = 0:1:9, V(:, j + 1) = sum((e*(wk.*xk.^j).’).*Y, 2); end
[Q, R] = qr(V, 0); D = sort(eig(Q’*A*Q));
```

The first three lines simply set up the problem and initialize a few variables for storing results. The fourth line solves the systems at each of the Chebyshev points, storing the results in `Y`, and the fifth line implements (5.11) to compute the basis, storing the results in `V`. In the last line, we form the projected eigenvalue problem as described in Section 5.5.1.

The results of running this code with the same random vector v used in the demonstration of Section 5.4 are shown Table 5.2. All 10 eigenvalues have been computed to full precision.

Computed eigenvalue	Error
0.000000000000000	1.76×10^{-16}
0.100000000000000	$0.00\times10^{+00}$
0.200000000000000	1.67×10^{-16}
0.300000000000000	2.78×10^{-16}
0.400000000000000	5.55×10^{-17}
0.500000000000000	2.78×10^{-16}
0.600000000000000	4.44×10^{-16}
0.699999999999999	1.11×10^{-15}
0.800000000000000	1.11×10^{-16}
0.900000000000002	2.11×10^{-15}

Table 5.2: Results of applying the reformulated method based on a Rayleigh–Ritz procedure to the test problem of Section 5.4. All eigenvalues are computed to full accuracy. No instabilities are observed.

5.5.5 Contour Integral Derivation of the Chebyshev Filter

While we arrived at the filter for rational interpolation in Chebyshev points via the equivalence established in Theorem 5.1, it is worth observing that it can also be obtained as a limit of filters derived from discretized contour integrals taken over certain ellipses that enclose the interval $[-1, 1]$. Before proceeding, we pause to outline an argument that shows this is the case.

Let D_r be the open disc in $\mathbb{C}$ with center 0 and radius $r > 1$. The ellipses we consider are the Bernstein ellipses that have the points ±1 as their foci. We recall from Section 1.2.6 that for $r > 1$, the Bernstein ellipse E_r is the image $J(D_r)$ of D_r under the Joukowski map $J(z) = (z + z^{-1})/2$. As $r \to 1$ from above, these ellipses collapse down to $[-1, 1]$.

If $\psi_0, \ldots, \psi_{K-1}$ are any K points that are equally spaced on the unit circle, then by transforming the integral (5.6), taken over ∂E_r, into one over ∂D_r via a change of variables using J and discretizing the result using the trapezoidal rule in these points, we obtain

$$\frac{1}{2\pi i}\int_{\partial E_r}(A - zI)^{-1}\,dz \approx \frac{1}{K}\sum_{k=0}^{K-1}(r\psi_k)\big(A - J(r\psi_k)I\big)^{-1}J'(r\psi_k).$$

Using the fact that $J'(z) = (z - z^{-1})/(2z)$ and letting $r \to 1$, one can show via straightforward computation that the filter obtained is

$$H(z) = \frac{1}{K}\sum_{k=0}^{K-1}\frac{i\,\mathrm{Im}\,\psi_k}{z - \mathrm{Re}\,\psi_k}.$$

Let $\psi_k = x_k + iT_{K-1}(x_k)$, $0 \le k \le K-1$. Another computation shows that $\psi_k^K = i$ for each k, so these points are the Kth roots of i and hence are equally spaced on the unit circle. Our filter then becomes

$$H(z) = \frac{i}{K}\sum_{k=0}^{K-1}\frac{T_{K-1}(x_k)}{z - x_k},$$

which is the same as (5.10), apart from a factor of i.

5.5.6 General Rational Filters and Remarks on the Literature

The use of a Rayleigh–Ritz procedure in conjunction with a rational filter for computing eigenvalues is not a new idea. For example, the shift-and-invert Arnoldi method [117], applied with a given shift σ, can be thought of as filtering the spectrum with powers of the function $1/(z-\sigma)$. This observation was extended by Ruhe to filters generated by arbitrary rational functions with his introduction of rational Krylov methods [114] in the 1980s. Contour integral methods and the method we discuss here are a particular type of rational Krylov method: they construct a rational Krylov subspace using the quadrature nodes or, more generally, the filter poles as shifts and then extract approximate eigenpairs from this subspace using the Rayleigh–Ritz technique. They differ from more traditional rational Krylov methods in the mechanics of how they build the subspace. Unlike the rational Arnoldi algorithm, which applies each shift to the starting vector in succession, these methods apply the shifts simultaneously and then take linear combinations to form the result. Doing things the latter way makes the method easier to parallelize, but all the shifts must be chosen in advance. In contrast, a method that employs the former approach can choose the shifts adaptively [30, 139].

We arrived at the concept of a rational filter by considering how discretized contour integrals and rational interpolation act on the resolvent, but it is not necessary to proceed in this way. It is equally possible to begin directly with (5.8) and ask how to choose the poles z_k and residues w_k to construct an effective filter. Eigenvalue algorithms based on rational filters have been explored extensively from this viewpoint in the Japanese literature by Murakami [84, 85, 86, 87, 88, 89], who refers to them as filter diagonalization methods. The term "filter diagonalization" comes from a class of closely related algorithms introduced by Neuhauser in the 1990s for calculating eigenstates of quantum mechanical systems that lie in a given energy interval [90, 91, 132, 142].

In [85], Murakami considers the problem of computing all the eigenvalues within a given real interval of a matrix pencil $A - \lambda B$, where A and B are both real symmetric and B is positive-definite. After introducing the concept of filter diagonalization methods, he discusses the result mentioned in Section 5.5.3 that one can obtain a filter $H(z)$ that decays as rapidly as possible as $|z| \to \infty$ by taking the residues to be the barycentric weights of the corresponding poles (though he does not appear to use exactly this language). Murakami actually proposes the use of the reciprocal Chebyshev filter that we have been considering, justifying it using the minimality properties of Chebyshev polynomials [133]. Ultimately, however, he sets it aside in favor of filters that have no poles on the real axis (and hence require complex arithmetic to implement), owing to the potential for numerical instabilities that can arise when one of the eigenvalues sought lies close to one of the filter poles. Murakami proposes a fix for these instabilities in [86], which we will discuss in Section 5.6.1.

In [87, 88, 89], Murakami goes on to consider the use of rational filters based on the four "classic" filter types used in analog circuit design [100]: Butterworth, Chebyshev, inverse Chebyshev,[4] and elliptic. Each of these filter types satisfies a different optimality condition with respect to certain criteria, and hence each may be expected to perform particularly well in certain circumstances. All of them have poles located off the real axis. Of these filters, the elliptic (also called Cauer or Zolotarev) filter is especially noteworthy because of its ability to attain a sharper transition across the boundary of the search interval than the other types. The price one pays for using this filter is that it does not decay to zero at infinity. This makes it well-suited to problems for which there are unwanted eigenvalues that lie outside but close to the interval of interest. On the other hand, if the desired portion of the spectrum is fairly well-separated from the rest, one will typically achieve greater accuracy by using a filter that decays at infinity. The use of elliptic filters in conjunction with the FEAST algorithm has been considered in [140] and is explored further in [53].

One can also consider filters derived from rational interpolation on an interval in points other than first-kind Chebyshev grids, i.e., filters which are reciprocals of polynomials other than Chebyshev polynomials. Recall that we based our choice of the first-kind Chebyshev points on their suitability for use as interpolation points, a property that stems from the fact that they cluster near the interval endpoints [133]. If one uses a Rayleigh–Ritz-based approach instead of one based on interpolation, one might imagine that this clustering property would be less important.

Nevertheless, we can still isolate two advantages to using the proposed filter based on the reciprocal Chebyshev polynomial. First, Chebyshev polynomials grow rapidly immediately outside of the interval $[-1, 1]$ compared to other polynomials of the same degree [112]. This means that filters based on their reciprocals will typically do a better job of suppressing unwanted eigenvalues outside but close to the interval of interest than will filters with the same number of poles that achieve the same asymptotic decay rate at infinity. We noted this advantage previously in Section 5.5.3 when comparing the proposed filter to the one derived from rational interpolation in roots of unity (recall Figure 5.2).

Second, the residues for the reciprocal Chebyshev filter, i.e., the barycentric weights for the first-kind Chebyshev points, are roughly uniform in magnitude, and hence the terms in the pole-residue expansion (5.10) are weighted roughly equally. Other point distributions may give rise to barycentric weights that do not have this property; for instance, the weights for equispaced points vary by factors which grow exponentially as the number of points increases [14]. Filters with the maximum asymptotic decay rate derived from such points may thus excessively weight the contributions from linear systems solved at some of the poles relative to others, potentially reducing accuracy.

[4] In spite of their names, the "Chebyshev" and "inverse Chebyshev" filters are not the same as the filters based on the reciprocals of Chebyshev polynomials considered here.

Original filter		Dropped pole	
Eigenvalue	Error	Eigenvalue	Error
0.049067674327428	7.6×10^{-17}	0.049067674327428	1.7×10^{-16}
0.100005648969141	5.6×10^{-06}	0.100000000000000	2.4×10^{-16}
0.200031541585367	3.2×10^{-05}	0.200000000000000	8.3×10^{-17}
0.300016487697101	1.6×10^{-05}	0.300000000000000	3.9×10^{-16}
0.411431339360006	1.1×10^{-02}	0.400000000000000	5.6×10^{-17}
0.502930792938639	2.9×10^{-03}	0.500000000000000	2.2×10^{-16}
0.600000216771745	2.2×10^{-07}	0.599999999999998	2.4×10^{-15}
0.700033492323077	3.3×10^{-05}	0.699999999999999	1.4×10^{-15}
0.800000015779351	1.6×10^{-08}	0.800000000000000	1.1×10^{-16}
0.900000000238297	2.4×10^{-10}	0.899999999999999	1.1×10^{-15}

Table 5.3: Eigenvalues and absolute errors for the example of Section 5.6.1 illustrating the handling of eigenvalues that fall extremely close to filter poles.

5.6 Practical Considerations

In the preceding sections, we have presented these methods in their simplest possible forms. In this section, we briefly discuss a few additional items which should be considered when realizing them in practice.

5.6.1 Eigenvalues Near Filter Poles

The method we have proposed based on the use of the reciprocal Chebyshev polynomial filter draws its strength from its placement of the filter poles within the interval of interest. While we have shown that this can be advantageous, it also has a potential drawback. If it happens that one of the eigenvalues of A, say λ, lies close to one of the filter poles, say, z_k, then $\|(A - z_k I)^{-1}\|$ will be large. Hence, the solutions to the linear systems at z_k will dominate those from the other poles, and the resulting filtered vectors will have large components in the direction of the eigenvectors of A corresponding to λ. If λ is sufficiently close to z_k, these components can overwhelm those in the directions of the other eigenvectors of A, degrading the accuracy of their computation and that of their corresponding eigenvalues, though λ itself will be computed highly accurately.

As an illustration, we consider the same problem from Section 5.4 but with the eigenvalue of A at 0 shifted to lie at the point $\cos(31\pi/64) + 10^{-14}$. Furthermore, instead of taking A to be diagonal, we set $A = Q^*DQ$, where D is a diagonal matrix of the specified eigenvalues and Q is a randomly generated 12×12 orthogonal matrix. Since $\cos(31\pi/64) \approx 0.049$ belongs to the 32-point first-kind Chebyshev grid on $[-1, 1]$, A has an eigenvalue very close to one of the filter poles. The results of running the same code from Section 5.5.4 with this new A are presented in the left half of Table 5.3. The eigenvalue near $\cos(31\pi/64)$ has been computed to full accuracy, while the other eigenvalues, especially those near the middle of the spectrum, have suffered badly.

As mentioned in Section 5.5.6, this phenomenon was noted by Murakami [85, 86], who refers to it as a "resonance problem." In [86], he presents two options for overcoming it: either drop the offending poles from the filter or shift them to lie somewhere else. The former is simpler and

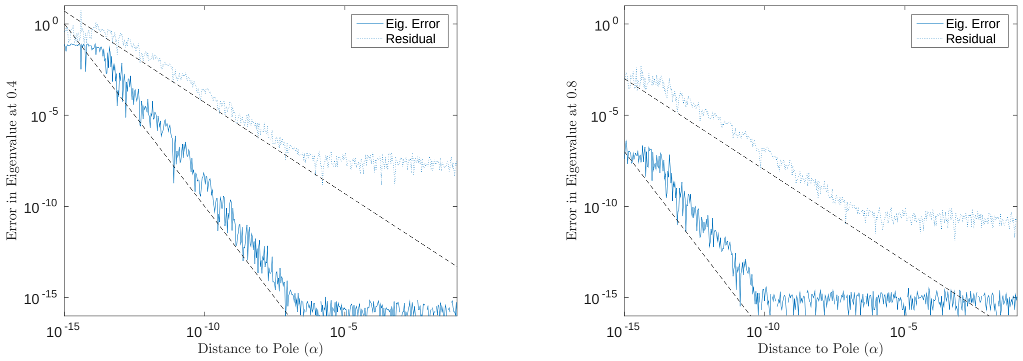


Figure 5.3: Behavior of residuals and errors in the eigenvalues at 0.4 (left) and 0.8 (right) as the eigenvalue near $\cos(31\pi/64)$ in the example considered in Section 5.6.1 gets close to that point.

needs no extra linear solves but requires that one accept the use of a filter with a slower asymptotic decay at infinity than that with which one began the computation. The latter does not have this disadvantage, since it keeps the total number of poles the same, but it is more expensive, requiring the solution of additional linear systems at the new poles. Murakami discusses these solutions in the context of non-Hermitian eigenvalue problems. Nevertheless, it is clear that they are also applicable in the Hermitian (and, in particular, the real symmetric) case, though he does not appear to mention this explicitly.

Under either option, one will need to recompute the filter residues to ensure that the resulting filter has the desired behavior. If one is working with filters of the type (5.9) and wishes to maintain the property that the recomputed filter has the maximum possible asymptotic decay rate at infinity, this amounts to calculating the barycentric weights for the new grid. Murakami provides explicit formulas for doing this in [86], which he phrases as updates to the weights for the original grid.

Before one can move to address this problem, however, one must first determine whether it has occurred. In [86], Murakami suggests looking at the factor by which application of the resolvent at a given filter pole (via the solution of the corresponding linear system) magnifies the norm of the starting vector and declaring the pole problematic if this factor exceeds a predetermined threshold. One could also consider simply looking at the computed eigenvalues and seeing if any are close to the filter poles; however, doing this may be subtle, as it is not clear how close an eigenvalue must be to a pole to be considered "too close," and the induced error may vary greatly between eigenvalues, as the results in the left half of Table 5.3 demonstrate.

Nevertheless, we note that the distance which qualifies as "too close" may be even smaller than one may expect at first, thanks to the Rayleigh quotient effect: an $O(\varepsilon)$ error in an approximation to an eigenvector induces an $O(\varepsilon^2)$ error in the Rayleigh-quotient estimate of the corresponding eigenvalue [134]. If an eigenvalue is a relative distance α from a pole, then since the terms in the partial fraction expansion for the filter function are inverse linear, we might anticipate induced errors in the nonresonant eigenvectors on the order of $\varepsilon_m \alpha^{-1}$, where ε_m is the machine epsilon, yielding errors in the nonresonant eigenvalues on the order of $\varepsilon_m^2 \alpha^{-2}$. Thus, the errors in the eigenvalues due to the resonance effect fall off rapidly as the distance α increases. This suggests that if one seeks to compute only eigenvalues and not eigenvectors, then this problem may be less of a cause for concern than it may seem initially.

These facts are illustrated in the plots of Figure 5.3, which we produced by varying the distance α between the eigenvalue near $\cos(31\pi/64)$ and that point in the numerical example just discussed. The solid lines show the errors in the approximations to the eigenvalues at 0.4 (left) and 0.8 (right), and the dotted lines show the associated residuals, a measure of the error in the corresponding eigenvectors. The dashed lines in both plots illustrate the decay rates of $O(\alpha^{-1})$ for the residuals and $O(\alpha^{-2})$ for the eigenvalue errors. The error in the eigenvalue at 0.4 reaches the level of machine precision for α larger than around 10^{-7}, while for 0.8, this occurs for α as small as 10^{-10}. The residuals at these values are relatively large; however, if one is not concerned with approximating the eigenvectors, this is not a problem.

Returning to our original numerical example and noting that the solution to the linear system at the pole at $\cos(31\pi/64)$ has a norm larger than that of the starting vector by a factor on the order of 10^{13}, we conclude that a resonance problem has occurred and decide to correct it by simply dropping this pole from the filter. Recalculating the barycentric weights using (1.5) and applying the new filter, we obtain the results in the right half of Table 5.3. All eigenvalues have now been obtained to full precision.

5.6.2 Determining the Subspace Size

In our descriptions above, we have made the assumption that the number s of eigenvalues within the region of interest is known in advance and taken the dimension d of the subspace used for the Rayleigh–Ritz procedure to be equal to this number. In practice, s will have to be calculated or estimated in some way. For modest-size Hermitian eigenvalue problems, this can be accomplished using the "spectrum slicing" technique based on Sylvester's law of inertia and the LDL^* decomposition [101]. For larger problems, stochastic techniques have been developed that use contour integrals to estimate the trace of the spectral projector onto the region of interest [27, 43]. It is not immediately obvious how to extend these latter techniques to work with arbitrary rational filters because they rely on the filter taking the same (or approximately the same) value at every eigenvalue in the search region. Further investigation is needed.

Actually, all that is required is that $d \geq s$, and it is often advantageous to take d to be larger than s even when s is known exactly. This is especially helpful if the spectrum of A is not well-separated so that there are eigenvalues close to but outside the search region that may not be adequately suppressed by the filter. Increasing d has the effect of incorporating eigenvectors corresponding to such eigenvalues within the search subspace so that they are computed instead of ignored. This results in a larger projected eigenvalue problem and the need to solve additional linear systems if block methods are used (see the next subsection). Nevertheless, the cost is typically far less than would be required to solve the problem by adding more filter poles to achieve the desired level of suppression. In [104], Polizzi suggests choosing $d \geq 1.5s$ as a rule of thumb.

While taking d to be too small yields poor results due to the influence of unwanted eigenvectors, if d is too large, one will typically find that some of the computed eigenpairs are spurious and need to be discarded. One way to deal with these is to check the residuals of the computed eigenpairs and eliminate those which are large. This is essentially what is done in the version 2.1 release of FEAST [130].

An alternative technique, proposed by Sakurai and coworkers [61, 62], uses the SVD to pare down the search subspace prior to solving the projected eigenvalue problem. The basis vectors for the subspace computed by applying the rational filter are gathered as columns into a matrix. One then computes the reduced SVD of this matrix and replaces the original basis vectors with the left singular vectors, omitting those corresponding to negligible singular values. This is very similar both in spirit and in execution to the techniques employed by `ratinterp` for eliminating spurious pole-zero pairs from rational interpolants briefly mentioned in Section 5.3.1. In [119], Sakurai and coworkers propose further that this technique can be used to help detect when one's initial choice of the subspace dimension is too small: if none of the singular values are negligible, a larger basis is probably needed.

5.6.3 Block Methods

If implemented exactly as described above, these methods cannot detect if an eigenvalue is derogatory, i.e., if it has geometric multiplicity greater than one. For the SS-H and rational interpolation methods, this follows from the fact that the scalarized resolvent (5.1) has only a simple pole at each of the eigenvalues of A, even if some of those eigenvalues have non-unit multiplicity. From the perspective of SS-RR, this occurs because the subspace is generated from the projected powers of A applied to a single initial vector v.

This problem can be addressed by using multiple starting vectors to build the subspace. Sakurai and coworkers introduced this technique for their algorithms in [61] and [62] under the name of the "block Sakurai–Sugiura method," while Polizzi used it from the outset in FEAST [104]. Doing this requires one to solve additional linear systems at each filter pole, but since systems corresponding to different starting vectors are independent, they can be solved in parallel.

Aside from being able to detect higher-multiplicity eigenvalues, an additional benefit to using multiple starting vectors is that it allows one to use fewer projected powers of A when building the subspace [62]. This is useful because higher powers weaken the filter. For instance, when using

(5.11) to apply the reciprocal Chebyshev polynomial filter we have been considering, the vector v_j is computed by filtering v with

$$H_j(z) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{x_k^j T_{K-1}(x_k)}{z - x_k} = \frac{z^j}{T_K(z)}, \qquad 0 \leq j \leq K-1,$$

where the second equality holds by (1.4). The higher the power j, the more slowly the filter decays as $|z| \to \infty$.

Allowing higher powers does, however, have the advantage of requiring fewer linear solves, so a balance must be struck. To date, there is no consensus about how this is best accomplished. In [119], Sakurai and coworkers propose the heuristic that the number of starting vectors be chosen so that the number of powers (including the zero power) employed to build the subspace is at most $K/4$. Polizzi, on the other hand, does not use higher powers at all in FEAST [104].

5.6.4 Outer Iteration

Finally, if the accuracy of the computed eigenpairs is not satisfactory, it can be improved by using a simple iterative procedure: just take the computed eigenvectors (or some linear combination(s) thereof) as starting vectors and repeat the process, filtering them to generate a new subspace and applying the Rayleigh–Ritz procedure again to get a new set of eigenpairs. This idea was proposed by Polizzi in [104] for FEAST, and it amounts to applying powers of the underlying filter or, equivalently, subspace iteration [130]. The disadvantage to doing this is that an additional set of linear solves is required for each iteration. Since the same filter is used each time, however, one can mitigate this cost by computing the LU factors of the resolvent at each filter pole during the first pass and then reusing them on subsequent passes.

5.7 Summary of the Proposed Algorithm

Taking into account some of the considerations discussed in the previous section, a practical version of the algorithm we have been discussing based on the reciprocal Chebyshev polynomial filter might proceed as follows:

1. Fix the matrix A and the search interval $[a, b]$. Choose the number K of filter poles and the maximum number M of powers of A that will be used to build the search subspace. Let $x_0, \ldots, x_{K-1}$ be the filter poles given by (1.14), rescaled to lie in $[a, b]$, and let $\nu_0, \ldots, \nu_{K-1}$ be the corresponding barycentric weights.

2. Compute or estimate the number s of eigenvalues of A in $[a, b]$, e.g., using Sylvester's law of inertia.

3. Decide the minimum dimension $d_{\min} \geq s$ of the search subspace and calculate the number L of starting vectors as $L = \lceil d_{\min}/M \rceil$. The search subspace dimension is then $d = ML$.

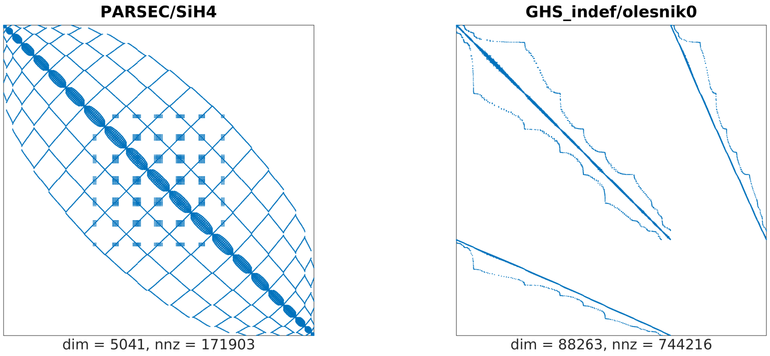


Figure 5.4: Sparsity patterns for the matrices of the test problems considered in Section 5.8.

4. Generate L random starting vectors, and gather them as columns into an $N \times L$ matrix V.

5. Factor $(A - x_k I) = L_k U_k$ at each filter pole x_k.

6. For each k, solve $L_k U_k W_k = V$ for W_k.

7. For each power $0 \leq j \leq M - 1$, calculate $R_j = \sum_{k=0}^{K-1} x_k^j \nu_k W_k$. Let $R = \begin{bmatrix} R_0 & \cdots & R_{M-1} \end{bmatrix}$.

8. Factor $R = XSY^*$ in an SVD. Let Q be the first r columns of X, where r is the number of singular values of R greater than a chosen tolerance.

9. Compute the eigenvalues λ_j and corresponding eigenvectors y_j of Q^*AQ. Discard those that lie outside the search interval. The remaining λ_j are approximations to the desired eigenvalues of A, and the $v_j = Qy_j$ are the corresponding approximate eigenvectors.

10. If an eigenvalue λ_j is too close to a filter pole, adjust the filter using one of the strategies outlined in Section 5.6.1 and return to step 7. Otherwise, proceed.

11. If greater accuracy is desired (as measured, e.g., by the size of the residuals of the approximate eigenpairs), reassign the L columns of V to be suitably chosen linear combinations of the vectors v_j, and return to step 6.

5.8 Numerical Examples

We close with a pair of examples illustrating the application of the proposed algorithm to larger problems. Our first test matrix A is the PARSEC/SiH4 matrix from the University of Florida Sparse Matrix Collection [24]. A is real symmetric with dimensions 5041×5041, and it has 171,903 nonzero entries. The sparsity pattern is shown in the left half of Figure 5.4.

We apply our method using the reciprocal Chebyshev polynomial filter with poles at 16 Chebyshev points of the first kind in $[-1, 1]$. Using the approach mentioned in Section 5.6.2 based on

Cheb. polynomial	Midpoint rule	Dense solver
$-\underline{0.995566528834}318$	$-\underline{0.995566528834}321$	$-\underline{0.995566528834}260$
$-\underline{0.6251586546426}99$	$-\underline{0.6251586546426}99$	$-\underline{0.6251586546426}78$
$-\underline{0.625158654640}977$	$-\underline{0.625158654640}979$	$-\underline{0.625158654640}871$
$-\underline{0.625158654638}943$	$-\underline{0.625158654638}941$	$-\underline{0.625158654638}861$
$\underline{0.034524054616}914$	$\underline{0.034524054616}922$	$\underline{0.034524054616}893$
$\underline{0.034524054618}691$	$\underline{0.034524054618}692$	$\underline{0.034524054618}906$
$\underline{0.0345240546202}45$	$\underline{0.0345240546202}46$	$\underline{0.0345240546202}10$
$\underline{0.045410512335}108$	$\underline{0.045410512335}109$	$\underline{0.045410512335}284$
$\underline{0.1769635801337}52$	$\underline{0.1769635801337}72$	$\underline{0.1769635801337}77$
$\underline{0.17696358013}7865$	$\underline{0.17696358013}7871$	$\underline{0.17696358013}8021$

Table 5.4: Selected eigenvalues for the PARSEC/SiH4 test problem of Section 5.8. Figures in each row that are the same in all three columns are underlined.

Cheb. polynomial	Midpoint rule	Dense solver
38 s	63 s	420 s

Table 5.5: Computation times on a single processor for the PARSEC/SiH4 test problem of Section 5.8. Use of multiple processors would allow a speedup of the first two figures by a factor of up to 16.

Sylvester's law of inertia, we find that A has 35 eigenvalues in $[-1, 1]$. We take the dimension d of our search subspace to be 72, roughly twice this value. We use a block Sakurai–Sugiura-like approach (see Section 5.6.3) using $16/4 = 4$ powers of A, following the rule of thumb from [119], and $72/4 = 18$ starting vectors. We do not employ any form of outer iteration (see Section 5.6.4).

Our computations were carried out in MATLAB R2013a on 1 core of a machine with twin 8-core Intel Xeon processors, clocked at 2.7 GHz, and 256 GB of RAM. The results are shown in Tables 5.4 and 5.5.

Table 5.4 displays a selection of the eigenvalues in $[-1, 1]$ (specifically, the lowest 10) computed by each of three methods. The results for the remaining eigenvalues are similar. The leftmost column shows those computed by the method just described. For comparison, the middle column shows approximations to the same eigenvalues computed using the same procedure but using a contour integral method based on applying the midpoint rule in 32 points (i.e., the trapezoidal rule in 32 roots of unity, shifted along the unit circle by an angle of $\pi/32$), exploiting the symmetry to require only 16 linear solves. This is the same as the number of solves required by the reciprocal Chebyshev filter, but they require complex arithmetic. Finally, this problem is small enough that dense methods can solve it in a reasonable amount of time, and the results of using MATLAB's built-in `eig` function are shown in the rightmost column. Digits in each eigenvalue that were computed the same for all three methods are underlined. All methods agree to at least 10 digits on all eigenvalues.

For our second example, we take A to be the GHS_indef/olesnik0 test matrix, also from the University of Florida Sparse Matrix Collection. This real symmetric matrix has dimensions 88,263 × 88,263 and 744,216 nonzero entries. Its sparsity pattern is plotted in the right half of Figure 5.4.

We search for the eigenvalues of A in $[1.005, 1.010]$, again using the reciprocal Chebyshev polynomial filter with poles at 16 Chebyshev points of the first kind. Employing Sylvester's law, we find

Cheb. polynomial		Midpoint rule		`eigs`	
Eigenvalue	Residual	Eigenvalue	Residual	Eigenvalue	Residual
$\underline{1.00504528402028}4$	8.4×10^{-09}	$\underline{1.00504528402028}0$	2.6×10^{-10}	$\underline{1.00504528402028}3$	5.4×10^{-13}
$\underline{1.0051034905467}54$	3.4×10^{-10}	$\underline{1.0051034905467}47$	4.7×10^{-10}	$\underline{1.0051034905467}46$	7.5×10^{-13}
$\underline{1.0052421279206}82$	8.1×10^{-10}	$\underline{1.0052421279206}72$	4.1×10^{-10}	$\underline{1.0052421279206}71$	4.7×10^{-13}
$\underline{1.00537930598631}8$	4.4×10^{-09}	$\underline{1.00537930598631}5$	3.5×10^{-10}	$\underline{1.00537930598631}4$	2.0×10^{-12}
$\underline{1.0054412884877}31$	5.0×10^{-09}	$\underline{1.0054412884877}25$	2.9×10^{-10}	$\underline{1.0054412884877}26$	5.3×10^{-13}
$\underline{1.00550225120147}6$	4.1×10^{-09}	$\underline{1.00550225120147}0$	4.4×10^{-10}	$\underline{1.00550225120147}0$	2.4×10^{-13}
$\underline{1.00558853719623}3$	2.1×10^{-10}	$\underline{1.00558853719623}1$	2.6×10^{-10}	$\underline{1.00558853719623}3$	3.4×10^{-13}
$\underline{1.00569118780895}8$	4.2×10^{-09}	$\underline{1.00569118780895}1$	3.3×10^{-10}	$\underline{1.00569118780895}0$	1.8×10^{-12}
$\underline{1.0058756983410}51$	1.3×10^{-09}	$\underline{1.0058756983410}42$	4.6×10^{-10}	$\underline{1.0058756983410}42$	1.6×10^{-12}
$\underline{1.0062544464068}22$	1.8×10^{-09}	$\underline{1.0062544464068}18$	5.4×10^{-10}	$\underline{1.0062544464068}22$	1.1×10^{-12}

Table 5.6: Selected eigenvalues and residuals for the GHS_indef/olesnik0 test problem of Section 5.8. Figures in the eigenvalues that are the same for all three methods are underlined.

Cheb. polynomial	Midpoint rule	`eigs`
42 s	61 s	11 s

Table 5.7: Computation times on a single processor for the GHS_indef/olesnik0 test problem of Section 5.8. Again, the use of multiple processors would allow a speedup of the first two figures.

that there are 44 eigenvalues of A in this interval. We take the search subspace dimension to be 88 and use 22 starting vectors, again limiting the number of powers of A to 4. This time, we employ one step of outer iteration to refine the eigenpairs.

Ten of the computed eigenvalues and their 2-norm relative residuals (defined for an approximate eigenvalue λ and corresponding eigenvector v as $\|Av - \lambda v\|/\|Av\|$) are displayed in the first two columns of Table 5.6. As with the previous example, we have also computed the same eigenvalues using an equivalent contour integral method based on the midpoint rule that requires the same number of linear solves. Finally, we performed the computation a third time using MATLAB's `eigs` function, based on ARPACK [75], to search for 44 eigenvalues near 1.0075, the midpoint of the target interval. All three methods agree to at least 10 digits in the displayed eigenvalues. The maximum residual for all eigenvalues computed using the reciprocal Chebyshev polynomial filter, including those not displayed in the table, is 2.1×10^{-8}. For the midpoint rule, it is 2.2×10^{-7}. For `eigs`, it is 2.1×10^{-12}

Timings for each of the methods applied to this problem are given in Table 5.7. The values all include the time required to count the eigenvalues in the search interval using Sylvester's law. As before, the method based on the reciprocal Chebyshev polynomial is faster than the one based on the midpoint rule, though the speedup is closer to a factor of 1.5 for this problem instead of 2. MATLAB's `eigs` is considerably faster than both; however, as in the previous example, since we are using only one core for our computations, we are not taking full advantage of the parallelism offered by the other methods. As before, these methods can be sped up by approximately a factor of 16, making their timings much more competitive.

Chapter 6

Conclusion

In the preceding chapters, we have presented three new contributions to the fields of interpolation and numerical analysis. In Chapter 2, we studied the numerical stability of the barycentric formula for trigonometric interpolation, finding that it possesses a subtle instability that we then showed how to correct. In Chapters 3 and 4, we studied the problem of trigonometric interpolation in perturbed equispaced grids. We proved that the Lebesgue constant for trigonometric interpolation in such grids grows at a rate that is at most algebraic, allowing us to prove theorems—to our knowledge, the first to appear in the literature—about the convergence of the interpolants and the corresponding quadrature scheme. While the results we were able to prove are likely not optimal, we formed conjectures, backed by extensive numerical evidence, as to what the optimal results may be. Finally, in Chapter 5, we showed how rational interpolation can be used as the basis for an algorithm for solving large-scale real symmetric eigenvalue problems that enjoys the same high degree of parallelism as existing methods based on contour integrals but uses only real arithmetic, saving a factor of two in computation time and storage.

There are many ways in which this work could be profitably continued. The conjectures in Chapter 3 present several direct opportunities for further work. While we believe that resolving these conjectures will be challenging, we also believe that they are tractable, and the process of resolving them may suggest further new ideas. Furthermore, all the statements in Chapter 3 possess analogues for polynomial interpolation, and examining these could form the basis for another potentially useful investigation. Another, more ambitious project might involve examining what extensions there are, if any, of the results in Chapter 3 for interpolation in two and three dimensions.

The material of Chapter 5 is also rife with opportunities for extension. We presented an example of a rational filter that is useful for symmetric (or, more generally, Hermitian) eigenvalue problems, but it is not clear that it is the "best" such filter, nor is it even clear what "best" might mean in this context. One possible avenue for research would be to explore techniques for designing rational filters both for general use and tailored for specific eigenvalue problems that arise in applications. More generally, one could consider designing filters for non-Hermitian or even nonlinear eigenvalue problems. There is some existing literature on designing filters for the latter [4, 5, 15, 137, 138], but there is still plenty of work to be done. Other possible projects include performing a careful study of the effects of using iterative methods to solve the linear systems involved (mandatory for very large problems) and developing extensions of rational filtering methods for solving multiparameter eigenvalue problems. The former would be especially useful from the point of view of a practitioner, while the latter would yield a new, highly parallel method for finding the solutions to systems of polynomial equations that lie in a given region in $\mathbb{C}^n$.

Bibliography

[1] L. V. Ahlfors, *Complex Analysis*, McGraw-Hill, Inc., 3rd ed., 1979.

[2] A. Aldroubi and K. Gröchenig, *Nonuniform sampling and reconstruction in shift-invariant spaces*, SIAM Rev., 43 (2001), pp. 585–620.

[3] A. Almansa, *Échantillonnage, interpolation et détection. Applications en imagerie satellitaire.*, Ph.D. thesis, École Normale Supérieure de Cachan, 2002.

[4] J. Asakura, T. Sakurai, H. Tadano, T. Ikegami, and K. Kimura, *A numerical method for nonlinear eigenvalue problems using contour integrals*, JSIAM Lett., 1 (2009), pp. 52–55.

[5] ——, *A numerical method for polynomial eigenvalue problems using contour integral*, Jpn. J. Ind. Appl. Math., 27 (2010), pp. 73–90.

[6] A. P. Austin, P. Kravanja, and L. N. Trefethen, *Numerical algorithms based on analytic function values at roots of unity*, SIAM J. Numer. Anal., 52 (2014), pp. 1795–1821.

[7] A. P. Austin and L. N. Trefethen, *Computing eigenvalues of real symmetric matrices with rational filters in real arithmetic*, SIAM J. Sci. Comput., 37 (2015), pp. A1365–A1387.

[8] A. P. Austin and K. Xu, *On the numerical stability of the second barycentric formula for trigonometric interpolation in shifted equispaced points.* To appear in IMA J. Numer. Anal., 2016.

[9] K. I. Babenko, *On conjugate functions*, Dokl. Akad. Nauk SSSR, 62 (1948), pp. 157–160.

[10] Z. Battles and L. N. Trefethen, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., 25 (2004), pp. 1743–1770.

[11] S. Bernstein, *Sur la meilleure approximation de $|x|$ par des polynomes de degrés donnés*, Acta Math., 37 (1914), pp. 1–57.

[12] J.-P. Berrut, *Baryzentrische Formeln zur Trigonometrischen Interpolation (I)*, Z. Angew. Math. Phys., 35 (1984), pp. 91–105.

[13] ——, *Baryzentrische Formeln zur Trigonometrischen Interpolation (II): Stabilität und Anwendung auf die Fourieranalyse bei ungleichabständigen Stützstellen*, Z. Angew. Math. Phys., 35 (1984), pp. 193–205.

[14] J.-P. Berrut and L. N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517.

[15] W.-J. Beyn, *An integral method for solving nonlinear eigenvalue problems*, Linear Algebra Appl., 436 (2012), pp. 3839–3863.

[16] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, *Julia: A fast dynamic language for technical computing*, (2012). arXiv:1209.5145v1 [cs.PL].

[17] J. P. Boyd, *Finding the zeros of a univariate equation: Proxy rootfinders, Chebyshev interpolation, and the companion matrix*, SIAM Rev., 55 (2013), pp. 375–396.

[18] L. Brutman, *On the Lebesgue function for polynomial interpolation*, SIAM J. Numer. Anal., 15 (1978), pp. 694–704.

[19] L. Carleson, *On convergence and growth of partial sums of Fourier series*, Acta Math., 116 (1966), pp. 135–157.

[20] E. W. Cheney, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966.

[21] E. W. Cheney and T. J. Rivlin, *A note on some Lebesgue constants*, Rocky Mountain J. Math, 6 (1976), pp. 435–440.

[22] J. H. Curtiss, *A stochastic treatment of some classical interpolation problems*, in Proceedings of the Fourth Berkeley Symposium on Mathematics and Probability, Volume 2: Contributions to Probability Theory, 1961, pp. 79–93.

[23] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 2nd ed., 1984.

[24] T. A. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Software, 38 (2011), pp. 1:1–1:25.

[25] C. de Boor and A. Pinkus, *Proof of the conjectures of Bernstein and Erdös concerning the optimal nodes for polynomial interpolation*, J. Approx. Theory, 24 (1978), pp. 289–303.

[26] C.-J. de la Vallée Poussin, *Sur la convergence des formules d'interpolation entre ordonnées équidistantes*, Acad. Roy. Belg. Bull. Cl. Sci., (1908), pp. 319–410.

[27] E. Di Napoli, E. Polizzi, and Y. Saad, *Efficient estimation of eigenvalue counts in an interval.* Submitted to Numer. Linear Algebra Appl., 2014.

[28] T. A. Driscoll and N. Hale, *Rectangular spectral collocation*, 36 (2016), pp. 108–132.

[29] T. A. Driscoll, N. Hale, and L. N. Trefethen, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.

[30] V. Druskin and V. Simoncini, *Adaptive rational Krylov subspaces for large-scale dynamical systems*, Systems Control Lett., 60 (2011), pp. 546–560.

[31] R. J. Duffin and J. J. Eachus, *Some notes on an expansion theorem of Paley and Wiener*, Bull. Amer. Math. Soc., 48 (1942), pp. 850–855.

[32] R. J. Duffin and A. C. Schaeffer, *A class of nonharmonic Fourier series*, Trans. Amer. Math. Soc., 72 (1952), pp. 341–366.

[33] M. Dupuy, *Le calcul numérique des fonctions par l'interpolation barycentrique*, C. R. Acad. Sci., 226 (1948), pp. 158–159.

[34] Ö. Eğecioğlu and Ç. K. Koç, *A fast algorithm for rational interpolation via orthogonal polynomials*, Math. Comp., 53 (1989), pp. 249–264.

[35] H. Ehlich and K. Zeller, *Auswertung der Normen von Interpolationsoperatoren*, Math. Ann., 164 (1966), pp. 105–112.

[36] P. Erdös, *Problems and results on the theory of interpolation. II*, Acta Math. Acad. Sci. Hung., 12 (1961), pp. 235–244.

[37] P. Erdös and P. Vértesi, *On the almost everywhere divergence of Lagrange interpolatory polynomials for arbitrary system of nodes*, Acta Math. Acad. Sci. Hung., 36 (1980), pp. 71–89. Erratum: [38].

[38] ——, *Correction of some misprints in our paper*, Acta Math. Acad. Sci. Hung., 38 (1981), p. 263.

[39] G. Faber, *Über die interpolatorische Darstellung stetiger Funktionen*, Jahresber. Dtsch. Math.-Ver., 23 (1914), pp. 192–210.

[40] G. Facciolo, A. Almansa, J.-F. Aujol, and V. Caselles, *Irregular to regular sampling, denoising, and deconvolution*, Multiscale Model. Simul., 7 (2009), pp. 1574–1608.

[41] L. Fejér, *Interpolation und konforme Abbildung*, Gött. Nachr., (1918), pp. 319–331.

[42] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann, *MPFR: A multiple-precision binary floating-point library with correct rounding*, ACM Trans. Math. Software, 33 (2007), pp. 13:1–13:15.

[43] Y. Futamura, H. Tadano, and T. Sakurai, *Parallel stochastic estimation method of eigenvalue distribution*, JSIAM Lett., 2 (2010), pp. 127–130.

[44] D. Gaier, *Lectures on Complex Approximation*, Birkhäuser, 1987.

[45] C. F. Gauss, *Theoria interpolationis methodo nova tractata*, in Werke, Vol. III, Dieterich, Göttingen, 1866, pp. 265–327.

[46] L. Gemignani, *Rational interpolation via orthogonal polynomials*, Comput. Math. Appl., 26 (1993), pp. 27–34.

[47] S. Goedecker, *Low complexity algorithms for electronic structure calculations*, J. Comput. Phys., 118 (1995), pp. 261–268.

[48] ——, *Linear scaling electronic structure methods*, Rev. Modern Phys., 71 (1999), pp. 1085–1123.

[49] P. Gonnet, S. Güttel, and L. N. Trefethen, *Robust Padé approximation via SVD*, SIAM Rev., 55 (2013), pp. 101–117.

[50] P. Gonnet, R. Pachón, and L. N. Trefethen, *Robust rational interpolation and least-squares*, Electron. Trans. Numer. Anal., 38 (2011), pp. 146–167.

[51] I. J. Good, *The colleague matrix, a Chebyshev analogue of the companion matrix*, Quart. J. Math., 2 (1961), pp. 61–68.

[52] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Elsevier, Burlington, MA, 7th ed., 2007.

[53] S. Güttel, E. Polizzi, P. T. P. Tang, and G. Viaud, *Zolotarev quadrature rules and load balancing for the FEAST eigensolver*, SIAM J. Sci. Comput., 37 (2015), pp. A2100–A2122.

[54] N. Hale and L. N. Trefethen, *New quadrature formulas from conformal maps*, SIAM J. Sci. Comput., 46 (2008), pp. 930–948.

[55] P. Henrici, *Barycentric formulas for interpolating trigonometric polynomials and their conjugates*, Numer. Math., 33 (1979), pp. 225–234.

[56] J. R. Higgins, *Sampling Theory in Fourier and Signal Analysis: Foundations*, Oxford University Press, Oxford, 1996.

[57] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2nd ed., 2002.

[58] ——, *The numerical stability of barycentric Lagrange interpolation*, IMA J. Numer. Anal., 24 (2004), pp. 547–556.

[59] E. Hlawka, *Interpolation analytishcer Funktionen auf dem Einheitskreis*, in Number Theory and Analysis, P. Turán, ed., Plenum, New York, NY, 1969, pp. 99–118.

[60] IEEE, *IEEE Standard for Floating-Point Arithmetic.* IEEE Std. 754-2008, 2008.

[61] T. Ikegami and T. Sakurai, *Contour integral eigensolver for non-Hermitian systems: A Rayleigh-Ritz-type approach*, Taiwanese J. Math., 14 (2010), pp. 825–837.

[62] T. Ikegami, T. Sakurai, and U. Nagashima, *A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method*, J. Comput. Appl. Math., 233 (2010), pp. 1927–1936.

[63] D. Jackson, *Über die Genauigkeit der Annäherung stetiger Funktionen durch ganze rationale Funktionen gegebenen Grades und trigonometrische Summen gegebener Ordnung*, Ph.D. thesis, University of Göttingen, 1911.

[64] ——, *On approximation by trigonometric sums and polynomials*, Trans. Amer. Math. Soc., 13 (1912), pp. 491–515.

[65] C. G. J. Jacobi, *Disquisitiones Analyticae de Fractionibus Simplicibus*, Ph.D. thesis, Univeristy of Berlin, 1825.

[66] M. I. Kadec, *The exact value of the Paley-Wiener constant*, Soviet Math. Dokl., 5 (1964), pp. 559–561.

[67] L. Kalmár, *Über Interpolation*, Mat. Fiz. Lapok, (1926), pp. 120–149.

[68] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, New York, NY, 1966.

[69] Y. Katznelson, *An Introduction to Harmonic Analysis*, Cambridge University Press, Cambridge, U.K., 3rd ed., 2004.

[70] O. Kis, *On the convergence of the trigonometrical and harmonical interpolation*, Acta Math. Acad. Sci. Hung., 7 (1956), pp. 173–200.

[71] V. A. Kotel'nikov, *On the transmission capacity of the "ether" and wire in electrocommunications*, in Material for the First All-Union Conference on Questions of Communication, *Izd. Red. Upr. Svyazi RKKA*, Moscow, 1933. Reprinted in Modern Sampling Theory: Mathematics and Applications, J. J. Benedetto and P. J. S. G. Ferreira, eds., Springer, New York, 2001, pp. 27–45.

[72] P. Kravanja and M. Van Barel, *A derivative-free algorithm for computing zeros of analytic functions*, Computing, 63 (1999), pp. 69–91.

[73] ——, *Computing the Zeros of Analytic Functions*, Springer, New York, 2000.

[74] C. Labreuche, *Problèmes Inverses en Diffraction d'Ondes Basés sur la Notion de Résonance*, Ph.D. thesis, Université de Paris IX, 1997.

[75] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

[76] E. Levin and E. B. Saff, *Potential theoretic tools in polynomial and rational approximation*, in Harmonic Analysis and Rational Approximation: Their Rôles in Signals, Control and Dynamical Systems, J.-D. Fournier, J. Grimm, J. Leblond, and J. R. Partington, eds., vol. 327 of Lecture Notes in Control and Information Science, Springer, Berlin, 2006, pp. 71–94.

[77] N. Levinson, *On non-harmonic Fourier series*, Ann. of Math. (2), 37 (1936), pp. 919–936.

[78] J. Marcinkiewicz, *Quelques remarques sur l'interpolation*, Acta Sci. Math. (Szeged), 8 (1936-7), pp. 127–130.

[79] W. F. Mascarenhas, *The stability of barycentric interpolation at the Chebyshev points of the second kind*, Numer. Math., 128 (2014), pp. 265–300.

[80] W. F. Mascarenhas and A. P. de Camargo, *The effects of rounding errors in the nodes on barycentric interpolation*, (2016). To appear in Numer. Math.

[81] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, CRC Press, Boca Raton, 2003.

[82] E. Meijering, *A chronology of interpolation: From ancient astronomy to modern signal and image processing*, Proc. IEEE, 90 (2002), pp. 319–342.

[83] J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres, *Handbook of Floating-Point Arithmetic*, Birkhäuser, Boston, 2010.

[84] H. Murakami, *An experiment of the filter diagonalization method for the banded generalized symmetric-definite eigenproblem*, IPSJ SIG Technical Report 59 (2007-HPC-110), Information Processing Society of Japan, June 2007.

[85] ——, *A filter diagonalization method by the linear combination of resolvents*, IPSJ Trans. Adv. Comput. Syst., 49 (2008), pp. 66–87.

[86] ———, *The filter diagonalization method for the unsymmetric matrix eigenproblem*, IPSJ SIG Technical Report 43 (2008-HPC-115), Information Processing Society of Japan, May 2008.

[87] ———, *Experiments of filter diagonalization method for real symmetric definite generalized eigenproblems by the use of elliptic filters*, IPSJ SIG Technical Report 1 (2010-HPC-125), Information Processing Society of Japan, June 2010.

[88] ———, *Filter designs for the symmetric eigenproblems to solve eigenpairs whose eigenvalues are in the specified interval*, IPSJ Trans. Adv. Comput. Syst., 3 (2010), pp. 1–21.

[89] ———, *Optimization of bandpass filters for eigensolver*, IPSJ SIG Technical Report 3 (2010-HPC-124), Information Processing Society of Japan, February 2010.

[90] D. Neuhauser, *Bound state eigenfunctions from wave packets: Time → energy resolution*, J. Chem. Phys., 93 (1990), pp. 2611–2616.

[91] ———, *Time-dependent reactive scattering in the presence of narrow resonances: Avoiding long propagation times*, J. Chem. Phys., 95 (1991), pp. 4927–4932.

[92] D. J. Newman, *Rational approximation to $|x|$*, Michigan Math. J., 11 (1964), pp. 11–14.

[93] J. Ortega-Cerdà and K. Seip, *Fourier frames*, Ann. of Math. (2), 155 (2002), pp. 789–806.

[94] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelphia, 2001.

[95] J. C. Oxtoby, *Measure and Category*, Springer-Verlag, New York, 1971.

[96] R. Pachón, *Algorithms for Polynomial and Rational Approximation*, Ph.D. thesis, University of Oxford, 2010.

[97] R. Pachón, P. Gonnet, and J. Van Deun, *Fast and stable rational interpolation in roots of unity and Chebyshev points*, SIAM J. Numer. Anal., 50 (2012), pp. 1713–1734.

[98] R. Pachón and L. N. Trefethen, *Barycentric-Remez algorithms for best polynomial approximation in the chebfun system*, BIT, 49 (2009), pp. 721–741.

[99] R. E. A. C. Paley and N. Wiener, *Fourier Transforms in the Complex Domain*, vol. XIX of American Mathematical Society Colloquium Publications, AMS, New York, 1934.

[100] T. W. Parks and C. S. Burrus, *Digital Filter Design*, Wiley, New York, 1987.

[101] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[102] B. S. Pavlov, *Basicity of an exponential system and Muckenhoupt's condition*, Soviet Math. Dokl., 20 (1979), pp. 655–659.

[103] A. Pinkus, *Weierstrass and approximation theory*, J. Approx. Theory, 107 (2000), pp. 1–66.

[104] E. Polizzi, *Density-matrix-based algorithm for solving eigenvalue problems*, Phys. Rev. B, 79 (2009), pp. 11512:1–6.

[105] G. Pólya, *Über die Konvergenz von Quadraturverfahren*, Math. Z., 37 (1933), pp. 264–286.

[106] H.-J. Rack and M. Reimer, *The numerical stability of evaluation schemes for polynomials based on the Lagrange interpolation form*, BIT, 22 (1982), pp. 101–107.

[107] T. Ransford, *Potential Theory in the Complex Plane*, Cambridge University Press, Cambridge, 1995.

[108] E. Remes, *Sur le calcul effectif des polynomes d'approximation de Tchebichef*, C. R. Acad. Sci., 199 (1934), pp. 337–340.

[109] ——, *Sur un procédé convergent d'approximations successives pour déterminer les polynomes d'approximation*, C. R. Acad. Sci., 198 (1934), pp. 2063–2065.

[110] M. Riesz, *Eine trigonometrische Interpolationsformel und einige Ungleichungen für Polynome*, Jahresber. Dtsch. Math.-Ver., 23 (1914), pp. 354–368.

[111] ——, *Formule d'interpolation pour la dérivée d'un polynome trigométrique*, C. R. Acad. Sci., 158 (1914), pp. 1152–1154.

[112] T. J. Rivlin, *The Chebyshev Polynomials*, Wiley, New York, 1974.

[113] W. Rudin, *Real and Complex Analysis*, McGraw-Hill, Inc., Boston, 3rd ed., 1987.

[114] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.

[115] C. Runge, *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Z. Math. Phys., 46 (1901), pp. 224–243.

[116] H. Rutishauser, *Vorlesungen über numerische Mathematik*, vol. 1, Birkhäuser Verlag, Basel, 1976.

[117] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, SIAM, Philadelphia, 2011.

[118] E. B. Saff and V. Totik, *Logarithmic Potentials with External Fields*, Springer-Verlag, Berlin, 1997.

[119] T. Sakurai, Y. Futamura, and H. Tadano, *Efficient parameter estimation and implementation of a contour integral-based eigensolver*, J. Algorithms Comput. Technol., 7 (2013), pp. 249–269.

[120] T. Sakurai and H. Sugiura, *A projection method for generalized eigenvalue problems using numerical integration*, J. Comput. Appl. Math., 159 (2003), pp. 119–128.

[121] T. Sakurai and H. Tadano, *CIRR: A Rayleigh-Ritz type method with contour integral for generalized eigenvalue problems*, Hokkaido Math. J., 36 (2007), pp. 745–757.

[122] H. E. Salzer, *Coefficients for facilitating trigonometric interpolation*, J. Math. and Phys., 27 (1948), pp. 274–278.

[123] ——, *New formulas for trigonometric interpolation*, J. Math. and Phys., 39 (1960), pp. 83–96.

[124] ——, *Lagrangian interpolation at the Chebyshev points $x_{n,\nu} \equiv \cos(\nu\pi/n)$, $\nu = 0(1)n$; some unnoted advantages*, Comput. J., 15 (1972), pp. 156–159.

[125] A. Schönhage, *Fehlerfortpflanzung bei Interpolation*, Numer. Math., 3 (1961), pp. 62–71.

[126] C. E. Shannon, *Communication in the presence of noise*, Proc. IRE, 37 (1949), pp. 10–21.

[127] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[128] J. Szabados and P. Vértesi, *Interpolation of Functions*, World Scientific, Singapore, 1990.

[129] G. Szegö, *Orthogonal Polynomials*, vol. XXIII of American Mathematical Society Colloquium Publications, AMS, New York, 1959.

[130] P. T. P. Tang and E. Polizzi, *FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection*, SIAM J. Sci. Comput., (2014), pp. 354–390.

[131] W. J. Taylor, *Method of Lagrangian curvilinear interpolation*, J. Res. Nat. Bur. Stand., 35 (1945), pp. 151–155.

[132] S. Toledo and E. Rabani, *Very large electronic structure calculations using an out-of-core filter-diagonalization method*, J. Comput. Phys., 180 (2002), pp. 256–269.

[133] L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, Philadelphia, 2013.

[134] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[135] L. N. Trefethen and J. A. C. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.

[136] A. H. Turetskii, *The bounding of polynomials prescribed at equally distributed points*, Proc. Pedag. Inst. Vitebsk, 3 (1940), pp. 117–127.

[137] M. Van Barel, *Designing rational filter functions for solving eigenvalue problems by contour integration*, Linear Algebra Appl., 502 (2016), pp. 346–365.

[138] M. Van Barel and P. Kravanja, *Nonlinear eigenvalue problems and contour integrals*, J. Comput. Appl. Math., 292 (2016), pp. 526–540.

[139] R. Van Beeumen, K. Meerbergen, and W. Michiels, *A rational Krylov method based on Hermite interpolation for nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 35 (2013), pp. A327–A350.

[140] G. Viaud, *The FEAST Algorithm for Generalised Eigenvalue Problems*, M.Sc. thesis, University of Oxford, Oxford, UK, 2012.

[141] F. T. A. W. Wajer, G. H. L. A. Stijnman, M. Bourgeois, D. Graveron-Demilly, and D. van Ormondt, *Magnetic resonance image reconstruction from nonuniformly sampled k-space data*, in Nonuniform Sampling: Theory and Practice, F. Marvasti, ed., Kluwer Academic/Plenum Publishers, New York, 2001, pp. 439–478.

[142] M. R. Wall and D. Neuhauser, *Extraction, through filter-diagonalization, of general quantum eigenvalues or classical normal mode frequencies from a small number of residues or a short-time segment of a signal. I. Theory and application to a quantum-dynamics model*, J. Chem. Phys., 102 (1995), pp. 8011–8022.

[143] J. L. Walsh, *Note on polynomial interpolation to analytic functions*, Proc. Natl. Acad. Sci. USA, 19 (1933), pp. 959–963.

[144] J. L. Walsh, *Interpolation and Approximation by Rational Functions in the Complex Domain*, vol. XX of American Mathematical Society Colloquium Publications, AMS, Providence, 5th ed., 1969.

[145] E. Waring, *Problems concerning interpolations*, Philos. Trans. R. Soc. Lond., 69 (1779), pp. 59–67.

[146] M. Webb, L. N. Trefethen, and P. Gonnet, *Stability of barycentric interpolation formulas for extrapolation*, SIAM J. Sci. Comput., 34 (2012), pp. A3009–A3015.

[147] K. Weierstrass, *Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen*, Sitzungsberichte der Akademie zu Berlin, (1885), pp. 633–639 and 789–805.

[148] H. Weyl, *Über die Gleichverteilung von Zahlen mod. Eins.*, Math. Ann., 77 (1916), pp. 313–352.

[149] E. T. Whittaker, *On the functions which are represented by the expansions of the interpolation-theory*, Proc. Roy. Soc. Edinburgh, 35 (1915), pp. 181–194.

[150] G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen, *Extension of Chebfun to periodic functions*, SIAM J. Sci. Comput., 37 (2015), pp. C554–C573.

[151] K. Xu, *The Chebyshev points of the first kind*, Appl. Numer. Math., 102 (2016), pp. 17–30.

[152] K. Yao and J. B. Thomas, *On some stability and interpolatory properties of nonuniform sampling expansions*, IEEE Trans. Circuit Theory, 14 (1967), pp. 404–408.

[153] N. Young, *An Introduction to Hilbert Space*, Cambridge University Press, Cambridge, UK, 1988.

[154] R. M. Young, *An Introduction to Nonharmonic Fourier Series*, Academic Press, San Diego, CA, revised first ed., 2001.

[155] A. Zygmund, *Trigonometric Series*, Cambridge University Press, Cambridge, UK, 2nd ed., 1959.