

# Algorithms for trigonometric polynomial and rational approximation



Mohsin Javed

Wolfson College

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

MT 2016



Amal

## Acknowledgements

My first and foremost thanks goes to my mentor, my teacher, and my supervisor Nick Trefethen, without whose immeasurable support and help during the whole duration of my DPhil, this thesis would never have been possible. He has been a constant source of inspiration since the time I first met him in late 2010 when I started my master's degree under his guidance. Everyone knows that he is a brilliant mathematician but not everyone knows how remarkable a human being he is. I could not have asked for a better supervisor.

My special thanks goes to my friend and office mate for four years Anthony Austin with whom I shared a passion for mathematics and software. He carefully read all the chapters of the thesis and helped me correct countless errors and mistakes. I am also extremely thankful to Yuji Nakatsukasa, Oliver Sète and Silviu Filip for their very illuminating and insightful comments. My thanks also goes to my office mates, Hadrien Montanelli and Hrothgar for providing such a stimulating atmosphere for me at work. I would also like to thank all members of the Chebfun team, past and present, whom I have had the good fortune to work with. The Numerical Analysis Group at Oxford is a remarkable place to do research and I would like to immensely thank all members of the group, and the administrative staff, especially Lotti Ekert, Helen Taylor and Sandhya Patel.

My very special thanks to Behnam Hashemi for keeping my spirits high by reciting verses from the works of Omar Khayyam<sup>1</sup>. I will always cherish those soul-enriching poetry recitals.

I am extremely thankful to the European Research Council (ERC) for their very kind and generous financial support to carry out this work<sup>2</sup>.

Finally, I am thankful to Sarah and Amal, who believed in me and kept me going, and to my family back in Pakistan who stood by me despite being thousands of miles away.

---

<sup>1</sup>Khayyam was an 11th century Persian mathematician, philosopher and poet.

<sup>2</sup>This DPhil research was supported by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/20072013)/ERC grant agreement 291068. The views expressed in this thesis are not those of the ERC or the European Commission, and the European Union is not liable for any use that may be made of the information contained here.

## Abstract

This thesis presents new numerical algorithms for approximating functions by trigonometric polynomials and trigonometric rational functions. We begin by reviewing trigonometric polynomial interpolation and the barycentric formula for trigonometric polynomial interpolation in Chapter 1. Another feature of this chapter is the use of the complex plane, contour integrals and phase portraits for visualising various properties and relationships between periodic functions and their Laurent and trigonometric series. We also derive a periodic analogue of the Hermite integral formula which enables us to analyze interpolation error using contour integrals. We have not been able to find such a formula in the literature.

Chapter 2 discusses trigonometric rational interpolation and trigonometric linearized rational least-squares approximations. To our knowledge, this is the first attempt to numerically solve these problems. The contribution of this chapter is presented in the form of a robust algorithm for computing trigonometric rational interpolants of prescribed numerator and denominator degrees at an arbitrary grid of interpolation points. The algorithm can also be used to compute trigonometric linearized rational least-squares and trigonometric polynomial least-squares approximations.

Chapter 3 deals with the problem of trigonometric minimax approximation of functions, first in a space of trigonometric polynomials and then in a set of trigonometric rational functions. The contribution of this chapter is presented in the form of an algorithm, which to our knowledge, is the first description of a Remez-like algorithm to numerically compute trigonometric minimax polynomial and rational approximations. Our algorithm also uses trigonometric barycentric interpolation and Chebyshev-eigenvalue based root finding.

Chapter 4 discusses the Fourier-Padé (called trigonometric Padé) approximation of a function. We review two existing approaches to the problem, both of which are based on rational approximations of a Laurent series. We present a numerical algorithm with examples and compute various type  $(m, n)$  trigonometric Padé approximants.

All algorithms presented in the thesis are implemented in the **Chebfun** software system (<http://www.chebfun.org>).

# Contents

<b>0</b>	<b>Introduction</b>	<b>1</b>
0.1	A word about Fourier analysis . . . . .	1
0.2	What this thesis is not about . . . . .	2
0.3	Earlier work and this thesis . . . . .	3
0.4	Chebfun and trigonometric approximations . . . . .	3
0.5	Contributions of the thesis . . . . .	3
0.6	Applications . . . . .	5
<b>1</b>	<b>Trigonometric Polynomial Interpolation</b>	<b>7</b>
1.1	Introduction . . . . .	7
1.2	Trigonometric polynomials . . . . .	8
1.2.1	Roots of a trigonometric polynomial . . . . .	9
1.2.2	Various symmetries in trigonometric polynomials . . . . .	9
1.3	Trigonometric interpolation at arbitrary points . . . . .	10
1.3.1	Interpolation at an odd number of arbitrary points . . . . .	11
1.3.2	Interpolation at an even number of arbitrary points . . . . .	12
1.4	Trigonometric interpolation at equispaced points . . . . .	12
1.4.1	Interpolation at an odd number of equispaced points . . . . .	14
1.4.2	Interpolation at an even number of equispaced points . . . . .	14
1.5	Lagrange form of a trigonometric interpolant . . . . .	14
1.6	Barycentric interpolation formula . . . . .	16
1.6.1	A property of the barycentric weights . . . . .	20
1.7	Trigonometric and Laurent series . . . . .	21
1.8	Contour integrals of periodic analytic functions . . . . .	26
1.9	Cauchy's integral formula for periodic analytic functions . . . . .	28
1.10	Hermite integral formula for trigonometric interpolation . . . . .	31
1.11	Lebesgue constant for trigonometric interpolation . . . . .	35
1.12	Connection between algebraic and trigonometric polynomial interpolation . . . . .	36
1.13	Numerical examples using Chebfun . . . . .	39

1.14	Conclusion . . . . .	49
<b>2</b>	<b>Trigonometric Rational Interpolation and Least-Squares</b>	<b>50</b>
2.1	Introduction . . . . .	50
2.2	Trigonometric rational interpolation and least-squares approximation at arbitrary points . . . . .	51
2.2.1	The interpolation problem . . . . .	52
2.2.2	The linearized least-squares problem . . . . .	53
2.2.3	Existence and uniqueness of trigonometric rational interpolants . . . . .	53
2.2.4	The choice of basis . . . . .	55
2.2.5	Solution of the linearized interpolation and least-squares approximation . . . . .	56
2.2.6	Handling even and odd functions . . . . .	58
2.3	Numerical Computation . . . . .	58
2.3.1	Basic algorithm . . . . .	58
2.4	Numerical Examples . . . . .	59
2.4.1	Reproducing simple trigonometric rational functions . . . . .	59
2.4.2	Rational approximations of functions with singularities close to the interval of approximation . . . . .	61
2.4.3	Interpolants in the complex plane . . . . .	63
2.5	The problem of spurious poles . . . . .	69
2.5.1	Spurious poles in exact arithmetic . . . . .	70
2.5.2	Spurious poles in floating point arithmetic . . . . .	71
2.5.3	Singular values and spurious poles . . . . .	75
2.6	Robustness . . . . .	76
2.7	Trigonometric least-squares approximation . . . . .	80
2.8	Conclusion . . . . .	81
<b>3</b>	<b>Trigonometric Best Approximation</b>	<b>82</b>
3.1	Introduction . . . . .	82
3.2	Best trigonometric polynomial approximation . . . . .	84
3.2.1	The Haar condition . . . . .	85
3.3	The Remez algorithm for trigonometric polynomial approximation: Theoretical background . . . . .	86
3.3.1	A sufficient condition for optimality . . . . .	87
3.3.2	A necessary condition for optimality . . . . .	88
3.3.3	Characterization of the best trigonometric polynomial approximation . . . . .	89
3.4	The Remez (exchange) algorithm for trigonometric polynomial approximation: Implementation details . . . . .	91

3.4.1	Computation of the trial polynomial using trigonometric interpolation . . .	92
3.4.2	Finding the new reference . . . . .	93
3.5	Numerical examples of best trigonometric polynomial approximations . . . . .	95
3.6	Best trigonometric rational approximation . . . . .	99
3.7	A Remez algorithm for computing best trigonometric rational approximation . .	104
3.8	Numerical examples of best trigonometric rational approximations . . . . .	107
3.9	Conclusion and future work . . . . .	111
<b>4</b>	<b>Trigonometric Padé Approximation</b>	<b>113</b>
4.1	Padé approximation . . . . .	114
4.2	Laurent-Padé approximation . . . . .	115
4.3	Trigonometric Padé approximation . . . . .	120
4.4	Numerical algorithm . . . . .	121
4.5	Numerical examples . . . . .	122
4.6	Extended rational approximation to a Laurent series . . . . .	126
4.7	Conclusion and future work . . . . .	128
	<b>Appendices</b>	<b>129</b>
A	Appendix: Formula for the levelled reference error $h_k$ . . . . .	130
	<b>Bibliography</b>	<b>132</b>

# Chapter 0

## Introduction

### 0.1 A word about Fourier analysis

Fourier analysis is one of the most fascinating and well-understood branches of mathematics. Science, engineering, physics, statistics, mathematics (both pure and applied) and in short, countless aspects of our modern lives in the technologically driven world of today, all have innumerable examples where Fourier analysis is used. The discovery and invention of Fourier methods is one of the most revolutionary ideas of modern mathematics, with far reaching effects in subjects as varied as mathematical analysis, molecular biology, econometrics and weather forecasting.

What's *deep* about Fourier analysis is its use as a framework for thinking about mathematical problems in addition to being a powerful technique for solving them. But what is Fourier analysis? One can spend a lifetime answering this question. However, it's not too difficult to give a fairly concise (albeit non-exhaustive) description of what constitutes Fourier analysis. Just as vector analysis is concerned with decomposing a vector in  $\mathbb{R}^n$  into its components along orthogonal axes — in a nutshell — Fourier analysis is concerned with decomposing a suitably smooth function into simpler functions which are orthogonal to each other. In the continuous setting, this decomposition gives rise to Fourier transforms, while in the discrete setting, one obtains a Fourier series. Fourier transforms and Fourier series are, of course, closely related to each other.

The central idea of Fourier series is to decompose a well behaved function  $f$  on an interval of length  $2\pi$  as a sum of sinusoidal functions, that is,

$$f(\theta) = a_0 + \sum_{k=1}^{\infty} (b_k \sin k\theta + a_k \cos k\theta), \quad (0.1.1)$$

where  $a_k$  and  $b_k$  are the Fourier coefficients defined by the integrals

$$a_k = \int_0^{2\pi} f(\theta) \cos k\theta \, d\theta, \quad b_k = \int_0^{2\pi} f(\theta) \sin k\theta \, d\theta. \quad (0.1.2)$$

It is very common to approximate  $f$  by its truncated Fourier series,

$$f(\theta) \approx a_0 + \sum_{k=1}^N (b_k \sin k\theta + a_k \cos k\theta), \quad (0.1.3)$$

where  $N$  is a non-negative integer. The truncated Fourier series is not only mathematically convenient, it has very nice approximation properties as well. The truncated Fourier series of a function  $f$  gives the best approximation of  $f$  out of an appropriate function space in the  $L^2$  norm.

## 0.2 What this thesis is not about

The  $L^2$  approximation of a function is the usual starting point of Fourier analysis. This thesis is **not** about the  $L^2$  approximation of a function. The fundamental aim of the thesis is to explore other kinds of trigonometric approximations of  $f$ , and more importantly, how to compute these approximations. Our final goal is to develop numerical algorithms which make it possible to compute these approximations accurately, in a stable manner and as fast as possible.

Besides the  $L^2$  approximation, what are then the other approximations of  $f$ ? There are quite a few possibilities and the list can be rather long. However, what follows is a list of trigonometric approximations that we will explore in this thesis.

1. Trigonometric polynomial interpolation. (Chapter 1)
2. Trigonometric rational interpolation. (Chapter 2)
3. Trigonometric linearized rational least-squares approximation. (Chapter 2)
4. Trigonometric polynomial least-squares approximation. (Chapter 2)
5. Best trigonometric polynomial approximation in the  $\infty$ -norm. (Chapter 3)
6. Best trigonometric rational approximation in the  $\infty$ -norm. (Chapter 3)
7. Trigonometric Padé approximation<sup>1</sup>. (Chapter 4)

---

<sup>1</sup>More commonly known as Fourier-Padé approximation. However, following the naming convention of Chebfun, in this thesis we have tried to use the word *trigonometric* instead of *Fourier* wherever possible.

### 0.3 Earlier work and this thesis

The thesis also does not include earlier (joint) work on unifying the Euler–Maclaurin formula and the geometric convergence of the trapezoidal rule for numerical integration of periodic functions [54] and work related to Euler–Maclaurin and Newton–Gregory interpolants [56]. However, as happens so often in research, in hindsight, that apparently unrelated earlier work turned out to be the starting point of the present thesis. One of the fundamental aims of the paper [56] was to find a function approximation whose exact integral will give us the Euler–Maclaurin formula. As shown in [56], this turns out to be a *mixed* polynomial approximation which has a trigonometric polynomial part and an algebraic polynomial part. At that time, an exciting research direction was to develop contour integral proofs of various results related to the Euler–Maclaurin and barycentric formulas. In particular, this interest resulted in a contour integral expression for the error in interpolating a periodic analytic function by trigonometric polynomials. Later on, trigonometric barycentric formulas were crucial in guiding our interest in trigonometric rational approximations, which are a major part of the present thesis. In fact, as the list in the previous section shows, four out of the seven trigonometric approximations discussed in the thesis are rational.

### 0.4 Chebfun and trigonometric approximations

In this thesis, we will use [Chebfun](http://www.chebfun.org)<sup>2</sup> in every chapter. Chebfun is an open source software system<sup>3</sup> for numerical computing with functions [15], [86]. The fundamental idea of Chebfun is to approximate functions by polynomial interpolants in Chebyshev points [85]. Building on this solid foundation, Chebfun was later extended to approximate periodic functions by trigonometric polynomial interpolants in equispaced points [102]. This thesis extends this idea further by including algorithms for various other trigonometric approximations. While the thesis itself is not about Chebfun, all the algorithms developed here are available in Chebfun. This makes Chebfun a very powerful laboratory to conduct various numerical experiments involving different kinds of trigonometric approximations.

### 0.5 Contributions of the thesis

The contributions of the thesis are as follows. In Chapter 1 we develop a periodic analogue of the Hermite integral formula for evaluating the error in trigonometric polynomial interpolation

---

<sup>2</sup>[www.chebfun.org](http://www.chebfun.org)

<sup>3</sup>[www.github.com/chebfun](http://www.github.com/chebfun)

using contour integrals and summarize our result in Theorem 1.7. We have not been able to find an existing derivation of the periodic analogue of the Hermite integral formula in the literature.

In Chapter 2 we present a robust numerical algorithm for trigonometric rational interpolation. To our knowledge, this is the first attempt to numerically solve the problem of interpolating a function by a trigonometric rational function. The contribution of this chapter is presented in the form of a robust algorithm which allows us to numerically compute trigonometric rational interpolants of prescribed numerator and denominator degrees at an arbitrary grid of interpolation points. The algorithm can also be used to compute linearized least-squares approximations by trigonometric rational functions.

In Chapter 3 we solve the problem of computing the trigonometric minimax approximation of a periodic function, first in a space of trigonometric polynomials and then in a set of bounded trigonometric rational functions. The contribution of this chapter is presented in the form of a numerical algorithm which can compute both best trigonometric polynomial and best trigonometric rational approximations. To our knowledge, this is the first description of a numerical algorithm which solves this problem. Our algorithm also uses state of the art trigonometric barycentric interpolation and Chebyshev-eigenvalue based root finding.

In Chapter 4 we discuss the possibility of computing the trigonometric Padé, otherwise known as the Fourier-Padé approximation of a function. We review two approaches to the problem both of which map the problem of a trigonometric Padé approximation to the approximation of the corresponding Laurent series. We implement one of these in Chebfun and present numerical examples for computing type  $(m, n)$  trigonometric Padé approximants for both  $m < n$ , and  $m \geq n$ . A possible robust numerical algorithm is also under investigation and we believe it might result in an SVD based algorithm for computing these approximations.

Finally, another contribution of the thesis is the availability and integration of all of its numerical algorithms with the open source and freely available [Chebfun](https://www.chebfun.org) software system<sup>4</sup>.

The contributions of the thesis can be summarized in the following table.

Ch.	Trigonometric problem	Contribution	Chebfun command
1	Polynomial interpolation	New Hermite integral formula	
1	Polynomial interpolation	Arbitrary interpolation nodes	<code>chebfun.interp1(tk, fk, 'trig')</code>
1	Barycentric formulas	Chebfun implementation	<code>trigBary(t, fk, tk)</code>
2	Rational interpolation	New robust algorithm	<code>trigratinterp(f, m, n)</code>
2	Rational least-squares	New robust algorithm	<code>trigratinterp(f, m, n, N)</code>
2	Polynomial least-squares	SVD based implementation	<code>trigratinterp(f, m, 0, N)</code>
3	Best polynomial approximation	New algorithm	<code>trigremez(f, m)</code>
3	Best rational approximation	New algorithm	<code>trigremez(f, m, n)</code>
4	Padé approximation	New algorithm	<code>trigpade(f, m, n)</code>

Figure 1: Contributions of the thesis in a nutshell

<sup>4</sup><http://www.chebfun.org> or <https://www.github.com/chebfun>

## 0.6 Applications

The thesis has four chapters and in each chapter we discuss at least one trigonometric approximation. In this section, we present a number of possible applications. These applications are by no means exhaustive and we hope that in the near future there will be plenty of other theoretical and practical applications of the ideas developed and discussed in this thesis. The theory and practice of approximating periodic functions is a rich field and we see it only growing in interesting directions in the coming days.

Before we discuss the applications chapter-wise, we would like to sketch the overarching big picture of applications where algorithms of this thesis will prove useful. We are now in the midst of a data revolution and there is more data available than ever before. A lot of this data is periodic and there is interest in developing specialized methods to approximate periodic data more efficiently and accurately. We envisage that this is the biggest area where the methods developed in this thesis will be useful in the future. As an example, take a look at Figure 2.2, where we use a trigonometric rational interpolant to approximate a periodic function to machine precision using data at only three points. The example is artificial but it drives home the message that if we know something about the underlying periodic function, we can develop methods which would be very hard to beat.

Let us now look at each chapter and discuss the relevant applications.

The key contribution of Chapter 1 is Theorem 1.7 which presents the trigonometric Hermite integral formula. The contour integral formula is an excellent example of using *complex* analysis to understand *real* problems. We believe that the formula is rich with potential new research directions and will help us change our general perspective of trigonometric interpolation and approximation. In particular, armed with this formula, we are now in a position to develop the potential theory aspects of trigonometric polynomial interpolation. (See [85, Ch. 12-13] for an analogous presentation of potential theory for algebraic polynomial approximation.)

In Chapter 2, we develop algorithms for computing trigonometric rational interpolants and least-squares approximations. These interpolants and approximations can be extremely useful in problems of data science where the underlying periodic function is smooth on the interval of approximation but has nearby singularities in the complex plane. Another application area is the approximation of functions with poles. If the function is meromorphic and periodic, there are almost no algorithms which make use of the periodicity of the problem. Our numerical experiments in this chapter suggest that it is much more efficient to use trigonometric rational approximations in such cases.

In Chapter 3, we develop algorithms for best trigonometric polynomial and rational approximations. (By *best* here we mean best in infinity-norm.) We believe that no one has computed such best approximations before. Our ability to compute these approximations enables us to conduct all sorts of interesting numerical experiments and make conjectures and possibly help prove theorems. For example, Figure 3.7 presents numerical evidence that the error curves in best trigonometric approximations are sinusoidal and it might be possible to prove a theorem to this effect. In another example, we make a conjecture regarding the convergence rate of best approximations of the absolute value function (Figure 3.12). A possible practical application of the algorithms of this chapter is in the field of digital filters. In particular, our algorithms make it possible to compute optimal filters with non-symmetric frequency responses. At the moment, such filters are used very rarely but this can change in the future.

In Chapter 4, we present a numerical algorithm for computing the trigonometric Padé approximation<sup>5</sup> of a function. Padé approximations are extremely useful in various applications but there are no specialized algorithms for computing Padé approximations of periodic functions. Applications of Padé approximations include the harmonic oscillator, convergence acceleration, integral equations, electrical circuits and many more. In all of these applications, if the underlying function is periodic, our algorithm can be used advantageously. Another interesting application is in statistics. Various probability density functions blow up at the origin (the gamma distribution is one such example) and to conduct statistical experiments with such distributions, we think that trigonometric Padé approximations can prove far superior than the traditional Fourier methods.

---

<sup>5</sup>As mentioned earlier, this is more commonly known as the Fourier–Padé approximation.

# Chapter 1

## Trigonometric Polynomial Interpolation

### 1.1 Introduction

Approximating functions by interpolation is a fascinating subject and one that lies at the very heart of numerical analysis. In this chapter we will discuss interpolation of periodic functions by trigonometric polynomials. The chapter presents various trigonometric analogues of results derived for algebraic polynomials in *Approximation Theory and Approximation Practice (ATAP)* [85].

Let us first set the notation to be used not only in this chapter but also throughout the whole thesis.

We begin by setting up the domain on which we approximate periodic functions. Our algorithms and their corresponding Matlab codes will approximate functions on a finite interval  $[a, b]$  where  $a, b \in \mathbb{R}$  and  $a < b$ . However, it is a routine matter in numerical analysis to approximate a function by first linearly mapping it onto a standard interval, then computing its approximation on that standard interval and finally mapping the approximation back to the original domain of definition. Throughout this thesis we will make use of two such standard intervals while exploring various algorithms:

$$[-1, 1] \quad \text{and} \quad [0, 2\pi].$$

We will see that for the different algorithms we discuss, the explanation and the notation become more natural using one of these two intervals.

Mathematically, however, both  $[-1, 1]$  and  $[0, 2\pi]$  should be understood as *periodic intervals*. For example, if we are considering the interval  $[0, 2\pi]$ , then the points  $\theta = 0$  and  $\theta = 2\pi$  are *identified*: if we are interpolating a periodic function on  $[0, 2\pi]$  and the point  $\theta = 0$  is one of the points of interpolation, then the point  $\theta = 2\pi$  can not be another point of interpolation because

$\theta = 0$  and  $\theta = 2\pi$  are the same. Similarly, all assertions about a function defined on  $[0, 2\pi]$  hold for  $\theta = 0$  and  $\theta = 2\pi$ , just as they do for  $\theta \in (0, 2\pi)$ . In other words, the point 0 (or  $2\pi$ ) of the periodic interval  $[0, 2\pi]$  is in no way special.

Throughout this thesis, we will approximate periodic functions on an interval. We make the following definition of a *continuous periodic* function, which makes sure that the continuity of such a function is preserved across the boundary of a period.

**Definition 1.1** (*Continuous periodic function*). *A function  $f$  defined on an interval  $[a, b]$  is called continuous periodic if  $f$  is continuous on  $[a, b]$  and  $f(a) = f(b)$ . For convenience, we will consider such a function to be defined on the whole real line via the formula,*

$$f(\theta) = f(\theta + kT), \quad (1.1.1)$$

where  $k$  is an integer and  $T = b - a$ .

We will denote the vector space of functions continuous on  $[a, b]$  by  $\mathcal{C}([a, b])$  and the space of continuous periodic functions on  $[a, b]$  by  $\mathcal{C}_p([a, b])$ . Note that  $\mathcal{C}_p([a, b]) \subseteq \mathcal{C}([a, b])$ .

## 1.2 Trigonometric polynomials

A trigonometric polynomial is a linear combination of harmonically related sines and cosines. We can define a trigonometric polynomial on an arbitrary interval  $[a, b]$ . However, without loss of generality, we shall define a trigonometric polynomial of degree  $m$  as a function  $t : [0, 2\pi] \rightarrow \mathbb{C}$  of the form

$$t(\theta) = a_0 + \sum_{k=1}^m (a_k \cos k\theta + b_k \sin k\theta), \quad (1.2.1)$$

where  $\{a_k\}$  and  $\{b_k\}$  are complex numbers.

We shall denote the  $2m + 1$  dimensional vector space of all trigonometric polynomials of degree  $\leq m$  by  $\mathcal{T}_m$ . We note that the set

$$\{1, \cos \theta, \sin \theta, \dots, \cos m\theta, \sin m\theta\}$$

is a basis for  $\mathcal{T}_m$ .

Another basis for  $\mathcal{T}_m$  is given by the set

$$\{e^{-im\theta}, \dots, e^{-i\theta}, 1, e^{i\theta}, \dots, e^{im\theta}\},$$

and the same polynomial  $t$  can also be expressed as

$$t(\theta) = \sum_{k=-m}^m c_k e^{ik\theta}. \quad (1.2.2)$$

The coefficients  $a_k, b_k$  and  $c_k$  are related by the equations

$$a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k}) \quad 1 \leq k \leq m, \quad (1.2.3)$$

and  $a_0 = c_0$ . Equivalently, we have

$$c_k = \frac{a_k - ib_k}{2}, \quad c_{-k} = \frac{a_k + ib_k}{2}, \quad 1 \leq k \leq m. \quad (1.2.4)$$

### 1.2.1 Roots of a trigonometric polynomial

Let us now define the number of roots of a trigonometric polynomial  $t(\theta)$  as the sum of the multiplicities of its distinct real roots in the interval  $[0, 2\pi)$ . We have the following theorem [103].

**Theorem 1.1.** *The number of roots of a trigonometric polynomial  $t \in \mathcal{T}_m$ , where  $t \not\equiv 0$ , does not exceed  $2m$ .*

*Proof.* Since  $t \not\equiv 0$ , we can write

$$t(\theta) = \sum_{k=-m}^m c_k e^{ik\theta},$$

where not all  $c_k$  are zero. Following [103, Ch. X], we can write

$$e^{im\theta} t(\theta) = p(z), \quad (1.2.5)$$

where  $z = e^{i\theta}$  and  $p$  is an algebraic power polynomial of degree  $2m$  in  $z$ . If  $\theta = \eta$  is a root of  $t$  of order  $k$ , i.e., if

$$t(\eta) = t'(\eta) = \dots = t^{(k-1)}(\eta) = 0,$$

while

$$t^{(k)}(\eta) \neq 0,$$

we can differentiate (1.2.5) successively showing that  $\omega = e^{i\eta}$  is a root of  $p$  of order  $k$ , and conversely. Therefore, if the number of roots of  $t$  exceeds  $2m$ , then so does the number of roots of  $p$ , taking into account the multiplicity. Since  $p$  has degree  $2m$ , by the fundamental theorem of algebra,  $p \equiv 0$ , which implies  $t \equiv 0$ , contrary to the hypothesis of the theorem [103, Ch. X].  $\square$

### 1.2.2 Various symmetries in trigonometric polynomials

Let us now explicitly describe the relationship between the coefficients of a trigonometric polynomial and various symmetries that the trigonometric polynomial might exhibit.

1. Real-valued: If  $t$  is real-valued then all the coefficients in the sine-cosine expansion (1.2.1) are also real-valued. This can be proved by equating (1.2.1) with its complex conjugate and observing that the coefficients and their complex conjugates are equal and hence real. However, if we express the same trigonometric polynomial in the complex exponential basis as given by (1.2.2), then the coefficients  $c_k$  have conjugate symmetry:

$$c_k = \bar{c}_{-k}. \quad (1.2.6)$$

Again, this can be shown by taking the complex conjugate of (1.2.2) and rearranging the series.

2. Even: If  $t$  is even, then all coefficients  $b_k$  in the sine-cosine expansion (1.2.1) are zero and  $t$  reduces to a *cosine polynomial*. For even  $t$ , the coefficients in the complex exponential form (1.2.2) are also even:

$$c_k = c_{-k}. \quad (1.2.7)$$

3. Odd: If  $t$  is odd, then all coefficients  $a_k$  are zero in the sine-cosine expansion (1.2.1) and  $t$  reduces to a *sine polynomial*. For odd  $t$ , the coefficients in the complex exponential form (1.2.2) are also odd:

$$c_k = -c_{-k}. \quad (1.2.8)$$

Note in particular that  $c_0$  is zero in this case.

**Definition 1.2** (*Balanced trigonometric polynomial*). A trigonometric polynomial (1.2.2) is called balanced if the highest order coefficients satisfy the condition [50]:

$$c_m = c_{-m}. \quad (1.2.9)$$

Equivalently, a trigonometric polynomial (1.2.1) is balanced if the highest order sine term has zero coefficient:

$$b_m = 0. \quad (1.2.10)$$

### 1.3 Trigonometric interpolation at arbitrary points

Let  $N$  be a positive integer and let  $\{f_j\}$ ,  $0 \leq j \leq N-1$ , be a set of complex numbers, which may or may not come from sampling a function  $f(\theta)$  at the points  $0 \leq \theta_0 < \theta_1 < \dots < \theta_{N-1} < 2\pi$ . The interpolation problem is to find a trigonometric polynomial  $t \in \mathcal{T}_m$  which interpolates these data, i.e.,

$$t(\theta_j) = f_j \quad 0 \leq j \leq N-1. \quad (1.3.1)$$

For algebraic polynomials, it makes no difference whether the number of interpolation points is even or odd. Given  $n + 1$  values on  $n + 1$  distinct points, we can find a unique algebraic polynomial interpolant of degree  $n$ . In fact, such an interpolant can be expressed using the Lagrange formula:

$$p(x) = \sum_{j=0}^n \left( \prod_{\substack{k=0 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k} \right) f_j \quad (1.3.2)$$

Expressing the algebraic polynomial  $p$  in the monomial basis by writing  $p(x) = \sum_{j=0}^n a_j x^j$ , we see that the number of coefficients can be even or odd.

However, for a trigonometric polynomial  $t$ , the number of coefficients is always odd as can be seen from (1.2.1). Therefore, for trigonometric polynomial interpolation, we do have to take into account the parity of the number of points of interpolation. If the number of interpolation points is odd, a unique solution exists. However, both existence and uniqueness are no longer guaranteed if the number of interpolation points is even [82]. Let us discuss both these cases separately.

### 1.3.1 Interpolation at an odd number of arbitrary points

If  $N$  is odd, we take  $m = (N - 1)/2$  and the following theorem guarantees the existence and uniqueness of an interpolant through an odd number of interpolation points:

**Theorem 1.2** (*Trigonometric interpolation*). *Given  $2m + 1$  distinct points*

$$0 \leq \theta_0 < \theta_1 < \cdots < \theta_{2m} < 2\pi$$

*and arbitrary numbers  $f_0, f_1, \dots, f_{2m}$ , real or complex, there is a unique trigonometric polynomial  $t \in \mathcal{T}_m$  of the form (1.2.1) such that*

$$t(\theta_k) = f_k, \quad 0 \leq k \leq 2m. \quad (1.3.3)$$

*Proof.* See [103, Ch. X]. □

Note that for the above theorem, the distribution of interpolation points is completely arbitrary.

### 1.3.2 Interpolation at an even number of arbitrary points

If  $N$  is even, the situation is slightly complicated. Recall from (1.2.1) that a trigonometric polynomial  $t \in \mathcal{T}_m$  has  $2m + 1$  degrees of freedom. If we seek an interpolant  $t \in \mathcal{T}_m$  where  $m = N/2$  then  $t$  has  $2m + 1 = N + 1$  degrees of freedom and the problem is under-determined. Hence, if  $m = N/2$ , the interpolant is not unique. On the other hand, taking  $m = N/2 - 1$  allows  $t$  to have only  $2m + 1 = N - 1$  degrees of freedom and the problem is over-determined.

In practice, interpolation at an even number of arbitrary points is usually handled by taking  $m = N/2$  and then enforcing an extra condition on the interpolant. For example, given an even number of points, one can enforce the condition that  $t$  must be *balanced*. We will further discuss the case of an even number of interpolation points, but only in the context of interpolation at equispaced points.

## 1.4 Trigonometric interpolation at equispaced points

Let us now consider the problem of interpolating a function by a trigonometric polynomial on a grid of equispaced points. As mentioned earlier, any interval  $[a, b]$  can be scaled to any other interval  $[c, d]$ . Since we are mainly concerned with periodic functions, we will either consider the periodic interval  $[0, 2\pi]$  or Chebfun's<sup>1</sup> default periodic interval  $[-1, 1]$ .

Let  $N$  be a positive integer and consider  $N$  equally spaced points  $\{\theta_j\} \in [0, 2\pi)$ :

$$\theta_j = \frac{2\pi j}{N}, \quad 0 \leq j \leq N - 1. \quad (1.4.1)$$

These points are shown in Figure 1.1.

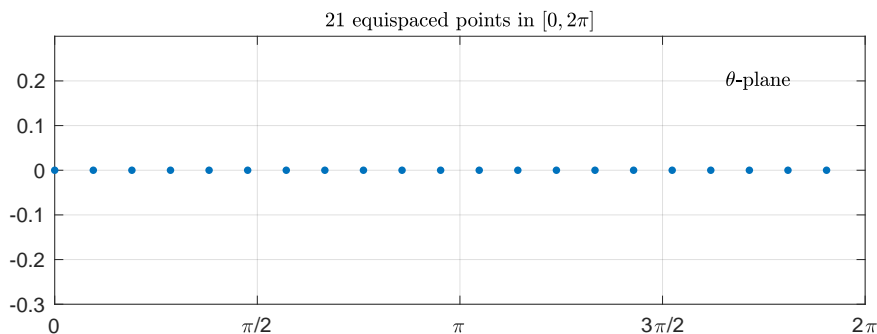


Figure 1.1: 21 equispaced points in  $[0, 2\pi)$ . Note that the left end-point is one of the equispaced points while the right is not.

<sup>1</sup>See Sec. 1.13 for more details about Chebfun.

We can think of these equispaced points as the arguments of  $N$  complex numbers  $\{z_j\}$  on the unit circle in the complex  $z$ -plane. The mapping

$$z = e^{i\theta} \tag{1.4.2}$$

takes the  $N$  equispaced points given by (1.4.1) to the  $N^{\text{th}}$  roots of unity on the unit circle. The following code uses the Chebfun command `trigpts` to compute these equispaced points and then maps them to the unit circle.

```
tt = trigpts(N, [0, 2*pi]);
zz = exp(1i*tt);
```

We can also see in Figure 1.2 that the mapping  $z = e^{i\theta}$  wraps the interval  $[0, 2\pi]$  around the circle and the two ends meet, making 0 and  $2\pi$  the same points in the  $z$ -plane.

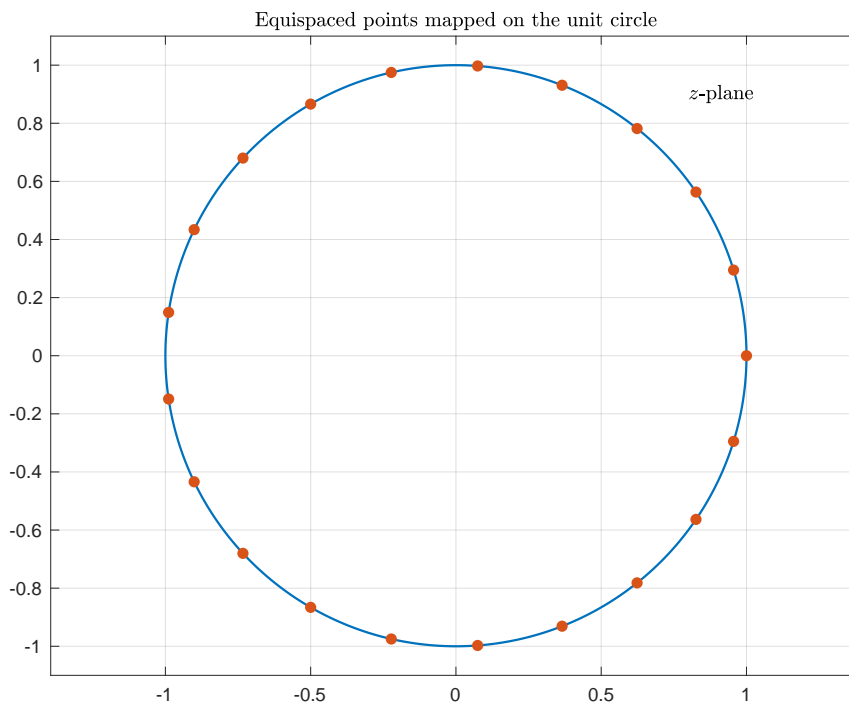


Figure 1.2: Equispaced points mapped on the unit circle

Trigonometric polynomial interpolants of smooth periodic functions at equispaced points have remarkable approximation properties [102, Sec. 3-4]. These interpolants also lead to geometrically convergent quadrature rules [30], [100]. In fact, equispaced grids are so ubiquitous that the term trigonometric polynomial interpolation is understood to mean trigonometric polynomial

interpolation at equispaced points. For the sake of completeness, we briefly mention the two cases of interpolation at an even and an odd number of equispaced points.

#### 1.4.1 Interpolation at an odd number of equispaced points

For an odd number of equispaced points, the interpolation problem has a unique solution. This follows from Theorem 1.2. Specifically, let  $N = 2m + 1$  be an odd positive integer and let  $\{f_j\}$ ,  $0 \leq j \leq N - 1$ , be a set of complex numbers, which may or may not come from sampling a function  $f(\theta)$  at  $N$  equispaced points in  $[0, 2\pi)$ . Then there exists a unique trigonometric polynomial  $t \in \mathcal{T}$  of degree  $m$  or less, which interpolates these data, i.e.,

$$t(\theta_j) = f_j \quad 0 \leq j \leq N - 1. \quad (1.4.3)$$

#### 1.4.2 Interpolation at an even number of equispaced points

The usual approach for even number of equally spaced interpolation points is to make use of *balanced* trigonometric polynomials as defined in definition 1.2, i.e. for  $N$  even, we will seek an interpolant  $t \in \mathcal{T}_m$ , where  $m = N/2$  and  $t$  is balanced. This choice has some advantages. For example, if we want to interpolate the saw-tooth function

$$f(\theta_k) = (-1)^k, \quad k = 0, \dots, N - 1, \quad N \text{ even,}$$

on an equispaced grid of points

$$\theta_k = \frac{2k\pi}{N}, \quad k = 0, 1, \dots, N - 1,$$

with  $m = N/2$ , the balanced trigonometric polynomial  $\cos(m\theta)$  is the function we seek. The complex-valued function  $e^{im\theta}$  also interpolates the data but it is not balanced. Indeed, the only real interpolant in this case is  $\cos(m\theta)$  [102], and forcing the interpolant to be balanced ensures that this is the interpolant we end up choosing.

### 1.5 Lagrange form of a trigonometric interpolant

We can express the trigonometric interpolant (1.3.3) in the so-called Lagrange form. If we let  $\ell_j$  be the trigonometric polynomial of degree  $m$  which takes the value 1 at  $\theta = \theta_j$  and vanishes at all of the remaining points  $\theta_k$ , then

$$t(\theta) = \sum_{j=0}^{2m} f_j \ell_j(\theta), \quad (1.5.1)$$

where  $\ell_j$  is given by

$$\ell_j(\theta_k) = \begin{cases} 1 & k = j, \\ 0 & k \neq j. \end{cases} \quad (1.5.2)$$

Corresponding to the interpolation nodes  $0 \leq \theta_0, \theta_1, \dots, \theta_{2m} < 2\pi$ , the trigonometric polynomials  $\ell_j, j = 0, 1, \dots, 2m$  are sometimes called the *trigonometric Lagrange polynomials*, also known as *fundamental trigonometric polynomials*. For example, we take  $m = 5$  and in Figure 1.3 plot the trigonometric Lagrange polynomial  $\ell_3$  on a grid of 11 equispaced points in  $[0, 2\pi)$ .

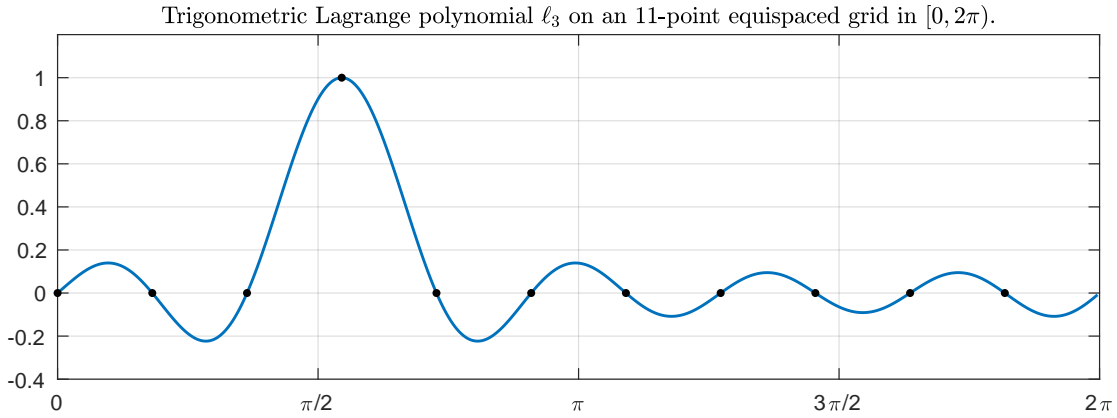


Figure 1.3: Trigonometric Lagrange polynomial  $\ell_3$  on a grid of 11 equispaced points in  $[0, 2\pi)$ .

It is easy to write down an explicit expression for  $\ell_j$  and there are various closely related forms. Following [103, Ch. X], we express the trigonometric Lagrange polynomials  $\ell_j$  by writing

$$\ell_j(\theta) = \prod_{\substack{k=0 \\ k \neq j}}^{2m} 2 \sin \frac{1}{2} (\theta - \theta_k) \bigg/ \prod_{\substack{k=0 \\ k \neq j}}^{2m} 2 \sin \frac{1}{2} (\theta_j - \theta_k). \quad (1.5.3)$$

The expression looks more complicated than it actually is. The denominator is just a scaling constant which makes sure that  $\ell_j$  takes the value 1 at the  $j^{\text{th}}$  node, i.e.,  $\ell_j(\theta_j) = 1$ . The numerator, and therefore the whole expression is a trigonometric polynomial of degree  $m$  with zeros at the right places. The form of (1.5.3) is well known and can be found in many textbooks as a standard representation for trigonometric Lagrange interpolants [50].

One explanation of the factor of 2 in each of the terms in (1.5.3) can be given using the Taylor expansion,

$$2 \sin \frac{1}{2} (\theta - \theta_j) = (\theta - \theta_j) + \frac{(\theta - \theta_j)^3}{24} + O((\theta - \theta_j)^5). \quad (1.5.4)$$

As a consequence, the Laurent expansion of  $1/(2 \sin \frac{1}{2}(\theta - \theta_j))$  has a residue of 1 at  $\theta = \theta_j$  [1]. This is mathematically convenient in certain contexts. Since this factor of 2 appears in each of the product term of the denominator as well as the numerator, it can be cancelled and following [16] one can write  $\ell_j$  as

$$\ell_j(\theta) = \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta - \theta_k) \bigg/ \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta_j - \theta_k). \quad (1.5.5)$$

This small change has no mathematical significance. However, numerically this change makes sure that both the denominator and the numerator appearing in the expression (1.5.5) are numbers with absolute values  $\leq 1$ .

The Lagrange form (1.5.1) leads to the trigonometric version of the barycentric interpolation formula, which we now discuss.

## 1.6 Barycentric interpolation formula

So far, we have looked at various forms of trigonometric polynomial interpolants but have not discussed how to evaluate such interpolants. This problem goes a long way back. In particular, Gauss addressed the topic and devised a method based on recursion to solve the problem of evaluating a trigonometric polynomial interpolant for equispaced data. This was what we now call the *Fast Fourier Transform*, abbreviated as FFT. Following the invention of electronic computers, the FFT was rediscovered and made popular by Cooley and Tukey in the 1960's [29]. (For a definitive reference, see [94].) The FFT approach to evaluate a trigonometric polynomial interpolant involves  $O(m \log m)$  work for computing the value of the interpolant at  $m$  equispaced points. The FFT is one of the most celebrated algorithms of computational mathematics and it works both ways; it can map the coefficients of a trigonometric polynomial to the values it takes on a grid of equispaced points and vice-versa it can map values back to coefficients. One major limitation of the FFT is that it is only applicable in the case of evaluation at equispaced points<sup>2</sup>.

As mentioned earlier, the FFT requires  $O(m \log m)$  work. However, analogous to the barycentric formula for evaluating algebraic polynomial interpolants [19], there is a trigonometric barycentric formula as well, which requires  $O(m)$  computations to evaluate the interpolant at  $m$  points. It also has the advantage of generalizing to sets of points other than the equispaced points. For a history and survey of barycentric formulae, see [19]. The trigonometric case was made popular by Berrut [16] and [17].

---

<sup>2</sup>There has been a lot of interest in developing a fast algorithm for the non-equispaced case. See [45], for example.

The trigonometric barycentric formula is derived from the trigonometric Lagrange form discussed earlier in the last section. We assume that we have  $2m + 1$  distinct interpolation nodes,

$$\theta_j \in [0, 2\pi), \quad j = 0, 1, \dots, 2m, \quad (1.6.1)$$

where  $m$  is a non-negative integer. Recall that a trigonometric polynomial interpolant can be written as a linear combination of trigonometric Lagrange polynomials, that is,

$$t(\theta) = \sum_{j=0}^{2m} f_j \ell_j(\theta). \quad (1.6.2)$$

Let us now look at the computational complexity of directly using the expression,

$$\ell_j(\theta) = \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta - \theta_k) \bigg/ \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta_j - \theta_k) \quad (1.6.3)$$

in (1.6.2) to evaluate  $t(\theta)$  for a given value of  $\theta$ . We need  $O(m)$  operations to evaluate  $\ell_j(\theta)$  and then another  $O(m)$  to add up all the terms in the summation (1.6.2). Therefore the total cost of evaluating  $t(\theta)$  at a given value of  $\theta$  is  $O(m^2)$ .

By a prudent rearrangement in (1.6.2), we can reduce this computational complexity to  $O(m)$ . First observe that for each value of  $j$ , the numerator in (1.6.3) is almost the same except that it is missing the factor  $\sin \frac{1}{2}(\theta - \theta_j)$ . Therefore, if we multiply each  $\ell_j(\theta)$  with its *missing factor*  $\sin \frac{1}{2}(\theta - \theta_j)$ , regardless of the value of  $j$ , we get the same function which has a root at every point of interpolation. We call this common function the *trigonometric node polynomial*  $\ell$  and define it by the equation,

$$\ell(\theta) = \prod_{k=0}^{2m} \sin \frac{1}{2}(\theta - \theta_k). \quad (1.6.4)$$

This allows us to rewrite (1.6.3) in the form

$$\ell_j(\theta) = \frac{\ell(\theta)}{2\ell'(\theta_j) \sin \frac{1}{2}(\theta - \theta_j)}. \quad (1.6.5)$$

We now define

$$\lambda_j := 1 \bigg/ \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta_j - \theta_k), \quad (1.6.6)$$

which can be compactly written as

$$\lambda_j = \frac{1}{2\ell'(\theta_j)}. \quad (1.6.7)$$

The trigonometric Lagrange polynomial (1.6.3) can now be expressed as

$$\ell_j(\theta) = \ell(\theta) \left( \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)} \right), \quad (1.6.8)$$

and the trigonometric Lagrange formula (1.6.2) takes the form

$$t(\theta) = \ell(\theta) \sum_{j=0}^{2m} \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)} f_j. \quad (1.6.9)$$

The above equation can also be called the *modified trigonometric Lagrange formula* or the *first form of the trigonometric barycentric interpolation formula* or the *type I trigonometric barycentric formula* [6]. According to Salzer [82], this formula was known to Gauss [36, p. 281].

The important computational observation is that the common factor  $\ell(\theta)$  in (1.6.9) makes sure that instead of a costly product of sines, each term inside the summation now has a single sine term of the form  $\lambda_j / \sin \frac{1}{2}(\theta - \theta_j)$ . If the numerical values of the weights  $\{\lambda_j\}$  are known, then given a value of  $\theta$ , one can use the form (1.6.9) to evaluate  $t(\theta)$  in  $O(m)$  operations<sup>3</sup>.

To compute all the weights  $\{\lambda_j\}$ , we can use the expression (1.6.6) and this will require  $O(m^2)$  operations. Note that (1.6.6) shows that the weights  $\{\lambda_j\}$  only need the grid points  $\{\theta_j\}$  for their computation, and in particular,  $\{\lambda_j\}$  can be computed independently from the point of evaluation  $\theta$  and the function values  $\{f_j\}$ . Therefore, if the interpolation grid  $\{\theta_j\}$  is fixed, we only need to compute the weights  $\{\lambda_j\}$  once. This is a huge advantage in practice because problems of interpolation are typically solved on a fixed grid of interpolation points. The function values to be interpolated or the point  $\theta$  where we wish to evaluate the interpolant can change but as long as the interpolation grid is fixed, we do not need to compute the weights  $\{\lambda_j\}$  again. In this scenario, the  $O(m^2)$  amount of work required for the computation of  $\{\lambda_j\}$ , reduces to a constant.

The story does not end here and an even bigger computational saving is possible for the most common case of interpolation in equispaced grids. For equispaced grids, one does not have to compute the weights  $\{\lambda_j\}$  even once since there are closed form formulas for their values (see Equation (1.6.17) and Theorem (1.6) below or Henrici [49]). The explicit formula (1.6.17) can be used to compute the weights  $\{\lambda_j\}$  for any equispaced grid of  $2m + 1$  points with  $m \geq 0$  in  $O(m)$  time.

Analogous to the second form of barycentric interpolation formula for algebraic polynomial interpolation, there is a *type II trigonometric barycentric formula* for trigonometric polynomial interpolation as well. To derive this formula, we make the key observation that the sum of all the trigonometric Lagrange polynomials  $\ell_j$  for  $j = 0, 1, \dots, 2m$ , should give us a trigonometric polynomial in  $\mathcal{T}_m$  which takes the value 1 at every point of the interpolation grid. The sum  $\sum_{j=0}^{2m} \ell_j(\theta)$  therefore interpolates the constant data 1. On the other hand, the constant trigonometric polynomial 1 also interpolates this data. Since trigonometric polynomial interpolants

---

<sup>3</sup>This is assuming that the sine function can be computed in constant time, which is in fact true.

are unique, we must have the equivalence

$$\sum_{j=0}^{2m} \ell_j(\theta) \equiv 1, \quad (1.6.10)$$

and using (1.6.8), gives us the identity

$$\ell(\theta) \sum_{j=0}^{2m} \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)} \equiv 1. \quad (1.6.11)$$

We now divide both sides of Equation (1.6.8) by this identity, which results in the cancellation of the common factor  $\ell(\theta)$  and we obtain,

$$\ell_j(\theta) = \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)} \bigg/ \sum_{k=0}^{2m} \frac{\lambda_k}{\sin \frac{1}{2}(\theta - \theta_k)}. \quad (1.6.12)$$

Finally, upon using the above expression of  $\ell_j$  in (1.6.2), we obtain the *second form of the trigonometric barycentric interpolation formula* or the *true trigonometric barycentric formula* or the *type II trigonometric barycentric formula*,

$$t(\theta) = \sum_{j=0}^{2m} \frac{\lambda_j f_j}{\sin \frac{1}{2}(\theta - \theta_j)} \bigg/ \sum_{j=0}^{2m} \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)}. \quad (1.6.13)$$

Following [85], we summarize our discussion of the type II trigonometric barycentric formula in the following theorem.

**Theorem 1.3** (*Trigonometric barycentric interpolation formula*). *The trigonometric polynomial interpolant through data  $\{f_j\}$  at  $2m + 1$  points  $\{\theta_j\} \in [0, 2\pi)$  is given by*

$$t(\theta) = \sum_{j=0}^{2m} \frac{\lambda_j f_j}{\sin \frac{1}{2}(\theta - \theta_j)} \bigg/ \sum_{j=0}^{2m} \frac{\lambda_j}{\sin \frac{1}{2}(\theta - \theta_j)}, \quad (1.6.14)$$

with the special case  $t(\theta) = f_j$  if  $\theta = \theta_j$  for some  $j$ , where the weights  $\{\lambda_j\}$  are defined by

$$\lambda_j = 1 \bigg/ \prod_{\substack{k=0 \\ k \neq j}}^{2m} \sin \frac{1}{2}(\theta_j - \theta_k). \quad (1.6.15)$$

*Proof.* The theorem follows from the above discussion. □

For equispaced points in  $[0, 2\pi)$ , the expression for the node polynomial  $\ell$  simplifies to

$$\ell(\theta) = \frac{1}{2^{2m}} \sin \left( \frac{2m+1}{2} \theta \right), \quad (1.6.16)$$

and the weights  $\{\lambda_j\}$  are given by the formula [49]:

$$\lambda_j = (-1)^j \frac{2^{2m}}{2m+1}. \quad j = 0, 1, \dots, 2m. \quad (1.6.17)$$

The last two equations allow us to simplify both the type I and type II formulas in the special case of equispaced points. The type I trigonometric barycentric formula in (1.6.9) simplifies to

$$t(\theta) = \frac{\sin\left(\frac{2m+1}{2}\theta\right)}{2m+1} \sum_{j=0}^{2m} \frac{(-1)^j}{\sin\frac{1}{2}(\theta-\theta_j)} f_j, \quad (1.6.18)$$

and since the weights  $\{\lambda_j\}$  appear both in the numerator and the denominator of (1.6.14), the constant factor of  $\frac{2^{2m}}{2m+1}$  cancels out and analogous to Theorem 5.2 in [85], we have the following theorem.

**Theorem 1.4** (*Trigonometric barycentric interpolation in equispaced points*). *The trigonometric polynomial interpolant through data  $\{f_j\}$  at  $2m+1$  equispaced points in  $[0, 2\pi)$  is*

$$t(\theta) = \sum_{j=0}^{2m} \frac{(-1)^j f_j}{\sin\frac{1}{2}(\theta-\theta_j)} \bigg/ \sum_{j=0}^{2m} \frac{(-1)^j}{\sin\frac{1}{2}(\theta-\theta_j)}, \quad (1.6.19)$$

with the special case  $t(\theta) = f_j$  if  $\theta = \theta_j$ .

*Proof.* See [49]. □

The numerical stability of trigonometric barycentric formulas is a fascinating topic but we will not discuss this in the present thesis. For that, we refer the reader to a very detailed analysis in Austin's PhD thesis [6] or the paper by Austin and Xu [8].

A Matlab code for trigonometric barycentric interpolation is freely available as part of Chebfun and can be downloaded from <https://www.github.com/chebfun/chebfun/>. The code was originally written by the author of this thesis and is mainly contained in the file `trigBary.m`.

### 1.6.1 A property of the barycentric weights

A property of the barycentric weights  $\{\lambda_j\}$  that appear in (1.6.9) and (1.6.14) is that they add up to zero. The result can be proved in various ways. We now prove it using contour integrals. In order to do this, we first prove another interesting result regarding trigonometric polynomials. Take a  $2\pi$ -periodic trigonometric polynomial  $t$  of degree  $\geq 1$  on the vertical strip  $S$ , where  $S$  is defined by the equation

$$S = \{\theta : 0 \leq \operatorname{Re}(\theta) < 2\pi\}. \quad (1.6.20)$$

Now consider  $\Gamma$ , a simple closed contour which encloses all the roots of  $t$  in  $S$  once in the positive direction. Then

$$\int_{\Gamma} \frac{1}{t(\theta)} d\theta = 0. \quad (1.6.21)$$

The proof is easy. Since all the roots of  $t$  are already enclosed in  $\Gamma$ , we can deform  $\Gamma$  into a rectangle of arbitrary large height with the sides of the rectangle running along the vertical

boundaries of  $S$ , without changing the value of the integral. Since  $t$  is periodic, the contributions from the left and the right legs of the deformed contour cancel each other out. The assumption about the degree of  $t$  now ensures that the integral must vanish as we take the top horizontal leg of the rectangular contour to  $+i\infty$  and the bottom horizontal leg to  $-i\infty$ .

This gives a contour integral proof of the fact that the sum of the barycentric weights of a trigonometric Lagrange interpolant is zero. This is easy to show. Suppose that  $\{\theta_k\}_{k=0}^{2m}$  are the  $2m + 1$  nodes used for interpolation and let us define the trigonometric node polynomial  $\ell$  as before. Now take a simple closed contour  $\Gamma$  which encloses all the interpolation nodes once in the positive direction. Then by the residue theorem and the above result

$$\int_{\Gamma} \frac{1}{\ell(\theta)} d\theta = (2\pi i) \sum_{k=0}^{2m} \frac{1}{\ell'(\theta_k)} = 0. \quad (1.6.22)$$

Since the  $k^{\text{th}}$  weight appearing in the type I trigonometric barycentric interpolation formula (1.6.9) is given by the expression

$$\lambda_k = \frac{1}{2\ell'(\theta_k)}, \quad (1.6.23)$$

we have thus shown that

$$\sum_{k=0}^{2m} \lambda_k = 0, \quad m \geq 1. \quad (1.6.24)$$

## 1.7 Trigonometric and Laurent series

A trigonometric polynomial by definition is a finite trigonometric series<sup>4</sup>. One can use trigonometric polynomials and trigonometric series to approximate functions from other spaces. If a periodic function is *sufficiently smooth*, we can find trigonometric polynomials of higher and higher degrees which will approximate  $f$  to any given degree of accuracy in a reasonably prescribed norm. This is the general idea behind expanding a function in its trigonometric series. The smoother the function, the easier it is to approximate. What we mean by that in the present context is that the trigonometric coefficients of a smooth function quickly become very small. For example, the  $2\pi$ -periodic entire function  $f(\theta) = e^{\sin(\theta)}$  has a uniformly convergent trigonometric series and to numerically approximate  $f$  up to machine precision on the interval  $[0, 2\pi]$ , we only need a degree-15 trigonometric polynomial<sup>5</sup>. However, if  $f$  is say, infinitely differentiable but not analytic, or if it is twice differentiable, or if it has a finite number of jump discontinuities, etc., then the convergence or divergence of the corresponding trigonometric series is a subtle question of analysis and mathematicians have spent their entire careers studying such cases carefully. As a result, we now know exactly what degree of smoothness of a

<sup>4</sup>It is more common to say Fourier series instead of trigonometric series, but for various reasons of consistency, here and elsewhere in the thesis, we have made the choice of using the word *trigonometric* wherever possible.

<sup>5</sup>Try the command `f = chebfun(@(t) exp(sin(t)), [0, 2*pi], 'trig')` in Chebfun.

function corresponds to what kind of convergence of its trigonometric series. For a fascinating account of the history of convergence of a trigonometric series and its importance in the historical evolution of mathematical analysis, we refer the reader to the book by Bressoud [21]<sup>6</sup>. For a textbook style treatment of the subject, see, for example, the books by Katznelson [60] and Körner [61]. Finally, it would be difficult not to find a result related to trigonometric series in the comprehensive two volume work by Zygmund [103]. The focus of this thesis, however, is on numerical algorithms for computing with periodic functions and we shall refrain from diving into such details of analysis or trying to prove the sharpest possible results for convergence. We will content ourselves by making a particular assumption about the smoothness of the function  $f$  that covers most practical and numerical applications. We shall assume that  $f$  is *Lipschitz continuous* on the periodic interval  $[0, 2\pi]$  with  $f(0) = f(2\pi)$ . Recall that this means that there is a constant  $C$  such that

$$|f(\theta_1) - f(\theta_2)| \leq C|\theta_1 - \theta_2|, \quad \theta_1, \theta_2 \in [0, 2\pi].$$

Topics related to continuity and convergence are discussed at length in analysis textbooks such as [79], [80] or [101]. For an excellent and beautiful textbook on convergence, we refer the reader to the book by Ferrar<sup>7</sup> [34].

Following [85], we present the following theorem about trigonometric series of Lipschitz continuous periodic functions.

**Theorem 1.5** (*Trigonometric Series*). *If  $f$  is Lipschitz continuous on  $[0, 2\pi]$  with  $f(0) = f(2\pi)$ , it has a uniformly and absolutely convergent trigonometric series,*

$$f(\theta) = \sum_{k=-\infty}^{\infty} c_k e^{ik\theta}, \tag{1.7.1}$$

and the coefficients  $c_k$  are given by the formula

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} f(\theta) e^{-ik\theta} d\theta. \tag{1.7.2}$$

*Proof.* See any standard text on Fourier analysis such as [60] or the comprehensive work by Zygmund [103]. □

Approximation of functions is usually considered on an interval, or on the unit circle, or on a periodic interval. If our functions live on an interval, Chebyshev polynomial interpolants through Chebyshev points are well suited for their numerical approximations. If our functions

---

<sup>6</sup>The first chapter of this book has the title *Crisis in Mathematics: Fourier's Series*.

<sup>7</sup>William Leonard Ferrar was a fellow and tutor of Hertford College, Oxford

live on the unit circle, Laurent polynomial interpolants through roots of unity are usually the best choice for approximations. And finally, if our functions live on a periodic interval, an excellent method to approximate them is usually by trigonometric polynomial interpolants through equispaced points. The Chebyshev setting for approximation of functions on an interval has been extensively dealt with in [85]. The aim of this thesis is to explore the trigonometric setting, which is equivalent to the Laurent setting, as we will shortly see. To establish this equivalence, we now move our discussion to the complex plane. In the trigonometric case, we deal with a function  $f$  of the complex variable  $\theta$  defined on an open neighbourhood of the periodic interval  $[0, 2\pi]$ :

$$\text{Trigonometric: } \theta \in [0, 2\pi], \quad f(\theta) \approx \sum_{k=-n}^n a_k e^{ik\theta}. \quad (1.7.3)$$

In the Laurent case, we deal with the variable  $z$  which ranges over an open neighbourhood of the unit circle in the complex  $z$ -plane. (For an introduction to Laurent series and related topics of complex analysis, see [2], [52], [64] or [81].) To make the connection between the trigonometric and the Laurent case, we define a function  $F(z)$  on the unit circle by the equation

$$F(z) = f(\theta), \quad (1.7.4)$$

where

$$z = e^{i\theta}. \quad (1.7.5)$$

Note that the above equation implies a unique value of  $z$  for each value of  $\theta$ . The series expansion of  $F$  now contains powers of both  $z$  and  $z^{-1}$  and it is called a *Laurent polynomial* [64]:

$$\text{Laurent: } |z| = 1, \quad F(z) \approx \sum_{k=-n}^n a_k z^k. \quad (1.7.6)$$

Let's see this beautiful interplay between the  $\theta$ -plane and the  $z$ -plane in action. As an example, we let  $f(\theta) = \sin(3\theta)$  and plot its phase portrait in Figure 1.4. (For a detailed exposition of phase portraits, see [99].) We can see that there are 6 real zeros in the interval  $[0, 2\pi)$ .

We then plot the phase portrait of  $F(z)$  in the  $z$ -plane where  $F(z) = f(\theta)$  under the mapping  $z = e^{i\theta}$  in Figure 1.5. We can also see in Figure 1.5 that the 6 zeros of  $f$  have now been transformed to the unit circle.

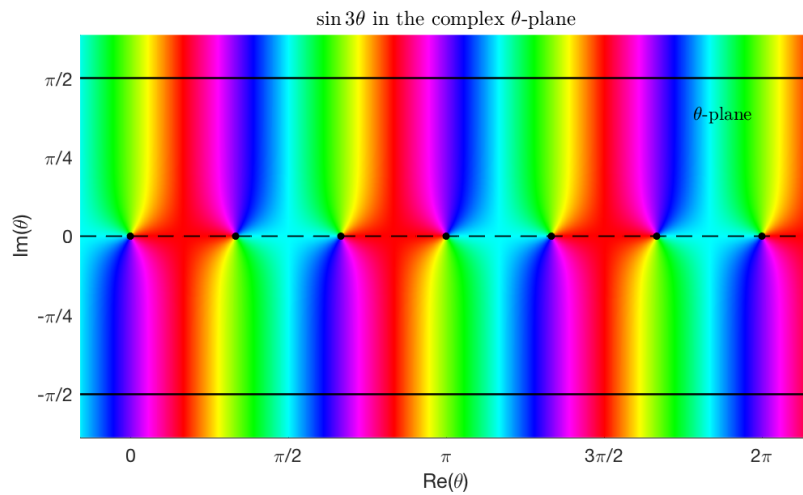


Figure 1.4: Phase portrait of  $f(\theta) = \sin(3\theta)$  in the  $\theta$ -plane. The zeros of  $f$  are shown as black dots, and we can see that they all lie on the real line (dashed).

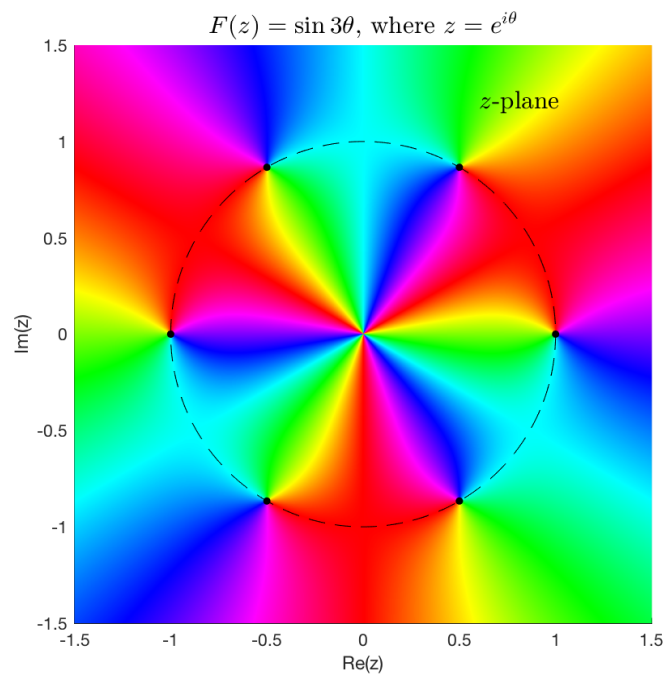


Figure 1.5: Phase portrait of  $F(z) = f(\theta)$  in the  $z$ -plane, where  $z = e^{i\theta}$ . The (dashed) unit circle is also shown. We can see that all the zeros of  $F$  (black dots) lie on the unit circle.

We now consider another example. Let

$$F(z) = \frac{2}{z - .6i} + \frac{1}{z - 2}. \quad (1.7.7)$$

This function has two simple poles; one at  $z = 0.6i$  and another at  $z = 2$ . The function has a simple zero at  $z = 4/3 + i/5$ . The phase portrait is shown in Figure 1.6. We can expand  $F$  in a Laurent series which converges in the annular region bounded by the two (solid) black circles as shown in Figure 1.6 [81]. This series is given by

$$F(z) = \frac{2}{z - .6i} + \frac{1}{z - 2} \quad (1.7.8)$$

$$= \frac{2}{z(1 - .6i/z)} + \frac{-1}{2(1 - z/2)} \quad (1.7.9)$$

$$= \frac{2}{z} \sum_{k=0}^{\infty} \left(\frac{.6i}{z}\right)^k + \frac{-1}{2} \sum_{k=0}^{\infty} \left(\frac{z}{2}\right)^k \quad (1.7.10)$$

$$= \frac{2}{.6i} \sum_{k=1}^{\infty} \left(\frac{.6i}{z}\right)^k - \frac{1}{2} \sum_{k=0}^{\infty} \left(\frac{z}{2}\right)^k, \quad (1.7.11)$$

and is valid in the annular region determined by the inequalities

$$0.6 < |z| < 2. \quad (1.7.12)$$

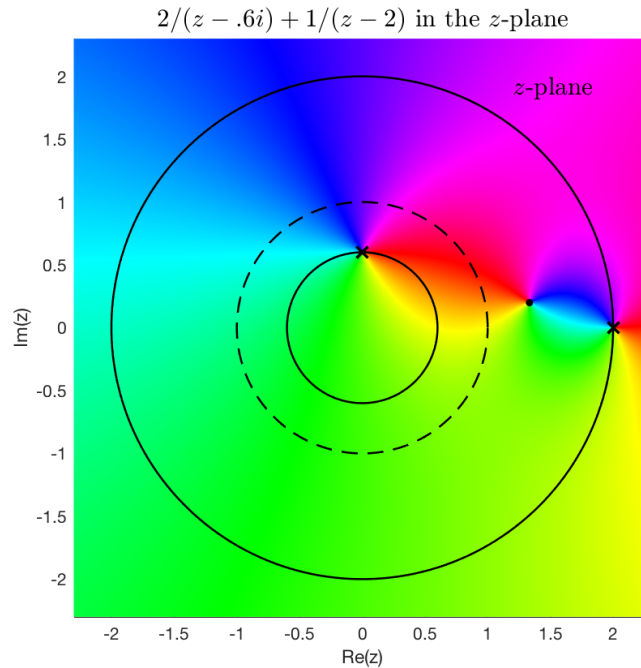


Figure 1.6: Phase portrait of the function  $F(z) = 2/(z - .6i) + 1/(z - 2)$  in the  $z$ -plane. The poles of  $F$  are shown by black crosses while the zero is shown by a black dot. The unit circle is represented by black dashes while the solid circles pass through the two poles of  $F$ .  $F$  can be expanded in a Laurent series which converges in the annular region between the two solid circles.

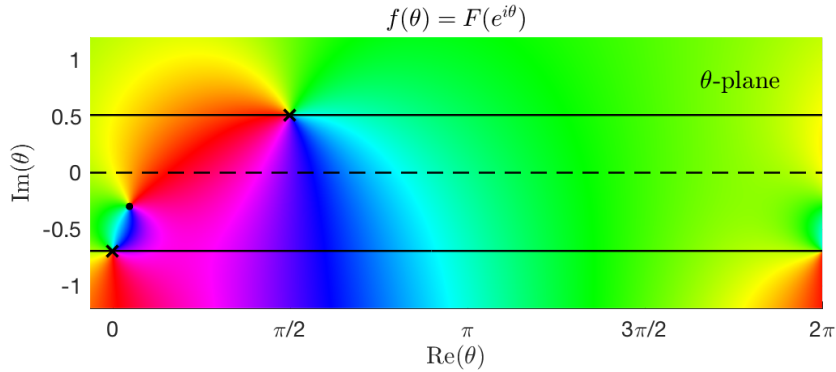


Figure 1.7: Phase portrait of  $f(\theta) = F(e^{i\theta})$  in the  $\theta$ -plane. The poles of  $f$  are shown by black crosses, while the zero is shown by a black dot. The real axis is plotted as a dashed line, while the solid lines pass through the two poles of  $f$ . Note that  $f$  is analytic in the strip between the two solid lines.

Let us now transplant everything to the  $\theta$ -plane. Corresponding to the Laurent series of  $F$ , we have a trigonometric series of  $f$  [31]:

$$f(\theta) = \frac{2}{.6i} \sum_{k=1}^{\infty} (.6i)^k e^{-ik\theta} - \frac{1}{2} \sum_{k=0}^{\infty} \left(\frac{1}{2^k}\right) e^{ik\theta}. \quad (1.7.13)$$

The annular region of analyticity of  $F(z)$  is transformed to a strip of analyticity of  $f(\theta)$ , as shown in Figure 1.7. This strip of analyticity is the region between the two solid lines and the above trigonometric series of  $f$  is only valid in this region.

## 1.8 Contour integrals of periodic analytic functions

We now consider contour integrals of periodic analytic functions in the  $\theta$ -plane. For an excellent introduction to contour integrals, see [74]. Again, the mapping

$$z = e^{i\theta}$$

will play a crucial role in transforming relatively unfamiliar contours in the  $\theta$ -plane to their well known images in the  $z$ -plane. Under this mapping, horizontal lines in the  $\theta$ -plane are mapped to circles in the  $z$ -plane. In particular, the real axis in the  $\theta$ -plane is mapped to the unit circle in the  $z$ -plane. The same mapping maps vertical lines in the  $\theta$ -plane to rays in the  $z$ -plane. In particular, the imaginary axis in the  $\theta$ -plane is mapped to the positive real axis in the  $z$ -plane.

An important example, which we shall refer to later, is the mapping from the  $\theta$ -plane of a rectangular contour of the kind shown in the left half of Figure 1.8 to a contour in the  $z$ -plane as shown in the right half of the same Figure.

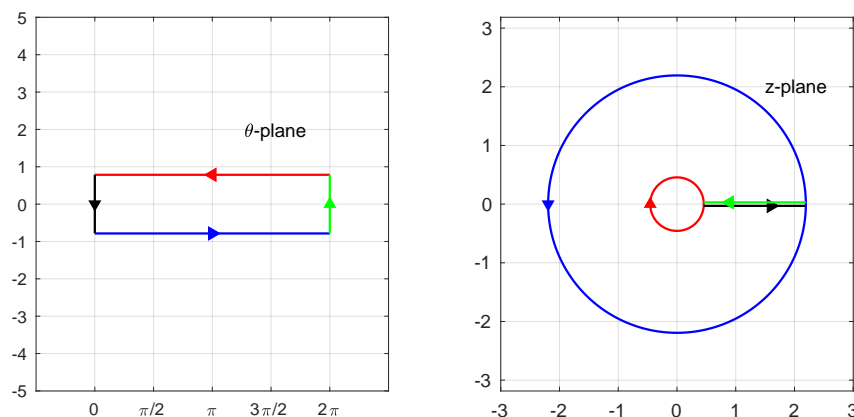


Figure 1.8: The mapping  $z = e^{i\theta}$  maps the rectangular contour on the left to the contour shown on the right. Each leg of the rectangular contour is coloured differently and the same colours are used for their mapped images. The green and the black parts of the contour on the right pass through the same points, but in opposite directions. These two lines are shown slightly apart from each other for visual aid.

Let us now consider a domain  $\Omega$  in the  $\theta$ -plane and a function  $f$  analytic in  $\Omega$ . We denote a simple closed rectangular contour of the kind shown in Figure 1.8 by  $\Gamma$  and assume that  $\Gamma \subseteq \Omega$  runs counter clockwise in the positive direction. We now consider the contour integral

$$\int_{\Gamma} f(\theta) d\theta. \quad (1.8.1)$$

The contour  $\Gamma$  is made up of four line segments and we can write

$$\int_{\Gamma} f(\theta) d\theta = \int_{\Gamma_{top}} f(\theta) d\theta + \int_{\Gamma_{left}} f(\theta) d\theta + \int_{\Gamma_{bottom}} f(\theta) d\theta + \int_{\Gamma_{right}} f(\theta) d\theta. \quad (1.8.2)$$

If we now assume further that  $f$  is periodic and the two vertical legs of the contour, namely  $\Gamma_{left}$  and  $\Gamma_{right}$  are one period apart, then the contribution of  $\Gamma_{left}$  cancels the contribution of  $\Gamma_{right}$ , that is,

$$\int_{\Gamma_{left}} f(\theta) d\theta = - \int_{\Gamma_{right}} f(\theta) d\theta. \quad (1.8.3)$$

The contributing top and bottom legs of the contour are shown in the left half of Figure 1.9. By mapping these two horizontal legs of  $\Gamma$  to the  $z$ -plane under the mapping  $z = e^{i\theta}$ , we obtain a contour which consists of two circles centred at the origin. This can be seen in the right half of Figure 1.9. We therefore get

$$\int_{\Gamma} f(\theta) d\theta = \int_{\Gamma_{top}} f(\theta) d\theta + \int_{\Gamma_{bottom}} f(\theta) d\theta. \quad (1.8.4)$$

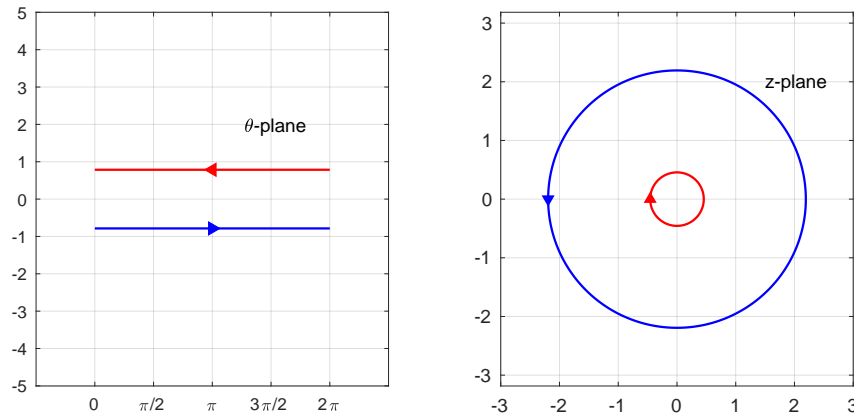


Figure 1.9: The contour integral of a periodic analytic function on a rectangular contour whose horizontal sides are one period apart has the same value as its contour integral over just the top and bottom line segments of the contour, as shown above in the left window. The contributions from the (not shown) left and right vertical legs of the contour cancel each other out due to periodicity. The mapping  $z = e^{i\theta}$  maps the contour on the left to the familiar concentric circles in the  $z$ -plane shown on the right.

This simple observation has far reaching consequences. For example, based on this observation, one can explain the geometric convergence of the trapezoidal rule for periodic analytic functions and the unification of this geometric convergence with the Euler–Maclaurin formula. We will not discuss these connections here and refer the interested reader to [54] and [91].

## 1.9 Cauchy’s integral formula for periodic analytic functions

Regardless of whether we are operating in the  $z$ -plane or the  $\theta$ -plane, Cauchy’s integral formula holds for any analytic function of a complex variable. For a function  $f$  analytic in a simply connected domain in the  $\theta$ -plane which contains a simple closed contour  $\Gamma$  and a point  $a$ , we have

$$f(a) = \frac{1}{2\pi i} \int_{\Gamma} \frac{f(\theta)}{\theta - a} d\theta, \quad (1.9.1)$$

where  $\Gamma$  encloses the point  $a$  once in the positive direction [2].

However, for  $f$  analytic and periodic, we now develop a periodic version of the Cauchy’s integral formula. What we mean by that is an integral formula which has the same qualitative properties as those possessed by Cauchy’s integral formula but in which the integrand is periodic. In short, we want to replace the non-periodic  $\frac{1}{\theta - a}$  factor appearing in Cauchy’s integral formula with a factor which is periodic but does not change the essential characteristics of Cauchy’s integral formula.

Let us first consider the complex  $z$ -plane and suppose that  $F$  is a function analytic in an annular region  $A$  defined by the circles  $C_1$  and  $C_2$ , centred at the origin and of radius  $R_1$  and  $R_2$ , respectively. That is,  $A$  is the set of points  $\{z : R_1 < |z| < R_2\}$ . We further assume that  $0 \leq R_1 < 1 < R_2 \leq \infty$  so that  $F$  is analytic on the unit circle and has a convergent Laurent series there,

$$F(z) = \sum_{k=-\infty}^{\infty} a_k z^k. \quad (1.9.2)$$

Cauchy's integral formula applies and for any  $a \in A$ , we can write

$$F(a) = \frac{1}{2\pi i} \int_{\Gamma} \frac{F(z)}{z-a} dz, \quad (1.9.3)$$

where  $\Gamma \subseteq A$  is a simple closed contour which encloses the point  $a$  once in the positive direction. The key ingredient in that formula is the factor

$$\frac{1}{z-a}. \quad (1.9.4)$$

We shall henceforth refer to this factor as the Cauchy kernel. We observe that the Cauchy kernel has the property that  $1/2\pi i$  times its integral is 1 over any simple closed contour  $\Gamma$  which encloses the point  $a$  once in the positive direction, that is,

$$\frac{1}{2\pi i} \int_{\Gamma} \frac{1}{z-a} = 1. \quad (1.9.5)$$

The Cauchy kernel is very much like the Dirac delta function. Its integral over  $\Gamma$  is 1 or 0 depending upon whether  $\Gamma$  encloses the point  $a$  or not. Also, Cauchy's integral formula can be thought of as the complex analyst's analogue of the real analyst's Dirac delta function with its famous sifting property. We have,

$$\text{Real: } \int_{-\infty}^{\infty} f(x)\delta(x-a) dx = f(a), \quad \text{Complex: } \frac{1}{2\pi i} \int_{\Gamma} \frac{f(z)}{z-a} dz = f(a). \quad (1.9.6)$$

The delta function samples  $f$  at a point on the real line [58]. Similarly, the Cauchy kernel samples the function  $f$  at the location of its pole in the complex plane. The analogy is not just a curious observation; the whole theory of hyperfunctions is based on it [41].

Let us now move to the  $\theta$ -plane. We assume that our functions are  $2\pi$ -periodic and analytic on a domain which contains the interval  $[0, 2\pi]$ . We wish to construct a function which is the periodic analogue of the Cauchy kernel. Let us call this function the periodic Cauchy kernel. We now list the properties which we think this kernel should have.

1. It should be periodic. The period can be arbitrary, but since we are working with  $2\pi$ -periodic functions, we require the periodic Cauchy kernel to have a period of  $2\pi$ .
2. Within any vertical strip of width  $2\pi$ , the periodic Cauchy kernel should be analytic, except at one point, where it should have a simple pole.

3. It should integrate to  $2\pi i$  over any simple closed contour which encloses its pole once in the positive direction, and integrate to 0 otherwise.
4. It should also have the sifting property of *sampling* a given function at the point where it has its pole.

Dictated by the above requirements, we select the function

$$\frac{1}{2} \cot \frac{\theta}{2} \quad (1.9.7)$$

as our periodic Cauchy kernel. (See [54] for a comparison, also compare Table 13.1 of [91].) We plot this function and its phase portrait in Figure 1.10. The Laurent expansion of the function  $\frac{1}{2} \cot \frac{\theta}{2}$  at  $\theta = 0$  is given by the equation

$$\frac{1}{2} \cot \frac{\theta}{2} = \frac{1}{\theta} - \frac{\theta}{12} - \frac{\theta^3}{720} + O(\theta^5), \quad (1.9.8)$$

which shows that the residue at  $\theta = 0$  is 1, that is,

$$\text{Res} \left( \frac{1}{2} \cot \frac{\theta}{2}, 0 \right) = 1. \quad (1.9.9)$$

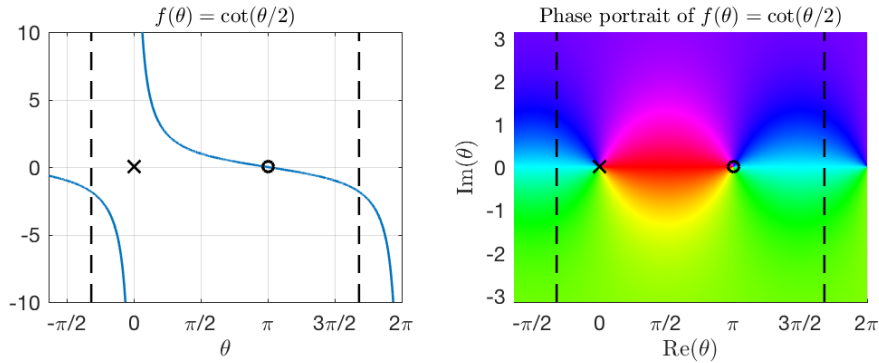


Figure 1.10: (Left) Plot of the function  $\cot(\theta/2)$ . (Right) Phase portrait of  $\cot(\theta/2)$ . The function is  $2\pi$ -periodic and the distance between the vertical black lines is exactly one period. We can also see that each period has one pole (black cross) and one zero (black circle).

The relationship between the periodic Cauchy kernel and the  $2\pi$ -periodic delta function, also known as the Dirac comb [41, Ch. 2, p. 68], is further highlighted by the beautiful equation [1, Ch. 3, p. 172],

$$\frac{1}{2} \cot \left( \frac{\theta}{2} \right) = \frac{1}{\theta} + \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \left( \frac{1}{\theta - 2k\pi} + \frac{1}{2k\pi} \right) = -2\pi i \sum_{k=-\infty}^{\infty} \delta(x - 2k\pi), \quad (1.9.10)$$

where the last equality is in the sense of hyperfunctions [41].

Let us assume now that  $f$  is a  $2\pi$ -periodic function of the complex variable  $\theta$ . Also assume that  $f$  is analytic in the rectangular strip  $S = \{\theta : 0 \leq \operatorname{Re}(\theta) < 2\pi, -b < \operatorname{Im}(\theta) < b\}$ , where  $b > 0$ . Let  $\Gamma$  be a simple closed contour in  $S$  enclosing the point  $a$  once in the positive direction. We then have

$$f(a) = \frac{1}{4\pi i} \int_{\Gamma} f(\theta) \cot \frac{1}{2}(\theta - a) d\theta. \quad (1.9.11)$$

This is the periodic version of Cauchy's integral formula that we were seeking.

## 1.10 Hermite integral formula for trigonometric interpolation

We now consider the trigonometric polynomial interpolation problem in the complex plane and derive the trigonometric version of the Hermite integral formula. A detailed presentation of the Hermite integral formula for algebraic polynomial interpolation can be found in [85, Ch. 11], which our present derivation closely follows. The result can hardly be called new and it is very much possible that one of the greats might have written down a periodic version of the Hermite integral formula. However, despite our best efforts, we<sup>8</sup> have not been able to find a periodic analogue of the Hermite integral formula in the literature we examined.

Without loss of generality, we assume that we are given a set of  $2m + 1$  distinct points  $\{\theta_j\}$  in the complex  $\theta$ -plane whose real parts satisfy the condition

$$0 \leq \operatorname{Re}(\theta_j) < 2\pi, \quad j = 0, 1, \dots, 2m. \quad (1.10.1)$$

Recall that the *trigonometric node polynomial*  $\ell$  for these points is defined by

$$\ell(\theta) = \prod_{j=0}^{2m} \sin \frac{1}{2}(\theta - \theta_j). \quad (1.10.2)$$

We can write the trigonometric Lagrange polynomial  $\ell_j$  in terms of  $\ell$  as

$$\ell_j(\theta) = \frac{\ell(\theta)}{2\ell'(\theta_j) \sin \frac{1}{2}(\theta - \theta_j)}. \quad (1.10.3)$$

Before we continue with the derivation of the Hermite integral formula for trigonometric polynomial interpolation, we prove that  $\ell_j$  is  $2\pi$ -periodic while  $\ell$  is not  $2\pi$ -periodic.

---

<sup>8</sup>We here means the author of this thesis and Anthony P. Austin who independently derived an equivalent trigonometric Hermite integral formula. Austin's version uses a complex exponential basis and employs a different definition for the node polynomial  $\ell(\theta)$ , resulting in a formula that is different in appearance from the formula we develop in this thesis (private communication).

**Theorem 1.6.** *Given an even number  $N \geq 2$ , and complex numbers  $\{\alpha_k\}$ , the product*

$$\prod_{k=1}^N \sin \frac{1}{2}(\theta - \alpha_k), \quad (1.10.4)$$

*is a  $2\pi$ -periodic function of the variable  $\theta$ .*

*Proof.* For  $N = 2$ , the product has the form

$$\sin \frac{1}{2}(\theta - \alpha_1) \sin \frac{1}{2}(\theta - \alpha_2). \quad (1.10.5)$$

Using the formula

$$\sin(a) \sin(b) = \frac{1}{2} [\cos(a - b) - \cos(a + b)], \quad (1.10.6)$$

we get

$$\sin \frac{1}{2}(\theta - \alpha_1) \sin \frac{1}{2}(\theta - \alpha_2) = \frac{1}{2} \left[ \cos \frac{1}{2}(\alpha_2 - \alpha_1) - \cos \left( \theta - \frac{\alpha_1 + \alpha_2}{2} \right) \right]. \quad (1.10.7)$$

On the right-hand side, the first term inside the square brackets is a constant while the second is a  $2\pi$ -periodic function of  $\theta$ . Hence the theorem is true for  $N = 2$ . For larger even numbers, we can pair up all the terms in the product and the theorem follows, since the product of periodic functions is also periodic.  $\square$

**Corollary 1.** *The trigonometric Lagrange polynomial  $\ell_j$  is  $2\pi$ -periodic.*

*Proof.* Each  $\ell_j$  is a product of  $2m$  terms of the form  $\sin \frac{1}{2}(\theta - \theta_k)$ . The result now follows from the above theorem.  $\square$

**Corollary 2.** *The trigonometric node polynomial  $\ell$  is  $4\pi$ -periodic but not  $2\pi$ -periodic.*

*Proof.*  $\ell(\theta)$  has the form

$$\ell(\theta) = \ell_j(\theta) \left( A_j \sin \frac{1}{2}(\theta - \theta_j) \right), \quad (1.10.8)$$

for any  $j = 0, 1, \dots, 2m$ , and some constants  $A_j$ . The function  $\ell_j$  is  $2\pi$ -periodic as shown above but the factor  $\sin \frac{1}{2}(\theta - \theta_j)$  is  $4\pi$ -periodic but not  $2\pi$ -periodic. Therefore,  $\ell$  is also  $4\pi$ -periodic but not  $2\pi$ -periodic.  $\square$

We return to the derivation of the trigonometric Hermite integral formula.

The trigonometric polynomial interpolant  $t \in \mathcal{T}_m$  to an arbitrary function  $f$  defined on the grid can be expressed as a linear combination of trigonometric Lagrange polynomials:

$$t(\theta) = \sum_{j=0}^{2m} f(\theta_j) \ell_j(\theta_j). \quad (1.10.9)$$

Let us try to write  $\ell_j(\theta)$  as a contour integral. In order to do this, we first need to carefully define a domain. Since the functions  $\ell_j$  and  $\ell$  are periodic and have infinitely many zeros in the complex  $\theta$ -plane, we therefore consider a simply connected region  $\Omega$  which contains the distinct points of interpolation  $\theta_0, \theta_1, \dots, \theta_{2m}$  and does not contain any other zero of  $\ell(\theta)$ . All our contours will lie in the domain  $\Omega$ .

Consider the simple closed contour  $\Gamma_j \subseteq \Omega$  which encloses the point  $\theta_j$  once in the positive direction and encloses no other point of interpolation, nor the point  $\theta$ . We can write

$$\frac{\ell(\theta)}{2\ell'(\theta_j) \sin \frac{1}{2}(\theta - \theta_j)} = \frac{1}{2\pi i} \int_{\Gamma_j} \frac{\ell(\theta)}{2\ell(\omega) \sin \frac{1}{2}(\theta - \omega)} d\omega. \quad (1.10.10)$$

The above equation follows from residue calculus [81]. First notice that the numerator  $\ell(\theta)$  inside the integral is constant with respect to  $\omega$ . Next, the only pole that  $\Gamma_j$  encloses is located at  $\theta_j$  and the corresponding residue is  $1/(2\ell'(\theta_j) \sin \frac{1}{2}(\theta - \theta_j))$ . Using the above equation together with (1.10.3), we arrive at a contour integral definition of the trigonometric Lagrange polynomial  $l_j$ , that is,

$$l_j(\theta) = \frac{1}{2\pi i} \int_{\Gamma_j} \frac{\ell(\theta)}{2\ell(\omega) \sin \frac{1}{2}(\theta - \omega)} d\omega. \quad (1.10.11)$$

If we now define another simple closed contour  $\Gamma' \subseteq \Omega$  which encloses all the points  $\{\theta_j\}$  once in the positive direction but not the point  $\theta$ , then the trigonometric polynomial interpolant  $t$  of a  $2\pi$ -periodic analytic function  $f$  can be expressed as the contour integral

$$t(\theta) = \frac{1}{2\pi i} \int_{\Gamma'} \frac{\ell(\theta)f(\omega)}{2\ell(\omega) \sin \frac{1}{2}(\theta - \omega)} d\omega. \quad (1.10.12)$$

Observe that by Theorem 1.6, the function  $f(\omega)/(2\ell(\omega) \sin \frac{1}{2}(\theta - \omega))$  is a  $2\pi$ -periodic function of  $\omega$ .

Finally, we define a simple closed contour  $\Gamma \subseteq \Omega$  which encloses all the points  $\{\theta_j\}$  and also the point  $\theta$  once in the positive direction. Not only does the contour  $\Gamma$  pick up all the residues due to  $\theta_j$  giving us  $t$  as above, it also picks up the residue at  $\theta$ , which is  $-f(\theta)$ . This gives us

$$t(\theta) - f(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\ell(\theta)f(\omega)}{2\ell(\omega) \sin \frac{1}{2}(\theta - \omega)} d\omega. \quad (1.10.13)$$

Multiplying both sides of the above equation by  $-1$  gives us the *trigonometric Hermite integral formula*:

$$f(\theta) - t(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\ell(\theta)f(\omega)}{2\ell(\omega) \sin \frac{1}{2}(\omega - \theta)} d\omega. \quad (1.10.14)$$

As mentioned in the beginning of this section, we have not been able to find this formula in the literature. To summarize, following Theorem 11.1 in [85], we present our result in the form of a theorem.

**Theorem 1.7** (*Trigonometric Hermite integral formula*). Let  $\theta_0, \theta_1, \dots, \theta_{2m}$  be distinct points in the complex plane satisfying

$$0 \leq \operatorname{Re}(\theta_j) < 2\pi, \quad j = 0, 1, \dots, 2m.$$

Let  $f$  be an analytic function in a simply connected region  $\Omega$ , which contains  $\{\theta_j\}$  and no other zeros of the function  $\ell(\theta) = \prod_{j=0}^{2m} \sin \frac{1}{2}(\theta - \theta_j)$ . Let  $\Gamma$  be a simple closed contour in  $\Omega$  enclosing the points  $\{\theta_j\}$  once in the positive direction. The trigonometric polynomial interpolant  $t \in \mathcal{T}_m$  to  $f$  at  $\{\theta_j\}$  is

$$t(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{f(\omega)(\ell(\omega) - \ell(\theta))}{2\ell(\omega) \sin \frac{1}{2}(\omega - \theta)} d\omega, \quad (1.10.15)$$

and if  $\Gamma$  encloses  $\theta$ , the error in the interpolant is

$$f(\theta) - t(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\ell(\theta)f(\omega)}{2\ell(\omega) \sin \frac{1}{2}(\omega - \theta)} d\omega. \quad (1.10.16)$$

*Proof.* The formula in (1.10.16) is the same as (1.10.14). To prove (1.10.15), observe that if  $\Gamma$  encloses  $\theta$ , then using residue calculus,  $f(\theta)$  can be written as

$$f(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{f(\omega)}{2 \sin \frac{1}{2}(\omega - \theta)} d\omega, \quad (1.10.17)$$

which is the same as

$$f(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\ell(\omega)f(\omega)}{2\ell(\omega) \sin \frac{1}{2}(\omega - \theta)} d\omega. \quad (1.10.18)$$

Now subtracting (1.10.14) from the above equation gives (1.10.15).

If  $\Gamma$  does not enclose  $\theta$ , then the integrand in (1.10.15) has no pole at  $\omega = \theta$  and the formula (1.10.15) still holds. This proves the theorem.  $\square$

Recall from Equation (1.6.16) that for an equispaced grid, the node polynomial  $\ell$  can be written as,

$$\ell(\theta) = \frac{1}{2^{2m}} \sin \left( \frac{2m+1}{2} \theta \right). \quad (1.10.19)$$

This gives us the following corollary.

**Corollary 3.** For an equispaced grid of  $2m+1$  points in  $[0, 2\pi)$ , the interpolant  $t$  is given by

$$t(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{f(\omega) \left( \sin \left( \frac{2m+1}{2} \omega \right) - \sin \left( \frac{2m+1}{2} \theta \right) \right)}{2 \sin \left( \frac{2m+1}{2} \omega \right) \sin \frac{1}{2}(\omega - \theta)} d\omega, \quad (1.10.20)$$

and if  $\Gamma$  encloses  $\theta$ , the error in the interpolant is

$$f(\theta) - t(\theta) = \frac{1}{2\pi i} \int_{\Gamma} \frac{\sin \left( \frac{2m+1}{2} \theta \right) f(\omega)}{2 \sin \left( \frac{2m+1}{2} \omega \right) \sin \frac{1}{2}(\omega - \theta)} d\omega. \quad (1.10.21)$$

To see how the trigonometric Hermite integral formula explains the error in trigonometric polynomial interpolation, suppose that  $f$  is analytic in a region containing  $[-1, 1]$  and the point of evaluation  $\theta$  as well as the points of interpolation  $\{\theta_j\}$  are all contained in the interval  $[-1, 1]$ . Let us now fix  $f$  and  $\theta$  in formula (1.10.16) and consider a fixed contour  $\Gamma$ . The quantities  $f(\omega)$  and  $\sin \frac{1}{2}(\omega - \theta)$  in (1.10.16) do not depend on  $m$  and the only term that depends on  $m$  is the ratio  $\ell(\theta)/\ell(\omega)$ , which we can write explicitly as,

$$\frac{\ell(\theta)}{\ell(\omega)} = \frac{\prod_{k=0}^{2m} \sin \frac{1}{2}(\theta - \theta_k)}{\prod_{k=0}^{2m} \sin \frac{1}{2}(\omega - \theta_k)}. \quad (1.10.22)$$

We also know that the sine function grows exponentially in the complex plane. As a result, if  $\omega$  is sufficiently far from  $\{\theta_j\}$ , for every  $\omega \in \Gamma$ , the ratio  $\ell(\theta)/\ell(\omega)$  will shrink exponentially as a function of  $m$ . Furthermore, if we can take  $\omega$  further away from  $\{\theta_j\}$  the ratio will shrink even faster. This deformation of the contour  $\Gamma$ , however, is only possible if  $f$  is analytic in a larger region and this explains why functions with singularities close to the interval of approximation converge slowly as compared to functions whose singularities are further away.

The kind of reasoning presented in the last paragraph shows how powerful potential theory in the complex plane can be for developing a deeper understanding of trigonometric interpolation. As another example, we know that potential theory for algebraic polynomial interpolation leads to the concept of a largest ellipse to which a function  $f$  can be analytically continued [85, Ch. 12]. Also related are the so called *Bernstein ellipses* [85, Ch. 12]. Analogously, one can develop a similar potential theory for trigonometric polynomial interpolation leading to the concept of horizontal strips of maximum width to which a periodic function  $f$  can be analytically continued.

## 1.11 Lebesgue constant for trigonometric interpolation

One can also think of trigonometric polynomial interpolation as a linear operator. Consider the Banach space  $\mathcal{C}_p([0, 2\pi])$  of continuous periodic functions and the Banach space  $\mathcal{T}_m$  of trigonometric polynomials of degree  $m$  on the interval  $[0, 2\pi]$ , where both spaces are equipped with the supremum norm. Given a grid of  $2m + 1$  distinct points  $\theta_0, \theta_1, \dots, \theta_{2m}$ , in  $[0, 2\pi)$ , the operator that maps a periodic function  $f \in \mathcal{C}_p([0, 2\pi])$  to its trigonometric polynomial interpolant  $t \in \mathcal{T}_m$ , such that

$$t(\theta_k) = f(\theta_k), \quad 0 \leq k \leq 2m,$$

is a bounded linear operator. The *Lebesgue constant* is simply the norm of this linear operator. For the given grid of interpolation points in  $[0, 2\pi)$ , the Lebesgue constant  $\Lambda$  is defined as

$$\Lambda = \sup_{\|f\|_\infty \leq 1} \|t\|_\infty. \quad (1.11.1)$$

Another way of looking at Lebesgue constants is as follows. First note that the Lebesgue constant is dependent on the grid of interpolation points. Now, if the maximum absolute value of the functions we are interpolating is 1, then the Lebesgue constant equals the maximum absolute value an interpolant can take. In other words, the Lebesgue constant measures how much larger an interpolant can be than the function or data it interpolates on a given set of points. Lebesgue constants are intimately connected with the condition number of the interpolation problem and that is the reason they are important from a practical point of view. See [85, Ch. 15] for more details.

For a grid of equispaced points, we have the following theorem.

**Theorem 1.8.** *The Lebesgue constant  $\Lambda$  for trigonometric polynomial interpolation in  $2m + 1$  equispaced points satisfies*

$$\Lambda \leq \frac{2}{\pi} \log m + \frac{5}{3}, \quad m = 1, 2, \dots . \quad (1.11.2)$$

Furthermore,  $\Lambda$  also satisfies the asymptotic relationship

$$\Lambda \sim \frac{2}{\pi} \log m, \quad (1.11.3)$$

as  $m \rightarrow \infty$ .

*Proof.* See [27]. □

The above theorem shows that equispaced points are well suited for trigonometric polynomial interpolation since the Lebesgue constants have a very slow logarithmic growth as the number of equispaced points increases. In fact, a stronger statement is true. Equispaced points are optimal in the sense that they minimize the Lebesgue constant for any fixed set of  $2m + 1$  points (see [32] for a proof).

This result is in sharp contrast with the famous fact that equispaced points are not good for *algebraic* polynomial interpolation. In fact, the Lebesgue constants for algebraic polynomial interpolation in equispaced points grow exponentially [90], [92].

## 1.12 Connection between algebraic and trigonometric polynomial interpolation

The problem of algebraic polynomial interpolation can be mapped to trigonometric polynomial interpolation. In this section we make this explicit.

First, suppose that we are given  $n + 1$  distinct points  $-1 \leq x_0 < x_1 < \dots < x_n \leq 1$  on the interval  $[-1, 1]$  and data  $f_0, f_1, \dots, f_n$  (complex or real) that we wish to interpolate at these

points. It is well known (see, for example, [31]) that there exists a unique algebraic polynomial  $p_n(x)$  of degree  $\leq n$  such that  $p_n(x_k) = f_k$  for  $0 \leq k \leq n$ .

Now define the mapping

$$x = \cos \theta, \tag{1.12.1}$$

which maps the interval  $-1 \leq x \leq 1$  onto the interval  $0 \leq \theta \leq \pi$ . Under the same mapping, any function  $f(x)$  defined on  $[-1, 1]$  is transformed into a function  $g(\theta) := f(\cos \theta) = f(x)$  and the original interpolation points  $x_k$  are mapped to points  $\theta_k$ , where  $x_k = \cos \theta_k$ . If we now define the trigonometric polynomial  $t_n(\theta)$  as

$$t_n(\theta) := p_n(\cos \theta), \tag{1.12.2}$$

then  $t_n$  by definition is a trigonometric (cosine) polynomial of degree  $\leq n$  since the  $n^{\text{th}}$  power of the cosine function can be expressed as a trigonometric (cosine) polynomial of degree  $n$ . Also,  $t_n$  interpolates the same data  $f_k$  at the points  $\theta_k$  for  $0 \leq k \leq n$ . Observe that the cosine polynomial  $t_n$  is an even function of  $\theta$ ; therefore  $t_n$  not only interpolates the pair  $(\theta_k, f_k)$  but also the pair  $(-\theta_k, f_k)$  for  $0 \leq k \leq n$ .

On the other hand, given  $n + 1$  points  $0 \leq \theta_0 < \theta_1 < \dots < \theta_n \leq \pi$  and data  $f_k$  at these points, we can find a cosine polynomial of degree  $\leq n$  which interpolates this data. It can also be shown that  $\cos n\theta$  can be written as powers of the cosine function with the highest power being  $n$ . Therefore, using the mapping (1.12.1), we can map the cosine interpolating polynomial to an algebraic polynomial interpolant.

Hence the problem of interpolation using ordinary algebraic polynomials on the interval  $[-1, 1]$  is equivalent to trigonometric interpolation using cosine polynomials on  $[0, \pi]$  [103, Ch. X].

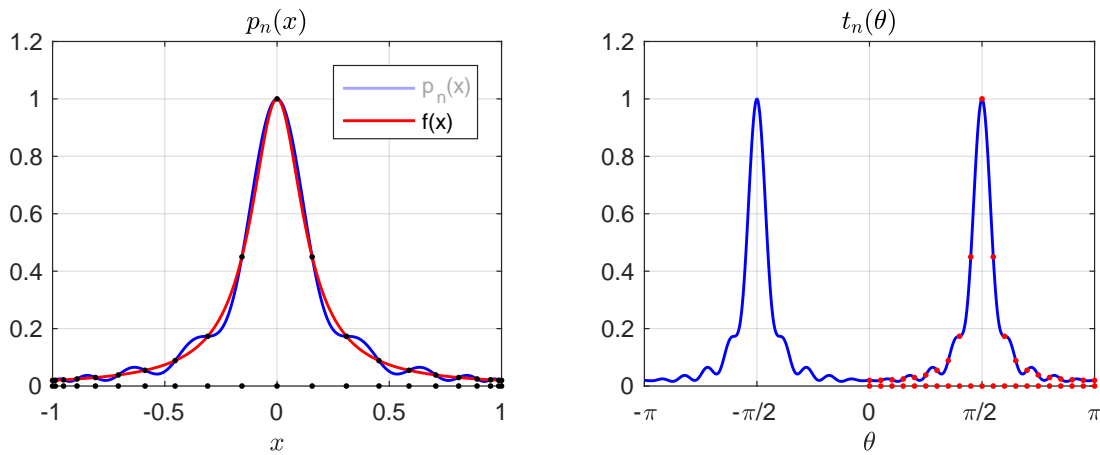


Figure 1.11: On the left we see the function  $f(x) = \frac{1}{1+50x^2}$  plotted in red together with its polynomial interpolant  $p_n(x)$  at 21 Chebyshev points plotted in blue. The Chebyshev points and the values interpolated by  $p_n$  at these points are shown by black dots. On the right we see the corresponding trigonometric (cosine) interpolant with the mapped equidistant points and the values interpolated at those points represented by red dots. Notice that the interval  $[-1, 1]$  on the left is mapped to the interval  $[0, \pi]$  on the right. The cosine polynomial is an even function of  $\theta$  and we plot it on the interval  $[-\pi, \pi]$ .

A special case of interest is algebraic polynomial interpolation at Chebyshev points in  $[-1, 1]$ , which is the same as trigonometric cosine interpolation at equidistant points in  $[0, \pi]$  due to the equivalence that we have just established and the fact that under the mapping defined by (1.12.1), Chebyshev points in  $[-1, 1]$  are mapped to equidistant points in  $[0, \pi]$ . In fact, every time a chebfun is constructed, this equivalence is invoked to utilize the FFT algorithm<sup>9</sup>. This equivalence is also used extensively in the design of digital filters, where even periodic functions are approximated by algebraic polynomials [67]. Figures 1.11 and 1.12 illustrate this beautiful equivalence.

<sup>9</sup>To be precise, the Discrete Cosine Transform (DCT) should be used, however, in the Matlab implementation of Chebfun a DCT is performed by computing an FFT on an array which concatenates the original data and its mirrored copy. See Battles [15] for more details.

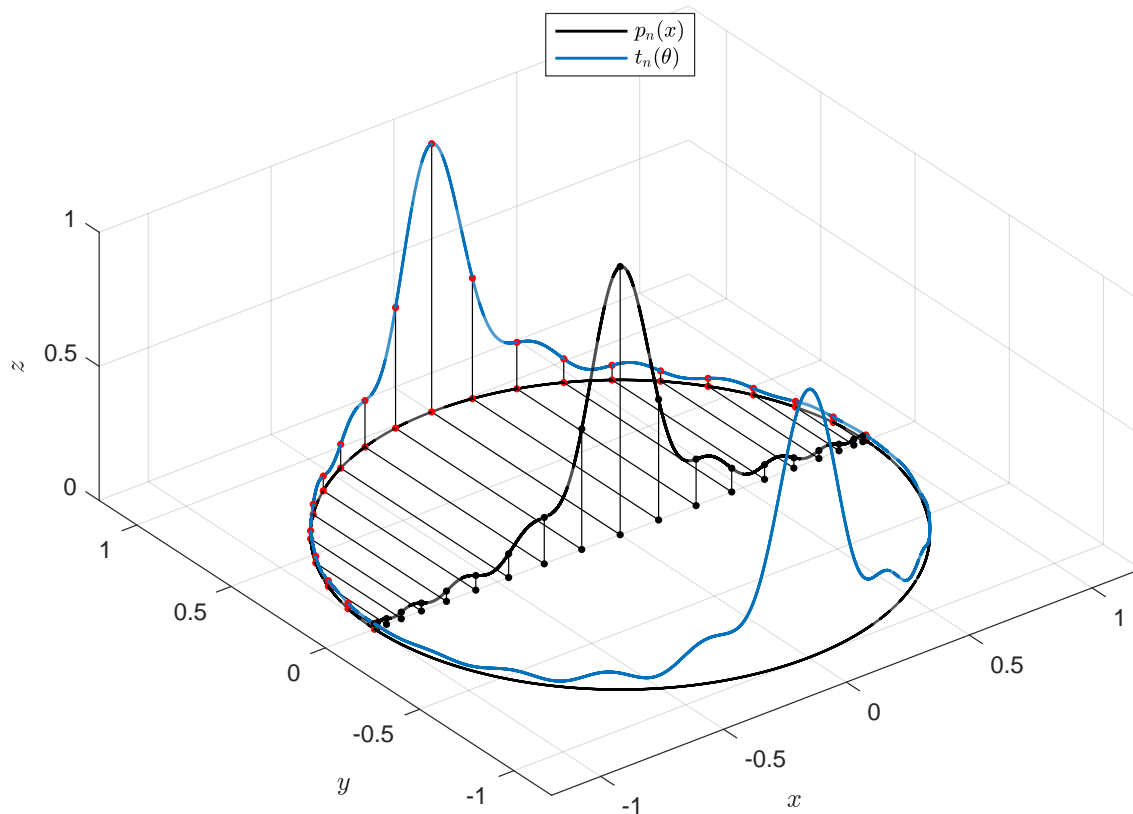


Figure 1.12: The polynomial interpolant of the function  $f(x) = \frac{1}{1+50x^2}$  at 21 Chebyshev points on  $[-1, 1]$  (black) and the corresponding trigonometric (cosine) interpolant (blue) wrapped around the unit circle. The black dots represent the Chebyshev points  $x_k$  and also the values  $p_n(x_k)$  interpolated by the polynomial interpolant. The red dots on and above the circle are the equidistant points  $\theta_k$  and the values  $t_n(\theta_k)$  interpolated by the trigonometric polynomial interpolant.

### 1.13 Numerical examples using Chebfun

The [Chebfun](http://www.chebfun.org)<sup>10</sup> software system is an open source project<sup>11</sup> written in object-oriented Matlab [15]. The fundamental principle of Chebfun is to compute numerically with functions instead of numbers and vectors [86]. Chebfun uses polynomial interpolants through Chebyshev points to approximate functions to relative machine precision. The approximation of a given function

<sup>10</sup>[www.chebfun.org](http://www.chebfun.org)

<sup>11</sup>[www.github.com/chebfun](https://www.github.com/chebfun)

is eventually stored as a vector of coefficients of its Chebyshev series. If the function is smooth enough, its Chebyshev series is rapidly convergent [85]. Chebfun uses an FFT based algorithm to automatically determine the number of coefficients needed to approximate a function to relative machine precision by converting the values of a function on a Chebyshev grid to the coefficients of its Chebyshev series [85]. The precise details of determining the right number of coefficients needed to approximate a function to a given degree of relative accuracy are given in [4].

To illustrate the idea of computing with functions, we consider the example<sup>12</sup> of the complicated looking function

$$f(x) = 1/(2 + \sin 22\pi x) + (1/2) \cos 13\pi x + 5e^{-80(x-.2)^2}.$$

To approximate  $f$  up to machine precision by a polynomial interpolant through Chebyshev points on the interval  $[-1, 1]$ , we can use the following Chebfun command.

```
f = chebfun(@(x) 1./(2+sin(22*pi*x))+cos(13*pi*x)/2+5*exp(-80*(x-.2).^2));
plot(f)
plotcoeffs(f, 'l')
```

The chebfun  $f$  and its Chebyshev coefficients are plotted in Figure 1.13. Chebfun determines that to approximate  $f$  to machine precision, a polynomial interpolant in 1803 Chebyshev points in  $[-1, 1]$  is needed.

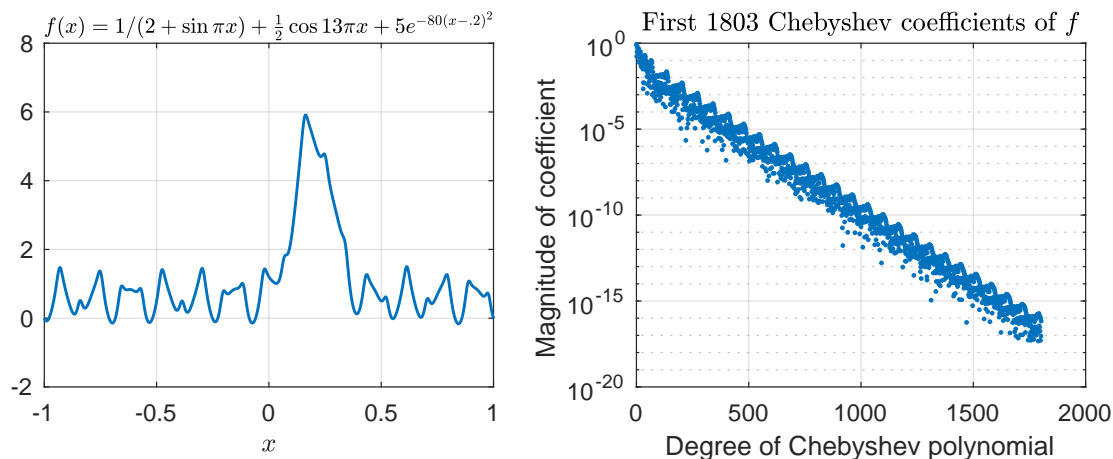


Figure 1.13: A function and its 1803 Chebyshev coefficients

<sup>12</sup>The examples in this section are modeled after various examples presented in the book *Approximation Theory and Approximation Practice* [85].

We can now consider  $\mathbf{f}$  as a function and numerically compute the roots, the maximum and the integral of  $f$  by the commands `roots(f)`, `max(f)` and `sum(f)` and so on. More examples can be seen on the [Chebfun](#) website<sup>13</sup>. The idea of computing with functions using Chebyshev polynomial interpolants was later extended to compute with periodic functions using trigonometric interpolants [102]. Due to an effort led by Grady Wright and others, including the author of this thesis, Chebfun now has the option of approximating periodic functions to machine precision by trigonometric polynomial interpolants through equispaced points [102]. The basic idea of automatically *resolving* a periodic function to machine precision is as follows. We know from Theorem 1.5 that any periodic function  $f$ , as long as it is Lipschitz continuous, has a convergent trigonometric series. Chebfun represents this periodic function as a finite trigonometric series of some degree  $m$ , storing both its values at  $2m + 1$  equidistant points and also, equivalently, its trigonometric series coefficients. To determine the right value of  $m$ , Chebfun samples the function at  $2m + 1$  points, converts this data to coefficients of a trigonometric series expansion of degree  $m$  and examines the resulting coefficients of the trigonometric series. If several of these in a row fall below a relative level of approximately  $10^{-15}$ , then the grid is judged to be fine enough and this is the right value of  $m$ . If this is not the case, the value of  $m$  is increased and the process is repeated. More details can be found in [4] and [102].

For example, to approximate the function  $f(\theta) = e^{.01 \cos \pi \theta}$  on the default interval  $[-1, 1]$ , and to examine the coefficients of its trigonometric series, we can type the following commands.

```
>> f = chebfun(@(t) exp(.01*cos(pi*t)), 'trig');
>> trigcoeffs(f)
ans =
    0.0000000000000026
    0.000000000026042
    0.000000020833464
    0.000012500104167
    0.005000062500260
    1.000025000156251
    0.005000062500260
    0.000012500104167
    0.000000020833464
    0.000000000026042
    0.0000000000000026
```

Notice that the first and the last coefficients are almost at the level of machine precision. The function and its Fourier coefficients are plotted in Figure 1.14.

---

<sup>13</sup>[www.chebfun.org/examples](http://www.chebfun.org/examples)

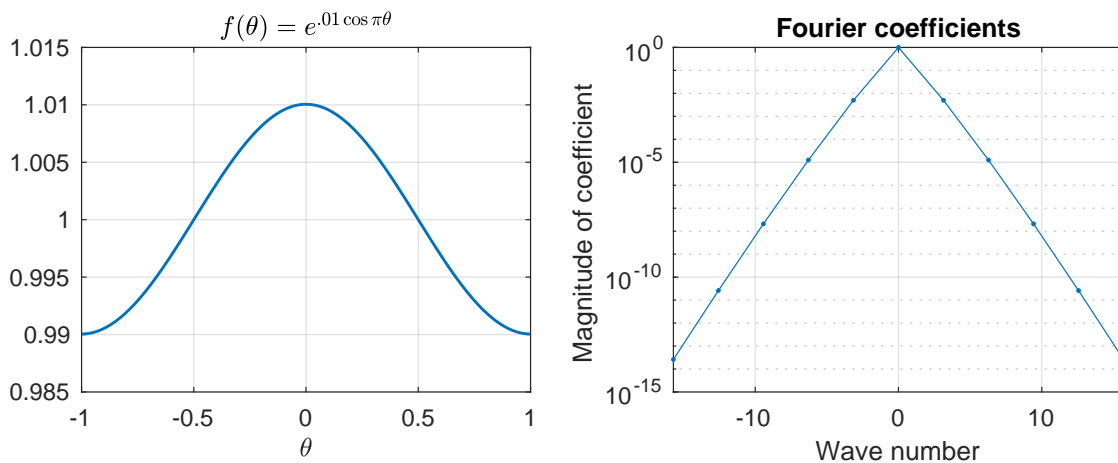


Figure 1.14: A function and its Fourier coefficients

We can revisit the function  $f(x) = 1/(2 + \sin 22\pi x) + (1/2) \cos 13\pi x + 5e^{-80(x-.2)^2}$ , which is periodic on  $[-1, 1]$  up to machine precision. Let us approximate  $f$  by a trigonometric polynomial interpolant using the command `chebfun` with the 'trig' flag:

```
f=chebfun(@(x)1./(2+sin(22*pi*x))+cos(13*pi*x)/2+5*exp(-80*(x-.2).^2), 'trig');
plot(f)
plotcoeffs(f, '.')
```

The result of this computation and the Fourier coefficients of `f` are now shown in Figure 1.15

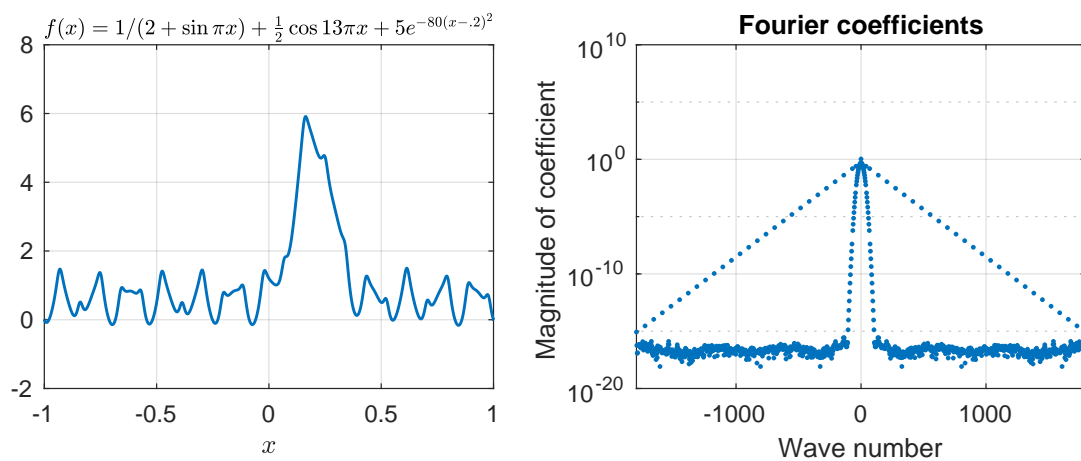


Figure 1.15: A function and its Fourier coefficients

We can also construct trigonometric polynomial interpolants to discontinuous and non-periodic functions as well. For example, to construct the degree 10 trigonometric interpolant to the discontinuous function  $f(\theta) = \text{sign}(\theta) - \theta$  through 21 equispaced points, we can call Chebfun in periodic mode by using the command `chebfun` with a second explicit argument of 21 and the 'trig' flag:

```
f = chebfun(@(t) sign(t) - t, 21, 'trig');
```

The chebfun `f` is plotted in Figure 1.16. Notice the behaviour at the left endpoint. Chebfun does not interpolate the function here but interpolates the mean value  $(f(1) + f(-1))/2$ , which is 0 in this case. If the function is continuous periodic (see Definition 1.1) on  $[-1, 1]$ , then of course  $f(1) = (f(1) + f(-1))/2 = f(-1)$  and no such anomalies occur.

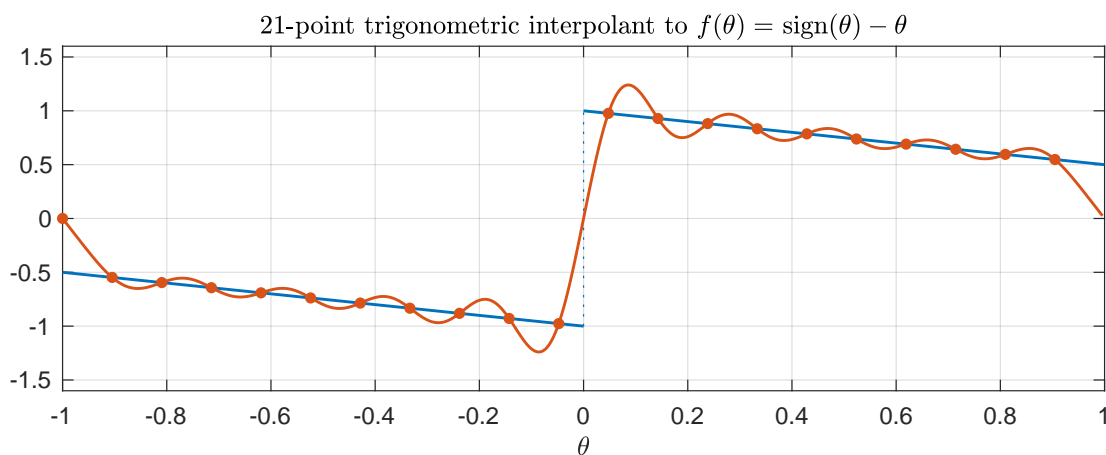


Figure 1.16: Trigonometric interpolant of a discontinuous function

Here is a more complicated function and its trigonometric polynomial interpolant in 101 points:

```
f = chebfun(@(t) sin(6*t) + sign(sin((t + exp(2*t))))), 101, 'trig');
```

The chebfun `f` is plotted in Figure 1.17 and we can see the Gibbs phenomenon [95] in action. For an interesting historical account of the Gibbs phenomenon, see [85, Ch. 9].

Chebfun has the capability to determine the location of points where a function is not smooth [69]. For example, Figure 1.17 shows that `f` is discontinuous at three points in  $(-1, 1)$  and using the 'splitting' option in Chebfun, we can compute a piecewise approximation of `f` consisting of four Chebyshev interpolants. This will get rid of the Gibbs phenomenon and the Chebfun approximation of `f` will be accurate to machine precision. This translates into a very powerful technique for computing with non-smooth functions [69]. However, such a solution is

not possible in the periodic case because the individual pieces of a periodic function are not necessarily periodic. For example, each of the four pieces of the periodic function shown in Figure 1.17 is non-periodic, and a piecewise periodic approximation will still suffer from the Gibbs phenomenon.

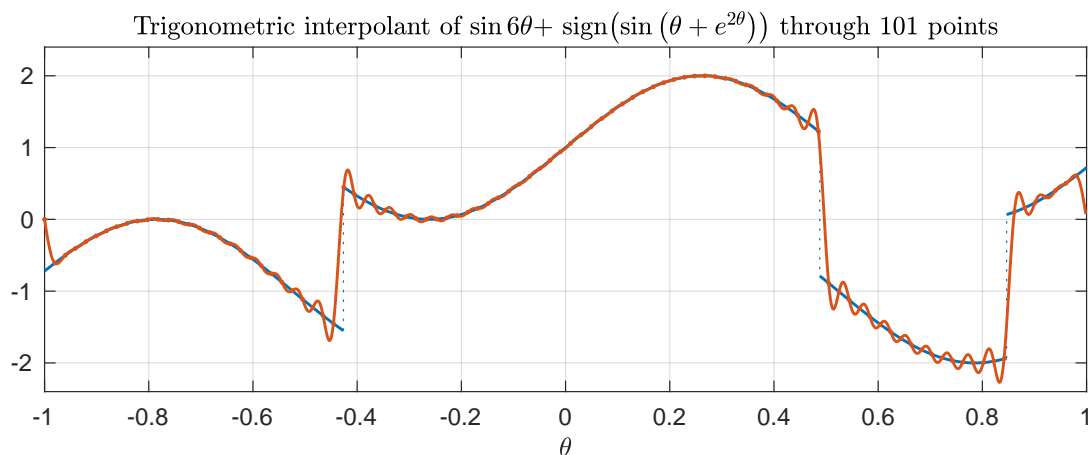


Figure 1.17: Trigonometric polynomial interpolant of a discontinuous function

Chebfun's capabilities are not limited to approximating real-valued functions only. We can also work with complex-valued functions of a real variable. In the next example, we consider a periodic complex-valued function on the interval  $[0, 2\pi]$  and approximate it by a trigonometric polynomial interpolant.

```
f = @(t) (3+sin(10*t)+sin(61*exp(.8*sin(t)+.7))).*exp(1i*t);
p = chebfun(f, [0, 2*pi], 'trig');
```

The plot in Figure 1.18 shows the image of  $[0, 2\pi]$  under  $f$ , which appears complicated but is nothing but a simple closed contour in the complex  $\theta$ -plane. Such complex-valued chebfuns are extremely useful in various applications. For example, if we were to compute the integral

$$I = \int_{\Gamma} g(t) dt, \tag{1.13.1}$$

where  $g$  is an analytic function and  $\Gamma$  is the contour shown in Figure 1.18, then we can parametrize the integral as

$$I = \int_{\Gamma} g(t) dt = \int_0^{2\pi} g(f(\theta))f'(\theta) d\theta. \tag{1.13.2}$$

Assuming  $g$  has been defined in Chebfun, we can compute  $I$  using the command,

```
I = sum(g(f).*diff(f))
```

Computing with functions makes it possible to effortlessly perform operations such as composition, multiplication, differentiation, integration and many more.

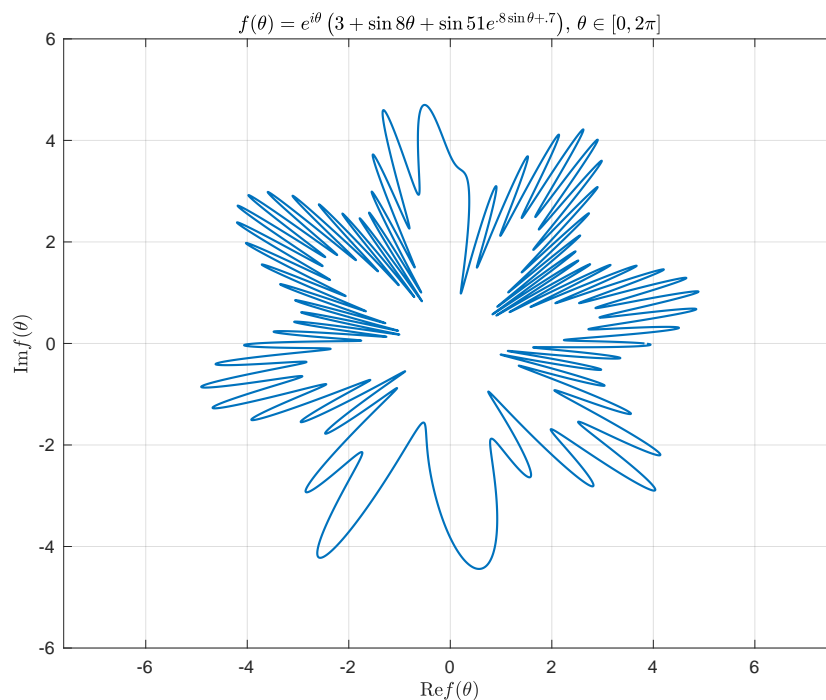


Figure 1.18: Trigonometric polynomial representation of a complex-valued function, which can also be considered as a parametrization of a closed contour.

Chebfun can also interpolate discrete data. To achieve this, we can use the `chebfun` command with the `'trig'` flag and instead of passing a function, pass a vector of data values. In this case, the data values are considered as samples of an unknown function on an equispaced grid. The number of grid points is taken equal to the number of data values passed and the outcome is a trigonometric polynomial interpolant of the appropriate order. If the number of data values is even, a balanced interpolant is returned. For example, to compute the trigonometric polynomial interpolant through 101 random data values in 101 equispaced points in  $[-1, 1)$ , we can write:

```
t = chebfun(-1 + 2*rand(101,1), 'trig');
```

The interpolant `t` is a trigonometric polynomial of degree 50 and can be seen in Figure 1.19. The example also highlights the robustness of trigonometric polynomial interpolation in equispaced points. Just as algebraic polynomial interpolants in Chebyshev points have superior approximation properties, trigonometric polynomial interpolants in equispaced points generate very good approximations of periodic functions.

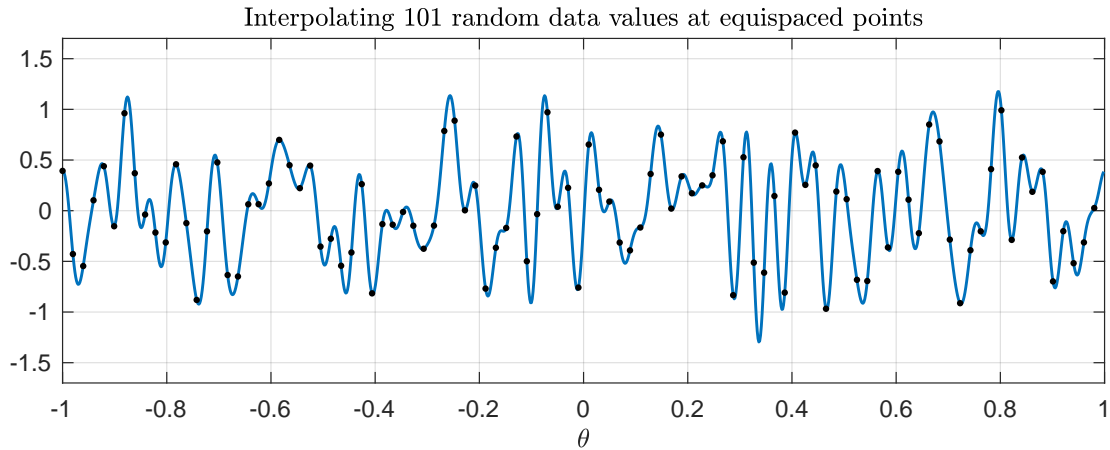


Figure 1.19: A trigonometric polynomial interpolant of random data at equispaced points

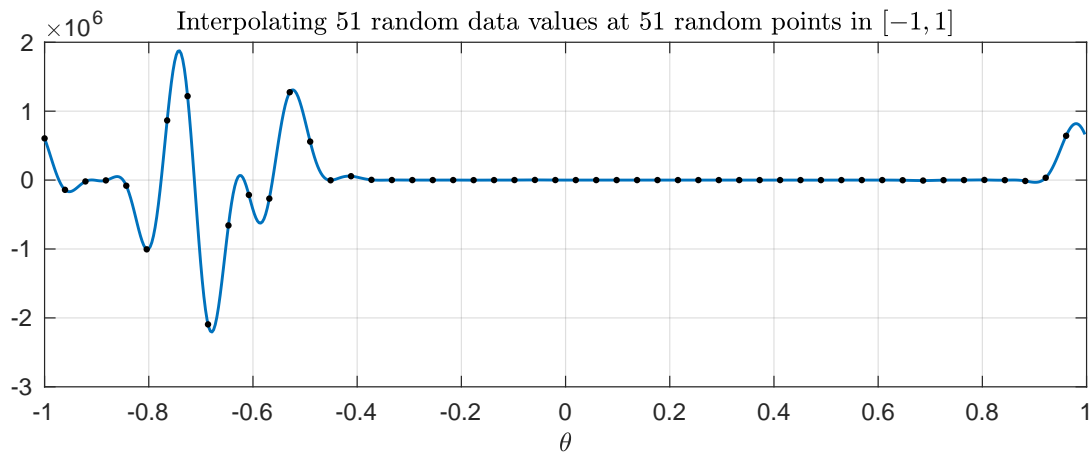


Figure 1.20: A trigonometric polynomial interpolant of random data at random points

On the other hand, we also know that ordinary polynomial interpolation using points which do not cluster quadratically at the end points of the interval of approximation is not a very good idea [85]. Similarly, trigonometric polynomial interpolation in points other than the equispaced points can lead to numerical difficulties. What follows is an example which highlights this problem. To illustrate this, we interpolate 51 random data values at 51 random points in the interval  $[-1, 1]$ . To achieve this, we use the command `chebfun.interp1` with the `'trig'` flag<sup>14</sup>

<sup>14</sup>This `'trig'` capability to interpolate data at arbitrary points by trigonometric polynomials was added to the existing `chebfun.interp1` command by the author of this thesis. The code is freely available from <https://www.github.com/chebfun/chebfun>.

```
th = -1 + 2*rand(51, 1); fk = -1 + 2*rand(51, 1);
t = chebfun.interp1(th, fk, 'trig', [-1, 1]);
```

This results in an interpolant with magnitude of the order  $10^6$ , indicating a very large Lebesgue constant. The interpolant is plotted in Figure 1.20.

One might think that the last example failed only because the data to be interpolated was random. In fact, this is not true. One can choose a function which is very smooth on the interval of approximation and yet run into severe problems while interpolating the function through random points. Let us consider the function

$$f(\theta) = \frac{1}{1 + 25 \sin^2(\pi\theta/2)}, \quad (1.13.3)$$

on the interval  $[-1, 1]$ . We shall refer to this function as a *periodic Runge function*. This function is analytic on  $[-1, 1]$ , but has poles in the complex  $\theta$ -plane. The two poles closest to the origin are located at the points

$$\pm \frac{i}{\pi} \sinh^{-1}(1/5) \approx \pm 0.12649i.$$

The function  $f$  and its phase portrait are plotted in Figure 1.21.

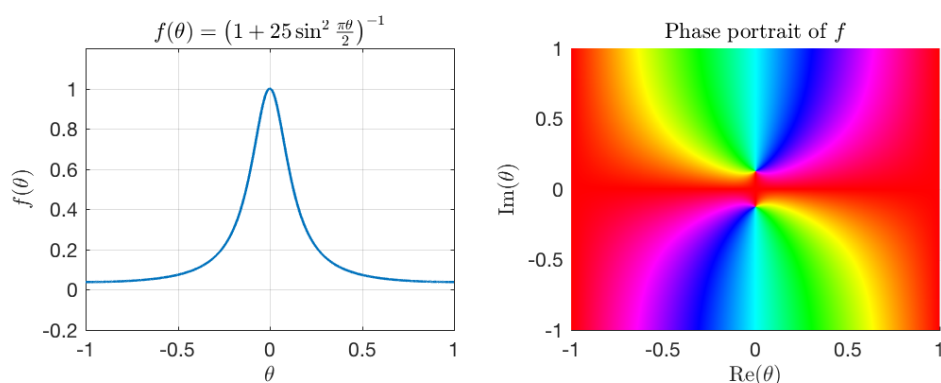


Figure 1.21: The *periodic Runge function*  $f(\theta) = 1/(1 + 25 \sin^2(\pi\theta/2))$  is plotted on the left. On the right, we plot the phase portrait of  $f$ , which shows the two symmetric poles of  $f$  located close to the origin on the imaginary axis.

Let us first approximate the function by trigonometric polynomial interpolants at 21 randomly chosen points in  $[-1, 1]$ . We can see 9 such interpolants in Figure 1.22, which shows that almost all of these interpolants behave erratically. One might argue that perhaps the number of interpolation points was not enough in the last experiment. However, increasing the number of random points of interpolation does not resolve the function. It actually makes the matter worse and the resulting interpolants wiggle with even larger global errors.

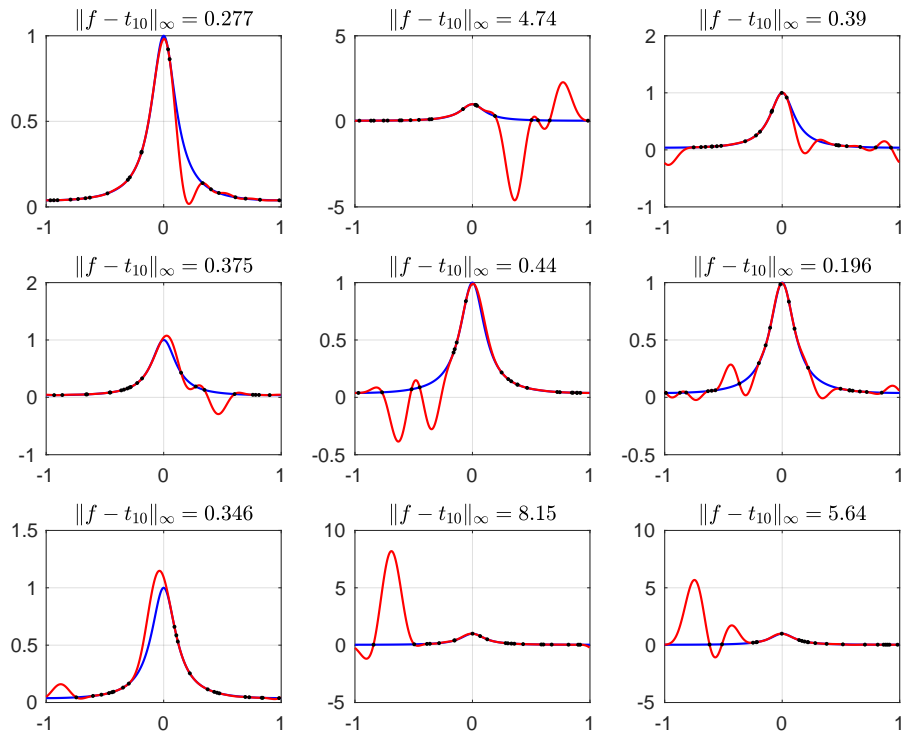


Figure 1.22: Trigonometric polynomial interpolants of degree 10 to the function  $f(\theta) = 1/(1 + 25(\sin \pi\theta/2)^2)$  at 21 random points. All interpolants exhibit poor approximation of the function  $f$ .

The pictures change completely when we start interpolating the function at equispaced points. The interpolants converge geometrically to the original function as shown in Figure 1.23.

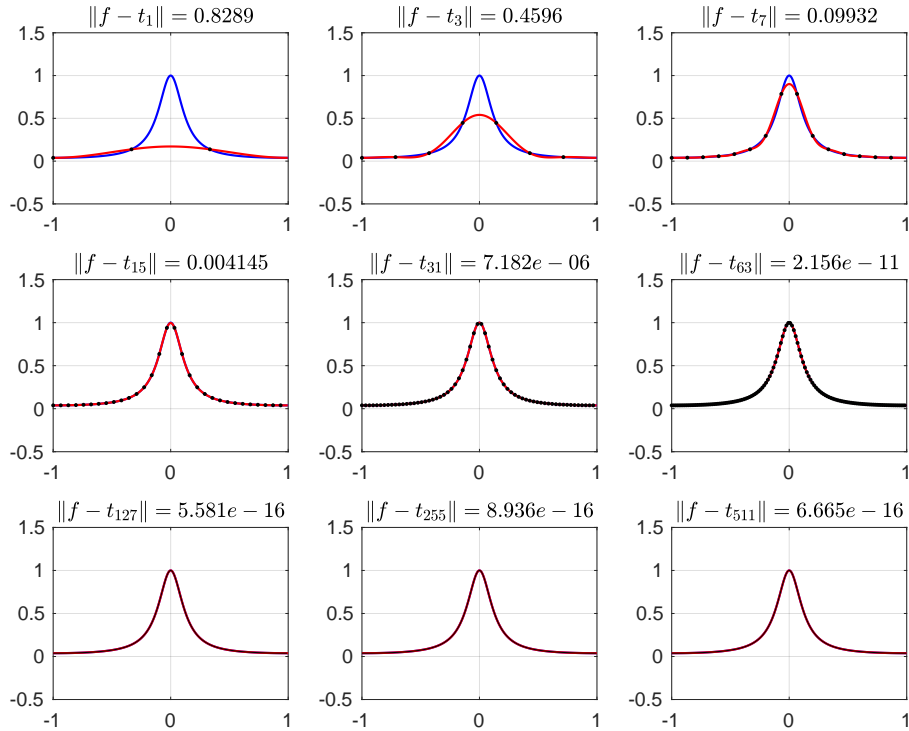


Figure 1.23: The power and robustness of trigonometric polynomial interpolants at equispaced points.

## 1.14 Conclusion

In this chapter we have reviewed trigonometric polynomial interpolation and the trigonometric barycentric formula for evaluating such interpolants. We then developed a trigonometric analogue of the Hermite integral formula which expresses the error in a trigonometric polynomial interpolant of a periodic analytic function as a contour integral. We believe that this formula and its derivation do not exist in the literature. Finally, we introduced Chebfun and illustrated its capabilities to approximate periodic functions by trigonometric polynomial interpolants.

After this grand tour of trigonometric polynomial interpolation, we move to trigonometric rational interpolation in the next chapter.

## Chapter 2

# Trigonometric Rational Interpolation and Least-Squares

In this chapter we focus on the problem of numerically computing trigonometric rational interpolants and trigonometric rational least-squares approximations of functions defined on an interval. There are existing algorithms for numerically computing rational interpolants and rational least-squares approximations of the form  $p/q$  where  $p$  and  $q$  are ordinary algebraic polynomials (see, for example [40], [68]), but we believe that this chapter describes for the first time an algorithm for computing trigonometric analogues of such approximations.

While describing the mathematics of the algorithms, we will assume that  $f$  is a function defined on a periodic interval<sup>1</sup>  $[a, b]$ . However, the software implementation of our algorithms is independent of the choice of any particular periodic interval.

### 2.1 Introduction

Approximation of functions by rational functions is a classical problem. There is a huge body of literature which discusses various variants of the problem. One of the oldest references is by Cauchy, published in 1821 [24], in which he views the problem of rational interpolation as a generalization of the Lagrange formula for polynomial interpolation [68]. In particular, Cauchy shows that if the prescribed rational interpolant has a numerator of degree  $m$  and a denominator of degree  $n$ , then setting the number of interpolation points equal to  $m + n + 1$  guarantees the uniqueness of the solution [68]. A couple of decades later in 1846, Jacobi presented his solution of the rational interpolation problem in [53]. Jacobi's solution provided a template for many algorithms that followed [68]. Walsh's treatise [96] is a classic reference on the theory of rational

---

<sup>1</sup>See the introductory section in Chapter 1 for an explanation of the term *periodic interval*.

interpolation. Other good references include the books by Braess [20], Cheney [25], Davis [31], and Rice [77]. For a numerically oriented introduction, see Trefethen [85, Ch. 26].

The literature mentioned above and numerous other books and papers discuss the problem of rational interpolation and rational least-squares approximation for the case where the approximants have the form  $p/q$ ,  $p$  and  $q$  being algebraic polynomials. Approximants where  $p$  and  $q$  are trigonometric polynomials are rarely discussed, if ever. In this chapter, we are going to discuss this case.

Gragg [43] was one of the first one to suggest numerical algorithms for trigonometric rational approximations. In [43], he presented an extension of Padé approximation for periodic functions, the so called “Fourier-Padé” approximations. However, the problem of trigonometric rational interpolation was not discussed. In [18], Berrut described an algorithm for constructing a barycentric rational trigonometric interpolant guaranteed to have no poles within the interval of approximation. However, the algorithm left the degree of numerator and denominator out of consideration and did not allow one to specify the type  $(m, n)$  of the interpolant. In [38], Geer also discusses trigonometric rational approximations but these are again “Fourier-Padé” approximations and not interpolants. In [14], Baltensperger discusses a special case of barycentric trigonometric rational interpolants but the main focus is on computing derivatives of such interpolants.

The situation is similar when it comes to software. While various numerical algorithms have been implemented to compute rational interpolants for the algebraic case, we know of no software to compute trigonometric rational interpolants and least-squares approximations.

In this chapter we will present a new algorithm for trigonometric rational interpolation. We will see that the problem of interpolation or least-squares approximation by trigonometric rational functions leads to a singular value decomposition of a certain matrix. The method developed in this chapter can be seen as the trigonometric analogue of an algorithm due to Pachón, Gonnet and Van Deun [68]. Since stability is a crucial problem for rational interpolation, following [40], we will also describe ways of making our algorithm robust.

## 2.2 Trigonometric rational interpolation and least-squares approximation at arbitrary points

We begin by discussing the problem of trigonometric rational interpolation and least-squares approximation for the general case when the function values to be interpolated or approximated are given on an arbitrary grid of distinct points in a periodic interval  $[a, b]$ . We continue to use the notation established already in Chapter 1.

A trigonometric rational function  $r$  is a periodic function that can be written as a ratio of two trigonometric polynomials  $p$  and  $q$ . For non-negative integers  $m$  and  $n$ , we say that  $r$  is of *type*  $(m, n)$  if  $p$  and  $q$  can be chosen such that  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$ , that is,

$$r(\theta) = \frac{p(\theta)}{q(\theta)} = \frac{a_0 + \sum_{k=1}^m (a_{2k-1} \sin \frac{2k\pi}{T} \theta + a_{2k} \cos \frac{2k\pi}{T} \theta)}{b_0 + \sum_{k=1}^n (b_{2k-1} \sin \frac{2k\pi}{T} \theta + b_{2k} \cos \frac{2k\pi}{T} \theta)}, \quad (2.2.1)$$

where  $T = b - a$ .

Note that it is not necessary for  $p$  and  $q$  to have exact degrees  $m$  and  $n$  respectively. We say that  $r = p/q$  is of *exact type*  $(m, n)$  if  $p$  and  $q$  are of exact degrees  $m$  and  $n$  respectively and there are no roots in common between  $p$  and  $q$ . In case  $r$  is identically zero, we say that it has the exact type  $(-\infty, 0)$ . We denote the set of trigonometric rational functions of type  $(m, n)$  defined on  $[a, b]$  by  $R_{mn}$ .

Given  $m \geq 0$  and  $n \geq 0$ , our aim is to compute interpolants and approximants of the function  $f$  defined on a periodic interval  $[a, b]$  from the set  $R_{mn}$ . The approximations, whether defined via interpolation or via least-squares, will be based on the values taken by  $f$  on a set of distinct grid points in the interval  $[a, b)$ .

### 2.2.1 The interpolation problem

Let us now state the problem of trigonometric rational interpolation. We define  $N = m + n$  and suppose that we are given  $2N + 1$  distinct points  $\theta_0, \theta_1, \dots, \theta_{2N}$  in the interval  $[a, b)$  and data in the form of  $2N + 1$  real or complex numbers  $f_k$ , where  $0 \leq k \leq 2N$ . Our problem is to find a trigonometric rational function  $r \in R_{mn}$  which satisfies the following interpolation conditions:

$$r(\theta_k) = \frac{p(\theta_k)}{q(\theta_k)} = f_k, \quad 0 \leq k \leq 2N, \quad (2.2.2)$$

where  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$ . This interpolation problem, as we will explain in Section 2.2.3, does not always have a solution. To avoid the problem of non-existence and for the numerical solution of the problem, our algorithm will start with the linearized version of (2.2.2):

$$p(\theta_k) - q(\theta_k) f_k = 0, \quad 0 \leq k \leq 2N. \quad (2.2.3)$$

Note that the number of degrees of freedom in the original problem (2.2.2) and in the linearized problem (2.2.3) are both equal to the number of equations: there are  $2m + 1$  degrees of freedom in  $p$ , and only  $2n$  degrees of freedom in  $q$  because we can always divide both  $p$  and  $q$  by a non zero constant without changing the solution. In practice, this division by a constant is used to enforce some kind of normalization on  $q$ . Therefore, there are  $2(m + n) + 1 = 2N + 1$  degrees of freedom and as many equations, both in (2.2.2) and (2.2.3), and it would seem that the count is right for the interpolation problem to be solvable. However, as we shall see shortly, this is not always the case.

### 2.2.2 The linearized least-squares problem

If the data is available at more than  $2(m+n)+1$  points, that is, when  $N > m+n$ , then there are more equations than unknowns. In this case, we define the error at a point of interpolation  $\theta_k$  as

$$e_k = p(\theta_k) - f_k q(\theta_k), \quad k = 0, 1, \dots, 2N, \quad (2.2.4)$$

and seek a solution such that the sum of the squares of the error at the points of interpolation is minimized. This is our definition of the linearized trigonometric rational least-squares problem: find  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$  which minimize the quantity

$$\|\mathbf{e}\|_{2N+1} := \sqrt{\sum_{k=0}^{2N} |p(\theta_k) - f_k q(\theta_k)|^2}, \quad (2.2.5)$$

where  $\mathbf{e}$  is the  $2N+1$ -vector

$$\mathbf{e} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{2N} \end{bmatrix}. \quad (2.2.6)$$

Besides the linearized least-squares problem, one can also consider the *true* non-linear least-squares problem of minimizing the quantity

$$\sum_{k=0}^{2N} \left| f_k - \frac{p(\theta_k)}{q(\theta_k)} \right|^2. \quad (2.2.7)$$

As one might expect, the above non-linear problem is much harder and of a completely different character. We will not consider this problem in this thesis.

### 2.2.3 Existence and uniqueness of trigonometric rational interpolants

The existence of a trigonometric rational function  $r \in R_{mn}$  which satisfies (2.2.2) is not always guaranteed. The situation is analogous to the non-existence of rational interpolants in the algebraic case [20, Ch. V, Sec. 3], [78, Sec. 5.4]. Consider for example the problem where we seek a trigonometric rational interpolant  $r \in R_{11}$  on the periodic interval  $[0, 2\pi]$ , where  $r$  satisfies the following conditions:

$$r(0) = 1, \quad r(\pi/4) = -1, \quad r(\pi/2) = 1, \quad r(\pi) = 1, \quad r(3\pi/2) = 1. \quad (2.2.8)$$

Since a function in  $R_{11}$  is determined by five parameters, the five interpolation conditions prescribed above should be sufficient to solve this problem. To solve the above system, we let

$$r(\theta) = \frac{p(\theta)}{q(\theta)} = \frac{a_0 + a_1 \sin \theta + a_2 \cos \theta}{b_0 + b_1 \sin \theta + b_2 \cos \theta} \quad (2.2.9)$$

where the coefficients of  $q$  are assumed to satisfy the normalization condition  $b_0^2 + b_1^2 + b_2^2 = 1$ .

Using all the interpolation conditions given in (2.2.8), except the condition  $r(\pi/4) = -1$ , and using the expression of  $r$  as above, we get a system of equations which forces

$$a_0 = b_0, \quad a_1 = b_1, \quad \text{and} \quad a_2 = b_2. \quad (2.2.10)$$

The form of  $r$  is therefore

$$r(\theta) = \frac{p(\theta)}{q(\theta)} = \frac{b_0 + b_1 \sin \theta + b_2 \cos \theta}{b_0 + b_1 \sin \theta + b_2 \cos \theta}, \quad (2.2.11)$$

and since all  $b_k$  are not zero due to the normalization condition, the numerator and the denominator cancel each other out and we get

$$r(\theta) \equiv 1. \quad (2.2.12)$$

Therefore, this particular interpolation problem has no solution<sup>2</sup> because we need  $r(\pi/4) = -1$ . More generally it can be shown that if a function in  $R_{11}$  takes the same value at four distinct points, it must be a constant. This phenomenon of non-existence in the present case of trigonometric rational interpolants may also be compared with the phenomenon of *unattainable points*, a term used to describe analogous problem of non-existence in the case of algebraic rational interpolants [78, Sec. 5.4].

While the existence of a solution for the original interpolation problem (2.2.2) is not guaranteed, a solution of the linearized problem (2.2.3), as we shall prove by construction, always exists. Let us take another look at the linearized problem (2.2.3) where we seek trigonometric polynomials  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$  such that

$$p(\theta_k) - f_k q(\theta_k) = 0, \quad 0 \leq k \leq 2N. \quad (2.2.13)$$

The above equation is trivially satisfied if  $p$  and  $q$  are identically zero. To avoid this degeneracy, we need to impose a normalizing condition on the denominator  $q$ . Let

$$q(\theta) = b_0 + \sum_{k=1}^n (b_{2k-1} \sin k\theta + b_{2k} \cos k\theta),$$

and let  $\mathbf{b}$  denote the  $(2n+1)$ -vector of coefficients  $b_k, 0 \leq k \leq 2N$ . The normalization condition that we impose on  $q$  is given as

$$\|\mathbf{b}\|_2 = 1. \quad (2.2.14)$$

As mentioned earlier, the existence of  $p$  and  $q$  satisfying the linearized problem (2.2.13) and (2.2.14) is guaranteed. However,  $r = p/q$  will not always satisfy the interpolation conditions of the original problem (2.2.2). There are two ways in which a solution of the linearized problem (2.2.13) may still be considered problematic [20]:

---

<sup>2</sup>We believe that this is the first example which explicitly shows the non-existence of trigonometric rational interpolants in general.

1. The linearized solution may give rise to quotients of the form  $0/0$ . For example, consider the following interpolation problem. The points of interpolation are

$$\theta_0 = -1, \quad \theta_1 = -1/2, \quad \theta_2 = 0, \quad \theta_3 = 1/4, \quad \text{and} \quad \theta_4 = 3/4.$$

The data to be interpolated are

$$f_0 = 1, \quad f_1 = 2, \quad f_2 = 1, \quad f_3 = 1, \quad \text{and} \quad f_4 = 1.$$

We now seek a type  $(1,1)$  trigonometric rational interpolant on the periodic interval  $[-1, 1]$ .

Formally, the degenerate function

$$r(\theta) = \frac{\cos(\pi\theta)}{\cos(\pi\theta)}$$

interpolates the data at all but one of the interpolation points. For  $\theta = -1/2$ , we see that  $r$  has the form  $0/0$ .

2. It may happen that the linearized solution satisfies the interpolation conditions (2.2.2) but has poles between the points of interpolation. This is not always a problem, for example, if we are interpolating the function  $1/\sin(x/2)$  on  $[-\pi, \pi]$  in a set of points not containing the origin, we would like the trigonometric rational interpolant to reproduce the pole at the origin. However, sometimes the interpolant might have poles where the underlying function has no poles. One such example is presented in Section 2.5 where we discuss the phenomenon of spurious poles.

These issues can hardly be avoided if we only consider interpolation. However, following [40], we shall see that the least-squares approximations are much more robust where we take  $N > m + n$  and solve the discrete least-squares problem (2.2.5) to find  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$  such that the discrete error  $\|\mathbf{e}\|_{2N+1}$  is minimized on the grid.

#### 2.2.4 The choice of basis

Before we present the solution of the linearized problem, let us say a few words about the choice of basis. For interpolation or least-squares approximation, it does not matter whether we use the complex exponential basis or the sine-cosine basis for the representation of trigonometric polynomials. However, as we shall see in the next chapter, algorithms used for computing the best trigonometric approximations in the supremum norm do depend upon the real-valuedness of the function to be approximated. The algorithms for complex minimax approximation are very different. In this thesis, we shall only be discussing minimax approximations of real-valued functions. In particular, the Remez algorithm for computing the minimax approximation

relies on real-valued trigonometric interpolation, both in the polynomial and the rational case. Therefore, it is advantageous to develop our algorithm in the sine-cosine basis, and unless otherwise stated, we will assume that a trigonometric polynomial  $t$  of degree  $n$ , defined on a periodic interval  $[a, b]$  has the form

$$t(\theta) = c_0 + \sum_{k=1}^n \left( c_{2k-1} \sin \left( \frac{2\pi k}{T} \theta \right) + c_{2k} \cos \left( \frac{2\pi k}{T} \theta \right) \right), \quad (2.2.15)$$

where  $T = b - a$  and the coefficients  $c_j$  are real or complex numbers.

### 2.2.5 Solution of the linearized interpolation and least-squares approximation

Let us now look at the linear algebra involved in solving the linearized interpolation and least-squares problem. Without loss of generality, consider the periodic interval  $[0, 2\pi]$  and let the degree- $n$  trigonometric polynomial  $q$  be expressed as

$$q(\theta) = b_0 + \sum_{k=1}^n (b_{2k-1} \sin k\theta + b_{2k} \cos k\theta). \quad (2.2.16)$$

Let the coefficients of  $q$  be represented by the  $(2n + 1)$ -vector  $\mathbf{b} \in \mathbb{C}^{2n+1}$ ,

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{2n} \end{bmatrix}, \quad (2.2.17)$$

such that  $\mathbf{b}$  satisfies the normalization

$$\|\mathbf{b}\|_2 = 1. \quad (2.2.18)$$

Note that other normalizations can also be considered but for numerical computations this is a natural choice since, as we shall see,  $\mathbf{b}$  will be computed as a block component of a singular vector of a matrix. (A similar normalization is considered by Gonnet, Pachón and Trefethen for the algebraic rational interpolation problem in [40].)

We also express the degree- $m$  trigonometric polynomial  $p$  as

$$p(\theta) = a_0 + \sum_{k=1}^m (a_{2k-1} \sin k\theta + a_{2k} \cos k\theta), \quad (2.2.19)$$

and let the coefficients of  $p$  be represented by the  $(2m + 1)$ -vector  $\mathbf{a} \in \mathbb{C}^{2m+1}$ ,

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2m} \end{bmatrix}. \quad (2.2.20)$$

We can express the interpolation conditions

$$p(\theta_k) - f(\theta_k)q(\theta_k) = 0, \quad 0 \leq k \leq 2N, \quad (2.2.21)$$

in matrix-vector form by writing

$$P\mathbf{a} - FQ\mathbf{b} = 0, \quad (2.2.22)$$

where  $F$  is a diagonal square matrix of size  $(2N + 1) \times (2N + 1)$ ,

$$F = \begin{bmatrix} f_0 & & & & \\ & f_1 & & & \\ & & \ddots & & \\ & & & & f_{2N} \end{bmatrix}, \quad (2.2.23)$$

$P$  is the  $(2N + 1) \times (2m + 1)$  matrix

$$P = \begin{bmatrix} 1 & \sin \theta_0 & \cos \theta_0 & \dots & \sin m\theta_0 & \cos m\theta_0 \\ 1 & \sin \theta_1 & \cos \theta_1 & \dots & \sin m\theta_1 & \cos m\theta_1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \sin \theta_{2N} & \cos \theta_{2N} & \dots & \sin m\theta_{2N} & \cos m\theta_{2N} \end{bmatrix}, \quad (2.2.24)$$

and  $Q$  is the  $(2N + 1) \times (2n + 1)$  matrix

$$Q = \begin{bmatrix} 1 & \sin \theta_0 & \cos \theta_0 & \dots & \sin n\theta_0 & \cos n\theta_0 \\ 1 & \sin \theta_1 & \cos \theta_1 & \dots & \sin n\theta_1 & \cos n\theta_1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \sin \theta_{2N} & \cos \theta_{2N} & \dots & \sin n\theta_{2N} & \cos n\theta_{2N} \end{bmatrix}. \quad (2.2.25)$$

We can rewrite (2.2.22) as

$$\begin{bmatrix} P & -FQ \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = 0, \quad (2.2.26)$$

which reduces our problem to that of finding the null-space of the matrix  $[P - FQ]$ . The dimension of the matrix  $[P - FQ]$  is  $(2N + 1) \times (2(m + n) + 2)$  and for interpolation, we take  $N = m + n$ , so the matrix has  $2N + 1$  rows and  $2N + 2$  columns. Therefore, there is always at least one non-zero solution to the linearized interpolation problem. We state this formally in the following theorem.

**Theorem 2.1.** *The linearized interpolation problem (2.2.3) always has at least one non-zero solution.*

*Proof.* This follows from the paragraph above. □

### 2.2.6 Handling even and odd functions

We now look at two special cases of symmetry in the interpolation problem. If the function to be interpolated has even or odd symmetry, i.e. if the points of interpolation are symmetric and if the data to be interpolated on those points is either even or odd, we can further simplify the matrices involved in the solution of the problem.

If the function  $f$  is even, then a type  $(m, n)$  interpolant  $r = p/q$  is a ratio of two cosine polynomials of degree  $m$  and  $n$ , respectively. On the other hand if  $f$  is an odd function, then a type  $(m, n)$  interpolant  $r = p/q$  is a ratio of a sine polynomial of degree  $m$  and a cosine polynomial of degree  $n$ .

In each case, the symmetry allows us to cut the problem size roughly in half. Suppose as before that  $N = m + n$  and the data is given at  $2N + 1$  symmetric points of a periodic interval. If the data has even or odd symmetry, we can discard data at  $N$  of the rightmost or the leftmost points of interpolation and solve a smaller problem of interpolating the remaining  $N + 1$  data points by an even or an odd trigonometric rational function. The only change in the derivation of the solution just presented happens in the columns of the matrices  $P$  and  $Q$ , given by equations (2.2.24) and (2.2.25), respectively. The columns of  $P$  and  $Q$  are now formed by samples of sines only or cosines only, depending on the symmetry of the function  $f$ .

## 2.3 Numerical Computation

### 2.3.1 Basic algorithm

We have thus far shown that the linearized trigonometric rational interpolation problem leads to the homogeneous system of linear equations (2.2.26). To compute the null space of this system in a numerically stable way, in our first basic algorithm, we use the singular value decomposition (SVD). Here is a short pseudo-code outline of our first algorithm for the numerical computation of trigonometric rational interpolation and least-squares approximation.

---

**Algorithm 1** Trigonometric Rational Interpolation and Least-Squares

---

```
1: procedure TRIGRATINTERP( $f, m, n, N, \theta_k$ )
2:   if isempty( $N$ ) then
3:      $N \leftarrow 2(m + n) + 1$  ▷ Assume interpolation if  $N$  not given
4:   if isempty( $\theta_k$ ) then
5:      $\theta_k \leftarrow \text{trigpts}(N)$  ▷ Use equispaced points if  $\theta_k$  not provided
6:    $f_k \leftarrow f(\theta_k)$  ▷ Sample  $f$  at  $\theta_k$ 
7:   Construct matrices  $P, Q$  and  $F$ 
8:   Find the null space of  $[P, -FQ]$  via SVD
9:   Extract the coefficients  $\mathbf{a}$  and  $\mathbf{b}$ 
10:  return the coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$ 
```

---

## 2.4 Numerical Examples

### 2.4.1 Reproducing simple trigonometric rational functions

Let us first present a very simple example of trigonometric rational interpolation. We pick the function  $f(\theta) = \tan(\pi\theta)$  on the interval  $[-1, 1]$ .  $f$  has poles at  $\theta = \pm 1/2$ , however, we do not sample  $f$  at these points and interpolate it in 5 equidistant points by a type  $(1, 1)$  rational function. Since  $\tan(\pi\theta) = \frac{\sin(\pi\theta)}{\cos(\pi\theta)}$  is a type  $(1, 1)$  rational function already, we expect the type  $(1, 1)$  rational interpolant to reproduce  $f$ .

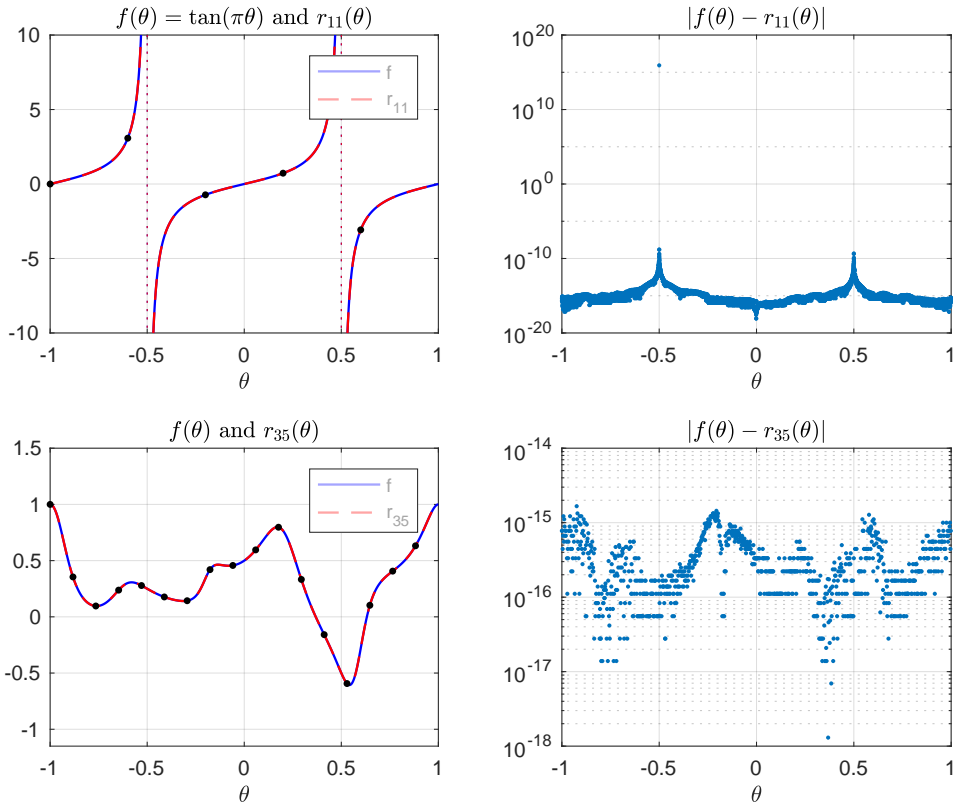


Figure 2.1: Reproducing simple trigonometric rational functions by rational interpolants. In the top row, we interpolate the function  $f(\theta) = \tan \pi\theta$  by a type  $(1, 1)$  interpolant at 5 equidistant points in  $[-1, 1]$ . Interpolated values are shown by black dots and the error at 10000 equispaced points in  $[-1, 1]$  is plotted in the top right window. The condition number for evaluating  $f$  in a small neighbourhood of the points  $\theta = \pm 1/2$  is large, and this results in spikes in the error plot at these two points. In the bottom row, we interpolate the function  $f(\theta) = (1 + \cos 2\pi\theta + \sin 3\pi\theta)/(3 + \sin 3\pi\theta + \cos 5\pi\theta)$  by a type  $(3, 5)$  interpolant at 17 equidistant points in  $[-1, 1]$ . Interpolated values are shown with black dots and the error at 1000 equispaced points in  $[-1, 1]$  (caused by floating-point rounding) is plotted in the bottom right window.

To invoke the algorithm, we execute the following lines of code, which set up the problem and compute the interpolant using the `trigratinterp` command.

```
t = chebfun('t'); m = 1; n = 1;
f = sin(pi*t)./cos(pi*t);
[p, q, r] = trigratinterp(f, m, n);
```

The result of the above computation is plotted in the upper part of Figure 2.1 and as expected, we see that the interpolant  $r$  is a very good numerical approximation of the original  $f$ . In the top right of Figure 2.1, we observe two spikes in the error at  $\theta = \pm 1/2$ . These can be explained by inspecting the condition number for evaluating the function  $f$ . We know that the relative condition number  $\kappa_f$  for evaluating a scalar function  $f$  at  $\theta$  is given by  $\kappa_f(\theta) = \left| \frac{\theta f'(\theta)}{f(\theta)} \right|$ , while the

absolute condition number  $c_f$  is given by  $c_f(\theta) = |f'(\theta)|$  [51], [87]. Therefore, for  $f(\theta) = \tan \pi\theta$ ,  $\kappa_f(\theta) = \left| \frac{\pi\theta}{\sin \pi\theta \cos \pi\theta} \right|$ , and  $c_f(\theta) = |\pi \sec^2 \theta|$ , and both  $\kappa_f(\theta) \rightarrow \infty$  and  $c_f(\theta) \rightarrow \infty$  as  $\theta \rightarrow \pm 1/2$ . This explains the spikes in the error shown in the top right window of Figure 2.1.

In the next example, we pick a slightly more complicated rational function by considering

$$f(\theta) = \frac{1 + \cos 2\pi\theta + \sin 3\pi\theta}{3 + \sin 3\pi\theta + \cos 5\pi\theta}$$

on the interval  $[-1, 1]$ . We now use our algorithm to interpolate  $f$  in 17 equidistant points with a type (3, 5) trigonometric rational interpolant. Since this is the type of  $f$  itself, we again see a good global match between  $f$  and its interpolant in the bottom row of Figure 2.1.

For the above examples and others that follow, while invoking the code `trigratinterp`, we omitted the explicit specification of the points of interpolation. We can, of course, easily specify the points of interpolation by saying:

```
[p, q, r] = trigratinterp(f, m, n, th)
```

In the line above, the arbitrary points of interpolation are passed via the vector `th`. The number of points in `th` should be greater than or equal to  $2(m + n) + 1$ .

#### 2.4.2 Rational approximations of functions with singularities close to the interval of approximation

The following example and the next bring out a recurring theme in numerical analysis. While approximating an analytic function on the real axis, one can not neglect the singularities of the function in the complex plane. Of course, rational functions have an obvious advantage over polynomials for approximating functions which have poles within the interval of approximation. However, in comparison to polynomials, rational functions also do a much better job of approximating even those functions which are otherwise analytic but have singularities close to the interval of approximation [7].

To illustrate this phenomenon, in the next example we consider the function

$$f(\theta) = \frac{1}{(1.01 + \sin 3\pi\theta)},$$

on the periodic interval  $[-1, 1]$ . The function  $a + \sin(b\pi\theta)$  has infinite zeros in the complex  $\theta$ -plane but we are only interested in the ones closest to the real interval  $[-1, 1]$ . It can be shown that the points  $\theta = \frac{1}{2b} - \frac{i}{2b} \log \left( -a \pm \sqrt{a^2 - 1} \right)$  are roots of  $a + \sin(b\pi\theta)$  and these are the closest roots to the interval  $[-1, 1]$ . Using this information, we can compute the poles of  $f$  which are closest to the interval  $[-1, 1]$ . These poles occur in conjugate pairs and have imaginary parts approximately equal to  $\pm 0.0236i$ .

Let us now interpolate  $f$  in 7 equidistant points in  $[-1, 1]$  by a type  $(0, 3)$  trigonometric rational interpolant.

```
f = chebfun(@(t) 1./(1.01 + sin(3*pi*t)), 'trig')
m = 0; n = 3;
[p, q, r] = trigratinterp(f, m, n)
```

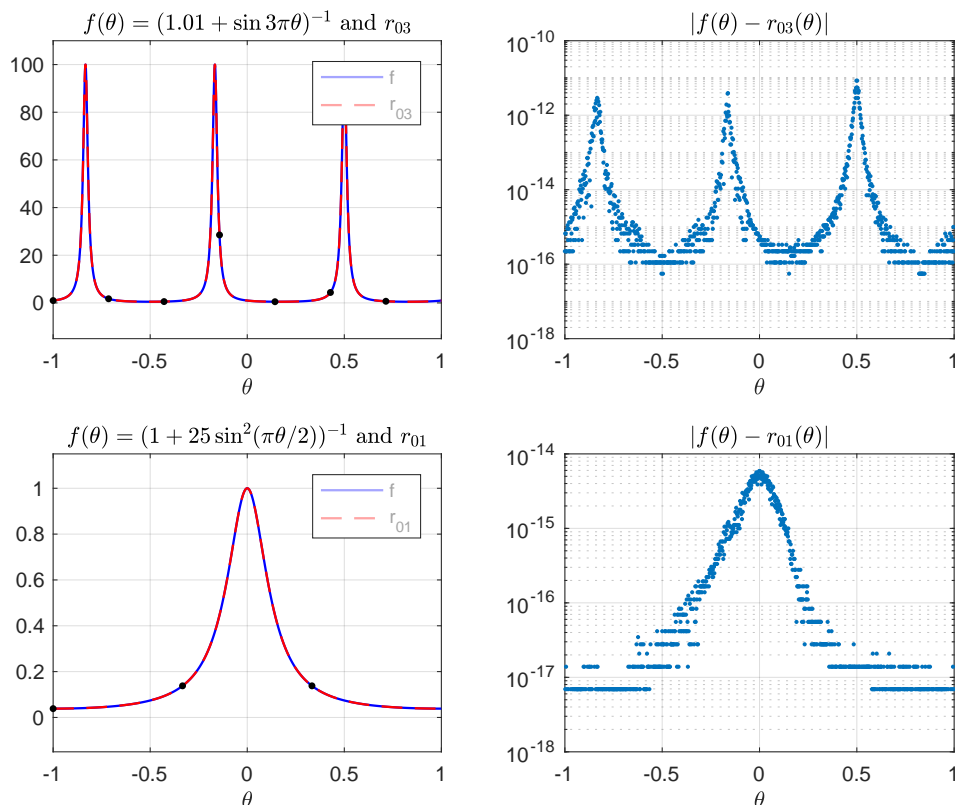


Figure 2.2: (Top row) On the top left we plot the type  $(0, 3)$  interpolant of the function  $1/(1.01 + \sin(3\pi\theta))$ . On the top right, we plot the error at 1000 equispaced points in  $[-1, 1]$ . The peaks in the error are caused by a large condition number for evaluating  $f$  at these points. (Bottom row) Interpolating the periodic Runge function. On the bottom left, we plot the trigonometric rational interpolant of  $f(\theta) = 1/(1 + 25(\sin \pi\theta/2)^2)$  and  $f$  itself. The interpolant is of type  $(0, 1)$  and interpolates  $f$  at only 3 equidistant points. These interpolated values are shown by black dots. The error at 1000 equispaced points in  $[-1, 1]$  is plotted in the bottom right window.

Since  $m$  and  $n$  are chosen to match the type of  $f$ , we can see in the top row of Figure 2.2 that the interpolant is indeed a very good numerical approximation of the original function. The periodic spikes in the error shown in the top right window can be explained using the absolute condition number for evaluating  $f$ , which is given by  $c_f(\theta) = |f'(\theta)| = \left| \frac{-3\pi \cos 3\pi\theta}{(1.01 + \sin 3\pi\theta)^2} \right|$ . The interpolant in this case is computed using function values at only 7 equidistant points. In comparison, Chebfun determines that a trigonometric polynomial interpolant at 1477 equidistant points or

a polynomial interpolant at 2249 Chebyshev points is needed to approximate  $f$  to machine precision.

In the next example, we consider the function

$$f(\theta) = \frac{1}{1 + 25(\sin \pi\theta/2)^2},$$

on the periodic interval  $[-1, 1]$ . We encountered this *periodic Runge function* earlier in Chapter 1 as well. Recall that  $f$  is analytic on the real line but has poles in the complex plane. The two poles closest to the origin are located at the points

$$\pm \frac{i}{\pi} \sinh^{-1}(1/5) \sim \pm 0.12649i.$$

Since  $f$  is a trigonometric rational function of exact type  $(0, 1)$ , we almost achieve machine precision by interpolating  $f$  in just 3 points. The results are plotted in the bottom row of Figure 2.2.

### 2.4.3 Interpolants in the complex plane

Trigonometric rational interpolation can be very effective in approximating periodic functions with poles in the complex plane. In our next example, we consider such a function. Let

$$f(\theta) = \frac{e^{\sin \pi\theta}}{1 + 25 \sin^2(\pi\theta/2)}, \quad (2.4.1)$$

and note that  $f$  has the same poles in the complex  $\theta$ -plane as the periodic Runge function discussed in the previous example. The phase portrait in Figure 2.3 shows the two poles of  $f$  closest to the origin.

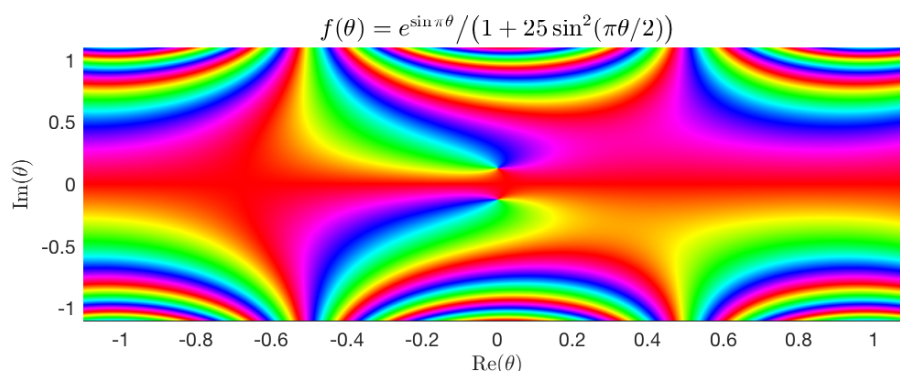


Figure 2.3: Phase portrait of the function  $f(\theta) = e^{\sin \pi\theta} (1 + 25 \sin^2(\pi\theta/2))^{-1}$ .

We now compute the following four kinds of interpolants of  $f$ :

1. algebraic-polynomial
2. trigonometric-polynomial
3. algebraic-rational
4. trigonometric-rational

To compute (1) and (2), we can say

```
f = @(t) exp(sin(pi*t))./(1 + 25*sin(pi/2*t).^2);  
p = chebfun(f);  
t = chebfun(f, 'trig');
```

Both chebfuns `p` and `t` automatically approximate  $f$  to machine precision on  $[-1, 1]$ . Their phase portraits are shown in Figure 2.4. It is known, due to a theorem by Jentzsch, that every point of the circle of convergence of a Taylor series is a limit point of zeros of the partial sums [57], [97]. In Figure 2.4 we see an analogous accumulation of zeros for a Chebyshev and a trigonometric series.

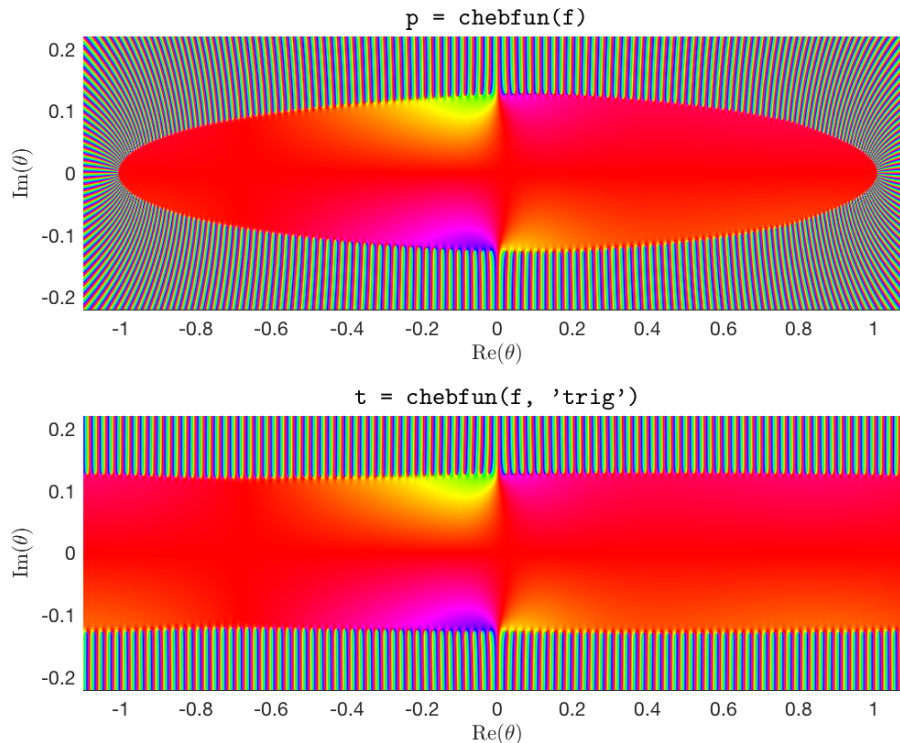


Figure 2.4: Polynomial interpolants of  $f(\theta) = e^{\sin \pi \theta} (1 + 25 \sin^2(\pi \theta / 2))^{-1}$ . *Top*: Phase portrait of  $\mathbf{p} = \text{chebfun}(f)$ . We know that  $\mathbf{p}$  is a polynomial interpolant of  $f$  based on its samples in Chebyshev points in  $[-1, 1]$ . We see an accumulation of zeros of  $\mathbf{p}$  around the Bernstein ellipse corresponding to the Chebyshev series of  $f$  [85]. *Bottom*: Phase portrait of  $\mathbf{t} = \text{chebfun}(f, \text{'trig'})$ . We know that  $\mathbf{t}$  is a trigonometric polynomial interpolant of  $f$  based on its samples in equispaced points in  $[-1, 1]$ . We see a similar accumulation of the zeros of  $\mathbf{t}$  roughly along the two horizontal lines which pass through the poles of  $f$ .

To compute the rational interpolants (3) and (4), we need to make a choice regarding the type  $(m, n)$  of the interpolants. We take  $m = 20$  and  $n = 6$  for the algebraic rational interpolant and use  $m = 10$  and  $n = 3$  for the trigonometric rational interpolant. This halving of  $m$  and  $n$  for the trigonometric case makes sure that the number of parameters in both the rational interpolants (3) and (4) are comparable. We can use the following commands in Chebfun.

```
[pm, qn, s] = ratinterp(f, 20, 6);
[pm, qn, r] = trigratinterp(f, 10, 3);
```

Note that both the interpolants  $\mathbf{s}$  and  $\mathbf{r}$  interpolate  $f$  at 27 points in  $[-1, 1]$ , the former in Chebyshev points, the latter in equidistant points. A phase portrait of the algebraic rational interpolant  $\mathbf{s}$  and the trigonometric rational interpolant  $\mathbf{r}$  is shown in Figure 2.5.

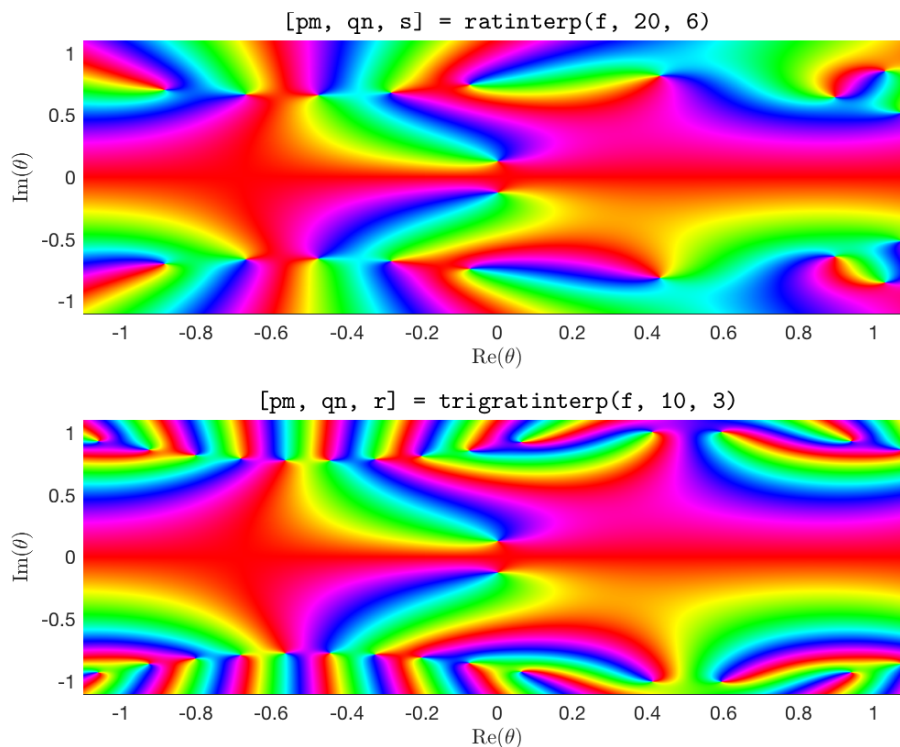


Figure 2.5: Rational interpolants of  $f(\theta) = e^{\sin \pi \theta} (1 + 25 \sin^2(\pi \theta / 2))^{-1}$ . *Top*: Phase portrait of type  $(20, 6)$  algebraic rational interpolant of  $f$ . *Bottom*: Phase portrait of type  $(10, 3)$  trigonometric rational interpolant of  $f$ . It is difficult to tell from these phase portraits whether the trigonometric rational interpolant  $\mathbf{r}$  is better than the algebraic rational interpolant  $\mathbf{s}$ . However, as we can see from the error plots contained in the bottom row of Figure 2.6,  $\mathbf{r}$  is better than  $\mathbf{s}$  in the entire region of the complex  $\theta$ -plane plotted above.

By looking at Figure 2.5, one can not tell whether the trigonometric rational interpolant  $\mathbf{r}$  is better than the algebraic rational interpolant  $\mathbf{s}$ . However,  $\mathbf{r}$  does a much better job than all the other interpolants, including  $\mathbf{s}$ . To ascertain this, in Figure 2.6, we show contour plots of the log of the absolute value of the error for each of the four interpolants  $\mathbf{p}$ ,  $\mathbf{t}$ ,  $\mathbf{s}$ , and  $\mathbf{r}$ . To compute the error in each case, we use a discrete grid of  $800 \times 800$  equispaced points in the rectangular region of the complex  $\theta$ -plane defined by the inequalities,

$$-1.1 \leq \operatorname{Re}(\theta) \leq 1.1, \quad -1 \leq \operatorname{Im}(\theta) \leq 1. \quad (2.4.2)$$

The bottom row of Figure 2.6 shows that in fact,  $\mathbf{r}$  approximates  $f$  orders of magnitude better than  $\mathbf{s}$  in the entire region plotted.

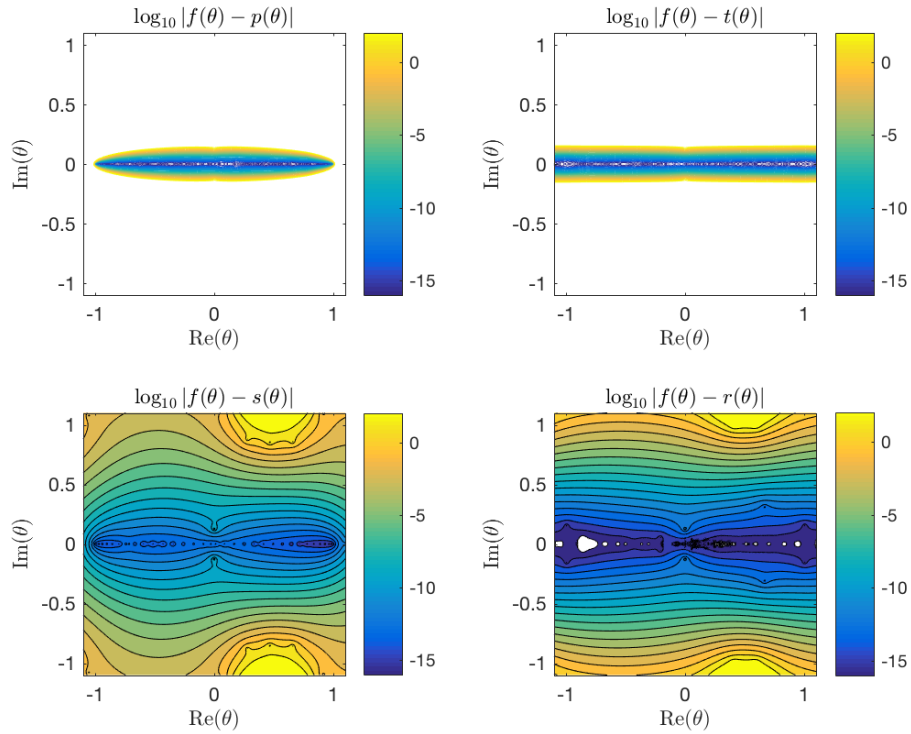


Figure 2.6: Contour plots of the log of the absolute value of the error for various interpolants of the function  $f(\theta) = e^{\sin \pi \theta} (1 + 25 \sin^2(\pi \theta/2))^{-1}$ . The error is computed on an equispaced grid of  $800 \times 800$  points in the rectangular region defined by the inequalities  $|\operatorname{Re}(\theta)| \leq 1.1$  and  $|\operatorname{Im}(\theta)| \leq 1$ . The top row contains error plots for polynomial interpolants of  $f$  whereas the bottom row plots the errors for rational interpolants. The left column contains plots of algebraic interpolants while the right column contains error plots of trigonometric interpolants. In the bottom right, we can see that the type (10, 3) trigonometric rational interpolant  $\mathbf{r}$  is orders of magnitude better than the type (20, 6) algebraic rational interpolant  $\mathbf{s}$ .

As another example, we select the 2-periodic function,

$$f(\theta) = 1/J_0(\sin \pi \theta), \quad (2.4.3)$$

where  $J_0$  is the zeroth order Bessel function of the first kind. We repeat the same process of interpolating  $f$  by the four interpolants  $\mathbf{p}$ ,  $\mathbf{t}$ ,  $\mathbf{s}$  and  $\mathbf{r}$  using Chebfun.

```
fh = @(t) 1./besselj(0, sin(pi*t));
p = chebfun(fh);
t = chebfun(fh, 'trig');
[p_m, q_n, s] = ratinterp(fh, 12, 12);
[p_m, q_n, r] = trigratinterp(fh, 6, 6);
```

The phase portrait of  $f$  is shown in Figure 2.7.

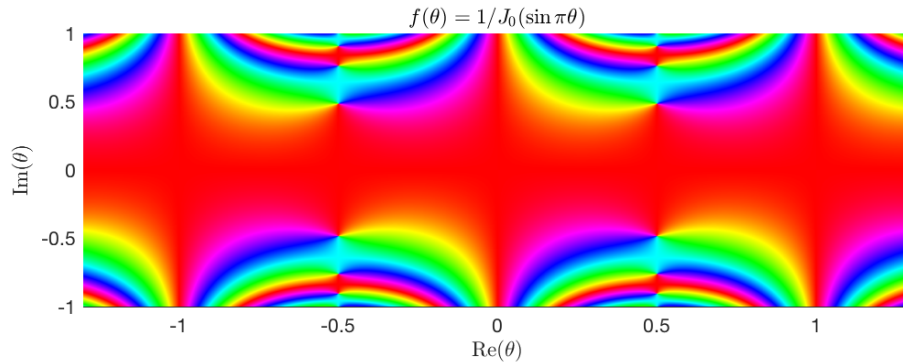


Figure 2.7: Phase portrait of the reciprocal of a periodic zeroth order Bessel function of the first kind,  $f(\theta) = 1/J_0(\sin \pi\theta)$ .

To compare the accuracy of the four interpolants, we again construct a discrete grid of  $800 \times 800$  equispaced points in a region of the complex  $\theta$ -plane defined by the inequalities,

$$-1.3 \leq \operatorname{Re}(\theta) \leq 1.3, \quad -1 \leq \operatorname{Im}(\theta) \leq 1. \quad (2.4.4)$$

The error for each of the four interpolants is plotted in Figure 2.8. Again, the trigonometric rational interpolant  $\mathbf{r}$  approximates  $f$  better than all the other interpolants.

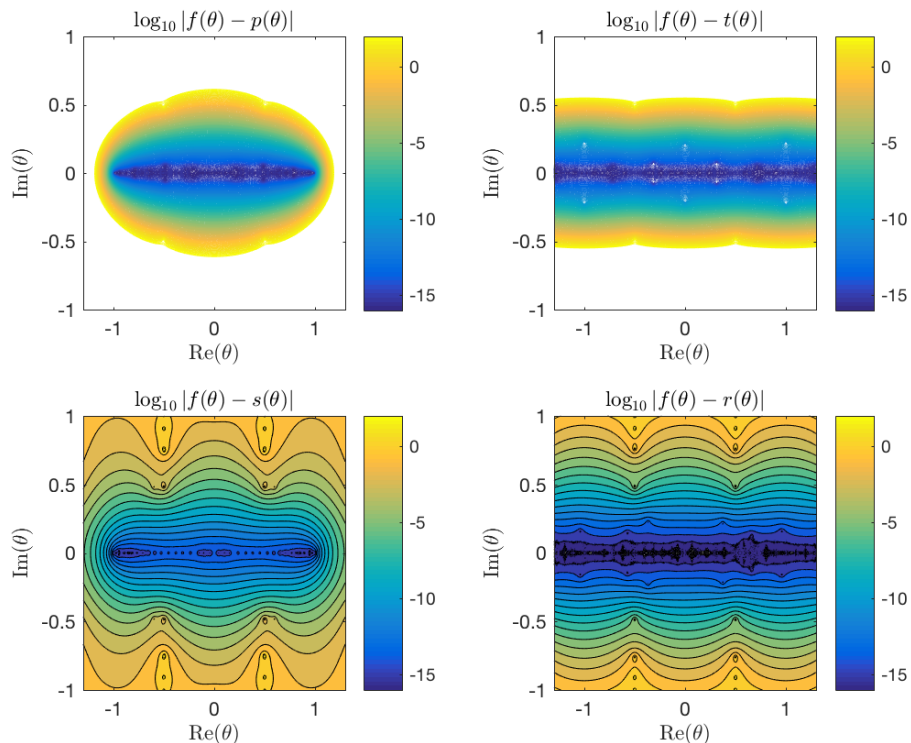


Figure 2.8: Contour-log plot of the log of the absolute value of the error for four different interpolants of the function  $f(\theta) = 1/J_0(\sin \pi\theta)$ . The error is computed on an equispaced grid of  $800 \times 800$  points in the rectangular region defined by the inequalities  $|\operatorname{Re}(\theta)| \leq 1.3$  and  $|\operatorname{Im}(\theta)| \leq 1$ . In the top left and right, we plot the error for algebraic and trigonometric polynomial interpolants of  $f$ , respectively. In the bottom row, the left column plots the error for the algebraic rational interpolant  $\mathbf{s}$  whereas the right column plots the error for the trigonometric rational interpolant  $\mathbf{r}$ . We can see that  $\mathbf{r}$  approximates  $f$  better than  $\mathbf{s}$ .

## 2.5 The problem of spurious poles

Numerical computation of rational approximations presents the challenge of spurious poles [40]. Spurious poles can occur due to two very different reasons.

1. Spurious poles can occur in exact arithmetic.
2. Spurious poles can be caused by rounding errors in floating point arithmetic.

Let us now discuss these two cases separately. The former is inherent to the mathematics of the problem and there is very little one can do to avoid these kind of spurious poles. The latter, however, is a numerical problem and as we shall see, can be avoided by using our proposed robust algorithm.

We first discuss spurious poles in exact arithmetic, followed by their discussion in floating point arithmetic.

### 2.5.1 Spurious poles in exact arithmetic

A trigonometric rational interpolant of a smooth, bounded function can have poles, even in exact arithmetic. In particular, the problem of rational interpolation is ill-posed in the sense that even a problem where the interpolant is uniquely defined, the solution may be discontinuously dependent on the data. To demonstrate the ill-posed nature of the interpolation problem, we present the following example<sup>3</sup>.

Consider the interpolation problem in which we seek a trigonometric rational interpolant  $r \in R_{11}$  on the periodic interval  $[-1, 1]$  such that the conditions

$$r(-1) = 1 = r(0), \quad r(-1/2) = 1 + \epsilon = r(1/2), \quad r(2/3) = 1 + \epsilon/2, \quad (2.5.1)$$

are satisfied. Note that  $\epsilon$  is a parameter in the conditions above.

As one can check, for any  $\epsilon$  there is a solution of the above interpolation problem in the form

$$r(\theta) = 1 + \frac{\epsilon \sin \pi \theta}{2 \sin \pi(\theta - 1/3)}. \quad (2.5.2)$$

However, the solution depends discontinuously on  $\epsilon$  in the sense that at  $\epsilon = 0$ , any change in  $\epsilon$  takes us from an interpolant with no poles, to an interpolant with poles. Specifically, for  $\epsilon = 0$  the solution is just the constant function 1. But for any non-zero  $\epsilon$ , the solution  $r$  is a rational function of exact type  $(1, 1)$  and moreover,  $r$  has poles at the points  $\theta = 1/3$  and  $\theta = -2/3$ . These poles are spurious and have nothing to do with the approximation of the underlying function. The rational interpolant  $r$  is plotted in Figure 2.9 for various values of  $\epsilon$ .

---

<sup>3</sup>This example is inspired by [85, Ch. 26], where the ill-posed nature of the algebraic rational interpolation problem is illuminated.

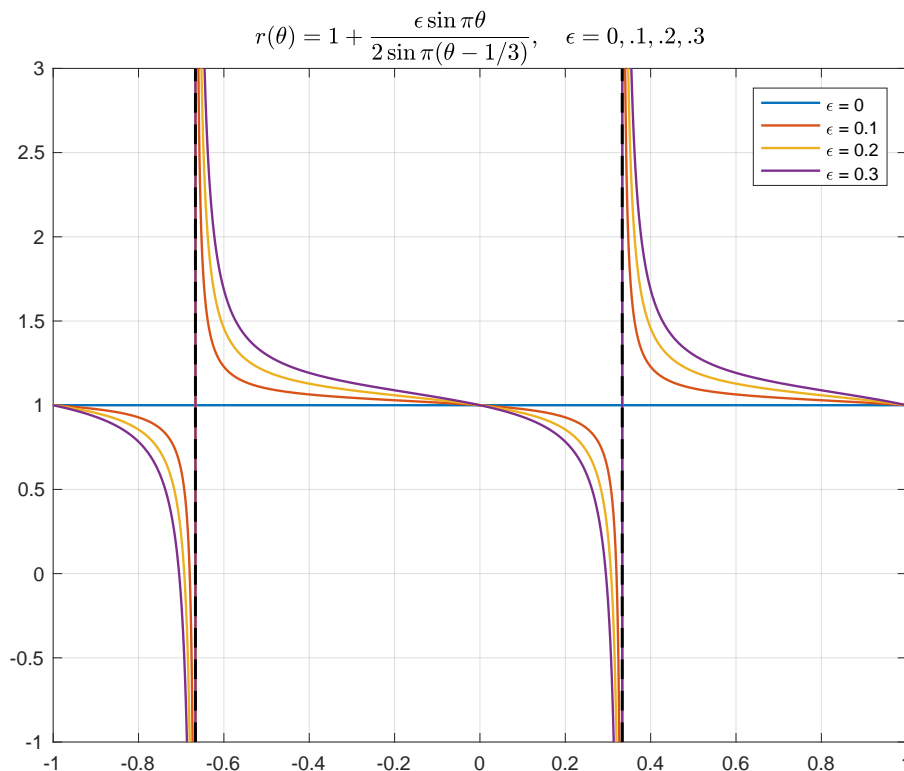


Figure 2.9: The trigonometric rational interpolant  $r(\theta) = 1 + \frac{\epsilon \sin \pi \theta}{2 \sin \pi(\theta - 1/3)}$  plotted for  $\epsilon = 0, 0.1, 0.2, 0.3$ . The function  $r$  satisfies the interpolation conditions  $r(-1) = 1 = r(0)$ ,  $r(-1/2) = 1 + \epsilon = r(1/2)$ , and  $r(2/3) = 1 + \epsilon/2$ . However,  $r$  is discontinuous with respect to the data parameter  $\epsilon$ . For  $\epsilon = 0$ ,  $r$  is the constant function 1, whereas for non-zero values of  $\epsilon$ ,  $r$  has spurious poles at  $\theta = 1/3$  and  $\theta = -2/3$ .

This example illustrates the ill-posed nature inherent to the mathematics of the problem of trigonometric rational interpolation. We will not consider such ill-posed problems in this thesis and concentrate on the second source of spurious poles, which is of a numerical nature and which we now discuss.

## 2.5.2 Spurious poles in floating point arithmetic

Spurious poles can also be caused by rounding errors that occur during the numerical computation of a rational interpolant. Henceforth, when we say spurious poles, unless otherwise stated, we shall mean spurious poles caused by rounding errors.

These spurious poles typically have very small residues. Recall that the residue of an analytic

function  $f$  at a simple pole  $\theta_0$  is given by the limit

$$\lim_{\theta \rightarrow \theta_0} (\theta - \theta_0) f(\theta) \quad (2.5.3)$$

In a sufficiently small neighbourhood of a spurious pole at  $\theta_0$ , the rational interpolant  $r$  behaves like having a zero at a point  $\hat{\theta}_0$  such that  $\hat{\theta}_0$  is numerically very close to  $\theta_0$ :  $\hat{\theta}_0 \approx \theta_0$ , and we can write the interpolant  $r$  as

$$r(\theta) = \epsilon \frac{(\theta - \hat{\theta}_0)}{(\theta - \theta_0)} + o(\theta - \theta_0), \quad (2.5.4)$$

for some small complex number  $\epsilon$  and an error term  $o(\theta - \theta_0)$ . Since a spurious pole usually has an associated zero nearby which cancels the effect of the pole further away from the location of the pole, spurious poles are also known in the literature as Froissart doublets, named after the physicist Marcel Froissart [85, Ch. 26]. As we shall see, the singular values of the matrix  $[P, -FQ]$  of Equation(2.2.26) play a central role in this discussion and we will examine these singular values in connection with the presence of the spurious poles.

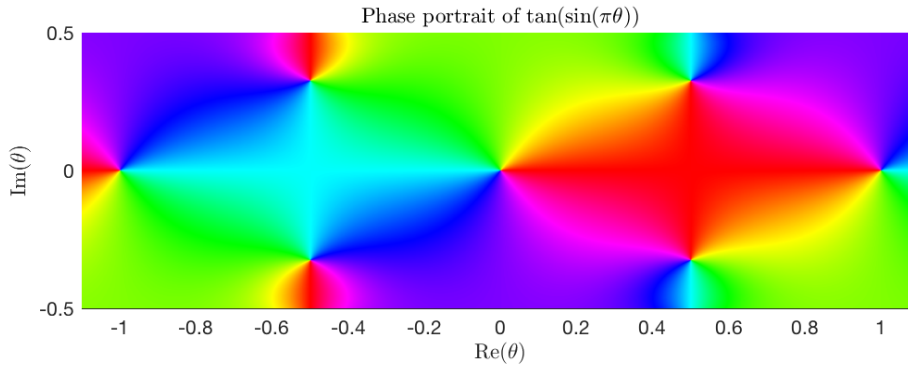


Figure 2.10: A phase portrait of the function  $f(\theta) = \tan(\sin(\pi\theta))$ . In the domain plotted, we can see three zeros on the real line and four poles symmetrically located as conjugate pairs.

To illustrate the problem of spurious poles caused by rounding errors, we take the example of the function  $f(\theta) = \tan(\sin(\pi\theta))$  on  $[-1, 1]$ . The function is analytic on the real  $\theta$ -line and meromorphic in the complex  $\theta$ -plane. In Figure 2.10, we can see the phase portrait of this function in a subset of the complex  $\theta$ -plane containing the real interval  $[-1, 1]$ . In particular, the figure shows the four poles of  $f$  closest to  $[-1, 1]$ . This suggests that a trigonometric rational interpolant with a denominator of degree 2 might be able to approximate these poles. However, we make no use of this information, and first interpolate  $f$  in 61 equispaced points in  $[-1, 1]$  by a type (15, 15) trigonometric rational function. The results are shown in Figure 2.11. We can see in the top plot that both the numerator  $p$  and the denominator  $q$  have zeros which are very close to each other.

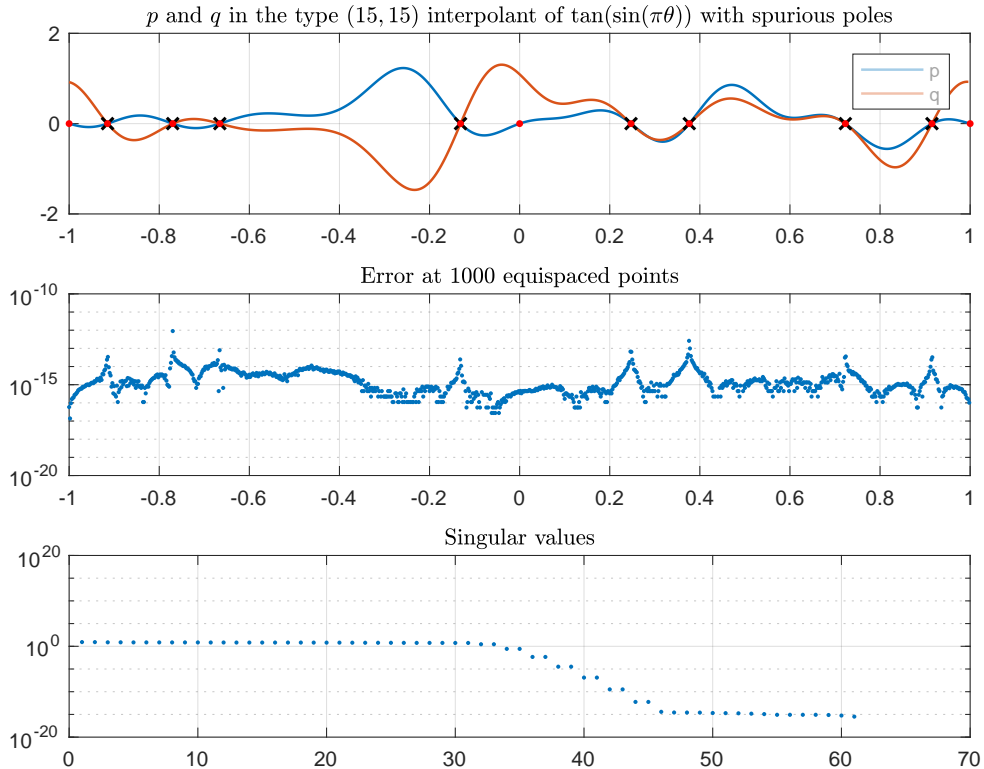


Figure 2.11: In the top row, we plot the numerator  $p$  and the denominator  $q$  of the type  $(15, 15)$  trigonometric rational interpolant of  $f(\theta) = \tan(\sin(\theta))$ . We can see the spurious poles (black crosses) and the matching zeros (red dots). In the middle row, we plot the error at 1000 equispaced points in  $[-1, 1]$ . The peaks in the error occur at the location of spurious poles where the condition number for evaluating  $p/q$  is large. In the bottom window, we plot the singular values associated with the interpolant and observe that quite a few of these singular values are close to machine precision. These small singular values are associated with spurious poles.

In fact, for this particular example, there are 8 spurious poles in  $[-1, 1]$  and using Chebfun’s `roots` command, we find that the associated zeros match with the poles to at least 14 digits of accuracy. The numerical values of the zeros of the denominator  $q$  and the numerator  $p$  are shown in left and middle columns of Table 2.1, respectively. The rightmost column of the same table shows that numerically, the zeros of  $p$  and  $q$  are almost indistinguishable.

Table 2.1 also helps in explaining the peaks of the error shown in the middle row of Figure 2.11. In a small neighbourhood of a spurious pole, there is a corresponding zero of  $p$  and this makes the condition number for evaluating  $p/q$  in the vicinity of that spurious pole large. To make this point clear, assume for the sake of simplicity that  $p(\theta) = \theta - \hat{\theta}_0$ , and  $q(\theta) = \theta - \theta_0$ , where

$\hat{\theta}_0 \approx \theta_0$ . The absolute condition number for evaluating  $r := p/q$  is given by

$$c_r(\theta) = |r'(\theta)| = \left| \frac{\hat{\theta}_0 - \theta_0}{(\theta - \theta_0)^2} \right|, \quad (2.5.5)$$

which is large as  $\theta \rightarrow \theta_0$ . One can also show that the relative condition number for evaluating  $r$  is also large as  $\theta \rightarrow \theta_0$ .

Roots of $q$ : $\theta_q$	Roots of $p$ : $\theta_p$	$ \theta_q - \theta_p $
$-9.147163564653427 \times 10^{-1}$	$-9.147163564653416 \times 10^{-1}$	$1.110223024625157 \times 10^{-15}$
$-7.701294841345929 \times 10^{-1}$	$-7.701294841345923 \times 10^{-1}$	$6.661338147750939 \times 10^{-16}$
$-6.655147102571504 \times 10^{-1}$	$-6.655147102571510 \times 10^{-1}$	$6.661338147750939 \times 10^{-16}$
$-1.311714036777694 \times 10^{-1}$	$-1.311714036777696 \times 10^{-1}$	$2.220446049250313 \times 10^{-16}$
$2.472240034197296 \times 10^{-1}$	$2.472240034197293 \times 10^{-1}$	$3.053113317719180 \times 10^{-16}$
$3.765212072098412 \times 10^{-1}$	$3.765212072098412 \times 10^{-1}$	$0.000000000000000 \times 10^{-16}$
$7.230128874625401 \times 10^{-1}$	$7.230128874625399 \times 10^{-1}$	$2.220446049250313 \times 10^{-16}$
$9.152362802626710 \times 10^{-1}$	$9.152362802626719 \times 10^{-1}$	$8.881784197001252 \times 10^{-16}$

Table 2.1: Spurious poles due to rounding errors in type (15, 15) interpolant of the function  $\tan(\sin(\pi\theta))$  on  $[-1, 1]$ . The rightmost column shows how close the zeros of the denominator  $q$  (left column) and the numerator  $p$  (middle column) are. Each number in the left column agrees with the number in the middle column to at least 14 digits of accuracy.

In the last example, we saw the occurrence of spurious poles while interpolating a function which had poles in the complex plane. However, spurious poles can also occur in interpolating functions which have no poles at all. Let us consider the entire function  $f(\theta) = e^{\sin \pi \theta}$  on  $[-1, 1]$  and interpolate it by a type (15, 15) trigonometric rational function in an equispaced grid of 61 points in  $[-1, 1)$ . The results are plotted in Figure 2.12 and in the top plot we can see 8 spurious pole-zero pairs. As a consequence, the condition number for evaluating  $p/q$  is large at these points and this results in spikes in the error shown in the middle row of Figure 2.12.

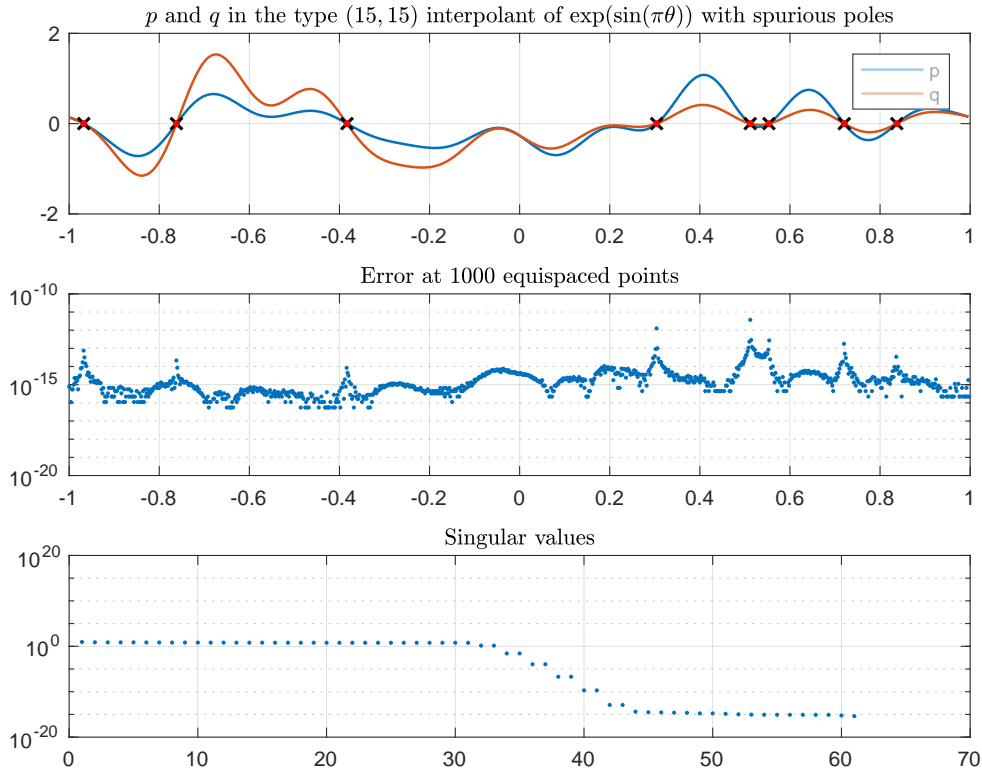


Figure 2.12: Interpolating the entire function  $f(\theta) = \exp(\sin \pi\theta)$  on  $[-1, 1]$ . In the top row, the numerator  $p$  and the denominator  $q$  of the type  $(15, 15)$  trigonometric rational interpolant are plotted. The spurious poles (black crosses) and the matching zeros (red dots) are also plotted. In the middle row, we plot the error at 1000 equispaced points in  $[-1, 1]$ . Again, the peaks of the error occur at the location of spurious poles, where the condition number for evaluating  $p/q$  is large. In the bottom row, all the singular values associated with the interpolant are plotted.

### 2.5.3 Singular values and spurious poles

In the bottom of Figures 2.11 and 2.12, we have also plotted the singular values of the matrix  $[P, -FQ]$  (see Equation (2.2.26)) associated with the rational interpolation problem in each case. In both the plots, we see that quite a few of the singular values are close to machine precision. These small singular values are responsible for spurious poles. The explanation for the appearance of spurious poles in our present case is the same as that presented in [40] for the algebraic case. When  $m$  and  $n$  are small, all the degrees of freedom in the interpolant are put to good use to achieve a good approximation, and thus the poles of the approximation tend to be placed at locations suitable for the function being approximated. As we increase  $m$  and  $n$ , or even as we increase  $m$  alone and keep  $n$  fixed, we begin to have more parameters available than needed to approximate  $f$  to machine precision. From this point onwards, the interpolant starts fitting rounding errors rather than the function values and this is when spurious poles

start to appear [40].

In the next section, we propose modifications in our original Algorithm 1 to improve this situation.

## 2.6 Robustness

To avoid spurious poles caused by rounding errors and to improve the robustness of our algorithm, following [40], we suggest the following modifications:

1. *Check the input data for symmetry:* As described in Section 2.2.6, we can reduce the problem size by a factor of 2 by taking into account even or odd symmetry in the data.
2. *Discard small trailing coefficients of  $p$  and  $q$ :* After checking for symmetry, we construct the appropriate matrix  $[P, -FQ]$ , compute the SVD and extract the coefficients of the numerator  $p$  and the denominator  $q$  as described in Algorithm 1. At this point, we discard the highest order coefficient of  $p$  if its absolute value is less than the prescribed tolerance  $\delta \geq 0$ . We then look at the next highest order coefficient of  $p$  and discard that too if its absolute value is less than  $\delta$ . We keep discarding coefficients until we find a coefficient which is greater than or equal to  $\delta$ . This step allows us to update the degree  $m$ . We then apply the same procedure on the coefficients of  $q$  and update the degree  $n$  as well.
3. *Remove contributions from numerically negligible singular values of the matrix  $[P, -FQ]$ :* The next key step of our robust algorithm is aimed at getting rid of spurious poles. In order to avoid spurious poles, our algorithm looks at the singular values of the matrix  $[P, -FQ]$  arranged in a non-increasing sequence and finds the last singular value above a given tolerance. The remaining singular values are defined as *small* and the algorithm reduces the degree of the denominator by one for every pair of small singular values. To be precise, let the singular values of  $[P, -FQ]$  be

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_J \geq 0, \quad (2.6.1)$$

and let the tolerance  $\delta \geq 0$ . Let us assume that there exists an integer  $j \geq 0$  such that

$$\sigma_j \geq \delta > \sigma_{j+1}. \quad (2.6.2)$$

We define the singular values

$$\sigma_{j+1} \geq \sigma_{j+2} \geq \dots \geq \sigma_J$$

as small singular values.

Let  $n_{ss} = J - j$  be the number of small singular values, then the degree of the denominator is updated according to the formula

$$n_{new} = n - \left\lfloor \frac{n_{ss}}{2} \right\rfloor. \quad (2.6.3)$$

The divide by 2 above is to account for the fact that 2 columns of the matrix  $[P, -FQ]$  account for a single degree of the trigonometric polynomials  $p$  or  $q$ . The matrix  $[P, -FQ]$  is now reconstructed with the number of columns adjusted according to  $n_{new}$  and the problem is solved again. The singular values of this new matrix are computed and examined again to see if there are still small singular values present. The process is repeated unless there is no pair of small singular values left.

A pseudo-code of the robust algorithm is presented in Algorithm 2 below.

---

**Algorithm 2** Robust Trigonometric Rational Interpolation and Least-Squares

---

```

1: procedure TRIGRATINTERP( $f, m, n, N, \theta_k, \delta$ )
2:   if isempty( $N$ ) then
3:      $N \leftarrow 2(m + n) + 1$  ▷ Assume interpolation if  $N$  not given
4:   if isempty( $\theta_k$ ) then
5:      $\theta_k \leftarrow \text{trigpts}(N)$  ▷ Use equispaced points if  $\theta_k$  not provided
6:    $f_k \leftarrow f(\theta_k)$  ▷ Sample  $f$  at  $\theta_k$ 
7:   while true do
8:     Construct matrices  $P, Q$  and  $F$  using the current values of  $m$  and  $n$ 
9:     Find the SVD of the matrix  $[P, -FQ]$ 
10:    Form the coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$ 
11:    Discard small trailing coefficients in  $\mathbf{a}$  and  $\mathbf{b}$ 
12:     $m \leftarrow (\text{length}(\mathbf{a}) - 1)/2$ 
13:     $n \leftarrow (\text{length}(\mathbf{b}) - 1)/2$ 
14:     $n_{ss} \leftarrow$  no. of singular values  $< \delta$ 
15:    if  $n_{ss} \geq 2$  and  $n - \left\lfloor \frac{n_{ss}}{2} \right\rfloor \geq 0$  then
16:       $n \leftarrow n - \left\lfloor \frac{n_{ss}}{2} \right\rfloor$ 
17:    else
18:      break
19:  return the coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$ 

```

---

An important point to notice is the fact that if we start with an interpolation problem, then  $N = m + n$  and the matrix  $[P, -FQ]$  has dimensions  $(2N + 1) \times (2N + 2)$ . However, in case a reduction in degrees  $m$  or  $n$  takes place, the problem becomes overdetermined and the solution is no longer an interpolant but a linearized least-squares solution. The reduction in  $m$  or  $n$  does not affect the number of rows of the matrix  $[P, -FQ]$ , but the number of columns are now reduced. The SVD computation in this case automatically computes a linearized least-squares solution with appropriate values of  $m$  and  $n$  determined by the algorithm.

The Matlab code which looks at the singular values and does the denominator degree reduction is shown below.

```

% s is the vector of singular values of [P, -F*Q]
n_big_sing_vals = find(abs(s) > tol, 1, 'last');
n_small_sing_vals = length(s) - n_big_sing_vals;
if ( fEven || fOdd )
    if ( n_small_sing_vals < 2 || ~robustness_flag )
        break
    else
        reduction = n_small_sing_vals;
    end
elseif ( n_small_sing_vals < 2 || ~robustness_flag )
    break
else
    reduction = floor(n_small_sing_vals/2);
end
n_new = n - reduction;
if ( n_new >= 0 )
    n = n_new;
end

```

Figure 2.13: Matlab code for robustness based on the magnitude of singular values

Continuing with the examples in the last section, we now revisit the problem of interpolating the function  $\tan(\sin(\pi\theta))$  by a type (15, 15) interpolant on  $[-1, 1]$ . This time, however, we use our robust algorithm and show the results obtained in Figure 2.14. In the top plot, we see that there are no spurious pole-zero pairs any more. In fact, the algorithm reduces the degree of the denominator as well as the numerator and determines that a type (13, 4) approximation results in all singular values of the associated matrix above the tolerance of  $10^{-15}$ . These singular values are plotted in the bottom row of Figure 2.14.

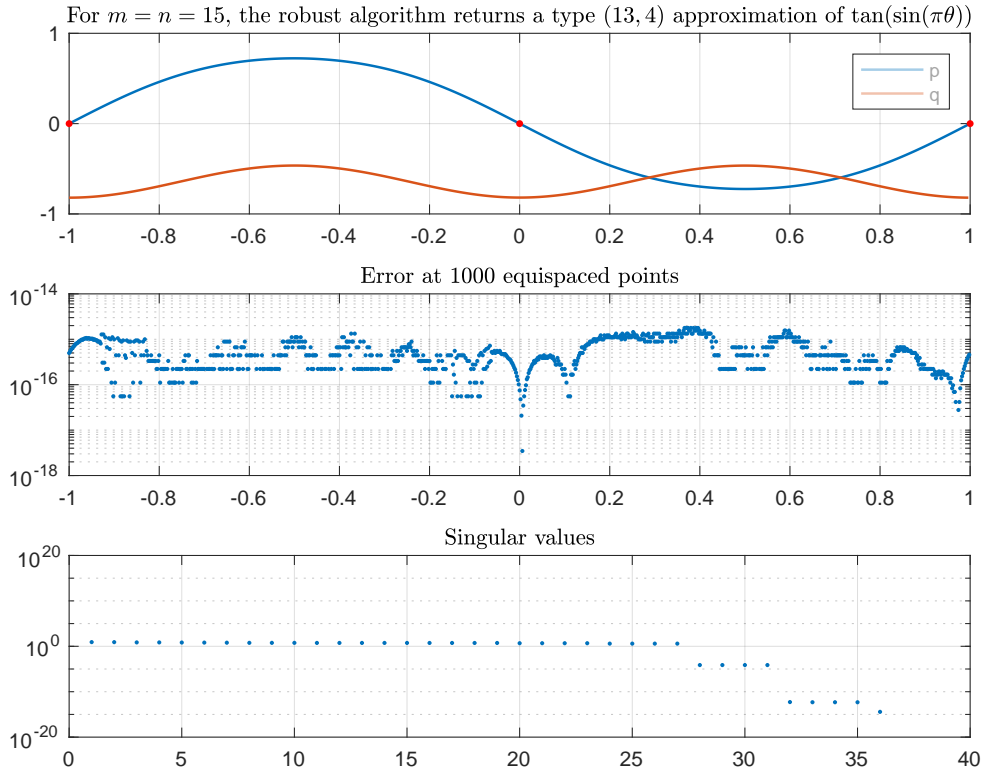


Figure 2.14: Robust approximation of  $f(\theta) = \tan(\sin(\theta))$ . In the top plot, we invoke the robust algorithm which reduces the degree of the denominator and the resulting plots of  $p$  and  $q$  show no spurious poles. In the middle row, we plot the error which does not exceed  $10^{-14}$  and in the bottom we plot the singular values, all of which are larger than the prescribed tolerance of  $10^{-15}$ . We can compare these plots with the plots in Figure 2.11, where no robustness is used.

The plots in Figure 2.15 show a similar improvement in the approximation of the function  $e^{\sin \pi\theta}$ .

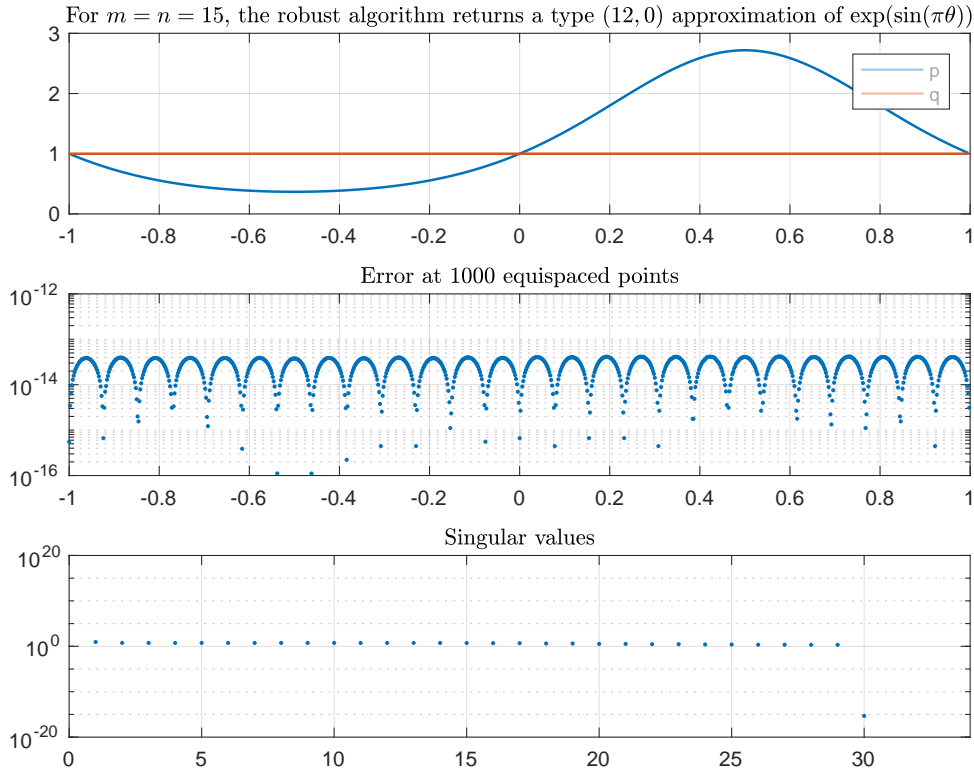


Figure 2.15: Robust trigonometric rational approximation of the entire function  $f(\theta) = e^{\sin \pi\theta}$  on  $[-1, 1]$ . In the top row, we plot the output of the robust algorithm which correctly identifies that  $f$  has no poles and automatically uses a type  $(12, 0)$  approximation. In the middle row we plot the error which does not exceed  $10^{-13}$  and in the bottom right we plot the singular values, all of which are larger than the prescribed tolerance of  $10^{-15}$ . We can compare the plots in this figure with those in Figure 2.12, where no robustness is used.

## 2.7 Trigonometric least-squares approximation

Our main focus in this chapter has been on trigonometric rational interpolation. As explained in the last section, our robust algorithm automatically computes a linearized rational least-squares approximation if a degree reduction occurs. However, we can also *start* with a least-squares problem. We only need to provide a value of  $N > m + n$ . The linear algebra of the problem is exactly the same and the final solution is computed via the SVD of the matrix  $[P, -FQ]$  of (2.2.26).

Moreover, we can also compute trigonometric polynomial least-squares approximations by setting the denominator degree  $n = 0$ . In this case the vector  $\mathbf{b}$  of (2.2.17) reduces to the number 1, while the matrix  $Q$  of (2.2.25) reduces to a column vector of all ones. As a result, the system

(2.2.26), which involves the matrix  $[P, -FQ]$  reduces to

$$\begin{bmatrix} P & -\mathbf{f} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} = 0, \quad (2.7.1)$$

where  $\mathbf{f}$  is the  $(2N + 1)$ -vector with components  $f_0, f_1, \dots, f_{2N}$ . The last equation can be re-written as

$$P\mathbf{a} = \mathbf{f}, \quad (2.7.2)$$

which is the well known matrix-vector formulation of the trigonometric polynomial least-squares problem.

## 2.8 Conclusion

In this chapter, we have presented a solution of the trigonometric rational interpolation and linearized least-squares problem. We believe that this problem has not been addressed before in the literature. We have also not been able to find software to compute such approximations. Our algorithm not only computes these trigonometric interpolants and approximants but also has the quality of being robust in the sense that it can avoid the problem of spurious poles caused by rounding errors. To achieve this robustness, the algorithm reduces the degrees of the denominator and the numerator of the approximant by inspecting the singular values of a matrix associated with the linearized problem. We have also shown numerical examples suggesting that trigonometric rational interpolants and approximants are well-suited to approximate periodic functions with poles in the complex plane. One can be quite optimistic that there are interesting applications where such trigonometric approximations are useful. Since a polynomial is a special case of a rational function, our algorithm can also be used to compute trigonometric polynomial least-squares approximations. The implementation of the algorithm is packaged with the Chebfun software system and can be downloaded with the latest copy of Chebfun from [www.chebfun.org/download](http://www.chebfun.org/download). Alternatively, the source code is freely available from [www.github.com/chebfun/chebfun/](http://www.github.com/chebfun/chebfun/).

## Chapter 3

# Trigonometric Best Approximation

### 3.1 Introduction

In this chapter<sup>1</sup>, we present numerical algorithms for finding minimax trigonometric polynomial and rational approximations of real-valued continuous periodic functions on an interval. We first discuss the case of minimax trigonometric polynomial approximation and building on that, we then extend our algorithm to the numerical computation of minimax trigonometric rational approximations. To our knowledge, there are no existing algorithms or software which compute such approximations. Inspired by the Remez algorithm [71], we will present the full details of such a numerical algorithm, together with its software implementation details.

The well known Remez algorithm is a nonlinear iterative procedure for finding minimax approximations. It is more than 80 years old and an account of its historical development can be found in [70], which focusses on the familiar case of approximation by algebraic polynomials. For an excellent implementation of the Remez algorithm for algebraic polynomial approximations, and for a new method of initialization, see the recent work of Filip [35]. As we shall see, with some variations, the same algorithm can be used to approximate continuous periodic functions by trigonometric polynomials.

After the advent of digital electronics, the Remez algorithm became popular in the 1970s due to its applications in digital filter design [71]. Since the desired filter response is typically an even periodic function of the frequency, the last 40 years have seen a sustained interest in finding the minimax approximation of even periodic functions in the space of cosine polynomials. This design problem, as shown in [71] and also in Section 1.12 of Chapter 1, after a simple change of variables, reduces to the problem of finding minimax approximations in the space of ordinary algebraic polynomials. There is a good deal of literature concerning such approximations, but

---

<sup>1</sup>Parts of this chapter have been adapted from a technical report [55] written for the Numerical Analysis (NA) Group, Mathematical Institute, Oxford, by L. N. Trefethen and the author of this thesis.

very little which deals with the general periodic case. What literature there is appears not to solve the general best approximation problem but only certain variations of it as used in digital filtering [59], [65]. In fact, we are unaware of any implementation of the Remez algorithm for this problem. This chapter presents such an implementation in Chebfun.

As mentioned in Chapter 1, Chebfun can represent periodic functions using trigonometric polynomial interpolants at equispaced points [102]. Our algorithms for computing best trigonometric polynomial and rational approximations work in the setting of this part of Chebfun. Our three major contributions are as follows.

1. *Interpolation at each step by second-kind trigonometric barycentric formula.* At each iteration of the Remez algorithm, we need to construct a trial trigonometric polynomial interpolant. Existing methods of constructing this interpolant rely on the first kind barycentric formula for trigonometric interpolation [65], [72]. We use the second kind barycentric formula, which was reviewed in Section 1.6 of Chapter 1. For a detailed discussion of the trigonometric barycentric formulae, see Berrut [16], [17], or the more recent work of Austin and Xu [8]. For a coherent presentation of various barycentric formulae, see [19].
2. *Location of maxima at each step by Chebfun eigenvalue-based root finder.* At each iteration, we are required to find the location where the maximum of absolute error in the approximation occurs. This extremal point is usually computed by evaluating the function on a dense grid and then finding the maximum on this grid [65], [71], [75]. Following [70], we compute the location of the maximum absolute error in a different way. Since the error function is periodic, we can represent it as a periodic chebfun and we can compute its derivative accurately. The roots of this derivative are then determined by the standard Chebfun algorithm, based on solving an eigenvalue problem for a colleague matrix [85, Ch. 18]. This allows us to compute the location of the maximum with considerable accuracy.
3. *High-level software framework of numerical computing with functions.* Whereas existing algorithms for best approximation produce parameters that describe an approximating function, our algorithm produces the function itself — a representation of it in Chebfun that can then be used for further computation. Thus we can calculate a best approximation in a single line of Chebfun, and further operations like evaluation, differentiation, maximization, or other transformations are available with equally simple commands. For example, our algorithm makes it possible to compute and plot the degree 10 minimax approximation of the function  $f(x) = 1/(2 + \sin 22\pi x) + (1/2) \cos 13\pi x + 5e^{-80(x-0.2)^2}$  with these commands:

```
>> x = chebfun('x');
>> f = 1./(2+sin(22*pi*x))+cos(13*pi*x)/2+5*exp(-80*(x-.2).^2);
```

```

>> tic, t = trigremez(f, 10); toc
Elapsed time is 1.968467 seconds.
>> norm(f-t, inf)
ans = 6.868203985976071e-01
>> plot(f-t)

```

Plots of the function  $f$ , its best trigonometric polynomial approximation  $t_{10}$  and the resulting error curve are shown in Figure 3.1.

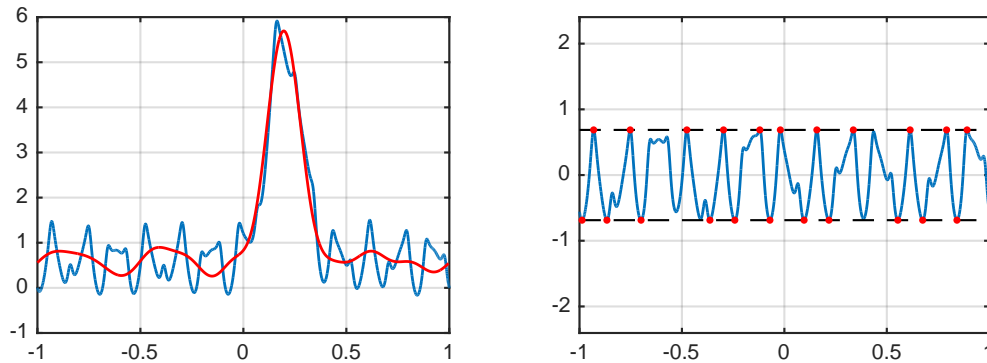


Figure 3.1: The function  $f(x) = 1/(2 + \sin 22\pi x) + (1/2) \cos 13\pi x + 5e^{-80(x-0.2)^2}$  and its best degree 10 trigonometric approximation (left). The error curve equioscillates between 22 extrema, marked by red dots (right).

The outline of the chapter is as follows. In Section 2, we define the problem and the associated vector space of trigonometric polynomials and present classical results guaranteeing the existence and uniqueness of the best trigonometric polynomial approximation. In Section 3, we present the theoretical background of the Remez algorithm for best trigonometric polynomial approximations. In Section 4 we present our algorithm and its implementation details. Section 5 consists of numerical examples. In Section 6, 7 and 8, we extend our algorithm to compute best trigonometric rational approximations, give the implementation details and present numerical examples. Finally, in the last section, we conclude the chapter while indicating possible future directions.

## 3.2 Best trigonometric polynomial approximation

Trigonometric polynomials are in general complex-valued, but in this chapter we shall only consider the real-valued case. The theory of minimax approximation of complex-valued functions

is considerably different (See, for example, [48], [59] or [84].) In this chapter, by a trigonometric polynomial of degree  $m$ , we mean a function  $t : [0, 2\pi] \rightarrow \mathbb{R}$  of the form:

$$t(\theta) = a_0 + \sum_{k=1}^m (a_k \cos k\theta + b_k \sin k\theta), \quad (3.2.1)$$

where  $a_k$  and  $b_k$  are real numbers. We continue to denote the  $2m + 1$  dimensional real vector space of all trigonometric polynomials of degree  $m$  or less by  $\mathcal{T}_m$ . We are interested in finding the best approximation in the  $\infty$ -norm,

$$\|t\|_\infty := \sup_{\theta \in [0, 2\pi]} |t(\theta)|. \quad (3.2.2)$$

Let us recall the definition of a continuous periodic function (Definition 1.1). We now formally state our problem.

**The problem:** Given a real-valued continuous periodic function  $f$  on  $[0, 2\pi]$ , and a nonnegative integer  $m$ , find a function  $t^* \in \mathcal{T}_m$  such that

$$\|f - t^*\|_\infty \leq \|f - t\|_\infty, \quad \forall t \in \mathcal{T}_m. \quad (3.2.3)$$

A best approximation exists, as can be proven easily; see for instance [25] or [73]. It is also unique. The proof of uniqueness relies on a property of the space  $\mathcal{T}_m$  which we now define.

### 3.2.1 The Haar condition

The Haar condition is a property of a finite dimensional vector space of functions. Let  $C(\Omega)$  be the space of real-valued continuous functions on a compact set  $\Omega$  and let  $V$  be a finite dimensional subspace of  $C(\Omega)$ . If  $V$  satisfies the Haar condition, then the uniqueness of the best approximation to a given  $f \in C(\Omega)$  from  $V$  is guaranteed. The existence of a best approximation is provable under less restrictive conditions [73]. However, for the uniqueness of the best approximation, the Haar condition becomes important.

There are various ways of characterizing the Haar condition; see [66] and [73], for example.

**Definition 3.1.** *Let  $V$  be an  $m$ -dimensional vector subspace of  $C(\Omega)$ . Then  $V$  satisfies the Haar condition if for every  $t \in V$ , where  $t \not\equiv 0$ , the number of distinct roots of  $t$  in  $\Omega$ , counting multiplicities, is strictly less than  $m$ .*

This is equivalent to the following [66]:

**Definition 3.2.** *Let  $V$  be an  $m$ -dimensional vector subspace of  $C(\Omega)$  and let  $\{\theta_k\}_{k=0}^{m-1}$  be any set of  $m$  distinct points in  $\Omega$ . Then  $V$  satisfies the Haar condition if there is a unique function  $t$  in  $V$  which interpolates arbitrary data  $\{f_i\}_{i=0}^{m-1}$ , i.e.  $t(\theta_i) = f_i$ ,  $i = 0, 1, \dots, m-1$ .*

Equivalently,  $V$  satisfies the Haar condition if for any basis  $\{t_i : i = 0, 1, \dots, m-1\}$  of  $V$  and any distinct set of  $m$  points in  $\Omega$ , the associated Vandermonde matrix is nonsingular.

**Theorem 3.1.** *The space  $\mathcal{T}_m$  satisfies the Haar condition.*

*Proof.* Since  $\mathcal{T}_m$  has dimension  $2m+1$ , we can prove this theorem by showing that any trigonometric polynomial  $t \in \mathcal{T}_m$ , where  $t \not\equiv 0$ , has at most  $2m$  roots in the interval  $[0, 2\pi)$ . However, this was shown earlier in Theorem 1.1 and the present theorem follows.  $\square$

It is interesting to note that the very similar vector spaces generated by the linear combinations of vectors in  $\mathcal{T}_m$  and just one of the functions  $\cos(m+1)\theta$  or  $\sin(m+1)\theta$  do not satisfy the Haar condition. For a simple example, consider the vector space generated by linear combinations of vectors in  $\mathcal{T}_0$  and say,  $\cos \theta$ , i.e. the vector space

$$\text{span}\{1, \cos \theta\}.$$

We now try to interpolate data  $\{1, 1\}$  at two points, where the first point  $\theta_1$  is arbitrary in  $(0, \pi)$  and the second point  $\theta_2$  is defined by  $\theta_2 = 2\pi - \theta_1$ . We now easily see that the constant function 1 and the function  $\cos \theta / \cos \theta_1$  both interpolate the data. These are distinct interpolants and hence the Haar condition is not satisfied. (See also the discussion in Section 1.3.2 of Chapter 1 about trigonometric polynomial interpolation in even number of points.)

Another way of proving that the Haar condition is not satisfied by this vector space is by noting that the function  $\cos \theta$ , which lies in this space, has two roots in  $[0, 2\pi)$ , while the dimension of the vector space itself is also 2.

Since  $\mathcal{T}_m$  satisfies the Haar condition, we can now assert the uniqueness of the best trigonometric polynomial approximation of a real-valued continuous periodic function. For the sake of completeness, we state this formally in the following theorem.

**Theorem 3.2.** *Let  $f$  be a real-valued continuous periodic function on  $[0, 2\pi]$ . The best degree- $m$  approximation of  $f$  out of  $\mathcal{T}_m$  is unique.*

*Proof.* See [25].  $\square$

### 3.3 The Remez algorithm for trigonometric polynomial approximation: Theoretical background

The Remez algorithm, also called the exchange algorithm, is a nonlinear iterative procedure which, in its usual setting, calculates the best approximation of a given function  $f \in C([a, b])$  from the vector space  $\mathcal{P}_m$  of algebraic polynomials of degree  $m$  or less. For a given  $f$ , starting

from an arbitrary initial condition, the algorithm converges to the best approximation of  $f$  in  $\mathcal{P}_m$ , and a proof of its convergence can be found in [73]. In the following sections we will see how the same algorithm, with certain modifications, can be used to find best approximations of continuous periodic functions in  $\mathcal{T}_m$ .

Let us now look at the details of the algorithm. We shall closely follow [73], adapting its discussion of algebraic polynomials to our problem of trigonometric polynomials.

Let  $t^*$  be the best approximation of a real-valued continuous periodic function  $f$  on  $[0, 2\pi]$  from the space  $\mathcal{T}_m$ . Let  $t_k$  be the  $k^{\text{th}}$  trial approximation in the iterative sequence and define the corresponding error at the  $k^{\text{th}}$  iteration as

$$e_k(\theta) = f(\theta) - t_k(\theta), \quad 0 \leq \theta < 2\pi. \quad (3.3.1)$$

Since we are approximating in the  $\infty$ -norm, the set of points at which  $e_k$  takes its extreme values is of interest. Let us denote this set by  $E_k$ ,

$$E_k = \{\theta \in [0, 2\pi) : |e_k(\theta)| = \|e_k(\theta)\|_\infty\}. \quad (3.3.2)$$

### 3.3.1 A sufficient condition for optimality

We suppose that at the  $k^{\text{th}}$  step,  $t_k$  is not the best approximation. We can then write the best approximation  $t^*$  as

$$t^*(\theta) = t_k(\theta) + \alpha t(\theta), \quad (3.3.3)$$

for some trigonometric polynomial  $t \in \mathcal{T}_m$ , such that  $\|t\|_\infty = 1$  and, without loss of generality, some real  $\alpha > 0$ . With this notation we have

$$e^* = f - t^* = f - t_k - \alpha t = e_k - \alpha t. \quad (3.3.4)$$

By definition of the best approximation, for  $\theta \in E_k$ ,  $e^*$  is strictly less than  $e_k$  in absolute value:

$$|e^*(\theta)| = |e_k(\theta) - \alpha t(\theta)| < |e_k(\theta)|, \quad \theta \in E_k. \quad (3.3.5)$$

Since  $\alpha > 0$ , this implies that for every  $\theta \in E_k$ ,  $t(\theta)$  has the same sign as  $e_k(\theta)$ . Thus, if  $t_k$  is not the best approximation, there is a function  $t \in \mathcal{T}_m$  which has the same sign as  $e_k$  at the points in  $E_k$ , i.e.

$$[f(\theta) - t_k(\theta)]t(\theta) > 0, \quad \theta \in E_k. \quad (3.3.6)$$

In other words, if there is no function  $t \in \mathcal{T}_m$  for which (3.3.6) holds, then  $t_k = t^*$ .

### 3.3.2 A necessary condition for optimality

We now prove the converse of the above statement: if there exists a trigonometric polynomial  $t \in \mathcal{T}_m$  for which the condition (3.3.6) holds, then  $t_k$  is not the best approximation, i.e. there exists a positive value of  $\alpha$  such that

$$\|f - (t_k + \alpha t)\|_\infty < \|f - t_k\|_\infty. \quad (3.3.7)$$

We again closely follow [73]. Let  $E$  denote a closed subset of the interval  $[0, 2\pi)$ . Also, let  $\alpha > 0$  and without loss of generality, assume that  $t \in \mathcal{T}_m$  satisfies the bound

$$|t(\theta)| \leq 1, \quad 0 \leq \theta < 2\pi. \quad (3.3.8)$$

We define  $E_o \subset E$  as the set of points in  $E$  for which  $e_k(\theta)$  and  $t(\theta)$  have opposite signs:

$$E_o = \{\theta \in E : t(\theta)e_k(\theta) \leq 0\}. \quad (3.3.9)$$

Since  $E_o$  is closed, any continuous function attains its bounds on  $E_o$ , and we can therefore define the number  $d$  as

$$d := \max_{\theta \in E_o} |e_k(\theta)|. \quad (3.3.10)$$

In case  $E_o$  is empty, we define  $d = 0$ . Since  $E_o \cap E_k$  is empty, we have

$$d < \max_{\theta \in E} |e_k(\theta)|. \quad (3.3.11)$$

We now prove that the inequality

$$\max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)| < \max_{\theta \in E} |f(\theta) - t_k(\theta)| \quad (3.3.12)$$

holds for

$$\alpha = \frac{1}{2} \left[ \max_{\theta \in E} |e_k(\theta)| - d \right]. \quad (3.3.13)$$

Since  $E$  is closed, we may let  $\gamma$  be an element of  $E$  such that

$$|f(\gamma) - t_k(\gamma) - \alpha t(\gamma)| = \max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)|. \quad (3.3.14)$$

If  $\gamma \in E_o$ , we have

$$\max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)| = |f(\gamma) - t_k(\gamma)| + |\alpha t(\gamma)| \leq d + \alpha < \max_{\theta \in E} |e_k(\theta)|,$$

where the last bound follows by inserting the value of  $d + \alpha$  from (3.3.13).

On the other hand if  $\gamma \notin E_o$  then the signs of  $e_k(\gamma)$  and  $t(\gamma)$  are the same, and we get the strict inequalities

$$\max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)| < \max \{|e_k(\gamma)|, |\alpha t(\gamma)|\} < \max_{\theta \in E} |e_k(\theta)|, \quad (3.3.15)$$

where in the last inequality, we have again used the definition of  $\alpha$  from (3.3.13) and the fact that  $t$  is bounded by 1. Therefore, we have shown that (3.3.12) holds.

We summarize the last two conditions as the following theorem.

**Theorem 3.3.** *Let  $E$  be a closed subset of  $[0, 2\pi)$ ,  $t_k$  be an element of  $\mathcal{T}_m$ , and  $E_k$  be the subset of  $E$  where the error  $e_k(\theta) = f(\theta) - t_k(\theta)$  takes its maximum absolute value. Then  $t_k$  is the function in  $\mathcal{T}_m$  that minimizes the expression*

$$\max_{\theta \in E} |f(\theta) - t(\theta)|, \quad t \in \mathcal{T}_m, \quad (3.3.16)$$

if and only if there is no function  $t \in \mathcal{T}_m$  such that

$$[f(\theta) - t_k(\theta)]t(\theta) > 0, \quad \theta \in E_k. \quad (3.3.17)$$

*Proof.* In the discussion above. □

### 3.3.3 Characterization of the best trigonometric polynomial approximation

Theorem 3.3 tells us that in order to find out if a trial approximation  $t_k$  is the best approximation or not, one only needs to consider the extreme values of the error function  $e_k(\theta) = f(\theta) - t_k(\theta)$ . Specifically, one should ask if the condition (3.3.6) holds for some function  $t \in \mathcal{T}_m$ . This allows us to characterize the best approximation by the sign changes of the error function. Since we are working in the space of trigonometric polynomials, condition (3.3.6) is rather easy to test. We make use of the fact that any function in  $\mathcal{T}_m$  has at most  $2m$  sign changes<sup>2</sup> in  $[0, 2\pi)$ . Therefore, if the error function changes sign more than  $2m$  times as  $\theta$  ranges in  $E_k$ , then  $t_k$  is the best approximation. Conversely, if the number of sign changes does not exceed  $2m$ , then we can choose the zeros of a trigonometric polynomial to construct  $t$  in  $\mathcal{T}_m$  so that the condition (3.3.6) is satisfied. This result is usually called the minimax characterization theorem [73] or the equioscillation theorem, which we state below.

**Theorem 3.4.** (*Equioscillation*) *Let  $f$  be a real-valued continuous periodic function on  $[0, 2\pi]$ . Then  $t^*$  is the best approximation of  $f$  from  $\mathcal{T}_m$  if and only if there exist  $2m + 2$  points  $\{\theta_i, i = 0, 1, \dots, 2m + 1\}$  such that the following conditions hold:*

$$0 \leq \theta_0 < \theta_1 < \dots < \theta_{2m+1} < 2\pi, \quad (3.3.18)$$

$$|f(\theta_i) - t^*(\theta_i)| = \|f - t^*\|_\infty, \quad i = 0, 1, \dots, 2m + 1, \quad (3.3.19)$$

---

<sup>2</sup>For example the function  $\sin \theta$  has exactly 2 sign changes in  $[0, 2\pi)$ , one at  $\theta = 0$  and another at  $\theta = \pi$ .

and

$$f(\theta_{i+1}) - t^*(\theta_{i+1}) = -[f(\theta_i) - t^*(\theta_i)], \quad i = 0, 1, \dots, 2m. \quad (3.3.20)$$

*Proof.* See [25] or [73]. □

A discrete version of the above theorem also holds, which we state below as a separate theorem. As we will see, this is one of the key results used by the Remez algorithm.

**Theorem 3.5.** *Let  $f$  be a continuous periodic function on  $[0, 2\pi]$  and let  $\{\theta_i\}$  be  $2m + 2$  points satisfying  $0 \leq \theta_0 < \theta_1 < \dots < \theta_{2m+1} < 2\pi$ . Then  $t^* \in \mathcal{T}_m$  minimizes the expression*

$$\max_i |f(\theta_i) - t(\theta_i)|, \quad t \in \mathcal{T}_m, \quad (3.3.21)$$

*if and only if*

$$f(\theta_{i+1}) - t^*(\theta_{i+1}) = -[f(\theta_i) - t^*(\theta_i)], \quad i = 0, 1, \dots, 2m. \quad (3.3.22)$$

*Proof.* See [73]. □

**Theorem 3.6.** *Let the conditions of Theorem 3.4 hold and let  $t^*$  be any element of  $\mathcal{T}_m$  such that the conditions*

$$\text{sign}[f(\theta_{i+1}) - t^*(\theta_{i+1})] = -\text{sign}[f(\theta_i) - t^*(\theta_i)], \quad i = 0, 1, \dots, 2m, \quad (3.3.23)$$

*are satisfied. Then the inequalities*

$$\min_i |f(\theta_i) - t^*(\theta_i)| \leq \min_{t \in \mathcal{T}_m} \max_i |f(\theta_i) - t(\theta_i)| \leq \min_{t \in \mathcal{T}_m} \|f - t\|_\infty \leq \|f - t^*\|_\infty \quad (3.3.24)$$

*hold. Also, the first inequality is strict unless all the numbers  $\{|f(\theta_i) - t^*(\theta_i)|, i = 0, 1, \dots, 2m + 1\}$  are equal.*

*Proof.* See [73]. □

The next theorem is contained in Theorem 3.6. However, to clearly establish a link between the sign alternation of the error function  $e_k$  on a discrete set and the error in the best approximation  $e^*$ , we state it separately.

**Theorem 3.7.** (*de la Vallée Poussin*) Let  $t_k \in \mathcal{T}_m$  and  $0 \leq \theta_0 < \theta_1 < \dots < \theta_{2m+1} < 2\pi$  be  $2m + 2$  points such that the sign of the error  $f(\theta_i) - t_k(\theta_i)$  alternates as  $i$  varies from 0 to  $2m + 1$ . Then, for every  $t \in \mathcal{T}_m$ ,

$$\min_i |f(\theta_i) - t_k(\theta_i)| \leq \max_i |f(\theta_i) - t(\theta_i)| \quad (3.3.25)$$

Also, the inequality is strict unless all the numbers  $\{|f(\theta_i) - t_k(\theta_i)|, i = 0, 1, \dots, 2m + 1\}$  are equal.

The above theorem tells us that if we construct  $t_k \in \mathcal{T}_m$  such that the error  $e_k$  alternates its sign as above, then a lower bound for  $e^*$  can be easily obtained: let  $t = t^*$  in the theorem above and we get

$$\min_i |f(\theta_i) - t_k(\theta_i)| \leq \max_i |f(\theta_i) - t^*(\theta_i)| \leq \|f - t^*\|_\infty \leq \|f - t_k\|_\infty. \quad (3.3.26)$$

This also gives us a way of bounding  $\|f - t_k\|_\infty$  by  $\|f - t^*\|_\infty$ . For it follows from the inequalities above that

$$1 \leq \frac{\|f - t^*\|_\infty}{\min_i |f(\theta_i) - t_k(\theta_i)|},$$

and now multiplying both sides of the inequality above by  $\|f - t_k\|_\infty$ , we get the bound

$$\|f - t_k\|_\infty \leq \left( \frac{\|f - t_k\|_\infty}{\min_i |f(\theta_i) - t_k(\theta_i)|} \right) \|f - t^*\|_\infty. \quad (3.3.27)$$

### 3.4 The Remez (exchange) algorithm for trigonometric polynomial approximation: Implementation details

The Remez algorithm, at each iteration, finds a set of points  $R_k = \{\theta_i, i = 0, 1, \dots, 2m + 1\}$  and a trigonometric polynomial  $t_k \in \mathcal{T}_m$  such that the conditions of Theorem 3.5 are satisfied. We will call the set  $R_k$  the  $k^{\text{th}}$  reference.

To start the algorithm, an initial reference  $R_0$  is chosen. This can be any set of points that satisfies the conditions

$$0 \leq \theta_0 < \theta_1 < \dots < \theta_{2m+1} < 2\pi. \quad (3.4.1)$$

The exchange algorithm is guaranteed to converge from any starting reference [73]. However, the number of iterations taken to achieve a certain accuracy is greatly affected by the choice of the initial reference. For the present case of trigonometric polynomials, we start by choosing  $2m + 2$  equally spaced points in  $[0, 2\pi)$ .

### 3.4.1 Computation of the trial polynomial using trigonometric interpolation

At the start of each iteration, a new reference is available which is different from the references of all the previous iterations. Given a reference  $R_k$ , first the trigonometric polynomial  $t_k$  is constructed that minimizes the expression

$$\max_{i=0,1,\dots,2m+1} |f(\theta_i) - t_k(\theta_i)|, \quad t_k \in \mathcal{T}_m. \quad (3.4.2)$$

Theorem 3.5 shows that this can be done by solving the linear system

$$f(\theta_i) - t_k(\theta_i) = (-1)^{i-1} h_k, \quad i = 0, 1, \dots, 2m + 1, \quad (3.4.3)$$

which also defines the unknown *levelled reference error*  $h_k$ .

However, instead of solving the linear system, we convert this problem to an interpolation problem. First, note that there are  $2m + 2$  unknown parameters: the unknown  $h_k$  and the  $2m + 1$  unknown coefficients of the trigonometric polynomial  $t_k$ . It turns out that the levelled reference error  $h_k$  can be found explicitly without solving the linear system (see Appendix A). Once this is done, we can re-write (3.4.3) as an interpolation problem:

$$t_k(\theta_i) = f(\theta_i) + (-1)^{i-1} h_k, \quad i = 0, 1, \dots, 2m + 1, \quad (3.4.4)$$

At a first look, it seems that this problem is overdetermined: it has  $2m + 2$  data points to interpolate and only  $2m + 1$  degrees of freedom in the coefficients of  $t_k$ . However, since  $h_k$  has been found as a part of the original problem, we can discard any one  $\theta_i$  from (3.4.4) and find the unique interpolant to the remaining data. The result is an interpolant that automatically interpolates the data at the discarded node.

This interpolation problem is solved using the type II trigonometric barycentric formula (see Equation (1.6.14) of Chapter 1 or [16], [19]):

$$t_k(\theta) = \sum_{i=0}^{2m} \frac{w_i t_k(\theta_i)}{\sin \frac{1}{2}(\theta - \theta_i)} \bigg/ \sum_{i=0}^{2m} \frac{w_i}{\sin \frac{1}{2}(\theta - \theta_i)}, \quad (3.4.5)$$

where

$$w_i^{-1} = \prod_{\substack{j=0 \\ j \neq i}}^{2m} \sin \frac{1}{2}(\theta_i - \theta_j), \quad i = 0, 1, \dots, 2m. \quad (3.4.6)$$

The formula (3.4.5) is implemented in Chebfun [102] and thus we can compute the interpolant  $t_k$  very easily. In fact, the complete code for constructing  $t_k$  is as follows:

```

dom = [0, 2*pi];
w = trigBaryWeights(xk);    % xk is the kth reference
sigma = ones(2*m+2, 1);
sigma(2:2:end) = -1;
hk = (w'*fk) / (w'*sigma); % Levelled reference error
uk = (fk - hk*sigma);      % Values to be interpolated
tk = chebfun(@(t) trigBary(t, uk, xk, dom), dom, 2*m+1, 'trig');

```

The real computational work in the short code above is done by the functions `trigBary` and `trigBaryWeights`<sup>3</sup>.

### 3.4.2 Finding the new reference

The next step of the algorithm is to find a new reference  $R_{k+1}$ . From Theorem 3.6, we see that the following bound holds:

$$|h_k| \leq \|f - t^*\|_\infty \leq \|f - t_k\|_\infty. \quad (3.4.7)$$

By increasing  $|h_k|$ , we get closer and closer to the best approximation. The exchange algorithm tries to choose the new reference so as to maximize the magnitude of the levelled reference error  $h_k$  at each iteration. One might think then that a classical optimization algorithm should be used to solve this problem. However, according to [73], the structure of  $h_k$  is such that the standard algorithms of optimization are not efficient.

We now look at ways of increasing the levelled reference error so that  $|h_{k+1}| > |h_k|$ . Let the new reference be  $R_{k+1} = \{\zeta_i, i = 0, 1, \dots, 2m+1\}$ , and let the new trial approximation be  $t_{k+1}$ . To ensure  $|h_{k+1}| > |h_k|$ , Theorem 3.6 tells us that the new reference must be chosen such that the old trial polynomial  $t_k$  oscillates on it (not necessarily equally) with a magnitude greater than or equal to  $|h_k|$ :

$$\text{sign}[f(\zeta_{i+1}) - t_k(\zeta_{i+1})] = -\text{sign}[f(\zeta_i) - t_k(\zeta_i)], \quad i = 0, 1, \dots, 2m, \quad (3.4.8)$$

and

$$|h_k| \leq |f(\zeta_i) - t_k(\zeta_i)|, \quad i = 0, 1, \dots, 2m+1, \quad (3.4.9)$$

where at least one of the inequalities is strict. If the above conditions are satisfied, then it follows from Theorem 3.7 that

$$|h_k| \leq \min_i |f(\zeta_i) - t_k(\zeta_i)| < \max_i |f(\zeta_i) - t_{k+1}(\zeta_i)| = |h_{k+1}|. \quad (3.4.10)$$

---

<sup>3</sup>The first Chebfun versions of the functions `trigBary` and `trigBaryWeights` were written by the author of this thesis.

This gives us a number of possibilities for choosing the new reference  $R_{k+1}$ . In the so called one-point exchange algorithm, it is obtained by exchanging a single point of  $R_k$  with the global maximum of the error function  $e_k$  in such a way that the oscillation of the error (3.4.8) is maintained. The other extreme is to define all points of the new reference as local extrema of the error  $e_k$  in such a way that conditions (3.4.8) and (3.4.9) are satisfied. Methods that can change every reference point at every iteration are usually more efficient than the one-point exchange algorithm in the sense that fewer iterations are required for convergence. In this chapter, we will not discuss the convergence properties of exchange algorithms; see [73].

Regardless of what exchange strategy is chosen, one needs to compute local or global extrema of the error function  $e_k = f - t_k$ . This is where our algorithm, following [70], uses Chebfun's root finder. This algorithm has three steps.

1. *Differentiate the error function.* Since the functions  $f$  and  $t_k$  are periodic, the error function  $e_k$  is also periodic. We can therefore compute the derivative  $e'_k$  in a numerically stable way by using the Fourier expansion coefficients of  $e_k$ . This is automated in Chebfun via the overloaded `diff` command.

```
ekp = diff(f-tk);
```

In case the error function  $e_k$  is non-differentiable on a finite number of points, Chebfun automatically finds these points and differentiates the error function piece-wise [69].

2. *Expansion in Chebyshev basis.* To find the roots of a periodic chebfun, one can solve a companion matrix eigenvalue problem. However, the straightforward implementation of this process will lead to an  $O(m^3)$  algorithm [102]. To use Chebfun's  $O(m^2)$  recursive interval subdivision strategy root finding algorithm, the function is first converted to a non-periodic chebfun. This can be done by issuing the command:

```
g = chebfun(ekp);
```

To be precise, the above line of code computes a Chebyshev interpolant of the function `ekp`, which is as good as a Chebyshev series expansion up to machine precision [85]. As an alternative to the subdivision strategy, one can also use the fast companion matrix eigensolver developed by Aurentz and others [3], [5] for root finding. We have not investigated this possibility.

3. *Find the roots.* Finally, we use the Chebyshev coefficients obtained in step 2 to compute the roots:

```
r = roots(g);
```

The above line computes roots as eigenvalues of a colleague matrix [15], [85].

The above three steps were spelled out to give algorithmic details. However, in Chebfun, they are all combined in a single line of code:

```
r = minandmax(f-tk, 'local');
```

Once the roots are computed, we can find the new reference  $R_{k+1}$  using either the one-point exchange or the multiple point exchange strategy.

Our algorithm uses the multiple point exchange strategy. Let  $W_k$  be the set of local extrema of  $e_k$ . We form the set  $S_k = \{\zeta : \zeta \in W_k \cup R_k, |e_k(\zeta)| \geq |h_k|\}$ . We then order the points in  $S_k$ , and for each subset of consecutive points with the same sign, we keep only one for which  $|e_k|$  is maximum. From this collection of points,  $R_{k+1}$  is formed by selecting  $2m + 2$  consecutive points that include the global maximum of  $e_k$ .

This completes one iteration of the exchange algorithm. The algorithm terminates when  $\|e_{k+1}\|_\infty$  is within a prescribed tolerance of  $|h_k|$ .

### 3.5 Numerical examples of best trigonometric polynomial approximations

Let us now look at some examples. The Chebfun command `trigremez(f, m)` finds the degree  $m$  best trigonometric polynomial approximation of a periodic continuous chebfun  $f$  on its underlying domain:

```
t = trigremez(f, m);
```

Here is a basic example for a function on the default interval  $[-1, 1]$ :

```
f = chebfun(@(x) exp(sin(8*pi*x)) + 5*exp(-40*x.^2), 'trig' )
t = trigremez(f, 4);
```

The plots can be seen in Figure 3.2.

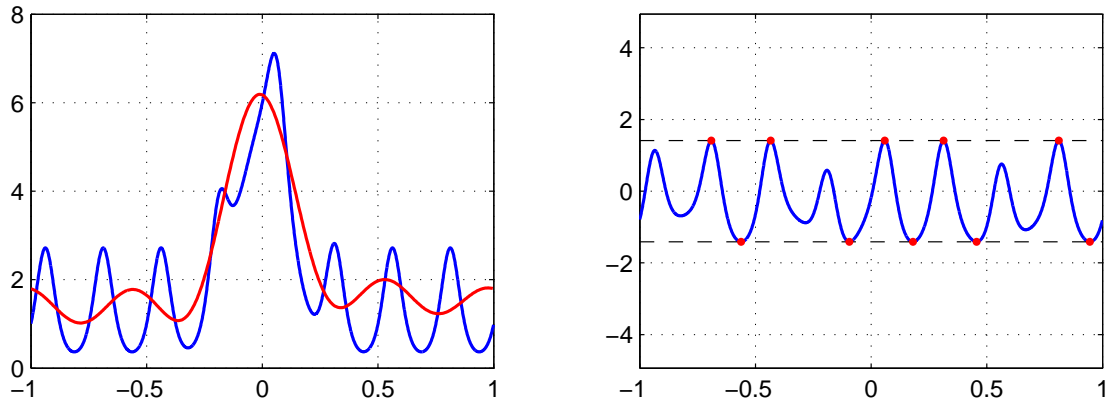


Figure 3.2: The function  $f(x) = e^{\sin 8\pi x} + 5e^{-40x^2}$ , its degree 4 best approximation (left) and the equioscillating error curve (right).

An interesting feature of `trigremez` is that we can also use it with `chebfuns` which are represented via their Chebyshev polynomial interpolants. As long as a function is continuous periodic, that is, the function values are the same at the endpoints, the theorems stated in the previous sections hold and it does not matter whether we represent the function as a Chebyshev or a trigonometric expansion.

Here is a modified version of an example of a zig-zag function taken from [85]. To make the function continuous across the periodic boundary, we have subtracted off an appropriate linear term:

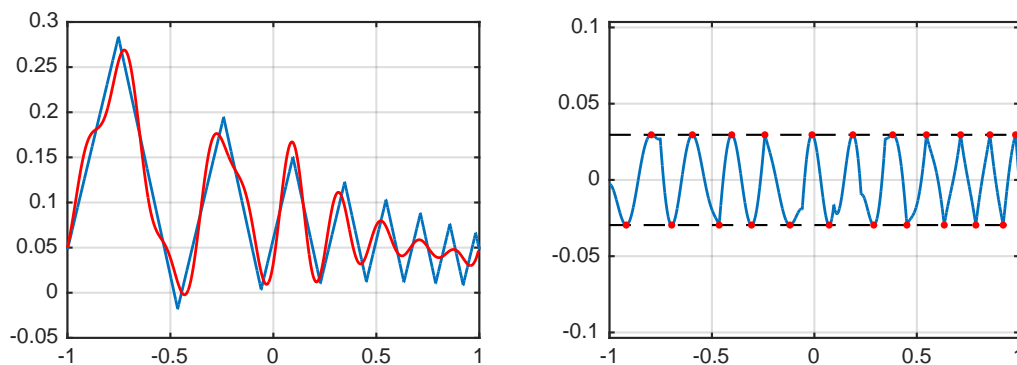


Figure 3.3: Degree 10 best approximation of a zig-zag function and the corresponding error curve.

```
x = chebfun('x');
```

```

g = cumsum(sign(sin(20*exp(x))));
m = (g(1)-g(-1))/2;
f = g - m*x;
t = trigremez(f, 10);

```

We can see the approximation and the error curve in Figure 3.3.

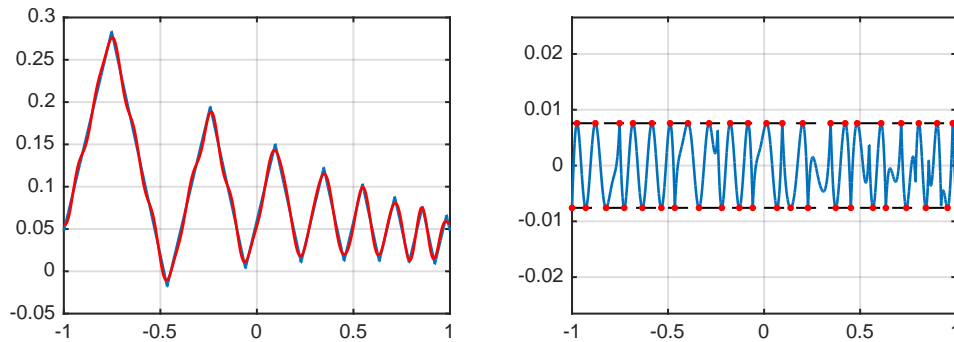


Figure 3.4: Degree 20 approximation of the same function.

Figure 3.4 is for the same problem but with a higher degree of approximation.

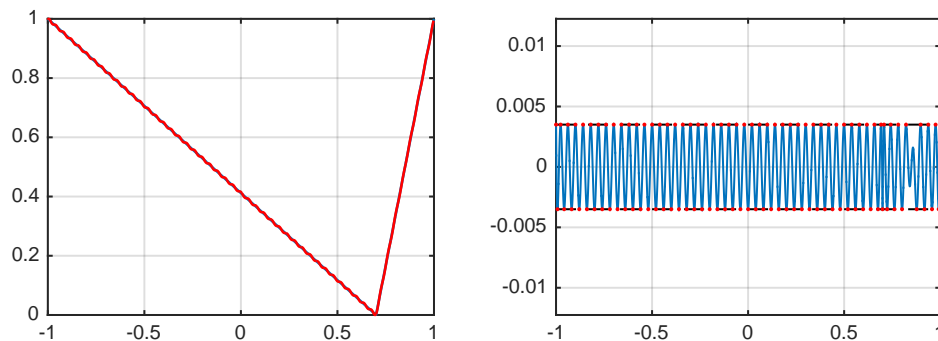


Figure 3.5: The shifted absolute value function and its degree 50 best approximation. The corresponding error curve is also shown.

Here is another example featuring the absolute value function, shifted to break the symmetry:

```

x = chebfun('x');
f = (-x+0.7)/(1+0.7).*(x<0.7) + (x-0.7)/(1-0.7).*(x>=0.7);
t = trigremez(f, 50);

```

The plots can be seen in Figure 3.5.

It is well known that for  $2\pi$ -periodic functions analytic on  $[0, 2\pi]$ , the degree- $n$  best approximation error decreases exponentially at the rate of  $O(e^{-(\alpha-\epsilon)n})$  for any  $\epsilon > 0$ . The parameter  $\alpha$  is the distance of the real line from the nearest singularity of the function [102]. As an example, we consider the function  $f(\theta) = (b + \cos \theta)^{-1}$  on the interval  $[-\pi, \pi]$  which has infinite poles in the complex  $\theta$ -plane, but the poles closest to the interval  $[-\pi, \pi]$  are the points  $\theta = -i \log(-b \pm \sqrt{b^2 - 1})$ . For  $b = 1.1$  the half-width of the strip of analyticity is  $\alpha \simeq 0.443568$ , while for  $b = 1.01$ ,  $\alpha \simeq 0.141304$ . In Figure 3.6, we see that the numerical decay of the error matches the predicted decay beautifully.

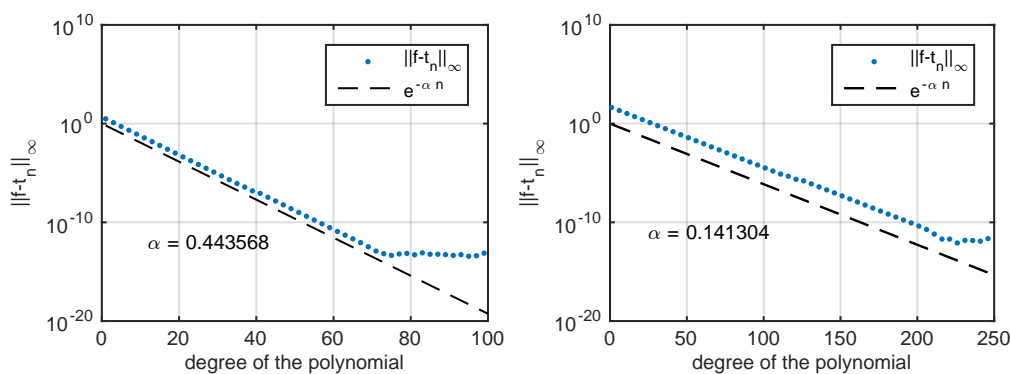


Figure 3.6: Error in the best approximation of  $f(\theta) = (b + \cos \theta)^{-1}$  as a function of the degree of the approximating polynomial.  $b = 1.1$  for the plot on the left,  $b = 1.01$  for the plot on the right.

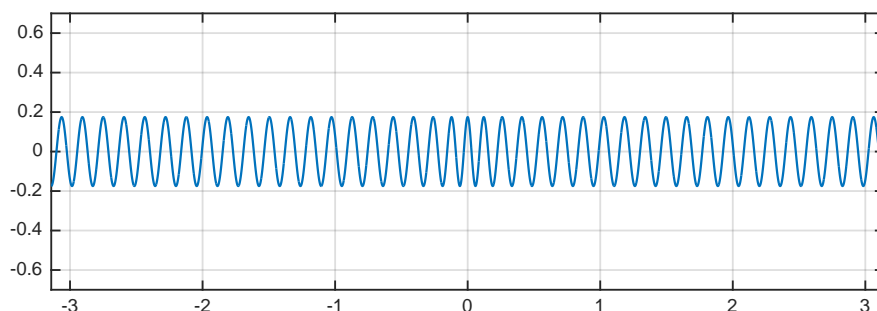


Figure 3.7: Almost sinusoidal error curve produced while approximating  $f(\theta) = (1.01 - \cos \theta)^{-1}$  by a degree 40 trigonometric polynomial.

A typical error curve of a best approximation in a space of algebraic polynomials looks approximately like a scaled Chebyshev polynomial. There are in fact precise theorems which prove that for functions which satisfy certain smoothness conditions, the error curves approach scaled versions of Chebyshev polynomials in the limit as the degree of approximation  $n \rightarrow \infty$ . As one might expect, in the space of trigonometric polynomials, the best approximation error curves

tend to look like sine waves, not Chebyshev polynomials. Asymptotic theorems to this effect would not be hard to prove but we are unaware of their presence in the literature. Experienced approximation theorists and practitioners are so used to seeing error curves that look like Chebyshev polynomials that it can be quite a surprise for them to see almost sinusoidal error curves. The following example illustrates this, and the almost sine-wave like error curve can be seen in Figure 3.7.

```
f = chebfun('1./(1.01-cos(x))', [-pi, pi], 'trig');
plot(f-trigremez(f,40)), ylim([-1 1])
```

### 3.6 Best trigonometric rational approximation

We now consider the problem of finding the best trigonometric *rational* approximation of a continuous periodic function on an interval. Without loss of generality, we consider the periodic interval  $[0, 2\pi]$  and let  $R_{mn}$  be the set of trigonometric rational functions of the form

$$r(\theta) = \frac{p(\theta)}{q(\theta)}, \quad (3.6.1)$$

where

$$p(\theta) = a_0 + \sum_{j=1}^m (a_j \cos j\theta + b_j \sin j\theta), \quad (3.6.2)$$

and

$$q(\theta) = c_0 + \sum_{j=1}^n (c_j \cos j\theta + d_j \sin j\theta), \quad (3.6.3)$$

and all the coefficients  $a_j, b_j, c_j$  and  $d_j$  are real.

In general, functions in  $R_{mn}$  may have poles in the interval  $[0, 2\pi]$ . Since continuous functions are bounded, for finding best trigonometric rational approximations, it is sufficient to search in the set  $\overline{R}_{mn} \subseteq R_{mn}$  of bounded rational functions, i.e., functions of the form  $r = p/q$ , such that  $q$  has no roots in  $[0, 2\pi]$ .

Let us recall the definition of a continuous periodic function (Definition 1.1). The formal statement of the problem is as follows.

**The problem:** Given a real-valued continuous periodic function  $f$  on  $[0, 2\pi]$ , and non-negative integers  $m$  and  $n$ , find a function  $r^* \in R_{mn}$  such that

$$\|f - r^*\|_\infty \leq \|f - r\|_\infty, \quad \forall r \in R_{mn}. \quad (3.6.4)$$

Neither the set  $\overline{R}_{mn}$ , nor the set  $R_{mn}$ , is a linear space. As a consequence, finding best trigonometric rational approximations in  $R_{mn}$  is a much harder problem in comparison with the problem of finding best trigonometric polynomial approximations that we discussed earlier in this

chapter. Before giving any details of a numerical algorithm for computing best trigonometric rational approximations, let us address the fundamental questions of existence and uniqueness.

The existence of  $r^* \in R_{mn}$  is guaranteed via the following theorem.

**Theorem 3.8.** *Given a real-valued continuous periodic function  $f$  on  $[0, 2\pi]$  and non-negative integers  $m$  and  $n$ , there exists a rational trigonometric function  $r(\theta) = p(\theta)/q(\theta)$  of the form (3.6.1) which best approximates  $f$  and without any loss of generality it can be assumed that the denominator  $q(\theta) > 0$  on  $[0, 2\pi]$ .*

*Proof.* See [25, Ch. 5, p. 157] or [26, Sec. 4]. □

The above theorem settles the question of existence. However, for characterization and uniqueness of best trigonometric rational approximation, we have to work a little harder. There are numerous references which discuss the problem of rational best approximations, mostly for the algebraic case. The work by Walsh [96] is a treatise on the subject of rational approximation but does not consider the trigonometric case. The work by Meinardus [66] is an excellent reference and presents a general theory of best rational approximations. The exposition in [66] does specialize to the two important cases of approximations by algebraic rational functions and approximations by ratios of sum of exponentials and proves the uniqueness of best approximation in each case. However, it does not discuss the trigonometric case. A classic reference on approximation theory is the two volume work by Rice [76], [77] and volume II is dedicated to nonlinear approximation. However, we have not been able to find a direct application of the theory presented in [77] to the case of trigonometric rational approximation. Another reference which presents theorems of existence and uniqueness for best rational approximations by quite a general class of functions is by Watson [98]. However, this too, does not discuss the trigonometric case. The book by Powell [73] discusses a numerical algorithm for algebraic rational approximations only, and even in that case, does not prove the existence and uniqueness of the best approximations. The books by Davis [31] and Rivlin [78] discuss rational approximations but not the trigonometric case. Cheney and Loeb discuss the trigonometric case in detail and give a specialized proof of the existence of the best trigonometric rational approximation [26]. However, quite surprisingly, they do not prove uniqueness. The general theory of best approximations developed by Cheney in [25] comes very close to proving uniqueness for our present case of best trigonometric rational approximations but can not be applied directly. Finally, the key paper which establishes the existence and uniqueness of the best trigonometric rational approximation is by Loeb [62, Sec. 4]. In the same paper, Loeb also gives a complete characterization of the best trigonometric rational approximation.

We prove in this section Theorem 3.10 which allows us to use theorems in [25] and [98] to give a precise characterization and a proof of uniqueness of best trigonometric rational approximations.

The theorem is not new and can be found in the paper by Loeb [62, Sec. 4]. However, in this section, we present a proof of Theorem 3.10 to give a coherent presentation of the subject.

Recall from Section 2.2 of Chapter 2 that given non-negative integers  $m$  and  $n$ , we say that a trigonometric rational function  $r = p/q$  is of *type*  $(m, n)$  if  $p$  and  $q$  can be chosen such that  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$ . We say that  $r = p/q$  is of *exact type*  $(\mu, \nu)$  if  $p$  and  $q$  are of exact degrees  $\mu$  and  $\nu$  respectively and there are no roots in common between  $p$  and  $q$ . In case  $r$  is identically zero, we say that it has the exact type  $(-\infty, 0)$ . Following [85, Ch. 24], we introduce the notion of *defect* of a trigonometric rational function  $r$  in  $R_{mn}$ .

**Definition 3.3** (*defect*). *Suppose  $r \in R_{mn}$  and has the exact type  $(\mu, \nu)$  where  $\mu \leq m$  and  $\nu \leq n$ . The defect of  $r$  in  $R_{mn}$  is the integer  $d$  between 0 and  $n$  defined by*

$$d = \min\{m - \mu, n - \nu\}. \quad (3.6.5)$$

*If  $r$  is identically zero, then regardless of the value of  $m$ , the defect of  $r$  in  $R_{mn}$  is  $n$ .*

The definition of defect is precisely connected with the maximum number of zeros a trigonometric rational function might have in the interval  $[0, 2\pi)$ . To see this connection, we suppose that  $r = p/q$  belongs to  $R_{mn}$  and has the exact type  $(\mu, \nu)$  and suppose further that  $u = s/t$  is another function in  $R_{mn}$ . Now consider the difference

$$r - u = \frac{p}{q} - \frac{s}{t} = \frac{pt - sq}{qt}, \quad (3.6.6)$$

which is a trigonometric rational function of type  $(\max\{\nu + n, m + \nu\}, n + \nu)$ . Since

$$m + n = \max\{\nu + n, m + \nu\} + \min\{m - \mu, n - \nu\}, \quad (3.6.7)$$

substituting the definition of  $d$  in the last term in the equation above and rearranging, we get

$$\max\{\nu + n, m + \nu\} = m + n - d. \quad (3.6.8)$$

This allows us to write the type of  $r - u$  in terms of the defect  $d$ , namely  $r - u$  is of type  $(m + n - d, 2n - d)$ . Theorem 1.1 of Chapter 1 now implies that  $r - u$  can have at most  $2(m + n - d)$  zeros in  $[0, 2\pi)$ . As we shall later see, this defect dependent zero count appears in almost all the results regarding equioscillation and uniqueness of best trigonometric rational approximations.

As mentioned earlier, the set  $R_{mn}$  is not a vector space, and this makes the problem harder. However, one can make considerable progress by defining a vector space which is closely related to  $R_{mn}$ . Towards this end, we first define the functions  $\psi_j$  and  $\kappa_j$  by the equations  $\psi_j(\theta) = \sin j\theta$ , and  $\kappa_j(\theta) = \cos j\theta$ . For a given  $r \in R_{mn}$ , we define  $\mathcal{T}_m + r\mathcal{T}_n$  as the vector subspace of  $C([0, 2\pi])$  spanned by the set

$$S = \{1, \psi_1, \kappa_1, \dots, \psi_m, \kappa_m, r, r\psi_1, r\kappa_1, \dots, r\psi_n, r\kappa_n\}. \quad (3.6.9)$$

We will shortly see that the vector space  $\mathcal{T}_m + r\mathcal{T}_n$  plays a crucial role in proving key results for the problem at hand. In order to prove that best rational trigonometric approximations are unique and are characterized by certain equioscillation of the error, we now follow [25] and prove two theorems (Theorem 3.9 and Theorem 3.10). It is noteworthy that while [25] establishes general theorems of characterization and uniqueness, those theorems cannot be directly applied to our present case of trigonometric rational approximations without proving that the vector space  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition and its dimension is related to the defect  $d$  of  $r$  in  $R_{mn}$ .

Our first theorem gives a bound on the dimension of the vector space  $\mathcal{T}_m + r\mathcal{T}_n$ , independent of the possible defect that  $r$  might have in  $R_{mn}$ . The bound is easy to prove and our proof follows a similar proof for the case of best algebraic rational approximations given in [25].

**Theorem 3.9.** *Let the dimension of  $\mathcal{T}_m + r\mathcal{T}_n$  be  $\delta$ . Then  $\delta \leq 2(m + n) + 1$ .*

*Proof.* If  $r \equiv 0$ , then  $\delta = 2m + 1$  and the theorem is true. Assume  $r \not\equiv 0$ . Since the number of elements in the spanning set  $S$  is  $2(m + n) + 2$ , we only need to show that  $S$  is linearly dependent. Since  $r \not\equiv 0$ , we can write

$$r(\theta) = \frac{p(\theta)}{q(\theta)} = \frac{a_0 + \sum_{k=1}^m (a_{2k-1} \sin k\theta + a_{2k} \cos k\theta)}{b_0 + \sum_{k=1}^n (b_{2k-1} \sin k\theta + b_{2k} \cos k\theta)}, \quad (3.6.10)$$

and therefore have the linear dependence

$$a_0 + \sum_{k=1}^m (a_{2k-1} \psi_k + a_{2k} \kappa_k) - \left[ b_0 r + \sum_{k=1}^n (b_{2k-1} r \psi_k + b_{2k} r \kappa_k) \right] = 0. \quad (3.6.11)$$

This implies that the dimension  $\delta$  of  $\mathcal{T}_m + r\mathcal{T}_n$  must be less than the number of elements in  $S$  and the theorem follows.  $\square$

Our next theorem proves that  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition defined earlier in the chapter (Definition 3.1) and also connects the dimension of  $\mathcal{T}_m + r\mathcal{T}_n$  with the defect  $d$  of  $r$  in  $R_{mn}$ . A similar result for the case of algebraic rational approximations has been proven in [25, Ch. 5, p. 162] and our proof closely follows that result. For a different proof of this theorem, see [62, Sec. 4].

**Theorem 3.10.** *Let  $r = p/q$ , with  $p \in \mathcal{T}_m$  and  $q \in \mathcal{T}_n$ ,  $q > 0$  on  $[0, 2\pi]$  and  $p/q$  irreducible. Assume also that  $p$  has exact degree  $\mu \leq m$  and  $q$  has exact degree  $\nu \leq n$ . Then the vector space  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition and is of dimension  $2(m + n - d) + 1$ , where  $d$  is the defect of  $r$  in  $R_{mn}$ .*

*Proof.* We first show that the dimension of  $\mathcal{T}_m + r\mathcal{T}_n$  is  $2(m + n - d) + 1$ . If  $r \equiv 0$ , then the dimension of  $\mathcal{T}_m + r\mathcal{T}_n$  is  $2m + 1$ . Also,  $p \equiv 0$ ,  $q \equiv 1$  and by definition  $r$  has the type

$(-\infty, 0)$ , that is,  $\mu = -\infty$ ,  $\nu = 0$  and the defect of  $r$  in  $R_{mn}$  is  $n$ . Therefore, in this case  $2(m + n - d) + 1 = 2m + 1$  and this is the dimension of  $\mathcal{T}_m + r\mathcal{T}_n$ .

Suppose now that  $r \neq 0$ . We have the dimension formula

$$\dim(\mathcal{T}_m + r\mathcal{T}_n) = \dim(\mathcal{T}_m) + \dim(r\mathcal{T}_n) - \dim(\mathcal{T}_m \cap r\mathcal{T}_n). \quad (3.6.12)$$

The dimension of  $\mathcal{T}_m$  is  $2m + 1$  and the dimension of  $r\mathcal{T}_n$  is  $2n + 1$ . Since  $r\mathcal{T}_n = \{(p/q)q_1 : \text{degree}(q_1) \leq n\}$ , and an element  $(p/q)q_1$  will belong also to  $\mathcal{T}_m$  if and only if  $q$  divides  $q_1$ , leaving a quotient of degree  $\leq m - \mu$ . In this event,  $q_1$  must be of the form  $qq_2$  with  $\text{degree}(q_2) \leq m - \mu$ . Since  $\text{degree}(q_1) \leq n$ , we must also have  $\text{degree}(q_2) \leq n - \nu$ . Thus

$$\dim(\mathcal{T}_m \cap r\mathcal{T}_n) = 2(\min\{n - \nu, m - \mu\}) + 1 = 2d + 1,$$

where we have used the definition (3.6.5) of the defect  $d$  in the last equality. Substituting this in the dimension formula gives

$$\dim(\mathcal{T}_m + r\mathcal{T}_n) = (2m + 1) + (2n + 1) - (2d + 1) = 2(m + n - d) + 1, \quad (3.6.13)$$

as was to be shown.

Now to prove that  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition, we only need to establish that its nontrivial elements can have at most  $2(m + n - d)$  roots in  $[0, 2\pi)$ . If one of its elements say  $p_1 + rq_1$  has  $2(m + n - d) + 1$  or more roots, then the trigonometric polynomial  $p_1q + pq_1$  also has  $2(m + n - d) + 1$  or more roots. But this is not possible, since this trigonometric polynomial is of degree at most  $\max\{m + \nu, n + \mu\} = m + n - d$  and by Theorem 1.1 can have at most  $2(m + n - d)$  roots. This proves that  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition and hence the theorem.  $\square$

We are now in a position to give a precise characterization of a best trigonometric rational approximation.

**Theorem 3.11** (Characterization of best trigonometric rational approximation). *In order that the irreducible trigonometric rational function  $r = p/q$  be a best approximation of  $f$  from the set  $R_{mn}$ , it is necessary and sufficient that the error  $f - r$  equioscillates on at least  $2(m + n - d) + 2$  points in  $[0, 2\pi)$ .*

*Proof.* Since we have already shown in Theorem 3.10 that the vector space  $\mathcal{T}_m + r\mathcal{T}_n$  satisfies the Haar condition, the theorem now follows from the *alternation theorem* given in [25, Ch. 5, pp. 161] or Theorem 9.3 in [98].  $\square$

**Theorem 3.12** (Uniqueness). *Let  $r^*$  be a best trigonometric rational approximation from  $R_{mn}$  to the function  $f$ . Then,  $r^*$  is unique.*

*Proof.* We have already shown via Theorem 3.10 that  $\mathcal{T}_m + r^*\mathcal{T}_n$  satisfies the Haar condition. The theorem now follows from the *unicity theorem* given in [25, Ch. 5, p. 164] or Theorem 9.4 in [98].  $\square$

### 3.7 A Remez algorithm for computing best trigonometric rational approximation

Let us now discuss the details of a Remez type numerical algorithm for computing the best trigonometric rational approximation of a function on an interval. As in the polynomial case, the first crucial step is to compute a best trigonometric rational approximation on a reference. An immediate problem is that the best approximation might suffer from a non-zero defect and this can cause problems for our algorithm since the defect cannot be determined a priori. We therefore assume that the defect  $d$  in the best approximation of  $f$  is 0.

The following theorem characterizes a best trigonometric rational approximation of discrete data on a reference in terms of the equioscillation of the error. For notational convenience, we define

$$N := m + n. \quad (3.7.1)$$

**Theorem 3.13.** *Let  $\{\theta_i : i = 0, 1, \dots, 2N + 1\}$  be a reference such that*

$$0 \leq \theta_0 < \theta_1 < \dots < \theta_{2N+1} < 2\pi. \quad (3.7.2)$$

*If  $r_k \in R_{mn}$  and if the equations*

$$r_k(\theta_i) + (-1)^i h_k = f(\theta_i), \quad i = 0, 1, \dots, 2N + 1, \quad (3.7.3)$$

*hold for some constant  $h_k$ , then  $r_k$  is the element in  $R_{mn}$  that minimizes the expression*

$$\max_{i=0,1,\dots,2N+1} |f(\theta_i) - r(\theta_i)|, \quad r \in R_{mn}. \quad (3.7.4)$$

*Proof.* Since the expression (3.7.4) has the value  $|h_k|$  when  $r$  is equal to  $r_k$ , it is sufficient to show that, if  $\tilde{r}$  is a function in  $R_{mn}$  that satisfies the condition

$$\max_{i=0,1,\dots,2N+1} |f(\theta_i) - \tilde{r}(\theta_i)| \leq |h_k|, \quad (3.7.5)$$

then  $\tilde{r}$  is equal to  $r_k$ . Equations (3.7.3) and (3.7.5) imply that each of the terms

$$\tilde{r}(\theta_i) - r_k(\theta_i) = [f(\theta_i) - r_k(\theta_i)] - [f(\theta_i) - \tilde{r}(\theta_i)], \quad i = 0, 1, \dots, 2N + 1 \quad (3.7.6)$$

is either zero or has the sign of  $(-1)^i h_k$ . This implies that the function  $(\tilde{r} - r_k)$  has at least  $2(m+n) + 1$  zeros in  $[0, 2\pi)$ . However, we may write  $(\tilde{r} - r_k)$  as the ratio of two trigonometric polynomials, where the degree of the numerator is at most  $m+n$  and by Theorem 1.1, it can have  $2(m+n)$  zeros at most. Therefore  $\tilde{r}$  is equal to  $r_k$ .  $\square$

If the conditions of Theorem 3.13 are satisfied and if  $r^*$  is the best approximation of  $f$  from the set  $R_{mn}$ , then it follows from Theorem 3.13 and from the definition of the best approximation that the bounds

$$|h_k| \leq \|f - r^*\|_\infty \leq \|f - r_k\|_\infty \quad (3.7.7)$$

must hold. From a computational point of view, once  $r_k$  is computed, we are back in a situation similar to the polynomial version of the algorithm. Given  $r_k$  corresponding to the  $k^{\text{th}}$  reference, we can use the same procedure as described for the polynomial case to move to the next reference and perform an iteration of the exchange algorithm such that  $h_{k+1} \geq h_k$ .

It remains to be shown how  $r_k$  can be computed. Towards that end, we now give details of the linear algebra involved in computing the best trigonometric rational approximation  $r_k$  on a reference. We suppose that  $r_k$  is the best approximation on a given reference and write  $r_k$  as

$$r_k(\theta) = \frac{a_0 + \sum_{j=1}^m (a_j \cos j\theta + b_j \sin j\theta)}{c_0 + \sum_{j=1}^n (c_j \cos j\theta + d_j \sin j\theta)}. \quad (3.7.8)$$

Since  $r_k \in R_{mn}$ , we can assume without loss of generality that

$$c_0 + \sum_{j=1}^n (c_j \cos j\theta + d_j \sin j\theta) = q(\theta) > 0, \quad \theta \in [0, 2\pi].$$

The system (3.7.3) can be explicitly written as

$$a_0 + \sum_{j=1}^m (a_j \cos j\theta_i + b_j \sin j\theta_i) = [f(\theta_i) - (-1)^i h_k] \left( c_0 + \sum_{j=1}^n (c_j \cos j\theta_i + d_j \sin j\theta_i) \right), \quad (3.7.9)$$

where  $i = 0, 1, \dots, 2N+1$ . We can also write these set of equations in matrix form as:

$$P\mathbf{p} = (F - h_k S) Q\mathbf{q} \quad (3.7.10)$$

where the  $(2N+2) \times (2m+1)$  matrix  $P$  is given by

$$P = \begin{bmatrix} 1 & \cos \theta_0 & \sin \theta_0 & \dots & \cos m\theta_0 & \sin m\theta_0 \\ 1 & \cos \theta_1 & \sin \theta_1 & \dots & \cos m\theta_1 & \sin m\theta_1 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & \cos \theta_{2N+1} & \sin \theta_{2N+1} & \dots & \cos m\theta_{2N+1} & \sin m\theta_{2N+1} \end{bmatrix} \quad (3.7.11)$$

and the  $(2N + 2) \times (2n + 1)$  matrix  $Q$  is given by

$$Q = \begin{bmatrix} 1 & \cos \theta_0 & \sin \theta_0 & \dots & \cos n\theta_0 & \sin n\theta_0 \\ 1 & \cos \theta_1 & \sin \theta_1 & \dots & \cos n\theta_1 & \sin n\theta_1 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & \cos \theta_{2N+1} & \sin \theta_{2N+1} & \dots & \cos n\theta_{2N+1} & \sin n\theta_{2N+1} \end{bmatrix}. \quad (3.7.12)$$

The matrix  $F$  is a diagonal square matrix of size  $(2N + 2) \times (2N + 2)$ ,

$$F = \begin{bmatrix} f(\theta_0) & & & & \\ & f(\theta_1) & & & \\ & & \ddots & & \\ & & & & f(\theta_{2N+1}) \end{bmatrix}, \quad (3.7.13)$$

while the matrix  $S$  is also a diagonal square matrix of size  $(2N + 2) \times (2N + 2)$  with the  $i^{\text{th}}$  diagonal entry equal to  $(-1)^i$ ,

$$S = \begin{bmatrix} 1 & & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & & (-1)^{2N+1} \end{bmatrix}. \quad (3.7.14)$$

Finally, the vectors containing the unknown coefficients  $\mathbf{p}$  and  $\mathbf{q}$  are given as

$$\mathbf{p} = \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_n \\ b_n \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} c_0 \\ c_1 \\ d_1 \\ \vdots \\ c_n \\ d_n \end{bmatrix}. \quad (3.7.15)$$

The system of equations (3.7.10) is non-linear because of the presence of the unknown levelled-error  $h_k$ . We can rewrite (3.7.10) as

$$\begin{bmatrix} P & -FQ \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = -h_k \begin{bmatrix} O & SQ \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}, \quad (3.7.16)$$

where  $O$  is the zero matrix of size  $(2N + 2) \times (2m + 1)$ . This is a generalized eigenvalue problem, with  $h_k$  as the eigenvalue and  $\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}$  as the eigenvector.

### 3.8 Numerical examples of best trigonometric rational approximations

Let us now look at some numerical examples of computing best trigonometric rational approximations. The Chebfun command `trigremez(f, m, n)` finds the type  $(m, n)$  best trigonometric approximation of a chebfun  $f$  on its underlying domain:

```
[p, q, r, error, status] = trigremez(f, m, n);
```

The outputs of the above `trigremez` command are as follows. The trigonometric polynomials  $p$  and  $q$  are periodic chebfuns forming the type  $(m, n)$  best trigonometric rational approximation of  $f$ . The Matlab function handle  $r$  is such that  $r(\tau) = p(\tau) ./ q(\tau)$ . The value of the minimax error is returned in `error`, while `status` is a Matlab structure with fields containing information about the final reference and number of iterations taken etc.

Here is a basic example in which we consider the 2-periodic function  $f(\theta) = e^{2\sin\pi\theta}$  on the interval  $[-1, 1]$  and compute its type- $(1, 1)$  best approximation. The result of this computation is shown in the left half of Figure 3.8. Since  $m = 1, n = 1$ , the error must equioscillate on at least  $2(m + n) + 2 = 6$  points in  $[-1, 1)$ . In fact, the error equioscillates exactly on 6 points and this is shown in the right half of Figure 3.8.

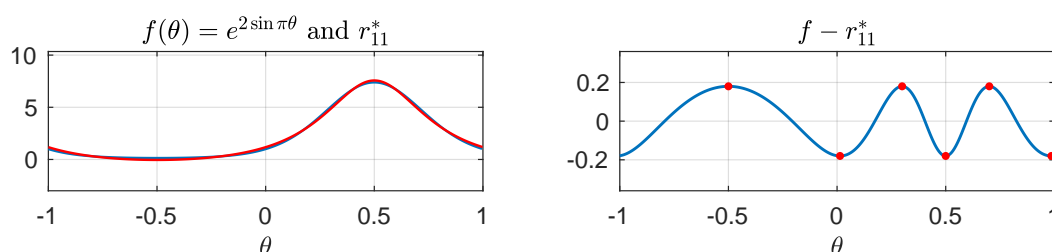


Figure 3.8: The function  $f(\theta) = e^{2\sin\pi\theta}$  and its best trigonometric rational approximation of type  $(1, 1)$  are plotted on the left. The error curve is plotted on the right which shows the equioscillation of the error at 6 points.

We carry on with the same example and using the `trigremez` command, compute the type  $(n, n)$  approximations of  $f$  for  $n = 2, 3, \dots, 7$ . The resulting error curves are shown in Figure 3.9. For  $n = 6$  the absolute error is of the order  $10^{-13}$  which is very close to the available tolerance of the algorithm. For  $n = 7$ , the best approximation has almost resolved the function to 14 digits of absolute accuracy or roughly 15 digits of relative accuracy.

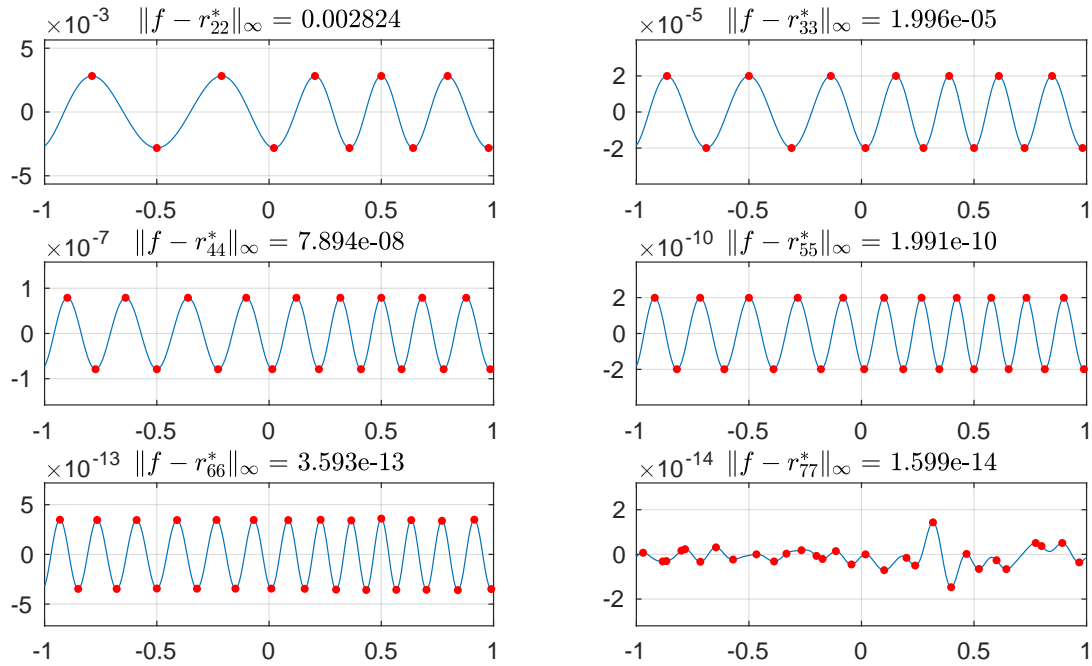


Figure 3.9: Equioscillating error curves for type  $(n, n)$  approximation of the function  $f(\theta) = e^{2 \sin \pi \theta}$  for  $n = 2, 3, \dots, 7$ . For the type  $(6, 6)$  approximation, the error curve (bottom left) is not exactly equioscillating since we are close to the tolerance of the algorithm. The type  $(7, 7)$  best approximation approximates the function close to machine precision and we see the effects of rounding errors on the error curve (bottom right).

We now consider the famous function  $f(\theta) = |\theta|$  on the interval  $[-1, 1]$  and compute its type  $(1, 1)$  best trigonometric approximation. The result of this computation is shown in the left half of Figure 3.10. The error curve is plotted in the right half of the same figure and we can see the equioscillation of the error on exactly  $2(m + n) + 2 = 6$  points in  $[-1, 1]$ .

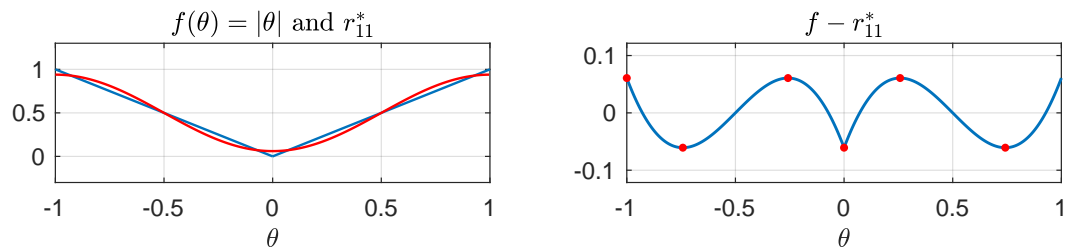


Figure 3.10: The function  $f(\theta) = |\theta|$  and its best trigonometric rational approximation of type  $(1, 1)$  are plotted on the left. The error curve is plotted on the right which shows the equioscillation of the error at 6 points.

We can in fact compute the type  $(n, n)$  approximations of  $f$  for  $n = 1, 2, \dots, 8$ . The resulting error curves are shown in Figure 3.11. We see the beautiful equioscillation of the error and the gradual concentration of the reference around the points where the function has a discontinuous derivative, namely the points  $\theta = -1, \theta = 0$ , and  $\theta = 1$ .

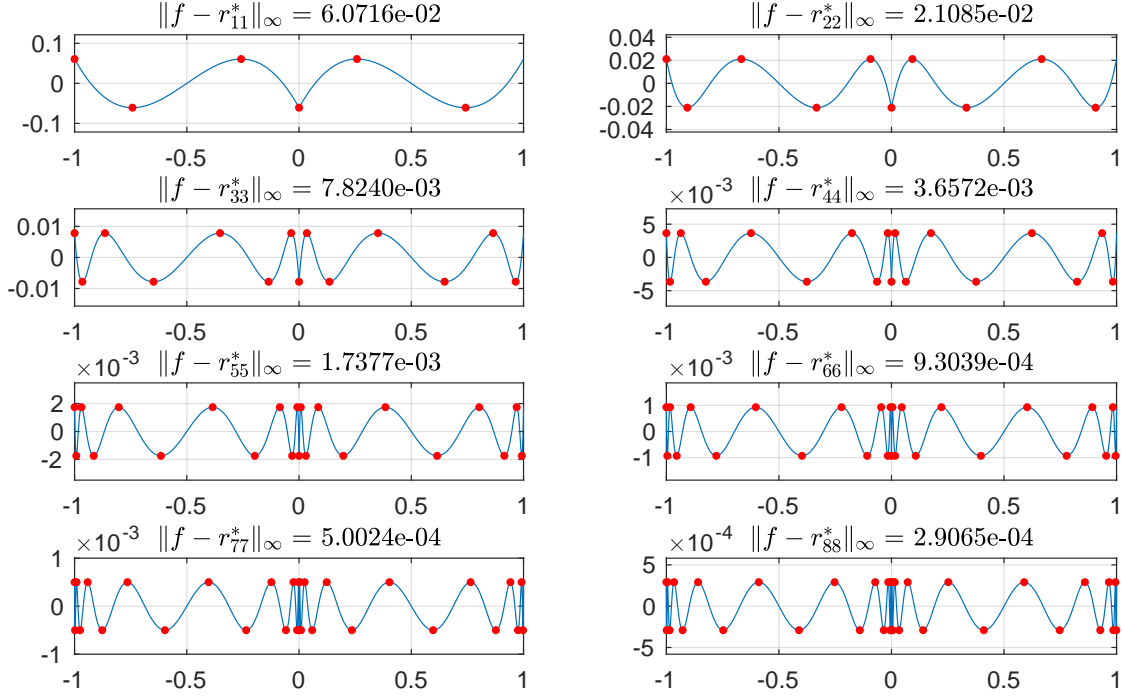


Figure 3.11: Equioscillating error curves for type  $(n, n)$  approximation of the function  $f(\theta) = |\theta|$  for  $n = 1, 2, \dots, 8$ .

In comparison with the previous example, the errors decrease much more slowly now. A comparison between the errors of the last two examples is shown in Figure 3.12. We now take a closer look at the rate of convergence for these two examples.

The function  $f(\theta) = e^{2 \sin \pi \theta}$  is entire and we can expand it explicitly into its trigonometric series using a Laurent series expansion of the function  $e^{\frac{z}{2}(t-t^{-1})}$  at  $t = 0$ . It is known that for complex numbers  $z, t, t \neq 0$ , we have the Laurent series expansion [101, Ch. XVII]

$$e^{\frac{z}{2}(t-t^{-1})} = \sum_{k=-\infty}^{\infty} J_k(z)t^k, \quad (3.8.1)$$

where  $J_k$  is the  $k^{\text{th}}$  order Bessel function of the first kind. If we substitute  $z = -ia$  and  $t = e^{ib\theta}$ , where  $a, b \in \mathbb{R}, b \neq 0$ , we obtain the trigonometric series,

$$e^{a \sin b\theta} = \sum_{k=-\infty}^{\infty} J_k(-ia)e^{ibk\theta}. \quad (3.8.2)$$

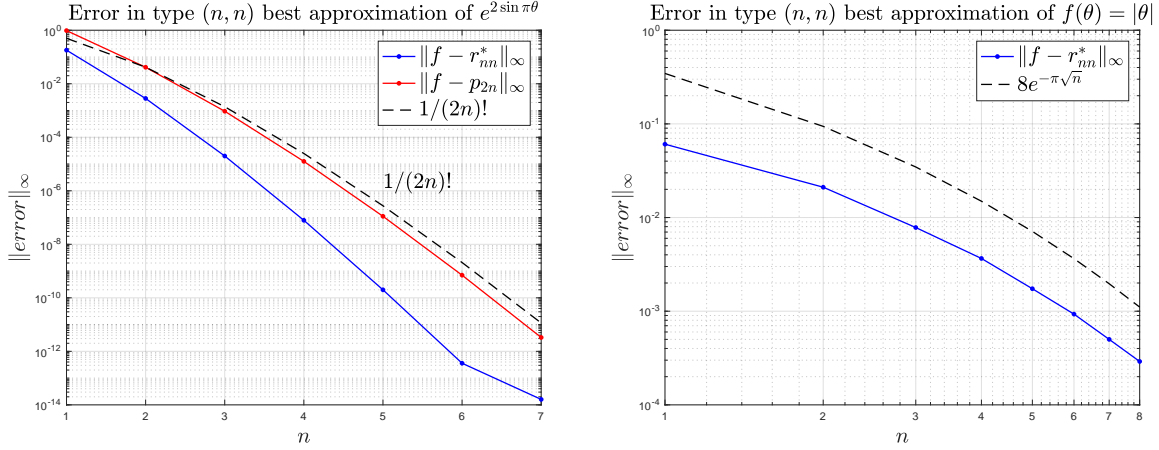


Figure 3.12: *Left:* Errors in type  $(n, n)$  best trigonometric rational approximations of  $f(\theta) = e^{2 \sin \pi \theta}$  plotted as a solid blue line using Matlab's `semilogy` plot. The solid red line plots the errors observed in trigonometric polynomial interpolants of degree  $2n$ . We see that these errors follow the predicted rate of  $1/(2n)!$ , as shown by the black dashed line. We also observe that the rate of convergence of the best trigonometric rational approximation in this case is even better than  $1/(2n)!$ . *Right:* We plot the errors in type  $(n, n)$  best trigonometric rational approximations of  $f(\theta) = |\theta|$  in blue. The errors are plotted on a `loglog` plot and hence the decay is much slower as compared to the picture on the left. However, the decay is still square root exponential. The black dashed line plots the function  $8e^{-\pi\sqrt{n}}$ , which is the asymptotic convergence rate of the type  $(n, n)$  best algebraic polynomial approximation of the absolute value function. We conjecture that the error in the best type  $(n, n)$  trigonometric approximation of the absolute value function also converges like  $O(e^{-\pi\sqrt{n}})$ .

Therefore, by putting  $a = 2$  and  $b = \pi$ , we get

$$e^{2 \sin \pi \theta} = \sum_{k=-\infty}^{\infty} J_k(-2i) e^{ik\pi\theta}. \quad (3.8.3)$$

For a non-negative integer  $k$ , the Taylor series expansion of  $J_k(z)$  at the point  $z = 0$  is given by [101, Ch. XVII],

$$J_k(z) = \sum_{j=0}^{\infty} \frac{(-1)^j}{j!(j+k)!} \left(\frac{z}{2}\right)^{2j+k}. \quad (3.8.4)$$

This gives,

$$J_k(-2i) = (-i)^k \sum_{j=0}^{\infty} \frac{1}{j!(j+k)!}, \quad (3.8.5)$$

where  $k$  is a non-negative integer. For negative orders, we use the relationship,

$$J_{-k}(z) = (-1)^k J_k(z), \quad k = 1, 2, \dots \quad (3.8.6)$$

It now follows from the last two equations that as  $|k|$  becomes large, we have the asymptotic relationship,

$$|J_k(-2i)| \sim \frac{1}{|k|!}. \quad (3.8.7)$$

Based on this observation, we can expect the errors incurred by trigonometric polynomial interpolants of degree  $n$  to converge super-exponentially at the rate  $1/n!$ . To make the comparison fair with best trigonometric rational approximations of type  $(n, n)$ , we use trigonometric polynomial interpolants of degree  $2n$ . The left half of Figure 3.12 shows that while the trigonometric polynomial interpolants converge at the expected rate of  $1/(2n)!$ , the rate of convergence of best trigonometric rational approximations is even faster than this.

We now turn our attention to the absolute value function. The best algebraic rational approximation of the absolute value function has been well studied (see, for example [85, Ch. 25]). Stahl showed in 1992 that the error in best type  $(n, n)$  algebraic rational approximation of  $|x|$  on  $[-1, 1]$  is asymptotically equal to  $8e^{-\pi\sqrt{n}}$  as  $n$  gets large [83]. In the right half of Figure 3.12, we see that the error in the best type  $(n, n)$  trigonometric rational approximation also seems to match the rate  $O(e^{-\pi\sqrt{n}})$ . However, a trigonometric rational function of type  $(n, n)$  has  $4n + 1$  parameters, whereas the best algebraic rational approximation with the same number of parameters is of type  $(2n, 2n)$  and converges at a much faster rate of  $8e^{-\pi\sqrt{2n}}$ . On the other hand, the function  $|\theta|$ , when viewed as a periodic function on  $[-1, 1]$ , has two singularities, one at  $\theta = -1$  (or equivalently at  $\theta = 1$ ) and the other at  $\theta = 0$ , whereas for algebraic rational approximations, the only singular point is  $\theta = 0$ . In other words, to approximate the absolute value function, a type  $(n, n)$  trigonometric function has twice the number of parameters available in a type  $(n, n)$  algebraic function but it also has to approximate twice the number of singularities. Therefore, based on this observation and our numerical experiment shown in the right half of Figure 3.12, we conjecture that the asymptotic rate of convergence of the type  $(n, n)$  best trigonometric rational approximation of the absolute value function is  $O(e^{-\pi\sqrt{n}})$ . A proof of the yet unknown true rate of convergence remains an open and interesting research problem.

### 3.9 Conclusion and future work

In this chapter, we have presented a Remez algorithm for finding the minimax trigonometric approximation of a function. We first considered the case of trigonometric polynomial approximations and then extend our algorithm to the rational case. The two key steps used in our algorithm are trigonometric interpolation and determining the extrema of the error function. In the case of polynomials, we use the second kind barycentric formula for interpolation. For finding the extrema, both for the polynomial and the rational case, we use Chebfun's root finding algorithm. The algorithm we developed is already a part of Chebfun<sup>4</sup> and can be used in solving real-world problems as well as for fundamental studies in approximation theory.

---

<sup>4</sup>To download the latest version of Chebfun, visit [www.chebfun.org](http://www.chebfun.org).

Our algorithm can easily be modified to design digital filters with real but asymmetric frequency responses, see [65]. Specifically, this can be achieved by making sure that during the iterations of the algorithm, no point of the reference set lies in the don't-care regions specified. This results in best approximation on a compact subset of  $[0, 2\pi]$ . Details will be reported elsewhere.

The computation of best approximations via the Remez algorithm and its variants is a nonlinear process. An alternative is the Carathéodory-Fejér (CF) method [88], based on singular values of a Hankel matrix of Chebyshev coefficients, which has been used to compute near-best approximations of algebraic polynomials [93], [47]. We are working on a CF algorithm for periodic functions based on the singular values of a matrix of trigonometric coefficients.

## Chapter 4

# Trigonometric Padé Approximation

In the final chapter of the thesis, we review trigonometric Padé approximation of a function and explore two approaches to the problem. We have decided to choose the name *trigonometric Padé* instead of the more common name of *Fourier-Padé*. The reason for this has to do with the naming convention adopted by Chebfun to use the prefix *trig* instead of *four* for most of its codes for approximation of periodic functions [102].

The mathematics of this chapter is not new and we will briefly discuss the history of Padé and trigonometric (Fourier) Padé approximations in the following sections. The numerical solution of the problem is non-trivial and a robust numerical algorithm is still under investigation. We believe that the idea of SVD robustness, applied successfully for computing Padé approximants [39], can be extended for computing Chebyshev–Padé and also for trigonometric Padé approximants. We are unaware of any existing algorithms which use SVD based robustness for computing these approximations.

The outline of this short chapter is as follows. In the first section, we briefly review the central idea of a Padé approximation. Since a trigonometric series can be mapped to a Laurent series and vice-versa, we next review Laurent–Padé approximations. We show that the trigonometric Padé approximation of a  $2\pi$ -periodic formal trigonometric series can be obtained by carrying out the following three steps.

1. Transform the given trigonometric series to a Laurent series.
2. Compute the Laurent–Padé approximation of the series obtained in the last step.
3. Transform the computed Laurent–Padé approximation back to a trigonometric Padé approximation.

This method is then implemented in a numerical algorithm and examples of computing various type  $(m, n)$  trigonometric Padé approximants are given.

We then briefly discuss an alternative Padé approximation to a Laurent series, proposed by Trefethen and Gutknecht [89]. The chapter is ended with a short conclusion and possible future directions.

## 4.1 Padé approximation

The Padé approximation is a generalization of Taylor series to rational functions. A Taylor approximant is a polynomial whereas a Padé approximant is a rational function. Let us suppose that  $f$  is a function of the complex variable  $z$  and  $f$  is holomorphic at the origin. Therefore,  $f$  has a convergent Taylor series expansion<sup>1</sup>,

$$f(z) = \sum_{k=0}^{\infty} c_k z^k, \quad (4.1.1)$$

valid in an open neighbourhood of the origin. For any non-negative integer  $m$ , if our problem is to find a degree- $m$  polynomial approximation  $p$  of  $f$  such that

$$(f - p)(z) = O(z^{m+1}), \quad (4.1.2)$$

then  $p$  is obtained by simply truncating the Taylor series of  $f$ . This is true because the Taylor series expansion of  $f$  satisfies the relation,

$$f(z) = \sum_{k=0}^m c_k z^k + O(z^{m+1}). \quad (4.1.3)$$

Now suppose that  $R_{mn}$  is the set of rational functions of type  $(m, n)$ , and instead of a polynomial approximation of  $f$ , we seek a rational function  $r \in R_{mn}$  such that

$$(f - r)(z) = O(z^{m+n+1}) \quad (4.1.4)$$

where

$$r(z) = \frac{a_0 + a_1 z + \cdots + a_m z^m}{b_0 + b_1 z + \cdots + b_n z^n}. \quad (4.1.5)$$

This is the key idea of a Padé approximant and some authors take (4.1.4) as the definition of a Padé approximation of  $f$  [13]. This seems a reasonable definition because  $r$  has  $n + m + 1$  free parameters and one can hope that the Taylor series expansion of  $r$  about the origin will match the first  $n + m + 1$  coefficients of the Taylor expansion of  $f$ . However, there are functions for which no  $r \in R_{mn}$  satisfies the condition (4.1.4). For such functions, the convergence order  $O(z^{m+n+1})$  is too much to ask for. However, if we relax the convergence order in (4.1.4) to be  $O(z^{\text{maximum}})$  instead, where the number

$$\text{maximum} \leq m + n + 1 \quad (4.1.6)$$

---

<sup>1</sup>A Taylor series expansion at  $z = 0$  is also called a Maclaurin series expansion.

denotes the maximum exponent possible in the order relation (4.1.4), then the existence and uniqueness of the Padé approximant of  $f$  is guaranteed. We will not discuss these fine differences here and state the following theorem from [85, Ch. 27].

**Theorem 4.1** (*Characterization of Padé approximants*). *For each  $m, n \geq 0$ , a function  $f$  has a unique Padé approximant  $r_{mn} \in R_{mn}$  defined by the condition*

$$(f - r_{mn})(z) = O(z^{\text{maximum}}), \quad (4.1.7)$$

*and a function  $r \in R_{mn}$  is equal to  $r_{mn}$  if and only if  $(f - r)(z) = O(z^{m+n+1-d})$ , where  $d$  is the defect of  $r$  in  $R_{mn}$ .*

It must be stated that while in most practical applications of the Padé approximant, the function being approximated has a convergent power series at the origin, this is not required. All one needs is a formal power series [13].

There is a huge body of literature for the Padé problem. We refer the reader to the classic book by Baker and Graves-Morris [13] which contains numerous results, examples, applications and references. For an introduction to numerical solutions of the problem, see [85, Ch. 27], which also contains references regarding the history of the problem. For a state of the art SVD based robust algorithm, see [39].

## 4.2 Laurent-Padé approximation

As mentioned in the last section, Padé approximations are usually constructed for functions which are analytic at the origin. As a natural consequence, most of the times the literature on Padé approximation either discusses the problem for the case when the function has a convergent power series at the origin, or discusses only the approximation of a formal power. However, for a number of applications, the function being approximated is only analytic in an annular region, i.e., has a convergent Laurent series. For such problems, it is more appropriate to consider the Padé approximation of a formal Laurent series, rather than the approximation of a formal power series.

The idea of extending Padé approximation to series other than a power series is not limited to the case of Laurent series only. Over the years, mathematicians have defined Padé type approximants for various formal series. A popular case is the Chebyshev-Padé approximation. (See, for example, [28], [33], [37], [43], [63] and [89].)

A considerable amount of work has been done in extending Padé approximations to Laurent series. Gragg's landmark review of the Padé table in [42] mentioned a possible extension of the idea of a Padé table to Laurent series but did not develop the idea further. A couple of

years later, Gragg and Johnson published [44] which was perhaps the first paper to discuss a generalization of Padé approximations to Laurent series. The first paper to mention the idea of a Fourier–Padé table was again by Gragg [43]. A few years later, Baker and Graves-Morris published their two volume treatise on Padé approximations [11], [12]. The second part of this widely popular work briefly presented *Padé–Fourier* approximants [12, Sec. 1.6], and highlighted the relatively nascent state of such approximations at that time. To quote in their own words [12, Sec. 1.6, pp. 62]:

*There is a host of different approaches to the Padé–Laurent problem. Construction of approximants with attractive mathematical properties and convergence theorems is still at an early stage for Padé–Laurent series, and much remains to be done.*

Three years later, Bultheel published his paper on the block structure of the Laurent–Padé table and showed that it contains rectangular blocks [22]. He also showed that these rectangular blocks become square in the case of Chebyshev–Padé because the corresponding Laurent series is symmetric [22]. Three years further down the line, Bultheel wrote a whole book on the subject of Laurent–Padé approximations [23]. Just about the same time, Trefethen and Gutknecht developed a different approach for arriving at a Padé approximation of a Laurent series and applied it to construct Chebyshev–Padé approximations [89].

There is some confusion about the definition a Laurent–Padé approximation. Even more confusingly, there is also a Padé–Laurent approximation which some authors define differently from a Laurent–Padé approximation. See, for example, [13, Sec. 7.4 and 7.5]. We will use the definition given by Gragg and Johnson [43]. Their idea of a Laurent–Padé approximant is as follows. Given a formal Laurent series,

$$F(z) = \sum_{k=-\infty}^{\infty} c_k z^k, \quad (4.2.1)$$

and non-negative integers  $m$  and  $n$ , the Laurent–Padé approximation of  $F$  is constructed by finding Laurent polynomials [64]  $p$  and  $q$  of degrees<sup>2</sup>  $m$  and  $n$  respectively such that the rational function  $r = p/q$  matches both negatively and positively indexed Laurent coefficients of  $F$  as far out as possible, that is, the coefficients  $c_{-N}, \dots, c_N$  are matched, where  $N$  is as large as possible [23]. To construct such an approximation, Gragg and Johnson split the Laurent series of  $F$  into its analytic and co-analytic parts [43],

$$F(z) = F^+(z) + F^-(z^{-1}), \quad (4.2.2)$$

and this splitting is done at the constant term  $c_0$ , which is divided equally between  $F^+(z)$  and  $F^-(z)$ . Then two type  $(\max\{m, n\}, n)$  Padé approximations  $r^+$  and  $r^-$  corresponding to the

---

<sup>2</sup>A Laurent polynomial of degree  $m$  is a function of the form  $\sum_{k=-m}^m c_k z^k$ .

power series  $F^+$  and  $F^-$ , respectively, are constructed. The sum of these two approximants is then defined as the Laurent–Padé approximation of  $F$  [43],

$$r(z) = \frac{p(z)}{q(z)} := r^+(z) + r^-(z^{-1}). \quad (4.2.3)$$

Note that it is not obvious that the sum on the right hand side above can be written as a ratio of two Laurent polynomials of degree  $m$  and  $n$  respectively. However, we will show that this is true regardless of whether  $m < n$  or not.

We now derive the Laurent–Padé approximation of a Laurent series following [13, Sec. 7.4] and [43]. Let us consider the formal Laurent series,

$$F(z) = \sum_{k=-\infty}^{\infty} c_k z^k. \quad (4.2.4)$$

Define

$$F^+(z) := \frac{c_0}{2} + \sum_{k=1}^{\infty} c_k z^k, \quad (4.2.5)$$

and

$$F^-(z^{-1}) := \frac{c_0}{2} + \sum_{k=1}^{\infty} c_{-k} z^{-k}, \quad (4.2.6)$$

such that

$$F(z) = F^+(z) + F^-(z^{-1}). \quad (4.2.7)$$

Let  $m$  and  $n$  be non-negative integers and let  $M = \max\{m, n\}$ . We now seek two type  $(M, n)$  Padé approximants  $r^+(z)$  and  $r^-(z^{-1})$  for the power series  $F^+(z)$  and  $F^-(z^{-1})$ , respectively, and then define the sum  $r^+(z) + r^-(z^{-1})$  as the Laurent–Padé approximation of  $F(z)$ . Gragg calls this approach the method of *additive splitting* [43].

In this section, wherever a summation symbol appears with a prime ( $\sum'$ ), the coefficient  $c_0$  is to be replaced with  $c_0/2$ . Consider the linear system  $L^+$ , defined by the equations

$$\sum_{j=0}^i {}'c_{(i-j)} \beta_j^+ = \alpha_i^+, \quad \text{if } 0 \leq i \leq M, \quad (4.2.8)$$

and

$$\sum_{j=0}^n c_{(i-j)} \beta_j^+ = 0, \quad \text{if } m+1 \leq i \leq m+n. \quad (4.2.9)$$

Similarly, consider the linear system  $L^-$ , defined by the equations

$$\sum_{j=0}^i {}'c_{-(i-j)} \beta_j^- = \alpha_i^-, \quad \text{if } 0 \leq i \leq M, \quad (4.2.10)$$

and

$$\sum_{j=0}^n c_{-(i-j)} \beta_j^- = 0, \quad \text{if } m+1 \leq i \leq m+n. \quad (4.2.11)$$

We now define four polynomials  $p^\pm$  and  $q^\pm$  as follows.

$$p^+(z) = \sum_{i=0}^M \alpha_i^+ z^i, \quad q^+(z) = \sum_{j=0}^n \beta_j^+ z^j \neq 0. \quad (4.2.12)$$

$$p^-(z^{-1}) = \sum_{i=0}^M \alpha_i^- z^{-i}, \quad q^-(z^{-1}) = \sum_{j=0}^n \beta_j^- z^{-j} \neq 0. \quad (4.2.13)$$

Let us define the notation

$$[g(z)]_{n_1}^{n_2} = \sum_{k=n_1}^{n_2} c_k z^k, \quad (4.2.14)$$

where  $g(z) = \sum_{k=-\infty}^{\infty} c_k z^k$ , and  $n_1, n_2$  are integers such that  $n_1 \leq n_2$ . Using this notation, we can think of  $q^\pm$  as polynomials satisfying the conditions,

$$[F(z)q^+(z)]_{m+1}^{m+n} = 0, \quad [F(z^{-1})q^-(z^{-1})]_{m+1}^{m+n} = 0, \quad (4.2.15)$$

while  $p^\pm$  can be interpreted as polynomials satisfying the conditions [13, Sec. 7.4]:

$$p^+(z) = [F(z)q^+(z)]_0^M, \quad p^-(z^{-1}) = [F(z^{-1})q^-(z^{-1})]_0^M. \quad (4.2.16)$$

Note that the degree of  $p$  is not necessarily  $m$  but  $M = \max\{m, n\}$ . This means that for  $m < n$ ,  $p$  is a polynomial of degree  $n$ .

Using these four polynomials, we define the following two Padé approximations of  $F^+(z)$  and  $F^-(z^{-1})$ , respectively,

$$r^+(z) = \frac{p^+(z)}{q^+(z)}, \quad r^-(z^{-1}) = \frac{p^-(z^{-1})}{q^-(z^{-1})}. \quad (4.2.17)$$

The type of both  $r^+$  and  $r^-$  is  $(\max\{m, n\}, n)$ . This seems problematic, since we are seeking an approximation of type  $(m, n)$ . However, our definition of the Laurent-Padé approximation of  $F$  will ensure that the final degrees of numerator are at most  $m$  and  $n$ , respectively. We define the Laurent-Padé approximation  $r$  of  $F$  by the expression

$$r(z) \equiv \frac{p(z)}{q(z)} = r^+(z) + r^-(z^{-1}), \quad (4.2.18)$$

where

$$p(z) = p^+(z)q^-(z^{-1}) + p^-(z^{-1})q^+(z), \quad (4.2.19)$$

and

$$q(z) = q^+(z)q^-(z^{-1}). \quad (4.2.20)$$

Since both  $q^+$  and  $q^-$  are polynomials of degree  $n$ , we see from the last expression (4.2.20) that  $q$  is indeed a Laurent polynomial of degree  $n$ . However, the expression (4.2.19) for  $p$  is rather involved and it is not clear whether  $p$  is a Laurent polynomial of degree  $m$ , especially when

$m < n$ . We now present the following lemma and its proof which shows that  $p$  is a Laurent polynomial of degree  $m$  for all non-negative integers  $m$  and  $n$ . The lemma and its proof are given in [13, Sec. 7.4].

**Lemma 4.1.** *The degree of the Laurent polynomial  $p$  defined by (4.2.19) does not exceed  $m$ .*

*Proof.* First consider the case when  $m \geq n$ . In this case,  $p$  has the form,

$$p(z) = \left( \sum_{j=0}^n \beta_j^- z^{-j} \right) \left( \sum_{i=0}^m \alpha_i^+ z^i \right) + \left( \sum_{j=0}^n \beta_j^+ z^j \right) \left( \sum_{i=0}^m \alpha_i^- z^{-i} \right). \quad (4.2.21)$$

There are two terms above and each is a product of two sums. All powers of  $z$  resulting from these products are between  $-m$  and  $m$ . Hence, when  $m \geq n$ ,  $p$  is a Laurent polynomial of degree  $m$  at most.

We now consider the case when  $m < n$ . The form of  $p$  is now,

$$p(z) = \left( \sum_{j=0}^n \beta_j^- z^{-j} \right) \left( \sum_{i=0}^n \alpha_i^+ z^i \right) + \left( \sum_{j=0}^n \beta_j^+ z^j \right) \left( \sum_{i=0}^n \alpha_i^- z^{-i} \right). \quad (4.2.22)$$

Let  $a_k$  be the coefficient of  $z^k$  in  $p$ . We want to show that  $a_k = 0$  for  $m < k \leq n$ . Using Cauchy's product rule,

$$a_k = \sum_{i=k}^n \alpha_i^+ \beta_{i-k}^- + \sum_{j=k}^n \alpha_{j-k}^- \beta_j^+. \quad (4.2.23)$$

Now using the definitions of  $\alpha_i^+$  and  $\alpha_i^-$  from (4.2.8) and (4.2.10) respectively, we arrive at,

$$a_k = \sum_{i=k}^n \beta_{i-k}^- \sum_{j=0}^i {}'c_{i-j} \beta_j^+ + \sum_{j=k}^n \beta_j^+ \sum_{i=0}^{j-k} {}'c_{i+k-j} \beta_i^- \quad (4.2.24)$$

$$= \sum_{i=k}^n \beta_{i-k}^- \sum_{j=0}^i {}'c_{i-j} \beta_j^+ + \sum_{i=k}^n \beta_{i-k}^- \sum_{j=i}^n {}'c_{i-j} \beta_j^+ \quad (4.2.25)$$

$$= \sum_{i=k}^n \beta_{i-k}^- \sum_{j=0}^n c_{i-j} \beta_j^+. \quad (4.2.26)$$

Using (4.2.9), we get  $a_k = 0$ . Hence, when  $m < n$ ,  $p$  is a Laurent polynomial of degree  $m$  at most.

This proves the lemma. □

It is now very easy to define a trigonometric Padé approximation in terms of a Laurent–Padé approximation. We show this in the following section.

### 4.3 Trigonometric Padé approximation

Let  $f$  be a  $2\pi$ -periodic function and assume that  $f$  is analytic<sup>3</sup> in a horizontal strip in the complex  $\theta$ -plane which contains the real axis, that is,

$$f(\theta) = \sum_{k=-\infty}^{\infty} c_k e^{ik\theta}. \quad (4.3.1)$$

We now move to the  $z$ -plane by letting  $z = e^{i\theta}$ , and defining  $F(z) := f(\theta)$ , i.e.,

$$F(z) = \sum_{k=-\infty}^{\infty} c_k z^k. \quad (4.3.2)$$

After this transformation, we are now back to (4.2.4) of the last section. Given  $m, n \geq 0$ , we construct a type  $(m, n)$  Laurent–Padé approximation of  $F$ , and arrive as before at (4.2.18),

$$r(z) \equiv \frac{p(z)}{q(z)} = r^+(z) + r^-(z^{-1}).$$

The type  $(m, n)$  trigonometric Padé approximation of  $f$  is now defined by substituting  $z = e^{i\theta}$  in  $r(z)$  [33], [13, Sec. 7.4],

$$r(e^{i\theta}) \equiv \frac{p(e^{i\theta})}{q(e^{i\theta})} = r^+(e^{i\theta}) + r^-(e^{-i\theta}) = \frac{p^+(e^{i\theta})}{q^+(e^{i\theta})} + \frac{p^-(e^{-i\theta})}{q^-(e^{-i\theta})}. \quad (4.3.3)$$

From (4.2.20), we find that

$$q(e^{i\theta}) = q^+(e^{i\theta})q^-(e^{-i\theta}), \quad (4.3.4)$$

and therefore,  $q(e^{i\theta})$  is a trigonometric polynomial of degree  $n$  at most. Also, from (4.2.19) it follows that

$$p(e^{i\theta}) = p^+(e^{i\theta})q^-(e^{-i\theta}) + p^-(e^{-i\theta})q^+(e^{i\theta}), \quad (4.3.5)$$

and applying Lemma 4.1, we find that  $p(e^{i\theta})$  is a trigonometric polynomial of degree  $m$  at most.

The method described in this section is implemented in Chebfun. We briefly describe this in the next section.

---

<sup>3</sup> The analyticity of  $f$  is not needed and one can begin with a formal trigonometric series. However, in most applications,  $f$  will be analytic on the real line. Examples of such applications include the harmonic oscillator, approximation of rational functions on the real line which have poles in the complex plane, convergence acceleration, integral equations, approximation of quantum mechanical equations, electrical circuits and many more [9], [10], [13].

## 4.4 Numerical algorithm

Let us refer back to the linear system defined by (4.2.9), which can be solved to obtain the coefficients  $\beta_j^+, j = 0, 1, \dots, n$  of the denominator polynomial  $q^+$ . Let  $\mathbf{b}$  denote the  $(n + 1)$ -vector whose  $j^{\text{th}}$  component is  $\beta_j^+$ . Looking at (4.2.9), we find that the vector  $\mathbf{b}$  lies in the null-space of a Toeplitz matrix  $C$  involving the coefficients  $c_k$ , that is,

$$C\mathbf{b} = \mathbf{0}. \tag{4.4.1}$$

We compute this using Matlab's `null` command, which amounts to saying that an SVD of  $C$  is computed.

Once the vector  $\mathbf{b}$  is computed, we normalize it by dividing each component of  $\mathbf{b}$  by its first component  $\beta_0^+$ . Next, the coefficients  $\alpha_j^+, j = 0, 1, \dots, M$  of the numerator polynomial  $p^+$  can be computed using (4.2.8), where  $\beta_j^+$  is taken to be 0 for  $j > n$ . Similarly, one can compute the coefficients of  $q^-$  and  $p^-$  by solving (4.2.11) and (4.2.10), respectively.

The coefficients of the numerator  $p$  defined by (4.2.19), and the denominator  $q$  defined by (4.2.20), are computed via discrete convolutions. Care must be taken in this step to account for positive and negative powers of  $z$  and appropriate zero padding<sup>4</sup> must be done before using Matlab's built-in `conv` command for convolution.

In the next step, we discard coefficients which are below a certain tolerance. This step is especially desirable if  $m < n$  since in this case the convolution sum corresponding to  $p$  results in the cancellation of coefficients of order greater than  $m$  and leaves some rounding errors. We discard these rounding errors using a tolerance of  $10^{-15}$ . Alternatively, one can leave these small rounding errors for Chebfun, which can automatically neglect very small coefficients, and is the next step of the algorithm.

The coefficients of  $p$  and  $q$  are now directly used to construct periodic chebfuns `p` and `q` using the Chebfun constructor with the `'coeffs'` and `'trig'` flags as shown below.

```
p = chebfun(pk, 'coeffs', 'trig');    % pk has the numerator coefficients
q = chebfun(qk, 'coeffs', 'trig');    % qk has the denominator coefficients
```

This completes our description of the numerical algorithm. Following other Chebfun commands for computing periodic approximations, we make our algorithm available under the command `trigpade`. In particular, the command

```
[p, q] = trigpade(f, m, n)
```

---

<sup>4</sup>By *zero padding* we mean adding appropriate number of zeros at either end of an array.

computes a type  $(m, n)$  trigonometric Padé approximant of the periodic chebfun  $\mathbf{f}$  and represents the approximant via periodic chebfuns  $\mathbf{p}$  and  $\mathbf{q}$ . Some examples where `trigpade` is used are presented in the next section.

## 4.5 Numerical examples

In our first example, we compute a type  $(1, 4)$  trigonometric Padé approximation of

$$f(\theta) = e^{\sin 3\pi\theta + \cos \pi\theta},$$

on  $[-1, 1]$ . Note that  $m = 1, n = 4$  and hence  $m < n$ . This case has been considered historically problematic [89]. However, the example shows that our algorithm works well at least for this smooth function and for small values of  $m$  and  $n$ . The function and the approximant  $r_{14}$  are plotted in the top row of Figure 4.1. The coefficients of  $f - p/q$  are also shown in the bottom row. We can see that the approximant is not defective and  $m + n = 5$  coefficients are matched on either side of the zeroth order coefficient.

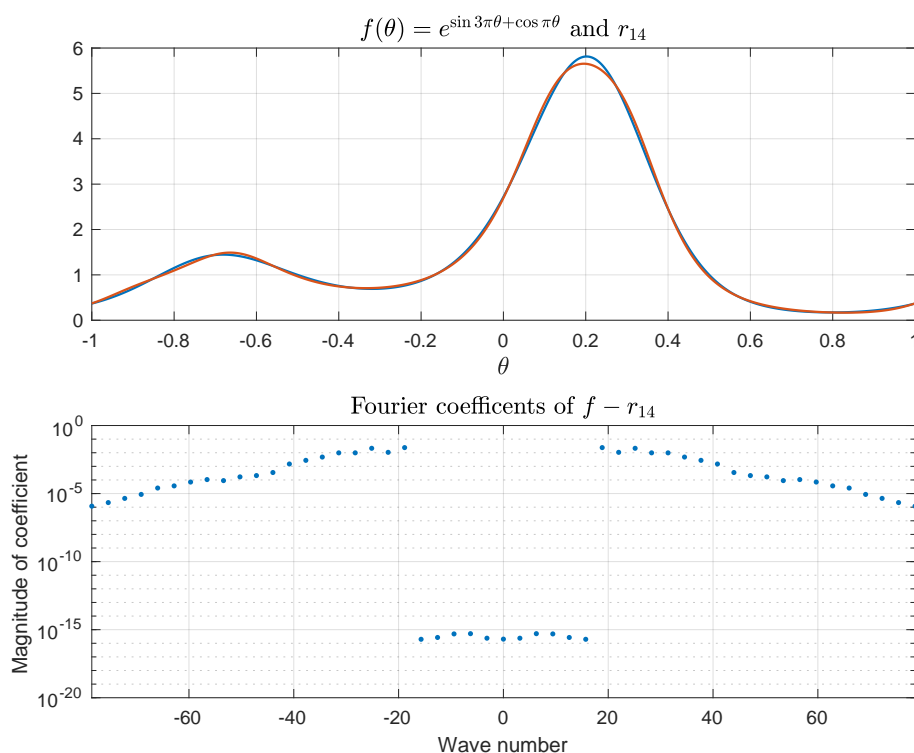


Figure 4.1: *Top*: Type  $(1, 4)$  trigonometric Padé approximation of  $f(\theta) = e^{\sin 3\pi\theta + \cos \pi\theta}$  on  $[-1, 1]$ . *Bottom*: The coefficients of the error  $f - r_{14}$  show that all coefficients of order less than or equal to  $m + n = 5$  are matched.

We can in fact compute type  $(m, 4)$  trigonometric Padé approximants for various values of  $m$ .

The range includes values of  $m$  less than  $n = 4$  and also greater than  $n = 4$ . The results are shown in Figure 4.2.

As another example of the case when  $m < n$ , we consider the trigonometric rational function,

$$f(\theta) = \frac{1 + \cos 2\pi\theta + \sin 3\pi\theta}{3 + \sin 3\pi\theta + \cos 5\pi\theta},$$

which we reproduced by a type (3,5) trigonometric rational interpolant in Section 2.4.1 of Chapter 2. We can reproduce  $f$  by a type (3,5) trigonometric Padé approximant as well:

```
>> fh = @(t) (1+cos(2*pi*t)+sin(3*pi*t))./(3 + sin(3*pi*t)+cos(5*pi*t));
f = chebfun(fh, 'trig');
[p, q] = triggade(f, 3, 5);
norm(f-p./q, inf)
ans =
    7.930614836965503e-16
```

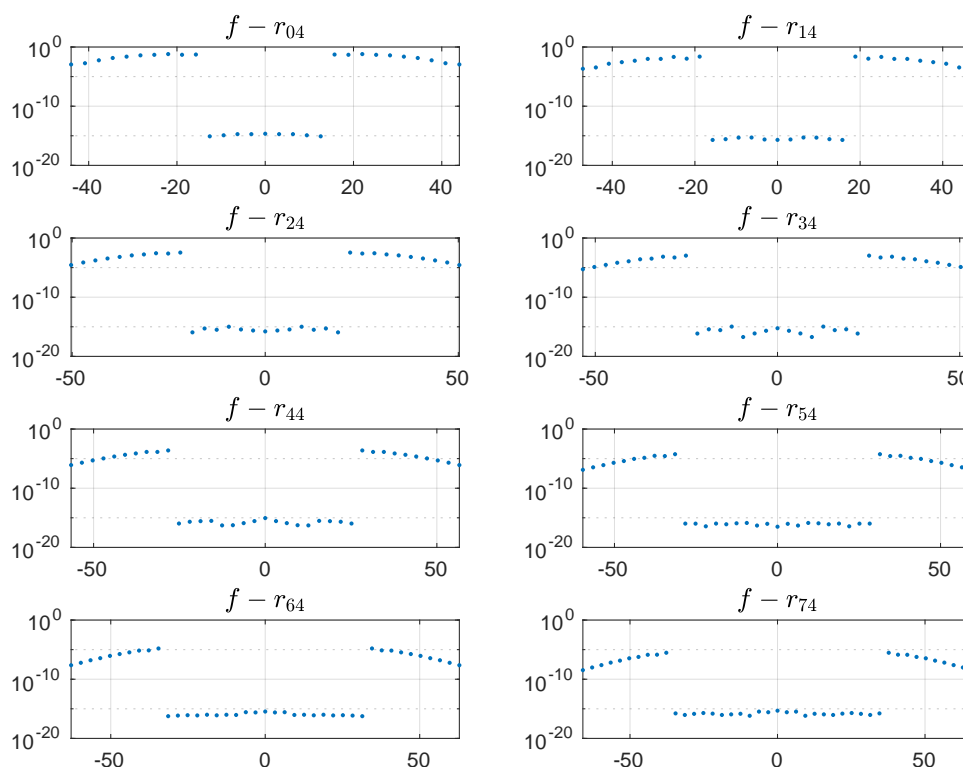


Figure 4.2: The absolute values of the Fourier coefficients of the error  $f - r_{mn}$  for various type  $(m, n)$  trigonometric Padé approximations of the function  $f(\theta) = e^{\sin 3\pi\theta + \cos \pi\theta}$  on  $[-1, 1]$ . In all cases shown, coefficients of order  $m + n$  are matched by the approximant.

In our next example, we numerically solve Example 1 from [13, Sec. 7.4, p. 381]. The original example is in the  $z$ -variable and concerns a Laurent–Padé approximation; however, we transform the problem to a trigonometric Padé approximation using  $z = e^{i\theta}$ . Here is a restatement of the example in the  $\theta$ -variable.

Find the trigonometric Padé approximation of type (1, 2) for

$$f(\theta) = b \sum_{i=0}^{\infty} h^i e^{i\theta} + c \sum_{i=0}^{\infty} k^{-i} e^{-i\theta}, \quad \text{for } h < 1, k > 1. \quad (4.5.1)$$

To solve the problem numerically, we let  $h = 1/9, k = 10, b = 2$  and  $c = 1$ . For our choice of  $h$  and  $k$ ,  $f$  can be approximated to machine precision by using the truncated series,

$$f(\theta) \approx b \sum_{i=0}^{N-1} h^i e^{i\theta} + c \sum_{i=0}^{N-1} k^{-i} e^{-i\theta}, \quad (4.5.2)$$

where  $N = 16$ . We can type the following code to solve our problem.

```
>> h = 1/9; k = 10;
>> b = 2; c = 1; N = 16;
>> c_k = [k.^(-N:-1).'; b+c; h.^(1:N).'];
>> f = chebfun(c_k, 'coeffs', 'trig');
>> [p, q] = trigpade(f, 1, 2);
```

The exact solution in the example shows that  $r$  has defect 1 in  $R_{12}$  and in fact,  $q$  has degree 1, which implies that a chebfun representation of  $q$  should be possible by a trigonometric polynomial interpolant through  $2(1) + 1 = 3$  points. Also,  $r$  reproduces  $f$ . Our algorithm detects this defect, and approximates  $f$  to machine precision, as shown by the output below.

```
>> length(q)           % i.e. number of points of interpolation used to represent q
ans =
     3
>> norm(f-p./q, inf)  % compute the maximum error
ans =
    1.872805921501850e-15
```

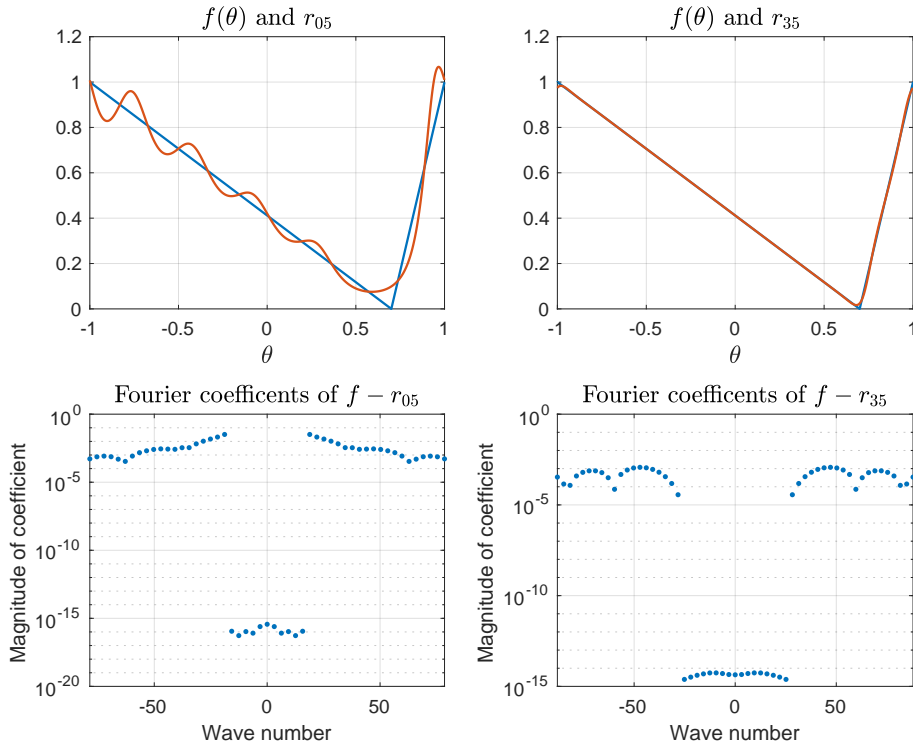


Figure 4.3: Type (0,5) and type (3,5) trigonometric Padé approximations of a shifted and stretched absolute value function on  $[-1, 1]$ .

In the next example, we compute type (0,4) and type (3,5) trigonometric Padé approximants of a shifted and stretched absolute value function. Let

$$f(\theta) = \begin{cases} -(\theta - a)/(1 + a) & \text{if } -1 \leq \theta \leq a, \\ (\theta - a)/(1 - a) & \text{if } a < \theta \leq 1, \end{cases}$$

Any non-zero value of  $a \in (-1, 1)$  makes sure that  $f$  doesn't have symmetric Fourier coefficients. The results for  $a = 0.7$  are shown in Figure 4.3.

In our final example, we compute a type (2,2) trigonometric Padé approximation of the absolute value function on  $[-1, 1]$ . The resulting error curve is compared with the error curve of the type (2,2) best trigonometric approximation in Figure 4.4

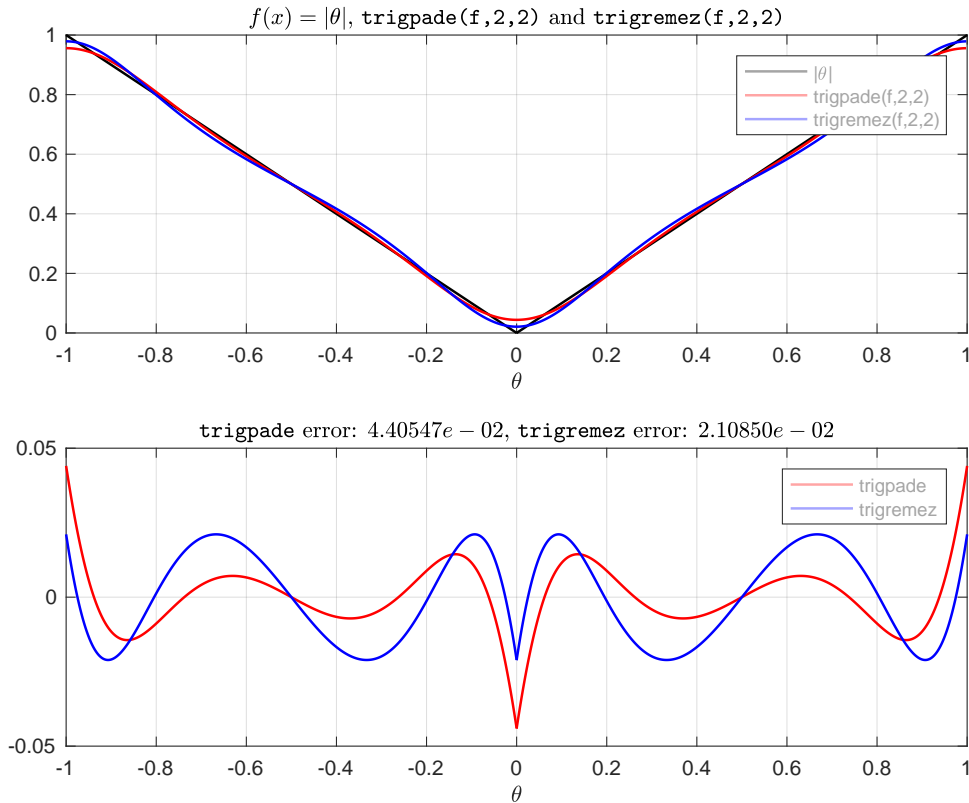


Figure 4.4: *Top*: Type (2, 2) trigonometric best and Padé approximations of the absolute value function on  $[-1, 1]$ . *Bottom*: The error curves show that at points away from the discontinuities, i.e., the points  $-1, 0$  and  $1$ , the trigonometric Padé approximant is better than the best approximation.

The method explained in Section 4.2 seems a natural way of extending the concept of a Padé approximation to a Laurent series and the numerical examples presented in this section seem promising. However, more thorough investigation and analysis of the algorithm and its mathematics are needed before conclusions can be drawn. It is also known that the approach discussed thus far breaks down if the denominator  $q$  has zeros inside the unit disk [89].

We now discuss a different approach to the problem.

## 4.6 Extended rational approximation to a Laurent series

Given the formal Laurent series (4.2.1), we have defined its Laurent–Padé approximation  $r = p/q$  in (4.2.18), where  $p$  and  $q$  are Laurent polynomials. Alternatively, one can also seek rational approximation of a formal Laurent series in a different set. Following Trefethen and Gutknecht

[89], we suppose that  $L$  is the set of formal Laurent series,

$$F(z) = \sum_{k=-\infty}^{\infty} c_k z^k, \quad (4.6.1)$$

and  $L_{-1}$  is the set of right-truncated formal Laurent series of the form

$$\sum_{k=-\infty}^{-1} c_k z^k. \quad (4.6.2)$$

A type  $(m, n)$  Padé approximation of an element in  $L$  is now sought in the set of *extended rational functions* [89]

$$\tilde{R}_{mn} = z^{m-n+1} (L_{-1} + R_{n-1, n}). \quad (4.6.3)$$

The elements of the set  $\tilde{R}_{mn}$  are formal expressions of the form

$$\tilde{r}(z) = r^-(z) + r^+(z) = r^-(z) + \frac{p(z)}{q(z)} = \sum_{k=-\infty}^{m-n} d_k z^k + \frac{\sum_{k=m-n+1}^m a_k z^k}{\sum_{k=0}^n b_k z^k}, \quad b_0 = 1. \quad (4.6.4)$$

A function  $\tilde{r} \in \tilde{R}_{mn}$  can be expressed as an extended rational function,

$$\tilde{r}(z) = \frac{\sum_{k=-\infty}^m a_k z^k}{\sum_{k=0}^n b_k z^k}, \quad b_0 = 1. \quad (4.6.5)$$

To construct a Padé approximation of  $F(z)$ , we split the series [46], [89]

$$F(z) = F^+(z) + F^-(z) = \sum_{k=-\infty}^{m-n} c_k z^k + \sum_{k=m-n+1}^{\infty} c_k z^k. \quad (4.6.6)$$

One can equivalently split the series at other indices as well, however, it is convenient to split  $F(z)$  as shown above [89]. We define

$$r^-(z) = F^-(z), \quad r^+(z) = z^{m-n+1} r_{n-1, n}^p, \quad (4.6.7)$$

where  $r_{n-1, n}^p$  is the type  $(n-1, n)$  Padé approximation of  $z^{n-m+1} F^+(z)$ , with the special case of  $r^+ \equiv 0$  if  $n = 0$ . A Padé approximant of the Laurent series  $F(z)$  is now defined as

$$\tilde{r}^\ell(z) := r^-(z) + r^+(z). \quad (4.6.8)$$

It is shown in [89] that an expression involving  $\tilde{r}^\ell(z)$  and the standard mapping

$$T_k(x) = \frac{1}{2} (z^{-k} + z^k), \quad (4.6.9)$$

where  $T_k$  is the  $k^{\text{th}}$  Chebyshev polynomial of the second kind, can be used to construct a Chebyshev–Padé approximation of the corresponding formal Chebyshev series on an interval.

We believe that analogously  $\tilde{r}^\ell(z)$  and the mapping  $z = e^{i\theta}$  might be used to construct a Padé approximation of a formal trigonometric series on a periodic interval. This problem is currently under investigation.

## 4.7 Conclusion and future work

In this chapter we first reviewed Padé and then Laurent–Padé approximation. Following Gragg [43], we showed how Laurent–Padé approximation may be used to construct trigonometric Padé approximation. We also discussed the implementation of this approach in Chebfun and presented numerical examples. We then briefly reviewed the approach suggested by Trefethen and Gutknecht [89] which seeks approximations of Laurent series in a set of extended rational functions. It appears that the construction of a Chebyshev–Padé approximant is very similar to the construction of a *real* trigonometric Padé approximant since the former is based on a symmetric, while the latter on a conjugate symmetric Laurent series [89]. The problems are so similar that it is hard to imagine that their solutions will differ. We also think that the idea of SVD based robustness can be extended to the computation of both trigonometric and Chebyshev–Padé approximations. These matters are currently under investigation.

# Appendices

## A Appendix: Formula for the levelled reference error $h_k$

Let us again consider the linear system (3.4.3):

$$f(\theta_j) - t_k(\theta_j) = (-1)^j h_k, \quad j = 0, 1, \dots, 2m + 1, \quad (\text{A.1})$$

where the points  $\theta_j \in [0, 2\pi)$  are distinct. The polynomial  $t_k$  is of degree  $m$ , which implies that there are  $2m + 1$  unknown parameters needed to completely determine  $t_k$ . However, the linear system consists of  $2m + 2$  equations with  $2m + 1$  unknowns corresponding to  $t_k$  and the unknown levelled error  $h_k$ . We will now show how the unknown  $h_k$  can be determined without solving the linear system.

We begin by expressing the degree  $m$  trigonometric polynomial  $t_k$  in a trigonometric Lagrangian basis. Since there is one extra collocation point, say  $\theta_j$ , we may define the  $j^{\text{th}}$  set of basis functions  $\beta_j$  as

$$\beta_j = \left\{ \ell_i^j \in \mathcal{T}_m, i = 0, 1, \dots, j-1, j+1, \dots, 2m+1 \right\}, \quad (\text{A.2})$$

where

$$\ell_i^j(\theta) := \prod_{\substack{\nu=0 \\ \nu \neq i, j}}^{2m+1} \sin \frac{1}{2}(\theta - \theta_\nu) \bigg/ \prod_{\substack{\nu=0 \\ \nu \neq i, j}}^{2m+1} \sin \frac{1}{2}(\theta_i - \theta_\nu). \quad (\text{A.3})$$

Note that the function  $\ell_i^j(\theta)$  is a degree  $m$  trigonometric polynomial that takes the value 1 at  $\theta = \theta_i$  and 0 at all other nodes except  $\theta = \theta_j$ , where it takes an unknown value. Therefore for a given  $j \in \{0, 1, \dots, 2m + 1\}$ , the polynomial  $t_k$  can be expressed as

$$t_k(\theta) = \sum_{\substack{i=0 \\ i \neq j}}^{2m+1} t_k(\theta_i) \ell_i^j(\theta) \quad (\text{A.4})$$

Using the above expression to evaluate  $t_k$  at  $\theta_j$ , we can write the original linear system (A.1) as:

$$f(\theta_j) - \sum_{\substack{i=0 \\ i \neq j}}^{2m+1} t_k(\theta_i) \ell_i^j(\theta_j) = (-1)^j h_k, \quad j = 0, 1, \dots, 2m + 1. \quad (\text{A.5})$$

We can simplify the number  $\ell_i^j(\theta_j)$  by writing

$$\ell_i^j(\theta_j) = \prod_{\substack{\nu=0 \\ \nu \neq i, j}}^{2m+1} \frac{\sin \frac{1}{2}(\theta_j - \theta_\nu)}{\sin \frac{1}{2}(\theta_i - \theta_\nu)} \quad (\text{A.6})$$

$$= \frac{\prod_{\substack{\nu=0 \\ \nu \neq j}}^{2m+1} \sin \frac{1}{2}(\theta_j - \theta_\nu)}{\prod_{\substack{\nu=0 \\ \nu \neq i}}^{2m+1} \sin \frac{1}{2}(\theta_i - \theta_\nu)} \times \frac{\sin \frac{1}{2}(\theta_i - \theta_j)}{\sin \frac{1}{2}(\theta_j - \theta_i)} = -\frac{w_i}{w_j}, \quad (\text{A.7})$$

where

$$w_j^{-1} = \prod_{\substack{\nu=0 \\ \nu \neq j}}^{2m+1} \sin \frac{1}{2} (\theta_j - \theta_\nu). \quad (\text{A.8})$$

This allows us to write (A.5) as

$$f(\theta_j)w_j - \sum_{\substack{i=0 \\ i \neq j}}^{2m+1} t_k(\theta_i)w_i = (-1)^j w_j h_k, \quad j = 0, 1, \dots, 2m+1. \quad (\text{A.9})$$

Now summing over  $j$  we get

$$\sum_{j=0}^{2m+1} f(\theta_j)w_j - \sum_{j=0}^{2m+1} \sum_{\substack{i=0 \\ i \neq j}}^{2m+1} t_k(\theta_i)w_i = \sum_{j=0}^{2m+1} (-1)^j w_j h_k. \quad (\text{A.10})$$

The double sum above collapses to zero when we use the fact that

$$\sum_{i=0}^{2m+1} t_k(\theta_i)w_i = 0. \quad (\text{A.11})$$

We can therefore rearrange (A.10) to write  $h_k$  as

$$h_k = \left( \sum_{j=0}^{2m+1} w_j f(\theta_j) \right) / \left( \sum_{j=0}^{2m+1} (-1)^j w_j \right). \quad (\text{A.12})$$

which is the required closed form expression.

# Bibliography

- [1] M. J. Ablowitz and A. S. Fokas. *Complex Variables: Introduction and Applications*. Cambridge University Press, 2003.
- [2] L. V. Ahlfors. *Complex Analysis: An Introduction to the Theory of Analytic Functions of One Complex Variable*. McGraw-Hill, Inc., 1953.
- [3] J. L. Aurentz, T. Mach, R. Vandebril, and D. S. Watkins. Fast and backward stable computation of roots of polynomials. *SIAM Journal on Matrix Analysis and Applications*, 36(3):942–973, 2015.
- [4] J. L. Aurentz and L. N. Trefethen. Chopping a Chebyshev series. *To appear in ACM Transactions on Mathematical Software*, 2017.
- [5] J. L. Aurentz, R. Vandebril, and D. S. Watkins. Fast computation of the zeros of a polynomial via factorization of the companion matrix. *SIAM Journal on Scientific Computing*, 35(1):A255–A269, 2013.
- [6] A. P. Austin. *Some New Results on and Applications of Interpolation in Numerical Computation*. PhD thesis, University of Oxford, 2016.
- [7] A. P. Austin, P. Kravanja, and L. N. Trefethen. Numerical algorithms based on analytic function values at roots of unity. *SIAM Journal on Numerical Analysis*, 52(4):1795–1821, 2014.
- [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 Journal of Numerical Analysis*, 2017.
- [9] G. A. Baker. *Essentials of Padé Approximants*. Academic Press, 1975.
- [10] G. A. Baker and J. L. Gammel. *The Padé Approximant in Theoretical Physics*. Academic Press, 1970.
- [11] G. A. Baker and P. R. Graves-Morris. *Padé Approximants, Part I: Basic Theory*. Addison-Wesley, 1981.

- [12] G. A. Baker and P. R. Graves-Morris. *Padé Approximants, Part II: Extensions and Applications*. Addison-Wesley, 1981.
- [13] G. A. Baker and P. R. Graves-Morris. *Padé Approximants*. Cambridge University Press, second edition, 1996.
- [14] R. Baltensperger. Some results on linear rational trigonometric interpolation. *Computers and Mathematics with Applications*, 43(6):737–746, 2002.
- [15] Z. Battles and L. N. Trefethen. An extension of MATLAB to continuous functions and operators. *SIAM Journal on Scientific Computing*, 25(5):1743–1770, 2004.
- [16] J.-P. Berrut. Baryzentrische formeln zur trigonometrischen interpolation (i). *Zeitschrift für Angewandte Mathematik und Physik*, 35(1):91–105, 1984.
- [17] J.-P. Berrut. Baryzentrische formeln zur trigonometrischen interpolation (ii) stabilität und anwendung auf die fourieranalyse bei ungleichabständigen stützstellen. *Zeitschrift für Angewandte Mathematik und Physik*, 35(2):193–205, 1984.
- [18] J.-P. Berrut. Rational functions for guaranteed and experimentally well-conditioned global interpolation. *Computers & Mathematics with Applications*, 15(1):1–16, 1988.
- [19] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.
- [20] D. Braess. *Nonlinear Approximation Theory*. Springer-Verlag, Berlin, 1986.
- [21] D. M. Bressoud. *A Radical Approach to Real Analysis*. MAA, second edition, 2007.
- [22] A. Bultheel. On the block structure of the Laurent-Padé table. In *Rational approximation and interpolation (Tampa, Fla., 1983)*, volume 1105 of *Lecture Notes in Math.*, pages 160–169. Springer, Berlin, 1984.
- [23] A. Bultheel. *Laurent Series and their Padé Approximations*. Birkhäuser, 1987.
- [24] A. L. Cauchy. *Cours d’analyse de l’Ecole Royale Polytechnique*, volume 1. Imprimerie Royale, 1821.
- [25] E. W. Cheney. *Introduction to Approximation Theory*. AMS Chelsea Publishing, Providence, Rhode Island, 1998.
- [26] E. W. Cheney and H. L. Loeb. Generalized rational approximation. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 1(1):11–25, 1964.
- [27] E. W. Cheney and T. J. Rivlin. A note on some Lebesgue constants. *Rocky Mountain J. Math.*, 6(3):435–440, 1976.

- [28] C. W. Clenshaw and K. Lord. Rational approximations from Chebyshev series. *Studies in Numerical Analysis*, pages 95–113, 1974.
- [29] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.
- [30] P. J. Davis. On the numerical integration of periodic analytic functions. In *On numerical approximation. Proceedings of a Symposium, Madison, April 21-23, 1958*, Publication no. 1 of the Mathematics Research Center, U.S. Army, the University of Wisconsin, pages 45–59. The University of Wisconsin Press, Madison, 1959.
- [31] P. J. Davis. *Interpolation and Approximation*. Dover Publications, 1975.
- [32] C. de Boor and A. Pinkus. Proof of the conjectures of Bernstein and Erdős concerning the optimal nodes for polynomial interpolation. *Journal of Approximation Theory*, 24(4):289–303, 1978.
- [33] T. A. Driscoll and B. Fornberg. A Padé-based algorithm for overcoming the Gibbs phenomenon. *Numerical Algorithms*, 26(1):77–92, 2001.
- [34] W. L. Ferrar. *A Text-book of Convergence*. The Clarendon Press, 1938.
- [35] S.-I. Filip. A robust and scalable implementation of the Parks-McClellan algorithm for designing FIR filters. *ACM Transactions on Mathematical Software*, 43(1):7:1–7:24, Aug 2016.
- [36] C. F. Gauss. Theoria interpolationis methodo novo tractata. *Werke, Vol. III, Dieterich, Göttingen*, 1866.
- [37] K. O. Geddes. Block structure in the Chebyshev–Padé table. *SIAM Journal on Numerical Analysis*, 18(5):844–861, 1981.
- [38] J. F. Geer. Rational trigonometric approximations using Fourier series partial sums. *Journal of Scientific Computing*, 10(3):325–356, 1995.
- [39] P. Gonnet, S. Güttel, and L. N. Trefethen. Robust Padé approximation via SVD. *SIAM Review*, 55(1):101–117, 2013.
- [40] P. Gonnet, R. Pachón, and L. N. Trefethen. Robust rational interpolation and least-squares. *Electronic Transactions on Numerical Analysis*, 38:146–167, 2011.
- [41] U. Graf. *Introduction to Hyperfunctions and their Integral Transforms: An Applied and Computational Approach*. Birkhäuser, Basel, 2010.
- [42] W. B. Gragg. The Padé table and its relation to certain algorithms of numerical analysis. *SIAM Review*, 14(1):1–62, 1972.

- [43] W. B. Gragg. Laurent, Fourier and Chebyshev–Padé tables. *Padé and Rational Approximation*, Academic Press, New York, pages 61–72, 1977.
- [44] W. B. Gragg and G. D. Johnson. The Laurent–Padé table. *Information Processing*, 74:632–637, 1974.
- [45] L. Greengard and J.-Y. Lee. Accelerating the nonuniform fast Fourier transform. *SIAM Review*, 46(3):443–454, 2004.
- [46] M. H. Gutknecht. Stable row recurrences for the Padé table and generically superfast lookahead solvers for non-Hermitian Toeplitz systems. *Linear Algebra and its Applications*, 188:351–421, 1993.
- [47] M. H. Gutknecht and L. N. Trefethen. Real polynomial Chebyshev approximation by the Carathéodory–Fejér method. *SIAM Journal on Numerical Analysis*, 19(2):358–371, 1982.
- [48] M. H. Gutknecht, L. N. Trefethen, et al. Real and complex Chebyshev approximation on the unit disk and interval. *American Mathematical Society*, 8(3), 1983.
- [49] P. Henrici. Barycentric formulas for interpolating trigonometric polynomials and their conjugates. *Numerische Mathematik*, 33(2):225–234, 1979.
- [50] P. Henrici. *Applied and Computational Complex Analysis: Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions*, volume 3. John Wiley & Sons, Inc., 1986.
- [51] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.
- [52] E. Hille. *Analytic Function Theory, Volume I*. Ginn and Company, Boston, 1959.
- [53] C. G. J. Jacobi. Über die Darstellung einer Reihe gegebner Werthe durch eine gebrochne Rationale Function. *Journal für die Reine und Angewandte Mathematik*, 30:127–156, 1846.
- [54] M. Javed and L. N. Trefethen. A trapezoidal rule error bound unifying the Euler–Maclaurin formula and geometric convergence for periodic functions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 470(2161):20130571, 2014.
- [55] M. Javed and L. N. Trefethen. The Remez algorithm for trigonometric approximation of periodic functions. Technical Report 15/03, Numerical Analysis Group, Mathematical Institute, Oxford, 2015.
- [56] M. Javed and L. N. Trefethen. Euler–Maclaurin and Gregory interpolants. *Numerische Mathematik*, 132(1):201–216, 2016.

- [57] R. Jentzsch. Untersuchungen zur theorie der folgen analytischer funktionen. *Acta Mathematica*, 41(1):219–251, 1916.
- [58] R. P. Kanwal. *Generalized Functions: Theory and Applications*. Birkhäuser, third edition, 2004.
- [59] L. J. Karam and J. H. McClellan. Complex Chebyshev approximation for FIR filter design. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(3):207–216, 1995.
- [60] Y. Katznelson. *An Introduction to Harmonic Analysis*. Cambridge University Press, third edition, 2004.
- [61] T. W. Körner. *Fourier Analysis*. Cambridge University Press, 1989.
- [62] H. L. Loeb. Approximation by generalized rationals. *SIAM Journal on Numerical Analysis*, 3(1):34–55, 1966.
- [63] H. J. Maehly. Methods for fitting rational approximations, Parts II and III. *Journal of the ACM*, 10(3):257–277, July 1963.
- [64] A. I. Markushevich and R. A. Silverman. *Theory of Functions of a Complex Variable*. American Mathematical Society, 2005.
- [65] M. McCallig. Design of digital FIR filters with complex conjugate pulse responses. *IEEE Transactions on Circuits and Systems*, 25(12):1103–1105, 1978.
- [66] G. Meinardus. *Approximation of Functions: Theory and Numerical Methods*. Springer, 1967.
- [67] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, Inc, second edition, 1999.
- [68] R. Pachón, P. Gonnet, and J. Van Deun. Fast and stable rational interpolation in roots of unity and Chebyshev points. *SIAM Journal on Numerical Analysis*, 50(3):1713–1734, 2012.
- [69] R. Pachón, R. B. Platte, and L. N. Trefethen. Piecewise-smooth chebfuns. *IMA journal of numerical analysis*, 30(4):898–916, 2010.
- [70] R. Pachón and L. N. Trefethen. Barycentric-Remez algorithms for best polynomial approximation in the chebfun system. *BIT Numerical Mathematics*, 49(4):721–741, 2009.
- [71] T. Parks and J. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Transactions on Circuit Theory*, 19(2):189–194, 1972.

- [72] S.-C. Pei and J.-J. Shyu. Design of complex FIR filters with arbitrary complex frequency responses by two real Chebyshev approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(2):170–174, 1997.
- [73] M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1981.
- [74] H. A. Priestley. *Introduction to Complex Analysis*. Oxford University Press, 2003.
- [75] L. Rabiner, J. H. McClellan, and T. W. Parks. FIR digital filter design techniques using weighted Chebyshev approximation. *Proceedings of the IEEE*, 63(4):595–610, 1975.
- [76] J. R. Rice. *The Approximation of Functions*, volume 1. Addison-Wesley Publishing Company, Inc., 1964.
- [77] J. R. Rice. *The Approximation of Functions*, volume 2. Addison-Wesley Publishing Company, Inc., 1969.
- [78] T. J. Rivlin. *An Introduction to the Approximation of Functions*. Blaisdell Publishing Company, 1969.
- [79] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Inc., New York, 1964.
- [80] W. Rudin. *Real and Complex Analysis*. McGraw-Hill Inc., New York, 3rd edition, 1986.
- [81] E. B. Saff and A. D. Snider. *Fundamentals of Complex Analysis*. Prentice-Hall, 2003.
- [82] H. E. Salzer. New formulas for trigonometric interpolation. *Journal of Mathematics and Physics*, 39(1-4):83–96, 1960.
- [83] H. Stahl. Best uniform rational approximation of  $|x|$  on  $[-1, 1]$ . *Matematicheskii Sbornik*, 183(8):85–118, 1992.
- [84] L. N. Trefethen. Near-circularity of the error curve in complex Chebyshev approximation. *Journal of Approximation Theory*, 31(4):344–367, 1981.
- [85] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.
- [86] L. N. Trefethen. Computing numerically with functions instead of numbers. *Communications of the ACM*, 58(10):91–97, September 2015.
- [87] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, 1997.
- [88] L. N. Trefethen and M. H. Gutknecht. The Carathéodory-Fejér method for real rational approximation. *SIAM Journal on Numerical Analysis*, 20(2):420–436, 1983.
- [89] L. N. Trefethen and M. H. Gutknecht. Padé, stable Padé, and Chebyshev-Padé approximation. In *Algorithms for Approximation*, pages 227–264. Clarendon Press, Oxford, 1987.

- [90] L. N. Trefethen and J. A. C. Weideman. Two results on polynomial interpolation in equally spaced points. *Journal of Approximation Theory*, 65(3):247–260, 1991.
- [91] L. N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.
- [92] A. H. Turetskii. The bounding of polynomials prescribed at equally distributed points. In *Proc. Pedag. Inst. Vitebsk*, volume 3, pages 117–127, 1940.
- [93] J. Van Deun and L. N. Trefethen. A robust implementation of the Carathéodory-Fejér method for rational approximation. *BIT Numerical Mathematics*, 51(4):1039–1050, 2011.
- [94] C. Van Loan. *Computational Frameworks for the Fast Fourier Transform*. SIAM, 1992.
- [95] A. Vretblad. *Fourier Analysis and its Applications*. Springer-Verlag, 2003.
- [96] J. L. Walsh. *Interpolation and Approximation by Rational Functions in the Complex Domain*. American Mathematical Society, 1935.
- [97] J. L. Walsh. The analogue for maximally convergent polynomials of Jentzsch’s theorem. *Duke Mathematical Journal*, 26(4):605–616, 1959.
- [98] G. A. Watson. *Approximation Theory and Numerical Methods*. Wiley, 1980.
- [99] E. Wegert. *Visual Complex Functions*. Birkhäuser, Basel, 2012.
- [100] J. A. C. Weideman. Numerical integration of periodic functions: A few examples. *The American Mathematical Monthly*, 109(1):21–36, 2002.
- [101] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis*. Cambridge University Press, fourth edition, 1927.
- [102] G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen. Extension of Chebfun to periodic functions. *SIAM Journal on Scientific Computing*, 37(5):C554–C573, 2015.
- [103] A. Zygmund. *Trigonometric Series*, volume I & II combined. Cambridge University Press, second edition, 1988.