

THE REMEZ ALGORITHM FOR TRIGONOMETRIC APPROXIMATION OF PERIODIC FUNCTIONS

MOHSIN JAVED^{*‡} AND LLOYD N. TREFETHEN^{†‡}

Abstract. In this paper we present an implementation of the Remez algorithm for trigonometric minimax approximation of periodic functions. There are software packages which implement the Remez algorithm for even periodic functions. However, we believe that this paper describes the first implementation for the general periodic case. Our algorithm uses Chebfun to compute with periodic functions. For each iteration of the Remez algorithm, to construct the approximation, we use the second kind barycentric trigonometric interpolation formula instead of the first kind formula. To locate the maximum of the absolute error, instead of dense sampling of the error function, we use Chebfun's eigenvalue based root finding method applied to the Chebyshev representation of the derivative of the underlying periodic function. Our algorithm has applications for designing FIR filters with real but asymmetric frequency responses.

Key words. Remez, Chebfun, trigonometric interpolation, best approximation, periodic functions, barycentric formula, root finding

AMS subject classifications. 41A50, 42A10, 42A15, 65T40

1. Introduction. The paper focusses on the classical problem of finding the minimax approximation of a real-valued periodic function in the space of trigonometric polynomials. 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 [10], which focusses on the familiar case of approximation by algebraic polynomials. As we shall see, after some variations, the same algorithm can be used to approximate continuous periodic functions by trigonometric polynomials.

After the advent of digital computers, the Remez algorithm became popular in the 1970s due to its applications in digital filter design [11]. 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 [11], 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 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 [7], [8]. In fact, we are unaware of any implementation of the Remez algorithm for this problem. This paper presents such an implementation in Chebfun.

Chebfun is an established tool for computing with functions [5]. Although it mainly uses Chebyshev polynomials to compute with non periodic functions, it has recently acquired trigonometric representations for periodic ones [16]. Our algorithm works in the setting of this part of Chebfun. Our three major contributions are as

^{*}javed@maths.ox.ac.uk

[†]trefethen@maths.ox.ac.uk

[‡]Oxford University Mathematical Institute, Oxford OX2 6GG, UK. This work is supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013)/ERC grant agreement no. 291068. The views expressed in this article 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.

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 [8], [12]. We use the second kind barycentric formula. For a detailed discussion of barycentric formulae, see [3].
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 [8], [11], [14]. Following [10], 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 [15, 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:

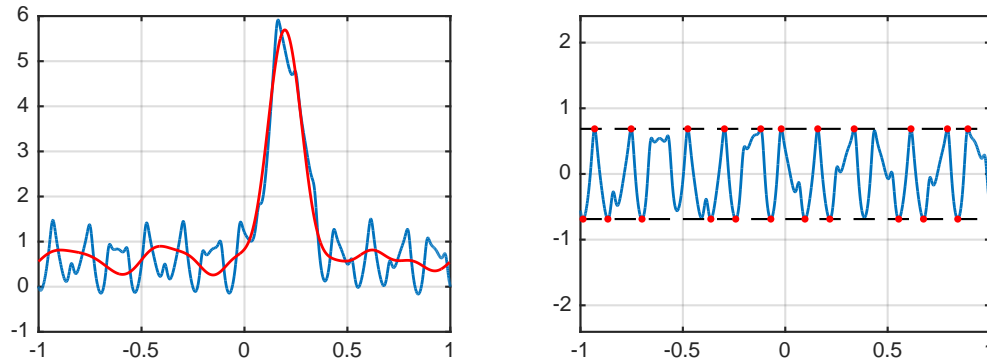


FIG. 1. The function $f(x) = 1/(2 + \sin 22\pi x) + (1/2) \cos 13\pi x + 5e^{-80(x-0.2)^2}$ and its degree 10 best trigonometric approximation (left). The error curve equioscillates between 22 extrema, marked by red dots (right).

```

>> 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)

```

The plot of the function f , its best approximation t_{10} and the resulting error curve are shown in Figure 1.

The outline of the paper 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 approximation. In Section 3, we present the theoretical background of the Remez algorithm for the trigonometric case. In Section 4 we present our algorithm and its implementation details. Section 5 consists of numerical examples. In Section 6 we conclude the paper while indicating future directions.

2. Trigonometric polynomials and best approximation. Before defining trigonometric polynomials, we define the domain on which these functions live, the quotient space

$$\mathbb{T} = \mathbb{R}/2\pi\mathbb{Z},$$

which is topologically equivalent to a circle. Loosely speaking, a function f defined on \mathbb{T} is a 2π -periodic function on \mathbb{R} . To be precise, we shall think of the function as defined on the interval $[0, 2\pi)$. A subset of $[0, 2\pi)$ will be considered open or closed if the corresponding set in \mathbb{T} is open or closed. Throughout this paper, we speak casually of the interval $[0, 2\pi)$ with the understanding that we are actually considering \mathbb{T} .

By a trigonometric polynomial of degree n , we mean a function $t : [0, 2\pi) \rightarrow \mathbb{R}$ of the form*:

$$(2.1) \quad t(\theta) = a_0 + \sum_{k=1}^n (a_k \cos k\theta + b_k \sin k\theta),$$

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

We shall denote the $2n + 1$ dimensional vector space of all trigonometric polynomials of degree n or less by \mathcal{T}_n . We note that the set $\{1, \cos \theta, \sin \theta, \dots, \cos n\theta, \sin n\theta\}$ is a basis for \mathcal{T}_n .

We are interested in finding the best approximation in the ∞ -norm,

$$(2.2) \quad \|t\|_{\infty} := \sup_{\theta \in [0, 2\pi)} |t(\theta)|.$$

The problem: Given a real-valued continuous function $f : \mathbb{T} \rightarrow \mathbb{R}$ and a non-negative integer n , find the function $t^* \in \mathcal{T}_n$ such that

$$(2.3) \quad \|f - t^*\|_{\infty} \leq \|f - t\|_{\infty}, \quad \forall t \in \mathcal{T}_n.$$

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

A best approximation exists, as can be proven easily; see for instance [13]. It is also unique. The proof of uniqueness relies on a property of the space \mathcal{T}_n which we now define.

2.1. The Haar condition. The Haar condition is a certain property of a finite dimensional vector space of functions. Let $C(\Omega)$ be the space of real-valued continuous functions on a compact set Ω 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 [13]. However, for the uniqueness of the best approximation, the Haar condition becomes important.

There are various ways of characterizing the Haar condition; see [9] and [13].

DEFINITION 1. *Let V be an n dimensional vector subspace of $C(\Omega)$. Then V satisfies the Haar condition if for every nonzero $t \in V$, the number of roots of t in Ω is strictly less than n .*

This is equivalent to the following [9]:

DEFINITION 2. *Let V be an n dimensional vector subspace of $C(\Omega)$ and let $\{\theta_k\}_{k=1}^n$ be any set of n distinct points in $[a, b]$. Then V satisfies the Haar condition if there is a unique function t in V which interpolates arbitrary data $\{f_i\}_{i=1}^n$, i.e. $t(\theta_i) = f_i$, $i = 1, 2, \dots, n$.*

Equivalently, V satisfies the Haar condition if for any basis $\{t_i : i = 1, 2, \dots, n\}$ of V and any distinct set of n points in Ω , the associated Vandermonde matrix is nonsingular.

THEOREM 3. *The space \mathcal{T}_n satisfies the Haar condition.*

Proof. We can prove this theorem by showing that any nonzero trigonometric polynomial $t \in \mathcal{T}_n$ has at most $2n$ roots in \mathbb{T} , i.e. in the interval $[0, 2\pi)$ [17, Chapter X]. We write t as

$$(2.4) \quad t(\theta) = a_0 + \sum_{k=1}^n (a_k \cos k\theta + b_k \sin k\theta) = \sum_{k=-n}^n c_k e^{ik\theta}.$$

Let $e^{i\theta} = z$ and consider the function

$$(2.5) \quad p(z) = e^{in\theta} t(\theta),$$

an algebraic polynomial of degree $2n$ in the variable z . The result now follows from the fundamental theorem of algebra. \square

It is interesting to note that the very similar vector spaces generated by the linear combinations of vectors in \mathcal{T}_n and just one of the functions $\cos(n+1)\theta$ or $\sin(n+1)\theta$ 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 θ_1 is arbitrary in $(0, \pi)$ and the second point θ_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.

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.

3. The Remez algorithm for periodic approximations: 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}_n of algebraic polynomials of degree n or less. For a given f , starting from an arbitrary initial condition, the algorithm converges to the best approximation of f in \mathcal{P}_n , and a proof of its convergence can be found in [13]. 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}_n .

Let us now look at the details of the algorithm. We shall closely follow [13], adapting its discussion of algebraic polynomials to our problem of trigonometric polynomials.

Let t^* be the best approximation of a continuous periodic function $f \in C([0, 2\pi))$ from the space \mathcal{T}_n . Let t_k be the k^{th} trial approximation in the iterative sequence and define the corresponding error at the k^{th} iteration as

$$(3.1) \quad e_k(\theta) = f(\theta) - t_k(\theta), \quad 0 \leq \theta < 2\pi.$$

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

$$(3.2) \quad E_k = \{\theta : \theta \in [0, 2\pi), \quad |e_k(\theta)| = \|e_k(\theta)\|_\infty\}.$$

3.1. A sufficient condition for optimality. We suppose that at the k^{th} step, t_k is not the best approximation. We can then write the best approximation t^* as

$$(3.3) \quad t^*(\theta) = t_k(\theta) + \alpha t(\theta),$$

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

$$(3.4) \quad e^* = f - t^* = f - t_k - \alpha t = e_k - \alpha t.$$

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

$$(3.5) \quad |e^*(\theta)| = |e_k(\theta) - \alpha t(\theta)| < |e_k(\theta)|, \quad \theta \in E_k.$$

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 $t \in \mathcal{T}_n$ which has the same sign as e_k at the points in E_k , i.e.

$$(3.6) \quad [f(\theta) - t_k(\theta)] t(\theta) > 0, \quad \theta \in E_k.$$

In other words, if there is no function $t \in \mathcal{T}_n$ for which (3.6) holds, then $t_k = t^*$.

3.2. A necessary condition for optimality. We now prove the converse of the above statement: if there exists a trigonometric polynomial $t \in \mathcal{T}_n$ for which the condition (3.6) holds, then t_k is not the best approximation, i.e. there exists a positive value of α such that

$$(3.7) \quad \|f - (t_k + \alpha t)\|_\infty < \|f - t_k\|_\infty.$$

We again closely follow [13]. Let E denote a closed subset of the interval $[0, 2\pi)$, which as always means more precisely a closed subset of \mathbb{T} . Also, let $\alpha > 0$ and without loss of generality, assume that $t \in \mathcal{T}_n$ satisfies the bound

$$(3.8) \quad |t(\theta)| \leq 1, \quad 0 \leq \theta < 2\pi.$$

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

$$(3.9) \quad E_o = \{\theta \in E : t(\theta)e_k(\theta) \leq 0\}.$$

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

$$(3.10) \quad d := \max_{\theta \in E_o} |e_k(\theta)|.$$

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

$$(3.11) \quad d < \max_{\theta \in E} |e_k(\theta)|.$$

We now prove that the inequality

$$(3.12) \quad \max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)| < \max_{\theta \in E} |f(\theta) - t_k(\theta)|$$

holds for

$$(3.13) \quad \alpha = \frac{1}{2} \left[\max_{\theta \in E} |e_k(\theta)| - d \right].$$

Since E is closed, we may let γ be an element of E such that

$$(3.14) \quad |f(\gamma) - t_k(\gamma) - \alpha t(\gamma)| = \max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)|.$$

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 (3.13).

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

$$(3.15) \quad \max_{\theta \in E} |f(\theta) - t_k(\theta) - \alpha t(\theta)| < \max\{|e_k(\gamma)|, |\alpha t(\gamma)|\} < \max_{\theta \in E} |e_k(\theta)|,$$

where in the last inequality, we have again used the definition of α from (3.13) and the fact that t is bounded by 1. Therefore, we have shown that (3.12) holds.

We summarize the last two conditions as the following theorem.

THEOREM 4. *Let E be a closed subset of $[0, 2\pi)$, i.e. closed in \mathbb{T} . Let t_k be any element of \mathcal{T}_n and let 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}_n that minimizes the expression*

$$(3.16) \quad \max_{\theta \in E} |f(\theta) - t(\theta)|, \quad t \in \mathcal{T}_n,$$

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

$$(3.17) \quad [f(\theta) - t_k(\theta)]t(\theta) > 0, \quad \theta \in E_k.$$

Proof. In the discussion above. \square

3.3. Characterization of the best approximation. Theorem 4 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 (3.6) holds for some function $t \in \mathcal{T}_n$. 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 (3.6) is rather easy to test. We make use of the fact that any function in \mathcal{T}_n has at most $2n$ sign changes[†] in \mathbb{T} . Therefore, if the error function changes sign more than $2n$ times as θ ranges in E_k , then t_k is the best approximation. Conversely, if the number of sign changes does not exceed $2n$, then we can choose the zeros of a trigonometric polynomial to construct t in \mathcal{T}_n so that the condition (3.6) is satisfied. This result is usually called the minimax characterization theorem [13] or the equioscillation theorem, which we state below.

THEOREM 5. (Equioscillation) *Let f be a continuous periodic function on $[0, 2\pi)$. Then t^* is the best approximation of f from \mathcal{T}_n if and only if there exist $2n + 2$ points $\{\theta_i, i = 1, 2, \dots, 2n + 2\}$ such that the following conditions hold:*

$$(3.18) \quad 0 \leq \theta_1 < \theta_2 < \dots < \theta_{2n+2} < 2\pi,$$

$$(3.19) \quad |f(\theta_i) - t^*(\theta_i)| = \|f - t^*\|_\infty, \quad i = 1, 2, \dots, 2n + 2,$$

and

$$(3.20) \quad f(\theta_{i+1}) - t^*(\theta_{i+1}) = -[f(\theta_i) - t^*(\theta_i)], \quad i = 1, 2, \dots, 2n + 1.$$

Proof. See [4] or [13]. \square

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 6. *Let f be a continuous periodic function on $[0, 2\pi)$ and let $0 \leq \theta_1 < \theta_2 < \dots < \theta_{2n+2} < 2\pi$ be $2n + 2$ points in $[0, 2\pi)$. Then $t^* \in \mathcal{T}_n$ minimizes the expression*

$$(3.21) \quad \max_i |f(\theta_i) - t(\theta_i)|, \quad t \in \mathcal{T}_n,$$

[†]For example the function $\sin(\theta)$ has exactly 2 sign changes in \mathbb{T} , one at $\theta = 0$ and another at $\theta = \pi$.

if and only if

$$(3.22) \quad f(\theta_{i+1}) - t^*(\theta_{i+1}) = -[f(\theta_i) - t^*(\theta_i)], \quad i = 1, 2, \dots, 2n+1.$$

Proof. See [13]. \square

THEOREM 7. *Let the conditions of Theorem 5 hold and let t^* be any element of \mathcal{T}_n such that the conditions*

$$(3.23) \quad \text{sign}[f(\theta_{i+1}) - t^*(\theta_{i+1})] = -\text{sign}[f(\theta_i) - t^*(\theta_i)], \quad i = 1, 2, \dots, 2n+1,$$

are satisfied. Then the inequalities

$$(3.24) \quad \min_i |f(\theta_i) - t^*(\theta_i)| \leq \min_{t \in \mathcal{T}_n} \max_i |f(\theta_i) - t(\theta_i)| \leq \min_{t \in \mathcal{T}_n} \|f - t\|_\infty \leq \|f - t^*\|_\infty$$

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

Proof. See [13]. \square

The next theorem is contained in Theorem 7. 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 8. (*de la Vallée Poussin*) *Let $t_k \in \mathcal{T}_n$ and $0 \leq \theta_1 < \theta_2 < \dots < \theta_{2n+2} < 2\pi$ be $2n+2$ points in $[0, 2\pi)$ such that the sign of the error $f(\theta_i) - t_k(\theta_i)$ alternates as i varies from 1 to $2n+2$. Then, for every $t \in \mathcal{T}_n$,*

$$(3.25) \quad \min_i |f(\theta_i) - t_k(\theta_i)| \leq \max_i |f(\theta_i) - t(\theta_i)|$$

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

The above theorem tells us that if we construct $t_k \in \mathcal{T}_n$ 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

$$(3.26) \quad \min_i |f(\theta_i) - t_k(\theta_i)| \leq \max_i |f(\theta_i) - t^*(\theta_i)| \leq \|f - t^*\|_\infty \leq \|f - t_k\|_\infty.$$

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

$$(3.27) \quad \|f - t_k\|_\infty \leq \left(\frac{\|f - t_k\|_\infty}{\min_i |f(\theta_i) - t_k(\theta_i)|} \right) \|f - t^*\|_\infty.$$

4. The Remez (exchange) algorithm for periodic functions: Implementation details. The Remez algorithm, at each iteration, finds a set of points $R_k = \{\theta_i, i = 1, 2, \dots, 2n+2\}$ and a trigonometric polynomial $t_k \in \mathcal{T}_n$ such that the conditions of Theorem 6 are satisfied. We will call the set R_k the k^{th} reference.

To start the algorithm, an initial reference R_1 is chosen. This can be any set of points that satisfies:

$$(4.1) \quad 0 \leq \theta_1 < \theta_2 < \cdots < \theta_{2n+2} < 2\pi.$$

The exchange algorithm is guaranteed to converge from any starting reference [13]. 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 $2n + 2$ equally spaced points in $[0, 2\pi)$.

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

$$(4.2) \quad \max_{i=1,2,\dots,2n+2} |f(\theta_i) - t_k(\theta_i)|, \quad t_k \in \mathcal{T}_n.$$

Theorem 6 shows that this can be done by solving the linear system

$$(4.3) \quad f(\theta_i) - t_k(\theta_i) = (-1)^{i-1} h_k, \quad i = 1, 2, \dots, 2n + 2,$$

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 $2n + 2$ unknown parameters: the unknown h_k and the $2n + 1$ unknown coefficients of the trigonometric polynomial t_k . It turns out the levelled reference error h_k can be found explicitly without solving the linear system (see Appendix). Once this is done, we can re-write (4.3) as an interpolation problem:

$$(4.4) \quad t_k(\theta_i) = f(\theta_i) + (-1)^{i-1} h_k, \quad i = 1, 2, \dots, 2n + 2,$$

At a first look, it seems that this problem is overdetermined: it has $2n + 2$ data points to interpolate and only $2n + 1$ 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 θ_i from (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 trigonometric barycentric formula of the second kind [2], [3]:

$$(4.5) \quad t_k(\theta) = \frac{\sum_{i=1}^{2n+1} w_i \csc \frac{1}{2} (\theta - \theta_i) t_k(\theta_i)}{\sum_{i=1}^{2n+1} w_i \csc \frac{1}{2} (\theta - \theta_i)},$$

where

$$(4.6) \quad w_i^{-1} = \prod_{j=1, j \neq i}^{2n+1} \sin \frac{1}{2} (\theta_i - \theta_j).$$

The formula (4.5) is implemented in Chebfun [16] 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*n+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*n+1, 'trig');

```

4.2. Finding the new reference. The next step of the algorithm is to find a new reference R_{k+1} . From Theorem 7, we see that the following bound holds:

$$(4.7) \quad |h_k| \leq \|f - t^*\|_\infty \leq \|f - t_k\|_\infty.$$

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 [13], 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 $R_{k+1} = \{\zeta_i, i = 1, 2, \dots, 2n+2\}$, and let the new trial approximation be t_{k+1} . To ensure $|h_{k+1}| > |h_k|$, Theorem 7 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 $|h_k|$:

$$(4.8) \quad \text{sign}[f(\zeta_{i+1}) - t_k(\zeta_{i+1})] = -\text{sign}[f(\zeta_i) - t_k(\zeta_i)], \quad i = 1, 2, \dots, 2n+1,$$

and

$$(4.9) \quad |h_k| \leq |f(\zeta_i) - t_k(\zeta_i)|, \quad i = 1, 2, \dots, 2n+2,$$

where at least one of the inequalities is strict. If the above conditions are satisfied, then it follows from Theorem 8 that

$$(4.10) \quad |h_k| \leq \min_i |f(\zeta_i) - t_k(\zeta_i)| < \max_i |f(\zeta_i) - t_{k+1}(\zeta_i)| = |h_{k+1}|.$$

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 (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 (4.8) and (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 paper, we will not discuss the convergence properties of exchange algorithms; see [13].

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 [10], 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 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);
```

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 $O(n^3)$ algorithm [16]. To use Chebfun's $O(n^2)$ recursive interval subdivision strategy root finding algorithm, the function is first converted to a nonperiodic 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 [15].

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 [1], [15].

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 $2n+2$ 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 $|h_k|$.

5. Numerical examples. Let us now look at some examples. The Chebfun command `trigremez(f, n)` finds the degree `n` best approximation of a periodic continuous chebfun `f` on its underlying domain `[a, b]`:

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

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 2.

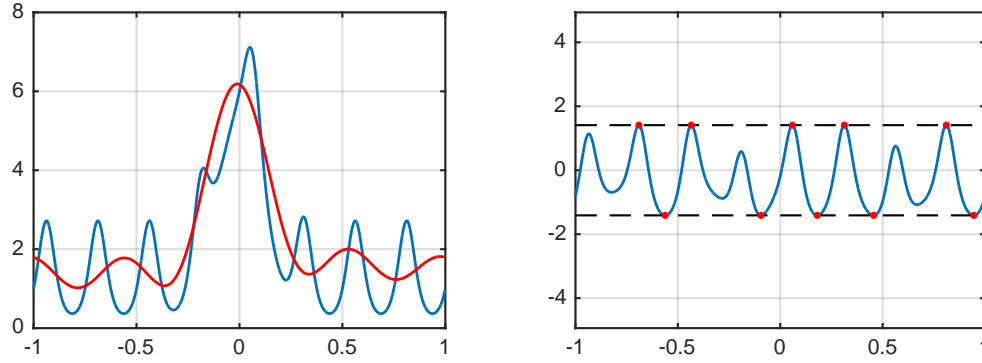


FIG. 2. The function $f(x) = e^{\sin 8\pi x} + 5e^{-40x^2}$, its degree 4 best approximation (left) and the equioscillating error curve (right).

An interesting feature of `trigremez` is that we can use it with nonperiodic as well as periodic chebuns. As long as a function is continuous in the periodic sense, i.e. the function values are the same at the endpoints, the theorems stated in the previous sections hold and it does not matter how whether we represent it as a Chebyshev (aperiodic) or a Fourier (periodic) expansion.

Here is a modified version of an example of a zig-zag function taken from [15]. To make the function continuous across the periodic boundary, we have subtracted off an appropriate linear term:

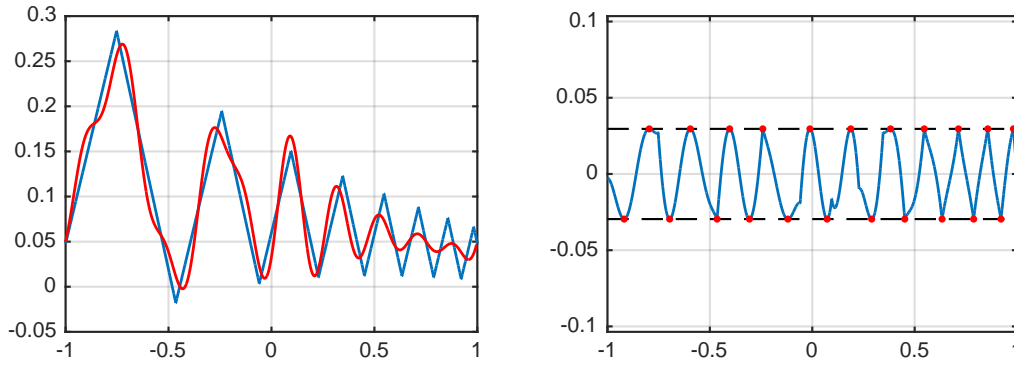


FIG. 3. Degree 10 best approximation of a zig-zag function and the corresponding error curve.

```
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.

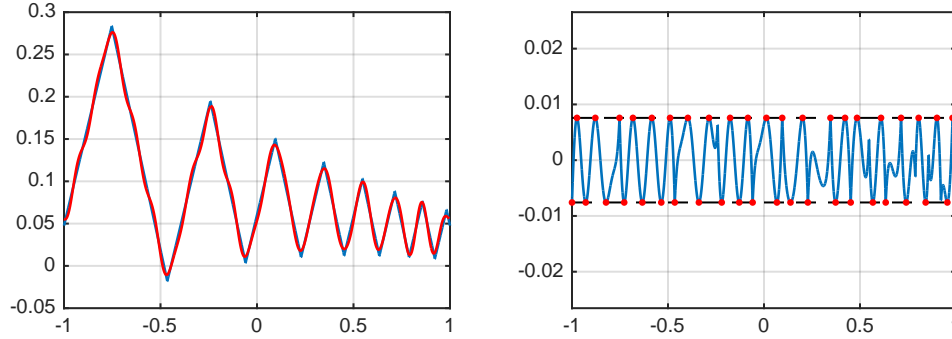


FIG. 4. Degree 20 approximation of the same function.

Figure 4 is for the same problem but with a higher degree of approximation.

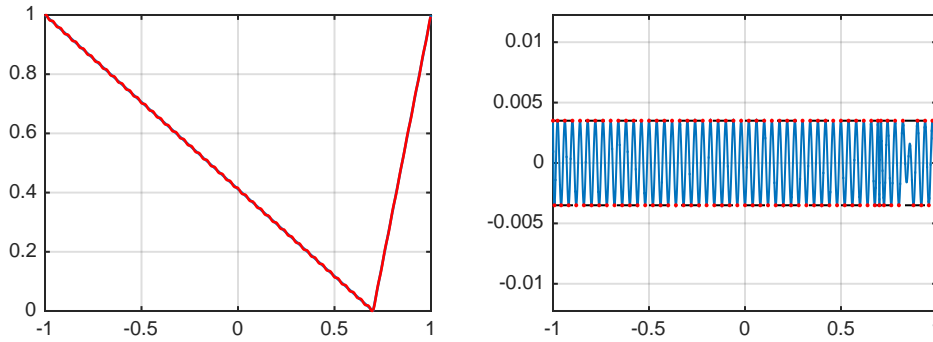


FIG. 5. The shifted absolute value function and its degree 50 best approximation. The corresponding error curve is also shown.

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 5.

It is well known that for functions analytic on $[0, 2\pi]$, the best approximation error decreases exponentially at the rate of $O(e^{-\alpha n})$, where n is the degree of the

best approximation and α is the distance of the real line from the nearest singularity of the function [16]. (More precisely we get $O(e^{-\alpha n})$ if f is bounded in the strip of analyticity defined by the nearest singularity. If not, we still have $O(e^{-(\alpha-\epsilon)n})$ for any $\epsilon > 0$, and there is no need for the ϵ if the closest singularities are just simple poles.) As an example, we consider the function $f(\theta) = (b + \cos \theta)^{-1}$ which has poles in the complex θ -plane along the line $\theta = 1$. 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 6, we see that the numerical decay of the error matches the predicted decay beautifully.

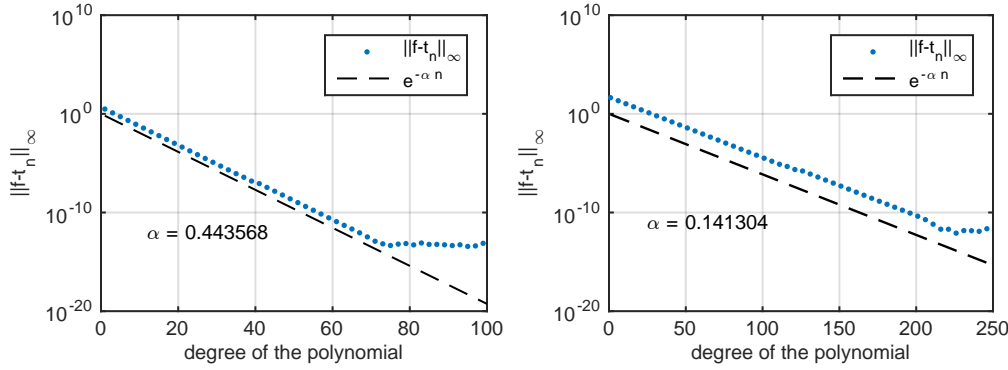


FIG. 6. Error in the best approximation of $f(\theta) = (b + \cos \theta)^{-1}$ as a function of the degree of the approximating polynomial. $b = 1.1$ for the plot on the left, $b = 1.01$ for the plot on the right.

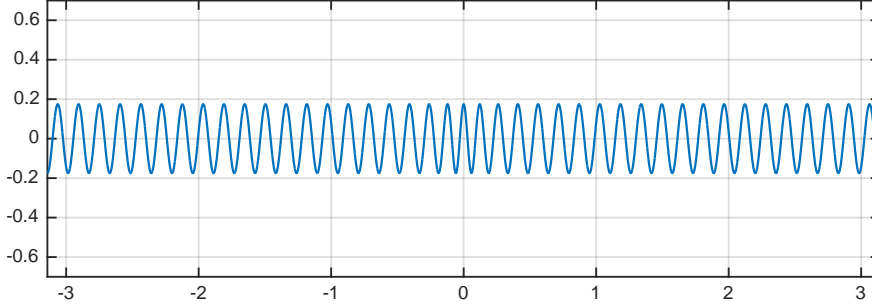


FIG. 7. Almost sinusoidal error curve produced while approximating $f(\theta) = (1.01 - \cos \theta)^{-1}$ by a degree 40 trigonometric polynomial.

Best approximations in spaces of algebraic polynomials yield error curves that look approximately like Chebyshev polynomials, and indeed, there are theorems to the effect that for functions satisfying appropriate smoothness conditions, the error curves approach multiples of Chebyshev polynomials as the degree approaches infinity. In trigonometric rather than algebraic best approximation, however, the error curves tend to look like sine waves, not Chebyshev polynomials. To the experienced eye, this can be quite a surprise. The following example illustrates this, and the almost sinusoidal error curve can be seen in Figure 7.

```
f = chebfun('1./(1.01-cos(x))', [-pi, pi], 'trig');
plot(f-trigremez(f, 40)), ylim([-1 1])
```

6. Conclusion and future work. In this paper, we have presented a Remez algorithm for finding the minimax approximation of a periodic function in a space of trigonometric polynomials. Two key steps are trigonometric interpolation and determining the extrema of the error function. We use the second kind barycentric formula for the former and Chebfun's root finding algorithm for the later. The algorithm is already a part of Chebfun[‡] and perhaps more than applications, its greatest use is for fundamental studies in approximation theory.

Our algorithm can easily be modified to design digital filters with real but asymmetric frequency responses, see [8]. 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 [15], based on singular values of a Hankel matrix of Chebyshev coefficients, which has been used to compute near-best approximations of algebraic polynomials [6]. We are working on a CF algorithm for periodic functions based on the singular values of a matrix of Fourier coefficients.

Appendix. Formula for the levelled reference error h_k . Let us again consider the linear system (4.3):

$$(A.1) \quad f(\theta_j) - t_k(\theta_j) = (-1)^{j-1} h_k, \quad j = 1, 2, \dots, 2n+2,$$

where the points $\theta_j \in [0, 2\pi)$ are distinct. The polynomial t_k is of degree n , which implies that there are $2n+1$ unknown parameters needed to completely determine t_k . However, the linear system consists of $2n+2$ equations with $2n+1$ 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 n trigonometric polynomial t_k in a trigonometric Lagrangian basis. Since there is one extra collocation point, say θ_j , we may define the j^{th} set of basis functions β_j as

$$(A.2) \quad \beta_j = \left\{ l_i^j \in \mathcal{T}_n, i = 1, 2, \dots, j-1, j+1, \dots, 2n+2. \right\},$$

where

$$(A.3) \quad l_i^j(\theta) := \prod_{\nu=1, \nu \neq i, j}^{2n+2} \frac{\sin \frac{1}{2}(\theta - \theta_\nu)}{\sin \frac{1}{2}(\theta_i - \theta_\nu)}.$$

Note that the function $l_i^j(\theta)$ is a degree n 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 $j \in \{1, 2, \dots, 2n+2\}$, the polynomial t_k can be expressed as

$$(A.4) \quad t_k(\theta) = \sum_{i=1, i \neq j}^{2n+2} t_k(\theta_i) l_i^j(\theta)$$

[‡]To download the latest version of Chebfun, visit www.chebfun.org.

Using the above expression to evaluate t_k at θ_j , we can write the original linear system (A.1) as:

$$(A.5) \quad f(\theta_j) - \sum_{i=1, i \neq j}^{2n+2} t_k(\theta_i) l_i^j(\theta_j) = (-1)^{j-1} h_k, \quad j = 1, 2, \dots, 2n+2.$$

We can simplify the number $l_i^j(\theta)$ by writing

$$(A.6) \quad l_i^j(\theta_j) = \prod_{\nu=1, \nu \neq i, j}^{2n+2} \frac{\sin \frac{1}{2}(\theta_j - \theta_\nu)}{\sin \frac{1}{2}(\theta_i - \theta_\nu)}$$

$$(A.7) \quad = \frac{\prod_{\nu=1, \nu \neq j}^{2n+2} \sin \frac{1}{2}(\theta_j - \theta_\nu)}{\prod_{\nu=1, \nu \neq i}^{2n+2} \sin \frac{1}{2}(\theta_i - \theta_\nu)} \times \frac{\sin \frac{1}{2}(\theta_i - \theta_j)}{\sin \frac{1}{2}(\theta_j - \theta_i)} = -\frac{w_i}{w_j},$$

where

$$(A.8) \quad w_j^{-1} = \prod_{\nu=1, \nu \neq j}^{2n+2} \sin \frac{1}{2}(\theta_j - \theta_\nu).$$

This allows us to write (A.5) as

$$(A.9) \quad f(\theta_j) w_j - \sum_{i=1, i \neq j}^{2n+2} t_k(\theta_i) w_i = (-1)^{j-1} w_j h_k, \quad j = 1, 2, \dots, 2n+2.$$

Now summing over j we get

$$(A.10) \quad \sum_{j=1}^{2n+2} f(\theta_j) w_j - \sum_{j=1}^{2n+2} \sum_{i=1, i \neq j}^{2n+2} t_k(\theta_i) w_i = \sum_{j=1}^{2n+2} (-1)^{j-1} w_j h_k.$$

The double sum above collapses to zero when we use the fact that

$$(A.11) \quad \sum_{i=1}^{2n+2} t_k(\theta_i) w_i = 0.$$

We can therefore rearrange (A.10) to write h_k as

$$(A.12) \quad h_k = \frac{\sum_{j=1}^{2n+2} w_j f(\theta_j)}{\sum_{j=1}^{2n+2} (-1)^{j-1} w_j}.$$

which is the required closed form expression.

REFERENCES

- [1] Z. BATTLES AND L. N. TREFETHEN, *An extension of MATLAB to continuous functions and operators*, SIAM Journal on Scientific Computing, 25 (2004), pp. 1743–1770.
- [2] J.-P. BERRUT, *Baryzentrische Formeln zur trigonometrischen Interpolation (i)*, Zeitschrift für angewandte Mathematik und Physik, 35 (1984), pp. 91–105.
- [3] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Review, 46 (2004), pp. 501–517.

- [4] E. W. CHENEY, *Introduction to Approximation Theory*, AMS Chelsea Publishing, Providence, Rhode Island, 1998.
- [5] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.
- [6] M. H. GUTKNECHT AND L. N. TREFETHEN, *Real polynomial Chebyshev approximation by the Carathéodory-Fejér method*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 358–371.
- [7] 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 (1995), pp. 207–216.
- [8] M. MCCALLIG, *Design of digital FIR filters with complex conjugate pulse responses*, IEEE Transactions on Circuits and Systems, 25 (1978), pp. 1103–1105.
- [9] G. MEINARDUS, *Approximation of Functions: Theory and Numerical Methods*, Springer, 1967.
- [10] R. PACHÓN AND L. N. TREFETHEN, *Barycentric-Remez algorithms for best polynomial approximation in the chebfun system*, BIT Numerical Mathematics, 49 (2009), pp. 721–741.
- [11] T. PARKS AND J. MCCLELLAN, *Chebyshev approximation for nonrecursive digital filters with linear phase*, IEEE Transactions on Circuit Theory, 19 (1972), pp. 189–194.
- [12] 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 (1997), pp. 170–174.
- [13] M. J. D. POWELL, *Approximation Theory and Methods*, Cambridge University Press, 1981.
- [14] L. RABINER, J. H. MCCLELLAN, AND T. W. PARKS, *FIR digital filter design techniques using weighted Chebyshev approximation*, Proceedings of the IEEE, 63 (1975), pp. 595–610.
- [15] L. N. TREFETHEN, *Approximation Theory and Approximation Practice*, SIAM, 2013.
- [16] G. B. WRIGHT, M. JAVED, H. MONTANELLI, AND L. N. TREFETHEN, *Extension of Chebfun to periodic functions*, SIAM J. Sci. Comp., (submitted).
- [17] A. ZYGMUND, *Trigonometric Series*, vol. I & II combined, Cambridge University Press, second ed., 1988.