university of groningen / faculty of science and engineering / mathematics and applied mathematics

Finding best minimax approximations with the Remez algorithm

Bachelor's Project Mathematics

October 2017

Student: E.D. de Groot

First supervisor: Dr. A.E. Sterk

Second assessor: Prof. Dr. H.L. Trentelman

Abstract

The Remez algorithm is an iterative procedure which can be used to find best polynomial approximations in the minimax sense. We present and explain relevant theory on minimax approximation. After doing so, we state the Remez algorithm and give several examples created by our Matlab implementation of the algorithm. We conclude by presenting a convergence proof.

Contents

1 Introduction

Roughly speaking, approximation theory is a branch of mathematics which deals with the problem of approximating a given function by some simpler class of functions. The general formulation of the best approximation problem can be given as follows. Given a subset Y of a normed linear space X and an element $x \in X$, can we find an element $y^* \in Y$ that is closest to x? That is, can we find a $y^* \in Y$ so that

$$||x - y^*|| \leq ||x - y||$$

for all $y \in Y$? Several questions can be asked.

- Under what conditions on Y does a best approximation exist?
- If it exists, is it unique?
- If it exists, how can we find it?
- What happens if we choose a different norm?

Answers to the questions formulated above can be found in standard books on approximation theory, such as [1] or the classical book by Cheney [2]. Another overview of different topics in approximation theory can be found in [3].

In some cases, the best approximation problem has a relatively simple solution; if the norm is induced by an inner product, the following recipe can be followed to find a best approximation to an element f in an inner product space X. Let $Y \subset X$ be a finite dimensional subspace and assume without loss of generality that $\{h_1, \ldots, h_n\}$ is an orthonormal basis for Y. Otherwise we can apply the Gram-Schmidt procedure to obtain an orthonormal basis. The following theorem now tells us how to find a best approximation to f.

Theorem 1. *Let $\{h_1, \ldots, h_n\}$ be an orthonormal set in an inner product space with norm defined by $||h|| = \langle h, h\rangle^{1/2}$. The expression $||\sum_{i=1}^{n} c_i h_i - f||$ is a minimum if and only if $c_i = \langle f, h_i\rangle$ for $i = 1, \ldots, n$* [2, p. 20].

One could say that the field approximation theory started in the year of 1853, when the famous Russian mathematician P.L. Chebyshev was working on the problem of translating the linear motion of a steam engine to the circular motion of a wheel [3, p. 1]. He formulated the following problem. Let $\mathbb{P}_n$ be the space of polynomials of degree $\leq n$. Given a continuous function $f : [a, b] \to \mathbb{R}$ and $n \in \mathbb{N}$, find a polynomial $p^* \in \mathbb{P}_n$ so that for all other polynomials $p \in \mathbb{P}_n$,

$$||f - p^*||_\infty \leq ||f - p||_\infty.$$

In this case, we call p^* the best *minimax approximation* to f from the set $\mathbb{P}_n$, where we denote by $||\cdot||_\infty$ the uniform norm, defined as

$$||g||_\infty = \max_{x \in [a,b]} |g(x)|$$

for a continuous function $g : [a, b] \to \mathbb{R}$. The existence of a point in $[a, b]$ maximizing $|f(x)|$ for $f \in C[a, b]$ is guaranteed by the following theorem, which is often taught in introductory courses on metric spaces.

Theorem 2. *A continuous real-valued function defined on a compact set in a metric space achieves its infimum and supremum on that set.*

Unfortunately, the recipe described earlier for finding best approximations in an inner product space is not applicable to the minimax approximation problem; the uniform norm does not satisfy the *parallellogram law* and is therefore not induced by an inner product [4, p. 29]. It is furthermore known that for finding best minimax approximations, we need to rely on iterative procedures [5, chapter 2]. This makes the problem of finding the best minimax approximation, in certain sense, a more difficult one than the problem of finding the best approximation in an inner product space. An important question here is the following.

- How can we find the best minimax approximation in practice?

In this thesis, we study the solution for the minimax approximation problem formulated earlier by the *Remez algorithm.*

Approximation theory has both a very applicable side, for example involving approximation algorithms which are used in industry and in science. On the other hand, there is a highly theoretical side, studying problems such as existence, uniqueness and characterization of best approximations [1, preface]. The present thesis is a mixture of both sides, focusing on theory behind minimax approximation, as well as on applying the theory on examples through the Remez algorithm.

The theory on minimax approximation presented in this thesis applies not only to minimax approximation by polynomials of some fixed degree, but is more general and considers approximation by *generalized polynomials.* A generalized polynomial p is a function of the form

$$p(x) = \sum_{i=1}^{n} c_i g_i(x),$$

where $c_1, \ldots, c_n$ are scalars and $g_1, \ldots, g_n$ are continuous functions. Generally we will require the system of functions $\{g_1, \ldots, g_n\}$ to satisfy the *Haar condition*, which we define in section 4. Examples of systems of functions satisfying the Haar condition can be found in [6] and [7]. An important system of functions which satisfies the Haar condition is $\{1, x, \ldots, x^n\}$, making our theory applicable to approximation by ordinary polynomials. Existence of a best approximation by a generalized polynomial is guaranteed by the following theorem, which students may recognize from functional analysis.

Theorem 3. *(Existence theorem) Let X be a normed linear space and let Y be a finite dimensional subspace of X. Then, for all $x \in X$ there is an element $y^* \in Y$ such that $||x - y^*|| = \inf_{y \in Y} ||x - y||$* [2, p. 20].

Moreover, under the Haar condition, the best approximation is unique.

Theorem 4. *(Haar's unicity theorem) The best approximation to a function $f \in C[a, b]$ is unique for all choices of f if and only if the system of continuous functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition* [2, p. 81].

The Remez algorithm, introduced by the Russian mathematician Evgeny Yakovlevich Remez in 1934 [5, section 1], is an iterative procedure which converges to the best minimax approximation of a given continuous function on the interval $[a, b]$ by a generalized polynomial $p = \sum_{i=1}^{n} c_i g_i$. The system $\{g_1, \ldots, g_n\}$ is part of the input and must be subject to the Haar condition. Today, the Remez algorithm has applications in filter design, see for example [8]. The statement of the algorithm and its convergence properties can be found in several books on approximation theory, for example [1, 2, 9]. Underlying theory on minimax approximation is treated in these books as well.

The main idea behind the Remez algorithm is based on the *alternation theorem*, to which section 4 is devoted. The alternation theorem provides us with a method to directly calculate the best minimax approximation on a *reference*, which is a discrete subset of $[a, b]$. In each iteration, the Remez algorithm computes the best minimax approximation on the reference obtained in the previous iteration and then adjusts the reference. The best minimax approximation on this new reference, computed in the next iteration, will then be a better approximation on the whole interval $[a, b]$. The initial reference is part of the input and may be chosen freely.

Computing the best minimax approximation on the reference is computationally not a difficult task; a corollary of the alternation theorem shows us that this is done by solving a linear system in n equations and n unknowns. In other steps in the algorithm however, we will need to calculate several local extrema of the *residual* function

$$r(x) = f(x) - p(x),$$

where f is the approximant and p is the best approximation on the reference. The residual function is not guaranteed to be differentiable and, moreover, will usually have many extrema. For fast computations, it is necessary to approximate the positions of these extrema in an efficient way. In [1, p. 86], the author suggests interpolating the residual function locally by a quadratic function to approximate the positions of local extrema.

The authors in [5] report high degrees of efficiency for computing best approximations, using the Remez algorithm as part of the chebfun software package. An important feature of this implementation is the use of the *Barycentric Lagrange interpolation formula*, which, as the authors claim in [10], deserves to be known as the standard method of polynomial interpolation. Furthermore, explicit examples of best approximations in the minimax sense can be found in [5] and [11].

This thesis is focused on explaining the theory behind the Remez algorithm and on creating examples using our own Matlab implementation of it. We will gives answers to the following questions.

- How does the Remez algorithm work?
- How well does our own implementation of the Remez algorithm perform? What are its limitations? Why?

Sections 2 and 3 treat the theory enabling us to present a proof the alternation theorem in section 4. Our main sources in this part are [1] and [2]. The alternation theorem allows us to understand the underlying mechanism of the Remez algorithm, which is stated in section 5. A highly efficient implementation of the Remez algorithm already exists and is included in the chebfun package. We will make a relatively simple implementation of the algorithm, test it in several examples and create tables to discuss its performance. Moreover, we will make plots clarifying how the algorithm works. Finally, we slightly extend the theory presented in the first sections and present a proof of the convergence of the Remez algorithm.

2 Convexity

Our goal in this section is to define convexity for linear spaces and to prove several theorems about convex sets. Results in this section are used in proofs of the *characterization theorem* and the *alternation theorem* in sections 3 and 4, respectively.

2.1 Definitions

Definition 1. *A subset A of a linear space is said to be convex if $f, g \in A$ implies that $\theta f + (1-\theta)g \in A$ for all $\theta \in [0,1]$.*

Intuitively this means that a subset A of a linear space is convex if, given any $a, b \in A$, the line segment joining a and b is contained in A as well. Notice that in the case $A = \mathbb{R}^2$, the line segment joining a and b consists precisely of all points $\theta a + (1-\theta)b$ with $\theta \in [0,1]$.

Definition 2. *Let A be a subset of a linear space. The convex hull $H(A)$ of A is the set consisting of all finite sums of the form $g = \sum \theta_i f_i$ such that $f_i \in A$, $\sum \theta_i = 1$ and $\theta_i \geq 0$. Sums in this form are called convex linear combinations.*

Let A be a subset of a linear space. Observe that for $a, b \in H(A)$ and $\theta \in [0,1]$, we have that $\theta a + (1-\theta)b$ is contained in $H(A)$, which shows that the convex hull of any subset of a linear space is convex, justifying the name. We conclude this subsection with the following observation.

Observation 1. *Assume that $A \subset B$ are subsets of a linear space. Let $a \in H(A)$. Then we can write*

$$a = \sum \theta_i a_i$$

with $a_i \in A$ and $\sum \theta_i = 1$. Because $a_i \in B$ for all i, it directly follows that $a \in H(B)$. This shows that $H(A) \subset H(B)$.

2.2 Results

Theorem 5. *(Carathéodory) Let A be a subset of an n-dimensional linear space. Every point in the convex hull of A can be expressed as a convex linear combination of no more than $n+1$ elements of A.*

Proof. Let $g \in H(A)$. Then, by definition of $H(A)$, we may write $g = \sum_{i=0}^{k} \theta_i f_i$ with $\sum_{i=0}^{k} \theta_i = 1$, $\theta_i \geq 0$ and $f_i \in A$ for all $i = 0, 1, \ldots, k$. Assume k is as small as possible. Then all the θ_i are nonzero; otherwise we could exclude the zero term, contradicting the minimality of k. The set $G = \{g_i = f_i - g : 0 \leq i \leq k\}$ is dependent since

$$\sum_{i=0}^{k} \theta_i g_i = \sum_{i=0}^{k} \theta_i (f_i - g) = g - g \sum_{i=0}^{k} \theta_i = 0.$$

Assume for contradiction that $k > n$. Then $G \setminus \{g_0\} = \{g_1, \ldots, g_k\}$ must be dependent because our linear space has dimension n by assumption. Because of the dependence, there exist scalars $\alpha_1, \ldots, \alpha_k$ such that $\sum_{i=1}^{k} \alpha_i g_i = 0$ and $\sum_{i=1}^{k} |\alpha_i| \neq 0$. Defining $\alpha_0 = 0$, we find that $\sum_{i=0}^{k} (\theta_i + \lambda \alpha_i) g_i = 0$ for all λ. Choose λ with the smallest possible absolute value such that one of the coefficients $\theta_i + \lambda \alpha_i$ vanishes, dropping one term of the sum $\sum_{i=0}^{k} (\theta_i + \lambda \alpha_i) g_i$. The other coefficients cannot become negative because λ was chosen with $|\lambda|$ small enough. Moreover, $\theta_0 + \lambda \alpha_0 = \theta_0 > 0$, hence not all coefficients vanish. Replacing g_i by $f_i - g$ gives us

$$0 = \sum_{i=0}^{k} (\theta_i + \lambda \alpha_i) g_i = \sum_{i=0}^{k} (\theta_i + \lambda \alpha_i)(f_i - g),$$

so that $g \sum_{i=0}^{k} (\theta_i + \lambda \alpha_i) = \sum_{i=0}^{k} (\theta_i + \lambda \alpha_i) f_i$. Our last step is to divide both sides of this last equality by $\sum_{i=0}^{k} (\theta_i + \lambda \alpha_i)$. Because one term in this last sum is zero, we have now expressed g using no more than k terms, contradicting minimality of k. Hence the assumption that $k > n$ is wrong and the conclusion of the theorem follows. □

In the proof of the next corollary, we make use of the following theorem from functional analysis.

Theorem 6. *All norms on a finite dimensional linear space are equivalent.*

Corollary 1. *The convex hull of a compact set is compact.*

Proof. We first prove that the set $B = \{(\theta_0, \ldots, \theta_n) : \theta_i \geq 0, \sum \theta_i = 1\}$ is compact being a closed and bounded subset of $\mathbb{R}^{n+1}$; first note that the set is bounded because the l^1 norm of each element is 1.

For showing that B is closed, let $(\theta_0^m, \ldots, \theta_n^m) \in B$ for all integers m and assume

$$\lim_{m \to \infty} (\theta_0^m, \ldots, \theta_n^m) = (\theta_0^*, \ldots, \theta_n^*).$$

Making again use of the fact that that all norms on finite dimensional space are equivalent, we may assume the sequence converges in the l^1 norm so that

$$\lim_{m\to\infty}\sum_{i=0}^{n}|\theta_i^m-\theta_i^*|=0.$$

This implies that for all i, $\lim_{m\to\infty}\theta_i^m=\theta_i^*\geq 0$. It is also clear that $\sum_{i=0}^n\theta_i^*=1$ because $\sum_{i=0}^n\theta_i^m=1$ for all m. This proves that the limit of the sequence is contained in B, so that B is closed. This proves that B is compact.

Now let X be a compact subset of a linear space with dimension n and let (v_k) be any sequence in $H(X)$. Our goal is to show that this sequence has a convergent subsequence with limit in $H(X)$. Apply theorem 5 to write $v_k=\sum_{i=0}^n\theta_{ki}x_{ki}$, where the x_{ki} belong to X. By compactness of the sets B and X, we can find a sequence (k_j) such that $\lim_{j\to\infty}\theta_{k_ji}:=\theta_i$ and $\lim_{j\to\infty}x_{k_ji}:=x_i\in X$ exist. From the previous part of the proof it is clear that $\theta_i\geq 0$ for all i and that $\sum_{i=0}^n\theta_i=1$. Thus (v_k) has a subsequence (v_{k_j}) converging to a limit in $H(X)$. This proves that $H(X)$ is compact. □

Theorem 7. *Every closed, convex subset of $\mathbb{R}^n$ has a unique point of minimum norm.*

Proof. Let $K\subset\mathbb{R}^n$ be closed and convex and let $d=\inf_{x\in K}||x||$. By definition of the infimum, there exists a sequence (x_n) in K such that $\lim_{n\to\infty}||x_n||=d$. We want to show that this sequence converges to a limit in K. To this end, apply the parallelogram law to write

$$||x_i-x_j||^2=2||x_i||^2+2||x_j||^2-4||\frac{1}{2}(x_i+x_j)||^2.$$

By convexity of K, the point $\frac{1}{2}(x_i+x_j)$ belongs to K as well. This implies that $||\frac{1}{2}(x_i+x_j)||\geq d$ so that

$$||x_i-x_j||^2\leq 2||x_i||^2+2||x_j||^2-4d^2.$$

We find that $\lim_{i,j\to\infty}||x_i-x_j||^2\leq 2d^2+2d^2-4d^2=0$ which shows that (x_n) is Cauchy, hence convergent in $\mathbb{R}^n$. The limit of this sequence is unique and is contained in K because K is closed. This completes the proof. □

For vectors $a=[a_1,\ldots,a_n]$ and $b=[b_1,\ldots,b_n]$ in $\mathbb{R}^n$, the standard inner product is given by $\langle a,b\rangle=\sum_{i=1}^n a_ib_i$.

Theorem 8. *(Theorem on linear inequalities) Let $U\subset\mathbb{R}^n$ be compact. For all $z\in\mathbb{R}^n$ there exists at least one $u\in U$ such that $\langle u,z\rangle\leq 0$ if and only if $\mathbf{0}\in H(U)$.*

Proof. ($\Longleftarrow$) Assume that $\mathbf{0}\in H(U)$. Then, by definition of $H(U)$, we can write $\mathbf{0}=\sum_{i=1}^m\theta_iu_i$ with $\theta_i\geq 0$, $\sum_{i=1}^m\theta_i=1$ and $u_i\in U$ for some positive integer m. For all $z\in\mathbb{R}^n$, $\sum_{i=1}^m\theta_i\langle u_i,z\rangle=\langle\sum_{i=1}^m\theta_iu_i,z\rangle=\langle\mathbf{0},z\rangle=0$. This cannot be true if $\langle u_i,z\rangle>0$ for all $i=1\ldots m$.

($\implies$) By contraposition. Assume $\mathbf{0} \notin H(U)$ and let $u \in U$ be arbitrary. Corollary 1 tells us that $H(U)$ is compact, hence it is closed. Now we apply theorem 7 to see that there is a $z \in H(U)$ such that $||z||$ is a minimum. Because $H(U)$ is convex, $\theta u + (1-\theta)z \in H(U)$ for $\theta \in [0,1]$. We apply the usual rules for expanding inner products to establish the following inequality:

$$\begin{aligned} 0 &\leq ||\theta u + (1-\theta)z||^2 - ||z||^2 \\ &= \langle \theta(u-z) + z, \theta(u-z) + z\rangle - \langle z, z\rangle \\ &= \langle \theta(u-z), \theta(u-z)\rangle + 2\langle \theta(u-z), z\rangle + \langle z, z\rangle - \langle z, z\rangle \\ &= \theta^2 ||u-z||^2 + 2\theta\langle u-z, z\rangle. \end{aligned}$$

The inequality above can only be true if $\langle u-z, z\rangle \geq 0$; if the last inner product were negative, we could choose θ small enough so that

$$\theta^2 ||u-z||^2 + 2\theta\langle u-z, z\rangle < 0.$$

Hence $\langle u-z, z\rangle = \langle u, z\rangle - \langle z, z\rangle > 0$ which implies $\langle u, z\rangle \geq \langle z, z\rangle > 0$, completing the proof. □

3 Characterization of the best polynomial approximation

As mentioned in the introduction, a well known problem in approximation theory is to find the best polynomial of degree n approximation to a function $f \in C[a,b]$ in the minimax sense. That is, we are interested in finding a polynomial P of degree n minimizing the quantity $\max_{x\in[a,b]} |f(x) - P(x)|$. The main result in this section however, the *characterization theorem*, discusses a somewhat more general setting, in which our degree n polynomial is replaced by a *generalized polynomial* $\sum_{i=1}^{n} c_i g_i(x)$. Here, $g_1, \ldots, g_n$ are continuous functions on the interval $[a,b]$.

A natural question to ask is whether a best approximation by a generalized polynomial always exists. Since the set of linear combinations of the functions $g_1, \ldots, g_n$ forms a finite dimensional subspace of $C[a,b]$, the existence of a best approximation from this subspace is guaranteed by the existence theorem, which was stated in the introduction.

Theorem 9. *(Characterization theorem) Let $f, g_1, \ldots, g_n$ be continuous functions on a compact metric space X and define a residual function*

$$r(x) = \sum_{i=1}^{n} c_i g_i(x) - f(x).$$

The coefficients $c_1, \ldots, c_n$ minimize $||r||_\infty = \max_{x\in X} |\sum_{i=1}^{n} c_i g_i(x) - f(x)|$ if and only if the zero vector is contained in the convex hull of the set

$$U = \{r(x)\hat{x} : |r(x)| = ||r||_\infty\},$$

where $\hat{x} = [g_1(x), \ldots, g_n(x)]^\intercal$.

Proof. Both implications are proven by contraposition.

($\Longleftarrow$) Assume that $||r||_\infty$ is not minimum. Then there is a vector $d = [d_1, \dots d_n] \in \mathbb{R}^n$ such that

$$||\sum_{i=1}^{n}(c_i - d_i)g_i - f||_\infty < ||\sum_{i=1}^{n} c_i g_i - f||_\infty,$$

that is,

$$||r - \sum_{i=1}^{n} d_i g_i||_\infty < ||r||_\infty. \tag{1}$$

Define $X_0 = \{x \in X : |r(x)| = ||r||_\infty\}$. Notice that this definition is justified because r is a continuous function on a compact set and therefore attains its extrema on that set. By inequality (1), we have for $x \in X_0$ that

$$(r(x) - \sum d_i g_i(x))^2 < r(x)^2.$$

By expanding the left hand side of this last inequality, we find that

$$\begin{aligned} & r(x)^2 - 2r(x)\sum d_i g_i(x) + (\sum d_i g_i(x))^2 < r(x)^2 \\ \implies & (\sum d_i g_i(x))^2 < 2r(x)\sum d_i g_i(x) \\ \implies & 0 < r(x)\sum d_i g_i(x) = \langle d, r(x)\hat{x}\rangle. \end{aligned} \tag{2}$$

Inequality (2) tells us that for the vector d, there is no vector $u \in U$ such that $\langle d, u\rangle \leq 0$. Furthermore, lemma 5 from appendix A tells us that the set U is compact. Therefore, the theorem on linear inequalities from section 2 tells us that $\mathbf{0} \notin H(U)$.

($\Longrightarrow$) Assume $\mathbf{0} \notin H(U)$. Then the theorem on linear inequalities tells us that there is a vector $d = [d_1, \dots, d_n]$ so that inequality (2) is valid for $x \in X_0$. The set X_0 is compact being a closed subset of the compact set X; assume $(x_n) \to x^* \in X$ with $x_n \in X_0$ for all n. Then $\lim_{n\to\infty} r(x_n) = r(x^*)$ by continuity of r. Thus $r(x^*) = ||r||_\infty$ since $r(x_n) = ||r||_\infty$ for all n, which implies $x^* \in X_0$. Because of the compactness of X_0, we can define the number $\epsilon = \min_{x\in X_0} r(x)\langle d, \hat{x}\rangle$ which is positive by inequality (2). Define

$$X_1 = \{x \in X : r(x)\langle d, \hat{x}\rangle \leq \epsilon/2\}.$$

This set is the pre-image of a closed set under a continuous function, hence closed. Again, this directly implies that X_1 is compact because it is a closed subset of the compact set X. Notice moreover that the sets X_0 and X_1 have an empty intersection. By compactness of the set X_1, $|r(x)|$ achieves its supremum $E < ||r||_\infty$ on X_1. We will prove that there is a $\lambda > 0$ such that $||r - \lambda\sum d_i g_i||_\infty < ||r||_\infty$, which means that the coefficients $c_1, \dots, c_n$ do not

minimize $||r||_\infty$. Take $x \in X_1$ and let $0 < \lambda < (||r|| - E)/||\sum d_i g_i||_\infty$. We apply the triangle inequality to see that

$$\begin{aligned} |r(x) - \lambda \sum d_i g_i(x)| &\leq |r(x)| + \lambda |\sum d_i g_i(x)| \\ &\leq E + \lambda ||\sum d_i g_i||_\infty \\ &< ||r||_\infty \end{aligned} \tag{3}$$

for all $x \in X_1$. Now take $x \notin X_1$ and choose λ such that $0 < \lambda < \epsilon/||\sum d_i g_i||_\infty^2$. Then

$$\begin{aligned} (r(x) - \lambda \sum d_i g_i(x))^2 &= r(x)^2 - 2\lambda r(x)\langle d, \hat{x}\rangle + \lambda^2 (\sum d_i g_i(x))^2 \\ &\leq r(x)^2 - 2\lambda\epsilon + \lambda^2 (\sum d_i g_i(x))^2 \\ &< ||r||_\infty^2 + \lambda(-\epsilon + \lambda ||\sum d_i g_i||_\infty^2) \\ &< ||r||_\infty^2 \end{aligned} \tag{4}$$

Inequalities (3) and (4) prove that the coefficients $c_1, \ldots, c_n$ do not minimize $||r||_\infty$. □

4 The alternation theorem

Theory presented in the previous sections will come together in the present section to prove the *alternation theorem.* The latter theorem is key to understanding the mechanism of the Remez algorithm. Moreover, a corollary of the alternation theorem, presented at the end of this section, is used explicitly in each iteration of the Remez algorithm.

4.1 The Haar condition

In upcoming results, the generalized polynomial $\sum_{i=1}^n c_i g_i(x)$ will be subject to the *Haar condition*, to be defined in a moment. At the end of this subsection, we prove a lemma which is used in the proof of the alternation theorem.

Definition 3. *Let $g_1, \ldots, g_n$ be continuous functions defined on the interval [a,b] and let $x_i \in [a, b]$ for all $1 \leq i \leq n$. The system $\{g_1, \ldots, g_n\}$ is said to satisfy the Haar condition if the determinant*

$$D[x_1, \ldots, x_n] = \begin{vmatrix} g_1(x_1) & \ldots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \ldots & g_n(x_n) \end{vmatrix} \tag{5}$$

is nonzero whenever $x_1, \ldots, x_n$ are all distinct.

The following example is important because it allows us to apply everything we prove about systems satisfying the Haar condition to degree n polynomials.

Example 1. *The system* $\{1, x, \ldots, x^n\}$ *satisfies the Haar condition. For this system, we have*

$$D[x_1, \ldots, x_{n+1}] = \begin{vmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{vmatrix},$$

the famous Vandermonde determinant. It can be shown by an induction proof that this determinant has the value

$$D = \prod_{0 \le j < i \le n} (x_i - x_j),$$

which does not vanish whenever $x_0, \ldots, x_n$ *are all distinct [2, p. 74].*

Example 2. *The system* $\{\sin x, \cos x\}$ *satisfies the Haar condition on any interval* $[a, b] \subset (k\pi, (k+1)\pi)$ *for* $k \in \mathbb{Z}$*: assume* $x_1, x_2 \in [a, b]$*. Then*

$$\begin{aligned} D[x_1, x_2] &= \begin{vmatrix} \sin x_1 & \cos x_1 \\ \sin x_2 & \cos x_2 \end{vmatrix} \\ &= \sin x_1 \cos x_2 - \cos x_1 \sin x_2 \\ &= \sin(x_1 - x_2) \neq 0 \end{aligned}$$

for $x_1 - x_2 \neq k\pi$*,* $k \in \mathbb{Z}$*.*

In the proof of the next lemma, we will make use of a useful but often forgotten theorem from linear algebra.

Theorem 10. *(Cramer's rule) Let* $Ax = b$ *with* A *an* n *by* n *matrix and* $x, b \in \mathbb{R}^n$*. Assume that* $\det(A) \neq 0$ *and let* A^i *denote the matrix which is the result of replacing the* i^{th} *column vector of* A *by the vector* b*. Then the* i^{th} *entry of the solution vector* x *is given by*

$$x_i = \frac{\det(A^i)}{\det(A)}.$$

The proof is excluded here as it is often part of the undergraduate curriculum on linear algebra. A proof can be found in standard linear algebra textbooks, see for example [12, p. 104].

Lemma 1. *Let* $\{g_1, \ldots, g_n\}$ *be a system of continuous functions defined on the interval* $[a, b]$ *and assume the Haar condition is satisfied. Assume that* $a \le x_1 < \cdots < x_n \le b$ *and* $a \le y_1 < \cdots < y_n \le b$*. Then the determinants* $D[x_1, \ldots, x_n]$ *and* $D[y_1, \ldots, y_n]$*, defined by (5), have the same sign.*

Proof. By contraposition. Assume that the conditions of the lemma are satisfied. Furthermore, assume without loss of generality that

$$D[x_1, \ldots, x_n] < 0 < D[y_1, \ldots, y_n], \tag{6}$$

otherwise interchange roles of $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. Because the functions g_i are continuous, the value of $D[x_1, \ldots, x_n]$ depends continuously on the x_i. We can therefore define the continuous function $f : [0, 1] \to \mathbb{R}$,

$$f(\lambda) = D[\lambda x_1 + (1 - \lambda)y_1, \ldots, \lambda x_n + (1 - \lambda)y_n]. \tag{7}$$

By the intermediate value theorem [13, p. 120] and assumption (6), there is a $\lambda^* \in (0, 1)$ such that $f(\lambda^*) = 0$. From the Haar condition it then follows that not all entries in the determinant

$$D[\lambda^* x_1 + (1 - \lambda^*)y_1, \ldots, \lambda^* x_n + (1 - \lambda^*)y_n]$$

are distinct; otherwise this determinant were nonzero. In other words, there is an $i \neq j$ such that

$$\lambda x_i + (1 - \lambda)y_i = \lambda x_j + (1 - \lambda)y_j,$$

which means that

$$\lambda(x_i - x_j) = (1 - \lambda)(y_j - y_i),$$

so that the quantities $x_i - x_j$ and $y_i - y_j$ have opposite signs, that is, not both sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ are in ascending order. □

Lemma 2. *Let $\{g_1, \ldots, g_n\}$ be a system of continuous functions defined on the interval $[a, b]$ and assume the Haar condition is satisfied. Assume that $a \leq x_0 < \cdots < x_n \leq b$ and assume that the constants $\lambda_0, \ldots, \lambda_n$ are nonzero. Additionally let*

$$A = \{\lambda_i \hat{x}_i : \hat{x}_i = [g_1(x_i), \ldots, g_n(x_i)], 0 \leq i \leq n\}.$$

Then $\mathbf{0} \in H(A)$ if and only if $\lambda_i \lambda_{i-1} < 0$ for $1 \leq i \leq n$, that is, the λ_i's alternate in sign.

Proof. Let the set A be as defined in the statement of the lemma. We have $\mathbf{0} \in H(A)$ if and only if there are constants $\theta_i > 0$, $i = 0, \ldots, n$ (if one of them were equal to zero, the Haar condition would be violated) such that

$$\sum_{i=0}^{n} \theta_i \lambda_i \hat{x}_i = \mathbf{0}. \tag{8}$$

Note here that we could normalize the vector $[\theta_0, \ldots, \theta_n]$ so that $\sum \theta_i = 1$. From equation (8) it follows that we can write

$$\hat{x}_0 = -\sum_{i=1}^{n} \frac{\theta_i \lambda_i}{\theta_0 \lambda_0} \hat{x}_i,$$

which we write as matrix-vector equation:

$$\begin{bmatrix} g_1(x_1) & \ldots & g_1(x_n) \\ \vdots & \ddots & \vdots \\ g_n(x_1) & \ldots & g_n(x_n) \end{bmatrix} \begin{bmatrix} \frac{-\theta_1 \lambda_1}{\theta_0 \lambda_0} \\ \vdots \\ \frac{-\theta_n \lambda_n}{\theta_0 \lambda_0} \end{bmatrix} = \begin{bmatrix} g_1(x_0) \\ \vdots \\ g_n(x_0) \end{bmatrix}.$$

We apply Cramer's rule to find

$$\frac{-\theta_i\lambda_i}{\theta_0\lambda_0} = \frac{D[x_1,\ldots,x_{i-1},x_0,x_{i+1},\ldots,x_n]}{D[x_1,\ldots,x_n]}. \tag{9}$$

We order the x_i's in the determinant in the numerator by moving x_0 $i-1$ places to the left, that is, the determinant changes sign $i-1$ times. By lemma (1), the numerator and denominator in (9) have the same sign once we placed x_0 $i-1$ places to the left. Hence $\operatorname{sgn}(\frac{\theta_i\lambda_i}{\theta_0\lambda_0}) = (-1)^{i-1}$. Since $\theta_0\lambda_0 > 0$ and $\theta_i > 0$ for all i, we get $\operatorname{sgn}(\lambda_i) = (-1)^i$ and conclude that the λ_i's alternate in sign.

To prove the converse direction, assume $\operatorname{sgn}(\lambda_i) = (-1)^i$. Then in the solution (9) to equation (8), we can choose all the θ_i strictly positive, which then implies that $\mathbf{0} \in H(A)$. Had we started with $\operatorname{sgn}(\lambda_i) = (-1)^{i+1}$ instead, we could multiply the solution vector by -1 still making the result of the sum in (8) equal to zero.

□

4.2 The alternation theorem

Theorem 11. *(Alternation theorem) Let $\{g_1,\ldots,g_n\}$ be a system of continuous functions satisfying the Haar condition and let X be a closed subset of $[a,b]$ containing at least $n+1$ points. Furthermore, let f be a continuous function defined on X and let r denote the residue function $r(x) = f(x) - \sum_{i=1}^n c_i g_i(x)$. The coefficients $c_1,\ldots,c_n$ minimize*

$$\max_{x\in X}|r(x)| = ||r||_\infty$$

if and only if

$$r(x_i) = -r(x_{i-1}) = \pm||r||_\infty$$

for $a \le x_0 < \cdots < x_n \le b$ with $x_0,\ldots,x_n \in X$.

Proof. ($\Longrightarrow$) Assume that the coefficients $c_1,\ldots,c_n$ minimize $||r||_\infty$. Denote the vector $[g_1(x),\ldots,g_n(x)]$ by $\hat{x}$ and define the set

$$U = \{r(x)\hat{x} : |r(x)| = ||r||_\infty, x \in X\}.$$

By the characterization theorem, $\mathbf{0} \in H(U)$, since we assumed that $c_1,\ldots,c_n$ minimize $||r||_\infty$. By Caratheodory's theorem, any element in $H(U)$ can be written as a convex linear combination of no more than $n+1$ elements from $U \subset \mathbb{R}^n$, that is, there exists an integer $k \le n$ and scalars $\lambda_0,\ldots,\lambda_k$, all strictly positive, such that

$$\mathbf{0} = \sum_{i=0}^{k} \lambda_i r(x_i)\hat{x}_i, \qquad r(x_i)\hat{x}_i \in U. \tag{10}$$

Here $\hat{x}_i$ is the vector $[g_1(x_i),\ldots,g_n(x_i)]$. By the Haar condition, in fact, we must have $k \ge n$ and we conclude $k = n$. Assume the x_i's are labeled in such a way that $a \le x_0 < \cdots < x_n \le b$. By equation (10), $\mathbf{0} \in H(A)$ where

$$A = \{\lambda_i r(x_i)\hat{x}_i : i = 0,\ldots,n\}.$$

Our previous lemma tells us that this is only possible if $\lambda_i r(x_i)\lambda_{i-1}r(x_{i-1}) < 0$ for $i = 1, \ldots, n$. Because the λ_i are all strictly positive, we must in fact have that the $r(x_i)$'s alternate in sign. Furthermore, note that $|r(x_i)| = ||r||_\infty$ for $i = 0, \ldots, n$ because $r(x_i)\hat{x}_i \in U$ for all i.

($\Longleftarrow$) Assume that $a \leq x_0 < \cdots < x_n \leq b$ for $x_0, \ldots, x_n \in X$ and $r(x_i) = -r(x_{i-1}) = \pm||r||_\infty$ for $i = 1, \ldots, n$. Then, since the $r(x_i)$'s alternate in sign, our previous lemma tells us that $\mathbf{0} \in H(B)$ where

$$\begin{aligned} B &= \{r(x_i)\hat{x}_i : i = 0, \ldots, n\} \\ &= \{r(x_i)\hat{x}_i : |r(x_i)| = ||r||_\infty, i = 0, \ldots, n\} \\ &\subset U, \end{aligned}$$

where U is the set we defined in the first part of the proof. Hence, using observation 1, $\mathbf{0} \in H(U)$ and we conclude by the characterization theorem that the coefficients $c_1, \ldots, c_n$ were chosen such that the uniform norm of the residue function

$$r(x) = f(x) - \sum_{i=1}^{n} c_i g_i(x)$$

is minimized on $[a, b]$.

□

In some simple cases, the best approximation to a function can be found by direct application of the alternation theorem, we give an example of how this can be done below.

Example 3. Let us find the best linear approximation $P(x) = c_0 + c_1 x$ to the function $f(x) = e^x$ on $[0, 1]$, i.e. our Haar system is $\{1, x\}$. By the alternation theorem, the error function $r = f - P$ must alternate at least three times. From figure 1 it is clear that the points of alternation are 0, 1 and a point ξ between the two. Furthermore, at the alternation points, the quantity $|f(x) - P(x)|$ is equal to $||r||_\infty := \epsilon$. We get the following equations:

$$\epsilon = f(0) - P(0) = 1 - c_0$$

$$-\epsilon = f(\xi) - P(\xi) = e^\xi - c_0 - c_1\xi$$

$$\epsilon = f(1) - P(1) = e - c_0 - c_1$$

Moreover, we use the fact that $f - P$ has an extreme value at ξ, that is,

$$0 = f'(\xi) - P'(\xi) = e^\xi - c_1.$$

Using the first and the fourth equation, we find $c_0 = 1 - \epsilon$ and $c_1 = e^\xi$. Next, we substitute these values for c_0 and c_1 in the third equation and solve for ξ to find that $\xi = \log(e - 1)$. Lastly we solve the second equation for ϵ and find that $\epsilon = \frac{2 - e + (e-1)\log(e-1)}{2} \approx 0.106$. The best linear approximation to e^x on $[0, 1]$ is $P(x) = (e - 1)x + 1 - \epsilon$.

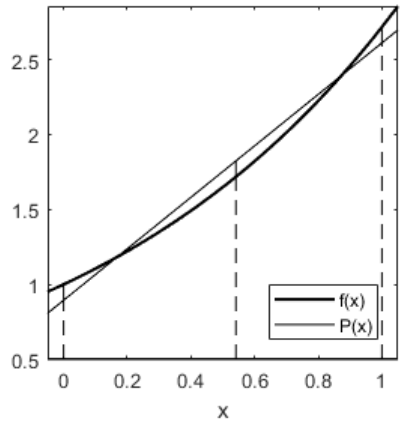

Figure 1: Linear approximation for e^x on $[0,1]$ determined by our implementation of the Remez algorithm in three iterations. The error function $P-f$ of the best approximation equioscillates on the points $0, 1$ and $\xi = \log(e-1) \approx 0.54$ as indicated in the figure.

4.3 An applicable corollary

The following result, which we present as a corollary of the alternation theorem, will be used explicitly in the first step of each iteration of the Remez algorithm.

Corollary 2. *Let $\{g_1, \ldots, g_n\}$ be a system of continuous functions satisfying the Haar condition and let $f \in C[a,b]$. Let r denote the residue function $r(x) = \sum_{i=1}^n c_i g_i(x) - f(x)$. Assume that $a \le x_0 < \cdots < x_n \le b$. The coefficients $c_1, \ldots, c_n$ minimizing $\max_{i=0,\ldots,n} |r(x_i)|$ are obtained by solving the following linear system of n equations and n unknowns*

$$\sum_{j=1}^{n} c_j[g_j(x_i) - (-1)^i g_j(x_0)] = f(x_i) - (-1)^i f(x_0), \quad i = 1, \ldots, n. \tag{11}$$

Proof. To make our notation shorter, we start by writing $p(x) = \sum_{j=1}^n c_j g_j(x)$. Applying the alternation theorem with $X = \{x_i : a \le x_0 < \cdots < x_n \le b\}$ gives us

$$f(x_{i+1}) - p(x_{i+1}) = -[f(x_i) - p(x_i)], \quad i = 0, 1, \ldots, n. \tag{12}$$

Let $h = f(x_0) - p(x_0)$. By the alternation theorem, $|r(x_i)| = |h|$ for all i. It follows now from equation (12) that

$$f(x_i) - p(x_i) = (-1)^i h \tag{13}$$

for all i. The last equation can be written as (take care not to confuse the

indices i and j!)

$$\begin{aligned} f(x_i) - \sum_{j=1}^{n} c_j g_j(x_i) &= (-1)^i [f(x_0) - \sum_{j=1}^{n} c_j g_j(x_0)] \\ \implies \sum_{j=1}^{n} c_j [g_j(x_i) - (-1)^i g_j(x_0)] &= f(x_i) - (-1)^i f(x_0) \end{aligned} \tag{14}$$

for $i = 1, \ldots, n$ (notice that $i = 0$ gives the trivial equation $0 = 0$). The system described by (14) is a linear system with n equations and n unknowns $c_1, \ldots, c_n$. The matrix belonging to the system is nonsingular; the alternation theorem combined with theorem 3 guarantees us the existence of the solution vector $c = [c_1, \ldots, c_n]$ for all $f \in C[a, b]$.

□

5 The Remez algorithm

In the present section, we describe the Remez algorithm. The Remez algorithm is an iterative procedure based on the alternation theorem, which can find the best approximation to a continuous function in the minimax sense. Stated more precisely, given $f \in C[a, b]$ and a system of continuous functions $\{g_1, \ldots, g_n\}$ satisfying the Haar condition, the algorithm will find a coefficient vector $c = [c_1, \ldots, c_n]$ minimizing the uniform norm of the function

$$r(x) = f(x) - \sum_{j=1}^{n} c_j^* g_j(x)$$

on the interval $[a, b]$. In practice we will stop the procedure once our approximation is close enough to the best approximation.

5.1 Two observations

Before stating the algorithm, we make two observations which may be helpful in understanding the steps of the Remez algorithm.

Observation 2. *Let $\{g_1, \ldots, g_n\}$ be a system of continuous functions satisfying the Haar condition, let $f \in C[a, b]$ and let $a \leq x_0 < \cdots < x_n \leq b$. Corollary 2 tells us that solving the linear system given by (11) gives us coefficients $c_1, \ldots, c_n$ which minimize the expression*

$$\max_{i=0,\ldots,n} |f(x_i) - \sum_{j=1}^{n} c_j^* g_j(x_i)|.$$

With the coefficients computed by solving system (11), define

$$r(x) = f(x) - \sum_{j=1}^{n} c_j g_j(x).$$

The alternation theorem tells us that

$$r(x_i) = -r(x_{i-1}) = \pm||r||_\infty, \quad i = 1, \ldots, n,$$

which means that the residue function r has a root in each interval (x_{i-1}, x_i).

Observation 3. *With the notation from observation 2, let*

$$A = \{\sum_{j=1}^{n} c_j^* g_j(x) : c_j^* \in \mathbb{R}, x \in [a, b]\}$$

and let P^* be the best approximation to f from the set A. Furthermore, let $P(x) = \sum_{j=1}^{n} c_j g_j(x)$ with coefficients as in observation 2. Choose $y \in [a, b]$ so that

$$|r(y)| = \max_{x \in [a,b]} |r(x)| = ||r||_\infty.$$

In [1, p. 86] it is found that

$$||f - P||_\infty \leq ||f - P^*||_\infty + \delta,$$

where

$$\delta = |r(y)| - |r(x_0)|.$$

Recall that $|r(x_0)| = \cdots = |r(x_n)|$ by the alternation theorem. In the algorithm which we will state below, we will stop iterating once δ is small enough, because this tells us that the computed approximation is close enough to the best approximation from A.

5.2 Statement of the Remez algorithm

Input: A function $f \in C[a, b]$, functions $g_1, \ldots, g_n \in C[a, b]$ so that the system $\{g_1, \ldots, g_n\}$ satisfying the Haar condition, the interval $[a, b]$, an initial reference $\{x_0, \ldots, x_n\}$, where $a \leq x_0 < \cdots < x_n \leq b$ and a constant $\delta > 0$ for the stopping criterion. We describe the steps of a generic iteration k.

Step 1: If $k = 1$, the reference $\{x_0, \ldots, x_n\}$ comes from the input, otherwise it is defined in the previous iteration. Solve linear system (11) to compute coefficients $c_1, \ldots, c_n$ minimizing the expression

$$\max_{i=0,\ldots,n} |f(x_i) - \sum_{j=1}^{n} c_j^* g_j(x_i)|.$$

Define $r(x) = f(x) - \sum_{j=1}^{n} c_j g_j(x)$ with the coefficients $c_1, \ldots, c_n$ just computed.

Note: The function $P(x) = \sum_{j=1}^{n} c_j g_j(x)$ is the approximation for f obtained in the present iteration, iteration k. The upcoming steps only influence the approximation computed in the next iteration, iteration $k + 1$.

Stopping criterion: Find $y \in [a, b]$ so that

$$|r(y)| = \max_{x \in [a,b]} |r(x)|.$$

Stop iterating if

$$|r(y)| - |r(x_0)| < \delta.$$

Otherwise continue in step 2.

Step 2: Define $z_0 = a$, $z_{n+1} = b$ and find a root z_i in (x_{i-1}, x_i) for all $i = 1, \ldots, n$. The existence of these roots was mentioned in observation 2.

Step 3: Let $\sigma_i = \operatorname{sgn} r(x_i)$. For all $i = 0, \ldots, n$, find $y_i \in [z_i, z_{i+1}]$ where $\sigma_i r(y)$ is a maximum with the property that $\sigma_i r(y_i) \geq \sigma_i r(x_i)$. Replace the reference $\{x_0, \ldots, x_n\}$ by $\{y_0, \ldots, y_n\}$.

Step 4: In the stopping criterion we computed $y \in [a, b]$ so that

$$|r(y)| = \max_{x \in [a,b]} |r(x)|.$$

If

$$|r(y)| = \max_{i=0,\ldots,n} |r(y_i)|,$$

go to step 1 of iteration $k + 1$ with the reference defined in step 3. If

$$|r(y)| > \max_{i=0,\ldots,n} |r(y_i)|,$$

include the point y in the set $\{y_0, \ldots, y_n\}$ and put it in the right position so that the elements are still in ascending order. Lastly, remove one of the y_i in such a way that $r(x)$ still alternates in sign on the resulting set. A description of how to accomplish this is given in appendix B. The resulting set is the new reference. Start in step 1 of iteration $k + 1$ with this new reference.

5.3 A visualization of the reference adjustment

In figure 2 it can be seen how the reference is adjusted in an iteration of the Remez algorithm. The figure shows the residue function in the first iteration for determining a second degree polynomial approximation for the function

$$f(x) = e^{-x} \sin(3\pi x) \cos(3\pi x) |\sin(2\pi x)|$$

on the interval $[0, 1]$. Notice that for a second degree polynomial approximation, our Haar system is $\{1, x, x^2\}$, that is, we have 3 basis elements. Therefore our initial reference must consist of 4 points. As initial reference we chose 4 equispaced nodes in $[0.2, 0.8]$. This choice for f and for the initial reference is a result of trial and error; the goal was to obtain plots in which it is visable how the reference is adjusted in each step of an iteration.

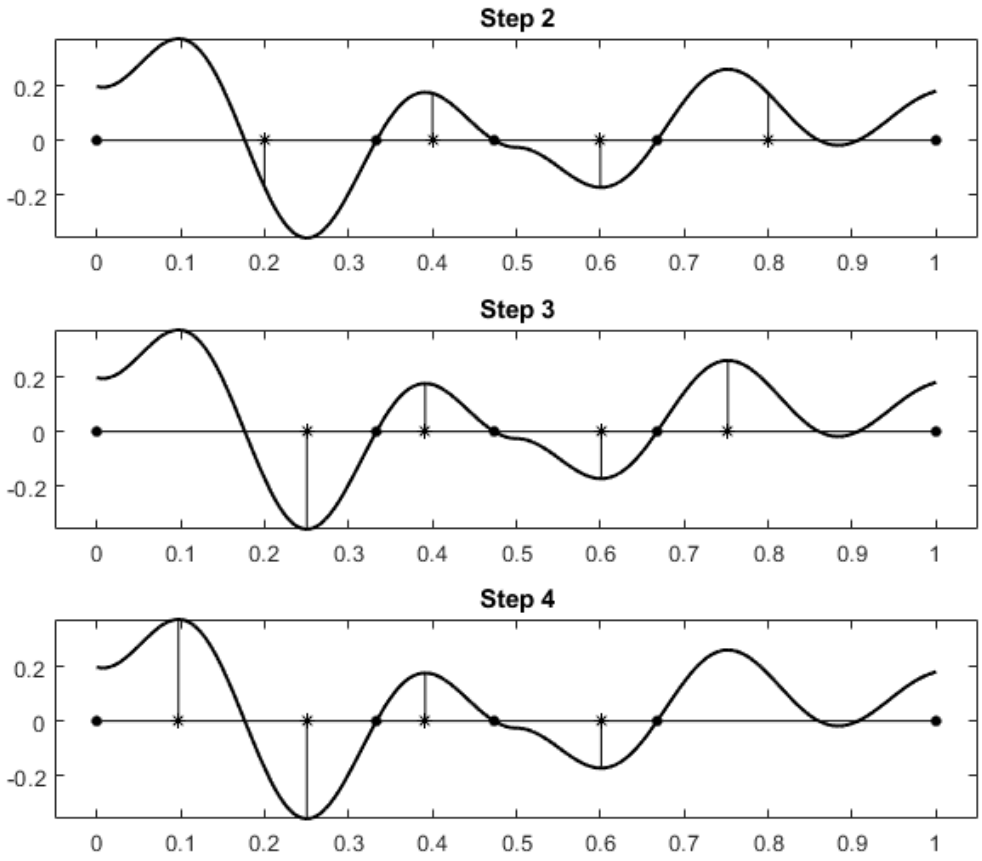

Figure 2: Residue function and reference points in steps $2, 3, 4$ of the first iteration of the Remez algorithm for determining a second degree polynomial approximation to the function $f(x) = e^{-x}\sin(3\pi x)\cos(3\pi x)|\sin(2\pi x)|$. In each step the locations of the reference points and the numbers $z_0, \ldots, z_5$ are indicated by stars and dots, respectively. In step 2, equioscillations on the initial reference are visible. In step 3, the reference points are re-located to the position of local maxima in accordance with the description of the algorithm. In the last step, the location of the global maximum of $|r|$ is included into the reference and one point is removed in the way described in step 4 of the statement of the algorithm. The residue function still alternates in sign on the resulting reference. The figure was created using our implementation of the Remez algorithm.

6 Testing our implementation of the Remez algorithm

In this section, we discuss several results obtained by using our Matlab implementation of the Remez algorithm. We discuss situations in which the implementation works well and discuss in what cases it may not give satisfactory results. We start by discussing some of the choices we made when making our implementation.

6.1 Comments about the implementation

Considerations

- Instead of letting the algorithm stop when $\delta = ||r||_\infty - |r(x_0)|$ is small enough, we will let it stop when $|\delta|$ is small enough. Theoretically, $\delta \geq 0$, but because in the algorithm $||r||_\infty$ becomes close to $|r(x_0)|$, situations may occur where δ becomes less than 0 due to computational errors.

- We use Matlab's built-in function `fminbnd` to locate local maxima of the residue function. For example, a maximum of a function f can be found on an interval by minimizing $-f$. In the third step of the algorithm it is necessary to find $y_i \in [z_i, z_{i+1}]$ where $\sigma_i r(y)$ is a maximum with the property that $\sigma_i r(y_i) \geq \sigma_i r(x_i)$. If `fminbnd` does not locate a maximum with this property, but e.g. a lower local maximum, we start a brute-force search for a maximum satsifying $\sigma_i r(y_i) \geq \sigma_i r(x_i)$. If no such maximum can be found, it must be that $\sigma_i r(x_i)$ is close to the absolute maximum on $[z_i, z_{i+1}]$ and we choose $y_i = x_i$.

- We have chosen $z_0 = a$ and $z_{n+1} = b$. These points are not generally zeroes of the residue function and in the search for a maximum in the intervals $[z_0, z_1]$ and $[z_n, z_{n+1}]$ we also evaluate the points z_0 and z_{n+1}, respectively.

- The global maximum of the residue function is located by a brute-force search. We evaluate the residue function in 10000 equispaced steps between $[0, 1]$ and find the maximum of these evaluations. The brute-force search should find the $x-$coordinate $x_{\max}$ of the maximum with an error of at most 10^{-4}.

Accuracy of the brute-force search We can use the mean value theorem (MVT) [13, p. 138] to say something about the accuracy of the value for $||r||_\infty$ found by use of our brute-force method, where r is again a residue function. Suppose that x_1 is the $x-$coordinate of the maximum found by the brute-force search and x_2 is the $x-$coordinate of the exact location of the maximum. Recall that the residue function r is continuous. Under the condition that

r is differentiable on (x_1, x_2), application the MVT and using the fact that $|x_1 - x_2| \leq 10^{-4}$ tells us that there exists a point $c \in (x_1, x_2)$ such that

$$\begin{aligned} |r(x_2) - r(x_1)| &= |r'(c)| \cdot |x_2 - x_1| \\ &\leq |r'(c)| \cdot 10^{-4}. \end{aligned} \tag{15}$$

If then, for example $|r'(x)| \leq 1$ for x in a neighbourhood of 10^{-4} around the location of the true maximum, we can conclude that the approximated value for $||r||_\infty$ differs at most by a value of 10^{-4} from the true value.

Assuming that r is differentiable, it can be noticed that in a small neighbourhood around the location of the maximum of $|r|$, $|r'|$ would only attain a value as large as 1 in rather extreme cases, e.g. when the plot of the residue function shows a sharp peak. High values for $|r'|$ at the endpoints of the interval of approximation should not cause problems because the brute-force search method evaluates r at these endpoints. Inspection of the plot of the residue function can indicate if it is reasonable to trust the value found for $||r||_\infty$. Unless indicated otherwise, we will, for the remainder of this section, accept the values found for $||r||_\infty$ by the brute-force method without further comments.

6.2 Examples

Comparison with two analytically determined results We compare two linear approximations determined by our implementation with two analytically determined best linear approximations, see table 1. Here, n is the number of iterations used, P_r^* is the approximation determined by the Remez algorithm and P_a^* is the analytically determined best approximation.

In both examples, $||f - p_r^*||_\infty$ and $||f - p_a^*||_\infty$ agree on 8 decimal digits. In these two examples the extrema of the residue function occur near the middle and at the endpoints of $[0, 1]$. A plot of the two residue functions is given in figure 3. Because our brute-force method for finding the extrema of the residue function evaluates the function at the endpoints, it will find these extrema in a precise way. We conclude that the quality of the approximation determined by our implementation of the Remez algorithm is in both cases satisfactory compared to the quality of the analytically determined best approximation.

$f(x)$	n	$\|\|f - P_r^*\|\|_\infty$	$\|\|f - P_a^*\|\|_\infty$	source
$\sin(\frac{1}{2}\pi x)$	2	0.105256830566	0.105256831176	[2, p. 76]
e^x	2	0.105933415992	0.105933416258	example 3

Table 1: Norms of the residue functions of two analytically determined best approximations on $[0, 1]$ versus approximations determined by our implementation of the Remez algorithm.

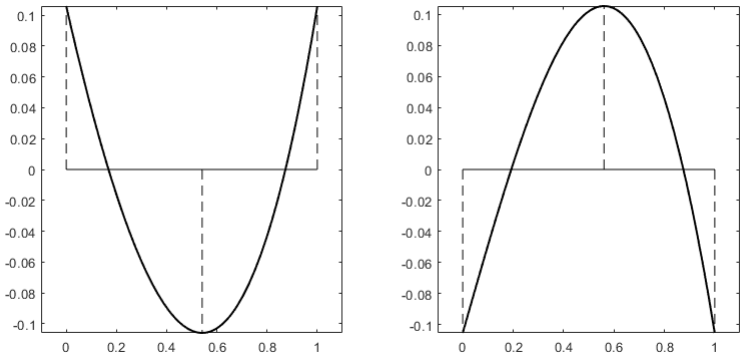

Figure 3: Residue function $f - P_r^*$ for $f(x) = e^x$ (left) and $f(x) = \sin(\frac{1}{2}\pi x)$ (right). The $x-$coordinates of the vertical lines are the reference points on which the approximation was computed.

Several polynomial approximations for one function Figure 4 shows six polynomial approximations for the function $f(x) = e^x \cos(2\pi x) \sin(2\pi x)$. The reason for choosing this function is because it has several extreme values on the interval $[0, 1]$. Choosing a function which is easy to approximate by low degree polynomials would make the plots less interesting, as the difference between the approximant and approximation would be hardly visible.

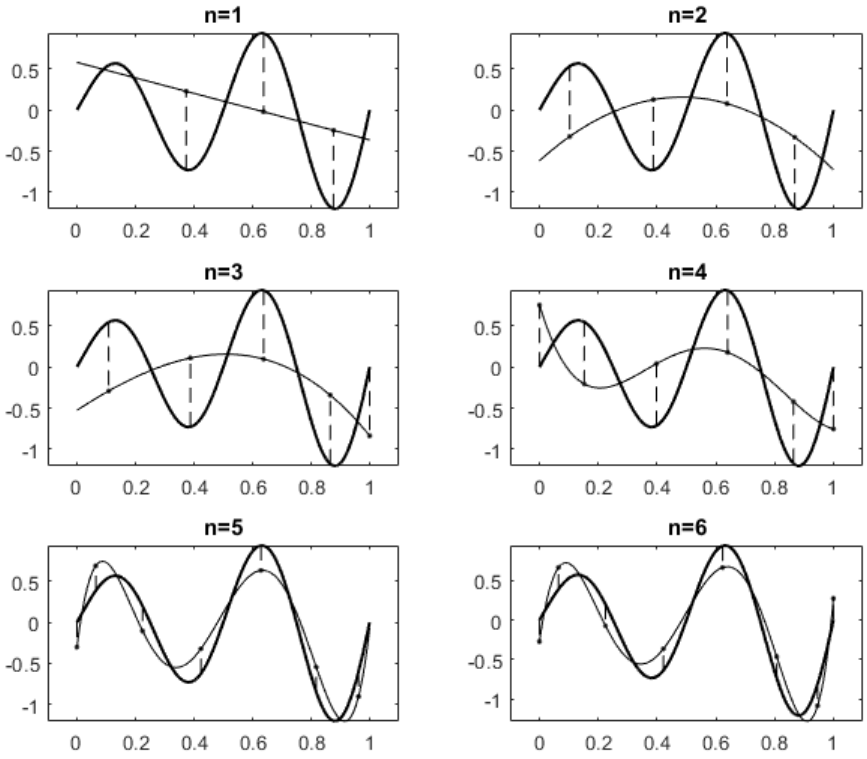

Figure 4: Best degree n polynomial approximations on $[0,1]$ for $f(x) = e^x \cos(2\pi x)\sin(2\pi x)$ determined by our implementation of the Remez algorithm. The algorithm was stopped when $|\delta| < 10^{-5}$. We started with an equispaced reference. The dashed lines connect the approximant and the approximation on the reference points on which the approximation was computed. The equioscillations on the reference points can be observed in the plots.

Table 2 shows the maximum of the error function and the number of iterations used for degree $n = 1, \ldots, 17$ polynomial approximations for f. We increased the degree of the approximation polynomial until failure. For $n = 18$, the implementation gave an error for the first time; the function `fzero` indicates that the residue function has the same sign on the endpoints between which a root should be found. This indicates that for $n = 18$, the condition number of the matrix in the linear system solved in the algorithm is too high, so that the computed residue function may not satisfy the alternation property anymore. Further, from the table we observe that higher degree polynomials give better approximations, as can be expected.

n	$\|\|f-P^*\|\|_\infty$	iterations	n	$\|\|f-P^*\|\|_\infty$	iterations
1	0.95484123	5	10	$4.3157657 \cdot 10^{-3}$	4
2	0.85490254	4	11	$1.1728234 \cdot 10^{-3}$	4
3	0.83717665	5	12	$2.6148995 \cdot 10^{-4}$	4
4	0.75385311	3	13	$7.9211438 \cdot 10^{-5}$	5
5	0.30308145	4	14	$1.0753948 \cdot 10^{-5}$	5
6	0.27180459	4	15	$3.8528423 \cdot 10^{-6}$	5
7	$7.6258611 \cdot 10^{-2}$	4	16	$3.6481047 \cdot 10^{-7}$	4
8	$4.5322638 \cdot 10^{-2}$	4	17	$1.6218161 \cdot 10^{-7}$	4
9	$1.1750510 \cdot 10^{-2}$	4	18	N/A	N/A

Table 2: Norms of the residue functions and number of iterations used for degree $n = 1, \ldots, 17$ approximations by the Remez algorithm to $f(x) = e^x \cos(2\pi x) \sin(2\pi x)$. For $n = 1, \ldots, n = 11$ we used $\delta = 10^{-5}$. We then decreased δ to 10^{-6} after $n = 11$ and to 10^{-7} after $n = 13$. For $n = 18$, our implementation fails for the first time.

Third degree polynomial approximations for several functions As another example, we apply our implementation to give third degree polynomial approximations for several functions. The plots are shown in figure 5. For each function, we compare how our implementation improves the initial approximation determined on the Chebyshev nodes, see table 3.

One may wonder if the points where the functions f_2, f_3, f_4 and f_6 are not differentiable can cause any problems in the determination of the approximation. Theoretically, this is not the case; in section 7, convergence of the Remez algorithm is proven and the only restriction on the approximant f is that it should be continuous on the interval of approximation.

In the implementation, we locate maxima using derivative-free methods. One possible computational problem however, may be in the determination of $||r||_\infty$ by the brute-force method. For example, the plot of f_4 suggests that $|r'|$ attains relatively high values in a small neighbourhood of the global minimum of f_4. If, for example, $|r'|$ takes a value of at most 10 near this extremum, the value found for $||r||_\infty$ using the brute-force method may differ with a value of 10^{-3} from the true value for $||r||_\infty$, according inequality (15). If more accurate results are desired, we could use a smaller stepsize for the brute-force search.

i	$f_i(x)$	$\|\|f_i - P_0^*\|\|_\infty$	$\|\|f_i - P_r^*\|\|_\infty$
1	$\sin(2\pi x)e^x$	0.4849165	0.3050985
2	$\|x - \frac{1}{2}\|$	0.0818596	0.0626497
3	$\tan(\frac{2}{5}\pi x)e^{-2x}\|x - 0.5\|$	0.0273625	0.0187958
4	$\sin(\frac{3}{2}\pi\|x - \frac{1}{2}\|)$	0.5011132	0.3749875
5	$\log(1.001 - x)$	3.0049850	1.1701450
6	$\|x - \frac{1}{4}\| \cdot \|x - \frac{1}{2}\| \cdot \|x - \frac{3}{4}\|$	0.0190784	0.0135400

Table 3: Norms of residue functions for the initial approximation P_0^* computed on the Chebyshev nodes and the approximation P_r^* determined with the implementation of the Remez algorithm. Iterations were stopped when $|\delta| < 10^{-3}$.

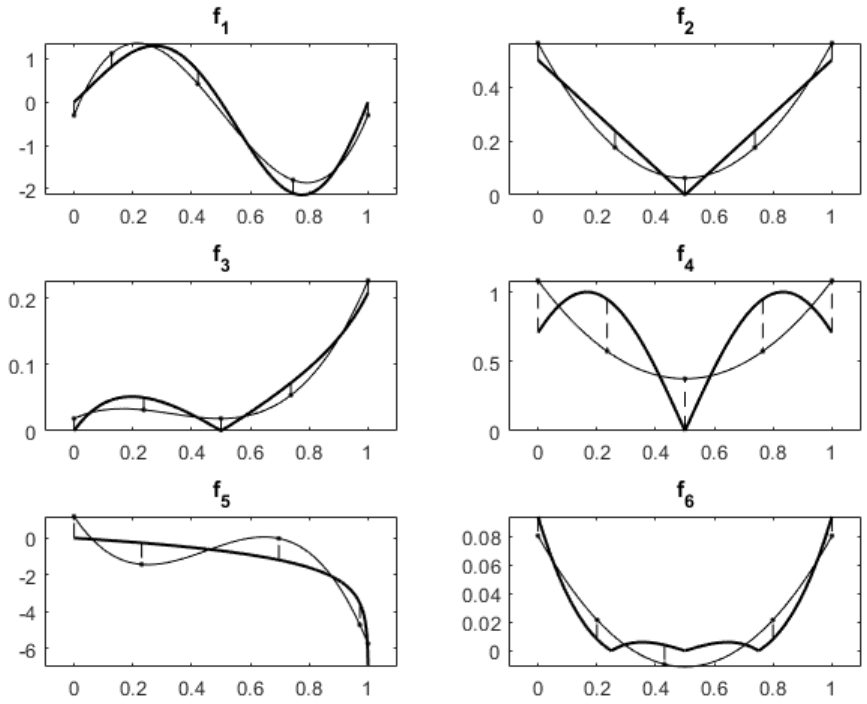

Figure 5: Third degree polynomial approximations to several functions. Iterations were stopped when $|\delta| < 10^{-3}$.

Table 3 shows the norms of the residue functions $||f - P_r^*||_\infty$. Here, P_0^* is the approximation determined on the initial reference, the Chebyshev nodes. We have included $||f - P_0^*||_\infty$ in the table to be able to see how the Remez

algorithm improves the first approximation on the Chebyshev nodes. In these examples, it can be seen that the Remez algorithm improves the approximation on the Chebyshev nodes by a factor between 1.3 and 2.6. Excluding f_5, the approximation is improved by a factor between 1.3 and 1.6 for each example.

In each of the examples, the approximation on the reference determined by the Remez algorithm is an improvement compared to the initial approximation on the Chebyshev nodes, as can be seen in table 3.

Approximation of four difficult functions We determine degree 10 polynomial approximations for four functions from [5, p. 734] and compare our results with the results from this paper. The norms of the residue functions can be found in table 4. Figure 6 shows the corresponding plots. One of the functions we approximate is

$$g(x) = \operatorname{sech}(10(0.5x + 0.3))^2 + \operatorname{sech}(100(0.5x + 0.1))^4 + \operatorname{sech}(1000(0.5x - 0.1))^6$$

and is given here because of the size of the expression.

Note that unlike the previous example, our interval of approximation is $[-1, 1]$. Also, because of the behaviour of the approximants $f_1, \ldots, f_4$, we use a grid of 10^5 equispaced steps for the brute-force search instead of 10^4 like in the previous examples. The iterations were stopped when $|\delta| < 10^{-4}$.

From table 4 we observe that for $i = 1, \ldots, 4$, $||f_i - P_r^*||_\infty$ differs no more than 10^{-5} from $||f_i - P_{cheb}^*||_\infty$, where P_r^* is the approximation determined by our implementation and P_{cheb}^* is the approximation determined by the implementation from [5]. Assuming the accuracy of the approximation determined by the implementation in the chebfun system, we conclude that our implementation can give satisfactory results concerning polynomial approximations up to degree 10, even when approximating functions with irregular behaviour such as a high number of oscillations, locally high derivatives and points at which the function is not differentiable.

i	$f_i(x)$	$\|\|f_i - P_r^*\|\|_\infty$	$\|\|f_i - P_{cheb}^*\|\|_\infty$
1	$\min\{\operatorname{sech}(3\sin(10x)), \sin(9x)\}$	0.335619522	0.335614142
2	$\max\{\sin(20x), e^{x-1}\}$	0.387232183	0.387232967
3	$g(x)$	0.499870795	0.499870789
4	$\sqrt{\lvert x - 0.1\rvert}$	0.114682217	0.114679540

Table 4: Comparison of norms of the residue functions of approximations P_r^* determined by our implementation with results from [5]. In the table, P_{cheb}^* is the approximation determined in [5].

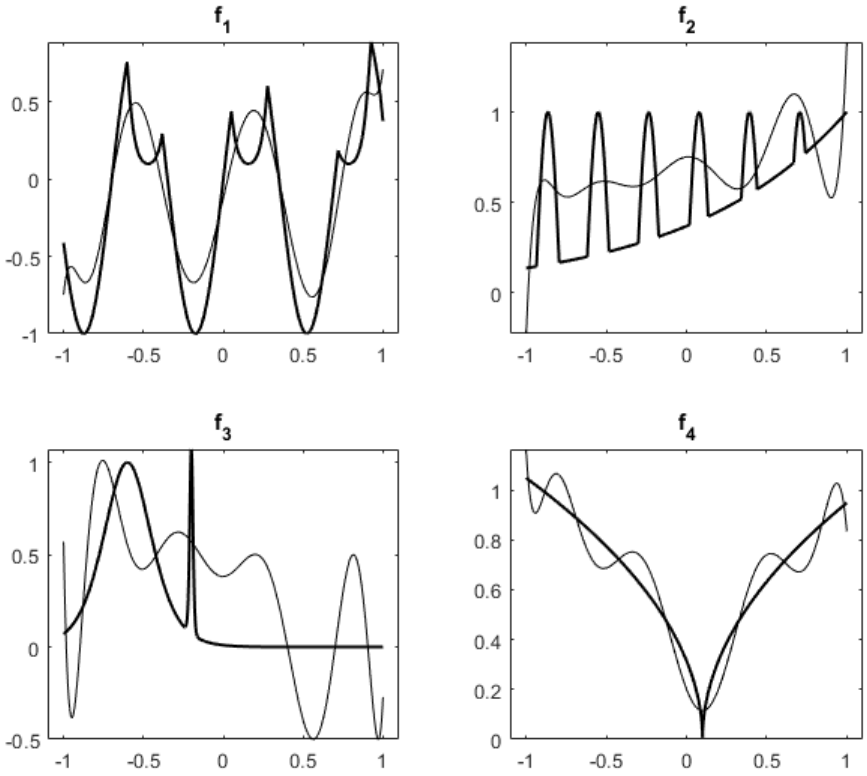

Figure 6: Degree 10 polynomial approximations for functions $f_1, \ldots, f_4$ as given by table 4. Again, the curve of the approximation is the thinner one. In the plot for f_3, there is another, sharper peak right of the one near the middle, which is not visible in the plot.

Approximations spanned by other Haar systems than the monomial basis It was mentioned earlier that the Remez algorithm can be used to find best approximations spanned by any system of continuous functions $\{g_1, \ldots, g_n\}$ as long as this system satisfies the Haar condition. In all the previous examples in this section our Haar system was the monomial basis $\{1, x, \ldots, x^n\}$.

An interesting comparison between different Haar systems would be to compare the number of basis functions which are needed to find approximations of a desired accuracy. We give a small example.

For the Haar systems

$$\begin{aligned} H_1 &= \{1, x, x^2, \ldots, x^n\} \\ H_2 &= \{1, e^{0.5x}, e^x, \ldots, e^{0.5nx}\} \\ H_3 &= \{1, e^x, e^{2x}, \ldots, e^{nx}\}, \end{aligned} \tag{16}$$

table 5 shows how many basis elements are needed from the bases H_1, H_2, H_3 in order to approximate the given functions f_i, $i = 1, \ldots, 6$ in such a way that $||r_i||_\infty < 0.05$, where r_i is the residue function corresponding to the approximation for f_i. The fact that $H2$ and $H3$ satisfy the Haar condition is mentioned in [7, p. 297].

It can be seen in table 5 that, for approximating $f_5(x) = \tan(0.45x)$, two less basis elements are needed in the basis H_3 compared to the monomial basis H_1. A plot of an approximation by exponential functions for f_5 is shown in figure 7. For $f_1, \ldots, f_4$ and f_6, the difference in number of basis elements needed from H_1, H_2, H_3 is at most 1.

For f_5, we check for the aforementioned bases how many basis elements are needed in order to satisfy $||r_5||_\infty < 0.005$, in order to see if the difference in number of basis elements needed from H_1, H_2, H_3 increases. We see from table 6 that the differences do not become more obvious in this case. It can be remarked how similar the bases H_1, H_2, H_3 are for these examples in terms of basis elements needed to obtain the prescribed quality of approximation.

i	$f_i(x)$	H_1	H_2	H_3
1	$\sqrt{x}$	4	5	5
2	$\log(x + 0.05)$	6	6	7
3	$e^x \cos(\pi x)$	5	4	4
4	$e^x \cos(2\pi x) \sin(2\pi x)$	9	9	9
5	$\tan(0.45\pi x)$	8	7	6
6	$(x+1)^{(x+1)^{(x+1)}}$	7	6	6

Table 5: Number of basis elements needed in Haar systems given by (16) to approximate f_i in such a way that $||r||_\infty < 0.05$ where r is the corresponding residue function.

i	$f_i(x)$	H_1	H_2	H_3
5	$\tan(0.45\pi x)$	11	10	9

Table 6: Number of basis elements needed in Haar systems given by (16) to approximate f_i in such a way that $||r||_\infty < 0.005$ where r is the corresponding residue function.

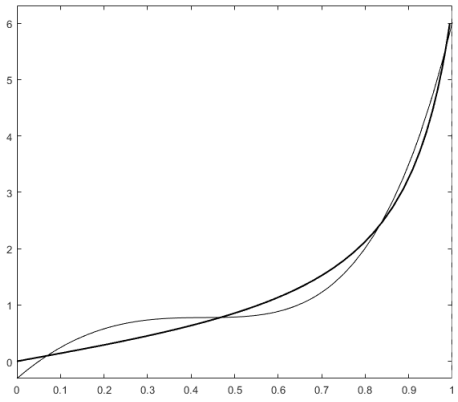

Figure 7: Approximation for $f_5(x) = \tan(0.45\pi x)$ spanned by the first four elements from Haar system H_2 given in (16).

6.3 Conclusion

We started by comparing results determined by our implementation with two analytically determined results and concluded that the results from the implementation were highly accurate in both cases. Moreover, we have obtained very reasonable results for determining polynomial approximations up to degree 10, as comparison with results from [5] showed. However, there are limitations to our implementation; two problems are discussed below.

- If the residue function contains sharp peaks, extrema may be missed in the brute-force search, or the value of $||r||_\infty$ may be determined inaccurately. In this case, the true value for $||r||_\infty$ may be much larger than the value determined by the brute-force search. Plotting the residue function can give us an indication whether we should trust our value for $||r||_\infty$. If higher accuracy is desired, a smaller step size for the brute-force search can be used.

- We saw an example where the implementation gave errors when trying to determine a degree 18 polynomial approximation. Using the monomial basis $\{1, x, \ldots, x^n\}$, the matrix in the linear system solved in the first step

of the algorithm, is the Vandermonde matrix. It is known that generally the condition number of this matrix increases exponentially as n increases [5, p. 728]. If the condition number is too high, the numerical solution of the linear system will be inaccurate. In the implementation described in [5], the problem of an increasing condition number is avoided by a suitable adjustment of the basis in each iteration. For details, see [5, p. 728].

7 Proof of convergence of the Remez algorithm

Our final goal is to give a proof of convergence of the Remez algorithm. It is known for many approximants the convergence of the Remez algorithm is quadratic [2, p. 98]. We will not prove this fact here, but instead conclude by presenting a proof of the linear convergence, which is valid for any continuous approximant. Before doing so, we will need to extend the theory presented in the first sections slightly more by proving the theorem of de La Vallée-Poussin and the strong unicity theorem. First we prove a short lemma about the Haar condition.

Lemma 3. *The system of continuous functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition if and only if no nontrivial generalized polynomial $\sum_{i=1}^{n} c_i g_i$ has more than $n-1$ distinct roots.*

Proof. ($\Longrightarrow$) Assume the Haar condition is satisfied. Then the matrix in the equation

$$\begin{bmatrix} g_1(x_1) & \ldots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \ldots & g_n(x_n) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{17}$$

is nonsingular whenever $x_1, \ldots, x_n$ are all distinct. In this case, the only solution is the trivial solution $c_1 = \cdots = c_n = 0$, that is, no nontrivial generalized polynomial can have n or more roots.
($\Longleftarrow$) By contraposition. Assume the Haar condition is not satisfied. Then there exist $x_1, \ldots, x_n$, all dinstinct, so that the matrix in equation (17) has rank $< n$. In this case, a nontrivial solution vector $[c_1, \ldots, c_n]^\intercal$ exists, hence the nontrivial generalized polynomial $\sum_{i=1}^{n} c_i g_i$ has the roots $x_1, \ldots, x_n$. □

7.1 Theorem of de La Vallée Poussin and the strong unicity theorem

The following theorem gives a lower bound for the largest deviation between the approximant and the best minimax approximation.

Theorem 12. *(de La Vallée Poussin) Assume the system of continuous functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition. Define*

$$E(f) = \inf ||P - f||_\infty,$$

where P ranges over all generalized polynomials $\sum_{i=1}^{n} c_i g_i$. Let P be a generalized polynomial such that $f - P$ is alternately positive and negative at $n+1$ consecutive points $x_i \in [a,b]$. Then

$$E(f) \geq \min_i |f(x_i) - P(x_i)|.$$

Proof. Assume for contradiction that

$$E(f) < \min_i |f(x_i) - P(x_i)|.$$

Then there exists a generalized polynomial P_0 so that

$$\max_{x\in[a,b]} |f(x) - P_0(x)| < \min_i |f(x_i) - P(x_i)|.$$

Now we write $P_0 - P = (f - P) - (f - P_0)$. From the inequality above it follows that the generalized polynomial $P_0 - P$ alternates in sign at the $n+1$ consecutive points x_i. But then $P_0 - P$ has n roots, contradicting the lemma we just proved. □

Theorem 13. *(Strong unicity) Assume the system of continuous functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition and let P^* be the best generalized polynomial approximation (spanned by $g_1, \ldots, g_n$) to $f \in C[a,b]$. Then there exists a constant $\gamma(f) > 0$ such that for all other generalized polynomials $P = \sum_{i=1}^{n} c_i g_i$, we have*

$$||f - P||_\infty \geq ||f - P^*||_\infty + \gamma ||P^* - P||_\infty.$$

Proof. First, let us consider the case where $||f - P^*||_\infty = 0$. Then we apply the triangle inequality to see

$$\begin{aligned} ||P - P^*||_\infty &= ||(f - P) - (f - P^*)||_\infty \\ &\leq ||f - P||_\infty + ||f - P^*||_\infty \\ &= ||f - P||_\infty. \end{aligned}$$

In this case we choose $\gamma = 1$.

Assume now that $||f - P^*|| > 0$. Let $r(x) = f(x) - P^*(x)$. Since P^* is assumed to be the best minimax approximation to f, the characterization theorem tells us that $\mathbf{0}$ is contained in the convex hull of the set

$$U = \{r(x)[g_1(x), \ldots, g_n(x)]^\intercal : |r(x)| = ||r||_\infty\}.$$

Hence, we may write

$$\mathbf{0} = \sum_{i=0}^{n} \theta_i r(x_i)[g_1(x_i), \ldots, g_n(x_i)]^\intercal,$$

with $\theta_i \geq 0$ and $\sum \theta_i = 1$. Now let $\sigma_i = \operatorname{sgn}(r(x_i))$ for $i = 0, \ldots, n$. After rescaling and possibly re-labeling, we may write

$$\mathbf{0} = \sum_{i=0}^{k} \theta_i \sigma_i [g_1(x_i), \ldots, g_n(x_i)]^\intercal$$

with $\theta_i > 0$. That is, an equation

$$0 = \sum_{i=0}^{k} \theta_i \sigma_i g_j(x_i)$$

holds for $j = 1, \ldots, n$. By the Haar condition, $k \geq n$. Caratheodory's theorem tells us that $k \leq n$ and we conclude $k = n$.

Let $Q(x) = \sum_{j=1}^{n} c_j g_j(x)$ be a generalized polynomial with norm 1. Then

$$\begin{aligned}
\sum_{i=0}^{n} \theta_i \sigma_i Q(x_i) &= \sum_{i=0}^{n} \theta_i \sigma_i \sum_{j=1}^{n} c_j g_j(x_i) \\
&= \sum_{j=1}^{n} c_j \sum_{i=0}^{n} \theta_i \sigma_i g_j(x_i) \\
&= 0.
\end{aligned}$$

By the Haar condition, lemma 3 tells us that the $\sigma_i Q(x_i)$ cannot all be zero. Hence, at least one of the $\sigma_i Q(x_i)$ must be strictly positive. This implies that $\max_i \sigma_i Q(x_i)$ is a strictly positive function of Q. The set

$$\{Q(x) = \sum_{j=1}^{n} c_j g_j(x) : ||Q||_\infty = 1\}$$

is compact as it is a closed and bounded subset of the finite dimensional linear space spanned by $\{g_1, \ldots, g_n\}$. Thus the number

$$\gamma = \min_{||Q||_\infty = 1} \max_i \sigma_i Q(x_i) \tag{18}$$

is strictly positive as it is the minimum of a strictly positive continuous function on a compact set. Now, let P be any generalized polynomial. There are two cases. If $P = P^*$, the inequality we want to prove follows trivially since in this case we may choose γ arbitrarily. Otherwise, the generalized polynomial

$$Q = \frac{P^* - P}{||P^* - P||_\infty}$$

has norm 1. By the definition of γ in (18), we have $\sigma_i Q(x_i) \geq \gamma$ for some index i. Consequently, for this index, $\sigma_i(P^* - P)(x_i) \geq \gamma ||P^* - P||_\infty$ and hence,

$$\begin{aligned}
||f - P||_\infty &\geq \sigma_i(f - P)(x_i) \\
&= \sigma_i(f - P^*)(x_i) + \sigma_i(P^* - P)(x_i) \\
&\geq ||f - P^*||_\infty + \gamma ||P^* - P||_\infty.
\end{aligned}$$

In the last step we used the fact that x_i is an element satisfying $f(x_i) - P^*(x_i) = r(x_i) = \pm ||r||_\infty$ and hence $\sigma_i r(x_i) = ||r||_\infty$. □

7.2 Proof of convergence

Theorem 14. *(Convergence of the Remez algorithm) Let P^k denote the generalized polynomial obtained in iteration k in the Remez algorithm and let P^* be the best minimax approximation spanned by the corresponding Haar system. Then an inequality of the form*

$$||P^k - P^*||_\infty \leq A\theta^k$$

with $0 < \theta < 1$ holds, which means that $P^k \to P^*$ uniformly.

Proof. We use the notation from the description of the Remez algorithm in section 5. At the end of iteration k, define

$$\alpha = \min_i |r(x_i)| = \max_i |r(x_i)|,$$

$$\beta = \max_i |r(y_i)| = ||r||_\infty,$$

$$\gamma = \min_i |r(y_i)|.$$

Notice here that the definition of α is justified because in the first step of the iteration, the best approximation on the reference $\{x_0, \ldots, x_n\}$ is computed. By the alternation theorem, the absolute values of the $r(x_i)$'s are all equal. Notice furthermore that in the last step of the iteration, an element $y \in [a, b]$ satisfying $r(y) = ||r||_\infty$ was included in the new reference $\{y_0, \ldots, y_n\}$, justifying the definition of β. The corresponding quantities obtained in the next iteration are denoted by α', β' and γ'.

Define $\beta^* = ||f - P^*||_\infty$, where P^* is the best minimax approximation as in the statement of the theorem. By the theorem of de La Vallée Poussin, $\beta^* \geq \gamma$. It is clear that $\beta \geq \beta^*$. From the definition of the new reference $\{y_0, \ldots, y_n\}$ it is furthermore clear that $\gamma \geq \alpha$. This gives us the following:

$$\alpha \leq \gamma \leq \beta^* \leq \beta. \tag{19}$$

In the beginning of the next iteration, the vector $c' = [c'_1, \ldots, c'_n]^\intercal$ minimizing

$$\max_i |f(y_i) - \sum_{j=1}^n c'_j g_j(y_i)|$$

is computed. By (13) in the proof of corollary 2, the coefficient vector c' is obtained by solving the linear system

$$(-1)^i h + \sum_{j=1}^n c'_j g_j(y_i) = f(y_i), \quad i = 0, \ldots, n$$

for the unknowns h and $c'_1, \ldots, c'_n$, where $h = r(y_0)$. The matrix of this system is nonsingular; in the proof of corollary 2 we observed that the solution vector

c' exists and is unique. The constant h is then determined uniquely by c'. We write this system as matrix-vector equation:

$$\begin{bmatrix} 1 & g_1(y_0) & \dots & g_n(y_0) \\ -1 & g_1(y_1) & \dots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^n & g_1(y_n) & \dots & g_n(y_n) \end{bmatrix} \begin{bmatrix} h \\ c'_1 \\ \vdots \\ c'_n \end{bmatrix} = \begin{bmatrix} f(y_0) \\ f(y_1) \\ \vdots \\ f(y_n) \end{bmatrix}.$$

Next, we use Cramer's rule to solve for h and obtain

$$h = \begin{vmatrix} f(y_0) & g_1(y_0) & \dots & g_n(y_0) \\ f(y_1) & g_1(y_1) & \dots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ f(y_n) & g_1(y_n) & \dots & g_n(y_n) \end{vmatrix} \div \begin{vmatrix} 1 & g_1(y_0) & \dots & g_n(y_0) \\ -1 & g_1(y_1) & \dots & g_n(y_1) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^n & g_1(y_n) & \dots & g_n(y_n) \end{vmatrix}.$$

Denote the minors corresponding to the column with ± 1 in the determinant in the denominator by M_i. The solution for h can then be written as

$$h = \frac{\sum_{i=0}^{n} f(y_i) M_i (-1)^i}{\sum_{j=0}^{n} M_j}. \tag{20}$$

If f itself in (20) is replaced by a generalized polynomial $P = \sum_{j=1}^{n} a_j g_j$, then the approximation on the reference $\{y_0, \dots, y_n\}$ is exact and hence $h = 0$, which means that

$$\sum_{i=0}^{n} P(y_i) M_i (-1)^i = 0.$$

This shows that in (20), the approximant $f(x)$ may be replaced by

$$r(x) = f(x) - \sum_{j=1}^{n} c'_j g_j(x),$$

leaving h unchanged. We use the fact that the $r(y_i)$'s alternate in sign to see that

$$\sum_{i=0}^{n} r(y_i)(-1)^i M_i = \pm \sum_{i=0}^{n} |r(y_i)| M_i. \tag{21}$$

Furthermore, because $y_0 < \dots < y_n$ and because the Haar condition is satisfied by $\{g_1, \dots, g_n\}$, lemma 1 tells us that all the minors M_i have the same sign. Hence, combining (21) with the expression for h in (20), we find that

$$\alpha' = |h| = \frac{\sum_{i=0}^{n} |M_i||r(y_i)|}{\sum_{j=0}^{n} |M_j|}. \tag{22}$$

Now let

$$\theta_i = \frac{|M_i|}{\sum_{j=0}^{n} |M_j|}. \tag{23}$$

Notice here that $\theta_i \in (0,1)$ because there are at least two minors. Combine (22) and (23) to obtain

$$\begin{aligned} \alpha' &= \sum_{i=0}^{n} \theta_i |r(y_i)| \\ &\geq \sum_{i=0}^{n} \theta_i \min_j |r(y_j)| \\ &= \sum_{i=0}^{n} \theta_i \gamma \\ &= \gamma \end{aligned} \tag{24}$$

where we used in the last step the fact that $\sum_{i=0}^{n} \theta_i = 1$.

Assume for now that throughout the iterations of the algorithm, the numbers θ_i remain larger than a fixed constant $1-\theta > 0$. This assumption will be justified in a lemma *after* this proof. We establish the following inequalities.

$$\begin{aligned} \gamma' - \gamma &\geq \alpha' - \gamma && \text{using } \gamma' \geq \alpha' \text{ from (24)} \\ &= \sum_{i=0}^{n} \theta_i (|r(y_i)| - \gamma) && \text{using } \sum \theta_i = 1 \\ &\geq \min_i \theta_i (\beta - \gamma) && \text{where } \beta = \max_i |r(y_i)| \\ &\geq (1-\theta)(\beta - \gamma) && \text{because } \theta_i \geq 1-\theta \text{ for all } i \\ &\geq (1-\theta)(\beta^* - \gamma) && \text{since } \beta \geq \beta^* \text{ by (19).} \end{aligned} \tag{25}$$

Furthermore,

$$\begin{aligned} \beta^* - \gamma' &= (\beta^* - \gamma) - (\gamma' - \gamma) \\ &\leq (\beta^* - \gamma) - (1-\theta)(\beta^* - \gamma) && \text{using } \gamma' - \gamma \geq (1-\theta)(\beta^* - \gamma) \\ &= \theta(\beta^* - \gamma). \end{aligned}$$

We label the values of γ and β in iteration k as $\gamma^{(k)}$ and $\beta^{(k)}$, respectively. Applying the inequality above k times, we find that

$$\begin{aligned} \beta^* - \gamma^{(k)} &\leq \theta^k (\beta^* - \gamma^{(0)}) \\ &= B\theta^k, \end{aligned}$$

where B is a nonnegative constant. We establish another inequality.

$$\begin{aligned}
\beta^{(k)} - \beta^* &\leq \beta^{(k)} - \gamma^{(k)} && \text{notice that } \gamma^{(k)} \leq \beta^* \\
&\leq \frac{1}{1-\theta}(\gamma^{(k+1)} - \gamma^{(k)}) && \text{by (25)} \\
&\leq \frac{1}{1-\theta}(\beta^* - \gamma^{(k)}) && \text{because } \gamma^{(k)} \leq \beta^* \\
&\leq \frac{1}{1-\theta} B\theta^k \\
&= C\theta^k && \text{with } C \text{ nonnegative.}
\end{aligned}$$

Lastly, we apply the strong unicity theorem, which tells us that there exists a constant $\gamma > 0$ such that

$$||f - P^*||_\infty + \gamma||P^* - P^k||_\infty \leq ||f - P^k||_\infty.$$

We complete the proof by establishing the following inequality:

$$\begin{aligned}
||P^* - P^k||_\infty &\leq \frac{||f - P^k||_\infty - ||f - P^*||_\infty}{\gamma} \\
&= \frac{\beta^{(k)} - \beta^*}{\gamma} \\
&\leq \frac{C}{\gamma}\theta^k \\
&= A\theta^k,
\end{aligned}$$

where A is a nonnegative constant.

□

7.3 Completing the argument

As we mentioned in the convergence proof, we still need to prove that the numbers θ_i are bounded from below by a strictly positive constant. Before proving this as a lemma, we mention the following famous theorem from approximation theory together with a generalized version of that theorem. The latter theorem will be used in the proof of the lemma.

Theorem 15. *Let the pairs $(x_i, y_i) \in \mathbb{R}^2$, $i = 0, \ldots, n$ be given. Then there exists a unique polynomial p of degree $\leq n$ so that $P(x_i) = y_i$ for $i = 0, \ldots, n$.*

One of the proofs of this theorem is found in [2, p. 58] and depends on the fact that the Vandermonde matrix has a nonzero determinant. Under the Haar condition on $\{g_1, \ldots, g_n\}$, we have instead of the Vandermonde matrix the matrix

$$G(x_1, \ldots, x_n) = \begin{bmatrix} g_1(x_1) & \ldots & g_n(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_n) & \ldots & g_n(x_n) \end{bmatrix}$$

with nonzero determinant, generalizing the theorem above to the following.

Theorem 16. *Let the pairs $(x_i, y_i) \in \mathbb{R}^2$, $i = 1, \ldots, n$ be given and assume the system of continuous functions $\{g_1, \ldots, g_n\}$ satisfies the Haar condition. Then there exists a unique generalized polynomial*

$$P(x) = \sum_{i=1}^{n} c_i g_i(x)$$

satisfying $P(x_i) = y_i$ for $i = 1, \ldots, n$.

Lemma 4. *The numbers*

$$\theta_i = \frac{|M_i|}{\sum_{j=0}^{n} |M_j|},$$

as defined in the proof theorem 14, are bounded away from 0.

Proof. By the Haar condition, $M_i \neq 0$ for all i. We will prove that $|M_i|$ is bounded away from zero by showing that in iteration k, we have an inequality

$$y_{i+1}^{(k)} - y_i^{(k)} \geq \epsilon > 0, \quad i = 0, \ldots, n, \tag{26}$$

where ϵ does not depend on k. By continuity of the determinant as a function of the y_i and by the Haar condition, this will then imply that in fact $|M_i|$ is bounded away from 0. We make furthermore the observation that

$$\sum_{j=0}^{n} |M_j| \leq R$$

for some $R \in \mathbb{R}^+$ because $\sum_{j=0}^{n} |M_j|$ is a continuous function from a compact subset of $\mathbb{R}^n$ to $\mathbb{R}$ and is hence bounded on this set. Proving that $|M_i|$ is bounded away from 0 will therefore prove that θ_i is bounded away from 0.

In the first iteration $k = 0$ it is clear from step 3 of the Remez algorithm that inequality (26) is valid for some $\epsilon > 0$. From now on assume that $k > 0$.

Assume for contradiction that inequality (26) is not valid. In that case, the sequence

$$[y_0^{(k)}, \ldots, y_n^{(k)}]$$

contains a subsequence converging to a point $[y_0^*, \ldots, y_n^*]$ in which $y_i^* = y_{i+1}^*$ for some i, because the original sequence lives in a closed and bounded subset of $\mathbb{R}^{n+1}$. Let P be the generalized polynomial of best approximation for f on $[y_0^*, \ldots, y_n^*]$. By theorem 16,

$$P(y_i^*) = f(y_i^*), \quad i = 0, \ldots, n. \tag{27}$$

Equation (22) tells us that $\alpha^{(1)} > 0$. By continuity of P and f, there exists a number $\epsilon > 0$ so that

$$|(P - f)(x_1) - (P - f)(x_2)| < \alpha^{(1)}. \tag{28}$$

whenever $|x_1 - x_2| < \epsilon$ and $x_1, x_2 \in [a, b]$. Choose $k \in \mathbb{N}$ so that

$$|y_i^{(k)} - y_i^*| < \epsilon, \quad i = 0, \ldots, n.$$

Then,

$$\begin{aligned} |(P - f)(y_i^{(k)}) - (P - f)(y_i^*)| &= |P(y_i^{(k)}) - f(y_i^{(k)})| && \text{by (27)} \\ &< \alpha^{(1)} && \text{by (28)} \end{aligned}$$

$$\implies \alpha^{(k+1)} = \min_{P = \sum c_i g_i} \max_{i=0,\ldots,n} |P(y_i^{(k)}) - f(y_i^{(k)})| < \alpha^{(1)},$$

contradicting the fact that $\alpha^{(k)}$ increases monotonically. The fact that $\alpha^{(k)}$ increases monotonically follows from inequality (24); the latter inequality shows that $\alpha' \geq \gamma$. It was furthermore observed in the beginning of the convergence proof that $\gamma \geq \alpha$, meaning that $\alpha' \geq \alpha$. $\square$

8 Conclusion

The goals for this thesis were to present relevant theory on minimax approximation, to explain how the Remez algorithm works and to create examples with a relatively simple self-built implementation of the algorithm.

We started by presenting several results on convexity which we used later on to prove the characterization theorem and ultimately the alternation theorem. The latter theorem plays a vital role in the mechanism of the Remez algorithm and its corollary is used explicitly in the first step of each iteration to compute the best minimax approximation on a reference. We have seen that in the remaining steps of an iteration, the reference is adjusted in order to obtain a better approximation in the next iteration. We included a plot, created by our implementation, to illustrate the reference adjustment.

In section 6, the focus was on creating examples with our implementation to illustrate what the implementation can and cannot do. We have seen that ultimately, for high degree polynomial approximations, the algorithm fails due to an increasing condition number in the matrix involved in the program. However, through examples, we have seen satisfactory results for polynomial approximations up to degree 10, even for functions with many oscillations, locally high derivatives and points at which the functions are not differentiable. For approximating these more 'difficult' functions we used a finer grid in the brute-force search involved in the program, in order to locate extrema of the residue function in a more accurate way.

We have seen that our implementation can be used to find approximations spanned by other Haar systems than the monomial basis, as was we did at the end of section 6.2 in order to make a comparison between three Haar systems.

We discussed that in extreme cases, the norm of the residue function determined using the brute-force search may be inaccurate. Making use of the mean value theorem we discussed that in case the residue function r has high values for $|r'|$ near an extremum, the actual value of $||r||_\infty$ may be much higher than the value determined using the brute-force search. In extreme cases, extrema could be missed by the brute-force search.

We have mentioned that high values for $|r'|$ near an endpoint of the interval of approximation do not cause problems for our implementation because our brute-force method is designed to evaluate r at these endpoints. We should however, be more careful when approximating a function showing sharp peaks in the plot, which may result in similar behaviour in the plot of the residue function and give high values for $|r'|$ near extrema of r. In this case we can refine the grid for the brute-force search accordingly, as we have done in the degree 10 polynomial approximation examples in section 6.2.

Difficulties in determining minimax approximations by (generalized) polynomials with the Remez algorithm are only of a computational nature; we concluded by presenting a proof of linear convergence of the algorithm, which guarantees convergence for any approximant as long as it is continuous on the interval of approximation.

A Technical detail

Lemma 5. *Let X be a compact metric space and let the functions $r, g_1, \ldots, g_n : X \to \mathbb{R}$ be continuous. Then the set*

$$U = \{r(x)\hat{x} : |r(x)| = ||r||_\infty\},$$

with $\hat{x} = [g_1(x), \ldots, g_n(x)]^\intercal$ and $x \in X$, is compact

Proof. We will show that U is compact by showing that it is a sequentially compact subset of $\mathbb{R}^n$. Let (u_k) be any sequence in U. Then for all $k \in N$ there exists an $x_k \in X$ such that $u_k = r(x_k)\hat{x}_k$ where $\hat{x}_k = [g_1(x_k), \ldots, g_n(x_k)]^\intercal$. Since (x_k) is a sequence in the compact, therefore sequentially compact, metric space X, (x_k) has a convergent subsequence, say (x_{k_j}), with limit $x^* \in X$. Then, by continuity of r,

$$\lim_{j\to\infty} r(x_{k_j}) = r(x^*), \tag{29}$$

which implies that $|r(x^*)| = ||r||_\infty$. By continuity of $g_1, \ldots, g_n$, we have

$$\lim_{j\to\infty} \hat{x}_{k_j} = [g_1(x^*), \ldots, g_n(x^*)]^\intercal := \hat{x}^*. \tag{30}$$

Combining (29) and (30) we find that

$$\lim_{j\to\infty} r(x_{k_j})\hat{x}_{k_j} = r(x^*)\hat{x}^*$$

with $|r(x^*)| = ||r||_\infty$, proving that $r(x^*)\hat{x}^* \in U$. This proves the sequential compactness of U.

□

B Adjustment of the reference

We describe how one of the y_i should be removed in the last step of an iteration in the Remez algorithm, following a strategy originally proposed by Remez himself [5, sec 2.2, 3.5]. In step 3 of the algorithm the set $\{y_0, \ldots, y_n\}$ was obtained where the y_i are in ascending order and where

$$\operatorname{sgn}(r(y_{i-1})) = -\operatorname{sgn}(r(y_i)), \quad i = 1, \ldots, n.$$

In words, r alternates in sign on the set $\{y_0, \ldots, y_n\}$. At the end of the first step of the present iteration, we found an element $y \in [a, b]$ satisfying

$$|r(y)| = \max_{x\in[a,b]} |r(x)|.$$

We included this point y in the set $\{y_0, \ldots, y_n\}$ on the right place so that the resulting set is still in ascending order. Our goal is to remove one of the y_i in such a way that r still alternates in sign on the resulting set. We will describe how this can be acomplished right now.

Case 1: Suppose that $y_0 < y < y_n$. In this case, the new set has two neighbouring points on which $r(x)$ has the same sign. We will keep the one of these two on which $|r(x)|$ has the largest value. This case may look as follows. After inserting y into $\{y_0, \ldots, y_n\}$, we have the following set:

$$\{y_0, \ldots, y_{i-1}^+, y^+, y_i^-, \ldots, y_n\},$$

where elements on which $r(x)$ is positive or negative are indicated by a superscript " $+$ " or " $-$ ", respectively. We have $|r(y)| > |r(y_{i-1})|$ and will therefore remove y_{i-1}. On the new set $\{y_0, \ldots, y_{i-2}^-, y^+, y_i^-, \ldots, y_n\}$, $r(x)$ alternates in sign. This set is the reference with which we start the next iteration.

Case 2: Suppose that $a \leq y < y_0$. Two situations may occur; if the new set has two neighbouring points on which $r(x)$ has the same sign, keep the point on which $|r(x)|$ has the largest value, as in the first case. This will give the desired reference. Otherwise, r is already alternating on the new set and we set

$$y_n = y_{n-1}, y_{n-1} = y_{n-2}, \ldots, y_0 = y,$$

resulting in an ordered set of $n+1$ points on which $r(x)$ is alternating. The result is that y is included in the set and the old value y_n is removed.

Case 3: Suppose that $y_n < y \leq b$. A point y_i to be removed can be chosen in a way analogous to the way described in the second case. In our Matlab implementation of the Remez algorithm, the correct element y_i is removed using a chain of if-statements.

C Matlab codes

In this section, the Matlab code for computing approximations by the Remez algorithm is given. The function `Remez2.m` executes one iteration of the Remez algorithm. The function `RemezExample.m` uses the function `Remez2.m` in a while-loop to execute the Remez algorithm. We leave the while-loop when the stopping criterion, introduced in the statement of the Remez algorithm, is satisfied. The function `Remez2.m` depends on several other functions, for example for finding extrema of the residue function. All these other functions are given below, ordered according to when they are used in `Remez2.m`.

The script below shows how the codes can be used to find a linear approximation for e^x on the interval $[0, 1]$.

```
f = @(x) exp(x);                % approximant
a=0;
b=1;
ref = linspace(a,b,3);          % equispaced nodes
basis = @(x) [1 x];             % we approximate f by
                                % a linear function
epsilon = 0.0001;
[approx,normr] = RemezExample(f,ref,basis,a,b,epsilon)

approx =

    @(x)dot(basis(x),lambda)


normr =

   0.105933415992418
```

Executing the algorithm `RemezExample.m`

```
%Computes approximation approx for function f on interval
    [a,b] and
%gives the infinity norm of the residue function, normr
    using Remez
%algorithm. Stops when delta <= epsilon.
%Input:     approximant f
%           initial reference ref
%           basis for Haar system basis
%           endpoints a,b of the interval [a,b]
%           epsilon for stopping criterion
%Output:    approximation for f approx
%           infinity norm of residue function normr
function [approx,normr] = RemezExample(f,ref,basis,a,b,
    epsilon)
```

```
delta = epsilon+1;      %Initialize delta, it must be > epsilon
while delta > epsilon
[ymaxr,r,newref,approx,delta]=Remez2(f,ref,basis,a,b);
ref = newref;           %reference for new iteration
end
normr = abs(r(ymaxr));  %compute infinity norm of r
```

Single iteration of the Remez algorithm `Remez2.m`

```
%Executes one iteration of the Remez algorithm.
%Input:     approximant f
%           initial reference ref
%           basis for Haar system basis
%           endpoints a,b of the interval [a,b]
%Output:    location of max of |r| ymaxr
%           residue function r
%           reference for next iteration refnew
%           approximation for f approx
%           delta for stopping criterion


function[ymaxr,r,newref,approx,delta] ...
= Remez2(f,ref,basis,a,b)

%compute best minimax approx on reference
[lambda] = minimaxsol2(f,ref,basis);

%define approximation and residue functions
approx = @(x) dot(basis(x),lambda);
r=@(x) f(x)- dot(basis(x),lambda);

h=r(ref(1));
[ymaxr] = findmaxr2(r,a,b);     %find location of max of r
normr = abs(r(ymaxr));          %compute ||r||
delta = abs(normr - abs(h));    %compute delta for stop crit
s=size(ref);
n=s(2)-1;

%Create vector z with roots
z(1) = a;
for i = 2:n+1
    z(i)=fzero(r,[ref(i-1) ref(i)]);
end
```

```
z(n+2)=b;

for i = 1:n+1                          %create trial set {y_0
    ,..., y_n}
    sigma = sign(r(ref(i)));
    newref(i) = fmaxsigmar(sigma,r,z(i), z(i+1),ref(i),a,
        b);
end

if a<=ymaxr && ymaxr<newref(1)         %inserting ymaxr
    into the reference
    if sign(r(ymaxr)) == sign(r(newref(1)))
        newref(1)=ymaxr;
    else
        for i = n+1:-1:2
        newref(i)=newref(i-1);
        end
    newref(1)=ymaxr;
    end
elseif newref(n+1)<=ymaxr && ymaxr<=b
    if sign(r(ymaxr)) == sign(r(newref(n+1)))
        newref(n+1) = ymaxr;
    else
        for i = 1:n
            newref(i)=newref(i+1);
        end
        newref(n+1)=ymaxr;
    end
else
    for i = 2:n+1
    if newref(i-1)<=ymaxr && ymaxr<newref(i)
        if sign(r(ymaxr)) == sign(r(newref(i-1)))
            newref(i-1) = ymaxr;
        elseif sign(r(ymaxr))== sign(r(newref(i)))
            newref(i) = ymaxr;
        end
    end
    end
end
end
```

Solving linear system to obtain best approximation on a reference `minimaxsol2.m`

```
%Solves linear system to obtain best approximation on
    reference ref
```

```
function [lambda] = minimaxsol2(f,ref,basis)
s=size(ref);
n=s(2)-1;

%Construct right hand side vector
for i = 2:(n+1)
    b(i-1)=f(ref(i))-power(-1,i-1)*f(ref(1));
end

%Construct matrix A
vector2=basis(ref(1));
for i=1:n
    vector1=basis(ref(i+1));
    for j=1:n
        A(i,j)=vector1(j)-power(-1,i)*vector2(j);
    end
end
b=transpose(b);
%Solve Ax=b
lambda = A\b;
```

Brute-force search for absolute maximum of residue function `findmaxr2.m`

```
%Brute force search for x-coordinate of maximum of r
function [ymaxr] = findmaxr2(r,a,b)
stepsize = (b-a)/10000;

%Create vector with function values
for i=1:10001
    valvector(i) = abs(r(a+stepsize*(i-1)));
end

%Find location of largest value
maxval = max(valvector);
for i=1:10001
    if valvector(i) == maxval
        ymaxr = a+stepsize*(i-1);
    end
end
end
```

Finding local maxima `fmaxsigmar.m`

```
%Finds desired local maxima of sigma*r
function [maxsol] = fmaxsigmar(sigma,r,lb,ub,xi,a,b)
f = @(x) -1*sigma*r(x);
```

```
4  maxsol = fminbnd(f,lb,ub);
5
6  %In case a local maximum was found with sigma*r(x_i)>
       sigma*r(y_i)
7  %we start a brute force search for a desired maximum
8  if sigma*r(xi)>sigma*r(maxsol)
9       stepsize = (ub-lb)/10000;
10      for i=1:10001
11          valvector(i) = sigma*r(lb+stepsize*(i-1));
12      end
13      maxval = max(valvector);
14      for i=1:10001
15          if valvector(i) == maxval
16              maxsol = lb+stepsize*(i-1);
17          end
18      end
19 end
20
21 %In case sigma*r(x_i) was already maximal
22 if sigma*r(xi)>sigma*r(maxsol)
23      maxsol=xi;
24 end
25
26 %Test values at endpoints of [a,b]
27 if lb == a && sigma*r(a)>sigma*r(maxsol)
28      maxsol=a;
29 end
30 if ub == b && sigma*r(b)>sigma*r(maxsol)
31      maxsol=b;
32 end
33 end
```

References

[1] M.J.D. Powell. *Approximation theory and methods*. Cambridge University Press, United States of America, 1981.

[2] E.W. Cheney. *Approximation Theory*. Chelsea Publishing Company, United States of America, second edition, 1982.

[3] N.L. Carothers. A Short Course on Approximation Theory. Bowling Green State University, 2009. Available online.

[4] H.S.V. de Snoo and A.E. Sterk. Functional Analysis An Introduction, January 2017.

[5] R. Pachón and L.N. Trefethen. Barycentric-Remez algorithms for best polynomial approximation in the chebfun system. *L.N. Bit Numer Math*, 49:721–741, 2009.

[6] H. Van de Vel. Haar intervals for trigonometric polynomials. *Journal of Computational and Applied Mathematics*, 5(4):265–267, 1979.

[7] C.B. Dunham. Families satisfying the Haar condition. *Journal of Approximation Theory*, 12:291–298, 1974.

[8] J.T. Lewis. Restricted Range Approximation and Its Application to Digital Filter Design. *Mathematics of Computation*, 29(130):522–539, April 1975.

[9] R.A. DeVore and G.G. Lorentz. *Constructive Approximation*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, United States of America, 1993.

[10] J. Berrut and L.N. Trefethen. Barycentric Lagrange Interpolation. *Society for Industrial and Applied mathematics*, 46(3):501–517, 2004.

[11] H.H. Denman. Minimax Polynomial Approximation. *Mathematics of Computation*, 20(94):257–265, 1966.

[12] S.J. Leon. *Linear Algebra with Applications*. Pearson Education Limited, United States of America, eighth edition, 2014.

[13] S. Abbott. *Understanding Analysis*. Springer Science+Business Media, Inc., United States of America, 2001.