ON THE FOURIER REPRESENTATION OF COMPUTABLE CONTINUOUS SIGNALS

Holger Boche and Ullrich J. Mönich

Technische Universität München
Lehrstuhl für Theoretische Informationstechnik

ABSTRACT

In this paper we study whether it is possible to decide algorithmically if the Fourier series of a continuous function converges uniformly. We show that this decision cannot be made algorithmically, because there exists no Turing machine that can decide for each and every continuous functions whether its Fourier series converges uniformly. Turing computability describes the theoretical feasible that can be implemented on a digital computer, hence the result shows that there exists no algorithm that can perform this decision.

***Index Terms*—** Turing computability, uniform convergence, Fourier series, continuous function

1. INTRODUCTION

In many signal processing applications the approximation of complicated functions by a small number of simple functions plays an important role [1–6]. There are numerous possibilities for the approximation processes, and many of them have proven to be useful, e.g., spline approximations [2], approximations for bandlimited signals using the Shannon sampling series [3], approximations with sparse signals [5] or wavelets [7–10], and Fourier series approximations [11]. Many of these approaches use orthonormal or biorthogonal systems [12]. Depending on the application, different signal representations have been proposed with properties that are tailored to the specific problem at hand [13, 14].

In this paper we consider the approximation of continuous periodic functions by Fourier series. Fourier series are widely used in signal theory, e.g., in the design of FIR filters [15]. In order to apply them practically, it is important to know their approximation behavior. The Fourier series of a 2π-periodic function f is given by

$$\sum_{k=-\infty}^{\infty} c_k(f)\,\mathrm{e}^{ikt}, \quad t \in [-\pi,\pi), \tag{1}$$

where

$$c_k(f) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)\,\mathrm{e}^{-ikt}\,\mathrm{d}t, \quad k \in \mathbb{Z}, \tag{2}$$

are the Fourier coefficients. For continuously differentiable functions, the Fourier series (1) is known to converge for all $t \in [-\pi,\pi)$. Further, for absolutely continuous functions the Fourier series (1) even converges uniformly. For many practical applications these simple sufficient conditions for convergence are either not satisfied or difficult to verify. Moreover, in the mathematical literature it is well-known that there exist continuous functions such that (1) diverges at some point $t \in [-\pi,\pi)$ [16, p. 72].

H. Boche was supported by the German Ministry of Education and Research (BMBF) within the national initiative for molecular communication under Grant 16KIS0914 and U. Mönich by the German Research Foundation (DFG) under grant BO 1734/20-1.

From an application's point of view it would be very helpful to have some criterion based on which it is possible to decide whether, for a given continuous function, the Fourier series (1) converges uniformly or not. In this paper we study if this question can be answered algorithmically, i.e., if we can find an algorithm that takes any computable continuous functions as an input and decides whether the Fourier series of this function converges uniformly. The existence of such an algorithm would be of importance for the computer-based signal and system design. Note that we restrict the class of continuous functions by requiring that they are computable, i.e., that they can be described algorithmically.

The proper framework to study this question is Turing computability. A Turing machine is an abstract device that manipulates symbols on a strip of tape according to certain rules [17–20]. Although the concept is very simple, a Turing machine is capable of simulating any given algorithm. Turing machines have no limitations with respect to memory or computing time, and hence provide a theoretical model that describes the fundamental limits of any practically realizable digital computer.

We will show that there exists no Turing machine, and hence no algorithm, that always can decide whether the Fourier series of a continuous computable function converges uniformly.

2. COMPUTABILITY

Before we present our main result, we review some basics of computability theory. Alan Turing introduced the concept of a computable real number in [17, 18]. A sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ is called computable sequence if there exist recursive functions a, b, s from $\mathbb{N}$ to $\mathbb{N}$ such that $b(n) \neq 0$ for all $n \in \mathbb{N}$ and

$$r_n = (-1)^{s(n)}\frac{a(n)}{b(n)}, \qquad n \in \mathbb{N}.$$

A recursive function is a function, mapping natural numbers into natural numbers, that is built of simple computable functions and recursions [21]. Recursive functions are computable by a Turing machine. A real number x is said to be computable if there exists a computable sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ such that $|x - r_n| < 2^{-n}$ for all $n \in \mathbb{N}$. By $\mathbb{R}_c$ we denote the set of computable real numbers. $\mathbb{R}_c$ is a field, i.e., finite sums, differences, products, and quotients of computable numbers are computable. A non-computable real number was for example constructed in [22].

There are several ways to define computability of functions, most importantly Turing/Borel computability, Markov computability, and Banach–Mazur computability. Out of these three, Banach–Mazur computability is the weakest form of computability, because any function that is computable with respect to the two other definitions is always Banach–Mazur computable. Conversely, any function that is not Banach–Mazur computable cannot be Turing

computable. A function $f\colon \mathbb{R}_c \to \mathbb{R}_c$ is called Banach–Mazur computable if f maps any given computable sequence $\{x_n\}_{n\in\mathbb{N}}$ of real numbers into a computable sequence $\{f(x_n)\}_{n\in\mathbb{N}}$ of real numbers. We will further use the important fact that every computable real function is continuous on $\mathbb{R}_c$ [23]. For a more detailed treatment of computability, see for example [19, 20, 23, 24], and for an example of a non-computable function [25].

By $C(\mathbb{T})$ we denote the space of all continuous 2π-periodic functions, equipped with the norm $\|f\|_{C(\mathbb{T})} = \max_{t\in[-\pi,\pi)}|f(t)|$. A function $f \in C(\mathbb{T})$ is computable in $C(\mathbb{T})$ if there exists a computable sequence $\{g_n\}_{n\in\mathbb{N}}$ of trigonometric polynomials such that $\|f - g_n\|_{C(\mathbb{T})} < 2^{-n}$ for all $n \in \mathbb{N}$. Let $C_{\mathrm{c}}(\mathbb{T})$ denote the set of all computable functions in $C(\mathbb{T})$. Note that $C_{\mathrm{c}}(\mathbb{T})$ has a linear structure.

3. MAIN RESULT

Before we state our main result, we introduce several abbreviations. Let

$$(S_N f)(t) = \sum_{k=-N}^{N} c_k(f)\,\mathrm{e}^{ikt}, \quad t \in [-\pi,\pi),$$

denote the N-th partial sum of the Fourier series, where the Fourier coefficients $c_k(f)$ are given by (2). There exist functions $f \in C(\mathbb{T})$ for which the Fourier series of f converges uniformly, i.e., we have

$$\lim_{N\to\infty} \|f - S_N f\|_{C(\mathbb{T})} = 0. \tag{3}$$

But there exist also functions $f \in C(\mathbb{T})$ for which the Fourier series of f does not converge uniformly to f, i.e., we have

$$\limsup_{N\to\infty} \|f - S_N f\|_{C(\mathbb{T})} > 0. \tag{4}$$

It can be shown that the set of functions $f \in C(\mathbb{T})$ that satisfy (4) is large, where the "size" can be characterized by different measures [26]. Let

$$\mathcal{U}_{\mathrm{c}}(\mathbb{T}) = \Big\{ f \in C_{\mathrm{c}}(\mathbb{T})\colon \lim_{N\to\infty} \|f - S_N f\|_{C(\mathbb{T})} = 0 \Big\}$$

denote the set of all functions $f \in C_{\mathrm{c}}(\mathbb{T})$ for which the Fourier series converges uniformly.

Remark 1. There exist infinitely many functions in $\mathcal{U}_{\mathrm{c}}(\mathbb{T})$. For example, all trigonometric polynomials with rational coefficients are in $\mathcal{U}_{\mathrm{c}}(\mathbb{T})$.

Clearly, it would be helpful to have an algorithm that can decide whether for a given $f \in C(\mathbb{T})$, we have (3) or (4). In order to be in a meaningful setting, we need to restrict ourselves to computable continuous functions $f \in C_{\mathrm{c}}(\mathbb{T})$, i.e., functions that can be described algorithmically. Note that, for $f \in C_{\mathrm{c}}(\mathbb{T})$, the Fourier coefficients defined by (2) and consequently the partial sums $S_N f$ are computable [24, Th. 5, p. 35]. Although, for $f \in C_{\mathrm{c}}(\mathbb{T})$, we have this computable sequence of very simple functions $\{S_N f\}_{N\in\mathbb{N}}$, we cannot always approximate f with this sequence, because there exist computable functions $C_{\mathrm{c}}(\mathbb{T})$ for which the Fourier series diverges. This will be shown in Lemma 1.

We ask whether it is possible to characterize the set $\mathcal{U}_{\mathrm{c}}(\mathbb{T})$ algorithmically. That is, we ask if there exists an algorithmic condition, which is necessary and sufficient for $f \in C_{\mathrm{c}}(\mathbb{T})$ to be in $\mathcal{U}_{\mathrm{c}}(\mathbb{T})$. The following theorem answers this question in the negative.

Theorem 1. *There exists no Turing machine that can decide for all $f \in C_c(\mathbb{T})$ whether $f \in \mathcal{U}_c(\mathbb{T})$.*

Theorem 1 shows that any algorithm that is forced to give a decision after a finite amount of time, needs to give wrong answers for some functions.

Before we present the proof of Theorem 1, we need to further formalize the problem and phrase it in the terminology of computability. Let

$$M\colon C_{\mathrm{c}}(\mathbb{T}) \to \{\text{"yes"}, \text{"no"}\}$$

denote the mapping that characterizes whether $f \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$, i.e., the mapping that satisfies

$$Mf = \text{"yes"} \quad \Leftrightarrow \quad f \in \mathcal{U}_{\mathrm{c}}(\mathbb{T}).$$

We will study the weakest form of computability here, namely Banach–Mazur computability. To this end, we encode "yes" with 1 and "no" with 0. Moreover, every computable function $f \in C_{\mathrm{c}}(\mathbb{T})$ is described by an algorithm P_f. The question now is: Does there exist a Turing machine TM with $TM(P_f) = 1$ if and only if $f \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$? For simplicity, we do not distinguish between a computable function f and the algorithm P_f describing it in the following.

For the proof of our main result, we need two lemmas.

Lemma 1. *We can construct a computable function $f_* \in C_c(\mathbb{T})$ such that*

$$\limsup_{N\to\infty} \|S_N f_*\|_{C(\mathbb{T})} = \infty. \tag{5}$$

We postpone the proof of Lemma 1 until the end of this section.

Remark 2. We actually show more than what is stated in Lemma 1. We construct a set of linearly independent computable functions $\{g_l\}_{l\in\mathbb{N}} \subset C_{\mathrm{c}}(\mathbb{T})$ and, for each $l \in \mathbb{N}$, two Turing computable functions $N_l\colon \mathbb{N} \to \mathbb{N}$ and $U_l\colon \mathbb{N} \to \mathbb{R}_c$ with $\lim_{k\to\infty} N_l(k) = \infty$ and $\lim_{k\to\infty} U_l(k) = \infty$, such that for the computable numbers $\|S_{N_l(k)} g_l\|_{C(\mathbb{T})}$ we have $\|S_{N_l(k)} g_l\|_{C(\mathbb{T})} > U_l(k)$. In other words, the divergence behavior is computable.

For functions $f \in C_{\mathrm{c}}(\mathbb{T}) \setminus \mathcal{U}_{\mathrm{c}}(\mathbb{T})$ we have

$$\limsup_{N\to\infty} \|f - S_N f\|_{C(\mathbb{T})} > 0,$$

however, (5) does not necessarily hold. Hence, we introduce the set

$$\mathcal{D}_{\mathrm{c}}(\mathbb{T}) = \Big\{ f \in C_{\mathrm{c}}(\mathbb{T})\colon \limsup_{N\to\infty} \|S_N f\|_{C(\mathbb{T})} = \infty \Big\},$$

i.e., the set of all functions $f \in C_{\mathrm{c}}(\mathbb{T})$ for which (5) holds. For our second lemma, we need to introduce several functions. Let $f_* \in C_{\mathrm{c}}(\mathbb{T})$ be the function from Lemma 1, $g \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$, and $\lambda_1, \lambda_2 \in \mathbb{R}_c \cap [0,1]$ with $\lambda_1 \neq \lambda_2$. Further, let $f_1 = (1-\lambda_1)g + \lambda_1 f_*$ and $f_2 = (1-\lambda_2)g - \lambda_2 f_*$. For $\mu \in \mathbb{R}_c \cap [0,1]$ we consider the computable function $F_\mu = (1-\mu)f_1 + \mu f_2$ and set $\psi(\mu) = \chi_{\mathcal{U}_{\mathrm{c}}(\mathbb{T})}(F_\mu)$.

Lemma 2. *For all $\lambda_1, \lambda_2 \in \mathbb{R}_c \cap [0,1]$, $\lambda_1 \neq \lambda_2$, the function ψ is not Banach–Mazur computable.*

Proof. Let $\lambda_1, \lambda_2 \in \mathbb{R}_c \cap [0,1]$, $\lambda_1 \neq \lambda_2$ be arbitrary but fixed. For $\mu \in \mathbb{R}_c \cap [0,1]$ we have

$$\begin{aligned} F_\mu(t) = {} & [(1-\mu)(1-\lambda_1) + \mu(1-\lambda_2)]g(t) \\ & + [(1-\mu)\lambda_1 - \mu\lambda_2]f_*(t). \end{aligned}$$

For $\hat{\mu} = \lambda_1/(\lambda_1+\lambda_2)$ we have

$$F_{\hat{\mu}}(t) = [(1-\hat{\mu})(1-\lambda_1) + \hat{\mu}(1-\lambda_2)]g(t).$$

Since $g \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$, it follows that that $F_{\hat{\mu}} \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$. For all other μ, i.e., $\mu \in \mathbb{R}_c \cap [0,1] \setminus \{\hat{\mu}\}$, we have $F_\mu \in \mathcal{D}_{\mathrm{c}}(\mathbb{T}) \subset C_{\mathrm{c}}(\mathbb{T}) \setminus \mathcal{U}_{\mathrm{c}}(\mathbb{T})$, because $f_* \in \mathcal{D}_{\mathrm{c}}(\mathbb{T})$. Thus, we see that

$$\psi(\mu) = \begin{cases} 1, & \mu = \hat{\mu}, \\ 0, & \mu \in \mathbb{R}_c \cap [0,1] \setminus \{\hat{\mu}\}. \end{cases}$$

It follows that for every computable sequence $\{\mu_n\}_{n\in\mathbb{Z}}$ of real numbers with $\lim_{n\to\infty} \mu_n = \hat{\mu}$ and $\mu_n \neq \hat{\mu}$, $n \in \mathbb{N}$, we have

$$0 = \lim_{n\to\infty} \psi(\mu_n) < \psi(\hat{\mu}) = 1,$$

i.e., ψ is a discontinuous function on $\mathbb{R}_c \cap [0,1]$. This implies that ψ is not Banach–Mazur computable, because every Banach–Mazur computable function is necessarily continuous [23]. □

Now we are in the position to prove Theorem 1.

Proof of Theorem 1. We prove the assertion by contradiction. Assume that there exists a Turing machine that for all $f \in C_{\mathrm{c}}(\mathbb{T})$ can decide whether $f \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$. This means we can construct a Turing machine $TM_1 \colon C_{\mathrm{c}}(\mathbb{T}) \to \{0,1\}$ with $TM_1(f) = 1$ if and only if $f \in \mathcal{U}_{\mathrm{c}}(\mathbb{T})$. For $\mu \in \mathbb{R}_c \cap [0,1]$ we have $\psi(\mu) = \chi_{\mathcal{U}_{\mathrm{c}}(\mathbb{T})}(F_\mu) = TM_1(F_\mu)$. Further, since F_μ is computable there exists a Turing machine $TM_2 \colon \mathbb{R}_c \cap [0,1] \to C_{\mathrm{c}}(\mathbb{T})$ with $TM_2(\mu) = F_\mu$. It follows that the concatenation of both Turing machines gives a Turing machine $TM_3 \colon \mathbb{R}_c \cap [0,1] \to \{0,1\}$ with $TM_3(\mu) = TM_1(TM_2(\mu)) = TM_1(F_\mu) = \psi(\mu)$. Let $\{\lambda_n\}_{n\in\mathbb{N}}$ be a computable sequence of real numbers. Then $\{q_n\}_{n\in\mathbb{N}}$ with $q_n = TM_3(\lambda_n) = \psi(\lambda_n)$, $n \in \mathbb{N}$, is a computable sequence. Hence, ψ maps the computable sequence $\{\lambda_n\}_{n\in\mathbb{N}}$ into the computable sequence $\{\psi(\lambda_n)\}_{n\in\mathbb{N}}$, or, in other words, ψ is Banach–Mazur computable. This is a contradiction, because in Lemma 1 we have already shown that ψ is not Banach–Mazur computable. □

Next we give a sketch of the remaining proof of Lemma 1.

Sketch of the proof of Lemma 1. For $N \in \mathbb{N}$, let

$$p_N(t) = \sum_{k=1}^{N} \frac{1}{k} \sin(kt), \quad t \in [-\pi, \pi).$$

We have $p_N \in C_{\mathrm{c}}(\mathbb{T})$, and it can be shown that

$$\|p_N\|_{C(\mathbb{T})} < \pi \tag{6}$$

for all $N \in \mathbb{N}$. For $M, N \in \mathbb{N}$ let

$$\begin{aligned} q_{M,N}(t) &= \mathrm{e}^{iMt}\, p_N(t) = \mathrm{e}^{iMt} \sum_{k=1}^{N} \frac{1}{k} \sin(kt) \\ &= -\frac{\mathrm{e}^{iMt}}{2i} \left(\sum_{k=1}^{N} \frac{1}{k}\, \mathrm{e}^{-ikt} - \sum_{k=1}^{N} \frac{1}{k}\, \mathrm{e}^{ikt} \right). \end{aligned}$$

Then, a short calculation shows that, for $M \geq N$,

$$(S_M q_{M,N})(t) = -\frac{1}{2i} \sum_{k=1}^{N} \frac{1}{k}\, \mathrm{e}^{-ikt}\, \mathrm{e}^{iMt}.$$

Hence, it follows that

$$\begin{aligned} |(S_M q_{M,N})(0)| &= \frac{1}{2} \sum_{k=1}^{N} \frac{1}{k} > \frac{1}{2} \sum_{k=1}^{N} \int_k^{k+1} \frac{1}{\tau}\, \mathrm{d}\tau \\ &= \frac{1}{2} \int_1^{N+1} \frac{1}{\tau}\, \mathrm{d}\tau = \frac{1}{2} \log(N+1). \end{aligned} \tag{7}$$

For $l \in \mathbb{N}$, let $N_l = 2^{(l^3)}$. We set $M_1 = 2^{(1^3)} = 2$, $M_2 = 2M_1 + N_2$, and, for general $k \in \mathbb{N}$,

$$M_{k+1} = 2M_k + N_{k+1} = 2M_k + 2^{((k+1)^3)}.$$

Having this specific construction rule, the sequences $\{N_l\}_{l\in\mathbb{N}}$ and $\{M_k\}_{k\in\mathbb{N}}$ are Turing computable. Let

$$f_*(t) = \sum_{l=1}^{\infty} \frac{1}{l^2} q_{M_l,N_l}(t), \quad t \in [-\pi, \pi). \tag{8}$$

We have

$$\|f_*\|_{C(\mathbb{T})} \leq \sum_{l=1}^{\infty} \frac{1}{l^2} \|p_{N_l}\|_{C(\mathbb{T})} < \pi \sum_{l=1}^{\infty} \frac{1}{l^2} = \frac{\pi^3}{6},$$

where the last inequality follows from (6). This shows that $f_* \in C(\mathbb{T})$. Further, using a similar calculation, we obtain, for $K \in \mathbb{N}$,

$$\begin{aligned} &\left\| f_* - \sum_{l=1}^{K} \frac{1}{l^2} q_{M_l,N_l} \right\|_{C(\mathbb{T})} = \left\| \sum_{l=K+1}^{\infty} \frac{1}{l^2} q_{M_l,N_l} \right\|_{C(\mathbb{T})} \\ &< \pi \sum_{l=K+1}^{\infty} \frac{1}{l^2} < \pi \sum_{l=K+1}^{\infty} \int_{l-1}^{l} \frac{1}{\tau^2}\, \mathrm{d}\tau = \pi \int_K^{\infty} \frac{1}{\tau^2}\, \mathrm{d}\tau = \frac{\pi}{K}. \end{aligned}$$

Thus, we see that the sequence of trigonometric polynomials $\{\sum_{l=1}^{K} \frac{1}{l^2} q_{M_l,N_l}\}_{K=1}^{\infty}$ is a computable sequence that convergences to f_*, effectively in K. Hence, we have $f_* \in C_{\mathrm{c}}(\mathbb{T})$. For $k \in \mathbb{N}$, we have

$$(S_{M_k} f_*)(t) = \sum_{l=1}^{k-1} \frac{1}{l^2} q_{M_l,N_l}(t) + \frac{1}{k^2} (S_{M_k} q_{M_k,N_k})(t).$$

It follows that

$$\begin{aligned} \left| (S_{M_k} f_*)(t) - \frac{1}{k^2} (S_{M_k} q_{M_k,N_k})(t) \right| &\leq \left\| \sum_{l=1}^{k-1} \frac{1}{l^2} q_{M_l,N_l} \right\|_{C(\mathbb{T})} \\ &< \pi \sum_{l=1}^{\infty} \frac{1}{l^2} = \frac{\pi^3}{6}, \end{aligned}$$

and consequently that

$$\begin{aligned} |(S_{M_k} f_*)(0)| &\geq \frac{1}{k^2} |(S_{M_k} q_{M_k,N_k})(0)| - \frac{\pi^3}{6} \\ &> \frac{1}{2k^2} \log(N_k + 1) - \frac{\pi^3}{6} > \frac{1}{2k^2} \log(2^{(k^3)}) - \frac{\pi^3}{6} \\ &= \frac{k}{2} \log(2) - \frac{\pi^3}{6} \end{aligned}$$

for all $k \in \mathbb{N}$, where we used (7) in the second line. This inequality shows that $\limsup_{N\to\infty} \|S_N f_*\|_{C(\mathbb{T})} = \infty$. At this point we already have proved the assertion of Lemma 1.

Next we prove the additional statement made in Remark 2. According to the construction of f_* we have

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} f_*(t)\, \mathrm{e}^{-int}\, \mathrm{d}t = 0$$

for all $n \leq -1$. Hence, for $l \geq 0$ the functions $g_l(t) = \mathrm{e}^{ilt} f_*(t)$, $t \in [-\pi, \pi)$, are linearly independent. Since f_* is computable, it follows that $g_l \in C_{\mathrm{c}}(\mathbb{T})$, $l \geq 0$. Further,

$$N_l(k) = l + M_k, \quad k \in \mathbb{N},$$

gives the Turing computable functions $N_l\colon \mathbb{N} \to \mathbb{N}$, and

$$U_l(k) = \frac{k}{2}\log(2) - \frac{\pi^3}{6}, \quad k \in \mathbb{N},$$

the Turing computable functions $U_l\colon \mathbb{N} \to \mathbb{R}_c$, the existence of which we claimed in Remark 2. □

4. DISCUSSION

In Section 3 we have seen that there exists no Turing machine that for all $f \in C_\mathrm{c}(\mathbb{T})$ can decide whether $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$. Next, we want to ask if maybe a weaker Turing machine exists: Does there exist a Turing machine $TM_{s\mathcal{U}}$ that stops exactly when $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$? Having such a Turing machine would not solve our original problem, because if $TM_{s\mathcal{U}}$ has not stopped after a certain number of steps, it could be that $TM_{s\mathcal{U}}$ simply did not yet "detect" that $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$ or that $f \in C_\mathrm{c}(\mathbb{T}) \setminus \mathcal{U}_\mathrm{c}(\mathbb{T})$. We could also ask the analogous question for the set $C_\mathrm{c}(\mathbb{T}) \setminus \mathcal{U}_\mathrm{c}(\mathbb{T})$: Does there exist a Turing machine $T_{s\mathcal{V}}$ that stops exactly when $C_\mathrm{c}(\mathbb{T}) \setminus \mathcal{U}_\mathrm{c}(\mathbb{T})$?

For convenience we introduce the set $\mathcal{V}_\mathrm{c}(\mathbb{T}) = C_\mathrm{c}(\mathbb{T}) \setminus \mathcal{U}_\mathrm{c}(\mathbb{T})$, consisting of all computable continuous functions for which the Fourier series is not uniformly convergent. Since

$$\big|\|S_N f\|_{C(\mathbb{T})} - \|f\|_{C(\mathbb{T})}\big| \leq \|f - S_N f\|_{C(\mathbb{T})},$$

we have

$$\lim_{N\to\infty} \big|\|S_N f\|_{C(\mathbb{T})} - \|f\|_{C(\mathbb{T})}\big| = 0$$

for all $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$. Thus, we see that

$$\mathcal{V}_\mathrm{c}(\mathbb{T}) = \left\{ f \in C(\mathbb{T})\colon \limsup_{N\to\infty} \big|\|S_N f\|_{C(\mathbb{T})} - \|f\|_{C(\mathbb{T})}\big| > 0 \right\}.$$

It is immediately clear that the Turing machines $TM_{s\mathcal{U}}$ and $TM_{s\mathcal{V}}$ cannot exist simultaneously.

Corollary 1. *There exist no two Turing machines $TM_{s\mathcal{U}}$ and $TM_{s\mathcal{V}}$ such that, for all $f \in C_c(\mathbb{T})$, $TM_{s\mathcal{U}}$ stops exactly when $f \in \mathcal{U}_c(\mathbb{T})$ and $TM_{s\mathcal{V}}$ stops exactly when $f \in \mathcal{V}_c(\mathbb{T})$.*

Proof. Assume that two such Turing machines exist. Then we can construct a new Turing machine

$$TM_{d\mathcal{U}}(f) = \begin{cases} 1, & TM_{s\mathcal{U}}(f) \text{ stops}, \\ 0, & TM_{s\mathcal{V}}(f) \text{ stops}. \end{cases}$$

That is, we would let run both Turing machines in parallel and stop if one of the Turing machines stops. Note that for $f \in C_\mathrm{c}(\mathbb{T})$ it is guaranteed that exactly one of the Turing machines stops. Hence, the Turing machine $TM_{d\mathcal{U}}$ would solve our original problem of deciding whether $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$, which is not possible according to Theorem 1. □

Moreover, for the Turing machine $TM_{s\mathcal{U}}$ we can answer the question of existence in the negative.

Theorem 2. *There exists no Turing machine $TM_{s\mathcal{U}}$ such that, for all $f \in C_c(\mathbb{T})$, $TM_{s\mathcal{U}}$ stops exactly when $f \in \mathcal{U}_c(\mathbb{T})$.*

Theorem 2 implies that the set of functions in $C_\mathrm{c}(\mathbb{T})$ with uniform convergence of the Fourier series cannot have a computable characterization. The question whether the Turing machine $TM_{s\mathcal{V}}$ exists is open.

Proof. We prove the assertion by contradiction. Assume that there exists a Turing machine $TM_{s\mathcal{U}}$ that stops exactly when $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$. Clearly, we have $\mathbf{0} \in \mathcal{U}_\mathrm{c}(\mathbb{T})$. Further, a result from [27] shows that there exists an infinite dimensional closed subspace $\mathcal{S} \subset C(\mathbb{T})$ such that

$$\limsup_{N\to\infty} \|S_N f\|_{C(\mathbb{T})} = \infty \tag{9}$$

for all $f \in \mathcal{S}$, $f \neq \mathbf{0}$. According to our construction of $\mathcal{S}$ in [27], the set $\mathcal{S} \cap C_\mathrm{c}(\mathbb{T})$ is not empty. We have $\mathbf{0} \in \mathcal{S}$, and for each $f \in \mathcal{S}$ with $\|f\|_{C(\mathbb{T})} > 0$ we have (9) and consequently $f \in \mathcal{V}_\mathrm{c}(\mathbb{T})$. Hence, for $f \in \mathcal{S} \cap C_\mathrm{c}(\mathbb{T})$, we have $f \in \mathcal{U}_\mathrm{c}(\mathbb{T})$ if and only if $\|f\|_{C(\mathbb{T})} = 0$. Thus, for $f \in \mathcal{S} \cap C_\mathrm{c}(\mathbb{T})$, $TM_{s\mathcal{U}}$ stops if and only if $\|f\|_{C(\mathbb{T})} = 0$.

Let $h \in \mathcal{S} \cap C_\mathrm{c}(\mathbb{T})$, $h \neq \mathbf{0}$. We have $\|h\|_{C(\mathbb{T})} \in \mathbb{R}_c$ and $\|h\|_{C(\mathbb{T})} > 0$. For $\lambda \in \mathbb{R}_c$, $\lambda \geq 0$, we set $g = \lambda h/\|h\|_{C(\mathbb{T})}$. Since $\lambda/\|h\|_{C(\mathbb{T})}$ is computable, it follows that $g \in \mathcal{S} \cap C_\mathrm{c}(\mathbb{T})$. Note that $\|g\|_{C(\mathbb{T})} = \lambda$.

We further use the fact that there exists a Turing machine $TM_>$ that stops if and only if $\|g\|_{C(\mathbb{T})} > 0$ [24, p. 14]. Using the Turing machines $TM_{s\mathcal{U}}$ and $TM_>$ we can construct a Turing machine $TM_z\colon \mathbb{R}_c \to \{0,1\}$ as follows: Given $\lambda \in \mathbb{R}_c$, we compute g as well as $\|g\|_{C(T)}$. Then we start the Turing machine $TM_{s\mathcal{U}}$ with g as input and the Turing machine $TM_>$ with $\|g\|_{C(T)}$ as input. We know that $TM_{s\mathcal{U}}$ stops if and only if $\lambda = \|g\|_{C(T)} = 0$, and that TM_g stops if and only if $\lambda = \|g\|_{C(T)} > 0$. This gives us a Turing machine TM_z with

$$TM_z(\lambda) = \begin{cases} 1, & \lambda > 0, \\ 0, & \lambda = 0. \end{cases}$$

However, such a Turing machine does not exist [24, p. 14]. □

5. RELATION TO PRIOR WORK

Although the convergence behavior of the Fourier series is important and a well studied topic in classical analysis [28], questions of computability have not caught much attention. The convergence of Fourier series for computable Lebesgue integrable functions was studied in [29], and it has been shown that the set of L^1-computable functions, whose Fourier series diverges a.e. is big in a certain sense. Further, a computable function $f(t)$, the Fourier series of which converges uniformly, but the convergence is not effective for $t = 0$, was given in [30]. In the present paper we constructed a computable continuous function with divergence. It would be practically interesting to have a Turing machine that always can decide whether the Fourier series of a continuous function converges uniformly. We have shown that such a Turing machine does not exist. Even the simpler question if there exists a Turing machine which accepts only those continuous functions for which the Fourier series converges uniformly, has to be answered in the negative. To the best of our knowledge there exist no further publications that treat this topic.

In general, it seems that the signal approximation, and in particular the possible divergence of approximation processes, is often not given the proper attention. One source of this insouciance are calculations that are based on distributions, which give a false sense of security. A closer analysis of those calculations sometimes reveals that the made claims are not justified [31, 32]. Recently, even for benign signal spaces, divergence phenomena could be shown for the Shannon sampling series [26,33–35]. This once more highlights that it would be desirable to have an algorithm that can decide whether a general approximation process converges or diverges for a given function. The results in this paper show that for the Fourier transform such an algorithm cannot exist.

6. REFERENCES

[1] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[2] M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[3] M. Unser, "Sampling–50 years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.

[4] P. L. Butzer and J. Lei, "Approximation of signals using measured sampled values and error analysis," *Communications in Applied Analysis. An International Journal for Theory and Applications*, vol. 4, no. 2, pp. 245–255, 2000.

[5] P. S. Huggins and S. W. Zucker, "Greedy basis pursuit," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3760–3772, Jul. 2007.

[6] G. Schmeisser and F. Stenger, "Sinc approximation with a Gaussian multiplier," *Sampling Theory in Signal and Image Processing*, vol. 6, no. 2, pp. 199–221, May 2007.

[7] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[8] P. Steffen, P. N. Heller, R. A. Gopinath, and C. . S. Burrus, "Theory of regular M-band wavelet bases," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3497–3511, Dec. 1993.

[9] G. Wang, J. Zhang, and G.-W. Pan, "Solution of inverse problems in image processing by wavelet expansion," *IEEE Transactions on Image Processing*, vol. 4, no. 5, pp. 579–593, May 1995.

[10] I. Daubechies, *Ten Lectures on Wavelets*. Siam, 1992, vol. 61.

[11] S.-C. Pei, M.-H. Yeh, and T.-L. Luo, "Fractional Fourier series expansion for finite signals and dual extension to discrete-time fractional Fourier transform," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2883–2888, Oct. 1999.

[12] D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty — I," *Bell System Technical Journal*, vol. 40, pp. 43–63, Jan. 1961.

[13] A. H. Tewfik, D. Sinha, and P. Jorgensen, "On the optimal choice of a wavelet for signal representation," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 747–765, Mar. 1992.

[14] J.-C. Pesquet, H. Krim, and H. Carfantan, "Time-invariant orthonormal wavelet representations," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1964–1970, Aug. 1996.

[15] L. R. Rabiner, "Techniques for designing finite-duration impulse-response digital filters," *IEEE Transactions on Communication Technology*, vol. 19, no. 2, pp. 188–195, Apr. 1971.

[16] Y. Katznelson, *An Introduction to Harmonic Analysis*. Cambridge University Press, 2004.

[17] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, Nov. 1936.

[18] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem. A correction," *Proceedings of the London Mathematical Society*, vol. s2-43, no. 1, pp. 544–546, Jan. 1937.

[19] K. Weihrauch, *Computable Analysis: An Introduction*. Springer-Verlag, 2000.

[20] G. S. Boolos, J. P. Burgess, and R. C. Jeffrey, *Computability and Logic*. Cambridge University Press, 2002.

[21] R. I. Soare, *Recursively Enumerable Sets and Degrees*, ser. Perspectives in Mathematical Logic. Springer-Verlag Berlin Heidelberg, 1987.

[22] E. Specker, "Nicht konstruktiv beweisbare Sätze der Analysis," *The Journal of Symbolic Logic*, vol. 14, no. 3, pp. 145–158, Sep. 1949.

[23] J. Avigad and V. Brattka, "Computability and analysis: the legacy of Alan Turing," in *Turing's Legacy: Developments from Turing's Ideas in Logic*, R. Downey, Ed. Cambridge University Press, 2014.

[24] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics*. Springer-Verlag, 1989.

[25] T. Rado, "On non-computable functions," *Bell System Technical Journal*, vol. 41, no. 3, pp. 877–884, May 1962.

[26] H. Boche, U. J. Mönich, and E. Tampubolon, "Spaceability and strong divergence of the Shannon sampling series and applications," *Journal of Approximation Theory*, vol. 222, pp. 157–174, Oct. 2017.

[27] H. Boche and U. J. Mönich, "Divergence behavior of sequences of linear operators with applications," *Journal of Fourier Analysis and Applications*, 2018.

[28] A. Zygmund, *Trigonometric Series*, 2nd ed. Cambridge University Press, 1993, vol. I and II.

[29] P. Moser, "On the convergence of Fourier series of computable Lebesgue integrable functions," *Electronic Notes in Theoretical Computer Science*, vol. 202, pp. 13–18, Mar. 2008.

[30] M. B. Pour-El and I. Richards, "Computability and noncomputability in classical analysis," *Transactions of the American Mathematical Society*, vol. 275, no. 2, pp. 539–560, Feb. 1983.

[31] H. Boche and U. J. Mönich, "Distributional behavior of convolution sum system representations," *IEEE Transactions on Signal Processing*, vol. 66, no. 19, pp. 5056–5065, Oct. 2018.

[32] H. Boche, U. Mönich, and B. Meinerzhagen, "Non-existence of convolution sum system representations," 2019, in preparation.

[33] H. Boche and U. J. Mönich, "There exists no globally uniformly convergent reconstruction for the Paley-Wiener space $\mathcal{PW}_\pi^1$ of bandlimited functions sampled at Nyquist rate," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3170–3179, Jul. 2008.

[34] H. Boche and U. J. Mönich, "Sampling of deterministic signals and systems," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2101–2111, May 2011.

[35] H. Boche and B. Farrell, "Strong divergence of reconstruction procedures for the Paley-Wiener space PW_π^1 and the Hardy space H^1," *Journal of Approximation Theory*, vol. 183, pp. 98–117, Jul. 2014.