ON THE FOURIER REPRESENTATION OF COMPUTABLE CONTINUOUS SIGNALS

Holger Boche and Ullrich J. Mönich
Technische Universität München
Lehrstuhl für Theoretische Informationstechnik

Introduction

- The approximation of functions is an important topic in signal processing.
- The approximation of continuous periodic functions by Fourier series has a long history.
- We study whether it is possible to decide algorithmically if the Fourier series of a continuous function converges uniformly.

Motivation

The Fourier series of a 2π-periodic function f is given by:

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

where

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

are the usual Fourier coefficients.

Convergence / divergence:

- For continuously differentiable functions, the Fourier series converges pointwise for all $t \in [-\pi, \pi)$.
- For absolutely continuous functions the Fourier series converges uniformly on all of $\mathbb{R}$.
- It is well-known that there exist continuous functions such that the Fourier series diverges at some point $t \in [-\pi, \pi)$.

Question: Given a continuous function, does the Fourier series converge uniformly or not?

Can this question be answered algorithmically?

- More precisely: Can we 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.

Turing Machines

- A Turing machine is an abstract device that manipulates symbols on a strip of tape according to certain rules.
- 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.

Computability Basics

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. 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 $n \in \mathbb{N}$.
$\mathbb{R}_c$: set of computable real numbers.

Computable Functions

Banach–Mazur computability

- There are several ways to define computability of functions: e.g. Turing/Borel, Markov, and Banach–Mazur computability.
- Banach–Mazur computability is the weakest form of computability. A function that is computable with respect to one of the two other definitions is Banach–Mazur 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.

Computability in $C(\mathbb{T})$

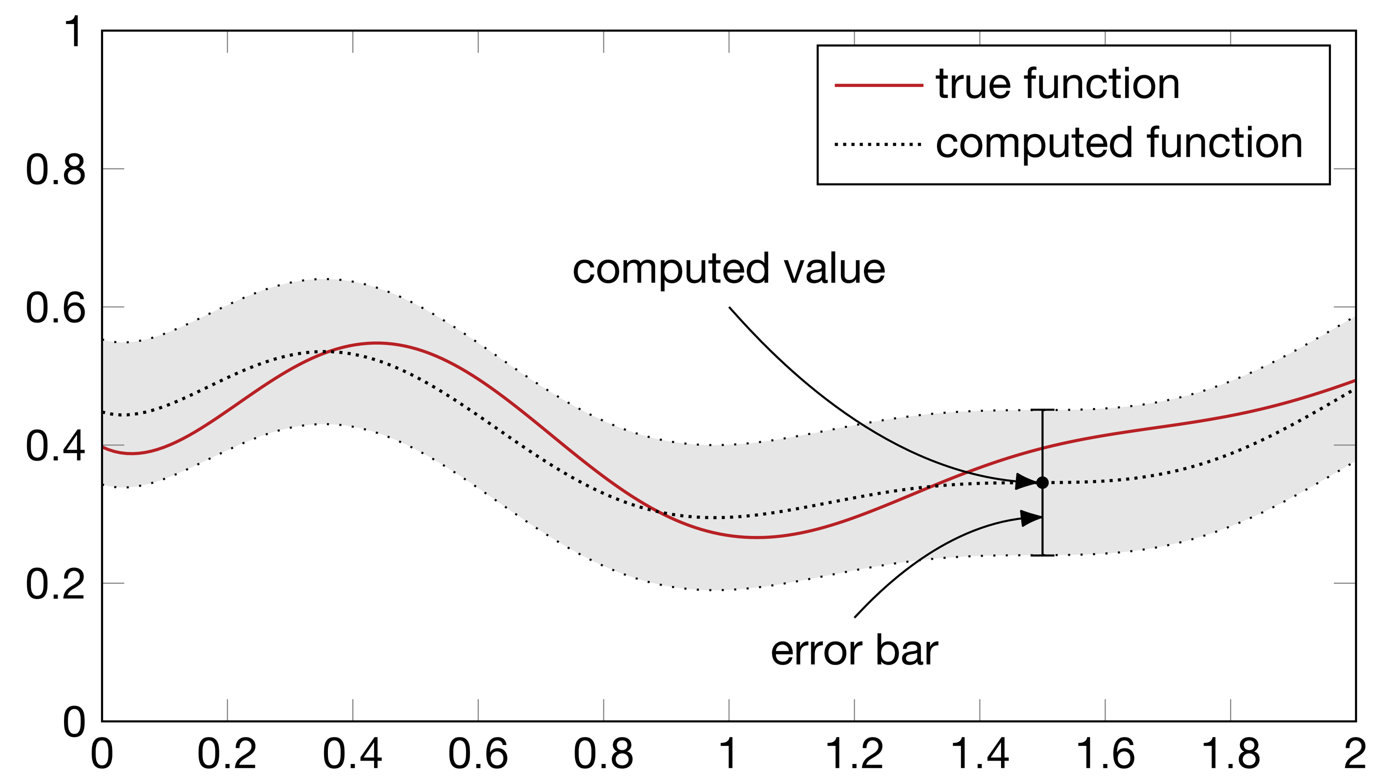
$C(\mathbb{T})$: space of all continuous 2π-periodic functions.
Norm: $\|f\|_{C(\mathbb{T})} = \max_{t\in[-\pi,\pi)} |f(t)|$.

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

$$\|f - g_n\|_{C(\mathbb{T})} < 2^{-n} \quad \text{for all } n \in \mathbb{N}.$$

$C_c(\mathbb{T})$: set of all computable functions in $C(\mathbb{T})$.

Control of the approximation error



Main Result

N-th partial sum of the Fourier series:

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

Set of all $f \in C_c(\mathbb{T})$ for which the Fourier series converges uniformly:

$$\mathfrak{U}_c(\mathbb{T}) = \left\{ f \in C_c(\mathbb{T}) \colon \lim_{N\to\infty} \|f - S_N f\|_{C(\mathbb{T})} = 0 \right\}.$$

- There exist infinitely many functions in $\mathfrak{U}_c(\mathbb{T})$. For example, all trigonometric polynomials with rational coefficients are in $\mathfrak{U}_c(\mathbb{T})$.

Is it possible to characterize the set $\mathfrak{U}_c(\mathbb{T})$ algorithmically?

Answer: There exists no algorithm that always can decide whether the Fourier series of a continuous computable function converges uniformly.

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

- Any algorithm that is forced to give a decision after a finite amount of time, needs to give wrong answers for some functions.

Semi-Decidability

Weaker question 1: Does there exist a Turing machine $TM_{\mathfrak{U}}$ that stops exactly when $f \in \mathfrak{U}_c(\mathbb{T})$?

- Such a Turing machine would not solve our original problem.
- If $TM_{\mathfrak{U}}$ has not stopped after a certain number of steps, it could be that $TM_{\mathfrak{U}}$ simply has not yet "detected" that $f \in \mathfrak{U}_c(\mathbb{T})$.

$\mathfrak{V}_c(\mathbb{T}) = C_c(\mathbb{T}) \setminus \mathfrak{U}_c(\mathbb{T})$: set of all computable continuous functions for which the Fourier series is not uniformly convergent.

Weaker question 2: Does there exist a Turing machine $TM_{\mathfrak{V}}$ that stops exactly when $f \in \mathfrak{V}_c(\mathbb{T})$?

We can answer both questions in the negative.

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

$\Rightarrow$ The set of functions in $C_c(\mathbb{T})$ with uniform convergence of the Fourier series cannot have a computable characterization.

Further applications: Computability of Fourier transform and spectral factorization.

H. Boche and U. J. Mönich, "Turing computability of the Fourier transform of bandlimited functions," in *Proceedings of the 2019 IEEE International Symposium on Information Theory*, 2019, accepted

H. Boche and V. Pohl, "On the algorithmic solvability of the spectral factorization and the calculation of the Wiener filter on Turing machines," in *Proceedings of the 2019 IEEE International Symposium on Information Theory*, 2019, accepted