Armenian Journal of Mathematics
Volume 14, Number 14, 2022, 1–31
https://doi.org/10.52737/18291163-2022.14.14-1-31

Acceleration of Convergence of Fourier Series Using the Phenomenon of Over-Convergence

A. Nersessian

Dedicated to the memory
of my colleague, a remarkable mathematician
and unusual personality Viktor Arzumanyan

Abstract. In recent publications of the author, the phenomenon of over-convergence was discovered, and a spectral method has been presented for accelerating the convergence of truncated Fourier series for smooth functions. On this basis, a certain parametric system that is biorthogonal to the corresponding segment of the Fourier system turned out to be unusually effective. This article reconsiders some approaches and makes some adjustments to previous publications. As a result, two improved schemes for the recovery of a function based on a finite set of its Fourier coefficients are proposed. Numerical experiments confirm a significant increase in the efficiency of corresponding algorithms in typical classes of smooth functions. In conclusion, some prospects for the development and generalization of the above approaches are discussed.

Key Words: Fourier Series, Acceleration of Convergence, Parametric Biorthogonalization, Spectral Methods, Over-Convergence Phenomenon
Mathematics Subject Classification 2010: 42A16, 42A20, 42A24, 34B05

1 Introduction

1.1 Acceleration of convergence of Fourier series

One of the classical tools of mathematics is the apparatus of Fourier series based on the orthogonal system $\{e^{i\pi kx}\}$, $k = 0, \pm 1, \pm 2, \ldots$, complete in $L_2[-1, 1]$. However, in practice, it is extremely limited because of the poor approximation of piece-wise smooth functions. Thus, in the case where a function has discontinuity points (taking also into account discontinuities at the ends of the interval $[-1,1]$), an intense oscillation arises in their neighborhood, and the uniform convergence is absent (the Gibbs phenomenon). Even at the dawn of twentieth century, A.N. Krylov ([1], 1907) proposed the following approach.

Let a piece-wise smooth function f be given on the segment $[-1,1]$ with the Fourier coefficients $\{f_s\}, s = 0, \pm 1, \ldots, \pm n,\ n \geq 1$, and with the following jumps of the function f or its derivatives up to the order $q \geq 1$ at the points $\{a_k\}$, $-1 < a_1 < \cdots < a_m = 1,\ 1 \leq m < \infty$:

$$A_{p,k} = f^{(k)}(a_p - 0) - f^{(k)}(a_p + 0), \quad A_{m,k} = f^{(k)}(-1) - f^{(k)}(1), \\ k = 0, 1, \ldots, q \geq 0,\ p = 1, \ldots, m. \tag{1}$$

In the neighborhoods of other points, we assume that $f \in C^{q+1}$. Let us construct a function $g = g(x)$, $x \in [-1,1]$ with Fourier coefficients $\{g_s\}$, $0 \leq |s| \leq n$, which has the same jumps at the same points, and $g \in C^{q+1}$ at the neighborhoods of other points. With jumps (1) in hand, one can construct, e.g., piecewise-polynomial g. As a result, 2-periodic extension of the function $F = (f - g)$ is q times continuously differentiable on the whole axis, and, therefore, it is possible to approximate f in the form

$$f(x) \simeq f_n(x) = g(x) + \sum_{s=-n}^{n} (f_s - g_s) e^{i\pi s x} \tag{2}$$

with the asymptotic error $r_n(x) = o(n^{-q})$ as $n \to \infty$ for $x \in [-1,1]$. However, computing the jumps $\{A_{sk}(f)\}$ directly by the function f seriously limits the scope of practical application of Krylov's method.

It was only at the end of 19th century that Knut Eckhoff ([2], 1993) managed to find approximate values for this jumps solving the following system of linear equations with the Vandermonde matrix by choosing the indexes $s = s_k$, $k = 1, 2, ..., m(q+1)$, $\theta n \leq |s_k| \leq n$, $0 < \theta = const < 1$:

$$f_s = -\frac{1}{2} \sum_{p=1}^{m} e^{-i\pi s\, a_p} \sum_{k=0}^{q} \frac{\tilde{A}_{p,k}}{(i\pi s)^{k+1}}, \quad s = s_1, s_2, ..., s_{m(q+1)}. \tag{3}$$

The use of only the Fourier coefficients allowed this "spectral" approach to effectively restore the function f using Bernoulli polynomials as a g. This polynomials can be calculated recursively

$$B_0(x) = 1, \quad B_k(x) = \int B_{k-1}(x)\, dx, \quad x \in [-1,1], \tag{4}$$

where the constant of integration is defined by the relation $\int_{-1}^{1} B_k(x)\, dx = 0$. The Fourier coefficients $\{b_{k,s}\}$, $s \in \mathbb{Z}$, of polynomial $B_k(x)$ have the following simple form

$$b_{k,s} = \begin{cases} 1, & s=0,\ k=0; \\ 0, & s=0,\ k=1,2,\dots; \\ \dfrac{(-1)^{s+1}}{\left(2(i\pi s)^k\right)}, & s\neq 0,\ k=1,2,\dots; \\ 0, & s\neq 0,\ k=0. \end{cases} \tag{5}$$

It is natural to call this approach *KE-method.* Unfortunately, it also has some disadvantages. First, for $n >> 1$, the approximate definition of the jumps $\{\tilde{A}_{p,k}\}$ (see (3)) predetermines a perceptible error, and second, the representation of the function by polynomials leads to an accumulation of errors and, ultimately, to a loss of stability.

Over the past three decades, in numerous works devoted to the scheme of the KE-method, not only polynomials were used as a function g when $g_s \neq 0$ for all s. Here we will not dwell even on a brief review since, as far as we know, the approach proposed below has not been used before. Those who wish can fill this gap, for example, in [2–11] and references therein.

1.2 Over-convergence phenomenon

In [3] (2004), for a smooth function f, a combination of the KE-method and a para-diagonal Padé approximant was realized. As a result of a number numerical experiments, it turned out that, as a rule, such an algorithm is “almost exact” on a certain *infinite-dimensional* space, although *only a finite number* of Fourier coefficients and almost exact jumps of f were used.

In [4], a similar situation was numerically confirmed for Fourier-Bessel series. Finally, a corresponding scheme was proposed in [5,8] for using similar phenomena for the eigenfunctions expansion of boundary value problems for ordinary differential equations on a finite segment.

In the case of a Fourier series, in papers [6] and [7], asymptotic errors were studied in terms of jumps (1) of a smooth function f. In [6], asymptotic estimates of approximation error is represented. In [7], the exact constant of the asymptotic error was estimated depending on the choice of integers $s = s_1, s_2, ..., s_{m(q+1)}$ in equation (3).

However, the results of numerical experiments can only serve as a basis for a hypothesis about the super-accuracy of summation truncated Fourier series in some infinite-dimensional spaces. The theoretical substantiation of this hypothesis (about the existence of “phenomenon of over-convergence”) was obtained and applied in the case of Fourier series quite recently in [9–12]. In [13], a similar phenomenon was established in a discrete case when the convergence of a finite Fourier series was accelerated.

Based on these results, two more efficient algorithms for the smooth function f are proposed, analyzed and numerically implemented below.

2 Some fast summation algorithms

We begin by presenting the basic constructions and formulas of the papers [9–12] that will be needed later.

2.1 Definitions

The classical definition of the partial sums of a Fourier series is focused on the traditional forms of summation. It is convenient for us to use a more general notation (see [9]).

Definition 1 *We call any sum of the form*

$$S_n(x) \overset{def}{=} \sum_{k\in D_n} f_k \exp(i\,\pi\,k\,x), \quad x\in[-1,1], \tag{6}$$

the truncated Fourier series, where

$$f_s = \frac{1}{2}\int_{-1}^{1} f(x)\, e^{-i\,\pi\,s\,x}\,dx, \quad s\in\mathbb{Z} \tag{7}$$

are Fourier coefficients of f, and $D_n = \{d_k\}$, $k = 1,\ldots,n$, is a set of n different integers ($n \geq 1$).

Definition 2 *Let $n \geq 1$ be a fixed integer. Consider a system of functions $U_n = \{\exp(i\,\pi\,\lambda_k\,x)\}$, $\lambda_k \in \mathbb{C}$, $x\in[-1,1]$, $k = 1,2,\ldots,n$, where $\{\lambda_k\}$ are arbitrary parameters. Consider the linear span $Q_n = span\{U_n\}$. We call a function $q \in Q_n$ as a* quasi-polynomial *of degree at most n.*

Note that $h^{-1}\big(\exp(i\,\pi\,(\lambda_k + h)\,x) - \exp(i\,\pi\,\lambda_k\,x)\big) \to i\,\pi\,x\exp(i\,\pi\,\lambda_k\,x)$ as $h \to 0$. Therefore, it is clear that $q \in Q_n$ if and only if either $q(x) \equiv 0$ or $q(x) = \sum_k P_{\beta_k}(x)\,\exp(i\,\pi\,\lambda_k\,x)$, where the polynomials $P_{\beta_k}(x) \not\equiv 0$ are of degree exactly β_k, and $m = \sum_k (1+\beta_k) \leq n$. The number m will be understood below as the degree of the quasi-polynomial q.

Remark 1 *Quasi-polynomials and their degrees are invariant under shifts of the argument. The same is preserved when multiplying a quasi-polynomial by a function $\exp(a\,x)$, $a = const \in \mathbb{C}$. When multiplying two quasi-polynomials of a general form, a quasi-polynomial of a higher degree than each of the factors is obtained, as a rule.*

2.2 Basic parametric biorthogonal system

Let $m \geq 1$ be an integer and D_m be a set of m integers. For $r \in D_m$, consider a set of m *non-integer parameters* $\{\lambda_k\} \subset \mathbb{C}$, $k \in D_m$, and the infinite sequence (for details, see [9] and [11])

$$t_{r,s} \overset{def}{=} (-1)^{s-r} \prod_{\substack{p \in D_m \\ p \neq r}} \frac{s-p}{r-p} \prod_{j \in D_m} \frac{r-\lambda_j}{s-\lambda_j}, \quad r \in D_m,\ s \in \mathbb{Z}. \tag{8}$$

Here the products $L_r(s) = \prod_{\substack{p \in D_m \\ p \neq r}} (s-p)/(r-p)$ are analogues of the well-known Lagrange basis for polynomial interpolation because $L_r(s) = \delta_{r,s}$, $r, s \in D_m$. Note also that $t_{r,s} = O(1/s)$, $s \to \infty$.

Remark 2 *We emphasize that formula (8) does not depend on the numbering of parameters $\{\lambda_k\}$. This numbering is provided for convenience.*

Considering that some parameters $\{\lambda_k\}$ can be repeated, (8) can be written as

$$t_{r,s} = (-1)^{s-r} \prod_{\substack{p \in D_m \\ p \neq r}} \frac{s-p}{r-p} \prod_{j \in D_{m_1}} \left(\frac{r-\lambda_j}{s-\lambda_j} \right)^{n_j}, \quad r \in D_m,\ s \in \mathbb{Z}, \tag{9}$$

where $D_{m_1} \subset D_m$, $\{n_q\}$ are corresponding positive integers, and $\sum_{j \in D_{m_1}} n_j = m$, $\lambda_p \neq \lambda_q$ if $p \neq q$. Let us call the number n_j the multiplicity of the parameter *λ_j.*

Further, we denote ($x \in [-1,1]$)

$$T_r(x) \overset{def}{=} \exp(i\,\pi\, r\, x) + \sum_{s \notin D_m} t_{r,s} \exp(i\,\pi\, s\, x), \quad r \in D_m,$$

$$f(x) \simeq F_m(x) \overset{def}{=} \sum_{r \in D_m} f_r T_r(x), \quad R_m(x) \overset{def}{=} f(x) - F_m(x). \tag{10}$$

It is obvious that system $\{T_r(x), 1/2 \exp(i\,\pi\, r\, x)\}$, $r \in D_m$, is biorthogonal on the segment $x \in [-1,1]$ and L_2-error of approximation $f(x) \simeq F_m(x)$ can be derived from the formula

$$\|R_m\|^2 = \sum_{s \notin D_m} \|f_s - \sum_{r \in D_m} f_r t_{r,s}\|^2. \tag{11}$$

2.3 Explicit form of the system $\{T_r\}$

2.3.1 Basic formulas

The following obvious formula ($\lambda \in \mathbb{C}$, $x \in [-1,1]$)

$$\exp(i\,\pi\,\lambda\, x) = \sum_{s=-\infty}^{\infty} \operatorname{sinc}(\pi\,(s-\lambda))\, \exp(i\,\pi\, s\, x), \tag{12}$$

where $\operatorname{sinc}(z) = \sin(z)/z$, $\operatorname{sinc}(0) = 1$, $z \in \mathbb{C}$, will play a key role in the below.

Consider the following quasi-polynomials associated with formula (12) and sequence (9)

$$\Lambda_{j,k}(x) \overset{def}{=} \frac{i^{2-k}}{2\,\pi^{k-1}(k-1)!}\,\frac{d^{k-1}}{d\,\lambda_j^{k-1}}\,(\csc(\pi\lambda_j)\,\exp(i\,\pi\,\lambda_j\,x)) =$$

$$\sum_{s=-\infty}^{\infty} \frac{(-1)^s\,\exp(i\,\pi\,s\,x)}{2\,(i\pi(s-\lambda_p))^k}, \quad \lambda_j \in \mathbb{C},\ j \in D_m,\ k \geq 1,\ x \in [-1,1]. \tag{13}$$

Theorem 1 [9, 10]. *Suppose the sequence* (9) *is given. Then the corresponding functions* $\{T_r\} \in Q_m$ *have the following explicit form*

$$T_r(x) = \sum_{j \in D_m} \sum_{k=1}^{n_j} c_{r,j,k}\, \Lambda_{j,k}(x), \quad r \in D_m,\ x \in [-1,1] \tag{14}$$

where

$$c_{r,j,k} = \frac{(-1)^{r+1}\,(i\,\pi)^k\,\prod_{p \in D_{m_1}}(r-\lambda_p)^{n_p}}{2\,(n_j-k)!\,\prod_{\substack{p \in D_m \\ p \neq r}}(r-p)}\,\frac{d^{n_j-k}}{d\lambda_j^{n_j-k}}\left(\frac{\prod_{\substack{p \in D_m \\ p \neq r}}(\lambda_j - p)}{\prod_{\substack{p \in D_{m_1} \\ p \neq j}}(\lambda_j-\lambda_p)^{n_p}}\right). \tag{15}$$

Comparing (14) with (12), we arrive at the conclusion that the system $\{T_r\}$ consists from a linear combination of functions $\{p_k(x)\exp(i\,\pi\,\lambda_k\,x)\}$, where $\{p_k\}$ are polynomials. Thus, each T_r is a quasi-polynomial.

2.3.2 Traditional scheme

Here we have in mind approximation (10) with *preassigned non-integer parameters* $\{\lambda_k\}$ (see [9]). In this case, we are actually dealing with a KE-like method if $m = 1$ (see Section 1.1), *but without solving the equation (3)*. Therefore, it can be said that this approach to the acceleration of convergence for Fourier series is completely spectral and does not use even approximate jumps of the function. In [8], this algorithm was called "universal".

2.3.3 Adaptive scheme

The adaptive algorithm is based on the *finding* parameters $\{\lambda_k\}$ *for each function f separately* according to its Fourier coefficients.

Let us describe the main idea of this approach. Together with the given set D_n and parameters$\{\lambda_q\}$, $q \in D_m$, *in symbolic form*, we choose a new set of integers $\tilde{D}_m = \{\tilde{d}_k\}$, $k = 1, \ldots, m$, $\tilde{D}_m \cap D_m = \emptyset$. Consider the following truncated Fourier series

$$S_{2m}(x) = \sum_{k \in D_m \cup \tilde{D}_m} f_k \exp(i\,\pi\,k\,x), \quad x \in [-1,1]. \tag{16}$$

To determine the parameters $\{\lambda_q\}$, $q \in D_m$, we *additionally use* Fourier coefficients $\{f_s\}$, $s \in \tilde{D}_m$, and solve the following system of equations by the least squares method

$$f_s - \sum_{r \in D_m} f_r t_{r,s} = 0, \quad s \in \tilde{D}_m \tag{17}$$

regarding parameters $\{\lambda_q\}$. Note that equation (16) is essentially nonlinear.

This approach actually optimizes the choice of parameters $\{\lambda_q\}$ for function f since it means minimizing the error corresponding to the set $\tilde{D}_m$ in formula (11). Note that it is possible to choose a set $\tilde{D}_{m_1}$, $m_1 > m$ instead (see Remark 3 in [9]).

The system $\{T_r\}$ from (14) is used to approximate f by F_n (see (11)).

2.4 Over-convergence phenomenon

The adaptive approximation is based on the *Algorithm* $\mathfrak{A}$ from [9, 11]. On its basis, the following theoretical result is obtained.

Theorem 2 (The phenomenon of the over-convergence [9, 10]) *Let $f \in Q_m$ (see Definition 2), the sets D_m, $\tilde{D}_m$, $D_n \cap \tilde{D}_m = \emptyset$ and the Fourier coefficients $\{f_s\}$, $s \in D_m \cup \tilde{D}_m$, of the function f be given. Denote by Λ the set of integer parameters in the approximation (7) $f \simeq \sum_k P_{m_k}(x) \exp(i\,\pi\,\mu_k\,x)$. In order for the approximation by Algorithm $\mathfrak{A}$ to be exact (that is, $f(x) \equiv F_m(x)$, $x \in [-1,1]$), it is necessary and sufficient that $\Lambda \subseteq D_m \cup \tilde{D}_m$.*

It is now natural to formulate the following definition of the acceleration of the convergence of a truncated Fourier series (see [9]).

Definition 3 *Let $f \in L_2[-1,1]$, the sets D_m, $\tilde{D}_m$ and the Fourier coefficients $\{f_s\}$, $s \in D_m \cup \tilde{D}_m$, of the f be given. Applying Algorithm $\mathfrak{A}$, we get parameters $\{\lambda_k\}$, $k \in D_m$. By accelerating the convergence of truncated Fourier series (15), we define the approximation F_m from (10).*

Remark 3 *There are no Fourier coefficients $\{f_r\}$, $r \in \tilde{D}_m$ in formula (10) for F_m. These coefficients are used in (16) to an "optimal" choice of parameters $\{\lambda_k\}$, $k \in D_m$.*

3 The updated scheme of approximation

3.1 Analysis of approximation (10)

Practical applications of approximation (10) have generally shown consistent performance. However, in some particular cases, numerical experiments have revealed a number of its shortcomings. Let us take a look at the main ones.

3.1.1 Traditional scheme

As the number of available m Fourier coefficients of the function f increases, the implementation of formulas in Section 2.2 leads to some computational problems. In particular, the number of arithmetic operations increases, and accordingly, more errors accumulate.

In addition, there are some problems with the practical choice of parameters depending on m. For example, one has to constantly control that the values of $\{\lambda_k\}$ are not "too close" to integers. Otherwise, the algorithm sometimes encounters type 0/0 indeterminacy and aborts.

3.1.2 Adaptive scheme

Problems listed above are much more difficult here since the solution of equation (16) by Algorithm $\mathfrak{A}$ is added (see [9, 11]).

Unfortunately, there is also an obvious omission. The fact is that if $f \notin Q_m$, then at the first stage of applying the above Algorithm $\mathfrak{A}$, the matrix of corresponding system can be degenerate. Then it is necessary to take into account the occurrence of corresponding discrepancy (see Remark 2 below).

Finally, it turns out that the occurrence of the above-mentioned uncertainties of type 0/0 is associated with insufficient justification for the generalization of formulas (8)–(14) for the case of having integer parameters in $\{\lambda_k\}$ (see Section 2.2 in [11]). As a result, it turns out that in Theorem 2, only the necessity of the last condition was proved.

3.2 Proposed approach

Below we propose a more general and practically more flexible scheme for the above (see Section 2) nonlinear spectral acceleration of convergence of Fourier series. In particular, the proof of Theorem 2 becomes complete.

3.2.1 Convergence acceleration formulas

The idea of the new approach is to use approximation (10) in the presence of $2m$ Fourier coefficients of f, *taking into account* the corresponding discrepancy. For a given partial sum $S_n(x)$ (see (6)), here we consider a more flexible scheme for restoring function f.

Now from D_n, we choose two sets D_m and $\tilde{D}_m$ ($D_m \cap \tilde{D}_m = \emptyset$) of m different integers ($m \le n/2$). Denote $\Omega_{m,n} = D_n \setminus (D_m \cup \tilde{D}_m)$ and consider the following modifications of the approximation F_m (see Section 2.2).

In the case of *traditional scheme*,

$$f \simeq F_{m,n}(x) = \sum_{r \in D_m \cup \tilde{D}_m} f_r T_r(x) + \sum_{s \in \Omega_{m,n}} (f_s - \sum_{r \in D_m \cup \tilde{D}_m} f_r t_{r,s}) \exp(i\,\pi\,s\,x). \quad (18)$$

Remark 4 *We emphasize that in order to implement approximation (18) for segment (17) of the Fourier series, no equations need to be solved:* you just need to choose $2m$ complex numbers *(see Section 2.3.2).*

In the case of *adaptive scheme*,

$$f \simeq \tilde{F}_{m,n}(x) = \sum_{r\in D_m} f_r T_r(x) + \sum_{s\in D_n\setminus D_m} (f_s - \sum_{r\in D_m} f_r t_{r,s}) \exp(i\,\pi\,s\,x). \quad (19)$$

We have the error $R_{m,n} = f - F_{m,n}$. Now in both cases above, the L_2-errors of these approximations take the form

$$\|R_{m,n}\|^2 = \sum_{s\notin D_n} \|f_s - \sum_{r\in D_p} f_r t_{r,s}\|^2$$

provided that $f \in L_2$, where $p = 2m$ for the traditional algorithm and $p = m$ for the adaptive one.

Remark 5 *In the case of even n, $n = 2m$, from (19), we have*

$$F_{m,m}(x) = \sum_{r\in D_m} f_r T_r(x) + \sum_{s\in D_{2m}\setminus D_m} (f_s - \sum_{r\in D_m} f_r t_{r,s}). \quad (20)$$

The first term on the right corresponds to approximation (10), and the second term corrects the discrepancy (see Section 3.1.2 above).

Above approximations was proposed earlier (see [9], Remark 3), but were not implemented numerically. Such an approximation has already been successfully applied in the discrete case for a finite Fourier series (see [13], formula (26)).

3.2.2 Eigen-parameters of smooth functions

Definition 4 *In the case of an adaptive algorithm, we denote by $\Lambda_{m,n}(f)$ the set of m parameters $\{\lambda_k\}$, $k \in D_m$ corresponding to approximation (19) for function f. In what follows, $\Lambda_{m,n}(f)$ will be called* eigen-parameters of *f.*

Remark 6 *These parameters have been earlier used in [3] based on the solution of equation (3). It was noted that the right-hand side of (3) is a part of a power series concerning $t = 1/s$, $s \to \infty$, and therefore, the para-diagonal Padé approximants can be applied (see Section 1.2). The zeros of denominators in these approximations actually plaid the role of eigen-parameters of the desired function f.*

The adaptability of approximation (19) for *each function* f (more precisely, for its given Fourier coefficients $\{f_r\}$, $r \in D_m$) lies in the use of this set of eigen-parameters. We emphasize that in practice, the set $\Lambda_{m,n}(f)$ depends not only on f, m and n, but also on the accuracy of setting the Fourier coefficients $\{f_s\}$ as well as on the properties of computational tools used in the realization of *Algorithm* $\mathfrak{A}$. For details, see Section 4.4.4 below.

We can say that the application of adaptive algorithm means the expansion of function f in terms of quasi-polynomials (19) using its *own biorthogonal system* $\{T_r(x), 1/2\exp(i\ \ pi\, r\, x)\}$, $r \in D_m$.

Remark 7 *The* KE-*method (see Section 1.1) is already one of the classical ones, and here we call* traditional *one of its analogs without solving equation (3). Although this algorithm also adapts to the function f through its Fourier coefficients* $\{f_k\}$*, the adaptation using eigen-parameters is much efficient (see Section 4 below) and is more "personal" for f. That is why we call such an algorithm (scheme, method, approximation, summation)* adaptive.

3.2.3 An asymptotic formula for Fourier coefficients

Let $f(x) \in C^{q-1}[-1,1]$ and $f^{(q)}(x) \in L_1[-1,1]$. Asymptotic of the Fourier coefficients $\{f_s\}$ corresponding to the given parameters $\{\lambda_k\}$ has the form (see Section **2.4** in [11])

$$f_s = \frac{(-1)^s\, P_{q-1}(s)}{2\,\pi^q \prod_{k=1}^{q}(s-\lambda_k)} + r_q(s), \tag{21}$$

where

$$r_q(s) = \frac{(-i)^q \int_{-1}^{1} e^{-i\,\pi\, s\, t} \prod_{k=1}^{q}(D_t - i\pi\lambda_k)\, f(t)\, dt}{2\,\pi^q \prod_{k=1`}^{q}(s-\lambda_k)},$$

and $P_{q-1}(s)$ is a corresponding polynomial of order at most $(q-1)$ whose coefficients depend on the jumps of function f and its derivatives at the ends of the segment $[-1,1]$ (see (1)). The symbol D_t means differentiation by t. It can be seen that for $\prod_{k=1}^{q}((D_t - i\pi\lambda_k)\, f(t) \equiv 0$ (that is, when f is a quasi-polynomial), $r_q(s) = 0$ for all s.

The remainder can also be exactly represented as

$$r_q(s) = \frac{(-i)^q \int_{-1}^{1} \left(\sum_{k=0}^{q} e_k f^{(q-k)}(x)\right) e^{-i\pi s x}\, dx}{2\,\pi^q \prod_{k=1}^{q}(s-\lambda_k)},$$

$$e_0 = 1, \quad e_k = (-1)^k \sum_{i_1<i_2<\cdots<i_k} \lambda_{i_1}, \ldots, \lambda_{i_k}.$$

The integrals in the numerator containing $f^{(q-k)}$, $k = 1, \ldots, q$, are of the order $O(s^{-1})$, $s \to \infty$. From here,

$$f_s = \frac{(-1)^s\, P_{q-1}(s)}{2\,\pi^q \prod_{k=1}^{q}(s-\lambda_k)} + \frac{(-i)^q \int_{-1}^{1} f^{(q)}(x)\, e^{-i\,\pi\, s\, x}\, dx}{2\,\pi^q s^q}(1 + O(s^{-1})). \tag{22}$$

3.2.4 New initial constructions

Let $\mathcal{P}_m(z)$, $z \in \mathbb{C}$, be a polynomial

$$\mathcal{P}_m(z) = \sum_{k=0}^{m} a_k\, z^k, \quad a_m = 1,\, a_k \in \mathbb{C},\, k = 0, \dots, m-1, \tag{23}$$

and $\{\lambda_k\}$, $k \in D_m$, be the set of its zeros. Then formula (8) can sometimes be represented in a form that is more convenient for us (see Section 3.5 below)

$$t_{r,s} \overset{def}{=} (-1)^{s-r} \Big(\prod_{\substack{p \in D_m \\ p \neq r}} \frac{s-p}{r-p} \Big) \frac{\mathcal{P}_m(r)}{\mathcal{P}_m(s)}, \quad r \in D_m,\ s \in \mathbb{Z}. \tag{24}$$

For the rest, we will use the results of Sections 2.3 and 3.2.1. This approach is not a formal procedure since it contributes to a reduction in the complexity of implemented algorithms (see Remark 7 below.)

3.3 The case of integer parameters in (8)

Is it possible to allow the presence of integer parameters in formula (8)? First of all, the answer to this question is important since such a case can happen when solving equation (16). No less important is the fact that it would be natural to include in our scheme the entire set of quasi-polynomials Q_m (see Definition 2 above).

This case was studied in [11] but the corresponding recommendations were not enough (see Section 3.1.2 above). Below there is a different approach that makes the situation clearer.

3.3.1 Some heuristic considerations for the adaptive case

The simplest quasi-polynomial is Euler‘s exponential $p(x) = \exp(i\,\pi\,\lambda\,x)$, $x \in [-1, 1]$, $\lambda \in \mathbb{C}$. If λ is not an integer, all its Fourier coefficients $\{p_s\}$ are nonzero (see (12)). Therefore, you can “feel”, “see” this function having only *some two* of its Fourier coefficients (see an example in Section 4.2 of [9]).

A completely different situation arises in the case of integer λ (that is, in the case when λ is an eigenvalue of the orthogonal Fourier system). Here we can “feel” the presence of $p(x)$ *only when we have* its Fourier coefficient g_λ since the rest of its coefficients are zero.

The meaning of Theorem 2 is that a similar situation holds for any quasi-polynomial $q(x) \in Q_n$ (see Definition 2). Let eigen-parameters $\{\lambda_k\}$ be given in symbolic form. Consider formula (9) in Remark 2, which takes into account the multiplicity of parameters

$$t_{r,s} = (-1)^{s-r} \prod_{\substack{p \in D_m \\ p \neq r}} \frac{s-p}{r-p} \prod_{j \in D_{m_1}} \left(\frac{r - \lambda_j}{s - \lambda_j} \right)^{n_j}, \quad r \in D_m, \ s \in \mathbb{Z}.$$

Let now for a $j = q \in D_m$, λ_q be an integer. Then we see that with $r \neq q$ in this formula, we can reduce the fraction, and instead of m, we actually have $(m-1)$, while instead of n_q, we have $(n_q - 1)$.

With $r = q$, we have $t_{r,s} = 0$ for all $s \neq q$, which means a Fourier exponent $T_q(x) = \exp(i \, \pi \, \lambda_q \, x)$. For $n_q = 1$, we have solved the problem of admitting an integer parameter to the adaptive algorithm scheme simultaneously reducing the number of operations in the construction of system $\{T_r(x), \, r \notin q\}$.

In the case $j = q \in \tilde{D_m}$, it would seem that the situation is different, and it is necessary to look for other ways. But taking into account formula (15) in Theorem 1, we see that similar reductions can be made for the coefficients $c_{r,j,k}$ and to come to the same conclusions.

3.3.2 Some details and recommendations

First of all, we note that the eigen-parameters $\{\lambda_k\}$ of function f are uniquely determined, and the problem of its integers arises only when constructing system $\{T_r\}$ (see Section 3.4 below). The second circumstance is that after abbreviations specified in Section 3.3.1, systems $\{T_r\}$, $\{\lambda_k\}$ and $\{f_k\}$ fall into “independent” parts.

For $n > 2m$ and $q \in D_n$, $q \notin \tilde{D_m}$, the algorithm continues ambiguously since there are several possibilities to change the location of set $\tilde{D_m}$ in D_n. Each user can act in his way.

Finally, we note that in case $\lambda_q \notin D_n$ not mentioned above, the situation again turns out to be ambiguous and requires clarifications. At the final stage of Algorithm $\mathfrak{A}$ (see the next section) in this case, it is provided to cancel the calculations already made. But the search for an approximate finding of function f ends here *when we can only* know the Fourier coefficients $\{f_s\}$, $s \in D_n$ (see (19)). Otherwise, we have the option of adding to D_n some new Fourier coefficients and run the algorithm again.

3.4 Refined Algorithm $\mathfrak{A}$

Let us present the main stages of the refined algorithm, which serves as the basis for substantiating the phenomenon of super-convergence (see Theorem 2 above).

Recall that our goal is to solve non-linear equation (16) (see Section 2.3.3). Here, instead of (8), we use representation (25).

Step 1. Using representation (25) and reducing equation (15) to the common denominator, we have

$$f_s\,\mathcal{P}_m(s) = \sum_{r\in D_m} (-1)^{s-r} f_r \,\Big(\prod_{\substack{p\in D_m\\ p\neq r}} \frac{s-p}{r-p}\Big)\,\mathcal{P}_m(r), \quad s\in \tilde{D}_m. \tag{25}$$

As a result, equations (15) takes the form of a linear system of equations with respect to the coefficients of polynomial $\mathcal{P}_m(z)$.

Step 2. Solve resulting equation by the least square method and find the polynomial $\mathcal{P}_m(z)$.

Step 3. Find all roots $\{\lambda_k\}\subset\mathbb{C}$ of polynomial $\mathcal{P}_m(z)$ and compute the values

$$(f_s - \sum_{r\in D_m} f_r\, t_{r,s}), \quad s\in \tilde{D_n}.$$

Step 4. a) If there are no integer values in $\{\lambda_k\}$, then at the output derive approximation formula (19).

b) If $\{\mu_p\}\neq\emptyset$, $\{\mu_p\}\subset\{\lambda_k\}$ is all integer values and $\{\mu_p\}\subset D_n$, then at the output derive an approximation formula using the descriptions described in Section 3.3.1 above.

c) If $\{\mu_p\}\neq\emptyset$, $\{\mu_p\}\subset\{\lambda_k\}$ is all integer values and there is $\mu_p\notin D_n$, then interrupt the calculations and go back to *Step 1* to follow recommendations of Section 3.3.2 above.

Remark 8 *Comparing the above procedures with the description of* Algorithm $\mathfrak{A}$ *in works [9, 11], one can notice two differences. First, the earlier equation (25) was solved only after some preliminary procedures, thus, here the complexity of algorithm is somewhat reduced.*

Second (and this is much more important), the final Step 4 *leads to a full justification for over-convergence in Theorem 2 based on the proofs given in [9] and [11].*

3.5 Some theoretical estimates of the convergence

3.5.1 The case of classical truncated Fourier series

To study the properties of proposed algorithms, further we restrict ourselves to the classical partial sums of the Fourier series

$$S_n(x) = \sum_{k=-n}^{n} f_k \exp(i\,\pi\,k\,x), \quad n\geq 1,\ x\in[-1,1]. \tag{26}$$

Now let us denote for integers $1\leq m\leq n$ (see Section 3.2.1),

$$\begin{gathered} D_m = \{n-m+1,\ldots,n\}, \quad \tilde{D}_m = \{-n,-n+1,\ldots,-n+m-1\}, \\ D_n = \{-n,-n+1\ldots,n\}. \end{gathered} \tag{27}$$

Here we allowed an "extra" Fourier coefficient f_0 for the sake of using the classical partial sum (26) (compare with (19)) but this does not matter.

The sets D_m and $\tilde{D}_m$ can be changed if there are integer values among the parameters $\{\lambda_k\}$ (see Sections 3.3 and 3.4).

Remark 9 *This choice of sets D_m and $\tilde{D}_m$ is primarily due to the asymptotic properties of the Fourier coefficients $\{f_s\}$ (see Section 3.2.3 above). Besides, we concluded that equation (17) is better solved for "tight" values for the indices s (see (27)).*

3.5.2 Rate of convergence for functions f of bounded smoothness

Here we are interested in an estimation of the error $R_{m,n}$ when m is fixed and $n- > \infty$ (see Section 3.2.1).

Theorem 3 *Let $f(x) \in C^{q-1}[-1,1]$, $2 \le q \le \infty$ and $q_1 = \min(2m, q)$ for traditional case and $q_1 = \min(m, q)$ for adaptive.*

1) *If $|f^{(q)}(x)| \le M$, $M = const > 0$, $x \in [-1,1]$, then*

$$\lim_{n\to\infty} n^{q_1-1} \|R_{m,n}\|_\infty \le \frac{M}{\pi^{q_1}}.$$

2) *If $f^{(q)}(x) \in L_1[-1,1]$, then*

$$\lim_{n\to\infty} n^{q_1-1} \|R_{m,n}\|_1 \le \frac{1}{\pi^{q_1}} \|f^{(q_1)}(x)\|_1.$$

3) *If $f^{(q)}(x) \in L_2[-1,1]$, then*

$$\lim_{n\to\infty} n^{q_1-1/2} \|R_{m,n}\|_2 \le \frac{1}{\pi^{q_1}} \|f^{(q_1)}(x)\|_2.$$

Proof. For the sake of simplicity, we will assume that all parameters $\{\lambda_k\}$ are non-integer.

We use asymptotic formula (22) for $q = q_1$. According to formula (18), it is necessary to estimate the value of $(f_s - \sum_{r\in D_m} f_r t_{r,s})$ at $|s| > n$, $n \to \infty$ (and additionally for $s \in \tilde{D}_m$ in (19)). To do this, it is enough to use the first (main) term of (22) on the right, since the rest have order of $o(n^{-q_1-1})$ and $m = const$ for $n \to \infty$. That is why

$$f_s - \sum_{r\in D_m} f_r t_{r,s} = O(n^{-q_1}) + \frac{(-1)^s P_{q_1-1}(s)}{2\,\pi^{q_1} \prod_{k=1}^{q_1} (s - \lambda_k)} - \sum_{r\in D_m} f_r t_{r,s}. \tag{28}$$

Consider the last sum in more detail, bearing in mind that asymptotic formula (22) can be applied to f_r according to choice (27) since $r = O(n)$ at

$n \to \infty$. That is why (see also (8))

$$\sum_{r\in D_m} f_r t_{r,s} = O(n^{-q_1}) + \sum_{r\in D_m} \frac{(-1)^s P_{q_1-1}(r)}{2\,\pi^{q_1} \prod_{k=1}^{q_1}(r-\lambda_k)} \prod_{\substack{p\in D_m\\ p\neq r}} \frac{s-p}{r-p} \prod_{j\in D_m} \frac{r-\lambda_j}{s-\lambda_j}$$

$$= O(n^{-q_1}) + (-1)^s \sum_{r\in D_m} \frac{P_{q_1-1}(r)}{2\,\pi^{q_1}} \prod_{\substack{p\in D_m\\ p\neq r}} \frac{s-p}{r-p} \prod_{j\in D_m} \frac{1}{s-\lambda_j}.$$

Hence, formula (28) takes the form

$$f_s - \sum_{r\in D_m} f_r t_{r,s} = O(n^{-q_1}) + \frac{(-1)^s \left(P_{q_1-1}(s) - \sum\limits_{r\in D_m} \prod\limits_{\substack{p\in D_m\\ p\neq r}} \frac{s-p}{r-p} P_{q_1-1}(r)\right)}{2\,\pi^{q_1} \prod\limits_{k=1}^{q_1} (s-\lambda_k)}.$$

Note that the expression in brackets of the numerator on the right is equal to zero by the accuracy of Lagrange interpolation at m nodes for polynomials of degree at most m. It remains to estimate the error $R_{m,n}$ from above (see Section 3.2.1) □

Remark 10 *The estimates in Theorem 3 are too rough for the adaptive algorithm because the eigen-parameters of function f cannot be used in proof.*

3.5.3 Rate of convergence for analytic functions f

Let us now consider the case when function f is analytic. Here our goal is to estimate the error $R_{n,m}$ at $m = n$ when $n \to \infty$. We can say that here we use the "maximum power" of the algorithms using a sharp increase in its complexity (see Section 3.1.2). Unfortunately, in this case too, we fail to use eigen-parameters of function being approximated.

Theorem 4 *Let $f(z)$ be analytic in a bounded closed domain D with Jordan rectifiable boundary ∂D and the segment $x = Re(z) \in [-1,1]$, $Im(z) = 0$, belongs to D.*

i. *If the distance of boundary points $t \in \partial D$ of the region D to segment $x \in [-1,1]$ is more than the value of $(\pi e)^{-1} \simeq 0.117$, then there exists $0 < \theta = const < 1$, such that*

$$|R_{m,n}(x)| < C\,\theta^n, \quad C = const > 0,\ x \in [-1,1],\ n \to \infty.$$

ii.*If $f(z)$ is an entire function, then in the last estimate the constant θ can be arbitrarily small.*

iii.*If $f(z)$ is an entire function of exponential type, then*

$$|R_{m,n}(x)| < \left(\frac{C}{n}\right)^n, \quad C = const > 0,\ n \to \infty.$$

Proof. Representing the function $f(x)$, $x \in [-1,1]$ in terms of the Cauchy integral along the contour ∂D and differentiating it n times, we obtain

$$f^{(n)}(x) = \frac{n!}{2\,\pi\,i} \int_{\partial D} \frac{f(t)}{(t-x)^n}\,dt, \quad x \in [-1,1]. \tag{29}$$

Let d be the distance between segment $x = Re(z) \in [-1,1]$ and ∂D. By the conditions of the theorem, function $f(t)$, $t \in \partial D$, is bounded and $d < (\pi e)^{-1}$. Obviously, there is a number $0 < \epsilon < 1$ such that $d < ((1-\epsilon)^2 \pi e)^{-1}$.

It follows from formula (29) and the above estimates that

$$|f^{(n)}(x)| < C_1 \sqrt{n}\,(1-\epsilon)^{2n}\,n! < C_2\,n!\,(1-\epsilon)^n, \quad C_1, C_2 = const.$$

On the other hand, according to the refined Stirling formula (see, for example, [16], Section 21.4-2), we have

$$\sqrt{2\pi n}\,\left(\frac{n}{e}\right)^n < n! < e^{\frac{1}{12}}\,\sqrt{2\pi n}\,\left(\frac{n}{e}\right)^n.$$

It remains to apply these estimates to the last term on the right-hand side of formula (28) for $q = n$, $|s| > n$ and follow the outline of the proof of Theorem 3. □

Remark 11 *Theorem 4 can be significantly strengthened by using the geometric properties of the boundary ∂D in the statement* i). *In particular, this applies to the case when f has only singularities in the form of poles in the complex plane (see, for instance, function $f_2(x)$ in (31) below).*

We note that earlier in [18], based on the KE-method with the choice of some function g (see the end of Section 1.1), estimate **i** of Theorem 4 was obtained for the case of piecewise-smooth function f, that is, analytic at smooth points.

4 Numerical implementation

Below we present information about our experiments to test the practical possibilities of the algorithms proposed above in Sections 3.2 and 3.5. Our goal is a detailed comparison of the *traditional* algorithm, whose simple scheme uses only Krylov's approach (see decomposition (2)), with the *adaptive* algorithm based on the recently discovered *over-convergence phenomenon.*

4.1 Test functions

Numerical experiments were carried out with the following five smooth functions.

$$f_1(x) = \begin{cases} 0, & -1 \le x < -1/3; \\ \dfrac{29}{20}\sin^5\left(\dfrac{3}{2}\left(x+\dfrac{1}{3}\right)\right), & -1/3 \le x \le 1; \end{cases} \tag{30}$$

$$f_2(x) = \frac{\cos(x)}{(2/3+i)-x}, \qquad f_3(x) = \frac{63}{89}\, e^{2i\left(x+\frac{2}{3}\right)^2},$$

$$f_4(x) = \left(x+\frac{41}{21}\right)\operatorname{sinc}(4\,x+\pi),\ f_5(x) = e^{-4i\pi x}\left(x+\frac{2i}{5}\right)+\frac{f_4(x)}{10}. \tag{31}$$

Here $f_1 \in C^5$ and $f_1^{(5)}$ is a piece-wise smooth function with two jumps: $A_{1,5} \simeq -1321$ at point $x=-1/3$ and $A_{2,5} \simeq 2610$ at point $x=1$ (see (1)). The other three functions are analytic in a neighborhood of segment [-1,1]. Function $f_2(z)$ is rational and has only one pole at $z = 2/3+i$. Functions f_3 and f_4 are entire. Function f_4 has an exponential type (see explanation after formula (9) above). The last function f_5 is a quasi-polynomial with an integer parameter perturbed by function $f_4/10$.

Bearing in mind experiments using different numbers of Fourier coefficients obtained with different accuracy, we have chosen functions whose Fourier coefficients are represented exactly (symbolically) using *Wolfram Mathematica* system.

For the sake of convenience (see Section 4.3.1 below), L_2-norms of all these functions are chosen to be approximately equal to one. More precisely, $|1-||f_k(x)||_2| < .01$ for all k.

4.2 Implementation notes

Let us present some details on the numerical implementation of the proposed algorithms.

4.2.1 Used algorithms

For our *traditional algorithm*, the following $2\,m$ parameters are applied (see (18))

$$\lambda_k = \frac{2}{\pi}\arctan\left(\frac{1}{5}+\frac{n}{33}\right)\,k, \quad n-m+1 \le |k| \le n, \tag{32}$$

which depend on n and m. For $m = const$ and $n \to \infty$, we have $|\lambda_k/k| \uparrow 1$. These parameters $\{\lambda_k\}$ are real and (at least for $1 \le m \le n \le 100$) non-integer.

Remark 12 *The traditional algorithm corresponding to formula (32) proved to be quite stable in this range. A similar choice was made in Section 5.1.1 of [9], and in Section 3.3, some other choices were indicated. In particular, if $\lambda_k = 0$ for all k, we arrive at a polynomial approximation. In addition, we indicate here a simple formula*

$$\lambda_k = \frac{n}{n+\epsilon}\, k, \quad \epsilon = const > 0,\, n-m+1 \le |k| \le n,$$

which naturally corresponds to the quasi-periodic interpolation algorithm proposed in [14] for a discrete case (see Remark 14 below).

Recall that *adaptive approximation* uses m eigen-parameters (see (19) and Remark 3), which are calculated by applying *Algorithm* $\mathfrak{A}$ (see Section 3.4).

4.2.2 Minimizing the complexity of the computation

Consider the implementation of formula (19) from the computational point of view. System $\{T_r\}$ has the following form (see (14) in Theorem 1)

$$T_r(x) = \sum_{j\in D_m} \sum_{k=1}^{n_j} c_{r,j,k}\, \varLambda_{j,k}(x), \quad r \in D_m,\ x \in [-1,1]$$

where constants $\{c_{r,j,k}\}$ are also given explicitly. Here we see that different functions $\{T_r\}$ contain different linear combinations of the *same quasi-polynomials* $\{\varLambda_{j,k}\}$. In approximation (19), expansion $\sum_{r\in D_m} f_r\, T_f(x)$ is used but it is much more advantageous to represent this expansion as a linear combination of quasi-polynomials $\{\varLambda_{j,k}\}$ (here $\{f_r\}$ are Fourier coefficients of f). Thus, we reduce the number of multiplications by about m times.

The same approach to formula (18) reduces the number of multiplications by about $2\,m$ times. As a result, only linear combinations of quasi-polynomials are produced at the outputs of our algorithms (see Section 2.1).

4.2.3 Some details of approximation codes

Approximation codes are written in *Wolfram Mathematica* language (see [17]). The calculations were performed on a standard personal computer.

This calculations were carried out using from 24 to 48 decimal places (from 24 to 48 *working precision* by *Wolfram Mathematica* terminology). For example, for $n = 8$ and $1 \le m \le 4$, we used 24 places, and for $n = 24$ and $5 \le m \le 8$, we used 48. The exceptions are cases $n = 24$, $m = 9$ and $m = 10$, where 64 decimal places were used (see Table 7 below). Choosing a given working precision for algorithms, we translate in advance the exact (symbolic) values of the Fourier coefficients into numerical of the same accuracy.

In some cases, it turns out to be useful to additionally process the data produced by *Algorithm* $\mathfrak{A}$. For example, when applying a working precision equal to w, we consider that two values of eigen-parameters λ_1 and λ_2 are equal to each other if $|\lambda_1 - \lambda_2| \leq w/3$. Such a seemingly rigid approach has justified itself in numerous experiments preventing cases of interruption of calculations due to the appearance of the uncertainty of type $0/0$.

With integer eigen-parameters, the implementation of shortcuts described in Section 3.3 can be difficult. Here, the following simple method is quite effective: designate such parameters symbolically and only at the output of the algorithm go by the limit to their numerical values.

The estimation of complexity of the algorithms used in the above conditions depends on many circumstances, especially in the adaptive case. Below to compare the relative complexity of traditional and adaptive algorithms, their execution times are used, including the symbolic representation as a quasi-polynomial in the output.

4.3 Numerical results

4.3.1 Designations

In Tables 1–10 below, the following designations are used:
Alg. — algorithm; *Trd.* — traditional; *Adp.* — adaptive; f_k — functions $\{f_k(x)\}$, $k = 1, 2, \dots, 5$.

Relative errors for approximations $\tilde{A}_{1,k}$ of jumps $A_{1,k}$ (see (1) above) are calculated by the formula

$$h_k = \begin{cases} \dfrac{|A_{1,k} - \tilde{A}_{1,k}|}{|A_{1,k}|}, & A_{1,k} \neq 0; \\ 0, & A_{1,k} = 0. \end{cases} \tag{33}$$

Relative L_2-error for an approximation $\tilde{f}$ of function f is calculated by the formula

$$\|f - \tilde{f}\| = \begin{cases} \dfrac{\|f - \tilde{f}\|_2}{\|f\|_2}, & \|f\|_2 \neq 0; \\ 0, & \|f\|_2 = 0. \end{cases} \tag{34}$$

It should also be noted that in Tables 4,5 and **7**, f'_k means the term-by-term differentiation of approximation $\tilde{f}_k$ concerning x. The denominator in formula (34) for $\|f'_k - \tilde{f}'_k\|$ is usually not unity.

Remark 13 *In the tables and figures below, nowhere do Fourier coefficients* $\{f_s\}$ *are not involved. Therefore, designations* f_k *refers only to the functions of Section 4.1.*

4.3.2 Results tables

Alg. ↓	n=8	n=12	n=16	n=20	n=24	n=28	n=32
Classic	5.7e-2	4.7e-2	4.1e-2	3.7e-2	3.4e-2	3.1e-2	2.9e-2
time(**sec.**)	1.6e-2	1.6e-2	3.1e-2	3.1e-2	3.1e-2	1 e-1	1.1 e-1

Table 1. *L_2-errors and operating time of classic Fourier orthogonal decomposition for $f_4(x)$.*

Alg ↓	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
Trd.	7.8e-3	1 e-3	1.4e-4	4.8e-5	1.3e-3	3.3e-4	6.4e-4	1.1e-3
sec.	3.1e-2	4.7e-2	1.2e-1	2e-1	3.3e-1	4.8e-1	6.6e-1	8.9e-1
Adp.	2.9e-4	2.2e-6	1 e-4	5.3e-6	1.1e-5	5.1e-7	1 e-5	5.6e-6
sec.	1.6e-2	1.6e-2	3.1e-2	3.1 e-2	4.7e-2	6.2e-2	6.2e-2	7.8e-2

Table 2. *L_2-errors and operating time for function $f_1(x)$ at n=16.*

Alg. ↓	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
Trd.	1 e-2	2 e-3	4.2-4	7.7e-5	3.1e-5	1.2e-4	2.5e-4	3.7e-4
sec.	3.1e-2	7.8e-2	1.7e-1	3.3e-1	5.2e-1	7.7e-1	1.1	1.5
Adp.	1 e-4	1.9e-7	5.5e-6	1.4e-7	2.7e-6	5.1e-7	7.1e-6	7.8e-7
sec.	1.6e-2	1.6e-2	3.1e-2	3.1e-2	4.7e-2	6.2e-2	7.8e-2	9.4e-2

Table 3. *L_2-errors and operating time for function $f_1(x)$ at n=24.*

Alg. ↓	m=2	m=3	m=4	m=5	m=6	m=7	m=8
Trd. f_2	1.2e-3	2.5e-4	5.3e-5	1.2e-5	2.5e-6	5.6e-7	1.2e-7
Trd. f_2'	1.5e-1	3.5e-2	8.4e-3	2 e-3	4.6e-4	1.1e-4	2.4e-5
sec.	1.2e-1	2.0e-1	3.4e-1	5.3e-1	8.0e-1	1.1	1.5
Adp. f_2	2.3e-8	1.6e-11	1.1e-14	1.1e-15	5.1e-18	8.6e-20	3.9e-21
Adp. f_2'	3.1e-6	2.6e-8	1.8e-12	2e-13	1e-15	1.9e-17	8.5e-19
sec.	3.1e-2	4.7e-2	6.2e-2	7.8e-2	3.6e-1	1.1e-1	1.1e-1

Table 4. *L_2-errors and operating time for functions $f_2(x)$ and $f_2'(x)$ at n=24.*

Alg. ↓	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
Trd. f_3	7.1e-3	1.4e-3	2.9e-4	6.3e-5	1.3e-5	2.9e-6	6.4e-7	1.4e-7
Trd. f_3'	2.3e-1	5.5e-2	1.3e-2	3.1e-3	7.2e-3	1.6e-4	3.8e-5	8.6e-6
sec.	1.1e-1	9.4e-2	1.9e-1	3.3e-1	5.3e-1	7.8e-1	1.1	1.5
Adp. f_3	4.2e-4	2.5e-7	3.8e-9	6.9e-12	1.4e-13	5.4e-16	1.4e-17	1.1e-19
Adp. f_3'	1.4e-2	1.1e-5	1.9e-7	3.7e-10	8.1e-12	3.4e-14	9.4e-16	7.6e-18
sec.	4.7e-2	1.6e-2	3.1e-2	4.7e-2	4.7e-2	6.2e-2	7.8e-2	1.1e-1

Table 5. *L_2-errors and operating time for functions $f_3(x)$ and $f_3'(x)$ at n=24.*

Alg.	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
Trd.	3.2e-3	2.4e-4	1.6e-5	1 e-6	5.4e-8	2.4e-9	9.1e-11	4.5e-11
sec.	2 e-1	6.2e-2	9.4e-2	1.2e-1	1.7e-1	2.3e-1	3 e-1	3.4e-1
Adp.	2.8e-4	5.9e-5	4.8e-7	1.3e-9	2.6e-11	8.5e-13	1.5e-12	2.7e-13
sec.	1.6e-2	3.1e-2	3.1e-2	3.1e-2	4.7e-2	6.2e-2	6.2e-2	6.2e-2

Table 6. *L_2-errors and operating time for function $f_4(x)$ at n=8.*

Alg. ↓	m=1	m=3	m=5	m=7	m=9	m=10
Trd.for f_4	5.9e-3	2.5e-4	1.2e-5	5.9e-7	3.0 e-8	6.7e-9
Trd.for f_4'	2.8e-1	1.7e-2	9.6e-4	5.3e-5	2.8e-6	6.6e-7
sec.	3.0e-1	2.3e-1	5.8e-1	1.1	1.9	2.4
Adp.for f_4	1.7e-5	2.1e-10	2.3e-17	5.6e-22	1.0 e-28	1.4e-30
Adp.for f_4'	8.8e-4	1.6e-8	2.0 e-15	5.6e-20	1.1e-26	2.2e-29
sec.	1.9e-1	6.2e-2	6.2e-2	1.2e-1	1.6e-1	1.9e-1

Table 7. *L_2-errors and operating time for functions $f_4(x)$ and $f_4'(x)$ at n=24.*

w ↓	m=2	m=3	m=4	m=5	m=6	m=7	m=8
12	1.9e-5	1.1e-6	2.0e-1	9.9e-1	1.0	1.0	1.0
16	1.9e-5	1.1e-6	3.6e-8	9.9e-1	1.0	∞	∞
24	1.9e-5	1.1e-6	3.6e-8	5.2e-8	9.9e-1	9.9e-1	9.9e-1
28	1.9e-5	1.1e-6	3.6e-8	5.2e-8	3.7 e-10	3.2-11	6.0e-11

Table 8. *Impact of computer's* $\boldsymbol{w}$*-digit precision to* L_2*-errors for function* $f_5(x)$ *at n=8 when adaptive algorithm is applied.*

$\mathbf{f}_k$ ↓	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
f_1	3e-3	2e-2	9e-6	7e-1	7.4	3.8	8.4	3.4	14
f_2	1e-9	3e-7	9e-6	1e-4	1e-3	6e-3	3e-2	1e-1	3e-1
f_3	2e-7	1e-6	7e-6	5e-5	6e-5	6e-4	9e-4	3e-3	6e-3
f_4	8e-12	9e-12	2e-9	2e-9	7e-8	8e-8	1e-6	2e-6	2e-5

Table 9. *Errors of relative jumps* $\{h_k\}$ *of some functions at* $n = m = 8$ *(see (33)). The adaptive algorithm is applied.*

$\mathbf{f}_k$ ↓	m=3	m=4
f_1	$\{-5 + 3.4i,\ 1.4i,$ $5 + 3.4i\}$	$\{-22.5 + .9i, -1.2 + .9i,$ $1.2 + .9i, 22.5 + .9i\}$
f_2	$\{-1.7 + .4i, .5 + .04i,$ $.27 - .03i\}$	$\{-3.0 + .75i, -1.5 + .23i,$ $-.48 - .006i, .26 - .001i\}$
f_3	$\{.37 + .54i, .7 + 2.4i,$ $2.3 + .51i\}$	$\{.97 + 1.0i, .01 + .27i,$ $2.0 - .27i, 2.9 - .98i\}$
f_4	$\{\ -1.6 - .16i, -.19i,$ $1.6 - .16i\}$	$\{-1.29 - .023i, -.5 + .05i,$ $.5 + .05i, 1.29 - .023i\}$
f_5	$\{\ -4.0, .1 - .46i,$ $1.76 + .56i\}$	$\{-4.0, -1.8 - .54i,$ $0.11 + .02i, 1.48 - .08i\}$

Table 10. *Some eigen-parameters of functions* $\{f_k(x)\}$*,* $k = 1, \ldots, 5$*, at n=24 (values are rounded off).*

4.4 Comments and recommendations

Let us make some comments on the numerical results characterizing the features of the method described in Section 3 above.

4.4.1 Efficiency and stability of algorithms

As a rule, we see a huge advantage of the adaptive algorithm in both accuracy and speed.

For example, at n=16, in the case of the function of bounded smoothness f_1, the smallest error of the traditional algorithm is achieved at $m = 6$ (see Table 2). At the same time, the same smallest error of the adaptive algorithm is achieved at $m = 1$ but for $m = 6$, the error is much smaller. It is worth noting here that (see Section 2.2 and formula (19) for $m = 1$) the system $\{T_r(x)\}$ consists of ... *only one function.*

Moreover, for $m = 2, 3$, eigen-parameters of function f_1 are determined *explicitly* but the accuracy of the adaptive algorithm is unattainable for the traditional one both for $n = 16$ and $n = 24$.

As for analytic functions f_2, f_3 and f_4, here the difference in efficiency of the algorithms increases many times over (see Tables 3–7).

If, for example, we pay attention to Tables 4, 5 and 7, which additionally contain data on the effectiveness of term-by-term differentiation of formulas (18) and (19), we conclude that the traditional algorithm cannot stand any comparison with the adaptive one.

Same can be said about the speed of adaptive algorithms. For example, if in Table 7 at $m = 1$ the time of the adaptive algorithm is three times less than that of the traditional one, then, at $m = 10$, this difference increases to twelve, although the adaptive algorithm additionally includes one pseudo-inversion of a 10×10-matrix and calculation of all roots for a polynomial of degree 10.

In the case of increased requirements for accuracy, it is recommended to use the maximum possible numerical capacity of the computer's processor used. This will also avoid the consequences of having integers in the set of eigen-parameters, as, for example, in the case of function f_5 (see Tables 7 and 9).

4.4.2 Errors of jumps

Given the exceptional properties of the adaptive algorithm, it is interesting to know how accurately it restores jumps of a smooth function (see (1)). In particular, we can compare our approach both with KE-method and with works [3–8] (see Introduction).

Table 9 shows the corresponding absolute values of the relative jump errors at $n = m = 8$. Comparing the results of Table 6 for function f_4, we conclude that, for example, even h_0 is about 31 times more than L_2-error for f_4. The remaining values of $\{h_k\}$ increase rapidly. Note that (see (1)) $|A_{1,0}| \simeq 5.3\,10^{-1}$, $|A_{1,4}| \simeq 90$ and $|A_{1,8}| \simeq 2.6\,10^{4}$.

A similar situation is observed for functions f_2 and f_3. The exception is function f_1, the fifth derivative of which has an additional unaccounted jump at point $x = -1/3$, and therefore, only its first four jumps can be considered approximately certain.

In general, we can say that it is impossible to consider the accuracy of restoring jumps of a function using the adaptive algorithm to the corresponding its excellent L_2-accuracy (see also Remark 14 below).

4.4.3 About eigen-parameters of smooth functions

The most important element of the adaptive algorithm is the determination of eigen-parameters of function f (see Section 3.2.2). They are calculated by applying *Algorithm* $\mathfrak{A}$ and are the most important characteristic of every smooth function f.

Table 10 shows some of their values at $n = 24$ in highly rounded form (these are 32-digit numbers in originals). Tables 2–4, 5 and 7 also refer to the case $n = 24$, thus, data can be compared.

First of all, we note that not only the differences in eigen-parameters of different functions but also their changes for the same function for different m. For example, the integer (-4) appears in the system of eigen-parameters of function f_5 both for $m = 3$ and $m = 4$ but, in fact, there is a non-zero value in the range from 7-th to the 11-th characters of the originals (see Section 4.2.3 above).

In this example, we ones more see how important it is to use high-bit arithmetic.

4.4.4 Approximate analytic continuation

The previous conclusions are also confirmed by observations of the propagation into a complex domain for approximation of function f that is analytic on the segment [-1,1].

Fig. 1 below shows the level lines of function $|f_2(z)|$, $z = x + yi$, in rectangle $x \in [-1.5, 1.3]$, $y \in [-1, .5]$ given in the aspect ratio of $1 : 1/2$. We see that approximately $.2 \leq |f_2(z)| \leq 1.4$. The white segment below the top boundary is near pole $z = 2/3 + i$ of function $f_2(z)$.

Fig. 2 and 3 show corresponding error level lines when the traditional and adaptive algorithms are applied at $m = 8$, respectively. It is seen that even at $n = 8$, the error of analytic continuation using the adaptive algorithm is about 10^4 times smaller. Interestingly, the smallest error (less than 10^{-12}) lies in the region outside of the segment of the original adaptive approximation.

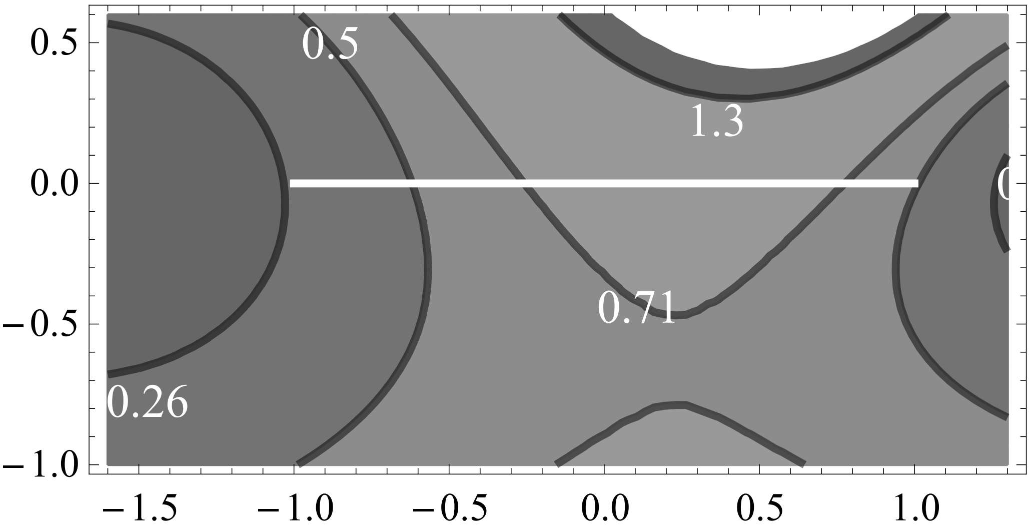

Figure 1: *Level lines of function* $|f_2(z)|$, $z = x + y\,i$, *in rectangle* $x \in [-1.5, 1.3]$, $y \in [-1, .5]$. *Segment* $x \in [-1, 1]$, $y = 0$ *is shown as a white line.*

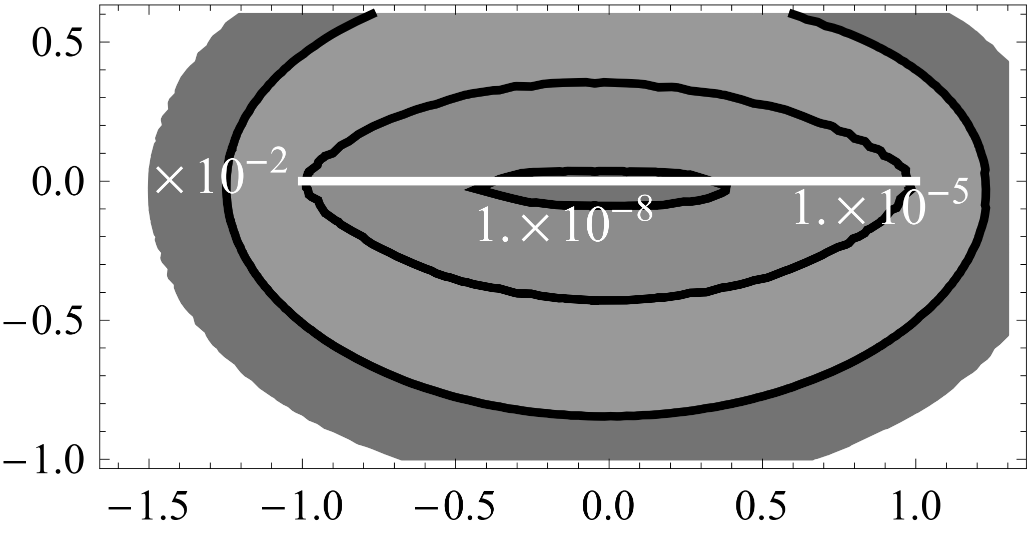

Figure 2: *Error level lines in rectangle* $z = x + y,\ i, x \in [-1.5, 1.3]$, $y \in [-1, .5]$ *when applying the traditional algorithm to function* $f_2(x)$, $-1 \leq x \leq 1$, *with* $n = m = 8$. *Segment* $x \in [-1, 1]$, $y = 0$ *is shown as a white line.*

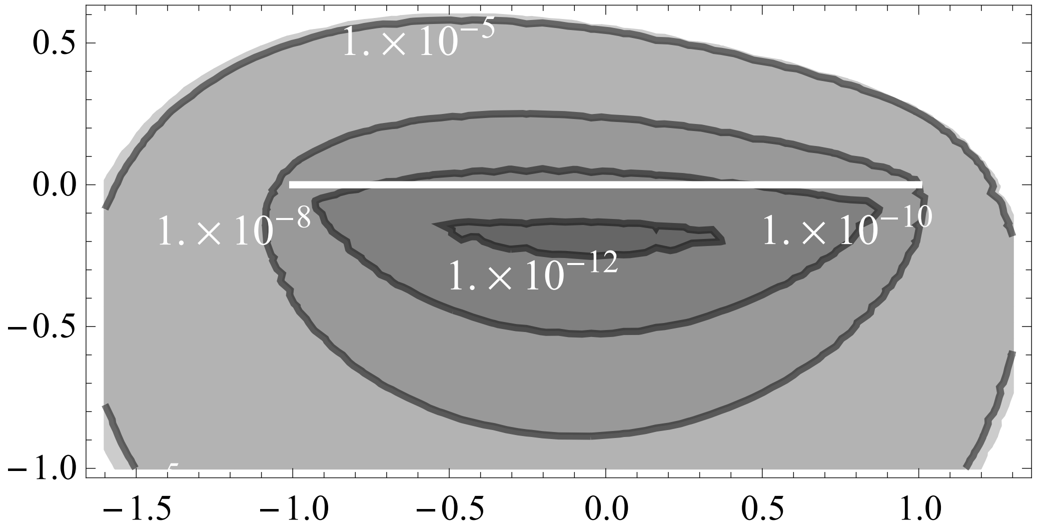

Figure 3: *Error level lines in rectangle $z = x + y$, $i, x \in [-1.5, 1.3]$, $y \in [-1, .5]$ when applying the adaptive algorithm to function $f_2(x)$, $-1 \leq x \leq 1$, with $n = m = 8$. Segment $x \in [-1, 1]$, $y = 0$ is shown as a white line.*

5 Conclusion

In conclusion, we briefly characterize the above results and show that the concept based on the apparatus of a parametric biorthogonal system makes it possible to naturally apply algorithms of traditional and adaptive types not only to the acceleration of convergence for classical Fourier series.

Remark 14 *The universality of our approach was confirmed in [13] in the case of an unusual interpolation on a finite grid. Here, too, a phenomenon of over-convergence was discovered, and the corresponding traditional and adaptive algorithms were proposed. Note that here the algorithm of quasi-periodic interpolation, previously studied in [14], is a particular case of traditional algorithms (see also Remark 12).*

From the practical point of view, these results are especially important since they allow us to effectively approximate a smooth function f on the entire segment [-1,1] by its values on a finite uniform set.

5.1 Summary

In papers [9] and [11], a new approach was proposed to accelerate the convergence of Fourier series based on the use of a parametric biorthogonal system. Along this path, two algorithms, called traditional and adaptive, have been developed. The first of them does not require solving equations for implementation, and the second (more accurate and complex) one is based on the application of super-convergence phenomenon which was discovered in [9] and requires solving nonlinear equation (17).

In this work, we correct and refine these results and propose algorithms that are more flexible and accurate in applications. The above numerical experiment shows the efficiency of approximations (18) and (19) in recovering a smooth function f from its 17-49 Fourier coefficients $\{f_s\}$ in the range $1 \leq m \leq 8$ for the used parameters. If f is a "difficult" function (for example, when it oscillates strongly), it is necessary to increase these values, which will require higher accuracy and more operations.

There is every reason to believe this traditional and adaptive algorithms will be in demand in many areas from engineering calculations that do not require high accuracy and speed to problems in celestial and cosmic mechanics, or tomography.

The usual approach to describing convergence of a functional series is to estimate the rate of decrease of its terms at infinity in one or another metric. The theoretical estimates obtained by us directly characterize the rate of convergence only for the traditional algorithm (see Section 3.5).

The adaptive algorithm is implemented using *Algorithm* $\mathfrak{A}$. Its theoretical basis is the over-convergence phenomenon (see Theorem 2 and Section 4.4.3), which provides for approximation an extremely rich and flexible class of quasi-polynomials. An estimate of convergence in terms of classical classes of smooth functions seems to us doubtful here.

Remark 15 *In studies involving the application of Padé approximants to the asymptotic formulas for the Fourier coefficients $\{f_k\}$ (see Section 1.2 above), some attempts were made to relate the numerical manifestations of the over-convergence phenomenon to the distribution of jumps $\{A_{1,k}\}$ of the function f (see (1)). In particular, the interesting estimates obtained in Theorems 5–7 of [7] showed that such connections are extremely complex and practically not informative.*

5.2 About nearest development and generalizations

The studies published in papers [3-13] as well as the above results make it possible to predict their natural development. Here we note the following directions:

- The case when an algorithm uses both traditional and adaptive methods is not used here. It would be interesting to know the practical properties of such a "hybrid" approach.

- A traditional algorithms for the multidimensional Fourier series were proposed in work [9] (Section 3.2). The implementation of similar adaptive algorithms for classical (and sine-cosine) Fourier series and interpolations in rectangular domains now seems quite possible.

- The same can be said about the problem of implementing the corresponding algorithms for a piecewise-smooth function f (see Section 1.1 and the work [18] mentioned above) for a given finite set of jump points. Here it is not difficult now to construct the corresponding parametric biorthogonal system. Of course, the complexity of the algorithms will increase. Previously, in [19], one similar simple problem was solved using the Padé approximant (see Section 1.2).

- Given what was said at the end of Section 4.4.2, it would be interesting to return to the approach of works [2–9] based on solving equation (3) and using the Padé approximant. There are now fast and stable methods for the inversion of the Vandermonde matrix (see, for example, [20]), and a numerical comparison can be made with the method of this paper. In addition, it is not yet clear whether these two technically different approaches are theoretically equivalent.

- It would be interesting to learn about the applicability of the above approaches to the Sturm-Liouville problem on the entire axis and semiaxis when their spectra is discrete.

5.3 On boundary value problems for ODE

The most promising is the application of the approaches of this work in the general theory of boundary value problems for ODE in a finite segment. First of all, this refers to the Sturm-Liouville problem where the implementation is the simplest.

Consider, for example, the following boundary value problem

$$L\,y = \lambda\,y, \quad L\,y = \sum_{k=0}^{n} p_k(x)\,y^{(k)}(x), \quad \lambda \in \mathbb{C},\ x \in [-1,1] \tag{35}$$

where $n \geq 2$, $q \geq 1$, $p_n(x) = 1$, $p_k(x) \in C^{k+n(q-1)}[-1,1]$, with regular boundary conditions $\{U_\mu = 0\}$, $\mu = 1, 2, \ldots, n$ (for details, see monograph [21]).

Let us outline a scheme for constructing the corresponding algorithms for accelerating the convergence of expansions by eigenfunctions of this problem. Let $\{\nu_k\}$ be the numbered system of all eigenvalues of this problem and $\{\phi_k(x), \psi_k(x)\}$ be the corresponding (generally speaking, biorthogonal) normalized system of eigenfunctions.

The analogue of sequence (24) here has the form

$$t_{r,s} \overset{def}{=} \frac{\psi_r(1)}{\psi_s(1)} \Big(\prod_{\substack{p \in D_m \\ p \neq r}} \frac{\nu_s - \nu_p}{\nu_r - \nu_p} \Big) \frac{\mathcal{P}_m(\nu_r)}{\mathcal{P}_m(\nu_s)}, \quad r \in D_n, \tag{36}$$

where D_m is a finite set of m integers. For simplicity, we assume here that the eigenvalues $\{\nu_k\}$ are not multiple and $\psi_k(1) \neq 0$ for all k.

Analogs of quasi-polynomials (see Definition 2) in this case have already been found in [5,8]. The role of function $\exp(i\,\pi\,\lambda\,x)$ is played by the function $G(x,1,\lambda)$, where G is the following Green's function for operator $L - \lambda\,I$

$$G(x,t,\lambda) = \sum_{\forall k} \frac{\phi_k(x)\,\overline{\psi_k(t)}}{\nu_k - \lambda}, \quad -1 \leq x,t \leq 1,\ \lambda \in \mathbb{C} \setminus \{\nu_k\}. \tag{37}$$

In practice, there is no need to use the infinite series in (37) on the right since it is known that Green's function can be represented explicitly using a given fundamental system of solutions of equation (35) (see Chapter I, §3 of [21]).

Let us dwell on the case of a self-adjoint problem for equation (35). The reader can make sure that all the main above stages of building both the traditional and adaptive algorithms are naturally applicable here, including the same type basic formulas as well as the same proof of the existence of the corresponding over-convergence phenomenon.

Acknowledgments. I express my gratitude to two anonymous referees, whose numerous objections and comments have significantly improved the exposition of this work.

References

[1] A.N. Krylov, *About approximate calculations. Lectures given in 1906* (in Russian), St. Petersburg, Typolithography of K. Birkenfeld, 1907.

[2] K.S. Eckhoff, Accurate and efficient reconstruction of discontinuous functions from truncated series expansions, Math. Comp., **61** (1993), no. 204, pp. 745–763. https://doi.org/10.1090/s0025-5718-1993-1195430-1

[3] A.B. Nersesyan, Quasi-polynomials of Bernoulli type and acceleration of convergence of Fourier series (in Russian). Reports of NAS RA, **104** (2004), no. 5, pp.280–286.

[4] A.B. Nersesyan, Acceleration of convergence of Fourier-Bessel series for piecewise smooth functions (in Russian). Reports of NAS RA, **105** (2005), no. 1, pp. 28–35.

[5] A.B. Nersesyan, Convergence acceleration of eigenfuncton expansion for one-dinentional bondary value problems (in Russian). Mathenatics in Higher School, "Tchartaraget", Yerevan, **1** (2005), no. 2, pp. 47–63.

[6] A.B. Nersesyan and A.V. Poghosyan, Asymptotic estimates for a nonlinear acceleration method for Fourier series (in Russian). Reports of NAS RA, **105** (2005), no. 4, pp. 309–316.

[7] R.H. Barkhudaryan and A.V. Poghosyan, On the convergence of some quasipolynomial approximation (in Russian). Reports of NAS RA, **107** (2007), no. 1, pp.13–19.

[8] A.B.Nersesyan, Acceleration of convergence of eigenfuncton expansions (in Russian). Reports of NAS RA, **107** (2007), no. 2, pp. 124–131.

[9] A. Nersessian, On an over-convergence phenomenon for Fourier series. Basic approach. Armen. J. Math., **10** (2018), no. 9, pp. 1–22. https://doi.org/10.52737/18291163-2018.10.9-1-22

[10] A. Nersessian, A correction to the article "On an over-convergence phenomenon for Fourier series. Basic approach". Armen. J. Math., **11** (2019), no. 2, pp. 1–2. https://doi.org/10.52737/18291163-2019.11.2-1-2

[11] A. Nersessian, Fourier tools are much more powerful than commonly thought. Lobachevskii J. Math., **40** (2019), no. 8, pp. 1122–1131. https://doi.org/10.1134/s1995080219080195

[12] A.Nersessian, A fast method for numerical realization of Fourier tools. In *Real Perspective of Fourier Transforms and Current Developments in Superconductivity*, Chapter 3, IntechOpen, 2021. https://doi.org/10.5772/intechopen.94186

[13] A. Nersessian, On some fast implementations of Fourier interpolation. In: Karapetyants A.N., Kravchenko V.V., Liflyand E., Malonek H.R. (Eds.) *Operator Theory and Harmonic Analysis*. OTHA 2020. Springer Proceedings in Mathematics and Statistics, **357** (2021), pp. 463–477. https://doi.org/10.1007/978-3-030-77493-6_27

[14] A. Nersesyan and N. Oganesyan. Quasiperiodic interpolation (in Russian). Reports of NAS RA, **101** (2001), no. 2, pp. 115–121.

[15] A. Zygmund, *Trigonomrtric series*, Vol. II, Cambridge, 1959.

[16] G.A. Korn and T.M. Korn, *Mathematical Handbook*, McGraw-Hill, 1968.

[17] S. Wolfram, *The Mathematica Book*, Wolfram Media, Champaign, IL., 2011.

[18] J. Geer and N.S. Banerjee, Exponentially accurate approximations to piece-wise smooth periodic functions, J. Sci. Comput., **12** (1997), no. 3, pp. 253–287.

[19] A.B. Nersesyan and R.H. Barkhudaryan, Convergence acceleration of decomposition by eigenfunctions of a model boundary value problem with a discontinuous coefficient (in Russian). Reports of NAS RA, **106** (2006), no. 1, pp. 5–12.

[20] A. Eisinberg and G. Fedele, On the inversion of the Vandermonde matrix, Appl. Math. Comput., **174** (2006), no. 2, pp. 1384–1397. https://doi.org/10.1016/j.amc.2005.06.014

[21] M.A.Naimark, *Linear differential operators* (in Russian), Second Edition, Moskow, "Nauka", 1969.

Anry B. Nersessian
Institute of mathematics,
National Academy of Sciences of the Republic of Armenia
Bagramian ave. 24B, 0019 Yerevan, Armenia.
nerses@instmath.sci.am

Please, cite to this paper as published in
Armen. J. Math., V. **14**, N. 14(2022), pp. 1–31
https://doi.org/10.52737/18291163-2022.14.14-1-31