ISSN 1995-0802, Lobachevskii Journal of Mathematics, 2022, Vol. 43, No. 6, pp. 1417–1427. © Pleiades Publishing, Ltd., 2022.

Generalized Discrete Fourier Transform on the Base of Lagrange and Hermite Interpolation Formulas

S. M. Sitnik[1*], O. E. Yaremko[2], and N. N. Yaremko[3**]**

(Submitted by A. B. Muravnik)

[1]Belgorod State National Research University (BelGU), Belgorod, 308015 Russia
[2]Moscow State Technological University, Moscow Machine Instrumental Institute, Moscow, 127055 Russia
[3]National University of Science and Technology, Moscow Institute of Steel and Alloys, Moscow, 119049 Russia

Received January 30, 2022; revised February 22, 2022; accepted March 10, 2022

Abstract—The article offers several generalizations of Discrete Fourier Transform. The theoretical basis for the generalizations is the interpolation formulas of Lagrange and Hermite. It has been established that each polynomial generates its own corresponding Discrete Fourier Transform. The paper proposes an algorithm for constructing new DFT generalizations. It is possible to use the Fast Fourier Transform (FFT) to build new generalizations. The technology for using the FFT-application in the MatLab system is described. We have revealed that the introduction and application of the Discrete Fourier Transform on the base of the interpolation formulas of Lagrange and Hermite allows us to build new generalizations with the necessary properties in practical applications. The authors' approach, as opposed to traditional methods of presentation, has a number of advantages: first, the simplicity and naturalness of the introduction of the Discrete Fourier Transform are achieved, and secondly, it is possible to construct the Generalized Discrete Fourier Transform with specified properties, which is not obvious under the standard approach.

DOI: 10.1134/S1995080222090256

Keywords and phrases: *discrete Fourier transform (DFT), barycentric Lagrange interpolation formula, Hermite interpolation formula, FFT-application.*

1. INTRODUCTION

The Discrete Fourier Transform (DFT) converts the finite sequence of function values on a equidistant nodes into a sequence of complex amplitudes of harmonic constituents. The inverse Discrete Fourier Transform is the partial sum of the Fourier series on a equidistant nodes. The complex-valued amplitudes of the harmonics are the coefficients of the partial sum of the Fourier series. DFT can be interpreted as a representation in the frequency area of the input sequence of signal values. DFT is widely used [4, 6, 9, 17, 20] to perform Fourier analysis in many practical applications. So, for example, DFT is used to effectively solve equations in private derivatives and perform convolution operation, multiplying large whole numbers, coding, filtering the signal analysis [13, 16, 25, 26]. These transforms are also important in transmutation theory [22, 23, 24].

A number of authors note the connection between DFT and the theory of trigonometric interpolation [2, 3, 18, 19, 20]. However, no proposals have been made to put the theory of interpolation as the basis for DFT. In this article, we propose to fill this gap and build a theory of Discrete Fourier Transforms based on the interpolation formulas of Lagrange and Hermite. The developed approach, in contrast to the traditional methods of presentation, makes it easy to introduce the concept of a discrete Fourier transform and opens up the possibility of its natural generalizations with given properties.

*E-mail: sitnik@bsu.edu.ru
**E-mail: yaremki@mail.ru

The main result of the article is in paragraph 3. The Generalizations of the Discrete Fourier Transform (GDFT) are built in 3.1; the most important cases for applications are considered in 3.2-3.4: 3.2 GDFT with new signal values; 3.3 GDFT for union of two series of equidistant nodes; 3.4 GDFT for union of signal values and its derivative values.

Literature review. The publications [8, 10, 11, 12] systematically set out the theoretical foundations and recommendations for the practical application of Fourier Transforms. The features of Discrete Fourier Transforms (DFT) are considered for the function of discrete argument posed by the numerical array of finite dimension. The Fast Fourier Transforms (FFT) algorithms and the Fourier Transforms standard Libraries are presented. The monograph [10] is devoted to the applications of the DFT for digital signal processing,which is characterized by great attention of the authors to the changing educational needs of the reader. The sources [1, 5, 7, 14, 17, 22–27] set out the theoretical foundations of DFT, FFT and FFT-algorithms and applications to different problems. The Lagrange interpolation formula is written in a baricentric form in the papers [2, 16] and it is mostly convenient for our purpose. The Hermite interpolation formula is explicitly recorded for the case of setting the values of the signal and its derivatives in [15, 18, 19], which seems to be convenient for practical applications.

Purpose and objectives of the study. The purpose of this article is to prove a Generalized Discrete Fourier Transforms with assigned properties based on Lagrange and Hermite Interpolation Formulas. To do this, we need to solve the following tasks:

—to modify the Lagrange and Hermite interpolation formulas;

—to establish a link between the Lagrange interpolation formula and DFT by selecting a base polynomial $p_N(x) = x^N - 1$;

—to develop the GDFT with a equidistant nodes of discrete time points with the ability to add a new signal value;

—to develop the GDFT with the ability to combine two non-intersecting series of equidistant nodes of discrete time points;

—to establish the GDFT on the base of Hermite interpolation formula for signal values and its derivative values on a equidistant nodes.

2. MATERIALS AND METHODS

Our study is based on the Barycentric Lagrange Interpolation Formula, which has the appearance in [2, 3, 16]

$$f(x) = \sum_{k=0}^{N-1} f(\varepsilon_k) \frac{p_N(x)}{p'_N(\varepsilon_k)(x-\varepsilon_k)}, \tag{1}$$

where polynomial $p_N(x)$ of N degree

$$p_N(x) = a_0 + a_1 x + ... + a_{N-1}x^{N-1} + x^N \tag{2}$$

all the roots of which $\{\varepsilon_k\}, k = 0, ..., N-1$ are simple. Formula (1) solves the problem of existing of a polynomial with minimum degree that takes these values in a given set of points. As it is known from [3], there exists the unique polynomial $f(x)$ at most of degree $N-1$ for which $f(\varepsilon_j) = y_j$, if N pairs of points are given

$$(\varepsilon_0, y_0), (\varepsilon_1, y_1), \dots, (\varepsilon_{N-1}, y_{N-1}).$$

Here all ε_j are different; N is the number of signal values measured over a period; $y_0, y_1, ..., y_{N-1}$ measured signal values in discrete time points with numbers $0, 1, ..., N-1$ that are inputs for direct DFT and output for inverse DFT; $\varepsilon_0\varepsilon_1, ..., \varepsilon_{N-1}$ are complex amplitudes of original signal harmonics. such amplitudes are the output for the direct transform and input for the inverse one. For further we need an investigation from the Viet theorem.

Theorem 1. *Let the polynomial of N degree $p_N(x) = a_0 + a_1x + ... + a_{N-1}x^{N-1} + x^N$ has simple roots $\{\varepsilon_k\}, k = 0, ..., N-1$. Then*

$$\frac{p_N(x)}{x-\varepsilon_k} = a_{0k} + a_{1k}x + ... + a_{N-2,k}x^{N-2} + x^{N-1},$$

where

$$a_{N-2,k} = a_{N-1} + \varepsilon_k, \quad a_{N-3,k} = a_{N-2} + a_{N-1}\varepsilon_k + \varepsilon_k^2, ...,$$
$$a_{0k} = a_1 + a_2\varepsilon_k + a_{N-1}\varepsilon_k^{N-2} + \varepsilon_k^{N-1}. \tag{3}$$

Proof. From the condition of the theorem we get

$$p_N(x) = \left(a_{0k} + a_{1k}x + ... + a_{N-2,k}x^{N-2} + x^{N-1}\right)(x - \varepsilon_k).$$

Equating the coefficients at x^l degree, we come to a system of linear equations

$$a_l = -a_{l,k}\varepsilon_k + a_{l-1,k}, \quad l = 1, ..., N-2, \quad a_0 = -a_{0,k}\varepsilon_k.$$

From the last equation we get $a_{0,k} = -\frac{a_0}{\varepsilon_k}$. Transform the found expression. Note that

$$a_0 = -a_1\varepsilon_k - ... - a_{N-1}\varepsilon_k^{N-1} - \varepsilon_k^N.$$

Then we find

$$a_{0k} = a_1 + a_2\varepsilon_k + a_{N-1}\varepsilon_k^{N-2} + \varepsilon_k^{N-1}.$$

From formula (3) we consistently find the values of the other coefficients.

Example 1. If $p_N(x) = x^N - 1$ it is of greatest interest. At the same time, we get

$$x^N - 1 = x^N - \varepsilon_k^N = (x - \varepsilon_k)\left(\varepsilon_k^{N-1} + \varepsilon_k^{N-2}x + \varepsilon_k^{N-3}x^2 + ... + \varepsilon_k x^{N-2} + x^{N-1}\right).$$

Consequently, the coefficients $a_{l,k}$ of the theorem 1 take form $a_{l,k} = \varepsilon_k^{N-1-l}$. Using (3) and taking into account $p'_N(\varepsilon_k) = N\varepsilon_k^{N-1}$ we get the expression of the interpolation polynomial

$$f(x) = \sum_{k=0}^{N-1} f(\varepsilon_k) \sum_{l=0}^{N-1} \frac{\varepsilon_k^{N-1-l}x^l}{N\varepsilon_k^{N-1}} = \frac{1}{N}\sum_{k=0}^{N-1} f(\varepsilon_k) \sum_{l=0}^{N-1} \bar{\varepsilon}_k^l x^l. \tag{4}$$

Changing the summation order on the right side of (4) we get a formula for an interpolation polynomial

$$f(x) = \frac{1}{N}\sum_{l=0}^{N-1} x^l \sum_{k=0}^{N-1} \bar{\varepsilon}_k^l f(\varepsilon_k).$$

If we put $x = \varepsilon_j$, we come to the formula of decomposition of the signal by the base $\varepsilon_j^0, \varepsilon_j^1, ..., \varepsilon_j^{N-1}$in the form

$$f(\varepsilon_j) = \frac{1}{N}\sum_{l=0}^{N-1} \varepsilon_j^l \sum_{k=0}^{N-1} \bar{\varepsilon}_k^l f(\varepsilon_k). \tag{5}$$

Definition 1. The discrete Fourier transform (DFT) of measured signal values

$$y_0 = f(\varepsilon_0), \quad y_1 = f(\varepsilon_1), ..., y_{N-1} = f(\varepsilon_{N-1})$$

is defined by the formula

$$Y_l = \sum_{k=0}^{N-1} \bar{\varepsilon}_k^l y_k \tag{6}$$

From (5) we get the formula of calculation for Discrete Fourier Transform

$$y_j = \frac{1}{N}\sum_{l=0}^{N-1} \varepsilon_j^l Y_l. \tag{7}$$

We rewrite formulas (6), (7) as usual. We use the roots $\varepsilon_k = e^{\frac{2\pi i}{N}k}, k = 0, 1, ..., N-1$ of polynomial $p_N(x) = x^N - 1$. Thus, the last formulas are transformed into formulas for DFT from [17]:

direct

$$Y_l = \sum_{k=0}^{N-1} e^{-\frac{2\pi i}{N} lk} y_k \tag{8}$$

inverse

$$y_j = \frac{1}{N} \sum_{l=0}^{N-1} e^{\frac{2\pi i}{N} lj} Y_l. \tag{9}$$

We have presented a way to introduce a direct and inverse DFT Discrete Fourier Transform based on the interpolation Lagrange formula.

3. MAIN RESULTS

3.1. The Generalizations of Discrete Fourier Transform (GDFT)

Use the Barycentric Lagrange Interpolation formula and the Hermite formula to generalize discrete Fourier transform.

Definition 2. The direct generalized discrete Fourier transform (GDFT) of measured signal values $y_0, y_1, ..., y_{N-1}$ will be determined by formula

$$Y_l = \sum_{k=0}^{N-1} \frac{a_{lk}}{p'_N(\varepsilon_k)} y_k, \quad l = 0, 1, ..., N-1; \quad y_k = f(\varepsilon_k), \tag{10}$$

the polynomial $p_N(x)$ is given by formula (2) and satisfies the conditions of Theorem 1.

Theorem 2. *If the sequence of numbers a_{lk} is set by (3), then the inverse Generalized Discrete Fourier Transform (GDFT) has the form*

$$y_j = \sum_{l=0}^{N-1} \varepsilon_j^l Y_l. \tag{11}$$

Proof. Based on the interpolation formula (1) and Theorem 1, we have

$$f(x) = \sum_{k=0}^{N-1} \frac{f(\varepsilon_k)}{p'_N(\varepsilon_k)} \sum_{l=0}^{N-1} a_{lk} x^l.$$

Put $x = \varepsilon_j$. Then the equality is done

$$y_j = f(\varepsilon_j) = \sum_{k=0}^{N-1} \frac{y_k}{p'_N(\varepsilon_k)} \sum_{l=0}^{N-1} a_{lk} \varepsilon_j^l.$$

Let's change the order of summing. We get

$$y_j = \sum_{l=0}^{N-1} \varepsilon_j^l \sum_{k=0}^{N-1} a_{lk} \frac{y_k}{p'_N(\varepsilon_k)} = \sum_{l=0}^{N-1} \varepsilon_j^l Y_l.$$

The theorem is proved.

Example 2. Let the polynomial $p_N(x) = x^N - 1$ be given. It is proved in paragraph 2 that DFT (8), (9) is a special case of GDFT, corresponding to the equidistant separation of the segment $[0, 2\pi]$ into N equal parts.

3.2. GDFT with New Signal Values

Consider the important case of adding a new signal value to an equidistant separation of the segment $[0, 2\pi]$ into N equal parts. To be more precise, we add one new value $y = f(\varepsilon)$ to the N measured signal values

$$y_0 = f(\varepsilon_0), \quad y_1 = f(\varepsilon_1), ..., y_{N-1} = f(\varepsilon_{N-1}).$$

We choose the polynomial $p(x)$ from theorem 1 in the form

$$p(x) = \left(x^N - 1\right)(x - \varepsilon), \quad \varepsilon \neq \varepsilon_k, \quad k = 0, 1, ..., N-1.$$

Then we get

$$\frac{p(x)}{x - \varepsilon_k} = a'_{0k} + a'_{1k}x + ... + a'_{N-2,k}x^{N-2} + x^N. \tag{12}$$

By equating the coefficients in the right and left parts of (12), we get

$$a'_{0k} = -\varepsilon_k^{N-1}\varepsilon, \quad a'_{lk} = \varepsilon_k^{N-l} - \varepsilon_k^{N-l-1}\varepsilon, \quad l = 1, .., N-1; \quad a'_{Nk} = 1. \tag{13}$$

Based on Theorem 1 direct GDFT is determined by the formula

$$\varepsilon_l = \sum_{k=0}^{N-1} \frac{a'_{lk}}{p'(\varepsilon_k)} y_k + \frac{a'_{lN}}{p'(\varepsilon)} y, \quad l = 0, 1, ..., N,$$

where

$$p'(\varepsilon_k) = N\varepsilon_k^{N-1}(\varepsilon_k - \varepsilon), \quad p'(\varepsilon) = \varepsilon^N - 1.$$

Taking into account (13) GDFT is set by formulas:

direct

$$Y_0 = -\sum_{k=0}^{N-1} \frac{\varepsilon}{N(\varepsilon_k - \varepsilon)} y_k - \frac{y}{\varepsilon^N - 1}, \quad Y_l = \sum_{k=0}^{N-1} \frac{\bar{\varepsilon}_k^l}{N} f(\varepsilon_k), \quad l = 1, ..., N-1, \tag{14}$$

$$Y_N = \sum_{k=0}^{N-1} \frac{\varepsilon_k}{N(\varepsilon_k - \varepsilon)} y_k + \frac{1}{(\varepsilon^N - 1)} y,$$

inverse

$$y_j = \sum_{l=0}^{N} \varepsilon_j^l Y_l; \quad y = \sum_{l=0}^{N} \varepsilon^l Y_l. \tag{15}$$

Note. From formulas (14), (15) we conclude that direct GDFT calculation, i.e. the calculation of complex amplitudes $Y_l, l = 1, ..., N-1$ can be performed in the FFT-application from MatLab. It is need to rewrite (5) and also return to the originals in the FFT-application.

$$y_j = \sum_{l=0}^{N-1} \varepsilon_j^l Y_l + \varepsilon_j^N Y_N; \quad y = \sum_{l=0}^{N-1} \varepsilon^l Y_l + \varepsilon^N Y_N.$$

Thus, this formula contains two addends, the first of which can be calculated in the FFT-application. Therefore, the newly introduced GDFT can be calculated in the FFT-application.

3.3. GDFT for Union of Two Series of Equidistant Nodes

Choose a polynomial $p(x) = \left(x^{N_1} - 1\right)\left(x^{N_2} + 1\right)$, where $N_1 < N_2$ are coprime integers. In this case, all zeros of the polynomial $p(x)$ are simple. We denote the zeros of the polynomial $p(x)$ by $\{\varepsilon_{1k}\}$, $k = 0, 1, ..., N_1 - 1$; $\{\varepsilon_{2m}\}$, $m = 0, 1, ..., N_2 - 1$,

$$\varepsilon_{1k} = e^{i\frac{2\pi lj}{N_1}}, \quad \varepsilon_{2k} = e^{i\frac{\pi(1+2j)l}{N_2}}. \tag{16}$$

Let further

$$\begin{aligned} \frac{p(x)}{x - \varepsilon_{1k}} &= a_{0k} + a_{1k}x + ... + a_{N-2,k}x^{N-2} + x^{N-1}, \quad N = N_1 + N_2, \\ \frac{p(x)}{x - \varepsilon_{2m}} &= b_{0k} + b_{1k}x + ... + b_{N-2,k}x^{N-2} + x^{N-1}, \quad N = N_1 + N_2. \end{aligned} \tag{17}$$

The geometric progression formula is applicable to calculate the coefficients

$$\frac{p(x)}{x - \varepsilon_{1k}} = \frac{x^{N_1} - 1}{x - \varepsilon_{1k}}\left(x^{N_2} + 1\right) = \sum_{p=0}^{N_1-1} x^p \varepsilon_{1k}^{N_1-1-p}\left(x^{N_2} + 1\right)$$

$$= \sum_{p=0}^{N_1-1} x^p \varepsilon_{1k}^{N_1-1-p} + \sum_{p=0}^{N_1-1} x^{N_2+p} \varepsilon_{1k}^{N_1-1-p} = \sum_{p=0}^{N_1-1} x^p \varepsilon_{1k}^{N_1-1-p} + \sum_{p=N_2}^{N_1+N_2-1} x^p \varepsilon_{1k}^{N_2-1-p}.$$

Then the formulas for coefficients from (17) take the form

$$a_{pk} = \begin{cases} \varepsilon_{1k}^{N_1-1-p}, & 0 \le p \le N_1 - 1, \\ 0, & N_1 \le p \le N_2 - 1, \\ \varepsilon_{1k}^{N_2-1-p}, & N_2 \le p \le N_1 + N_2 - 1. \end{cases}$$

Similarly, we calculate the b_{pk}. For this we expand the polynomial from (17) in powers of x

$$\frac{p(x)}{x - \varepsilon_{2k}} = \frac{x^{N_2} + 1}{x - \varepsilon_{2k}}\left(x^{N_1} - 1\right) = \sum_{p=0}^{N_2-1} x^p \varepsilon_{2k}^{N_2-1-p}\left(x^{N_1} - 1\right)$$

$$= -\sum_{p=0}^{N_2-1} x^p \varepsilon_{2k}^{N_2-1-p} + \sum_{p=0}^{N_2-1} x^{N_1+p} \varepsilon_{2k}^{N_2-1-p}$$

$$= -\sum_{p=0}^{N_2-1} x^p \varepsilon_{2k}^{N_2-1-p} + \sum_{p=N_1}^{N_1+N_2-1} x^{N_1+p} \varepsilon_{2k}^{N_1+N_2-1-p} = -\sum_{p=0}^{N_1-1} x^p \varepsilon_{2k}^{N_2-1-p}$$

$$+ \sum_{p=N_1}^{N_2-1} x^{N_1+p}\left(\varepsilon_{2k}^{N_1+N_2-1-p} - \varepsilon_{2k}^{N_2-1-p}\right) + \sum_{p=N_2}^{N_1+N_2-1} x^{N_1+p} \varepsilon_{2k}^{N_1+N_2-1-p}.$$

As a result, we get

$$b_{pk} = \begin{cases} -\varepsilon_{2k}^{N_2-1-p}, & 0 \le p \le N_1 - 1, \\ \left(\varepsilon_{2k}^{N_1+N_2-1-p} - \varepsilon_{2k}^{N_2-1-p}\right), & N_1 \le p \le N_2 - 1, \\ \varepsilon_{2k}^{N_1+N_2-1-p}, & N_2 \le p \le N_1 + N_2 - 1. \end{cases}$$

In the example the Barycentric Lagrange Interpolation formula has the form

$$f(x) = \sum_{k=1}^{N_1-1} y_{1k} \frac{p(x)}{p'(\varepsilon_{1k})(x - \varepsilon_{1k})} + \sum_{k=1}^{N_2-1} y_{2k} \frac{p(x)}{p'(\varepsilon_{2k})(x - \varepsilon_{2k})}, \tag{18}$$

where $y_{1k} = f(\varepsilon_{1k}), y_{2k} = f(\varepsilon_{2k})$. The calculations show that

$$p'_N(\varepsilon_{1l}) = N_1 \varepsilon_{1l}^{N_1-1}\left(\varepsilon_{1l}^{N_2}+1\right), p'_N(\varepsilon_{2l}) = N_2 \varepsilon_{2l}^{N_2-1}\left(\varepsilon_{2l}^{N_1}-1\right). \quad (19)$$

Similarly to paragraph 3.1 from (18) we get the formulas for GDFT:
direct

$$Y_l = \sum_{k=1}^{N_1-1} \frac{\bar{\varepsilon}_{1k}^l}{N_1\left(\varepsilon_{1k}^{N_2}+1\right)} y_{1k} - \sum_{k=1}^{N_2-1} \frac{\bar{\varepsilon}_{2k}^l}{N_2\left(\varepsilon_{2k}^{N_1}-1\right)} y_{2k}, \quad 0 \le l \le N_1 - 1,$$

$$Y_l = \sum_{k=1}^{N_2-1} \frac{\bar{\varepsilon}_{2k}^l}{N_2} y_{2k}, \quad N_1 \le l \le N_2 - 1,$$

$$Y_l = \sum_{k=1}^{N_1-1} \frac{\bar{\varepsilon}_{1k}^{l-N_2}}{N_1\left(\varepsilon_{1k}^{N_2}+1\right)} y_{1k} + \sum_{k=1}^{N_2-1} \frac{\bar{\varepsilon}_{2k}^{l-N_1}}{N_2\left(\varepsilon_{2k}^{N_1}-1\right)} y_{2k}, \quad N_2 \le l \le N_1 + N_2 - 1. \quad (20)$$

For the inverse GDFT (20) we substitute the expressions (16) and convert it to the form

$$y_{1j} = \sum_{l=0}^{N_1+N_2-1} e^{i\frac{2\pi l j}{N_1}} Y_l, y_{2j} = \sum_{l=0}^{N_1+N_2-1} \varepsilon_{2j}^l e^{i\frac{\pi(1+2j)l}{N_2}} Y_l. \quad (21)$$

Note. The decomposition of the signal is made by harmonics set on the union of two series of equidistant nodes

$$\left\{\frac{2\pi i j}{N_1}\right\}, \quad j = 0, 1, ..., N_1 - 1, \quad \left\{\frac{(1+2j)\pi i}{N_2}\right\}, \quad j = 0, 1, ..., N_2 - 1.$$

At the same time, the signal values y_{1j} are restored on the first equidistant nodes, and the values y_{2j} on the second ones.

3.4. GDFT for Union of Signal Values and its Derivative Values

The Hermite formula [15] solves the problem of existing the polynomial of minimal degree that takes the values in a given set of points, the derivative of which takes certain values on the same set of points, cf. [15]. As it is known, for $2N$ pairs of numbers

$$(\varepsilon_0,\ y_0),\ (\varepsilon_1,\ y_1), \dots,\ (\varepsilon_{N-1},\ y_{N-1}),$$
$$(\varepsilon_0,\ y'_0),\ (\varepsilon_1,\ y'_1), \dots,\ (\varepsilon_{N-1},\ y'_{N-1}),$$

where all ε_j are different, there is a unique polynomial $f(x)$ of degree no more than $2N-1$ for which $f(\varepsilon_j) = y_j, f'(\varepsilon_j) = y'_j$. Here are the following designations: $2N$ the number of signal values and values of its derivatives measured over the period $x_n, n = 0, ..., N-1$

$$y_0, y_1, ..., y_{N-1},$$
$$y'_0, y'_1, ..., y'_{N-1},$$

which are input for direct transform and output for the inverse one. In the case of roots of multiplicity 2, it is necessary to use the Hermite interpolation formula. The standard Hermite formula contains divided difference [19] and therefore is not suitable for our purposes. Let's present a more convenient version of the formula.

Theorem 3. *The Interpolation Hermite polynomial has the form*

$$f(x) = \sum_{i=0}^{N-1} \frac{2p_{2N}(x)}{p''_{2N}(\varepsilon_i)(x-\varepsilon_i)^2}\left(y_i\left(1 - \frac{p'''_{2N}(\varepsilon_i)}{3p''_{2N}(\varepsilon_i)}(x-\varepsilon_i)\right) + y'_i(x-\varepsilon_i)\right). \quad (22)$$

Proof. Note that if $x=\varepsilon_j$ all the addends in the right part of the formula are zero except the one with the j number

$$f(\varepsilon_j)=2\lim_{x\to\varepsilon_j}\frac{p_{2N}(x)}{p''_{2N}(\varepsilon_j)(x-\varepsilon_j)^2}\left(y_j\left(1-\frac{p'''_{2N}(\varepsilon_j)}{3p''_{2N}(\varepsilon_j)}(x-\varepsilon_j)\right)+y'_j(x-\varepsilon_j)\right)=y_j.$$

Similarly, it is proven that $f'(\varepsilon_j)=y'_j$. Consider the polynomial $p_{2N}(x)=\left(x^N-1\right)^2$. Let ε_i be the roots of polynomial $p_{2N}(x)$. Each of the roots has a multiplicity 2. Then we get

$$p''_{2N}(\varepsilon_i)=2N^2\varepsilon_i^{2N-2},\quad p'''_{2N}(\varepsilon_i)=6N^2(N-1)\varepsilon_i^{2N-3}.$$

Thus, Hermite interpolation formula takes the form

$$f(x)=2\sum_{i=0}^{N-1}\frac{p_{2N}(x)}{2N^2\varepsilon_i^{2N-2}(x-\varepsilon_i)^2}\left(y_i\left(1-\frac{N^2(N-1)\varepsilon_i^{2N-3}}{N^2\varepsilon_i^{2N-2}}(x-\varepsilon_i)\right)+y'_i(x-\varepsilon_i)\right).$$

After simplifications we get

$$f(x)=\frac{1}{N^2}\sum_{i=0}^{N-1}\frac{\varepsilon_i^2p_{2N}(x)}{(x-\varepsilon_i)^2}\left(y_iN-y'_i\varepsilon_i\right)+x\left(y'_i-(N-1)\varepsilon_i^{-1}y_i\right).$$

Taking into account the elementary identity

$$\varepsilon_k^2\left(\frac{x^N-1}{x-\varepsilon_k}\right)^2=\sum_{l=0}^{2N-2}(N-|N-l-1|)\,x^l\bar{\varepsilon}^l.$$

We get the decomposition of the interpolation polynomial $f(x)$ by degrees of x

$$f(x)=\sum_{i=0}^{N-1}\frac{\left(y'_i-(N-1)\varepsilon_i^{-1}y_i\right)}{N^2}\left(\sum_{k=0}^{N-1}(k+1)\varepsilon_i^{-k}x^{k+1}+\sum_{k=N}^{2N-2}(2N-1-k)\varepsilon_i^{-k}x^{k+1}\right)$$
$$+\sum_{i=0}^{N-1}\frac{\left(y_iN-y'_i\varepsilon_i\right)}{N^2}\left(\sum_{k=0}^{N-1}(k+1)\varepsilon_i^{-k}x^k+\sum_{k=N}^{2N-2}(2N-1-k)\varepsilon_i^{-k}x^k\right).$$

Accept the designations

$$F_k=\sum_{i=0}^{N-1}\varepsilon_i^{-k}\left(y_iN-y'_i\varepsilon_i\right),\quad G_k=\sum_{i=0}^{N-1}\varepsilon_i^{-k}\left(y'_i-(N-1)\varepsilon_i^{-1}y_i\right). \tag{23}$$

Then

$$f(x)=\frac{1}{N^2}\sum_{k=0}^{N-1}(k+1)(F_k+xG_k)x^k+\frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)(F_k+xG_k)x^k, \tag{24}$$

$$f'(x)=\frac{1}{N^2}\sum_{k=0}^{N-1}(k+1)G_kx^k+\frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)G_kx^k$$
$$+\frac{1}{N^2}\sum_{k=1}^{N-1}(k+1)(F_k+xG_k)kx^{k-1}+\frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)(F_k+xG_k)kx^{k-1}.$$

Put in each of the formulas (24) $x=\varepsilon_j$, we get the expressions

$$y_j=\frac{1}{N^2}\sum_{k=0}^{N-1}(k+1)(F_k+\varepsilon_jG_k)\varepsilon_j^k+\frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)(F_k+\varepsilon_jG_k)\varepsilon_j^k.$$

Similarly, for the values of the derivative, we receive accordingly

$$y'_j = \frac{1}{N^2}\sum_{k=0}^{N-1}(k+1)\,G_k\varepsilon_j^k + \frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)\,G_k\varepsilon_j^k$$

$$+\frac{1}{N^2}\sum_{k=1}^{N-1}(k+1)\,(F_k+\varepsilon_j G_k)\,k\varepsilon_j^{k-1} + \frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)\,(F_k+\varepsilon_j G_k)\,k\varepsilon_j^{k-1}.$$

Because of the equality $\varepsilon_k^N = 1$, the periodic condition is fulfilled $F_{k+N} = F_k, G_{k+N} = G_k$. Consequently, the signal decomposition formula into harmonic signals allows simplification

$$y_j = \frac{1}{N^2}\sum_{k=0}^{N-1}(k+1)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k + \frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k$$

$$= \frac{1}{N^2}\sum_{k=0}^{N-2}(k+1)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k + \frac{(F_{N-1}+\varepsilon_j G_{N-1})\,\varepsilon_j^{-1}}{N}$$

$$+\frac{1}{N^2}\sum_{k=N}^{2N-2}(2N-1-k)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k = \frac{1}{N^2}\sum_{k=0}^{N-2}(k+1)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k$$

$$+\frac{1}{N^2}\sum_{k=0}^{N-2}(2N-1-N-k)\,(F_k+\varepsilon_j G_k)\,\varepsilon_j^k + \frac{(F_{N-1}+\varepsilon_j G_{N-1})\,\varepsilon_j^{N-1}}{N}$$

$$= \frac{N}{N^2}\sum_{k=0}^{N-2}(F_k+\varepsilon_j G_k)\,\varepsilon_j^k + \frac{(F_{N-1}+\varepsilon_j G_{N-1})\,\varepsilon_j^{N-1}}{N} = \frac{1}{N}\sum_{k=0}^{N-1}(F_k+\varepsilon_j G_k)\,\varepsilon_j^k.$$

Similarly, formula for the decomposition of the derivative into a sum of harmonic signals can be obtained

$$y'_j = \sum_{k=0}^{N-1} G_k\varepsilon_j^k + \frac{N-1}{N}\sum_{k=0}^{N-1}F_k\varepsilon_j^{k-1}.$$

Definition 3. Two sets of complex amplitude F_k, G_k defined by formulas

$$F_k = \sum_{i=0}^{N-1}\varepsilon_i^{-k}\left(y'_i N - y_i\varepsilon_i^{-1}\right), \tag{25}$$

$$G_k = \sum_{i=0}^{N-1}\varepsilon_i^{-k}\left(y'_i - (N-1)\,\varepsilon_i^{-1}y_i\right), k = 0, 1, ..., N-1$$

is named the *direct GDFT* for signal values and their derivatives

$$y_0, y_1, ..., y_{N-1},$$
$$y'_0, y'_1, ..., y'_{N-1}.$$

As above, the inverse GDFT on sets of complex amplitudes F_k, G_k restores the values of the signal and the values of its derivatives

$$y_j = \frac{1}{N}\sum_{k=0}^{N-1}(F_k+\varepsilon_j G_k)\,\varepsilon_j^k, \tag{26}$$

$$y'_j = \sum_{k=0}^{N-1} G_k\varepsilon_j^k + \frac{N-1}{N}\sum_{k=0}^{N-1}F_k\varepsilon_j^{k-1}.$$

Note. The presence in formulas (25), (26) not only the values of the signal, but also the values of its derivative increases the quality of signal processing.

4. RESULTS OF TESTING THE PROPOSED APPROACHES IN PRACTICE

The calculation of complex amplitudes of $Y_l, l = 0, 1, ..., N-1$ harmonic signals can be performed in the fft-application from MatLab [9]. You can also restore the signal by its complex amplitudes in the fft-application. Program order $Y = \text{fft}(X)$ calculates the DFT of the observed signal values X using the Fast Fourier Transform Algorithm (FFT) [4, 6, 9, 11]. Note that the newly introduced GDFT can be implemented in the fft-application. If you add one new value (see p.3.2) you can implement the calculation algorithm in the fft-application from MatLab. If it is necessary to combine two series of equidistant nodes (see p. 3.3 formulas (19),(20)) FFT algorithm can be easily transformed.

Description the calculation of direct and inverse GDFT's, associated with the Hermite formula, is possible to produce by Fast Fourier Transform Algorithm in the fft and ifft applications from MatLab according to the following:

first, we define the complex amplitudes of the observed signal $Y = \text{fft}(\frac{y}{\varepsilon}, n)$ and the complex amplitudes of the observed signal derivatives $Y' = \text{fft}(y', n)$, where y, y' are vectors signal and its derivatives,

second, we get two vectors of complex amplitudes by formulas $F = NY' - Y, G = Y' - (N-1)Y$,

thirdly, we will determine of the harmonic components of the signal and its derivatives is carried out by the formulas (26) in the form

$y = \text{ifft}(\frac{1}{N}(F + \varepsilon G), n), \quad y' = \text{ifft}(\frac{1}{N}(G + (N-1)\bar{\varepsilon}F), n).$

5. CONCLUSIONS AND FUTURE STEPS

The article summarizes and generates the Discrete Fourier Transforms, based on the Lagrange and Hermite interpolation formulas; signal analysis apparatus was improved. New options for the considered generalizations of Discrete Fourier Transforms were appeared:

—the ability to add new signal values without significantly changing the FFT algorithm

—the possibility of combining two independent samples of signal observations,

—the ability to add signal derivatives.

Based on the presented results, you can develop algorithms for filtering, coding signals by modifying well-known ones. As a result, the quality of signal processing is improved.

To solve the problem it is necessary

1. Explore properties new GDFT.
2. Create an algorithm to add some new more than one signal values to process GDFT.
3. Create new GDFT based on generalized interpolation Hermite formula to process of signal values and its derivatives to the second order.

REFERENCES

1. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (Dover, New York, 1972).
2. J.-P. Berrut and A. Welscher, "Fourier and barycentric formulae for equidistant Hermite trigonometric interpolation," Appl. Comput. Harmon. Anal. **23**, 307–320 (2007).
3. J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange Interpolation," SIAM Rev. **46**, 501–517 (2000).
4. R. E. Blahut, *Fast Algorithms for Digital Signal Processing* (Addison-Wesley, Reading, MA, 1985).
5. R. N. Bracewell, *The Fourier Transform and its Applications* (McGraw-Hill, Reading, 1986).
6. E. O. Brigham, *The Fast Fourier Transform and its Applications* (Prentice-Hall, Englewood Cliffs, 1988).
7. C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms* (Wiley, Chichester, NY, 1985).
8. M. Frigo, *Implementing FFTs in Practice, Fast Fourier Transforms, Connexions* (Rice University, Houston TX, 2008).
9. M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT" in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing* (1998), Vol. 3, pp. 1381–1384.
10. R. Gonzalez and R. Woods, *Digital Image Processing,* 2nd ed. (Prentice Hall, Upper Saddle River, 2002).
11. M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the history of the fast Fourier transform," IEEE ASSP Mag. **1** (4), 14–21 (1984).

12. S. G. Johnson and M. Frigo, "A modified split-radix FFT with fewer arithmetic operations," IEEE Trans. Signal Proces. **55** (1), 11–119 (2007).
13. N. S. Bakhvalov and M. Yu. Orekhov, "Fast methods of solving Poisson's equation," USSR Comput. Math. Math. Phys. **22** (6), 107–114 (1982).
14. E. Kendall and An. Atkinson, *Introduction to Numerical Analysis,* 2nd ed. (Wiley, NY, 1988).
15. R. Kress, "On general Hermite trigonometric interpolation," Numer. Math. **20**, 125–138 (1972).
16. A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing,* 2nd ed. (Prentice Hall, Upper Saddle River, 1999).
17. T. M. Peters and J. Williams, *The Fourier Transform in Biomedical Engineering. Applied and Numerical Harmonic Analysis* (Birkhäuser, Boston, 1998).
18. K. P. Sahakyan, "Hermite trigonometric interpolation," East J. Approx. **12**, 441–449 (2006).
19. A. Spitzbart, "A generalization of Hermite's interpolation formula," Am. Math. Mon. **67**, 42–46 (1960).
20. G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen, "Extension of Chebfun to periodic functions," SIAM J. Sci. Comput. **37**, 554–573 (2015).
21. A. Zygmund, *Trigonometric Series* (Cambridge Univ. Press, 1988), Vol. 2.
22. V. V. Katrakhov and S. M. Sitnik, "The transmutation method and boundary-value problems for singular elliptic equations," Sovrem. Mat. Fundam. Napravl. **64**, 211–426 (2018).
23. E. L. Shishkina and S. M. Sitnik, *Transmutations, singular and Fractional Differential Equations with Applications to Mathematical Physics* (Elsevier, Amsterdam, 2020).
24. S. M. Sitnik and E. L. Shishkina, *Transmutation Method for Differential Equations with Bessel Operators* (Fizmatlit, Moscow, 2019) [in Russian].
25. S. M. Sitnik, A. S. Timashov, and S. N. Ushakov, "On the correctness of finite-rank approximations by series of shifted Gaussians," Lobachevskii J. Math. **41**, 423–429 (2020).
26. S. M. Sitnik, A. S. Timashov, and S. N. Ushakov, "Signal approximations by shifted Gaussians: A direct approach by finite linear systems," J. Phys.: Conf. Ser. **1479**, 012046 (2020).
27. D. K. Durdiev, E. L. Shishkina, and S. M. Sitnik, "The explicit formula for solution of anomalous diffusion equation in the multi-dimensional space," Lobachevskii J. Math. **42**, 1264–1273 (2021).