Armenian Journal of Mathematics
Volume 6, Number 1, 2014, 1–15

An Economical Method for Compression of Fourier Cosine Transformations

Anry Nersessian

Institute of Mathematics,
National Academy of Sciences of Armenia
Bagramian ave. 24/5, 0019 Yerevan, Armenia
nerses@instmath.sci.am

Abstract

A two-dimentional Krilov-Eckhoff type method for smooth functions is presented. A corresponding compression scheme for DCT is derived. Numerical results are presented and discussed. Some possible applications to Image Processing is suggested.

Key Words: multilevel Fourier series and interpolations, economical methods, DCT, compression, image processing, GPEG type algorithms
Mathematics Subject Classification 2000: 41A58,42A20, 65B99,42A15,94A08

1 Introduction

1.1

It is well known that the reconstruction of $f \in C[-1,1]$ by its finite number Fourier coefficients $\{f_n\}_{n=-N}^{N}$

$$f_n = \frac{1}{2}\int_{-1}^{1} f(x)e^{-i\pi nx}dx \tag{1}$$

or discrete Fourier coefficients $\{\widehat{f}_n\}_{n=-N}^{N}$

$$\widehat{f}_n = \frac{1}{2N+1}\sum_{k=-N}^{N} f(x_k)e^{-i\pi n x_k},\ x_k = \frac{2k}{2N+1} \tag{2}$$

by truncated Fourier series

$$S_N(f,x) = \sum_{n=-N}^{N} f_n e^{i\pi nx} \tag{3}$$

or by trigonometric interpolation

$$I_N(f,x) = \sum_{n=-N}^{N} \widehat{f}_n e^{i\pi nx} \tag{4}$$

is highly effective ([1]) for 2-periodic $f \in C^\infty(R)$. When the approximated function has a point of discontinuity the above mentioned approximations lead to the Gibbs phenomena with poor pointwise and L_2 convergences.

1.2

An efficient approach of convergence acceleration of S_N and I_N by subtracting a polynomial representation of discontinuities (jumps) of the function and its derivatives was suggested by Krylov in 1906, [2]. Lanczos in 1966, [3], independently developed the same but more formal approach. He has introduced a basic system of polynomials $B(k;x)$ playing a central role in the method and pointed out a close relation between the $B(k;x)$ and Bernoulli polynomials. Jones and Hardy in 1970, [4], and Lyness in 1974, [5], considered convergence acceleration of trigonometric interpolation by polynomial subtraction. Since then, this approach is widely considered in the context of Fourier series and trigonometric interpolation ([6]-[12]).

The key problem in the Krylov-Lanczos method is the approximation of exact jump values. Ordinarily, such values are unknown and, in general, only Fourier coefficients or discrete Fourier coefficients of a given function may be specified. Even if arbitrary pointwise values of the function can be calculated, the approximation of jump values via finite differences is not recommended for this purpose ([5]). Similarly, in the case of a uniform grid, finite difference approximations are notoriously unreliable. Moreover, in many applications the Fourier coefficients can be calculated but pointwise values and derivatives are not explicitly available.

As noted in [13], the previously mentioned lack of robust methods for the approximation of jump values was the main reason why the polynomial subtraction technique has not been utilized more extensively. The first attempt towards more robust approach was initiated by Gottlieb and all, [14], by utilizing step functions in the reconstruction of discontinuous functions. The general approach was established by Eckhoff in [13]. It was based on the observation that the Fourier coefficients themselves contain sufficient information to reconstruct the jump values. Hence, such values could be approximated with sufficient accuracy using only the coefficients. The fundamental aspect of Eckhoff's method is the approximation of jumps by solving a linear system of equations. Further investigation of the Eckhoff approximation and interpolation were organized in a series of papers [6], [17]-[19].

More specifically, let f be a piecewise smooth function on $[-1,1]$ with jump points $\{a_k\}$, $-1 = a_0 < a_1 < \cdots < a_{l-1} < 1$, $2 \le l < \infty$. Suppose that $f \in C^{q+1}$, $q \ge 0$ on each segment $[a_k, a_{k+1}]$, $k = 1, \cdots, l-2$, and also on the segments $[-1, a_1]$, $[a_{l-1}, 1]$.

Denote by

$$A_{sk} = f^{(k)}(a_s + 0) - f^{(k)}(a_s - 0),\ k = 0, \cdots, q;\ s = 1, \cdots, l-1,$$

$$A_{0k} = f^{(k)}(-1) - f^{(k)}(1),\ k = 0, \cdots, q$$

the jumps of f and its derivatives at the points $\{a_s\}$.

By means of integration by parts we get for $n \ne 0$

$$f_n = \sum_{s=0}^{l-1} e^{-i\pi n a_s} \sum_{k=0}^{q} \frac{A_{sk}}{2(i\pi n)^{k+1}} + \frac{1}{2(i\pi n)^{q+1}} \int_{-1}^{1} f^{(q+1)}(t) e^{-i\pi n t} dt. \tag{5}$$

Consider now Bernoulli polynomials $\{B_k\}$, $k = 1, 2, \cdots$ with Fourier coefficients $\{B_{k,n}\}$

$$B_{k,n} = \begin{cases} 0, & n = 0 \\ \dfrac{(-1)^{n+1}}{2(i\pi n)^{k+1}}, & n = \pm 1, \pm 2, \ldots \end{cases}$$

On the real line Bernoulli polynomials are considered as 2-periodic piecewise smooth functions with "jump points" $a_k = 2k+1$, $k = 0, \pm 1, \pm 2, \cdots$.

Expansion (5) leads to the representation

$$f(x) = \sum_{n=-\infty}^{\infty} f_n e^{i\pi n x} = U(x) + V(x) \tag{6}$$

where

$$U(x) = -\sum_{s=0}^{l-1} \sum_{k=0}^{q-1} A_{sk} \sum_{n=-\infty}^{\infty} e^{i\pi n (x - a_s + 1)} B_{k,n}$$

is a piecewise polynomial function consisting of "shifted" Bernoulli polynomials and $V \in C^q(\mathbf{R})$. Hence, the sequence

$$S_N(f, x) = U(x) + S_N(V, x) \tag{7}$$

converges to f with the rate $o(N^{-q})$, $N \to \infty$ as the coefficients $\{V_n\}$ of V tend to zero with the rate $o(n^{-q-1})$, $n \to \infty$.

If the position of singularities a_s are known then approximate values of jumps can be extracted from the following system of linear equations

$$f_n = U_n,\ n = n_1, \ldots, n_{\ell q}. \tag{8}$$

1.3

Polynomial subtraction method for bivariate truncated Fourier series

$$S_N(f,x,y) = \sum_{n,m=-N}^{N} f_{nm} e^{i\pi(nx+my)}, \ \ f_{nm} = \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1} f(x,y) e^{-i\pi(nx+my)} dxdy \tag{9}$$

and trigonometric interpolation

$$I_N(f,x,y) = \sum_{n,m=-N}^{N} \widehat{f}_{nm} e^{i\pi(nx+my)}, \tag{10}$$

$$\widehat{f}_{nm} = \frac{1}{(2N+1)^2} \sum_{k,s=-N}^{N} f(x_k, x_s) e^{-i\pi(nx_k+mx_s)} \tag{11}$$

was investigated in [6], [21], [22].

If f is smooth on $[-1,1]\times[-1,1]$ then the application of the polynomial correction method can be performed sequentially for each variable x and y.

Denote

$$f^{(k,s)}(x,y) = \frac{\partial^{k+s} f(x,y)}{\partial^k x \partial^s y},$$

$$a(k,y) = f^{(k,0)}(1,y) - f^{(k,0)}(-1,y), \ \ b(s,x) = f^{(0,s)}(x,1) - f^{(0,s)}(x,-1),$$

$$a_m(k) = \frac{1}{2}\int_{-1}^{1} a(k;y) e^{-i\pi my} dy, \ \ b_n(s) = \frac{1}{2}\int_{-1}^{1} b(s;x) e^{-i\pi nx} dx,$$

$$a^{(s)}(k,y) = f^{(k,s)}(1,y) - f^{(k,s)}(-1,y), \ \ b^{(k)}(s,x) = f^{(k,s)}(x,1) - f^{(k,s)}(x,-1),$$

$$a_m^{(s)}(k) = \frac{1}{2}\int_{-1}^{1} a^{(s)}(k;y) e^{-i\pi my} dy, \ \ b_n^{(k)}(s) = \frac{1}{2}\int_{-1}^{1} b^{(k)}(s;x) e^{-i\pi nx} dx,$$

$$c(k,s) = f^{(k,s)}(1,1) - f^{(k,s)}(1,-1) - f^{(k,s)}(-1,1) + f^{(k,s)}(-1,-1).$$

The following lemma from [21] is crucial for bivariate approximations.

Lemma 1 *Let $f^{(k,s)} \in C(D)$, $k,s = 0,\dots,q-1$ and $f^{(q-1,q-1)} \in AC(D)$. Then the following relations hold:*

$$f_{n,0} = \sum_{k=0}^{q-1} B_n(k) a_0(k) + \frac{1}{4(i\pi n)^q}\int_{-1}^{1}\int_{-1}^{1} f^{(q,0)}(x,y) e^{-i\pi nx} dxdy, \ \ n \neq 0,$$

$$f_{0,m} = \sum_{s=0}^{q-1} B_m(s) b_0(s) + \frac{1}{4(i\pi m)^q}\int_{-1}^{1}\int_{-1}^{1} f^{(0,q)}(x,y) e^{-i\pi my} dxdy, \ \ m \neq 0,$$

$$f_{n,m} = \sum_{k=0}^{q-1} B_n(k) a_m(k) + \sum_{s=0}^{q-1} B_m(s) b_n(s) - \sum_{k,s=0}^{q-1} B_n(k) B_m(s) c(k,s)$$
$$+ \frac{1}{4(i\pi n)^q (i\pi m)^q}\int_{-1}^{1}\int_{-1}^{1} f^{(q,q)}(x,y) e^{-i\pi(nx+my)} dxdy, \ \ n,m \neq 0.$$

In view of Lemma 1, and taking into account

$$f(x,y) = \sum_{n=-\infty}^{\infty}{}' \sum_{m=-\infty}^{\infty}{}' f_{n,m} e^{i\pi(nx+my)} + \sum_{n=-\infty}^{\infty}{}' f_{n,0} e^{i\pi nx} + \sum_{m=-\infty}^{\infty}{}' f_{0,m} e^{i\pi my} + f_{0,0},$$

where primes indicate that zero terms are omitted, we get the main representation

$$f(x,y) = G(x,y) + F(x,y), \tag{12}$$

where

$$G(x,y) = \sum_{k=0}^{q-1} B(k,x)a(k,y) + \sum_{s=0}^{q-1} B(s,y)b(s,x) - \sum_{k,s=0}^{q-1} B(k,x)B(s,y)c(k,s), \tag{13}$$

and F is a relatively smooth function $F^{(k,s)} \in C(R^2)$, $k,s = 0,\dots,q-1$. Future details of the acceleration convergence and the compression see in [21].

Below we present an algorithm which is more convenient for the compression in the case of two-dimensional Fourier Cosine series and corresponding interpolations.

2 Fourier Cosine Transformations

2.1 One-dimensional case

Consider a Fourier series and an interpolation based on the complete orthonormal system in $L_2[0,1]$

$$\{\phi_n(x)\} = \{1\} \bigcup \{2\, cos \pi nx\}, \tag{14}$$

$n = 1,2,\dots,\ x \in [0,1],\ \phi_0(x) \equiv 1.$

This basis is chosen mainly due to its possible practical applications in Image Processing. Note that this system corresponds to the boundary problem $y''(x) = \lambda y(x)$, $y'(0) = y'(1) = 0$.

2.1.1 Furier Cosine Series

The truncated Fourier series for the function $f(x) \in C^2[0,1]$, corresponding to the system (14), has the following form

$$S_N(f;x) = \sum_{n=1}^{N} f_n\, \phi_n(x),\ \ f_n = \int_0^1 f(x)\phi_n(x)dx \tag{15}$$

The cosine-coefficients $\{f_n\}$ have the following asymptotic representation when $n \to \infty$

$$f_n = \int_0^1 f(x)\phi_n(x)\,dx = \frac{2\left((-1)^n f'(1) - f'(0)\right)}{\pi^2 n^2} + r_n,$$

$$r_n = -\frac{2\int_0^1 f''(x)\phi_n(x)\,dx}{\pi^2 n^2} = o(1/n^2) \tag{16}$$

For $f \in C^{2m}[0,1]$, $m \geq 2$, extending the integration by parts for r_n, we obtain an estimation for the asymptotic series f_n by powers of $(1/\pi^2 n^2)$ with accuracy of the order $o(1/n^{2m})$.

We introduce the following polynomials

$$P(x) = \frac{1}{4}(x-1)^2,\ Q(x) = P(1-x) = \frac{1}{4}x^2 \tag{17}$$

It is clear that the corresponding coefficients with respect to the system (14) have the form

$$\begin{aligned} p_0 &= \int_0^1 P(x)\,dx = \frac{1}{3},\ p_n = \int_0^1 P(x)\phi_n(x)\,dx = \frac{1}{\pi^2 n^2}, \\ q_0 &= \int_0^1 Q(x)\,dx = \frac{1}{3},\ q_n(x) = \int_0^1 Q(x)\phi_n(x)\,dx = \frac{(-1)^n}{\pi^2 n^2} \end{aligned} \tag{18}$$

Remark 1. *It is not difficult to see that this coefficients correspond to the main term in (16), and thus, the polynomials P and Q can play the role of Bernoulli polynomials (see the Introduction) in KE- method for Fourier Cosine Transformation (FCT, i.e. cosine-series).*

2.1.2 Discrete Cosine Transforms and Interpolations

Discrete analogue of Fourier cosine-series has four versions (see, for instance [23]). We are interested in the following transformation DCTII and its conversion into DCTIII.

$$\text{DCTII}: \quad \widehat{u}_s = \tfrac{1}{\sqrt{N}} \sum_{r=1}^N u_r \cos\left(\tfrac{\pi}{N}(r-1/2)(s-1)\right),\ s = 1,2,\ldots,N, \tag{19}$$

$$\text{DCTIII}: \quad u_r = \tfrac{1}{\sqrt{N}} \left(\widehat{u}_1 + 2\sum_{s=2}^N \widehat{u}_s \cos\left(\tfrac{\pi}{N}(s-1/2)(r-1)\right)\right),\ r = 1,2,\ldots,N \tag{20}$$

Let the vector $u_r = f((r-1/2)/N)$ of values of the function f over the set $\{(r-1/2)/N\}$, $r = 1,2,\ldots,N$ be given. We call the discrete (DCT)-transformation of those values the vector $\widehat{u}$ with components (19) and denote it by $\widehat{f}_s = \widehat{u}_s$, $s = 1,2,\ldots,N$.

Note that according to (19) the corresponding DCT- interpolation $I_N(f;x)$ of a function f is given by the formula

$$I_N(f;x) = \sum_{s=1}^N \widehat{f}_s\,\phi_s(x),\ x \in [0,1],\ N \geq 1 \tag{21}$$

It is easy to see (for instance, using Wolfram Mathematica [23]) that DCT- transformations of values of polynomials $P(x)$ and $Q(x)$ have the following explicit forms:

$$\widehat{p}_0 = \frac{-1-8N^2}{48\,N^{3/2}},\ \widehat{p}_n = \frac{1}{8\,N^{3/2}}\cot\left(\frac{\pi(n-1)}{2\,N}\right)\csc\left(\frac{\pi(n-1)}{2\,N}\right),\ \widehat{q}_n = (-1)^n\,\widehat{p}_n,\ n > 1. \tag{22}$$

2.2 Two-dimensional case

We set

$$S_N(f;x,y) = \sum_{n,m=1}^{N} f_{nm}\,\phi_n(x)\,\phi_m(y),\ f_{nm} = \int_0^1\int_0^1 f(x,y)\phi_n(x)\,\phi_m(y)dxdy \tag{23}$$

as a partial sum of the expansion of a function $f(x,y) \in C^2([0,1]\times[0,1])$ into the Fourier cosine-series.

An analogue of asymptotic formula (16) for this case can be obtained using the integrating by parts. We will need the analogue of the formula (19) (for two-dimensional DCTII), which has here the following form $(s,p = 1,2,\ldots,N)$

$$\widehat{f}_{sp} = \frac{1}{N}\sum_{r,q=1}^{N} u_{rq}\cos\Big(\frac{\pi}{N}(r-1/2)(s-1)\Big)\cos\Big(\frac{\pi}{N}(q-1/2)(p-1)\Big) \tag{24}$$

where $\{u_{rq}\}$, $r,q = 1,2,\ldots,N$, stands for the values of f on the given cosine set:

$$u_{rq} = f((r-1/2),(s-1/2)),\ r,q = 1,2,\ldots,N \tag{25}$$

The interpolation $I_N(f;x,y)$ for a function $f(x,y)$ is the following

$$I_N(f;x,y) = \sum_{s,p=1}^{N} \widehat{f}_{sp}\,\phi_s(x)\,\phi_p(y),\ \ x,y\in[0,1],\ \ N\ge 1 \tag{26}$$

An analogue of the formula (19) (well known two-dimensional inverse DCTII) for $\{u_{rq}\}$, $r,q = 1,2,\ldots,N$, corresponds to the values of $I_N(f;x,y)$ on the considering discrete set (in the case of (25) they are the exact values of f on the cosine grid).

Remark 2. *It is noteworthy, that the main versions of the KE-method for Fourier series and interpolations (see Introduction and [24]) are equivalent to the replacement of most oscillating terms in the Fourier system by polynomials or other functions, which are not restricted with boundary conditions. In our case, some terms in the system $\{\phi_n(x)\,\phi_m(y)\}$ can be replaced by smooth functions of two variables.*

3 More compression without convergence acceleration

3.1 About compression

The notion "compression" is related to the reduction of the memory to keep certain arrays of data with an acceptable level of accuracy. Let us explain this in our case.

In one-dimensional case from the practical point of view (see Introduction and the item 2.1.1 above) we can omit in the truncated series (15) some coefficients f_n, $\|f_n\| < \epsilon$, where ϵ

is a given number. We will call it (here and in two-dimensional case) **ϵ-level compression**. It is clear, that the number of remained coefficients will be so less as rapidly f_n tends to zero.

The application of the one-dimensional KE-method led to the acceleration of convergence of appropriate series and interpolations for smooth functions by switching to the expansions with coefficients tending to zero faster. We say that **the compression increases** if one applies compression of the same acceptable (in some sense) level ϵ, but the complexity of the corresponding algorithms grows insignificantly.

The same phenomenon is observed in known multi - dimensional versions of KE - method, because they are based on one-dimensional versions with respect to each spatial coordinate (see for instance [6, 21,22]). However, in this case the complexity of appropriate algorithms is essentially growing, and it is natural to find alternative approaches to certain applications. Below we present one of them.

3.2 Our concept

The easiest way to explain the idea of the present work is to refer to the Fourier cosine-series. For this purpose let us compare the formulas (15) and (23). They can be rewritten in terms of corresponding errors RS_N:

$$RS_N(f,x) = f(x) - S_N(f,x) = \sum_{n=N+1}^{\infty} f_n\, \phi_n(x), \tag{27}$$

$$RS_N(f;x,y) = f(x,y) - S_N(f;x,y) = \sum_{n,m>N} f_{nm}\, \phi_n(x)\, \phi_m(y) \tag{28}$$

Taking into account the asymptotic of coefficients f_n , when $n \to \infty$, we conclude that $RS_N(f,x) = O(N^{-2})$, $N \to \infty$ in general case, moreover the main contribution is made by the first terms of the series (27). Therefore, (see Introduction and ***Remark 2***) the application of the KE - method accelerates the convergence and increases the compression simultaneously.

The situation is different in the case of (28). Here, for instance (compare with Lemma 1 in Introduction), when $m = \text{const}$, $N \to \infty$, $f_{nm} = O(N^{-2})$, but when $N, M \to \infty$, $f_{NM} = O((NM)^{-2}$.

Our approach is based on (see ***Remark 2***) the replacement of four members of the system $\{\phi_n(x)\, \phi_m(y)\}$ by the linear combination of the following functions (see (17))

$$\{P(x)\, P(y), P(x)\, P(1-y), P(1-x)\, P(y), P(1-y)\, P(1-y)\}.$$

In this work we change the following functions:

$$\{\phi_N(x)\, \phi_N(y),\ \phi_N(x)\, \phi_{N-1}(y), \phi_{N-1}(x)\, \phi_N(y) \text{ and } \phi_{N-1}(x)\, \phi_{N-1}(y)\}.$$

Such a choice is conditioned by the fact, that the main term of the asymptotic expansion of coefficients is more explicit distinguished. For example, when $f \in C^4$

$$f_{NN} = const \cdot N^{-4} + O(N^{-8}),\ N \to \infty \tag{29}$$

In this way we expect the increasing of compression and at the same time we don't expect the acceleration of convergence.

3.3 Realization of the algorithm

Note that the previous arguments hold also for interpolation (26) (hence, also for two-dimensional DCTII).

Focusing on the discrete case now, note (see (24)) that especially the matrix $\{\widehat{f}_{sp}\}$ is used as a subject of compression in **JPEG algorithm** for Image Processing. By this analogy, sometimes we call **pixels** the elements of considered numerical matrices.

Let us adduce the basic steps of the algorithm for realization of the mentioned concept.

Step 1. Let us consider the linear combination

$$A(x,y) = a_1\, P(x)\, P(y) + a_2\, P(1-x)\, P(y) + a_3\, P(x)\, P(1-y) + a_4\, P(1-y)\, P(1-y), \tag{30}$$

where coefficients should be determined later. Using the formulae in item 2.1.1, we get the following representation for the function $A(x,y)$ (see (22)).

$$\widehat{A}_{rq} = (a_1 + (-1)^r\, a_2 + (-1)^q\, a_3 + (-1)^{r+q}\, a_4)\, \widehat{p}_r\, \widehat{p}_q,\ 2 \le r, q \le N, \tag{31}$$

where

$$\widehat{p}_n = \frac{1}{8\,N^{3/2}} \cot\left(\frac{\pi(n-1)}{2\,N}\right) \csc\left(\frac{\pi(n-1)}{2\,N}\right),\ \widehat{q}_n = (-1)^n\, \widehat{p}_n,\ n \ge 2. \tag{32}$$

Step 2. Let the DCTII-transform $U = \{\widehat{f}_{rq}\}$ is given (see (24)). It is easy to see that the system of four equations

$$\widehat{A}_{rq} = \widehat{f}_{rq},\ \ r = N, N-1,\ \ q = N, N-1 \tag{33}$$

with respect to the vector $a = (a_1,\, a_2,\, a_3,\, a_4)^{tr}$ is uniquely solvable in explicit form $a = D\, M\, u$, where

$u = (\widehat{u}_{NN},\, \widehat{u}_{N(N-1)},\, \widehat{u}_{(N-1)N},\, \widehat{u}_{(N-1)(N-1)})^{tr}$ and

$$D = \begin{pmatrix} (2\widehat{p}_N)^{-2} & 0 & 0 & 0 \\ 0 & (4\widehat{p}_{N-1}\widehat{p}_N)^{-1} & 0 & 0 \\ 0 & 0 & (4\widehat{p}_{N-1}\widehat{p}_N)^{-1} & 0 \\ 0 & 0 & 0 & (2\widehat{p}_{N-1})^{-2} \end{pmatrix},$$

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ (-1)^{-N} & (-1)^{-N} & -(-1)^{-N} & -(-1)^{-N} \\ (-1)^{-N} & -(-1)^{-N} & (-1)^{-N} & -(-1)^{-N} \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

Step 3. After the determination of the vector a we completement $(N-1)\times(N-1)$-dimentional matrix to the $N\times N$-dimensional, using the values $\widehat{p}_1$ from (22). We call the resulting matrix the **correction matrix** C. Let us construct the **remainder matrix** R by the formula

$$R = U - C = \{r_{nm}\},\ 1 \le n, m \le N, \tag{34}$$

Step 4. Make a chosen ϵ- level compression of the remainder matrix R and denote it by $R^\epsilon = \{r^\epsilon_{nm}\}$.

Step 5. Using DCTIII (see (20)) to $\{r^\epsilon_{nm}\}$ we get a matrix $V^\epsilon = \{v^\epsilon_{nm}\}$. At the desired output of the algorithm we obtain the **final compressed matrix** $F^\epsilon = V^\epsilon + C$.

Final output interpolation has the form

$$\tilde{I}_N(f;x,y) = \sum_{n,m=1}^{N} r^\epsilon_{nm}\phi_n(x)\phi_m(y) + A(x,y) \tag{35}$$

Remark 3. *The corresponding algorithm for a truncated Fourier series differs from the interpolation only by the fact that in* ***Step 2.*** $\{f_{nm}\}$ *coefficients are used (see (23)) instead* $\{\widehat{A}_{rq}\}$*, and in the matrix* C *instead of* $\{\widehat{p}_j\}$ *the values of* $\{p_j\}$ *from (18) are used.*

Remark 4. *To find the vector* a*, only four multiplication operations were required. For the remainder matrix* R *it was necessary* N^2 *summations. After the* ϵ*- level compression of the remainder matrix* R*, nonzero pixels of the matrix* R^ϵ *and four coordinates of the vector* a *should be stored in the memory. In comparison with the DCTII algorithm (where* ϵ*- level compression is directly applied to the matrix* U*) additional cost is rather cheap.*

4 Numerical Results

4.1 Test Functions

Our test functions are

$$f_1(x,y) = 6.66\, J_{1.33}\left(\frac{1-\sin\left((x-1/3)\,y\right)}{2+x\,y}\right)$$

$$f_2(x,y) = .25\, J_{2.12}\left(3 - \frac{\sin(x+4\,y)+\cos(6\,x-2\,y)}{1.7-x\ (y-1/3)}\right)$$

where J_ν is the Bessel function of the first kind. These functions are positive and their maximal values are equal about 1. We see that $f_1(x, y)$ is very smooth but $f_2(x, y)$ is sharply variable in the directions of both axes.

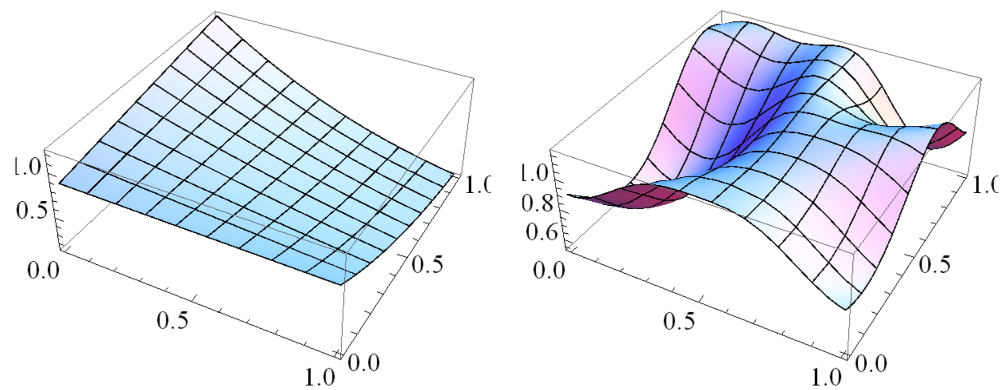

Figure 1: Graphs of test functions f_1 (left) and f_2 (right)

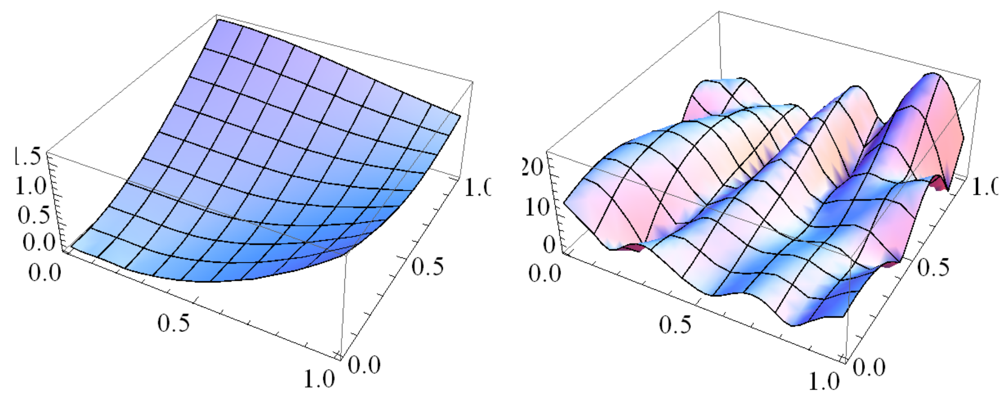

Figure 2: Graphs of $|\nabla f_1|$ (left) and $|\nabla f_2|$ (right)

Remark 5. *It follows from the two-dimensional analogue of (16) that the asymptotic properties of the coefficients* $\widehat{f}_{sp}$ *and* f_{sp} *(compare with Lemma 1 in the Introduction) depend on the behavior of the gradient of the function* $f(x, y)$ *on the boundary.*

Let us present some results of our numerical experiments. Calculations were performed using Wolfram Mathematica 9 for $N = 8, 12, 16$. Below only the algorithm for discrete case was applied for comparison with the DCT-compression algorithm.

The the figures and tables below present the data on comparative compression of DCT and our algorithm called by **KE+DCT**.

4.2 Compression possibilities

Figures below correspond to (see the item 3.1 above) three compression levels $l = (\epsilon_1, \epsilon_2, \epsilon_3)$.

If $f = f_1, l = (2.6 \times 10^{-7}, 2.2 \times 10^{-6}, 5 \times 10^{-3})$ and if $f = f_2, l = (1.9 \times 10^{-7}, 3.1 \times 10^{-6}, 5.1 \times 10^{-3})$ (compare with Tables 1-4 below),

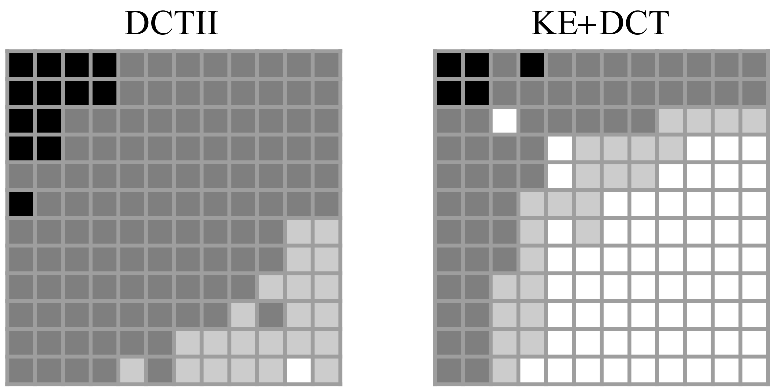

Figure 3: The table of $\{\widehat{f_{sp}}\}$, (ee (24)), for $f = f_1$ (left) and (see Step 3 above) for the corresponding remainder matrix R (right), N=12. Colored pixels correspond: white for $|\widehat{u}_{sp}| \leq 2.6 \times 10^{-7}$, light grey for $2.6 \times 10^{-7}| < |\widehat{u}_{sp}| \leq 2.2 \times 10^{-6}$, dark grey for $2.2 \times 10^{-6} < |\widehat{u}_{sp}| \leq 5 \times 10^{-3}$.

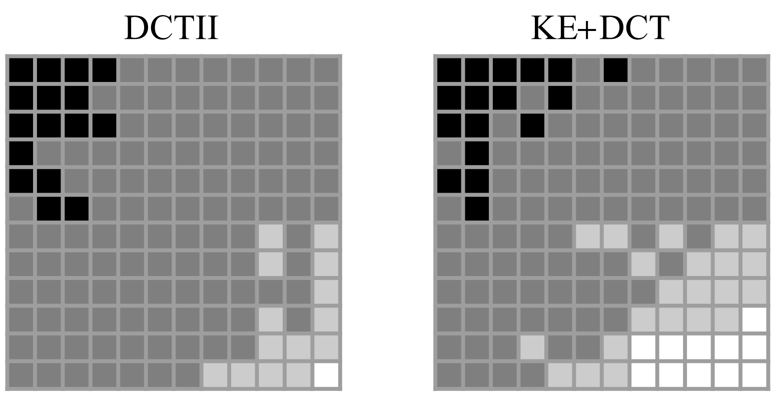

Figure 4: The table of $\{\widehat{f_{sp}}\}$, (ee (24)), for $f = f_2$ (left) and (see Step 3 above) for the corresponding remainder matrix R (right), N=12. Colored pixels correspond: white for $|\widehat{u}_{sp}| \leq 1.9 \times 10^{-7}$, light grey for $1.9 \times 10^{-7}| < |\widehat{u}_{sp}| \leq 3.1 \times 10^{-6}$, dark grey for $3.1 \times 10^{-6} < |\widehat{u}_{sp}| \leq 5.1 \times 10^{-3}$.

The first data of three compression levels equals to the minimum of the values $\{|u_{NN}|, |u_{N(N-1)}|, |u_{(N-1)N}$ (see white pixels at the left pictures in Figures 3 and 4)

Main characteristics for $f_1(h, y)$ and $f_2(x, y)$ are given in the tables 1,2. In the columns "Nonzero pixels" the data for DCT compression algorithm are given in the first row of the

2×3 matrices. In the column "Saved percents" the negative percents mean the nonzero pixels excess.

N	Compression levels	Nonzero pixels	Saved percents
8	$\begin{pmatrix} 3.3\,e{-6} & 3.2\,e{-5} & 1.3\,e{-2} \end{pmatrix}$	$\begin{pmatrix} 54 & 52 & 6 \\ 30 & 29 & 3 \end{pmatrix}$	$\begin{pmatrix} 44.4 & 44.2 & 50 \end{pmatrix}$
12	$\begin{pmatrix} 2.6\,e{-7} & 4\,e{-6} & 5.6\,e{-3} \end{pmatrix}$	$\begin{pmatrix} 128 & 121 & 13 \\ 62 & 56 & 5 \end{pmatrix}$	$\begin{pmatrix} 51.6 & 53.7 & 61.5 \end{pmatrix}$
16	$\begin{pmatrix} 4.5\,e{-8} & 9.5\,e{-7} & 3\,e{-3} \end{pmatrix}$	$\begin{pmatrix} 231 & 217 & 16 \\ 102 & 91 & 7 \end{pmatrix}$	$\begin{pmatrix} 55.8 & 58.1 & 56.3 \end{pmatrix}$

Table 1. *Compression information for* f_1

N	Compression levels	Reminder pixels	Saved percents
8	$\begin{pmatrix} 4\,e{-7} & 5.7\,e{-6} & 6.5\,e{-3} \end{pmatrix}$	$\begin{pmatrix} 64 & 62 & 15 \\ 60 & 59 & 16 \end{pmatrix}$	$\begin{pmatrix} 6.3 & 4.8 & -6.7 \end{pmatrix}$
12	$\begin{pmatrix} 1.9\,e{-7} & 3.1\,e{-6} & 5.1\,e{-3} \end{pmatrix}$	$\begin{pmatrix} 143 & 129 & 16 \\ 133 & 111 & 17 \end{pmatrix}$	$\begin{pmatrix} 7 & 14 & -6.3 \end{pmatrix}$
16	$\begin{pmatrix} 2.7\,e{-8} & 6.2\,e{-7} & 2.6\,e{-3} \end{pmatrix}$	$\begin{pmatrix} 255 & 233 & 24 \\ 224 & 189 & 23 \end{pmatrix}$	$\begin{pmatrix} 12.2 & 18.9 & 4.2 \end{pmatrix}$

Table2. *Compression information for* f_2

4.3 Compression errors

Some final errors after restoration of values of $f_1(x, y)$ and of $f_2(x, y)$ on the two-dimensional cosine grid are presented in the tables 3,4.

Error norms→	L_∞ -errors by compression levels			L_2 -errors by compression levels		
DCT	8.2 e-6	2.1 e-4	7 e-2	4.5 e-6	8.1 e-5	2.8 e-2
KE+DCT	1.4 e-5	1.1 e-4	7.3 e-2	6.7 e-6	4.5 e-5	2.1 e-2

Table 3. *Errors of the algorithms for function* f_1 *on the cosine grid, when* $N = 8$.

Error norms→	L_∞ -errors by compression levels			L_2 -errors by compression levels		
DCT	1 e-8	1.1 e-6	5.4 e–3	5.3 e-9	3.1 e-7	1.4 e-3
KE+DCT	4.4 e-8	1.7 e-6	7.7 e-3	1.1 e-8	3.8 e-7	1.5 e-3

Table 4. *Errors of the algorithms for function* f_2 *on the cosine grid, when* $N = 16$.

Let us call conventionally the **lossless algorithm** the case, when the L_∞- error is less than 0.05 percents of $Max|f|$.

Remark 6. *According to our experiments, the KE+DCT algorithm is best for a lossless compression (see first two compression levels in Tables 3,4 above). It allows to save on average about 20-30 percents of the memory when* $8 \leq N \leq 32$.

5 Conclusions

From the presented and some other our numerical experiments, the following conclusions related to compression problems can be done:

1. For $f \in C^2$, the proposed algorithm is effective when applied to a lossless compression.

2. The smaller the maximum value of the gradient norm $|\nabla f|$ is, the more efficient is the proposed algorithm.

3. The bigger N is, the more efficient is the proposed algorithm. In the above experiment, the values $N = 8, 12, 16$ are chosen so as to make small changes comparing with the classical JPEG-choice, $N = 8$.

4. For $f \notin C^2$ the proposed algorithm makes the compression worse comparing to the DCT algorithm, however the compression errors are practically unchanged.

5. It would be natural to investigate the possibilities of the implementation our KE+DCT lossless algorithm to the compression part of JPEG- algorithm (see ***Remark 6***).

6. It seems interesting to apply the proposed conception to three- and multidimensional Cosine-Fourier series, Cosine-Fourier interpolations and DCTII.

References

[1] A. Zygmund, Trigonometric series. Vol. I, Cambridge Mathematical Library (3rd ed.), Cambridge University Press, 2002.

[2] A. Krylov. On approximate calculations, Lectures delivered in 1906 (in Russian), St. Petersburg 1907, Tipolitography of Birkenfeld.

[3] C. Lanczos, Discourse on Fourier Series, Oliver and Boyd, Edinburgh, 1966.

[4] W. B. Jones, and G. Hardy, *Accelerating convergence of trigonometric approximations*, Math. Comp. **24**(111)(1970), 547–560.

[5] J. N. Lyness, *Computational techniques based on the Lanczos representation*, Math Comp. **28**(1974), 81–123.

[6] B. Adcock, Modified Fourier expansions: theory, construction and applications, PhD thesis, 2010.

[7] G. Baszenski, F.-J. Delvos, and M. Tasche. *A united approach to accelerating trigonometric expansions*, Comput. Math. Appl. **30**(36)(1995), 3349.

[8] D. Batenkov, and Y. Yomdin. *Algebraic Fourier reconstruction of piecewise smooth functions,* Mathematics of Computation, **81**(277)(2012), 277–318.

[9] J. Geer and N. S. Banerjee, *Exponentially accurate approximations to piece-wise smooth periodic functions*, J. Sci. Comput. **2**(3)(1997), 253–287.

[10] A. Nersessian, and A. Poghosyan, *Accelerating the convergence of trigonometric series*, Central European Journal of Math. 3, **4**(2006), 435-448.

[11] A. Nersessian, and A. Poghosyan, *On a rational linear approximation of Fourier series for smooth functions*, J. Sci. Comp., **26**(1) (2006), 111–125.

[12] A. Poghosyan, *Asymptotic behavior of the Krylov-Lanczos interpolation*, Analysis and Applications **7**(2) (2009), 199–211.

[13] K. S. Eckhoff, *Accurate and efficient reconstruction of discontinuous functions from truncated series expansions*, Math. Comp. **61** (1993), 745-763.

[14] D. Gottlieb, L. Lustman, and S.A. Orszag, *Spectral calculations of one dimensional inviscid flows with shocks*, SIAM J. Sci. Stat. Comput. **2**(1981), 296-310.

[15] S. Abarbanel, D. Gottlieb, and E. Tadmor, *Spectral methods for discontinuous problems*, in Numerical Methods for Fluid Dynamics II, ed. by K.W. Morton and M.J. Baines, Oxford University Press, (1986), 128-153.

[16] W. Cai, D. Gottlieb, and C.-W. Shu, *Essentially nonoscillatory spectral Fourier methods for shock wave calculations*, Math. Comp. **52** (1989), 389–410.

[17] A. Barkhudaryan, R. Barkhudaryan, and A. Poghosyan, *Asymptotic behavior of Eckhoff's method for Fourier series convergence acceleration*, Analysis in Theory and Applications **23**(3) (2007), 228-242.

[18] A. Poghosyan, *On an autocorrection phenomenon of the Krylov-Gottlieb-Eckhoff method,* IMA Journal of Numerical Analysis, **31**(2)(2011), 512–527.

[19] A. Poghosyan, *Asymptotic behavior of the Eckhoff's method for convergence acceleration of trigonometric interpolation*, Analysis in Theory and Applications **26**(3)(2010), 236–260.

[20] A.Nersessian and A.Poghosyan, ”Asymptotic errors of accelerated two-dimensional trigonometric approximations”, In: G.A.Barsegian, H.G.W.Begehr, H.G.Ghazaryan, A.Nersessian (Ed.): *Complex Analysis, Differential Equations and Related Topics*, Yerevan, Armenia, September 17-21, 2002, ”Gitutjun” Publishing House, Yerevan, Armenia, 2004, pp. 70-78.

[21] A. Nersessian and A. Poghosyan, *The Convergence Acceleration of Two-Dimensional Fourier Interpolation*. Armenian Journal of Mathematics, vol. 1, no. 1, 2008, pp. 50-63.

[22] A. Poghosyan, *Asymptotic behavior of the Eckhoff approximation in bivariate case*. Analysis in Theory and Applications, **28**(4)(2012), 329–362.

[23] S. Wolfram, The Mathematica Book, fourth edition, Cambridge University Press.

[24] A. Nersessian, *Convergence acceleration of Fourier series revisited.*, Armenian Journal of Mathematics, Volume 3 , Issue 4, 2011, pp. 152-161.