

A Fast Discrete Transform for a Class of Simultaneously Diagonalizable Matrices

Antonio Boccuto[1], Ivan Gerace[1(✉)], and Valentina Giorgetti[2]

[1] Dipartimento di Matematica e Informatica, Via Vanvitelli, 1, 06123 Perugia, Italy
{antonio.boccuto,ivan.gerace}@unipg.it
[2] Dipartimento di Matematica e Informatica, viale G. B. Morgagni, 67/A, 50134 Firenze, Italy
valentina.giorgetti@unifi.it

Abstract. We introduce a new class of simultaneously diagonalizable real matrices, the γ-matrices, which include both symmetric circulant matrices and a subclass of the set of all reverse circulant matrices. We define some algorithms for fast computation of the product between a γ-matrix and a real vector. We proved that the computational cost of a multiplication between a γ-matrix and a real vector is of at most $\frac{7}{4}\, n\, \log_2 n + o(n\, \log_2 n)$ additions and $\frac{1}{2}\, n\, \log_2 n + o(n\, \log_2 n)$ multiplications. Our algorithm can be used to improve the performance of general discrete transforms for multiplications of real vectors.

Keywords: Symmetric circulant matrix · Reverse circulant matrix · Discrete real Fast Fourier transform

1 Introduction

Recently, there have been several studies about simultaneously diagonalizable real matrices. In particular, in investigating the preconditioning of Toeplitz matrices (see, e.g., [9]), several matrix algebras are considered, among which the class of all circulant matrices (see, e.g., [7,12]), the family of all ω-circulant matrices (see, e.g., [10]), the set of all τ-matrices (see, e.g., [2]) and the family of all matrices diagonalized by the Hartley transform (see, e.g., [3]).

In this paper we focus on approximation of real symmetric Toeplitz matrices. Thus, the involved data are always *real*, so that it is not advisable to use the classical complex Discrete Fast Fourier Transform. We investigate a particular class of simultaneously diagonalizable real matrices, whose elements we call γ-*matrices*. Such a class, similarly as that of the matrices diagonalizable by the Hartley transform (see, e.g., [3]) includes symmetric circulant matrices and a subclass of the family of all reverse circulant matrices. Symmetric circulant matrices have several applications to ordinary and partial differential equations (see, e.g., [13]), images and signal restoration (see, e.g., [6]), graph theory (see,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022, LNCS 13375, pp. 214–231, 2022.
https://doi.org/10.1007/978-3-031-10522-7_16

e.g., [8]). Reverse circulant matrices have different applications, for instance in exponential data fitting and signal processing (see, e.g., [1]).

Successively, we deal with the problem of real fast transform algorithms (see, e.g., [11,15]). In particular we define some algorithms for fast computation of the product between a γ-matrix and a real vector and between two γ-matrices. We show that the computational cost of a multiplication between a γ-matrix and a real vector is of at most $\frac{7}{4}\, n \log_2 n + o(n \log_2 n)$ additions and $\frac{1}{2}\, n \log_2 n + o(n \log_2 n)$ multiplications. The proposed algorithm can be used to improve the performance of different discrete transforms given for multiplications of real vectors. In [5] it is shown how the γ-matrices can be suitably used for preconditioning Toeplitz symmetric matrices, which are widely used in several practical problems and applications. Moreover in [4], by means of various experimental results, it is shown how such a matrix class approximates symmetric Toeplitz matrices better than some other classes used in the literature.

The paper is structured as follows. In Sect. 2 we characterize γ-matrices, in Sect. 3 we define the real fast transforms by means of which it is possible to implement the product between a γ-matrix and a real vector.

2 Spectral Characterization of γ-matrices

In order to present a new class of simultaneously diagonalizable matrices, we first define the following matrix. Let n be a fixed positive integer, and $Q_n = (q_{k,j}^{(n)})_{k,j}$, $k, j = 0, 1, \ldots, n-1$, where

$$q_{k,j}^{(n)} = \begin{cases} \alpha_j \cos\left(\dfrac{2\pi\, k\, j}{n}\right) & \text{if } 0 \le j \le \lfloor n/2 \rfloor, \\ \alpha_j \sin\left(\dfrac{2\pi\, k\, (n-j)}{n}\right) & \text{if } \lfloor n/2 \rfloor \le j \le n-1, \end{cases} \tag{1}$$

$$\alpha_j = \begin{cases} \dfrac{1}{\sqrt{n}} = \overline{\alpha} \text{ if } j = 0, \text{ or } j = n/2 \text{ if } n \text{ is even}, \\ \sqrt{\dfrac{2}{n}} = \widetilde{\alpha} \text{ otherwise}, \end{cases} \tag{2}$$

and put

$$Q_n = \left(\mathbf{q}^{(0)} \,\middle|\, \mathbf{q}^{(1)} \,\middle|\, \cdots \,\middle|\, \mathbf{q}^{(\lfloor \frac{n}{2} \rfloor)} \,\middle|\, \mathbf{q}^{(\lfloor \frac{n+1}{2} \rfloor)} \,\middle|\, \cdots \,\middle|\, \mathbf{q}^{(n-2)} \,\middle|\, \mathbf{q}^{(n-1)}\right), \tag{3}$$

where

$$\mathbf{q}^{(0)} = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^T = \frac{1}{\sqrt{n}}\, \mathbf{u}^{(0)}, \tag{4}$$

$$\mathbf{q}^{(j)} = \sqrt{\frac{2}{n}} \left(1 \quad \cos\left(\frac{2\pi j}{n}\right) \cdots \cos\left(\frac{2\pi j(n-1)}{n}\right)\right)^T = \sqrt{\frac{2}{n}}\, \mathbf{u}^{(j)},$$
$$\mathbf{q}^{(n-j)} = \sqrt{\frac{2}{n}} \left(0 \quad \sin\left(\frac{2\pi j}{n}\right) \cdots \sin\left(\frac{2\pi j(n-1)}{n}\right)\right)^T = \sqrt{\frac{2}{n}}\, \mathbf{v}^{(j)}, \tag{5}$$

$j = 1, 2, \ldots, \lfloor \frac{n-1}{2} \rfloor$. Moreover, when n is even, set

$$\mathbf{q}^{(n/2)} = \frac{1}{\sqrt{n}} \Big(1 \; -1 \;\; 1 \; -1 \cdots -1\Big)^T = \frac{1}{\sqrt{n}}\, \mathbf{u}^{(n/2)}. \tag{6}$$

In [14] it is proved that all columns of Q_n are orthonormal, and thus Q_n is an orthonormal matrix.

Now we define the following function. Given $\boldsymbol{\lambda} \in \mathbb{C}^n$, $\boldsymbol{\lambda} = (\lambda_0\, \lambda_1 \cdots \lambda_{n-1})^T$, set

$$\operatorname{diag}(\boldsymbol{\lambda}) = \Lambda = \begin{pmatrix} \lambda_0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & \lambda_1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \lambda_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \lambda_{n-2} & 0 \\ 0 & 0 & 0 & \ldots & 0 & \lambda_{n-1} \end{pmatrix},$$

where $\Lambda \in \mathbb{C}^{n\times n}$ is a diagonal matrix.

A vector $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\boldsymbol{\lambda} = (\lambda_0\, \lambda_1 \cdots \lambda_{n-1})^T$ is said to be *symmetric* (resp., *asymmetric*) iff $\lambda_j = \lambda_{n-j}$ (resp., $\lambda_j = -\lambda_{n-j}$) $\in \mathbb{R}$ for every $j = 0, 1, \ldots, \lfloor n/2 \rfloor$.

Let Q_n be as in (3), and $\mathcal{G}_n$ be the space of the matrices *simultaneously diagonalizable* by Q_n, that is

$$\mathcal{G}_n = \operatorname{sd}(Q_n) = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}),\, \boldsymbol{\lambda} \in \mathbb{R}^n\}.$$

A matrix belonging to $\mathcal{G}_n$, $n \in \mathbb{N}$, is called γ-*matrix*. Moreover, we define the following classes by

$$\mathcal{C}_n = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}),\, \boldsymbol{\lambda} \in \mathbb{R}^n,\, \boldsymbol{\lambda} \text{ is symmetric}\},$$

$$\mathcal{B}_n = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}),\, \boldsymbol{\lambda} \in \mathbb{R}^n,\, \boldsymbol{\lambda} \text{ is asymmetric}\}.$$

Note that the class $\mathcal{C}_n$ corresponds to the set of all circulant symmetric matrices, while $\mathcal{B}_n$ is a particular subclass of the set of the reverse circulant matrices. We have the following.

Theorem 1. (see [5]) *One has*

$$\mathcal{G}_n = \mathcal{C}_n \oplus \mathcal{B}_n, \tag{7}$$

where $\oplus$ *is the orthogonal sum.*

Any matrix belonging to the class $\mathcal{G}_n$ will be called γ-*matrix*.

3 Multiplication Between a γ-matrix and a Real Vector

In this section we deal with the problem of computing the product

$$\mathbf{y} = G\mathbf{x}, \tag{8}$$

where $G \in \mathcal{G}_n$. Since G is a γ-matrix, there is a diagonal matrix $\Lambda^{(G)} \in \mathbb{R}^{n\times n}$ with $G = Q_n \Lambda^{(G)} Q_n^T$, where Q_n is as in (1). To compute

$$\mathbf{y} = Q_n \Lambda^{(G)} Q_n^T \mathbf{x},$$

we proceed by doing the next operations in the following order:

$$\mathbf{z} = Q_n^T \mathbf{x}; \tag{9}$$

$$\mathbf{t} = \Lambda^{(G)} \mathbf{z}; \tag{10}$$

$$\mathbf{y} = Q_n \mathbf{t}. \tag{11}$$

To compute the product in (9), we define a new fast technique, which we will call *inverse discrete sine-cosine transform* (IDSCT), while to do the operation in (11), we define a new technique, which will be called *discrete sine-cosine transform* (DSCT).

From now on, we assume that the involved vectors $\mathbf{x}$ belong to $\mathbb{R}^n$, where $n = 2^r$, $r \geq 2$, and we put $m = n/2$, $\nu = n/4$.

3.1 The IDSCT Technique

In this subsection we present the technique to compute (9). Given $\mathbf{x} \in \mathbb{R}^n$, we denote by

$$\mathrm{IDSCT}(\mathbf{x}) = Q_n^T\,\mathbf{x} = \begin{pmatrix} \alpha_0\,\mathrm{C}_0(\mathbf{x}) \\ \alpha_1\,\mathrm{C}_1(\mathbf{x}) \\ \vdots \\ \alpha_m\,\mathrm{C}_m(\mathbf{x}) \\ \alpha_{m+1}\,\mathrm{S}_{m-1}(\mathbf{x}) \\ \vdots \\ \alpha_{n-1}\,\mathrm{S}_1(\mathbf{x}) \end{pmatrix}, \tag{12}$$

where α_j, $j = 0, 1, \ldots, n-1$, are as in (2),

$$\mathrm{C}_j(\mathbf{x}) = \mathbf{x}^T \mathbf{u}^{(j)}, \quad j = 0, 1, \ldots, m, \tag{13}$$

$$\mathrm{S}_j(\mathbf{x}) = \mathbf{x}^T \mathbf{v}^{(j)}, \quad j = 1, 2, \ldots, m-1, \tag{14}$$

where $\mathbf{u}^{(j)}$ and $\mathbf{v}^{(j)}$ are as in (4)–(6).

For $\mathbf{x} = (x_0\, x_1\, \cdots x_{n-1})^T \in \mathbb{R}^n$, let η (resp., ζ): $\mathbb{R}^n \to \mathbb{R}^m$ be the function which associates to any vector $\mathbf{x} \in \mathbb{R}^n$ the vector consisting of all even (resp., odd) components of $\mathbf{x}$, that is

$$\eta(\mathbf{x}) = \eta\,\mathbf{x} = (x'_0\, x'_1\, \ldots\, x'_{m-1}),\ \text{where } x'_p = x_{2p},\ p = 0, 1, \ldots, m-1;$$
$$\zeta(\mathbf{x}) = \zeta\,\mathbf{x} = (x''_0\, x''_1\, \ldots\, x''_{m-1}),\ \text{where } x''_p = x_{2p+1},\ p = 0, 1, \ldots, m-1.$$

Proposition 1. (see, e.g., [5,12]) *For $k = 0, 1, \ldots, m$, we have*

$$\mathrm{C}_k(\mathbf{x}) = \mathrm{C}_k(\eta\,\mathbf{x}) + \cos\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{C}_k(\zeta\,\mathbf{x}) - \sin\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{S}_k(\zeta\,\mathbf{x}). \tag{15}$$

Moreover, for $k = 1, 2, \ldots, \nu$, it is

$$\mathrm{S}_k(\mathbf{x}) = \mathrm{S}_k(\eta\,\mathbf{x}) + \cos\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{S}_k(\zeta\,\mathbf{x}) + \sin\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{C}_k(\zeta\,\mathbf{x}). \tag{16}$$

Lemma 1. *For $k = 0, 1, \ldots, \nu - 1$, let $t = m - k$. We have*

$$\mathrm{C}_t(\mathbf{x}) = \mathrm{C}_k(\eta\,\mathbf{x}) - \cos\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{C}_k(\zeta\,\mathbf{x}) + \sin\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{S}_k(\zeta\,\mathbf{x}). \tag{17}$$

Furthermore, for $k = 1, 2, \ldots, \nu - 1$ we get

$$\mathrm{S}_t(\mathbf{x}) = \mathrm{S}_k(\eta\,\mathbf{x}) - \cos\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{S}_k(\zeta\,\mathbf{x}) + \sin\Big(\frac{\pi\, k}{m}\Big)\,\mathrm{C}_k(\zeta\,\mathbf{x}). \tag{18}$$

Proof. We begin with (17). For $t = \nu + 1, \nu + 2, \ldots, m$, we have

$$\begin{aligned}
\mathrm{C}_t(\mathbf{x}) &= \sum_{p=0}^{m-1} x_{2p} \cos\Big(\frac{4\pi tp}{n}\Big) + \sum_{p=0}^{m-1} x_{2p+1} \cos\Big(\frac{2\pi t(2p+1)}{n}\Big) \\
&= \sum_{p=0}^{m-1} x'_p \cos\Big(2\pi p - \frac{2\pi kp}{m}\Big) + \sum_{p=0}^{m-1} x''_p \cos\Big(\pi(2p+1) - \frac{\pi k(2p+1)}{m}\Big) \\
&= \sum_{p=0}^{m-1} x'_p \cos\Big(\frac{2\pi kp}{m}\Big) - \cos\Big(\frac{\pi k}{m}\Big) \sum_{p=0}^{m-1} x''_p \cos\Big(\frac{2\pi kp}{m}\Big) \\
&\quad + \sin\Big(\frac{\pi k}{m}\Big) \sum_{p=0}^{m-1} x''_p \sin\Big(\frac{2\pi kp}{m}\Big) \\
&= \mathrm{C}_k(\eta\,\mathbf{x}) - \cos\Big(\frac{\pi k}{m}\Big)\mathrm{C}_k(\zeta\,\mathbf{x}) + \sin\Big(\frac{\pi k}{m}\Big)\mathrm{S}_k(\zeta\,\mathbf{x}).
\end{aligned}$$

Now we turn to (18). For $t = \nu + 1, \nu + 2, \ldots, m - 1$, it is

$$\begin{aligned}
\mathrm{S}_t(\mathbf{x}) &= \sum_{p=0}^{m-1} x_{2p} \sin\Big(\frac{4\pi tp}{n}\Big) + \sum_{p=0}^{m-1} x_{2p+1} \sin\Big(\frac{2\pi t(2p+1)}{n}\Big) \\
&= \sum_{p=0}^{m-1} x'_p \sin\Big(2\pi p - \frac{2\pi kp}{m}\Big) + \sum_{p=0}^{m-1} x''_p \sin\Big(\pi(2p+1) - \frac{\pi k(2p+1)}{m}\Big) \\
&= -\mathrm{S}_k(\eta\,\mathbf{x}) + \sin\Big(\frac{\pi k}{m}\Big)\mathrm{C}_k(\zeta\,\mathbf{x}) + \cos\Big(\frac{\pi k}{m}\Big)\mathrm{S}_k(\zeta\,\mathbf{x}). \qquad \square
\end{aligned}$$

Note that

$$\mathrm{C}_0(\mathbf{x}) = \mathrm{C}_0(\eta\,\mathbf{x}) + \mathrm{C}_0(\zeta\,\mathbf{x}), \tag{19}$$

$$\mathrm{C}_m(\mathbf{x}) = \mathrm{C}_0(\eta\,\mathbf{x}) - \mathrm{C}_0(\zeta\,\mathbf{x}). \tag{20}$$

Since $\mathrm{S}_\nu(\mathbf{x}) = 0$ whenever $\mathbf{x} \in \mathbb{R}^m$, then

$$\mathrm{C}_\nu(\mathbf{x}) = \mathrm{C}_\nu(\eta\,\mathbf{x}) - \mathrm{S}_\nu(\zeta\,\mathbf{x}) = \mathrm{C}_\nu(\eta\,\mathbf{x}) \tag{21}$$

and

$$\mathrm{S}_\nu(\mathbf{x}) = \mathrm{S}_\nu(\eta\,\mathbf{x}) + \mathrm{C}_\nu(\zeta\,\mathbf{x}) = \mathrm{C}_\nu(\zeta\,\mathbf{x}). \tag{22}$$

We call $\rho : \mathbb{R}^n \to \mathbb{R}^m$ that function which reverses all components of a given vector $\mathbf{x} \in \mathbb{R}^n$ but the 0-th component, and σ (resp., α): $\mathbb{R}^n \to \mathbb{R}^n$ that function which associates to every vector $\mathbf{x} \in \mathbb{R}^n$ the double of its symmetric (resp., asymmetric) part, namely

$$\rho(\mathbf{x}) = (x_0\, x_{n-1}\, x_{n-2} \cdots x_2\, x_1)^T,$$
$$\sigma(\mathbf{x}) = \sigma\,\mathbf{x} = \mathbf{x} + \rho(\mathbf{x}), \qquad \alpha(\mathbf{x}) = \alpha\,\mathbf{x} = \mathbf{x} - \rho(\mathbf{x}).$$

Note that

$$\mathbf{x} = \frac{\sigma\,\mathbf{x} + \alpha\,\mathbf{x}}{2}. \tag{23}$$

Now we state the next technical results.

Lemma 2. *Let* $\mathbf{a} = (a_0\, a_1 \cdots a_{n-1})^T$, $\mathbf{b} = (b_0\, b_1 \cdots b_{n-1})^T \in \mathbb{R}^n$ *be such that* $\mathbf{a}$ *is symmetric and* $\mathbf{b}$ *is asymmetric. Then,* $\mathbf{a}^T\,\mathbf{b} = 0$.

Proof. First of all, we observe that $b_0 = b_{n/2} = 0$. So, we have

$$\begin{aligned}\mathbf{a}^T\,\mathbf{b} &= \sum_{j=0}^{n-1} a_j\, b_j = \sum_{j=1}^{n-1} a_j\, b_j = \sum_{j=1}^{n/2-1} a_j\, b_j + a_{n/2}\, b_{n/2} + \sum_{j=n/2+1}^{n} a_j\, b_j \\ &= \sum_{j=1}^{n/2-1} a_j\, b_j + \sum_{j=1}^{n/2-1} a_{n-j}\, b_{n-j} = \sum_{j=1}^{n/2-1} a_j\, b_j - \sum_{j=1}^{n/2-1} a_j\, b_j = 0. \qquad \square\end{aligned}$$

Corollary 1. *Let* $\mathbf{x} \in \mathbb{R}^n$ *be a symmetric vector and* $k \in \{0, 1, \ldots, n-1\}$. *Then, we get*

$$\mathrm{S}_k(\mathbf{x}) = 0. \tag{24}$$

Proof. Note that $\mathrm{S}_k(\mathbf{x}) = \mathbf{x}^T\,\mathbf{v}^{(k)}$. The formula (24) follows from Lemma 2, since $\mathbf{x}$ is symmetric and $\mathbf{v}^{(k)}$ is asymmetric. $\square$

Corollary 2. *Let $\mathbf{x} \in \mathbb{R}^n$ be an asymmetric vector and $k \in \{0, 1, \dots, n-1\}$. Then, we have*

$$\mathrm{C}_k(\mathbf{x}) = 0. \tag{25}$$

Proof. Observe that $\mathrm{C}_k(\mathbf{x}) = \mathbf{x}^T \mathbf{u}^{(k)}$. The formula (25) is a consequence of Lemma 2, because $\mathbf{u}^{(k)}$ is symmetric and $\mathbf{x}$ is asymmetric. □

Lemma 3. *Let $\mathbf{x} \in \mathbb{R}^n$ and $k \in \{0, 1, \dots, m\}$. Then we get*

$$C_k(\sigma\,\mathbf{x}) = 2\,C_k(\mathbf{x}). \tag{26}$$

Moreover, if $k \in \{1, 2, \dots, m-1\}$, then

$$\mathrm{S}_k(\alpha\,\mathbf{x}) = 2\,S_k(\mathbf{x}). \tag{27}$$

Proof. We begin with proving (26). For every $j \in \{0, 1, \dots, n-1\}$ we have

$$\begin{aligned} C_k(\sigma\,\mathbf{x}) &= 2x_0 + \sum_{j=1}^{n-1} x_j u_j^{(k)} + \sum_{j=1}^{n-1} x_{n-j} u_j^{(k)} \\ &= 2x_0 + \sum_{j=1}^{n-1} x_j u_j^{(k)} + 2\sum_{j=1}^{n-1} x_j u_j^{(k)} = 2\sum_{j=0}^{n-1} x_j u_j^{(k)} = 2\,C_k(\mathbf{x}). \end{aligned}$$

Now we turn to (27). For any $j \in \{0, 1, \dots, n-1\}$, since $v_0^{(k)} = 0$, we get

$$\begin{aligned} S_k(\alpha\,\mathbf{x}) &= \sum_{j=1}^{n-1} x_j v_j^{(k)} - \sum_{j=1}^{n-1} x_{n-j} v_j^{(k)} = \sum_{j=1}^{n-1} x_j v_j^{(k)} + \sum_{j=1}^{n-1} x_{n-j} v_{n-j}^{(k)} \\ &= 2\sum_{j=1}^{n-1} x_j v_j^{(k)} = 2\sum_{j=0}^{n-1} x_j v_j^{(k)} = 2\,S_k(\mathbf{x}). \qquad \square \end{aligned}$$

Observe that from (12), (23), (24) and (25), taking into account that $\sigma\,\mathbf{x}$ is symmetric and $\alpha\,\mathbf{x}$ is asymmetric, we obtain

$$\mathrm{IDSCT}(\mathbf{x}) = Q_n^T\,\mathbf{x} = \begin{pmatrix} \alpha_0\,\dfrac{\mathrm{C}_0(\sigma\,\mathbf{x})}{2} \\ \alpha_1\,\dfrac{\mathrm{C}_1(\sigma\,\mathbf{x})}{2} \\ \vdots \\ \alpha_m\,\dfrac{\mathrm{C}_m(\sigma\,\mathbf{x})}{2} \\ \alpha_{m+1}\,\dfrac{\mathrm{S}_{m-1}(\alpha\,\mathbf{x})}{2} \\ \vdots \\ \alpha_{n-1}\,\dfrac{\mathrm{S}_1(\alpha\,\mathbf{x})}{2} \end{pmatrix}. \tag{28}$$

Thus, to compute IDSCT($\mathbf{x}$), we determine the values $\mathrm{C}_k(\sigma\,\mathbf{x})$ for $k = 0, 1, \ldots, m$ and $\mathrm{S}_k(\alpha\,\mathbf{x})$ for $k = 1, 2, \ldots, m-1$, and successively we multiply such values by the constants $\alpha_k/2$, $k = 0, 1, \ldots, n-1$. Now we give the following.

Lemma 4. *Let $\mathbf{x} \in \mathbb{R}^n$ be symmetric. Then for $k = 1, 2, \ldots, \nu - 1$ we get*

$$\mathrm{C}_k(\mathbf{x}) = \mathrm{C}_k(\eta\,\mathbf{x}) + \frac{1}{2\,\cos\!\left(\dfrac{\pi\,k}{m}\right)}\,\mathrm{C}_k(\sigma\,\zeta\,\mathbf{x}). \tag{29}$$

Moreover, if $t = m - k$, then

$$\mathrm{C}_t(\mathbf{x}) = \mathrm{C}_k(\eta\,\mathbf{x}) - \frac{1}{2\,\cos\!\left(\dfrac{\pi k}{m}\right)}\,\mathrm{C}_k(\sigma\,\zeta\,\mathbf{x}). \tag{30}$$

Proof. We begin with (29). Since $\mathbf{x}$ is symmetric, we have

$$\mathrm{S}_k(\mathbf{x}) = \mathrm{S}_k(\eta\,\mathbf{x}) = 0. \tag{31}$$

From (16) and (31) we obtain

$$\mathrm{S}_k(\zeta\,\mathbf{x}) = -\frac{\sin\!\left(\dfrac{\pi k}{m}\right)}{\cos\!\left(\dfrac{\pi k}{m}\right)}\,\mathrm{C}_k(\zeta\,\mathbf{x}). \tag{32}$$

From (15), (26) and (32) we get (29). The equality in (30) follows from (17), (26) and (32). □

Now we define the following algorithm:

function CS($\mathbf{x}, n$)
if $n = 4$ **then**
  $c_0 = x_0 + 2x_1 + x_2$;
  $c_1 = x_0 - x_2$;
  $c_2 = x_0 - 2x_1 + x_2$;
else
  $\widetilde{\mathbf{c}}$ =CS($\eta\,\mathbf{x}, n/2$);
  $\overline{\mathbf{c}}$ =CS($\sigma\,\zeta\,\mathbf{x}, n/2$);
  for $k = 1, 2, \ldots n/4 - 1$ **do**
    $aux = 1/(2\cos(2\,\pi\,k/n))\,\overline{c}_k$;
    $c_k = \widetilde{c}_k + aux$;
    $c_{n/2-k} = \widetilde{c}_k - aux$;
  end for
  $aux = \overline{c}_0/2$;
  $c_0 = \widetilde{c}_0 + aux$;
  $c_{n/4} = \widetilde{c}_{n/4}$;
  $c_{n/2} = \widetilde{c}_0 - aux$;
end if
return c

Lemma 5. *For each symmetric vector* $\mathbf{x} \in \mathbb{R}^n$*, the vector* $\eta\,\mathbf{x} \in \mathbb{R}^m$ *is symmetric.*

Proof. Let $\widetilde{\mathbf{x}} = \eta\,\mathbf{x}$, where $\mathbf{x}$ is a symmetric vector. So, we have

$$\widetilde{x}_j = x_{2j} = x_{n-2j} = x_{2(n/2-j)} = \widetilde{x}_{n/2-j}, \quad j = 0, 1, \dots m. \tag{33}$$

Since $\widetilde{\mathbf{x}} \in \mathbb{R}^m$, from (33) it follows that $\widetilde{\mathbf{x}}$ is symmetric. □

Theorem 2. *Let* $\mathbf{x} \in \mathbb{R}^n$ *be a symmetric vector and* $\mathbf{c} = \mathrm{CS}(\mathbf{x}, n)$*. Then,*

$$c_j = \mathrm{C}_j(\mathbf{x}) = \mathbf{x}^T \mathbf{u}^{(j)}, \quad j = 0, 1, \dots, m.$$

Proof. First of all we prove that, when we call the function CS, the first variable is symmetric. Concerning the external call, the vector $\mathbf{x}$ is symmetric in CS. Now we consider the internal calls of CS at lines 7 and 8. By Lemma 5, $\eta\,\mathbf{x}$ is symmetric. Hence, line 7 of our algorithm satisfies the assertion. Furthermore, it is readily seen that the assertion is fulfilled also by line 8.

Now, let us suppose that $n = 4$. Then,

$$c_0 = \mathbf{x}^T\,\mathbf{u}^{(0)} = x_0 + x_1 + x_2 + x_3.$$

Since $\mathbf{x}$ is even, we get the formula at line 3 of the algorithm CS. Moreover,

$$\mathbf{u}^{(1)} = (1 \quad 0 \quad -1 \quad 0)^T,$$

and hence

$$c_1 = \mathbf{x}^T\,\mathbf{u}^{(1)} = x_0 - x_2,$$

obtaining the formula at line 4 of the algorithm CS. Furthermore,

$$\mathbf{u}^{(2)} = (1 \quad -1 \quad 1 \quad -1)^T,$$

and thus we have

$$c_2 = \mathbf{x}^T\,\mathbf{u}^{(2)} = x_0 - 2\,x_1 + x_2,$$

since $\mathbf{x}$ is symmetric. So, we obtain the formula at line 5 of the algorithm CS.

Now we consider the case $n > 4$. Since $\mathbf{x}$ is symmetric, from Lemma 4 we deduce that line 11 of the algorithm CS gives the correct value of $\mathrm{C}_k(\mathbf{x})$ for $k = 1, 2, \dots, \nu - 1$. As $\mathbf{x}$ is symmetric, from Lemma 4 we obtain that line 12 gives the exact value of $\mathrm{C}_{m-k}(\mathbf{x})$ for $k = 1, 2, \dots, \nu - 1$.

From (19) and (26) we deduce that line 15 of the function CS gives the correct value of $\mathrm{C}_0(\mathbf{x})$, and by (20) and (26) we obtain that line 17 of the function CS gives the exact value of $\mathrm{C}_m(\mathbf{x})$. Furthermore, by virtue of (21), line 16 of the function CS gives the correct value of $\mathrm{C}_\nu(\mathbf{x})$. This completes the proof. □

Theorem 3. *Suppose to have a library in which the values* $1/(2\cos(2\pi k/n))$ *have been stored for* $n = 2^r$, $r \in \mathbb{N}$, $r \geq 2$, *and* $k \in \{0, 1, \ldots, n-1\}$. *Then, the computational cost of a call of the function* CS *is given by*

$$A(n) = \frac{3}{4} n \log_2 n - \frac{1}{2} n + 1, \tag{34}$$

$$M(n) = \frac{1}{4} n \log_2 n + \frac{1}{2} n - 2, \tag{35}$$

where $A(n)$ (*resp.,* $M(n)$) *denotes the number of additions* (*resp., multiplications*) *requested to compute* CS.

Proof. Concerning line 8, we first compute $\sigma\, \zeta\, \mathbf{x}$. To do this, $\nu - 1$ sums and 2 multiplications are required. To compute the **for** loop at lines 9–13, $2\,\nu - 2$ sums and $\nu - 1$ multiplications are necessary. Moreover, to compute lines 14–17, 2 sums and one multiplication are necessary. Thus, the total number of additions is given by

$$A(n) = \frac{3}{4} n - 1 + 2\, A\Big(\frac{n}{2}\Big), \tag{36}$$

while the total number of multiplications is

$$M(n) = \frac{1}{4} n + 2 + 2\, M\Big(\frac{n}{2}\Big). \tag{37}$$

Concerning the initial case, we have

$$A(4) = 5, \quad M(4) = 2.$$

For every $n = 2^r$, with $r \in \mathbb{N}$, $r \geq 2$, let $A(n)$ be as in (34). We get $A(4) = 6 - 2 + 1 = 5$. Now we claim that the function $A(n)$ defined in (34) satisfies (36). Indeed, we have

$$A\Big(\frac{n}{2}\Big) = \frac{3}{8} n\, (\log_2 n - 1) - \frac{1}{4} n + 1,$$

and hence

$$\frac{3}{4} n - 1 + 2\, A\Big(\frac{n}{2}\Big) = \frac{3}{4} n - 1 + \frac{3}{4} n\, (\log_2 n - 1) - \frac{1}{2} n + 2 = A(n),$$

getting the claim.

Moreover, for any $n = 2^r$, with $r \in \mathbb{N}$, $r \geq 2$, let $M(n)$ be as in (35). One has $M(4) = 2 + 2 - 2 = 2$. Now we claim that the function $M(n)$ defined in (35) fulfils (37). It is

$$M\Big(\frac{n}{2}\Big) = \frac{1}{8} n\, (\log_2 n - 1) + \frac{1}{4} n - 2,$$

and hence

$$\frac{1}{4} n + 2 + 2\, M\Big(\frac{n}{2}\Big) = \frac{1}{4} n + 2 + \frac{1}{4} n\, (\log_2 n - 1) + \frac{1}{2} n - 4 = M(n),$$

obtaining the claim. □

Lemma 6. *Let* $\mathbf{x} \in \mathbb{R}^n$ *be asymmetric. Then, for* $k = 1, 2, \dots, \nu - 1$ *it is*

$$\mathrm{S}_k(\mathbf{x}) = \mathrm{S}_k(\eta\,\mathbf{x}) + \frac{1}{2\,\cos\Big(\dfrac{\pi k}{m}\Big)}\,\mathrm{S}_k(\alpha\,\zeta\,\mathbf{x}). \tag{38}$$

Moreover, if $t = m - k$, *then*

$$\mathrm{S}_t(\mathbf{x}) = -\mathrm{S}_k(\eta\,\mathbf{x}) + \frac{1}{2\,\cos\Big(\dfrac{\pi k}{m}\Big)}\,\mathrm{S}_k(\alpha\,\zeta\,\mathbf{x}). \tag{39}$$

Proof. We begin with (38). As $\mathbf{x}$ is asymmetric, we have

$$\mathrm{C}_k(\mathbf{x}) = \mathrm{C}_k(\eta\,\mathbf{x}) = 0. \tag{40}$$

From (15) and (40) we deduce

$$\mathrm{C}_k(\zeta\,\mathbf{x}) = \frac{\sin\Big(\dfrac{\pi k}{m}\Big)}{\cos\Big(\dfrac{\pi k}{m}\Big)}\,\mathrm{S}_k(\zeta\,\mathbf{x}). \tag{41}$$

From (16), (27) and (41) we get (38).

The relation (39) follows from (18), (27) and (41). □

Now we define the following algorithm:

function SN$(\mathbf{x}, n)$
if $n = 4$ **then**
  $s_1 = 2\,x_1$;
else
  $\widetilde{\mathbf{s}}$ =SN$(\eta\,\mathbf{x}, n/2)$;
  $\overline{\mathbf{s}}$ =SN$(\alpha\,\zeta\,\mathbf{x}, n/2)$;
  for $k = 1, 2, \dots n/4 - 1$ **do**
    $aux = 1/(2\cos(2\,\pi\,k/n))\,\overline{s}_k$;
    $s_k = \widetilde{s}_k + aux$;
    $s_{n/2-k} = aux - \widetilde{s}_k$;
  end for
  $s_{n/4} = 0$;
  for $j = 0, 2, \dots, n/4 - 2$ **do**
    $s_{n/4} = s_{n/4} + x_{2j+1} - x_{2j+3}$
  end for
  $s_{n/4} = 2\,s_{n/4}$
end if
return s

Lemma 7. *For every asymmetric vector* $\mathbf{x} \in \mathbb{R}^n$, *the vector* $\eta\,\mathbf{x} \in \mathbb{R}^m$ *is asymmetric.*

Proof. Let $\overline{\mathbf{x}} = \eta\,\mathbf{x}$, where $\mathbf{x}$ is asymmetric. Then,

$$\overline{x}_j = x_{2j} = -x_{n-2j} = -x_{2(n/2-j)} = -\overline{x}_{n/2-j}, \quad j = 0, 1, \ldots m. \tag{42}$$

Since $\widetilde{\mathbf{x}} \in \mathbb{R}^m$, from (42) it follows that $\overline{\mathbf{x}}$ is asymmetric. □

Lemma 8. *Given any asymmetric vector* $\mathbf{x} \in \mathbb{R}^n$*, set* $\widehat{\mathbf{x}} = \zeta\,\mathbf{x}$*. Then,*

$$\mathrm{S}_\nu(\widehat{\mathbf{x}}) = 2\sum_{j=0}^{\nu-1}(-1)^j\,\widehat{x}_{2j+1}. \tag{43}$$

Proof. Let $\mathbf{x} \in \mathbb{R}^n$ be any asymmetric vector. Then, $\widehat{\mathbf{x}} = \zeta\,\mathbf{x} \in \mathbb{R}^m$. It is not difficult to check that

$$\widehat{x}_j = -\widehat{x}_{n-1-j}, \quad j = 0, 1, \ldots n-1. \tag{44}$$

Since $\mathbf{u}^{(\nu)} = (1\ -1\ \ldots 1\ -1)^T$ whenever $\mathbf{u}^{(\nu)} \in \mathbb{R}^m$, the equality in (43) follows from (22) and (44). □

Theorem 4. *Given an asymmetric vector* $\mathbf{x} \in \mathbb{R}^n$*, let* $\mathbf{s} = \mathrm{SN}(\mathbf{x}, n)$*. We get*

$$s_j = \mathrm{S}_j(\mathbf{x}) = \mathbf{x}^T\mathbf{v}^{(j)}, \quad j = 1, 2, \ldots, m-1.$$

Proof. First we prove that, when we call the function SN, the first variable is asymmetric. Concerning the external call, we have that $\mathbf{x}$ is asymmetric. Now we consider the internal calls of SN at lines 5 and 6. By Lemma 7, $\eta\,\mathbf{x}$ is asymmetric. Therefore, line 5 of our algorithm fulfils the assertion. Moreover, it is easy to see that line 6 satisfies the assertion too.

Now, let us suppose that $n = 4$. We have

$$\mathbf{v}^{(1)} = (0 \quad 1 \quad 0 \quad -1)^T,$$

and hence

$$s_1 = \mathbf{x}^T\,\mathbf{v}^{(1)} = x_1 - x_3 = 2x_1,$$

because $\mathbf{x}$ is asymmetric. Therefore, we obtain the formula at line 3 of the algorithm SN.

Now we consider the case $n > 4$. As $\mathbf{x}$ is asymmetric, from Lemma 6 we get that line 9 of the function SN gives the right value of $\mathrm{S}_k(\mathbf{x})$ for $k = 1, 2, \ldots, \nu-1$. Since $\mathbf{x}$ is asymmetric, from Lemma 6 we obtain that line 10 of the function SN gives the exact value of $\mathrm{S}_{m-k}(\mathbf{x})$ for $k = 1, 2, \ldots, \nu-1$. Thanks to Lemma 8 and asymmetry of $\mathbf{x}$, lines 12–16 give the exact value of $\mathrm{S}_\nu(\mathbf{x})$. This ends the proof. □

Theorem 5. *Suppose to have a library in which the values* $1/(2\cos(2\,\pi\,k/n))$ *have been stored for* $n = 2^r$*,* $r \in \mathbb{N}$*,* $r \geq 2$*, and* $k \in \{0,\ 1, \ldots, n-1\}$*. Then, the computational cost of a call of the function* SN *is given by*

$$A(n) = n\,\log_2 n - \frac{11}{4}n + 3, \tag{45}$$

$$M(n) = \frac{1}{4} n \log_2 n - \frac{1}{4} n, \tag{46}$$

where $A(n)$ (*resp.,* $M(n)$) *denotes the number of additions* (*resp., multiplications*) *required to compute* CS.

Proof. Concerning line 6, we first compute $\alpha\, \zeta\, \mathbf{x}$. To do this, $\nu - 1$ sums and no multiplications are required. To compute the **for** loop at lines 7–11, $2\,\nu - 2$ sums and $\nu - 1$ multiplications are necessary. Concerning the **for** loop at lines 13–15, ν sums are necessary. Finally, at line 16, one multiplication is necessary. Thus, the total number of additions is

$$A(n) = n - 3 + 2\, A\Big(\frac{n}{2}\Big), \tag{47}$$

and the total number of multiplications is

$$M(n) = \frac{1}{4} n + 2\, M\Big(\frac{n}{2}\Big). \tag{48}$$

Concerning the initial case, we have

$$A(4) = 0, \quad M(4) = 1.$$

For each $n = 2^r$, with $r \in \mathbb{N}$, $r \geq 2$, let $A(n)$ be as in (45). We get $A(4) = 8 - 11 + 3 = 0$. Now we claim that the function $A(n)$ defined in (45) satisfies (47). Indeed, we have

$$A\Big(\frac{n}{2}\Big) = -\frac{11}{8} n + \frac{1}{2} n\, (\log_2 n - 1) + 3.$$

Therefore,

$$n - 3 + 2\, A\Big(\frac{n}{2}\Big) = n - 3 - \frac{11}{4} n + n\, (\log_2 n - 1) + 6 = A(n),$$

which gives the claim.

Moreover, for every $n = 2^r$, with $r \in \mathbb{N}$, $r \geq 2$, let $M(n)$ be as in (46). One has $M(4) = 2 - 1 = 1$. Now we claim that the function $M(n)$ defined in (46) fulfils (48). Indeed,

$$M\Big(\frac{n}{2}\Big) = -\frac{1}{8} n + \frac{1}{8} n\, (\log_2 n - 1),$$

and hence

$$\frac{1}{4} n + 2\, M\Big(\frac{n}{2}\Big) = \frac{1}{4} n - \frac{1}{4} n + \frac{1}{4} n\, (\log_2 n - 1) = M(n),$$

getting the claim. □

Corollary 3. *Let* $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y}$ =IDSCT($\mathbf{x}$), $\mathbf{c}$ = CS($\sigma\,\mathbf{x}, n$) *and* $\mathbf{s}$ = SN($\alpha\,\mathbf{x}, n$). *Then,*

$$y_j = \begin{cases} \dfrac{\alpha_j\, c_j}{2}, & \text{if } j \le m; \\ \dfrac{\alpha_j\, s_{n-j}}{2}, & \text{otherwise.} \end{cases} \tag{49}$$

Proof. Since $\sigma\,\mathbf{x}$ is symmetric, from Theorem 2 we obtain

$$c_j = \mathrm{C}_j(\sigma\,\mathbf{x}), \quad j = 0, 1, \ldots, m.$$

Moreover, as $\alpha\,\mathbf{x}$ is asymmetric, from Theorem 4 we get

$$s_j = \mathrm{S}_j(\alpha\,\mathbf{x}), \quad j = 1, 2, \ldots, m-1.$$

Therefore, from (28) we deduce (49). □

To complete the computation of IDSCT($\mathbf{x}$) as in (28), we have to compute $\sigma\,\mathbf{x}$ and $\alpha\,\mathbf{x}$ and to multiply every entry of the result of CS($\sigma\,\mathbf{x}, \alpha\,\mathbf{x}, n$) by $\alpha_k/2$, $k = 0, 1, \ldots, n-1$. The computational cost of these operations is $O(n)$. Therefore, the total cost for the computation of (9) is $\frac{7}{4}\, n\, \log_2 n + o(n\, \log_2 n)$ additions and $\frac{1}{2}\, n\, \log_2 n + o(n\, \log_2 n)$ multiplications.

3.2 Computation of the Eigenvalues of a γ-Matrix

In this subsection we find an algorithm to compute the expression in (10). We first determine the eigenvalues of the matrix G in (8), in order to know the matrix $\Lambda^{(G)}$ in (10).

Since $G \in \mathcal{G}_n$ and $\mathcal{G}_n = \mathcal{C}_n \oplus \mathcal{B}_n$, we determine two matrices $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$, C =circ($\mathbf{c}$), B =rcirc($\mathbf{b}$). Observe that

$$\begin{aligned} g_{0,0} &= c_0 + b_0, & g_{\nu,\nu} &= c_0 + b_m, \\ g_{0,m} &= c_m + b_m, & g_{\nu,m+\nu} &= c_m + b_0. \end{aligned} \tag{50}$$

By solving the system in (50), we find c_0, b_0, c_m, b_m. Knowing c_0, from the principal diagonal of G it is not difficult to determine the numbers b_{2i}, $i = 1, 2, \ldots, \nu - 1$. If we know these quantities, it is possible to determine the numbers c_{2i}, $i = 1, 2, \ldots, \nu - 1$, from the first row of G. Moreover, note that

$$\begin{aligned} g_{0,1} &= c_1 + b_1, & g_{1,2} &= c_1 + b_3, \\ g_{0,3} &= c_3 + b_3, & g_{m-1,m+2} &= c_3 + b_1. \end{aligned} \tag{51}$$

By solving the system in (51), we obtain c_1, b_1, c_3, b_3. If we know c_1, then from the first diagonal superior to the principal diagonal of G it is not difficult to determine the numbers b_{2i+1}, $i = 2, 3, \ldots, \nu - 1$. Knowing these values, it is not difficult to find the quantities c_{2i+1}, $i = 2, 3, \ldots, \nu - 1$, from the first row of

G. It is possible to prove that the computational cost of all these operations is $O(n)$. Note that

$$G\,\mathbf{q}^{(j)} = C\,\mathbf{q}^{(j)} + B\,\mathbf{q}^{(j)} = \left(\lambda_j^{(C)} + \lambda_j^{(B)}\right)\,\mathbf{q}^{(j)}, \quad j = 0, 1, \ldots, n-1, \quad (52)$$

where the $\lambda_j^{(C)}$'s and the $\lambda_j^{(B)}$'s are the j-th eigenvalues of C and B, and the orders are given by

$$\lambda_j^{(C)} = \mathbf{c}^T \mathbf{u}^{(j)} \quad (53)$$

for $j = 0,\, 1, \ldots, n-1$, and

$$\lambda_j^{(B)} = \mathbf{b}^T \mathbf{u}^{(j)} \quad (54)$$

for $j = 0,\, 1, \ldots \lfloor \frac{n-1}{2} \rfloor$. Moreover, for $j = 1,\, 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, we get

$$\lambda_{n-j}^{(B)} = -\lambda_j^{(B)} \quad (55)$$

(see [5, Theorems 5 and 9]).

Proposition 2. *Given two symmetric vectors* $\mathbf{c}$, $\mathbf{b} \in \mathbb{R}^{\mathbf{n}}$ *such that*

$$\sum_{t=0}^{n-1} b_t = 0 \quad \textit{and} \quad \sum_{t=0}^{n-1} (-1)^t b_t \,=\, 0,$$

set $\mathbf{d}^{(C)} = \mathrm{CS}(\mathbf{c}, n)$ *and* $\mathbf{d}^{(B)} = \mathrm{CS}(\mathbf{b}, n)$. *Then the eigenvalues of* $G = \mathrm{circ}(\mathbf{c}) + \mathrm{rcirc}(\mathbf{b})$ *are given by*

$$\begin{aligned} &\lambda_0^{(G)} = d_0^{(C)}; \qquad \lambda_j^{(G)} = d_j^{(C)} + d_j^{(B)}, \quad j = 1, 2, \ldots m-1; \\ &\lambda_m^{(G)} = d_m^{(C)}; \qquad \lambda_j^{(G)} = d_{n-j}^{(C)} - d_{n-j}^{(B)}, \quad j = m+1, m+2, \ldots n-1. \end{aligned} \quad (56)$$

Proof. Since $\mathbf{c}$ and $\mathbf{b}$ are symmetric, from Theorem 2 we obtain

$$d_j^{(C)} = \mathrm{C}_j(\mathbf{c}) = \mathbf{c}^T \mathbf{u}^{(j)}, \quad d_j^{(B)} = \mathrm{C}_j(\mathbf{b}) = \mathbf{c}^T \mathbf{u}^{(j)}, \quad j = 0, 1, \ldots, m.$$

From (53)–(55) we get that $\lambda_j^{(C)} = d_j^{(C)}$ and $\lambda_j^{(B)} = d_j^{(B)}$ for $j = 0,\, 1, \ldots, m$, where $\lambda_j^{(C)}$ (resp., $\lambda_j^{(B)}$) are the eigenvalues of $\mathrm{circ}(\mathbf{c})$ (resp., $\mathrm{rcirc}(\mathbf{b})$). From (52)–(55) we obtain (56). □

To complete the computation of (10), we have to multiply the diagonal matrix $\Lambda^{(G)}$ by $\mathbf{z}$. The cost of this operation consists of n multiplications. Thus, by Theorem 3, the total cost to compute (10) is of $\frac{3}{2}\, n \log_2 n + o(n \log_2 n)$ additions and $\frac{1}{2}\, n \log_2 n + o(n \log_2 n)$ multiplications.

3.3 The DSCT Technique

Now we compute $\mathbf{y} = Q_n\,\mathbf{t} = \mathrm{DSCT}(\mathbf{t})$. Let $\overline{\alpha}$ and $\widetilde{\alpha}$ be as in (2). By (1), for $j = 0,\, 1, \ldots, m$ it is

$$y_j = \overline{\alpha}\, t_0 + (-1)^j\, \overline{\alpha}\, t_{n/2} + \widetilde{\alpha} \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\, t_k + \widetilde{\alpha} \sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\, t_{n-k} = \mathrm{DSCT}_j(\mathbf{t}), \tag{57}$$

and for $j = 1,\, 2, \ldots, m-1$ we have

$$\begin{aligned} y_{n-j} &= \overline{\alpha}\, t_0 + (-1)^j\, \overline{\alpha}\, t_{n/2} + \widetilde{\alpha} \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,(n-j)}{n}\Big)\, t_k \\ &\quad + \widetilde{\alpha} \sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,(n-j)}{n}\Big)\, t_{n-k} \\ &= \overline{\alpha}\, t_0 + (-1)^j\, \overline{\alpha}\, t_{n/2} + \widetilde{\alpha} \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\, t_k \\ &\quad - \widetilde{\alpha} \sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\, t_{n-k} = \mathrm{DSCT}_{n-j}(t). \end{aligned} \tag{58}$$

Now we define the functions $\varphi,\, \vartheta : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\begin{aligned} \varphi(\mathbf{t}) &= \overline{\mathbf{t}} = (t_0\ t_1\ \ldots t_{m-1}\ t_m\ t_{m-1}\ \ldots\ t_1), \\ \vartheta(\mathbf{t}) &= \widetilde{\mathbf{t}} = (0\ t_{n-1}\ t_{n-2}\ \ldots t_{m+1}\ 0\ -t_{m+1}\ \ldots\ -t_{n-1}). \end{aligned} \tag{59}$$

The following result holds.

Theorem 6. *For $j = 1,\, 2, \ldots, m-1$, it is*

$$\mathrm{DSCT}_j(\mathbf{t}) = \frac{\widetilde{\alpha}}{2}(\mathrm{C}_j(\overline{\mathbf{t}}) - t_0 - (-1)^j t_m + \mathrm{S}_j(\widetilde{\mathbf{t}})) + \overline{\alpha}\, t_0 + (-1)^j\, \overline{\alpha}\, t_m \tag{60}$$

and

$$\mathrm{DSCT}_{n-j}(\mathbf{t}) = \frac{\overline{\alpha}}{2}(\mathrm{C}_j(\overline{\mathbf{t}}) - t_0 - (-1)^j t_m - \mathrm{S}_j(\widetilde{\mathbf{t}})) + \overline{\alpha}\, t_0 + (-1)^j\, \overline{\alpha}\, t_m. \tag{61}$$

Moreover, one has

$$\mathrm{DSCT}_0(\mathbf{t}) = \frac{\widetilde{\alpha}}{2}(\mathrm{C}_0(\overline{\mathbf{t}}) - t_0 - t_m) + \overline{\alpha}\, t_0 + \overline{\alpha}\, t_m; \tag{62}$$

$$\mathrm{DSCT}_m(\mathbf{t}) = \frac{\widetilde{\alpha}}{2}(\mathrm{C}_m(\overline{\mathbf{t}}) - t_0 - t_m) + \overline{\alpha}\, t_0 + \overline{\alpha}\, t_m. \tag{63}$$

Proof. We have

$$\begin{aligned}
\mathrm{C}_j(\overline{\mathbf{t}}) = \overline{\mathbf{t}}^T \mathbf{u}^{(j)} &= \overline{t}_0 + (-1)^j \overline{t}_m + \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,\overline{t}_k + \sum_{k=m+1}^{n-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,\overline{t}_k \\
&= t_0 + (-1)^j\, t_m + \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,t_k + \sum_{k=1}^{m-1} \cos\Big(\frac{2\,(n-k)\,\pi\,j}{n}\Big)\,\overline{t}_{n-k} \\
&= t_0 + (-1)^j t_m + \sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,t_k + \sum_{k=1}^{m-1} \cos\Big(\frac{2\,(n-k)\,\pi\,j}{n}\Big)\,t_k \qquad (64) \\
&= t_0 + (-1)^j t_m + 2\sum_{k=1}^{m-1} \cos\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,t_k;
\end{aligned}$$

$$\begin{aligned}
\mathrm{S}_j(\widetilde{\mathbf{t}}) = \widetilde{\mathbf{t}}^T \mathbf{v}^{(j)} &= \sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,\widetilde{t}_k + \sum_{k=m+1}^{n-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,\widetilde{t}_k \\
&= \sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,t_{n-k} - \sum_{k=1}^{m-1} \sin\Big(\frac{2\,(n-k)\,\pi\,j}{n}\Big)\,t_{n-k} \qquad (65) \\
&= 2\sum_{k=1}^{m-1} \sin\Big(\frac{2\,k\,\pi\,j}{n}\Big)\,t_{n-k}.
\end{aligned}$$

From (57) and (64) we obtain (60), (62) and (63), while from (58) and (65) we get (61). □

The computational cost of the valuation of the functions φ and ϑ is linear, and the cost of the call of the functions CS and SN to determinate the values $\mathrm{C}_j(\overline{\mathbf{t}})$ for $j = 0,\ 1, \cdots, m$ and $\mathrm{S}_j(\widetilde{\mathbf{t}})$ for $j = 1,\ 2, \cdots, m$ is of $\frac{7}{4}\,n\,\log_2 n + o(n\,\log_2 n)$ additions and $\frac{1}{2}\,n\,\log_2 n + o(n\,\log_2 n)$ multiplications. The remaining computations for $\mathrm{DSCT}(\mathbf{t})$ are of $O(n)$. Thus, the total cost to compute $\mathrm{DSCT}(\mathbf{t})$ is of $\frac{7}{4}\,n\,\log_2 n + o(n\,\log_2 n)$ additions and $\frac{1}{2}\,n\,\log_2 n + o(n\,\log_2 n)$ multiplications. Therefore, the total cost to compute (8) is given by $5\,n\,\log_2 n + o(n\,\log_2 n)$ additions and $\frac{3}{2}\,n\,\log_2 n + o(n\,\log_2 n)$ multiplications. □

4 Conclusions

In this article we presented a new class of simultaneously diagonalizable real matrices, which can be expressed as the direct sum between a symmetric circulant matrix and a suitable reverse circulant matrix. Such matrices are used in several applications. We constructed a fast discrete transform in order to implement the multiplication between such a matrix and a real vector. We proved that the computational cost of such a transform is very low, especially in terms of multiplications. Our transform can be used also to compute multiplications

between other classes of simultaneously diagonalizable real matrices and real vectors. Thus, our algorithm can be used to improve the performance of various discrete transforms proposed for multiplications of real vectors.

Acknowledgments. This work was partially supported by University of Perugia, G.N.A.M.P.A. (Italian National Group of Mathematical Analysis, Probability and Applications) and I.N.d.A.M. (Italian National Institute of Higher Mathematics).

References

1. Andrecut, M.: Applications of left circulant matrices in signal and image processing. Mod. Phys. Lett. B **22**, 231–241 (2008)
2. Bini, D., Di Benedetto, F.: A new preconditioner for the parallel solution of positive definite Toeplitz systems. In: Second Ann. Symp. Parallel Algorithms and Architecture, Crete, Greece, pp. 220–223 (1990)
3. Bini, D., Favati, P.: On a matrix algebra related to the discrete Hartley transform. SIAM J. Matrix Anal. Appl. **14**, 500–507 (1993)
4. Boccuto, A., Gerace, I., Giorgetti, V.: Image Deblurring: a Class of Matrices Approximating Toeplitz Matrices. viXra 2201.0155 (2022)
5. Boccuto, A., Gerace, I., Giorgetti, V., Greco, F.: γ-matrices: a new class of simultaneously diagonalizable matrices. arXiv:2107.05890 (2021)
6. Carrasquinha, E., Amado, C., Pires, A.M., Oliveira, L.: Image reconstruction based on circulant matrices. Signal Process. Image Commun. **63**, 72–80 (2018)
7. Chan, R.: The spectrum of a family of circulant preconditioned Toeplitz systems. SIAM J. Numer. Anal. **26**, 503–506 (1989)
8. Codenotti, B., Gerace, I., Vigna, S.: Hardness results and spectral techniques for combinatorial problems on circulant graphs. Linear Algebra Appl. **285**, 123–142 (1998)
9. Di Fiore, C., Zellini, P.: Matrix algebras in optimal preconditioning. Linear Algebra Appl. **335**, 1–54 (2001)
10. Fischer, R., Huckle, T.: Using ω-circulant matrices for the preconditioning of Toeplitz systems. Selçuk J. Appl. Math. **4**, 71–88 (2003)
11. Frigo, M., Johnson, S.G.: A modified split-radix FFT with fewer arithmetic operations. IEEE Trans. Signal Process. **55**(1), 111–119 (2006)
12. Gerace, I., Pucci, P., Ceccarelli, N., Discepoli, M., Mariani, R.: A preconditioned finite element method for the p-Laplacian parabolic equation. Appl. Numer. Anal. Comput. Math. **1**, 155–164 (2004)
13. Győri, I., Horváth, L.: Utilization of circulant matrix theory in periodic autonomous difference equations. Int. J. Differ. Equ. **9**, 163–185 (2014)
14. Lei, Y.J., Xu, W.R., Lu, Y., Niu, Y.R., Gu, X.M.: On the symmetric doubly stochastic inverse Eigenvalue problem. Linear Algebra Appl. **445**, 181–205 (2014)
15. Shao, X., Johnson, S.G.: Type-II/III DCT/DST algorithms with reduced number of arithmetic operations. IEEE Signal Process. **88**(6), 1553–1564 (2008)