Journal of Computational Mathematics
Vol.28, No.6, 2010, 864–878.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1004-m3193

ON BLOCK MATRICES ASSOCIATED WITH DISCRETE TRIGONOMETRIC TRANSFORMS AND THEIR USE IN THE THEORY OF WAVE PROPAGATION*

Nikolaos L. Tsitsas
School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Iroon Polytechniou 9, GR-15780, Athens, Greece
Email: ntsitsas@esd.ntua.gr

Abstract

Block matrices associated with discrete Trigonometric transforms (DTT's) arise in the mathematical modelling of several applications of wave propagation theory including discretizations of scatterers and radiators with the Method of Moments, the Boundary Element Method, and the Method of Auxiliary Sources. The DTT's are represented by the Fourier, Hartley, Cosine, and Sine matrices, which are unitary and offer simultaneous diagonalizations of specific matrix algebras. The main tool for the investigation of the aforementioned wave applications is the efficient inversion of such types of block matrices. To this direction, in this paper we develop an efficient algorithm for the inversion of matrices with U-diagonalizable blocks (U a fixed unitary matrix) by utilizing the U-diagonalization of each block and subsequently a similarity transformation procedure. We determine the developed method's computational complexity and point out its high efficiency compared to standard inversion techniques. An implementation of the algorithm in Matlab is given. Several numerical results are presented demonstrating the CPU-time efficiency and accuracy for ill-conditioned matrices of the method. The investigated matrices stem from real-world wave propagation applications.

Mathematics subject classification: 65F05, 65T50, 74J20, 78A40, 15A09.
Key words: Discrete Trigonometric transforms, Block matrices, Efficient inversion algorithms, Wave radiation and scattering, Numerical methods in wave propagation theory.

1. Introduction

Discrete Trigonometric transforms (DTT's) play a significant role in wave scattering and radiation theory, signal processing, physics, and numerical linear algebra. Representative examples constitute the discrete Fourier transforms (DFT's), the discrete Hartley transforms (DHT's), the discrete Cosine transforms (DCT's), and the discrete Sine transforms (DST's). Their primary contribution lies in the significant reduction of the complexity in the associated mathematical problems. For example, applications of such appropriate transforms in differential and integral equations reduce them to algebraic equations, whose solutions are more easily obtained, see, e.g., [1, 2]. Moreover, in harmonic analysis as well as in signal processing the DFT decomposes a signal sequence into its frequency components [3]. It is important to note that this wide applicability of the DTT's is mainly justified by the existence of fast algorithms, that allow the transforms computations within $\mathcal{O}(n\log_2 n)$ (instead of $\mathcal{O}(n^2)$ when performing directly the matrix-vector product of length n) [4–6].

* Received August 17, 2009 / Revised version received November 9, 2009 / Accepted December 7, 2009 / Published online August 9, 2010 /

In particular, block matrices associated with DTT's arise in concrete physical and technological applications including: (i) the solution of wave scattering and radiation problems with the Method of Moments (MoM) [7, 8], (ii) the investigation and optimization of numerical methods for electromagnetic scattering problems, such as the Method of Auxiliary Sources (MAS) [9]- [10], (iii) the numerical solution of integral equations with the Boundary Element Method (BEM) [11], (iv) optical imaging [12], (v) image compression [13], (vi) efficient preconditioning of Toeplitz systems [14]. Besides, we point out that these applications exhibit the essential role of the inversion of such types of complex block matrices for the derivation of formulas determining the error bounds of numerical methods as well as for the numerical or semi-analytical computation of solutions [15].

The specific DTT's mentioned above are represented by the Fourier, Hartley, Cosine, and Sine matrices. These unitary matrices offer simultaneous diagonalizations of specific matrix algebras, including circulants, skew-circulants, Toeplitz-plus-Hankel, tridiagonal [16]. These matrix algebras are unified by considering the algebra Diag(U) of all U-diagonalizable matrices, for U a fixed unitary matrix.

For the mathematical modelling of the above mentioned wave applications we develop in this paper an efficient method for the inversion of an $m \times m$ block matrix $A = [A_{ij}]$ with U-diagonalizable blocks of order n $(i, j = 1, \ldots, m)$. First, we consider the diagonalizations $A_{ij} = U\Lambda_{ij}U^*$, where Λ_{ij} is the diagonal matrix containing the eigenvalues of A_{ij}, and hence the inversion of A is reduced to that of the $m \times m$ block matrix $\Lambda = [\Lambda_{ij}]$ with diagonal blocks of order n. For the inversion of Λ we construct, by using concepts of Graph Theory, an appropriate permutation matrix P so that the matrix $P\Lambda P^T = \mathrm{diag}(\Lambda'_1, \Lambda'_2, \ldots, \Lambda'_n)$ is block-diagonal with Λ'_k invertible $m \times m$ full matrices. The inverse $\Lambda^{-1} = [L_{ij}]$ of Λ is then determined by inverting each block Λ'_k with a standard LU direct solver. Finally, the inverse of A is given by the inverse block-diagonalization that is $A^{-1} = [UL_{ij}U^*]$. An implementation in Matlab of the above described algorithmic inversion is given in the Appendix.

For any one of the choices of U, that is Fourier, Hartley, Cosine, and Sine matrices, the matrix multiplications $U^*A_{ij}U$ and $UL_{ij}U^*$, appearing in the block-diagonalizations, are computed by applying the DTT's for $n \neq 2^p$ or the fast Trigonometric transforms (FTT's) for $n = 2^p$, that is the fast Fourier, Hartley, Cosine, and Sine transforms [4]- [6]. Hence, the computational complexity (i.e. the total number of required scalar complex multiplications) of the inversion algorithm is $n\mathcal{O}(m^3) + 2m^2n^2$ for $n \neq 2^p$ and $n\mathcal{O}(m^3) + 2m^2\mathcal{O}(n\log_2 n)$ for $n = 2^p$. This shows that the developed method is far more efficient than the LU decomposition applied to the original matrix A, having complexity $\mathcal{O}(m^3n^3)$. We note that in several wave applications the order n of each block may be chosen equal to 2^p by selecting suitable discretizations of the scatterer's or radiator's surface [7]- [10].

On the other hand, the above described inversion method can be also applied for the efficient determination of the eigenvalues of a matrix A with U-diagonalizable blocks. Specifically, the eigenvalues of A are those of all blocks Λ'_k and thus their computation requires $n\mathcal{O}(m^3) + m^2\mathcal{O}(n\log_2 n)$ multiplications. Besides, we notice the parallel nature of the proposed inversion algorithm, since the inversion of each specific block Λ'_k can be handled by a different processing unit. For a discussion on parallel algorithms for inverting block matrices which arise in inverse wave scattering theory see [17].

Several numerical results are presented exhibiting the efficiency of the proposed method and highlighting its beneficial contribution in the numerical implementation of certain scattering and radiation applications. We compare in terms of CPU time the developed algorithmic inversion with the LU method, applied to the initial matrix A, and investigate the achieved high speedup for various matrices with circulant, skew-circulant, Toeplitz-plus-Hankel, and tridiagonal blocks, appearing in real-world radiation applications, and in particular in the modelling of radiation by circular-loop antennas with the MoM [7]- [8]. Moreover, we examine the accuracy of the proposed inversion method for the case of ill-conditioned matrices, arising in the numerical solution of electromagnetic scattering problems by layered objects with the MAS [9]- [10]. The achieved inversion error is significantly reduced by the application of the developed method compared with that of the LU decomposition. This fact is justified numerically by the decrease of the conditions numbers of the blocks Λ'_k, which have to be inverted by the developed algorithm, and a related discussion is included.

2. Preliminaries, Notations and Terminology

A *Discrete Trigonometric Transform* (DTT) of length n is a linear transformation

$$T_M : \mathbb{C}^n \to \mathbb{C}^n, \quad T_M\mathbf{x} = M\mathbf{x}, \tag{2.1}$$

where M is a Fourier, Hartley, Cosine, or Sine square matrix of order n. The respective DTT is named the *Discrete Fourier, Hartley, Cosine, or Sine Transform* (DFT, DHT, DCT, or DST) of length n. We collect the 4 types I, II, III, and IV of each one of the $n \times n$ Fourier, Hartley, Cosine, and Sine matrices, commonly used in the literature (see, e.g., [6, 16, 18]) by $(j, k \in \{0, \ldots, n-1\})$

$$\left[F_n^{\mathrm{I}}\right]_{jk} = \frac{1}{\sqrt{n}} w_n^{jk}, \qquad \left[F_n^{\mathrm{II}}\right]_{jk} = \frac{1}{\sqrt{n}} w_{2n}^{j(2k+1)}, \tag{2.2a}$$

$$\left[F_n^{\mathrm{III}}\right]_{jk} = \frac{1}{\sqrt{n}} w_{2n}^{(2j+1)k}, \qquad \left[F_n^{\mathrm{IV}}\right]_{jk} = \frac{1}{\sqrt{n}} w_{4n}^{(2j+1)(2k+1)}, \tag{2.2b}$$

$$\left[H_n^{\mathrm{I}}\right]_{jk} = \frac{1}{\sqrt{n}} \mathrm{cas}\left(\frac{2\pi jk}{n}\right), \qquad \left[H_n^{\mathrm{II}}\right]_{jk} = \frac{1}{\sqrt{n}} \mathrm{cas}\left(\frac{2\pi j(2k+1)}{2n}\right), \tag{2.3a}$$

$$\left[H_n^{\mathrm{III}}\right]_{jk} = \frac{1}{\sqrt{n}} \mathrm{cas}\left(\frac{2\pi (2j+1)k}{2n}\right), \qquad \left[H_n^{\mathrm{IV}}\right]_{jk} = \frac{1}{\sqrt{n}} \mathrm{cas}\left(\frac{2\pi (2j+1)(2k+1)}{4n}\right), \tag{2.3b}$$

$$\begin{aligned}
\left[C_n^{\mathrm{I}}\right]_{jk} &= \sqrt{\frac{2}{n-1}} \delta_j \delta_k \epsilon_j \epsilon_k \cos\left(\frac{jk\pi}{n-1}\right), \qquad &\left[C_n^{\mathrm{II}}\right]_{jk} &= \sqrt{\frac{2}{n}} \delta_j \cos\left(\frac{j(k+\frac{1}{2})\pi}{n}\right), \\
\left[C_n^{\mathrm{III}}\right]_{jk} &= \sqrt{\frac{2}{n}} \delta_k \cos\left(\frac{(j+\frac{1}{2})k\pi}{n}\right), \qquad &\left[C_n^{\mathrm{IV}}\right]_{jk} &= \sqrt{\frac{2}{n}} \cos\left(\frac{(j+\frac{1}{2})(k+\frac{1}{2})\pi}{n}\right),
\end{aligned} \tag{2.4}$$

$$\begin{aligned}
\left[S_n^{\mathrm{I}}\right]_{jk} &= \sqrt{\frac{2}{n+1}} \sin\left(\frac{(j+1)(k+1)\pi}{n+1}\right), \qquad &\left[S_n^{\mathrm{II}}\right]_{jk} &= \sqrt{\frac{2}{n}} \epsilon_j \sin\left(\frac{(j+1)(k+\frac{1}{2})\pi}{n}\right), \\
\left[S_n^{\mathrm{III}}\right]_{jk} &= \sqrt{\frac{2}{n}} \epsilon_k \sin\left(\frac{(j+\frac{1}{2})(k+1)\pi}{n}\right), \qquad &\left[S_n^{\mathrm{IV}}\right]_{jk} &= \sqrt{\frac{2}{n}} \sin\left(\frac{(j+\frac{1}{2})(k+\frac{1}{2})\pi}{n}\right).
\end{aligned} \tag{2.5}$$

where $w_n = \exp(-2\pi i/n)$, $i^2 = -1$,

$$\mathrm{cas}(t) := \cos(t) + \sin(t),$$

$\delta_0 = 1/\sqrt{2}$, δ_p=1 for $p \in \{1,\dots,n-1\}$, ϵ_q=1 for $q \in \{0,\dots,n-2\}$, and $\epsilon_{n-1} = 1/\sqrt{2}$. For consistency with literature, matrix entries are indexed throughout from 0 to $n-1$.

The DHT's of type II-IV are utilized in signal processing applications [19]. The DFT's and DHT's of Type IV are usually mentioned as *generalized* or *fractional* transforms. For an extensive survey on DCT's and DST's see [6].

The unitary Fourier, Hartley, Cosine, and Sine matrices diagonalize specific classes of matrices (circulants, skew-circulants, Toeplitz-plus-Hankel, tridiagonal, e.t.c). To formalize a unified situation, we introduce, in a similar way to [16], for a *fixed* unitary $n \times n$ complex matrix U, the term *matrix algebra of U-diagonalizable* $n \times n$ *matrices*:

$$\mathrm{Diag}(U) = \{U\Lambda U^*, \Lambda \text{ diagonal}\} = \{A : U^*AU \, \text{diagonal}\}. \tag{2.6}$$

$\mathrm{Diag}(U)$ is a commutative matrix algebra, referred to as *U-matrix algebra*. The matrices of $\mathrm{Diag}(U)$ are named *U-diagonalizable matrices*. The columns of U constitute a universal set of eigenvectors for *all* elements of the U-matrix algebra.

Now, the following notations enable us to determine certain specific U-matrix algebras.

First, we consider the direct sums of the matrices

$$J'_n = 1 \oplus J_{n-1} = \mathrm{diag}(1, J_{n-1}) \quad \text{and} \quad J''_n = (-1) \oplus J_{n-1} = \mathrm{diag}(-1, J_{n-1}),$$

where J_n the counteridentity matrix of order n (having ones on the main anti-diagonal and zeros elsewhere). A vector $\mathbf{a} \in \mathbb{C}^n$ is called J'_n*-even*, J''_n*-even*, J'_n*-odd*, or J''_n*-odd* whenever $J'_n\mathbf{a} = \mathbf{a}$, $J''_n\mathbf{a} = \mathbf{a}$, $J''_n\mathbf{a} = -\mathbf{a}$, or $J''_n\mathbf{a} = -\mathbf{a}$.

Let $\mathcal{C}_n$ and $\mathcal{SC}_n$ be the spaces of all circulant $A = \mathrm{circ}(a_1, a_2, \dots, a_n)$, and skew-circulant $A = \mathrm{scirc}(a_1, a_2, \dots, a_n)$, $n \times n$ matrices ([20], p. 66, p. 83). We denote by $\mathcal{C}_n^0$, $\mathcal{C}_n^1$ and $\mathcal{SC}_n^0$, $\mathcal{SC}_n^1$ the spaces of all circulant $\mathrm{circ}(\mathbf{a})$ and skew-circulant matrices $\mathrm{scirc}(\mathbf{a})$ with J'_n-even, J'_n-odd and J''_n-even, J''_n-odd vectors $\mathbf{a}$.

An $n \times n$ matrix $T_n = [t_{jk}]$ is called symmetric Toeplitz if there exists a vector $\mathbf{t} = [t_0, t_1, \dots, t_{n-1}]$ with $t_{jk} = t_{|j-k|}$, and thus T_n is determined by the vector $\mathbf{t}$ of its first row. Besides, an $n \times n$ matrix $HA_n = [h_{jk}]$ is called Hankel if there exists a vector $\mathbf{h} = [h_0, h_1, \dots, h_{2n-2}]$ with $h_{jk} = \mathbf{h}(j+k-1)$ and hence HA_n is determined by the vectors $\mathbf{h}(0{:}n-1)$ and $\mathbf{h}(n-1{:}2n-2)$ of its first column and last row. By $\mathcal{T}_n$ and $\mathcal{HA}_n$ we denote the vector spaces of symmetric Toeplitz and Hankel $n \times n$ matrices respectively. To this respect, we note that $\mathcal{T}_n$ contains $\mathcal{C}_n^0$ and $\mathcal{SC}_n^1$, while $\mathcal{HA}_n$ contains $J'_n\mathcal{C}_n^1$, $J''_n\mathcal{SC}_n^0$, $J_n\mathcal{C}_n^1$, and $J_n\mathcal{SC}_n^0$.

Furthermore, the family of the $n \times n$ tridiagonal matrices

$$\mathcal{B}_n(\boldsymbol{\beta}) = \frac{1}{2}\begin{bmatrix} \beta_1 & \beta_2 & & & & \\ \beta_5 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \vdots & \ddots & \vdots & \\ & & & 1 & 0 & \beta_6 \\ & & & & \beta_3 & \beta_4 \end{bmatrix} \tag{2.7}$$

is diagonalized by each one of the Cosine and Sine matrices for $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6]$ with values given in Table 2.1 (see also [6]). Besides, we introduce here two additional parameters β_5 and β_6 (extending the case $\beta_5 = \beta_6 = 1$ considered in [18]) so that the 8 matrices $\mathcal{B}_n$ are diagonalized by the *unitary* matrices C_n^X and S_n^X of (2.4) and (2.5) (for $\mathrm{X} \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$).

Table 2.1: The coefficient vectors $\boldsymbol{\beta}^c_{\mathrm{X}}$ and $\boldsymbol{\beta}^s_{\mathrm{X}}$ for each type of the DCT's and DST's.

	$\boldsymbol{\beta}^c_{\mathrm{X}}$
DCT-I	$(0,\sqrt{2},\sqrt{2},0,\sqrt{2},\sqrt{2})$
DCT-II	$(1,1,1,1,1,1)$
DCT-III	$(0,\sqrt{2},1,0,\sqrt{2},1)$
DCT-IV	$(1,1,1,-1,1,1)$
	$\boldsymbol{\beta}^s_{\mathrm{X}}$
DST-I	$(0,1,1,0,1,1)$
DST-II	$(-1,1,1,-1,1,1)$
DST-III	$(0,1,\sqrt{2},0,1,\sqrt{2})$
DST-IV	$(-1,1,1,1,1,1)$

Now, under the above considerations and summarizing results of [16, 18], we have

$$\begin{aligned}
&\mathrm{Diag}(F_n^{\mathrm{I}}) = \mathrm{Diag}(F_n^{\mathrm{III}}) = \mathcal{C}_n, && \mathrm{Diag}(F_n^{\mathrm{II}}) = \mathrm{Diag}(F_n^{\mathrm{IV}}) = \mathcal{SC}_n,\\
&\mathrm{Diag}(H_n^{\mathrm{I}}) = \mathcal{C}_n^0 \oplus J_n'\mathcal{C}_n^1, && \mathrm{Diag}(H_n^{\mathrm{II}}) = \mathcal{SC}_n^1 \oplus J_n''\mathcal{SC}_n^0,\\
&\mathrm{Diag}(H_n^{\mathrm{III}}) = \mathcal{C}_n^0 \oplus J_n\mathcal{C}_n^1, && \mathrm{Diag}(H_n^{\mathrm{IV}}) = \mathcal{SC}_n^1 \oplus J_n\mathcal{SC}_n^0,\\
&\mathrm{Diag}(C_n^{\mathrm{X}}) = \mathcal{B}_n(\boldsymbol{\beta}^c_{\mathrm{X}}), && \mathrm{Diag}(S_n^{\mathrm{X}}) = \mathcal{B}_n(\boldsymbol{\beta}^s_{\mathrm{X}}),
\end{aligned} \tag{2.8}$$

where $\oplus$ denotes the *orthogonal sum of linear subspaces* of $\mathbb{C}^{n\times n}$ with respect to the Frobenius inner product [16]. In particular, from (2.8), $\mathrm{Diag}(H_n^X)$ are subspaces of $\mathcal{T}_n + \mathcal{HA}_n$, and hence every matrix of $\mathrm{Diag}(H_n^X)$ may be expressed as a symmetric Toeplitz plus Hankel matrix.

Besides, concrete wave propagation applications, analyzed in [7]- [10], demonstrate the significance of the generalization of the notion of U-diagonalizable matrices to that of *matrices with U-diagonalizable blocks*, i.e. $m\times m$ block matrices $A = \left[A_{ij}\right]$ with A_{ij} $n\times n$ U-diagonalizable matrices.

3. Inversion Algorithm

We develop an algorithmic approach for the inversion of an $m\times m$ block matrix $A = \left[A_{ij}\right]$, with U-diagonalizable blocks of order n based on: (i) the block-diagonalization of A and (ii) a similarity transformation algorithm for the inversion of the block matrix, determined by the eigenvalues of the blocks A_{ij}.

For the development of the inversion algorithm we utilize essentially the following

Theorem 3.1. *Let U be a unitary matrix of order n and $A = \left[A_{ij}\right]$ an $m\times m$ block matrix with U-diagonalizable blocks of order n. Then, A is block-diagonalized as*

$$A = \mathrm{diag}(U,U,...,U)\,[\Lambda_{ij}]\,\mathrm{diag}(U^*,U^*,...,U^*), \quad \Lambda_{ij} = \mathrm{diag}(\lambda_{ij}^1,\lambda_{ij}^2,...,\lambda_{ij}^n), \tag{3.1}$$

where λ_{ij}^k are the eigenvalues of A_{ij} ($k\in\{1,\dots,n\}$ and $i,j\in\{1,\dots,m\}$) and all Λ_{ij} have as eigenvectors the columns of U.

Besides, if A is invertible, the inverse A^{-1} has the block-diagonalization

$$A^{-1} = \mathrm{diag}(U,U,...,U)\,[L_{ij}]\,\mathrm{diag}(U^*,U^*,...,U^*), \quad \Lambda^{-1} = \left[L_{ij}\right]. \tag{3.2}$$

Eq. (3.2) reduces the inversion of A to that of the $m\times m$ block matrix $\Lambda = \left[\Lambda_{ij}\right]$ with diagonal blocks of order n. The inverse Λ^{-1} is computed below by applying a similarity transformation algorithm.

A similar block-diagonalization to that of the previous Theorem has also been considered in [21] for the particular case of matrices with circulant blocks (each diagonalized by the Fourier matrix). However, the inversion of the respective matrix Λ in [21] is obtained by a recursive method, different than the one presented hereafter.

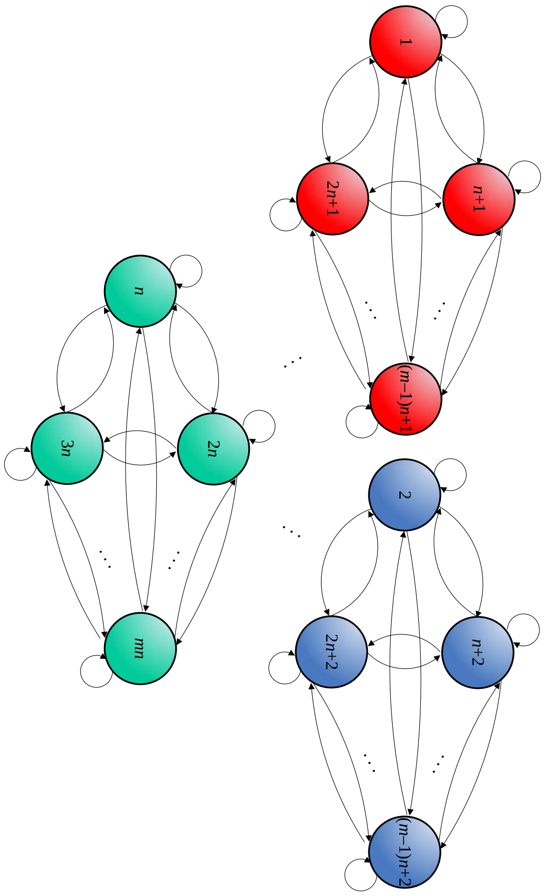


Fig. 3.1. Graph $G(\Lambda)$ of the $m \times m$ block matrix $\Lambda = \left[\Lambda_{ij}\right]$ with diagonal blocks of order n

The basic idea for the inversion of Λ lies in the application of a suitable similarity transformation, making the matrix with diagonal blocks Λ similar to a block-diagonal matrix Λ'. To this direction, we construct the graph $G(\Lambda)$ of Λ (see Fig. 3.1), where only the diagonal elements of the blocks Λ_{ij} are considered, including the zero ones. In view of Fig. 3.1, the graph $G(\Lambda)$ is a union of n disjoint subgraphs each with m nodes. Thus, taking into account this decomposition of the graph, we construct the permutation matrix

$$P = \left[\mathbf{e}_1, \mathbf{e}_{n+1}, \ldots, \mathbf{e}_{(m-1)n+1}, \mathbf{e}_2, \mathbf{e}_{n+2}, \ldots, \mathbf{e}_{(m-1)n+2}, \ldots, \mathbf{e}_n, \mathbf{e}_{2n}, \ldots, \mathbf{e}_{mn}\right]^T,$$

where $(\mathbf{e}_i)_{i\in\{1,\ldots,mn\}}$ the standard base of $\mathbb{R}^{mn}$. By the definition of P we get the similarity transformation

$$P\Lambda P^T = \Lambda'_1 \oplus \Lambda'_2 \oplus \cdots \oplus \Lambda'_n \equiv \Lambda', \tag{3.3}$$

with

$$[\Lambda'_k]_{ij} = \lambda^k_{ij} \qquad (k \in \{1,\ldots,n\}, \quad i,j \in \{1,\ldots,m\}).$$

Thus, so far the original $m \times m$ block matrix Λ with diagonal blocks Λ_{ij} of order n has been transformed to the block-diagonal matrix Λ' with n full blocks Λ'_k of order m.

Furthermore, the inverse Λ'^{-1} of Λ' is determined by inverting each block Λ'_k with a standard direct solver such as the LU decomposition. We note that the blocks Λ'_k are invertible in case the matrix Λ is invertible and the inverse Λ'^{-1} is given by

$$\Lambda'^{-1} = \Lambda_1'^{-1} \oplus \Lambda_2'^{-1} \oplus \cdots \oplus \Lambda_n'^{-1}, \quad [\Lambda_k'^{-1}]_{ij} = \ell_{ij}^k. \tag{3.4}$$

Now, $\Lambda^{-1}=\left[L_{ij}\right]$ is derived from Λ'^{-1} by applying the inverse similarity transformation

$$\Lambda^{-1} = P^T \Lambda'^{-1} P, \tag{3.5}$$

which implies that the $n \times n$ diagonal blocks L_{ij} of the matrix Λ^{-1} are of the form

$$L_{ij} = \mathrm{diag}(\ell_{ij}^1, \ell_{ij}^2, ..., \ell_{ij}^n). \tag{3.6}$$

Finally, application of the block-diagonalization (3.2) to Λ^{-1} gives the inverse A^{-1} of A.

4. Computational Complexity of the Inversion

The computational complexity, which we consider, expresses the total number of required scalar complex multiplications. For the determination of the computational complexity $C(m,n)$ of the proposed algorithm for the inversion of a matrix A with U-diagonalizable blocks we need the complexities: (i) C_{bdA} of the block-diagonalization of A, (ii) C_S of the similarity transformation algorithm for the inversion of Λ, and (iii) $C_{bdA^{-1}}$ of the block-diagonalization of A^{-1}. Clearly, the complexity $C(m,n)$ is the sum

$$C(m,n) = C_{bdA}(m,n) + C_S(m,n) + C_{bdA^{-1}}(m,n). \tag{4.1}$$

The unitary matrix U is introduced to unify the 16 cases of the Fourier, Hartley, Cosine, and Sine matrices, described in (2.2)-(2.5). For these choices of U the matrix multiplications $U^* A_{ij} U$ and $U L_{ij} U^*$, appearing in the block-diagonalizations (3.1) and (3.2), may be computed by applying the discrete (DTT's) or the fast (FTT's) Trigonometric transforms.

The following Theorem gives the complexity $C(m,n)$ of the algorithmic inversion

Theorem 4.1. *Let $A = \left[A_{ij}\right]$ be an invertible $m \times m$ block matrix with U-diagonalizable blocks of order n, where U represents any one of F_n^{X}, H_n^{X}, C_n^{X}, or S_n^{X} for $\mathrm{X} \in \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$. Then, the computational complexity C(m,n) of the algorithmic inversion of A, developed in Section 3, is given by*

$$C(m,n) = \begin{cases} n\mathcal{O}(m^3) + 2m^2n^2, & \log_2 n \notin \mathbb{N}, \\ n\mathcal{O}(m^3) + 2m^2\mathcal{O}(n\log_2 n), & \log_2 n \in \mathbb{N}. \end{cases} \tag{4.2}$$

Proof. The complexity C_{bdA} is that of the determination of the blocks Λ_{ij} of (3.1), which are computed for $n \neq 2^p$ by the DTT's (2.1)-(2.5), requiring n^2 multiplications, and for $n=2^p$ by the FTT's, requiring $O(n\log_2 n)$ multiplications (see [5], [6]), yielding

$$C_{bdA}(m,n) = \begin{cases} m^2n^2, & \log_2 n \notin \mathbb{N}, \\ m^2\mathcal{O}(n\log_2 n), & \log_2 n \in \mathbb{N}. \end{cases}$$

Furthermore, C_S is determined by the complexity of the inversions of the n blocks Λ'_k of order m of matrix Λ' in (3.3) with an LU solver, resulting $C_S(m,n) = n\mathcal{O}(m^3)$. Note that the similarity transformations (3.3) and (3.5) do not contribute to C_S, since they involve only rows and columns interchanges.

Finally, in order to compute the block-diagonalization (3.2), we apply the inverse DTT's or FTT's to the blocks L_{ij} of (3.6). Since both the DTT's and the inverse DTT's require n^2 multiplications and the respective FTT's $\mathcal{O}(n\log_2 n)$ multiplications, we conclude that $C_{bdA^{-1}}(m,n) = C_{bdA}(m,n)$. Thus, the desired (4.2) follows by (4.1). $\square$

The following remarks concern useful aspects of the inversion's computational complexity

(i) *Efficient computation of the eigenvalues of matrices with U-diagonalizable blocks*: by combining (3.1) and (3.3) we have

$$A = (U \oplus U \oplus \cdots \oplus U)\ P^T\,(\Lambda'_1 \oplus \Lambda'_2 \oplus \cdots \oplus \Lambda'_n)\,P\ (U^* \oplus U^* \oplus \cdots \oplus U^*).$$

Thus, the eigenvalues of A coincide with those of all blocks Λ'_k and their complexity is

$$C_{eig}(m,n) = \begin{cases} n\mathcal{O}(m^3) + m^2n^2, & \log_2 n \notin \mathbb{N}, \\ n\mathcal{O}(m^3) + m^2\mathcal{O}(n\log_2 n), & \log_2 n \in \mathbb{N}. \end{cases} \tag{4.3}$$

The factor 2 in C of (4.2) does not appear in C_{eig} of (4.3), since for the eigenvalues computation are required only the direct (and not the inverse) DTT's or FTT's.

(ii) *Parallel nature of the proposed algorithm*: for n large and m small enough, the block-diagonalizations (3.1) and (3.2) require m^2 processing nodes for the execution of each one of the direct and inverse DTT's, and then the complexity is decreased to

$$C_{par1}(m,n) = n\mathcal{O}(m^3) + 2\mathcal{O}(n\log_2 n).$$

Moreover, for m large and n small enough, the inversion of each block Λ'_k may be assigned to a different one from n processing nodes and the complexity reduces to

$$C_{par2}(m,n) = \mathcal{O}(m^3) + 2m^2\mathcal{O}(n\log_2 n).$$

(iii) From (4.2), the algorithmic inversion works best for $n = 2^p$, where the block- diagonalizations are computed at "FTT-speed". On the other hand, the complexity of the LU decomposition and the numerical inversion of the matrix A is of order $C_{LU}(m,n) = \mathcal{O}(m^3n^3)$ and hence the proposed algorithmic inversion is much faster than the LU method (as shown also in the numerical results of Section 5).

(iv) The accurate numerical solutions of certain electromagnetic scattering and radiation problems [7–10] require large n and small m (usually 2 or 4), yielding

$$C(n) = \begin{cases} \mathcal{O}(n^2), & \log_2 n \notin \mathbb{N} \\ \mathcal{O}(n\log_2 n), & \log_2 n \in \mathbb{N} \end{cases}, \qquad C_{LU}(n) = \mathcal{O}(n^3),$$

and thus $C_{LU}(n)$ is at least one order of magnitude greater than $C(n)$.

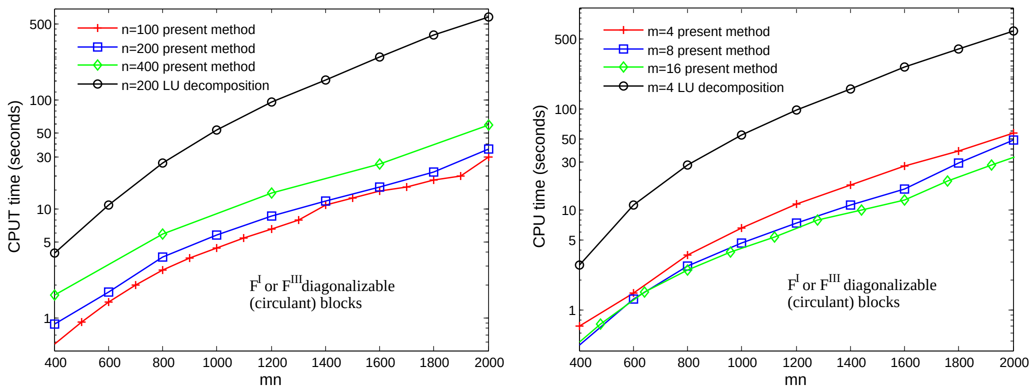
5. Numerical Results

In this Section we present numerical results for the developed inversion algorithm with respect to (i) the execution (CPU) time and (ii) the accuracy for ill-conditioned matrices. The investigated block matrices stem from real-world scattering and radiation applications. The algorithm is implemented by the Matlab function *inv_unit_diagon_blocks*, given in the Appendix. The numerical results are compared with the LU decomposition method, using Gaussian elimination with partial pivoting, and the efficiency of the developed algorithm is discussed. All numerical experiments are implemented in Matlab 7 with double precision arithmetic and executed in an Intel Pentium M processor 1.60 GHz with 504 MB of RAM.

5.1. Execution Time

First, we consider matrices with circulant and skew-circulant blocks, appearing in the real-world application of the modelling of radiation by circular-loop antenna arrays with the Method of Moments [7, 8]. For this application m represents the number of circular loops, while n the number of basis functions utilized. Figs. 5.1a and 5.1b depict the CPU time for the inversions with respect to mn of the present method and the LU decomposition for $m \times m$ block matrices with F_n^{I}- or F_n^{III}-diagonalizable circulant blocks of order n for fixed (a) n=100, 200, 400 and (b) m=4, 8, 16. The situation is similar in the case of block matrices with F_n^{II} or F_n^{IV}-diagonalizable skew-circulant blocks. The developed algorithmic inversion is clearly faster than the LU inversion and the logarithmic scale in Fig. 5.1 emphasizes its lower complexity. This CPU-time improvement, achieved by the developed method, may lead to the efficient and fast modelling of electrically large circular-loop antenna arrays.

Furthermore, we consider matrices with symmetric Toeplitz-plus-Hankel blocks, composed of random real entries, chosen from a normal distribution with mean zero, variance one, and standard deviation one. The Toeplitz and the Hankel counterparts of these blocks, belonging to H_n^{X} matrix algebras, are constructed by following the general guidelines of [16], modified into the present context in Table 5.1.

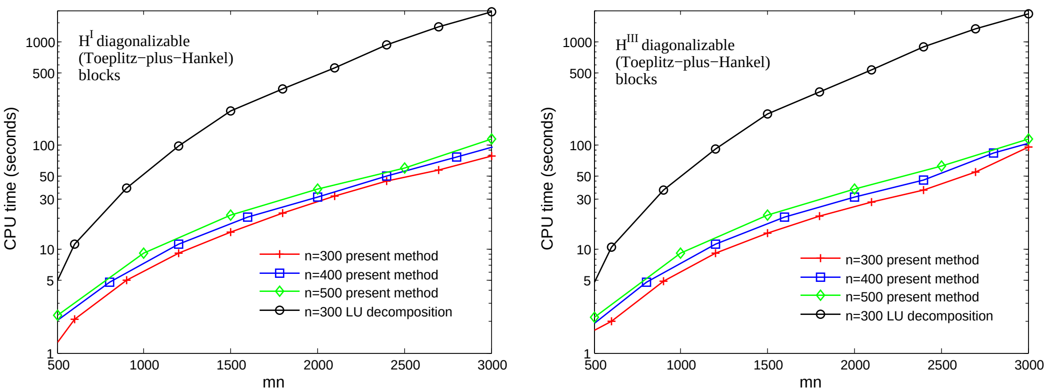


(a) (b)

Fig. 5.1. CPU time for the inversions with respect to mn of the present method and the LU decomposition for $m \times m$ block matrices with F_n^{I}- or F_n^{III}-diagonalizable circulant blocks of order n for: (a) n=100, 200, 400 and (b) m=4, 8, 16, appearing in modelling of radiation by circular-loop antenna arrays.

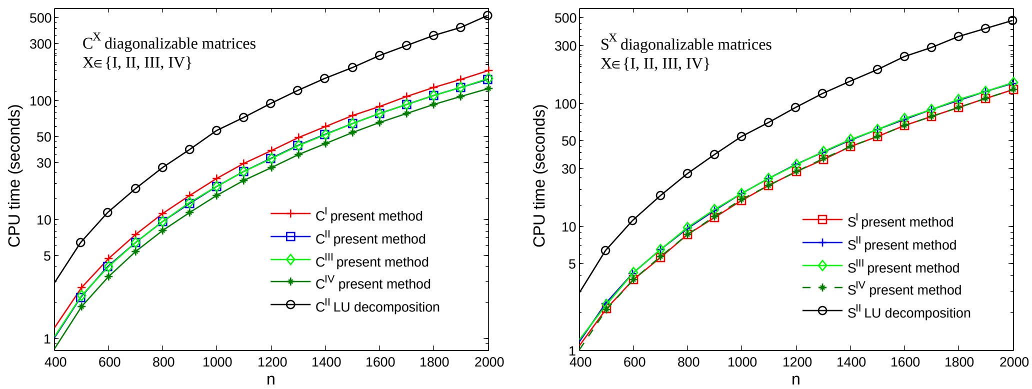
Figs. 5.2a and 5.2b show the CPU time for the inversions with respect to mn of the present method and the LU decomposition for $m \times m$ block matrices with (a) H_n^{I}- and (b) H_n^{III}-diagonalizable Toeplitz-plus-Hankel blocks of order n for n=300, 400, 500. By the statements of Fig. 5.2 the present algorithmic inversion is carried out in a few seconds even for large mn. However, the computational cost of the LU inversion increases rapidly with mn and this fact makes the application of this inversion prohibitive. Moreover, for arbitrary fixed n (m) the difference between the CPU time of the LU and the present inversion method increases significantly with m (n). Similar conclusions hold for block matrices with H_n^{II} or H_n^{IV}-diagonalizable Toeplitz-plus-Hankel blocks.

Finally, we consider tridiagonal matrices of the form (2.7). Figs. 5.3a and 5.3b depict the inversion CPU time with respect to n of the present method and the LU decomposition for (a) C_n^{X}- and (b) S_n^{X}-diagonalizable matrices for X=I, II, III, IV. For large n the achieved speedup of the developed method compared to the LU decomposition increases rapidly. For fixed n the CPU time of the algorithmic inversion does not vary significantly with X.



(a) (b)

Fig. 5.2. CPU time for the inversions with respect to mn of the present method and the LU decomposition for $m \times m$ block matrices with (a) H_n^{I}- and (b) H_n^{III}-diagonalizable Toeplitz-plus-Hankel blocks of order n for n=300, 400, 500.



(a) (b)

Fig. 5.3. CPU time for the inversions with respect to n of the present method and the LU decomposition for (a) C_n^{X}- and (b) S_n^{X}-diagonalizable matrices for X=I, II, III, IV.

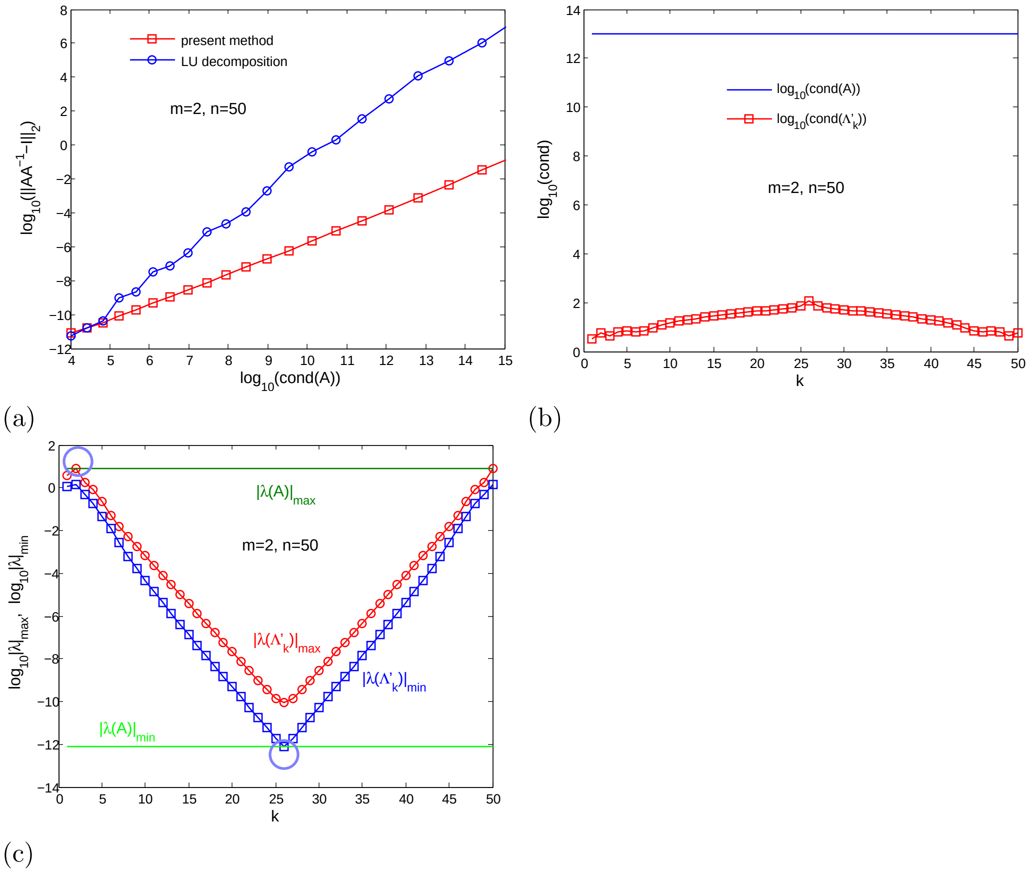


Fig. 5.4. (a) Inversion error vs condition number of a matrix A with circulant blocks for m=2, n=50, as obtained by the present method and the LU decomposition, (b) Condition number of the matrices Λ'_k, (c) Maximum and minimum eigenvalues of A and Λ'_k. The considered matrices appear in the numerical solution of wave scattering problems with the Method of Auxiliary Sources.

Table 5.1: The vectors $\mathbf{t}$ and $\mathbf{h}$, which determine respectively the Toeplitz and the Hankel counterparts of the symmetric Toeplitz-plus-Hankel matrices of H_n^{X} matrix algebras. For X=I, III and X=II, IV $\mathbf{t}$ is the first row of the circulant and skew-circulant matrix. For X=I, II and X=III, IV $\mathbf{h}$ is the first column and last row respectively of the corresponding Hankel matrix. ($\mathbf{x} \in \mathbb{C}^n$)

X	$\mathbf{x}^+$	$\mathbf{x}^-$	$\mathbf{t}$	$\mathbf{h}$
I	$\mathbf{x}^+ = \frac{1}{2}(\mathbf{x} + J'_n\mathbf{x})$	$\mathbf{x}^- = \frac{1}{2}(\mathbf{x} - J'_n\mathbf{x})$	$H_n^{\mathrm{I}}\mathbf{x}^+$	$H_n^{\mathrm{I}}\mathbf{x}^-$
II	$\mathbf{x}^+ = \frac{1}{2}(\mathbf{x} + J_n\mathbf{x})$	$\mathbf{x}^- = \frac{1}{2}(\mathbf{x} - J_n\mathbf{x})$	$H_n^{\mathrm{II}}\mathbf{x}^+$	$H_n^{\mathrm{II}}\mathbf{x}^-$
III	$\mathbf{x}^+ = \frac{1}{2}(\mathbf{x} + J'_n\mathbf{x})$	$\mathbf{x}^- = \frac{1}{2}(\mathbf{x} - J'_n\mathbf{x})$	$H_n^{\mathrm{I}}\mathbf{x}^+$	$H_n^{\mathrm{I}}\mathbf{x}^-$
IV	$\mathbf{x}^+ = \frac{1}{2}(\mathbf{x} + J_n\mathbf{x})$	$\mathbf{x}^- = \frac{1}{2}(\mathbf{x} - J_n\mathbf{x})$	$H_n^{\mathrm{II}}\mathbf{x}^+$	$-H_n^{\mathrm{II}}\mathbf{x}^-$

5.2. Accuracy

We examine the accuracy of the proposed inversion method for ill-conditioned block matrices, appearing in the real-world application concerning the numerical solution of electromagnetic scattering problems by layered objects with the Method of Auxiliary Sources (MAS) [9]- [10]. Here, m is related to the number of auxiliary surfaces utilized, and n to the number of auxiliary

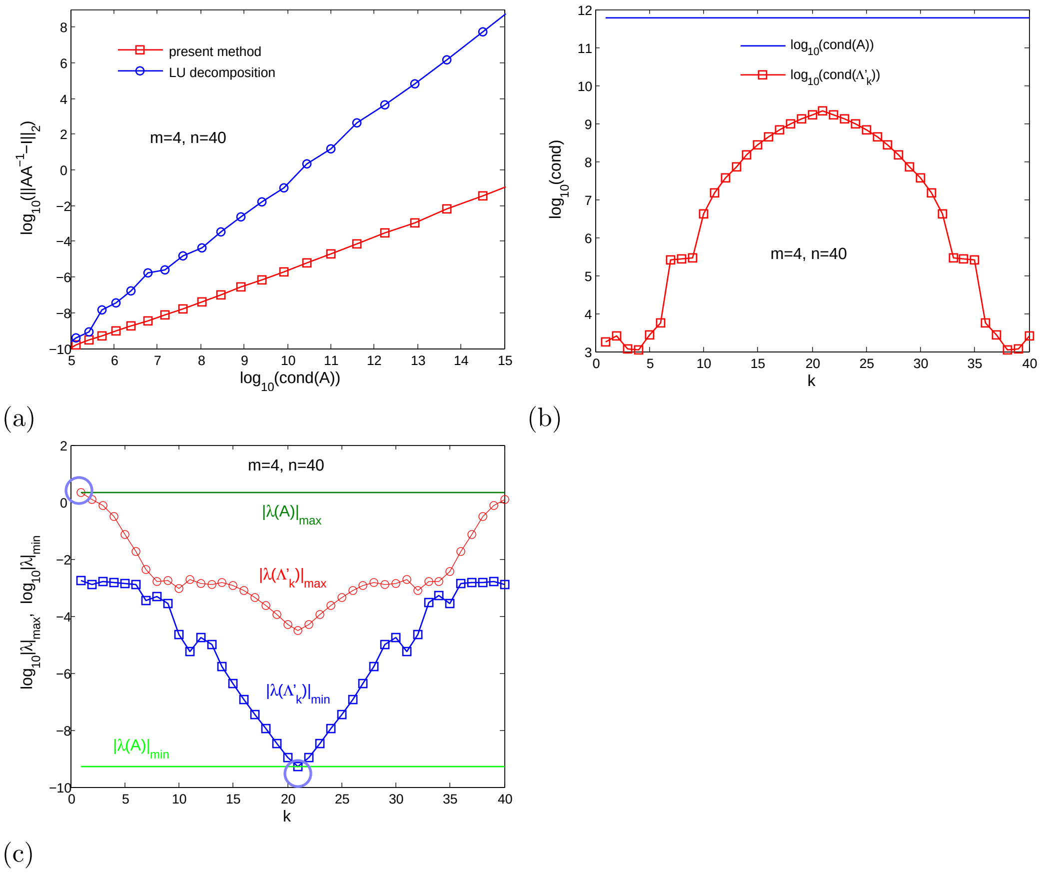


Fig. 5.5. Same as in Fig. 5.4, except for m=4, n=40.

sources considered in each auxiliary surface. Figs. 5.4a, 5.4b, and 5.4c depict respectively: (a) the inversion error $\|AA^{-1} - I\|_2$ as a function of the condition number of a matrix A with circulant blocks with m=2, n=50, as obtained by the present method and the LU decomposition, (b) the condition number of the matrices Λ'_k appearing in the similarity transformation algorithm of Section 3 for a specific choice of an ill-conditioned matrix A, and (c) the maximum and minimum eigenvalues of A and Λ'_k. Moreover, Fig. 5.5 depicts the quantities mentioned in Fig. 5.4 but for m=4, n=40. The circles in Figs. 5.4c and 5.5c indicate the blocks Λ'_k containing the maximum and minimum eigenvalues of the initial matrix A.

By the statements of Figs. 5.4a and 5.5a, the achieved inversion error for ill-conditioned matrices A is significantly reduced by the application of the developed method compared with that of the LU decomposition, applied to A. This fact is justified numerically in Figs. 5.4b and 5.5b by the decrease of the conditions numbers of the blocks Λ'_k, which have to be inverted by the present algorithm. Hence, the problem of inverting an ill-conditioned matrix A ($\mathrm{cond}(A) = 10^{13}$ for m=2, n=50 and $\mathrm{cond}(A) = 6 \cdot 10^{11}$ for m=4, n=40) is decomposed to that of inverting the n matrices Λ'_k, which are no longer ill-conditioned ($\mathrm{cond}(\Lambda'_k) \leq 10^{2.06}$ for m=2, n=50 and $\mathrm{cond}(\Lambda'_k) \leq 10^{9.34}$ for m=4, n=40). This decrease of the condition numbers of matrices Λ'_k is further justified by the *shift of the eigenvalues* of matrix A to the eigenvalues of the blocks Λ'_k. More precisely, we observe from Figs. 5.4c and 5.5c that the maximum and minimum eigenvalues of A are shifted to the blocks Λ'_2 and Λ'_{26} for m=2, n=50 and to Λ'_1 and Λ'_{21} for m=4, n=40 respectively.

Thus, the utilization of the developed inversion method may overcome the numerical instability problems, arising in the MAS, and hence enlarge its applicability range.

Appendix

The Matlab function *inv_unit_diagon_blocks* implements the algorithmic inversion of Section 3. This function calls the functions *dtt* and *idtt*, which implement the DTT's and the Inverse DTT's. The function *eigenmatrix* appearing in *dtt* and *idtt* generates the 16 Fourier, Hartley, Cosine, and Sine matrices, described in (2.2)-(2.5).

```
function [inv_A]=inv_unit_diagon_blocks(A,m,U,type)
% Input:  A = mxm complex block-matrix with nxn U-diagonalizable blocks
%         m = number of blocks in one block-row of  A
%         U = 'F' (Fourier), 'H' (Hartley), 'C' (Cosine), 'S' (Sine)
%         type = transformation type: 1 (I), 2 (II), 3 (III), 4 (IV)
% Output: inv_A = the inverse of A
n=length(A)/m;
% Block-Diagonalization of A: creation of Lambda by using the DTT's
for i=1:m
    for j=1:m
        A_block=A((i-1)*n+1:i*n,(j-1)*n+1:j*n);
        Lambda((i-1)*n+1:i*n,(j-1)*n+1:j*n)=dtt(A_block,n,U,type);
    end
end
% Similarity Transformation: Lambda_p=P*Lambda*P';
p=1; p=horzcat(p,[1:m-1]*n+1); p_new=p; for j=2:n,  p_new=p_new+1;
p=horzcat(p,p_new); end Lambda_p=Lambda(p,:);
Lambda_p=Lambda_p(:,p);
% Computation of the inverse inv_Lambda_p of Lambda_p;
for i=1:n
    Lambda_p_block=Lambda_p((i-1)*m+1:i*m,(i-1)*m+1:i*m);
    inv_Lambda_p((i-1)*m+1:i*m,(i-1)*m+1:i*m)=inv(Lambda_p_block);
end
% Inverse Similarity Transformation: inv_Lambda=P'*inv_Lambda_p*P;
p=1; p=horzcat(p,[1:n-1]*m+1); p_new=p; for j=2:m,  p_new=p_new+1;
p=horzcat(p,p_new); end inv_Lambda=inv_Lambda_p(p,:);
inv_Lambda=inv_Lambda(:,p);
% Block-Diagonalization of inv_A by imposing the IDTT's in inv_Lambda
for i=1:m
    for j=1:m
        inv_Lambda_block=inv_Lambda((i-1)*n+1:i*n,(j-1)*n+1:j*n);
        if U=='F'
            inv_A_vector=idtt(inv_Lambda_block,n,U,type);
            inv_A_block=gallery('circul',inv_A_vector);
            if (type==2) | (type==4)
```

```
                inv_A_block=triu(inv_A_block)-tril(inv_A_block,-1);
            end
        else if (U=='H') | (U=='C') | (U=='S')
                 inv_A_block=idtt(inv_Lambda_block,n,U,type);
            end
        end
        inv_A((i-1)*n+1:i*n,(j-1)*n+1:j*n)=inv_A_block;
    end
end
end
function out=dtt(X,n,U,type)
if U=='F'
    if (type==1) | (type==3)
        T=eigenmatrix(n,U,1); out=sqrt(n)*diag(T*X(1,:).');
    end
    if (type==2) | (type==4)
        T=eigenmatrix(n,U,1);
        W=diag(exp(-i*pi*[0:n-1]/n)); Wvec=conj(W(1,:)); WXvec=Wvec*X;
        help=WXvec*diag(exp(-i*pi*[0:n-1]/n)); out=sqrt(n)*diag(T*help.');
    end
end
if (U=='H') | (U=='C') | (U=='S'), T=eigenmatrix(n,U,type); out=T'*X*T; end
end
function out=idtt(X,n,U,type)
T=eigenmatrix(n,U,type);
if U=='F',  Tvec=T(1,:); TXvec=Tvec*X; out=TXvec*T';
else if (U=='H') | (U=='C') | (U=='S'), out=T*X*T'; end
end
end
```

Acknowledgment. The work of N. L. Tsitsas was supported by the State Scholarships Foundation, while he was a post-doctoral research scholar.

References

[1] H. Ammari, An Introduction to Mathematics of Emerging Biomedical Imaging, Mathematics and Applications, vol. 62, Springer-Verlag, Berlin, 2008.

[2] H. Ammari, An inverse initial boundary value problem for the wave equation in the presence of imperfections of small volume, *SIAM J. Control Optim.*, **41** (2002), 1194-1211.

[3] H. Ammari and, H. Kang, Reconstruction of Elastic Inclusions of Small Volume via Dynamic Measurements, *Appl. Math. Opt.*, **54** (2006), 223-235.

[4] J.W. Cooley and J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comput.*, **19** (1965), 297-301.

[5] R.N. Bracewell, The Hartley Transform, Oxford University Press, New York, 1986.

[6] V. Britanak, P. Yip, and K.R. Rao, Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations, Academic Press, 2006.

[7] H.T. Anastassiu, Fast, simple and accurate computation of the currents on an arbitrarily large circular loop antenna, *IEEE T. Antenn. Propag.*, **54** (2006), 860-866.

[8] P.J. Papakanellos, N.L. Tsitsas, and H.T. Anastassiu, Efficient Modeling of Radiation and Scattering for a Large Array of Loops, *IEEE T. Antenn. Propag.*, **58** (2010), 999-1002.

[9] N.L. Tsitsas, E.G. Alivizatos, H.T. Anastassiu, D.I. Kaklamani, Optimization of the method of auxiliary sources (MAS) for scattering by an infinite cylinder under oblique incidence, *Electromagnetics*, **25** (2005), 39-54.

[10] N.L. Tsitsas, E.G. Alivizatos, H.T. Anastassiu and D.I. Kaklamani, Optimization of the method of auxiliary sources (MAS) for oblique incidence scattering by an infinite dielectric cylinder, *Electr. Eng. (Archiv für Electrotechnik),* **89** (2007), 353-361.

[11] S. Rjasanow, Effective algorithms with circulant-block matrices, *Linear Algebra Appl.*, **202** (1994), 55-69.

[12] S.E. Reichenbach, J.C. Burton, and K.W. Miller, Comparison of algorithms for computing the two-dimensional discrete Hartley transform, *J. Opt. Soc. Am. A*, **6** (1989), 818-822.

[13] G. Mandyam, N. Ahmed, and N. Magotra, Lossless Image Compression Using the Discrete Cosine Transform, *J. Vis. Commun. Image Representation.*, **8** (1997), 21-26.

[14] G. Strang, A Proposal for Toeplitz Matrix Calculations, *Stud. Appl. Math.*, **74** (1986), 171-176.

[15] G. Dassios, M. Hadjinicolaou, G. Kamvyssas, A.N. Kandili, On the polarizability potential for two spheres, *Int. J. Eng. Sci.*, **44** (2006), 1520-1533.

[16] A. Arico, S. Serra-Capizzano, and M. Tasche, Fast and Numerically Stable Algorithms for Discrete Hartley Transforms and Applications to Preconditioning, *Commun. in Information and Systems*, **5** (2005), 21-68.

[17] T. Apostolopoulos, G. Dassios, A parallel algorithm for solving the inverse scattering moment problem, *J. Comput. Appl. Math.*, **42** (1992), 63-77.

[18] M. Püschel and J.M.F. Moura, The Algebraic Approach to the Discrete Cosine and Sine Transforms and their Fast Algorithms, *SIAM J. Comput.*, **32** (2003), 1280-1316.

[19] N.-C. Hu, H.-I. Chang, O.K. Ersoy, Generalized discrete Hartley transforms, *IEEE Trans. Signal Processing*, **40** (1992), 2931-2940.

[20] P.J. Davis, *Circulant Matrices*, Chelsea Publishing, N.Y., 1994.

[21] N.L. Tsitsas, E.G. Alivizatos, and G.H. Kalogeropoulos, A recursive algorithm for the inversion of matrices with circulant blocks, *Appl. Math. Comput.*, **188** (2007), 877-894.