

## Computational algorithms for determinant evaluation of cyclic $m$ -diagonal Toeplitz matrices

Maryam Shams Solary \*

Ensieh Sadeghy <sup>†</sup>

*Department of Mathematics*

*Payame Noor University*

*P. O. Box 19395-3697*

*Tehran*

*Iran*

---

### Abstract

Cyclic  $m$ -diagonal matrices possessing a Toeplitz structure have been extensively studied in the contemporary literature. In this work, we propose two efficient numerical procedures for calculating the determinant of general cyclic  $m$ -diagonal matrices possessing a Toeplitz structure. The first procedure is based on converting the matrix into a 2-by-2 block matrix format, subsequently leveraging the inverse of this reformulated matrix. The second technique is based on a conversion to an upper triangular block matrix. The performance of these new algorithms is then benchmarked against established methods from prior research.

---

**Subject Classification (2020):** 15B05, 65F40, 15A15.

**Keywords:** Determinant, Block matrix, Cyclic,  $m$ -diagonal, Toeplitz matrix

### 1. Introduction

In this study, we focus on the cyclic  $m$ -diagonal Toeplitz matrix of the form

---

\* ORCID-ID: 0000-0002-8911-3903

---

\* E-mail: [shamssolary@pnu.ac.ir](mailto:shamssolary@pnu.ac.ir); (Corresponding Author)  
[shamssolary@gmail.com](mailto:shamssolary@gmail.com)

<sup>†</sup> E-mail: [sara320s@yahoo.com](mailto:sara320s@yahoo.com)

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_x & & & b_y & \cdots & b_3 & b_2 & b_1 \\ b_1 & a_1 & a_2 & \cdots & a_x & & & b_y & \cdots & b_3 & b_2 \\ b_2 & b_1 & a_1 & \cdots & \cdots & a_x & & & b_y & \cdots & b_3 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & a_x & & & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & a_x & & & b_y \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ b_y & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & a_x \\ a_x & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_4 & \cdots & a_x & & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_3 & a_4 & \cdots & a_x & & & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_2 & a_3 & a_4 & \cdots & a_x & & b_y & \cdots & b_2 & b_1 & a_1 \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (1)$$

where  $x = \left(\frac{m-1}{2}\right) + 1, y = \left(\frac{m-1}{2}\right)$ , we assume that  $a_i, b_j \in \mathbb{R}$  and  $a_x \neq 0$ ,  $n \geq 9$ . We undertook this work because  $m$ -diagonal matrices and cyclic  $m$ -diagonal matrices are frequently used in several areas of mathematics and various branches of numerical analysis, such as spline approximation, boundary value problems, and linear dynamical systems, [1-4].

Building on the algorithms for cyclic pentadiagonal and heptadiagonal Toeplitz matrices established in [5-7], this work generalizes these methods to the broader class of cyclic  $m$ -diagonal Toeplitz matrices.

We present a generalization of the determinant-finding process for cyclic  $m$ -diagonal Toeplitz matrices via two efficient computational algorithms. Some comparison tables have been designed in each case to compare the implementation of the algorithms with each other and also with the results of Matlab software. Here, the superscript  $T$  marked the transpose operation. Recent studies on related structured matrices, such as quasi anti-tridiagonal and block tridiagonal matrices [8, 9], further highlight the importance of this class, yet the generalization of efficient algorithms from the specific cases of cyclic pentadiagonal ( $m = 5$ ) and heptadiagonal ( $m = 7$ ) Toeplitz matrices to the general cyclic  $m$ -diagonal case remains a pertinent problem, which this paper addresses.

## 2. Main Results

We commence this section by extending the findings of [5-7] to the evaluation of the determinant for cyclic  $m$ -diagonal matrices possessing a Toeplitz structure, specifically for

$$m = 9, 11, 13, \dots, 2k + 1, \quad k \in \mathbb{N},$$

### 2.1 A Determinant Evaluation Algorithm for Cyclic $m$ -diagonal Matrices

Let the column vectors of matrix  $A$ , defined in equation (1), be denoted by  $\{C_1, C_2, C_3, \dots, C_n\}$ . Now, consider a matrix  $S$  of the form

is achieved by repositioning the last  $\frac{m-1}{2}$  columns to become the first  $\frac{m-1}{2}$  columns. This transformation yields an approximately upper triangular Toeplitz matrix,  $S$ , whose column vectors exhibit a structured pattern of non-zero elements  $\{C_{n-(\frac{m-1}{2}-1)}, C_{n-(\frac{m-1}{2}-2)}, \dots, C_n, C_1, \dots, C_{(\frac{m-1}{2})}\}$ .

if  $\binom{m-1}{2} = 2k$  then

and if  $\binom{m-1}{2} = 2k - 1$  then

when  $k = 1, 2, 3, \dots$

$$L = \begin{bmatrix} D & 0 \\ S & E \end{bmatrix} \in \mathbb{R}^{(n+m-1) \times (n+m-1)}, \quad (4)$$

where

$$D = \begin{bmatrix} a_x & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \vdots & \ddots & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ a_2 & \dots & a_x & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ a_1 & a_2 & \dots & a_x & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ b_1 & a_1 & a_2 & \dots & a_x & 0 & \dots & \dots & \dots & \dots & 0 \\ b_2 & b_1 & a_1 & a_2 & \dots & a_x & 0 & \dots & \dots & \dots & 0 \\ \vdots & b_2 & b_1 & a_1 & a_2 & \dots & a_x & 0 & \dots & \dots & 0 \\ b_{y-1} & \dots & b_2 & b_1 & a_1 & a_2 & \dots & a_x & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{(m-1) \times n},$$

$$E = \begin{bmatrix} 0 & \dots & 0 & a_x & \dots & a_2 & a_1 & b_1 & b_2 & \dots & b_{y-1} \\ 0 & \dots & \dots & 0 & a_x & \dots & a_2 & a_1 & b_1 & b_2 & \vdots \\ 0 & \dots & \dots & \dots & 0 & a_x & \dots & a_2 & a_1 & b_1 & b_2 \\ 0 & \dots & \dots & \dots & 0 & a_x & \dots & a_2 & a_1 & b_1 & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 & a_x & \dots & a_2 & a_1 & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & a_x & \dots & a_2 & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 & a_x & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 & a_x & \vdots \end{bmatrix}^T \in \mathbb{R}^{n \times (m-1)}.$$

The matrix  $S$ , previously defined as an  $n$ -by- $n$  quasi upper triangular Toeplitz matrix, will now be used. We now extend some results from [5] concerning cyclic  $m$ -diagonal matrices with Toeplitz structure.

To achieve this objective, we examine a system of linear equations of dimension  $(n + m - 1) \times (n + m - 1)$  that possesses a lower triangular Toeplitz structure, expressed as

$$L\eta = e_1,$$

where  $e_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^{(n+m-1) \times 1}$  and  $L$  is an  $(n + m - 1)$ -by- $(n + m - 1)$  matrix thus forms a lower triangular matrix as described in equation (4). Solving these equations leads to the following recurrence formula:

$$\eta_i = \begin{cases} \frac{1}{a_x} & \text{if } i = 1 \\ -\frac{a_{x-1}}{a_x} \eta_1 & \text{if } i = 2 \\ -\frac{a_{x-1}}{a_x} \eta_2 + \frac{a_{x-2}}{a_x} \eta_1 & \text{if } i = 3 \\ -\frac{a_{x-1}}{a_x} \eta_3 + \frac{a_{x-2}}{a_x} \eta_2 + \frac{a_{x-3}}{a_x} \eta_1 & \text{if } i = 4 \\ \vdots & \vdots \\ -\frac{a_{x-1}}{a_x} \eta_{m-2} + \frac{a_{x-2}}{a_x} \eta_{m-3} + \cdots + \frac{a_1}{a_x} \eta_{\frac{m-1}{2}} + \frac{b_1}{a_x} \eta_x + \cdots + \frac{b_{y-1}}{a_x} \eta_1 & \text{if } i = m-1 \\ -\frac{a_{x-1}}{a_x} \eta_{i-1} + \frac{a_{x-2}}{a_x} \eta_{i-2} + \frac{a_{x-3}}{a_x} \eta_{i-3} + \cdots + \frac{a_1}{a_x} \eta_{i-\frac{m-1}{2}} + \frac{b_1}{a_x} \eta_{i+x} + \frac{b_2}{a_x} \eta_{i+(x+1)} + \cdots + \frac{b_y}{a_x} \eta_{i-(m-1)} & \text{if } i = m, \dots, n \\ -\frac{a_{x-1}}{a_x} \eta_n + \frac{a_{x-2}}{a_x} \eta_{n-1} + \cdots + \frac{a_1}{a_x} \eta_{n-\frac{m-1}{2}-1} + \frac{b_1}{a_x} \eta_{n-\frac{m-1}{2}} + \cdots + \frac{b_y}{a_x} \eta_{n-(m-2)} + a_x \eta_1 & \text{if } i = n+1 \\ -\frac{a_{x-1}}{a_x} \eta_{n+1} + \frac{a_{x-2}}{a_x} \eta_n + \cdots + \frac{a_1}{a_x} \eta_{n-\frac{m-1}{2}-2} + \frac{b_1}{a_x} \eta_{n-\frac{m-1}{2}-1} + \cdots + \frac{b_y}{a_x} \eta_{n-(m-3)} + a_x \eta_2 + \frac{a_{x-1}}{a_x} \eta_1 & \text{if } i = n+2 \\ -\frac{a_{x-1}}{a_x} \eta_{n+2} + \frac{a_{x-2}}{a_x} \eta_{n+1} + \cdots + \frac{a_1}{a_x} \eta_{n-\frac{m-1}{2}-3} + \frac{b_1}{a_x} \eta_{n-\frac{m-1}{2}-2} + \cdots + \frac{b_y}{a_x} \eta_{n-(m-4)} + a_x \eta_3 + \frac{a_{x-1}}{a_x} \eta_2 + \frac{a_{x-2}}{a_x} \eta_1 & \text{if } i = n+3 \\ \vdots & \vdots \\ -\frac{a_{x-1}}{a_x} \eta_{n+(m-2)} + \frac{a_{x-2}}{a_x} \eta_{n+(m-3)} + \cdots + \frac{b_y}{a_x} \eta_n + a_x \eta_{m-1} + \eta_{m-1} & \text{if } i = n+(m-1) \end{cases}$$

by using the forward substitution. All cases of  $\eta_i$  will be merged. Subsequently the following theorem is deduced:

**Theorem 1:** Consider the lower triangular matrix  $L$  from (4) of dimension  $(n+m-1)$ -by- $(n+m-1)$ . Its inverse,  $L^{-1}$ , has a first column represented as. Then

$$\eta_i = -\eta_{i-n} - \frac{1}{a_x} \sum_{j=i}^{i+m-2} r_{j-i}(\eta_{j-m-1} + \eta_{j-n-m-1}), \text{ for } i = 2, 3, \dots, n+m-1, \quad (5)$$

where

$$r_0 = b_y, r_1 = b_{y-1}, r_2 = b_{y-2}, \dots, r_{m-2} = a_{x-1}, \eta_1 = \frac{1}{a_x}, \eta_k = 0$$

for all  $k \leq 0$ .

Let  $L^{-1}$  be partitioned in the block form

$$L^{-1} = \begin{bmatrix} U & Q \\ P & V \end{bmatrix},$$

where  $U$ ,  $Q$ ,  $P$  and  $V$  are matrices of dimensions  $n \times (m-1)$ ,  $n \times n$ ,  $(m-1) \times (m-1)$  and  $(m-1) \times n$ , respectively. Applying Lemma 2.2 of [5] and Theorem 1 yields

$$P = \begin{bmatrix} \eta_{n+1} & \eta_n & \eta_{n-1} & \cdots & \eta_{n-m-3} \\ \eta_{n+2} & \eta_{n+1} & \eta_n & \cdots & \eta_{n-m-4} \\ \eta_{n+3} & \eta_{n+2} & \eta_{n+1} & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_{n+m-1} & \eta_{n+m-2} & \eta_{n+m-3} & \cdots & \eta_{n+1} \end{bmatrix}.$$

Thus,  $p_{ij} = \eta_{n+i-j+1}$  for  $1 \leq i, j \leq (m-1)$ . From Theorem 2.1 of [5] and relations (2), (3) in Section 2.1, we obtain that:

if  $\left(\frac{m-1}{2}\right) = 2k$  then

$$\det(A) = \det(S) = \det(L.P) = (a_x)^{n+m-1} \cdot \det(P)$$

and if  $\left(\frac{m-1}{2}\right) = 2k-1$  then

$$\det(A) = \begin{cases} \det(S) = \det(L.P) = (a_x)^{n+m-1} \cdot \det(P) & \text{if } n = 2k+1, \\ -\det(S) = -\det(L.P) = -(a_x)^{n+m-1} \cdot \det(P) & \text{if } n = 2k, \end{cases} \quad (6)$$

since  $\det(L) = a_x^{n+m-1}$ . Thus,  $\det(A)$  can be determined through the evaluation of  $\det(P)$ . Applying the Leibniz formula,

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n a_{\sigma(i), i}, \quad (7)$$

The  $\det(P)$  is shown by the following expression:

$$\det(P) = \sum_{\sigma \in S_{m-1}} \text{sgn}(\sigma) \prod_{i=1}^{m-1} P_{\sigma(i), i} = \sum_{\sigma \in S_{m-1}} \text{sgn}(\sigma) \prod_{i=1}^{m-1} \eta_{n+\sigma(i)-i+1} \quad (8)$$

The results established herein apply to arbitrary complex numbers

$$a_1, a_2, \dots, a_x, b_1, b_2, \dots, b_y \in \mathbb{C}.$$

The determinant of matrix  $A$  in (1) is evaluated by applying the following results:

**Theorem 8:** Consider an  $n$ -by- $n$  cyclic  $m$ -diagonal Toeplitz matrix  $A$ , and let  $L$  be the associated lower triangular Toeplitz matrix given in (4). Let  $\eta$  be the column vector explained in Theorem 1. Then

$$\det(A) = a_x^{n+m-1} \cdot \sum_{\sigma \in S_{m-1}} \text{sgn}(\sigma) \prod_{i=1}^{m-1} \eta_{n+\sigma(i)-i+1}.$$

where  $\text{sgn}(\cdot)$  is the sign function of a permutation in the permutation group  $S_{m-1}$ .  $W$

## 2.2 Computing the Determinant of Cyclic $m$ -diagonal Matrices by a Numerical Algorithm

In this section, we present an improved and efficient determinant computation for cyclic  $m$ -diagonal Toeplitz matrices, which extends the methodology developed earlier. The algorithm is based on an explicit formula derived in Section 2.1. Note that the computation of  $\eta_i$  in Theorem 2 involves repeated division by the constant  $a_x$ . Therefore, let

$$a'_1 = \frac{a_1}{a_x}, a'_2 = \frac{a_2}{a_x}, \dots, a'_x = 1, b'_1 = \frac{b_1}{a_x}, b'_2 = \frac{b_2}{a_x}, \dots, b'_y = \frac{b_y}{a_x}$$

to reduce the operation count. If we substitute the modified entries into the definition of matrix  $A$  (1), resulting in a modified matrix  $A'$ .

This leads to the relation

$$\det(A) = a_x^n \cdot \det(A')$$

and a corresponding lower triangular Toeplitz matrix  $L'$ , defined analogously to  $L$  in (4) but using the updated elements  $a'_1, a'_2, \dots, a'_x, b'_1, b'_2, \dots, b'_y$ . Applying Theorem 1 to the system  $L'\eta' = e_1$  yields

$$\eta'_i = -\eta'_{i-n} - \sum_{j=i}^{i+m-2} r'_{j-i} (\eta'_{j-m-1} + \eta'_{j-n-m-1}), \text{ for } i = 2, 3, \dots, n+m-1, \quad (9)$$

where  $r'_0 = b'_y, r'_1 = b'_{y-1}, \dots, r'_{m-2} = a'_{x-1}, \eta'_1 = 1$  and  $\eta'_k = 0$  for all  $k \leq 0$ . Following the same approach, applying Theorem 2.1 of [5] in conjunction with relation (3) from Section 2.1, yields

$$\det(A') = \det(L' \cdot P'),$$

with  $\det(L') = 1$ , and

$$P' = \begin{bmatrix} \eta'_{n+1} & \eta'_n & \cdots & \eta'_{n-m-3} \\ \eta'_{n+2} & \eta'_{n+1} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \eta'_{n+m-1} & \eta'_{n+m-1} & \cdots & \eta'_{n+1} \end{bmatrix} \quad (10)$$

then:

$$\det(A) = (a_x)^{n+m-1} \cdot \det(A') = (a_x)^{n+m-1} \cdot \det(P')$$

The resulting algorithm can be constructed as follows:

**Algorithm DCMT.1****Step 1.**

Provide as input the sequences:  $a_1, a_2, \dots, a_x, b_1, b_2, \dots, b_y$ , and the integers  $n$  and  $m$ .

**Step 2.**

Evaluate the vector:  $\eta'_1, \eta'_2, \dots, \eta'_{n+m-1}$  using equation (9) and construct the matrix  $P'$  according to (10).

**Step 3.**

Compute:  $\det(P')$  by applying the formula given in (8).

**Step 4.**

Output: if  $\left(\frac{m-1}{2}\right) = 2k$

$$\det(A) = (a_x)^{n+m-1} \cdot \det(P')$$

if  $\left(\frac{m-1}{2}\right) = 2k - 1$  then

$$\det(A) = \begin{cases} (a_x)^{n+m-1} \cdot \det(P') & \text{if } n = 2k + 1, \\ -(a_x)^{n+m-1} \cdot \det(P') & \text{if } n = 2k, \end{cases}$$

### 2.3 On Determinant Computation Using Block Upper Triangular Transformations

Let  $\{C_1, C_2, C_3, \dots, C_n\}$  denote the columns of matrix  $A$  as defined in (1). By exchanging the last  $\left(\frac{m-1}{2}\right)$  columns by swapping columns to position them as the initial columns, we get a Toeplitz matrix  $T$  with an approximately lower triangular form. Its column vectors are reordered as

$$\{C_{\left(\frac{m-1}{2}+1\right)}, C_{\left(\frac{m-1}{2}+2\right)}, \dots, C_n, C_1, C_2, C_{\left(\frac{m-1}{2}\right)}\},$$

forming the following structure:

$$T = \begin{bmatrix} a_x & & & b_y & b_{y-1} & \cdots & b_2 & b_1 & a_1 & a_2 & \cdots \\ \vdots & a_x & & & b_y & b_{y-1} & \cdots & b_2 & b_1 & a_1 & a_2 \\ a_2 & \cdots & a_x & & & b_y & b_{y-1} & \cdots & b_2 & b_1 & a_1 \\ a_1 & a_2 & \cdots & a_x & & & b_y & b_{y-1} & \cdots & b_2 & b_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & b_y & b_{y-1} & \cdots & b_2 \\ b_2 & \ddots & \ddots & \ddots & \ddots & & & & b_y & b_{y-1} & \vdots \\ \vdots & b_2 & b_1 & a_1 & a_2 & \cdots & a_x & & & b_y & b_{y-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & b_y \\ b_y & b_{y-1} & \cdots & b_2 & b_1 & a_1 & a_2 & \cdots & a_x & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & b_y & b_{y-1} & \cdots & b_2 & b_1 & a_1 & a_2 & \cdots & a_x \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (11)$$

now we prepared  $\left(\frac{m-1}{2}\right)$  column exchanges in the matrix reordering above, then we deduce the following relation:

if  $\left(\frac{m-1}{2}\right) = 2k$  then

$$\det(A) = \det(T)$$

and if  $\left(\frac{m-1}{2}\right) = 2k - 1$  we have

$$\det(A) = \begin{cases} \det(T) & \text{if } n = 2k + 1 \\ -\det(T) & \text{if } n = 2k \end{cases} \quad (12)$$

for  $k = 1, 2, 3, \dots$

The matrix  $T$  from (11) can therefore be transformed into the form of a 2-by-2 block upper triangular matrix:

$$U = \begin{bmatrix} a_x \cdot I_{n-m-1} & U_{12} \\ 0 & U_{22} \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (13)$$

with  $U_{12}$  and  $U_{22}$  of sizes  $(n - m - 1) \times (m - 1)$  and  $(m - 1) \times (m - 1)$ , respectively. This matrix  $U$  can be derived from the original matrix  $T$  by selecting appropriate entries  $\omega_{i+1,i}, \omega_{i+2,i}, \omega_{i+3,i}, \omega_{i+4,i}, \dots, \omega_{i+m-1,i}$  in the following transformation matrices:

$$\Omega_1 = \begin{bmatrix} 1 & & & & & & & & \\ \omega_{21} & 1 & & & & & & & \\ \omega_{31} & & 1 & & & & & & \\ \omega_{41} & & & 1 & & & & & \\ \omega_{51} & & & & 1 & & & & \\ \vdots & & & & & 1 & & & \\ \omega_{(m-1)1} & & & & & & 1 & & \\ \omega_{m1} & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{bmatrix},$$

$$\Omega_i = \begin{bmatrix} I_{i-1} & & & & & & & & \\ & 1 & & & & & & & \\ & \omega_{i+1,i} & 1 & & & & & & \\ & \omega_{i+2,i} & & 1 & & & & & \\ & \omega_{i+3,i} & & & 1 & & & & \\ & \omega_{i+4,i} & & & & 1 & & & \\ & \vdots & & & & & 1 & & \\ & \omega_{i+m-1,i} & & & & & & 1 & \\ & & & & & & & & I_{n-i-m-1} \end{bmatrix},$$

This holds for  $i = 2, 3, \dots, n - m - 1$ , where  $I_{i-1}$  and  $I_{n-i-m-1}$  are identity matrices of order  $i - 1$  and  $n - i - m - 1$ , respectively. In general, the coefficients  $\omega_{i+1,i}, \omega_{i+2,i}, \omega_{i+3,i}, \omega_{i+4,i}, \dots, \omega_{i+m-1,i}$  are defined by:

$$\omega_{i+1,i} = -\frac{a_{x-1}}{a_x}, \omega_{i+2,i} = -\frac{a_{x-2}}{a_x}, \omega_{i+3,i} = -\frac{a_{x-3}}{a_x},$$

$$\omega_{i+4,i} = -\frac{a_{x-4}}{a_x}, \dots, \omega_{i+m-1,i} = -\frac{b_y}{a_x},$$

for  $i = 1, 2, 3, \dots, n - m - 1$ .

Defining the composite transformation matrix  $\Omega = \Omega_{n-m-1}\Omega_{n-m-2} \dots \Omega_1$ , we obtain the relation

$$\Omega T = U,$$

where matrix  $U$  is the 2-by-2 block upper triangular defined in (13).

Since we can deduce from the definition of  $\Omega_i$  that  $\det(\Omega) = 1$ , we directly obtain the following relations:

If  $\left(\frac{m-1}{2}\right) = 2k$  then

$$\det(A) = \det(T) = \det(\Omega T) = \det(U) = a_x^{n-m-1} \cdot \det(U_{22}).$$

If  $\left(\frac{m-1}{2}\right) = 2k - 1$  we have

$$\det(A) = \begin{cases} \det(T) = \det(U) = (a_x)^{n-m-1} \cdot \det(U_{22}) & \text{if } n = 2k + 1, \\ -\det(T) = -\det(U) = -(a_x)^{n-m-1} \cdot \det(U_{22}) & \text{if } n = 2k, \end{cases} \quad (14)$$

$$k = 1, 2, 3, \dots$$

#### 2.4 Numerical Algorithm via Block Upper Triangular Transformation

To lower the operation count in the method of Subsection 2.3, we introduce the scaled constants:

$$a'_{x-1} = \frac{a_{x-1}}{a_x}, a'_{x-2} = \frac{a_{x-2}}{a_x}, \dots, a'_1 = \frac{a_1}{a_x}$$

$$b'_1 = \frac{b_1}{a_x}, b'_2 = \frac{b_2}{a_x}, \dots, b'_y = \frac{b_y}{a_x}.$$

The scaling circumvents multiple divisions by  $a_x$  when converting  $T$  into a block upper triangular form. Applying the scaled constants to the original matrix definition in (1) results in the transformed matrix  $A'$ . The determinants are related by

$$\det(A) = (a_x)^n \cdot \det(A'). \quad (15)$$

For the matrix  $A'$ , we apply the same procedure as for  $A$ . This leads to a corresponding nearly matrix  $T'$ , analogous to  $T$  in (11), but with the entries  $a'_1, a'_2, \dots, a'_x, b'_1, \dots, b'_y$ . Similarly, we define the transformation matrices  $\Omega'_i$  analogous to  $\Omega_i$ .  $\Omega' = \Omega'_{n-m-1}\Omega'_{n-m-2} \dots \Omega'_1$  we have

$$\Omega' T' = \begin{bmatrix} I_{n-m-1} & U'_{12} \\ O & U'_{22} \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (16)$$

where  $U'_{12}$  and  $U'_{22}$  are matrices of size  $(n-m-1)$ -by- $(m-1)$  and  $(m-1)$ -by- $(m-1)$ , respectively. In addition, based on the structure of  $\Omega'_i$  and simple matrix multiplication.

We have

(17)

if  $\binom{m-1}{2} = 2k$  then

$$\det(A) = (a_x)^n \cdot \det(A') = (a_x)^n \cdot \det(\Omega' T') = (a_x)^n \cdot \det(U'_{22})$$

and if  $\binom{m-1}{2} = 2k - 1$  we have

(18)

for  $k = 1, 2, 3, \dots$ . We obtain  $W_1, W_2, W_3$  to compute  $U'_{22}$

(19)

(20)

$$W_2 = \begin{bmatrix} 1 & & & & & & & & \\ a'_{x-1} & 1 & & & & & & & \\ \vdots & a'_{x-1} & 1 & & & & & & \\ a'_2 & \vdots & a'_{x-1} & \ddots & & & & & \\ a'_1 & a'_2 & \vdots & \ddots & 1 & & & & \\ b'_1 & a'_1 & a'_2 & \vdots & a'_{x-1} & 1 & & & \\ b'_2 & \ddots & \ddots & \ddots & \vdots & a'_{x-1} & 1 & & \\ \vdots & \ddots & b'_1 & a'_1 & a'_2 & \vdots & a'_{x-1} & 1 & \\ b'_y & \cdots & b'_2 & b'_1 & a'_1 & a'_2 & \cdots & a'_{x-1} & 1 \end{bmatrix} \quad (21)$$

yields

$$U'_{22} = W_1 * W_2 + W_3.$$

An algorithm for computing  $n$  th-order cyclic  $m$  -diagonal Toeplitz determinants is now described.

#### Algorithm DCMT.2

##### Step 1.

Input the parameters  $a_1, a_2, \dots, a_x$ ,  $b_1, b_2, \dots, b_y$ , and the integers  $n$  and  $m$ .

##### Step 2. Initialise:

$$\omega_1^{(1)} = -\frac{a_{x-1}}{a_x}, \omega_2^{(1)} = -\frac{a_{x-2}}{a_x}, \dots, \omega_{m-1}^{(1)} = -\frac{b_y}{a_x}.$$

**For**  $i = 2, 3, \dots, n - (m - 1)$  **Compute**

$$\begin{aligned} \omega_1^{(i)} &= \omega_2^{(i-1)} + \omega_1^{(1)} * \omega_1^{(i-1)}, \\ \omega_2^{(i)} &= \omega_3^{(i-1)} + \omega_2^{(1)} * \omega_1^{(i-1)}, \\ \omega_3^{(i)} &= \omega_4^{(i-1)} + \omega_3^{(1)} * \omega_1^{(i-1)}, \\ \omega_4^{(i)} &= \omega_5^{(i-1)} + \omega_4^{(1)} * \omega_1^{(i-1)}, \\ &\vdots \\ \omega_{m-1}^{(i)} &= \omega_{m-1}^{(1)} * \omega_1^{(i-1)} \end{aligned}$$

End.

**Step 3. Construct the  $(m - 1)$ -by- $(m - 1)$  matrices in (19), (20), (21)**

$$U'_{22} = W_1 * W_2 + W_3.$$

##### Step 4.

Output:

if  $(\frac{m-1}{2}) = 2k - 1$  we have

$$\det(A) = \begin{cases} (a_x)^n \cdot \det(U'_{22}) & \text{if } n = 2k + 1, \\ -(a_x)^n \cdot \det(U'_{22}) & \text{if } n = 2k \end{cases} \text{ and } k = 1, 2, 3, 4, \dots$$

if  $(\frac{m-1}{2}) = 2k$  then  $\det(A) = (a_x)^n \cdot \det(U'_{22})$ .

### 3. Numerical Experiments

This section presents the results of numerical experiments comparing the performance of the proposed algorithms (DCMT.1 and DCMT.2) with MATLAB's built-in function  $\det()$ . All examples were conducted using

MATLAB R2024a in double-precision arithmetic. The presented work generalizes the methodologies found in references [5-7, 10]. To ensure numerical stability, the same inductive process employed in these referenced algorithms is utilized; the details are omitted here to avoid repetition.

**Example 1:** The computation starts with determining the determinant of an  $n$ -by- $n$  cyclic 9-diagonal Toeplitz matrix by:

$$A = \begin{bmatrix} 0.3 & 0.1 & 0.2 & 0.5 & 1 & & & & 0.2 & 0.8 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.1 & 0.2 & 0.5 & 1 & & & 0.2 & 0.8 & 0.5 & \\ 0.5 & 0.2 & 0.3 & 0.1 & 0.2 & 0.5 & 1 & & & 0.2 & 0.8 & \\ 0.8 & 0.5 & 0.2 & 0.3 & 0.1 & 0.2 & 0.5 & 1 & & & 0.2 & \\ 0.2 & 0.8 & 0.5 & 0.2 & 0.3 & 0.1 & 0.2 & 0.5 & 1 & & & \\ & \ddots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 & \\ 1 & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0.5 & \\ 0.5 & 1 & & & 0.2 & 0.8 & 0.5 & 0.2 & 0.3 & 0.1 & 0.2 & \\ 0.2 & 0.5 & 1 & & & 0.2 & 0.8 & 0.5 & 0.2 & 0.3 & 0.1 & \\ 0.1 & 0.2 & 0.5 & 1 & & & 0.2 & 0.8 & 0.5 & 0.2 & 0.3 & \end{bmatrix}.$$

For matrix sizes  $n = 100, 500, 1000, 2000, 3000, 5000$ , we compute the determinant using both the MATLAB built-in function and the proposed algorithms. The results are summarized in Table 1.

**Table 1**  
**Determinant computation results for Example 1**

| $n$  | DCMT.1 algorithm            | DCMT.2 algorithm            | MATLAB                      |
|------|-----------------------------|-----------------------------|-----------------------------|
| 100  | 1.020257617310737           | 1.020257617310737           | 1.020257617310737           |
| 500  | 2.299756225300392           | 2.299756225300393           | 2.299756225300393           |
| 1000 | 9.796608428903742           | 9.796608428903756           | 9.796608428903735           |
| 2000 | 25.455310646806186          | 25.455310646806208          | 25.455310646806161          |
| 3000 | 1.319072699945494e<br>+ 002 | 1.319072699945495e<br>+ 002 | 1.319072699945494e<br>+ 002 |
| 5000 | 3.672361276185642e<br>+ 003 | 3.672361276185577e<br>+ 003 | 3.672361276185603e<br>+ 003 |

The result show that our algorithms generated almost the same value as the MATLAB. Meanwhile, the mean value of the CPU times in the computation of the determinant are shown in Table 2 and then we show the results in Fig.1.

**Table 2**  
**Execution times (CPU seconds) for Example 1**

| Algorithms | $n$ | 100    | 500    | 1000   | 2000   | 3000   | 5000   |
|------------|-----|--------|--------|--------|--------|--------|--------|
| DCMT.1     |     | 0.0132 | 0.0034 | 0.0036 | 0.0053 | 0.0094 | 0.0166 |
| DCMT.2     |     | 0.0127 | 0.0022 | 0.0029 | 0.0040 | 0.0050 | 0.0100 |
| MATLAB     |     | 0.0212 | 0.0087 | 0.0098 | 0.0174 | 0.0331 | 0.0560 |

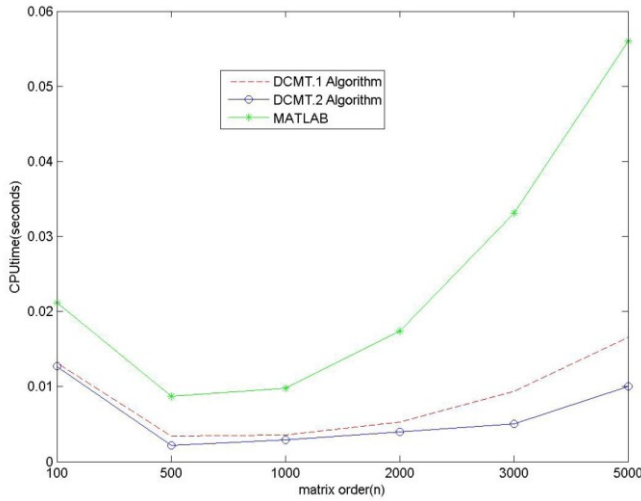


**Figure 1**  
Average CPU time (log scale) for Example 1

The determinant of an  $n$ -by- $n$  cyclic 11-diagonal Toeplitz matrix, structured as follows, is computed next:

$$A = \begin{bmatrix} 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & & 0.1 & 0.2 & 0.3 & 0.8 & 0.5 \\ 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & & 0.1 & 0.2 & 0.3 & 0.8 \\ 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 & 0.5 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ 1 & & & & & & & & & & & 1 \\ 0.5 & 1 & & & 0.1 & 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.5 & 1 & & & 0.1 & 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 & 0.2 \\ 0.2 & 0.3 & 0.5 & 1 & & & 0.1 & 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.5 & 1 & & & 0.1 & 0.2 & 0.3 & 0.8 & 0.5 & 0.3 + h^2 \end{bmatrix}$$

Here, we set  $n = \frac{1}{h}$ . For  $h = 0.01, 0.02, 0.05, 0.001, 0.002, 0.005$ , we compute the determinant using the proposed algorithms, and the results, along with those from MATLAB's built-in function, are presented in Table 3.

**Table 3**  
Determinant computation results for Example 2

| h     | DCMT.1 algorithm    | DCMT.2 algorithm    | MATLAB              |
|-------|---------------------|---------------------|---------------------|
| 0.01  | -0.607894416774315  | -0.607894416774315  | -0.607894416774315  |
| 0.02  | -1.452696702629045  | -1.452696702629049  | -1.452696702629048  |
| 0.05  | -26.632177070529409 | -26.632177070529398 | -26.632177070529380 |
| 0.001 | -1.000100152606399  | -1.000100152606399  | -1.000100152606399  |
| 0.002 | -1.002709541783609  | -1.002709541783609  | -1.002709541783609  |
| 0.005 | -1.375861160656002  | -1.375861160656004  | -1.375861160656003  |

The mean CPU time for computing the determinant is presented in Table 4, with the results visualized in Fig. 2.

**Table 4**  
**Execution times (CPU seconds) for Example 2**

| Algorithms | $h$ | 0.05   | 0.02   | 0.01   | 0.005  | 0.002  | 0.001  |
|------------|-----|--------|--------|--------|--------|--------|--------|
| DCMT.1     |     | 0.0040 | 0.0047 | 0.0048 | 0.0080 | 0.0090 | 0.0102 |
| DCMT.2     |     | 0.0032 | 0.0040 | 0.0043 | 0.0053 | 0.0059 | 0.0085 |
| MATLAB     |     | 0.0052 | 0.0075 | 0.0090 | 0.0095 | 0.0102 | 0.0189 |

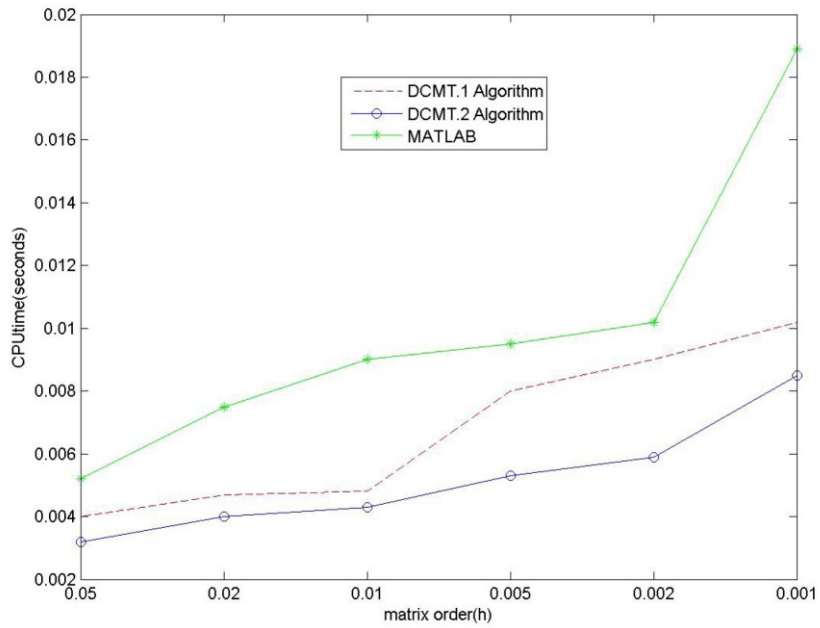


**Figure 2**  
**Average CPU time (log scale) for Example 2**

**Example 3:** Our computation proceeds to an  $n$ -by- $n$  cyclic 13-diagonal Toeplitz matrix.

$$A = \begin{bmatrix} 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 \\ 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & \\ 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & 0.1 & 0.8 & 0.5 & 0.4 & \\ 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & 0.1 & 0.8 & 0.5 & \\ 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & 0.1 & 0.8 & \\ 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & 0.1 & \\ 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & & & & & & & & & & & & 1 \\ 1 & & & & & & & & & & & & & & & 0.4 \\ 0.4 & 1 & & & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 1 & & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & \\ 0.2 & 0.3 & 0.4 & 1 & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.5 & 0.2 & \\ 0.5 & 0.2 & 0.3 & 0.4 & 1 & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.2 & \\ 0.3 & 0.5 & 0.2 & 0.3 & 0.4 & 1 & & 0.1 & 0.8 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & \end{bmatrix}.$$

Here, we set  $n = 500, 1000, 2000, 3000, 4000, 5000$ . Table 5 presents the numerical results obtained from MATLAB's built-in  $\det()$  function alongside the proposed algorithms (DCMT.1 and DCMT.2). The corresponding mean CPU times for the determinant computation are given in Table 6, and the results are illustrated in Fig. 3 for  $n = 500, 1000, 2000, 3000, 4000, 5000$ .

**Table 5**  
**Determinant computation results for Example 3**

| $n$  | DCMT.1 algorithm       | DCMT.2 algorithm       | MATLAB                 |
|------|------------------------|------------------------|------------------------|
| 500  | 3.879186583187889      | 3.879186583187878      | 3.879186583187880      |
| 1000 | 26.039995308460966     | 26.039995308460981     | 26.039995308460945     |
| 2000 | 3.351319213216030e+002 | 3.351319213216018e+002 | 3.351319213216020e+002 |
| 3000 | 6.207442548562271e+003 | 6.207442548562463e+003 | 6.207442548562406e+003 |
| 4000 | 1.166386300975372e+005 | 1.166386300975419e+005 | 1.166386300975375e+005 |
| 5000 | 2.137986642437513e+006 | 2.137986642437167e+006 | 2.137986642437664e+006 |

**Table 6**  
**Execution times (CPU seconds) for Example 3**

| Algorithms | $n$ | 500    | 1000   | 2000   | 3000   | 4000   | 5000   |
|------------|-----|--------|--------|--------|--------|--------|--------|
| DCMT.1     |     | 0.0099 | 0.0094 | 0.0113 | 0.0144 | 0.0179 | 0.0227 |
| DCMT.2     |     | 0.0050 | 0.0046 | 0.0052 | 0.0046 | 0.0045 | 0.0046 |
| MATLAB     |     | 0.0100 | 0.0125 | 0.0221 | 0.0351 | 0.0526 | 0.0915 |

In this example, we demonstrate that Algorithm DCMT.2 achieves reduced CPU times. In summary, the computational complexity of the proposed algorithms is lower than, or comparable to, that of other recent algorithms implemented in MATLAB.

The results indicate that our algorithms produce values nearly identical to those from MATLAB's built-in  $\det()$  function, while requiring less computational time. We propose the investigation of block-based structures for

computing determinants of cyclic  $m$ -diagonal matrices as a direction for future research, see [10, 11].

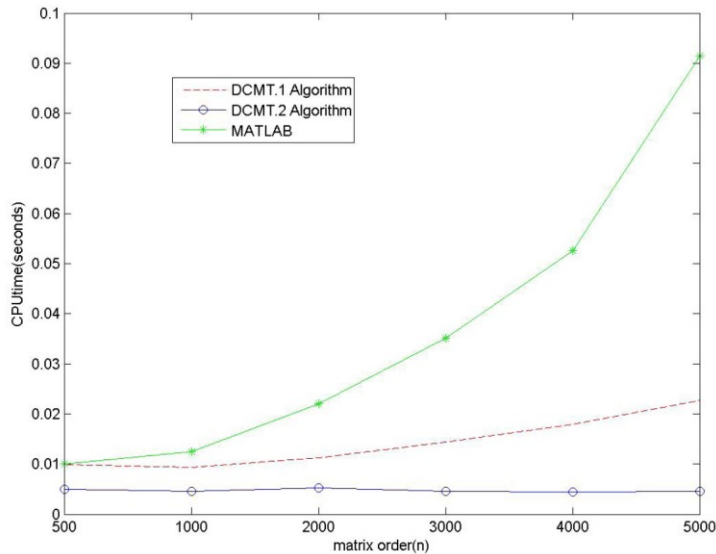


**Figure 3**  
Average CPU time (log scale) for Example 3

## References

- [1] S. Barnett, *Matrices: Methods and Applications*. Oxford, U.K.: Oxford Univ. Press (1990).
- [2] M. Dana and R. Yousefi, "Formulas for the Drazin inverse of matrices with new conditions and its applications," *Int. J. Appl. Comput. Math.*, vol. 4, no. 4, Art. no. 128 (2018).
- [3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press (1996).
- [4] I. M. Navon, "PENT: A periodic pentadiagonal systems solver," *Commun. Appl. Numer. Methods*, vol. 3, no. 1, pp. 63-69 (Jan./Feb. 1987).
- [5] J. T. Jia and S. M. Li, "A numerical algorithm for the determinant evaluation of cyclic pentadiagonal matrices with Toeplitz structure," *Numer. Algorithms*, vol. 71, no. 2, pp. 337-348 (Feb. 2016).
- [6] J. T. Jia and S. M. Li, "On determinants of cyclic pentadiagonal matrices with Toeplitz structure," *Comput. Math. Appl.*, vol. 73, no. 2, pp. 304-309 (Jan. 2017).

- [7] M. Shams Solary and E. Sadeghy, "Determinant for cyclic heptadiagonal matrices with Toeplitz structure," *Afr. Mat.*, vol. 31, pp. 407-422 (2020).
- [8] D. A. H. Ahmed, "On the characteristic polynomial and eigenvalues for block tridiagonal matrices," *J. Discrete Math. Sci. Crypt.*, vol. 25, no. 6, pp. 1745-1756 (2022).
- [9] C. M. da Fonseca, "On some quasi anti-tridiagonal matrices," *J. Discrete Math. Sci. Crypt.*, vol. 24, no. 4, pp. 1043-1052 (2021).
- [10] M. Shams Solary, "Computational properties of pentadiagonal and anti-pentadiagonal block band matrices with perturbed corners," *Soft Comput.*, vol. 24, pp. 301-309 (2020).
- [11] M. Shams Solary, "From matrix polynomial to determinant of block Toeplitz-Hessenberg matrix," *Numer. Algorithms*, vol. 94, pp. 1421-1434 (2023).

*Received July, 2025*