

A New Algorithm for Computing the Determinant and the Inverse of a Pentadiagonal Toeplitz Matrix*

Yuehui Chen

Department of Mathematics and Information Science, Zhangzhou Normal University,
Zhangzhou, China
Email: yuehuich@fjzs.edu.cn

Received January 25, 2013; revised February 26, 2013; accepted March 4, 2013

Copyright © 2013 Yuehui Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

An effective numerical algorithm for computing the determinant of a pentadiagonal Toeplitz matrix has been proposed by Xiao-Guang Lv and others [1]. The complexity of the algorithm is $(9n + 3)$. In this paper, a new algorithm with the cost of $(4n + 6)$ is presented to compute the determinant of a pentadiagonal Toeplitz matrix. The inverse of a pentadiagonal Toeplitz matrix is also considered.

Keywords: Pentadiagonal Matrix; Toeplitz Matrix; Determinant; Nonsingular; Inverse

1. Introduction

Pentadiagonal Toeplitz matrix linear systems often occur in several fields such as numerical solution of differential equations, interpolation problems, boundary value problems [1-5], etc. In these areas, the determinants and the inversions of pentadiagonal Toeplitz matrices are considered. In recent years they have become one of the most important and active research field of applied mathematics and computational mathematics increasingly.

In [2], E. Kilic, M. Ei-Mikkawy presented a fast and reliable algorithm with the cost of $(11n - 17)$ for evaluating special n th-order pentadiagonal Toeplitz determinants. In [1], X.G. Lv, T.Z. Huang, J. Le presented an algorithm with the cost of $(9n + 3)$ for calculating the determinant of a pentadiagonal Toeplitz matrix and an algorithm for calculating the inverse of a pentadiagonal Toeplitz matrix.

In this paper, we present new algorithms for computing the determinant and the inverse of an n -by- n pentadiagonal Toeplitz matrix. The complexity of the algorithms are $(4n + 6)$ and $(n^2 + 5n)$ respectively.

This paper is organized as follows: in Section 2, we present some useful notations and lemmas. In Section 3, we are going to derive new two algorithms. Finally, we give an numerical examples to show the performance of our algorithms in Section 4.

2. Notations and Preliminaries

Definition 2.1 Let $B = (b_{ij})$ be an $n \times n$ matrix. B is called persymmetric if it symmetric about its northeast-southwest diagonal, i.e., $b_{ij} = b_{n-j+1, n-i+1}$ for all i and j .

Definition 2.2 Form as

$$H = (h_{i-j})_{i,j=1}^n = \begin{pmatrix} h_0 & h_{-1} & h_{-2} & \cdots & h_{-n+1} \\ h_1 & h_0 & h_{-1} & \cdots & h_{-n+2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{n-2} & \cdots & h_1 & h_0 & h_{-1} \\ h_{n-1} & h_{n-2} & \cdots & h_1 & h_0 \end{pmatrix}$$

is called Toeplitz matrix.

Toeplitz matrices are all persymmetric matrices.

Lemma 2.1 [6] Let H be an $n \times n$ matrix. Then

(1) H is persymmetric matrix if and only if

$$J_n H^T J_n = H;$$

(2) If H is nonsingular Toeplitz matrix, H^{-1} is also a Toeplitz matrix, where J_n is the $n \times n$ exchange matrix, i.e., $J_n = [e_n, e_{n-1}, \dots, e_1]$, e_i is the i th column of identity matrix I_n of order n .

Without loss of generality, we suppose $n \geq 6$ in the paper. By computing simply, we have the following conclusion:

Lemma 2.2 Let \tilde{T} be an $(n-2) \times (n-2)$ Toeplitz matrix

*Project supported by the Natural Science Foundation of Fujian Province, China (2012D139).

(2) Compute

$$a_i = -(aa_{i-1} + ba_{i-2} + ca_{i-3} + da_{i-4}),$$

$$i = 1, 2, \dots, n+1;$$

(3) Compute $\det(\mathbf{T}) = a_n^2 - a_{n-1}a_{n+1}$.

Algorithm 3.2 (Computing \mathbf{T}^{-1})

(1) Using (3.1), calculate d_1, d_2, d_3, Δ ;

(2) Using (3.2), calculate $\tilde{\mathbf{M}} = (s_{ij})_{(n-2) \times 2}$;

(3) Using (3.4), calculate $\tilde{\mathbf{M}}_2 = (t_{ij})_{(n-2) \times 2}$;

(4) Using (3.3) and (3.5), calculate $\tilde{\mathbf{M}}_1, \tilde{\mathbf{D}}$;

(5) Calculate $\mathbf{M}^{-1} = \frac{1}{\Delta} \begin{pmatrix} \tilde{\mathbf{M}}_1 & \tilde{\mathbf{M}}_2 \\ \tilde{\mathbf{M}}_2^T \mathbf{J}_{n-2} & -\tilde{\mathbf{D}} \end{pmatrix}$;

(6) Calculate $\mathbf{T}^{-1} = \tilde{\mathbf{P}}\mathbf{M}^{-1}$.

Let us now have a look at the number of multiplications and divisions executed by Algorithm 3.1 and 3.2. For Algorithm 3.1, in Step 2, it takes about $4(n+1)$ operations. Step 3 amounts to 2 operations. On the whole, we need about $(4n+6)$ operations to compute $\det(\mathbf{T})$. Algorithm 3.1 is better than E. Kilic's algorithm [2] with the cost of $(11n-17)$ and X.G. Lv's algorithm [1] with the cost of $(9n+3)$. For Algorithm 3.2, in Step 1, it takes 4 operations. Step 2 amounts to $(n-2)$ operations. Step 3 amounts to $4(n-2)$ operations. The cost of step 4 is about n^2 , we make use of the persymmetric matrix. Therefore, we need about $(n^2 + 5n)$ operations computing \mathbf{T}^{-1} . Our algorithm is better than X.G. Lv's algorithm [1] with the cost of $\left(\frac{3}{2}n^2 + 13n\right)$.

4. An Example

Consider the pentadiagonal Toeplitz matrix as

$$\mathbf{T} = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 \end{pmatrix}.$$

By Algorithm 3.1, we have

$$a_1 = -1, a_2 = -1, a_3 = 2, a_4 = 0,$$

$$a_5 = -2, a_6 = 1, a_7 = 1.$$

So

$$\det(\mathbf{T}) = a_6^2 - a_5a_7 = 3.$$

Using Algorithm 3.2, we obtain

$$d_1 = -2, d_2 = -1, d_3 = 1, \Delta = -3,$$

$$\tilde{\mathbf{M}} = \begin{pmatrix} 1 & 2 \\ 1 & -1 \\ -2 & -2 \\ 0 & 2 \end{pmatrix}, \tilde{\mathbf{M}}_2 = \begin{pmatrix} 3 & -3 \\ 0 & 3 \\ -4 & 2 \\ 2 & -4 \end{pmatrix},$$

$$\tilde{\mathbf{M}}_2^T \mathbf{J}_4 = \begin{pmatrix} 2 & -4 & 0 & 3 \\ -4 & 2 & 3 & -3 \end{pmatrix}, \tilde{\mathbf{M}}_1 = \begin{pmatrix} 3 & 0 & -6 & 3 \\ -3 & 3 & 3 & -6 \\ -1 & -1 & 3 & 0 \\ 2 & -1 & -3 & 3 \end{pmatrix},$$

$$\mathbf{M}^{-1} = \frac{1}{3} \begin{pmatrix} -3 & 0 & 6 & -3 & -3 & 3 \\ 3 & -3 & -3 & 6 & 0 & -3 \\ 1 & 1 & -3 & 0 & 4 & -2 \\ -2 & 1 & 3 & -3 & -2 & 4 \\ -2 & 4 & 0 & -3 & 1 & 1 \\ 4 & -2 & -3 & 3 & 1 & -2 \end{pmatrix}.$$

So

$$\mathbf{T}^{-1} = \frac{1}{3} \begin{pmatrix} 4 & -2 & -3 & 3 & 1 & -2 \\ -2 & 4 & 0 & -3 & 1 & 1 \\ -3 & 0 & 6 & -3 & -3 & 3 \\ 3 & -3 & -3 & 6 & 0 & -3 \\ 1 & 1 & -3 & 0 & 4 & -2 \\ -2 & 1 & 3 & -3 & -2 & 4 \end{pmatrix}.$$

REFERENCES

- [1] X. G. Lv, T. Z. Huang and J. Le, "A Note on Computing the Inverse and the Determinant of a Pentadiagonal Toeplitz Matrix," *Applied Mathematics and Computation*, Vol. 206, No. 1, 2008, pp. 327-331. [doi:10.1016/j.amc.2008.09.006](https://doi.org/10.1016/j.amc.2008.09.006)
- [2] E. Kilic and M. Ei-Mikkawy, "A Computational Algorithm for Special nth Order Pentadiagonal Toeplitz Determinants," *Applied Mathematics and Computation*, Vol. 199, No. 2, 2008, pp. 820-822. [doi:10.1016/j.amc.2007.10.022](https://doi.org/10.1016/j.amc.2007.10.022)
- [3] J. M. McNally, "A Fast Algorithm for Solving Diagonally Dominant Symmetric Pentadiagonal Toeplitz Systems," *Journal of Computational and Applied Mathematics*, Vol. 234, No. 4, 2010, pp. 995-1005. [doi:10.1016/j.cam.2009.03.001](https://doi.org/10.1016/j.cam.2009.03.001)
- [4] S. S. Nemani, "A Fast Algorithm for Solving Toeplitz Penta-Diagonal Systems," *Applied Mathematics and Computation*, Vol. 215, No. 11, 2010, pp. 3830-3838.
- [5] Y. H. Chen and C. Y. Yu, "A New Algorithm for Computing the Inverse and the Determinant of a Hessenbert Matrix," *Applied Mathematics and Computation*, Vol. 218, 2011, pp. 4433-4436. [doi:10.1016/j.amc.2011.10.022](https://doi.org/10.1016/j.amc.2011.10.022)
- [6] G. H. Golub and C. F. Van Loan, "Matrix Computations," 3rd Edition, Johns Hopkins University Press, Baltimore and London, 1996, pp. 193-202.