Malaysian Journal of Mathematical Sciences 9(3): 349–365 (2015)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

A Fast and Reliable Algorithm for Evaluating n-th Order k-Tridiagonal Determinants

Moawwad El-Mikkawy *[1] and Faiz Atlan[2]

[1,2] *Mathematics Department, Faculty of Science, Mansoura University, Egypt*

E-mail: m_elmikkawy@yahoo.com
**Corresponding author*

ABSTRACT

The main objective of the current article is to present a fast and reliable algorithm for evaluating n-th order k-tridiagonal determinants with Toeplitz structure. Additionally, a modified algorithm for evaluating the general n-th order k-tridiagonal determinants is proposed. Numerical tests and illustrative examples are also given.

Keywords: k -Tridiagonal matrix, LU factorization, Determinant, Toeplitz matrix, MATLAB.

1. Introduction

The general tridiagonal matrix $T_n = [t_{ij}]_{i,j=1}^{n}$ in which $t_{ij} = 0$ for $|i-j| > 1$ can be written in the form:

$$T_n = \begin{bmatrix} d_1 & a_1 & 0 & \dots & 0 \\ b_1 & d_2 & a_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & b_{n-2} & d_{n-1} & a_{n-1} \\ 0 & \dots & 0 & b_{n-1} & d_n \end{bmatrix}. \quad (1)$$

Tridiagonal matrices frequently appear in a variety of applications such as parallel computing, cubic spline interpolation, telecommunication system analysis, and in numerous other fields of science and engineering. In many of these areas the evaluation of tridiagonal determinants is required. There has been a considerable amount of work concerning tridiagonal matrices of the form (1). The interested reader may refer to, for example, the references Aiat and Elouafi (2008), El-Mikkawy (2004b), El-Mikkawy and Atlan (2014a), El-Mikkawy and Karawia (2006), El-Mikkawy and Rahmo (2008), Huang and McColl (1997), Kilic (2008a), Lewis (1982), Mallik (2001), Ran et al. (2009), Yamamoto (2001) and the references therein.
A general $n \times n$ tridiagonal matrix of the form (1) can be stored in $3n-2$ memory locations, rather than n^2 memory locations for a full matrix, by using three vectors $\mathbf{a} = [a_1, a_2, \dots, a_{n-1}]$, $\mathbf{b} = [b_1, b_2, \dots, b_{n-1}]$ and $\mathbf{d} = [d_1, d_2, \dots, d_n]$. This is always a good habit in computation in order to save memory space. To study tridiagonal matrices it is very convenient to introduce a vector $\mathbf{e}$ in the following way El-Mikkawy (2004a):

$$\mathbf{e} = [e_1, e_2, ..., e_n], \quad (2)$$

where

$$e_1 = d_1, \qquad e_i = d_i - \frac{b_{i-1}\, a_{i-1}}{e_{i-1}}, \quad i = 2, 3, ..., n. \quad (3)$$

Consider the $n \times n$ matrix, T_n defined in (1). Denote by:

$$f_1 = |d_1| = d_1, \quad f_i = \begin{vmatrix} d_1 & a_1 & 0 & \dots & 0 \\ b_1 & d_2 & a_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & b_{i-2} & d_{i-1} & a_{i-1} \\ 0 & \dots & 0 & b_{i-1} & d_i \end{vmatrix}, \ i = 2, 3, ..., n. \quad (4)$$

It is known that El-Mikkawy (2003), Hager (1988) the determinants in (4) satisfy a three-term recurrence:

$$f_i = d_i f_{i-1} - b_{i-1} a_{i-1} f_{i-2}, \quad i = 1, 2, \ldots, n, \tag{5}$$

where the initial values for f_i are $f_{-1} = 0$ and $f_0 = 1$. In particular, we have:

$$\det(T_n) = f_n. \tag{6}$$

In El-Mikkawy (2004a), the author developed a symbolic algorithm, called **DETGTRI** to compute general n-th order tridiagonal determinants in linear time. A more general tridiagonal matrix is the k-tridiagonal matrix $T_n^{(k)} = [\hat{t}_{ij}]_{i,j=1}^n$ which can be written in the form:

$$T_n^{(k)} = [\hat{t}_{ij}]_{i,j=1}^n = \begin{bmatrix} d_1 & 0 & \ldots & 0 & a_1 & 0 & \ldots & 0 \\ 0 & d_2 & 0 & \ldots & 0 & a_2 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ldots & \ddots & \ddots & 0 \\ 0 & \ldots & \ddots & \ddots & \ddots & \ldots & \ddots & a_{n-k} \\ b_1 & 0 & \ldots & \ddots & \ddots & \ddots & \ldots & 0 \\ 0 & b_2 & \ddots & \ldots & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ldots & 0 & d_{n-1} & 0 \\ 0 & \ldots & 0 & b_{n-k} & 0 & \ldots & 0 & d_n \end{bmatrix}, \quad n \geq 3. \tag{7}$$

These kind of matrices have attracted much attention in recent years. For the matrix $T_n^{(k)}$ in (7), $\hat{t}_{ij} = 0$ for all $i, j = 1, 2, ..., n$ except for $|i - j| = 0$ or k, where $k \in \{1, 2, ..., n-1\}$. For $k \geq n$, the matrix $T_n^{(k)}$ is a diagonal matrix and $T_n^{(1)} = T_n$. The k-tridiagonal matrix plays an important role in describing generalized k-Fibonacci numbers El-Mikkawy and Sogabe (2010), Sogabe and El-Mikkawy (2011), Yilmaz and Sogabe (2014). The symbolic algorithm, **k-DETGTRI** in El-Mikkawy (2012) can be used to compute the determinant of the matrix in (7) in linear time. It is a generalization of the **DETGTRI** algorithm in El-Mikkawy (2004a). For related work, see Asci et al. (2012), El-Mikkawy and Atlan (2014a,b), Jia et al. (2013), Sogabe and El-Mikkawy (2011), Yalciner (2011), Yilmaz and Sogabe (2014). When we consider matrices of the form (7) there is no need to store the zero elements. The nonzero elements of the matrix can be stored in $3n - 2k$ memory locations by using three vectors $\mathbf{a} = [a_1, a_2, \ldots, a_{n-k}]$, $\mathbf{b} = [b_1, b_2, \ldots, b_{n-k}]$ and $\mathbf{d} = [d_1, d_2, \ldots, d_n]$. To study k-tridiagonal matrices it is advantageous to introduce an n-component vector

$\mathbf{c} = [c_1, c_2, ..., c_n]$ whose components are given by:

$$c_i = \begin{cases} d_i, & \text{if } i = 1, 2, ..., k \\ d_i - \frac{b_{i-k}\, a_{i-k}}{c_{i-k}}, & \text{if } i = k+1, k+2, ..., n. \end{cases} \tag{8}$$

With the help of the vector $\mathbf{c}$ in (8), we may formulate the following basic result whose proof may be found in Asci et al. (2012), Burden and Faires (2001), El-Mikkawy (2012).

Theorem 1.1. *Let $T_n^{(k)}$ be a k-tridiagonal matrix in (7) for which $c_i \neq 0$, for $i = 1, 2, ..., n$. Then the Doolittle LU factorization of $T_n^{(k)}$ is given by:*

$$T_n^{(k)} = L_n^{(k)}\, U_n^{(k)}, \tag{9}$$

where

$$L_n^{(k)} = \begin{bmatrix} 1 & 0 & \dots & & & & \dots & 0 \\ 0 & 1 & \ddots & & & & & \vdots \\ \vdots & 0 & \ddots & \ddots & & & & \\ 0 & \vdots & \ddots & \ddots & \ddots & & & \\ \frac{b_1}{c_1} & \ddots & \vdots & \ddots & \ddots & \ddots & & \\ 0 & \frac{b_2}{c_2} & \ddots & \dots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \dots & 0 & \ddots & 0 \\ 0 & \dots & 0 & \frac{b_{n-k}}{c_{n-k}} & 0 & \dots & 0 & 1 \end{bmatrix},$$
$$U_n^{(k)} = \begin{bmatrix} c_1 & 0 & \dots & 0 & a_1 & 0 & \dots & 0 \\ 0 & c_2 & 0 & \dots & 0 & a_2 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \dots & \ddots & \ddots & 0 \\ & & \ddots & \ddots & \ddots & \dots & \ddots & a_{n-k} \\ & & & \ddots & \ddots & \ddots & \dots & 0 \\ & & & & \ddots & \ddots & 0 & \vdots \\ \vdots & & & & & \ddots & c_{n-1} & 0 \\ 0 & \dots & & & & \dots & 0 & c_n \end{bmatrix}, \tag{10}$$

and

$$\det(T_n^{(k)}) = \prod_{i=1}^{n} c_i, \tag{11}$$

where c_1, c_2, ..., c_n are given by (8).

The evaluation of n-th order k-tridiagonal determinant has been considered by some authors for the special case $k = 2$. See for instance Borowska et al. (2013) and Kilic (2008b). To the best of our knowledge there is no closed form determinant for n-th order k-tridiagonal determinants in the literature. The subject of the current paper is therefore to consider a closed-form determinants of k-tridiagonal matrices having Toeplitz structure. Also the modification of a recent algorithm for evaluating the general k-tridiagonal determinants Sogabe and Yilmaz (2014) will be taken into account.
The paper is organized as follows. The main results are given in the next section. In Section 3, a modified algorithm for evaluating the general n-th order k-tridiagonal determinants is proposed. Some illustrative examples are given in Section 4. Some concluding remarks are presented in Section 5.

2. Main Results

In this section we are going to evaluate the determinants of k-tridiagonal Toeplitz matrices in (7) for which, $d_i = d,\ i = 1, 2, ...n,\ a_i = a$ and $b_i = b,\ i = 1, 2, ...n - k$.

Theorem 2.1. *Consider the k-tridiagonal matrix of Toeplitz structure $T_n^{(k)}(b, d, a)$ defined by:*

$$T_n^{(k)}(b,d,a) = \begin{bmatrix} d & 0 & \dots & 0 & a & 0 & \dots & 0 \\ 0 & d & 0 & \dots & 0 & a & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \dots & \ddots & \ddots & 0 \\ 0 & \dots & \ddots & \ddots & \ddots & \dots & \ddots & a \\ b & 0 & \dots & \ddots & \ddots & \ddots & \dots & 0 \\ 0 & b & \ddots & \dots & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \dots & 0 & d & 0 \\ 0 & \dots & 0 & b & 0 & \dots & 0 & d \end{bmatrix}. \quad (12)$$

If $n = mk + r,\ r = 0, 1, 2, ..., k - 1,$ then we have

$$\det(T_n^{(k)}(b,d,a)) = \Big[\det(T_{m+1}(b,d,a))\Big]^r \Big[\det(T_m(b,d,a))\Big]^{k-r}. \quad (13)$$

Proof. By using (11), we have

$$\det(T_n^{(k)}(b,d,a)) = \prod_{p=1}^{n} c_p, \tag{14}$$

where

$$c_1 = c_2 = ... = c_k = d$$

and

$$c_i = d - \frac{a\,b}{c_{i-k}}, \qquad i = k+1, k+2, ..., n. \tag{15}$$

At this point let us consider the following second order linear homogeneous difference equation El-Mikkawy (2003), Hager (1988):

$$u_i = d\,u_{i-1} - b\,a\,u_{i-2}, \quad i = 1, 2, \ldots, n, \tag{16}$$

where the initial values for u_i are $u_{-1} = 0$ and $u_0 = 1$.
Since $n = mk + r,\ r = 0, 1, ..., k-1$ the values of $c_1,\ c_2,\ ..., c_n$ in (15) are now given by:

$$\begin{aligned}
&c_1 = c_2 = ... = c_k = \tfrac{d}{1} = \tfrac{u_1}{u_0},\\
&c_{k+1} = c_{k+2} = ... = c_{2k} = d - \tfrac{ab}{c_1} = d - \tfrac{ab}{\frac{u_1}{u_0}} = \tfrac{du_1 - ab\,u_0}{u_1} = \tfrac{u_2}{u_1},\\
&c_{2k+1} = c_{2k+2} = ... = c_{3k} = d - \tfrac{ab}{c_{k+1}} = d - \tfrac{ab}{\frac{u_2}{u_1}} = \tfrac{du_2 - ab\,u_1}{u_2} = \tfrac{u_3}{u_2},\\
&\vdots\\
&c_{(m-1)k+1} = c_{(m-1)k+2} = ... = c_{mk} = d - \tfrac{ab}{c_{(m-2)k+1}} = d - \tfrac{ab}{\frac{u_{m-1}}{u_{m-2}}} = \tfrac{du_{m-1} - ab\,u_{m-2}}{u_{m-1}} = \tfrac{u_m}{u_{m-1}},\\
&c_{mk+1} = c_{mk+2} = ... = c_{mk+r} = d - \tfrac{ab}{c_{(m-1)k+1}} = d - \tfrac{ab}{\frac{u_m}{u_{m-1}}} = \tfrac{du_m - ab\,u_{m-1}}{u_m} = \tfrac{u_{m+1}}{u_m},
\end{aligned} \tag{17}$$

having used (16). By using (17) into (14), we obtain

$$\begin{aligned}
\det(T_n^{(k)}(b,d,a)) &= \prod_{p=1}^{n} c_p = \prod_{p=1}^{mk+r} c_p = \Big[c_1\,c_2\,...c_k\Big]\Big[c_{k+1}\,c_{k+2}\,...c_{2k}\Big]...\\
&\quad ...\Big[c_{(m-1)k+1}\,c_{(m-1)k+2}\,...c_{mk}\Big]\Big[c_{mk+1}\,c_{mk+2}\,...c_{mk+r}\Big]\\
&= [\frac{u_1}{u_0}]^k[\frac{u_2}{u_1}]^k...[\frac{u_m}{u_{m-1}}]^k[\frac{u_{m+1}}{u_m}]^r\\
&= \left[(\frac{u_1}{u_0})(\frac{u_2}{u_1})...(\frac{u_m}{u_{m-1}})(\frac{u_{m+1}}{u_m})\right]^r\left[(\frac{u_1}{u_0})(\frac{u_2}{u_1})...(\frac{u_{m-1}}{u_{m-2}})(\frac{u_m}{u_{m-1}})\right]^{k-r}\\
&= (u_{m+1})^r(u_m)^{k-r},
\end{aligned}$$

on simplification. Consequently, we get

$$\begin{aligned}\det(T_n^{(k)}(b,d,a)) &= \Big[\det(T_{m+1}^{(1)}(b,d,a))\Big]^r \Big[\det(T_m^{(1)}(b,d,a))\Big]^{k-r} \\ &= \Big[\det(T_{m+1}(b,d,a))\Big]^r \Big[\det(T_m(b,d,a))\Big]^{k-r},\end{aligned}$$

having used (6). ■

It is to be noted that if n is divisible by m, and therefore $r = 0$, the last result reduces to the form

$$\det(T_n^{(k)}(b,d,a)) = \Big[\det(T_{\frac{n}{k}}(b,d,a))\Big]^k. \tag{18}$$

For the sake of completeness, we need to compute $T_m(b,d,a)$ and $T_{m+1}(b,d,a)$ in (13). To do this we may solve the difference equation in (16). For this purpose, we have to consider the discriminant, Δ which is given by:

$$\Delta = d^2 - 4ab. \tag{19}$$

Three cases will be considered accordingly.

Case (i): $\Delta > 0$.
Case (ii): $\Delta = 0$.
Case (iii): $\Delta < 0$.

For **Case (i)**, the solution of (16) is given by:

$$u_m = \det(T_m(b,d,a)) = \frac{1}{\sqrt{\Delta}}\Big(\alpha^{m+1} - \beta^{m+1}\Big), \tag{20}$$

where $\alpha = \frac{1}{2}(d + \sqrt{\Delta})$ and $\beta = \frac{1}{2}(d - \sqrt{\Delta})$.
For **Case (ii)**, we have:

$$u_m = \det(T_m(b,d,a)) = (m+1)(\frac{d}{2})^m. \tag{21}$$

Finally, for **Case (iii)**, we get:

$$u_m = \det(T_m(b,d,a)) = \frac{-i}{\sqrt{-\Delta}}\Big(\delta^{m+1} - \mu^{m+1}\Big), \tag{22}$$

where $i = \sqrt{-1}$, $\delta = \frac{1}{2}(d + i\sqrt{-\Delta})$ and $\mu = \frac{1}{2}(d - i\sqrt{-\Delta})$.

Summarizing, we have:

$$\det(T_n^{(k)}(b,d,a)) = \begin{cases} \left(\frac{1}{\sqrt{\Delta}}\right)^k \left(\alpha^{\frac{n}{k}+1} - \beta^{\frac{n}{k}+1}\right)^k, & \text{if } r = 0 \text{ and } \Delta > 0, \\ (\frac{n}{k}+1)^k (\frac{d}{2})^n, & \text{if } r = 0 \text{ and } \Delta = 0, \\ \left(\frac{-i}{\sqrt{-\Delta}}\right)^k \left(\delta^{\frac{n}{k}+1} - \mu^{\frac{n}{k}+1}\right)^k, & \text{if } r = 0 \text{ and } \Delta < 0, \\ \left(\frac{1}{\sqrt{\Delta}}\right)^k \left(\alpha^{m+2} - \beta^{m+2}\right)^r \left(\alpha^{m+1} - \beta^{m+1}\right)^{k-r}, & \text{if } r \neq 0 \text{ and } \Delta > 0, \\ (m+2)^r (m+1)^{k-r} (\frac{d}{2})^n, & \text{if } r \neq 0 \text{ and } \Delta = 0, \\ \left(\frac{-i}{\sqrt{-\Delta}}\right)^k \left(\delta^{m+2} - \mu^{m+2}\right)^r \left(\delta^{m+1} - \mu^{m+1}\right)^{k-r}, & \text{if } r \neq 0 \text{ and } \Delta < 0. \end{cases} \tag{23}$$

Two interesting remarks are given in order:

Remark 2.1. *If $n < 2k$, then we have $m = 1$ and $r = n - k$. Consequently, for all values of Δ, we obtain:*

$$\det(T_n^{(k)}(b,d,a)) = d^{2k-n}(d^2 - ab)^{n-k}. \tag{24}$$

Remark 2.2. *By using De Moivre's theorem Arfken (1985), we see that if $\Delta < 0$, then (22) can also be written in the form:*

$$\det(T_m(b,d,a)) = \frac{2(ab)^{\frac{m+1}{2}}}{\sqrt{-\Delta}} \sin(m+1)\theta, \tag{25}$$

where $\theta = \arctan(\frac{\sqrt{-\Delta}}{d})$.

With the help of (23), we are now in a position to formulate the following result:

Algorithm 2.1. (An algorithm for computing the determinant of $T_n^{(k)}(b,d,a)$ in (12)).

To compute the determinant of $T_n^{(k)}(b,d,a)$ in (12), we may proceed as follows:

INPUT: Order of the matrix n, value of k, the values, a, b and d.
OUTPUT: The determinant of $T_n^{(k)}(b,d,a)$ in (12).

Step 1: Find the values of m and r which satisfy $n = mk + r$.
Step 2: Set $\Delta = d^2 - 4ab$.
Step 3: **if** $\Delta > 0$ **then**
Compute: $\alpha = \frac{1}{2}(d + \sqrt{\Delta})$, $\beta = \frac{1}{2}(d - \sqrt{\Delta})$, and
$$\det(T_n^{(k)}(b,d,a)) = \left(\frac{1}{\sqrt{\Delta}}\right)^k \left(\alpha^{m+2} - \beta^{m+2}\right)^r \left(\alpha^{m+1} - \beta^{m+1}\right)^{k-r}.$$
elseif $\Delta = 0$ **then**
Compute: $\det(T_n^{(k)}(b,d,a)) = (m+2)^r(m+1)^{k-r}(\frac{d}{2})^n$.
else
Compute: $\delta = \frac{1}{2}(d + i\sqrt{\Delta})$, $\mu = \frac{1}{2}(d - i\sqrt{\Delta})$, and
$$\det(T_n^{(k)}(b,d,a)) = \left(\frac{-i}{\sqrt{-\Delta}}\right)^k \left(\delta^{m+2} - \mu^{m+2}\right)^r \left(\delta^{m+1} - \mu^{m+1}\right)^{k-r}.$$
end if.

Algorithm 2.1 will be referred to as **DETkTOEP-I**.

3. A Modified Algorithm for Evaluating the General n-th Order k-Tridiagonal Determinants

Following Sogabe and Yilmaz (2014), let n, k be natural numbers such that $n \geq 3$ and $k = 1, 2, ..., n-1$. For each values of n and k, consider the associated set $A_n = \{1, 2, ..., n\}$. The equivalence class $[p]$, $p \in \{1, 2, ..., k\}$ is defined by:

$$[p] = \{i \in A_n \text{ such that } i \equiv p \bmod (k)\}. \tag{26}$$

For $j = 1, 2, ..., k$, let N_j denotes the number of elements of the equivalence class $[j]$. Then we have

$$A_n = \bigcup_{p \in \{1,2,\ldots,k\}} [p], \tag{27}$$

$$[p] \cap [q] = \phi \text{ for } p \neq q,\ p,\ q \in \{1, 2, ..., k\} \tag{28}$$

and

$$\sum_{j=1}^{k} N_j = n. \tag{29}$$

For example, if $n = 10$ and $k = 4$, then we have the equivalence classes $[1] = \{1, 5, 9\}$, $[2] = \{2, 6, 10\}$, $[3] = \{3, 7\}$ and $[4] = \{4, 8\}$. Therefore $N_1 = N_2 = 3$ and $N_3 = N_4 = 2$. Very recently in Sogabe and Yilmaz (2014), the authors constructed a numeric algorithm for evaluating general n-th order determinants of k-tridiagonal type (Algorithm 1, p. 99). In fact this algorithm works properly only if $n \geq 2k$. The motivation of this section is therefore to modify this algorithm to work for $n < 2k$ as well. We begin by noticing that if $n < 2k$, then we have:

$$N_1 = N_2 = ... = N_{n-k} = 2 < 3 \tag{30}$$

and

$$N_{n-k+1} = N_{n-k+2} = ... = N_k = 1 < 3. \tag{31}$$

Additionally, if $n = 2k$, then n is necessarily even and $N_j = 2 < 3$, for $j = 1, 2, ..., k$. For $n > 2k$, the value of N_1 is at least 3. Armed with the above results, we may formulate the following modified numeric algorithm which is breakdown-free.

Algorithm 3.1. (A modification of Algorithm 1 in Sogabe and Yilmaz (2014), p. 99, p= -1).

To compute the determinant of a general k-tridiagonal matrix in (7), we may proceed as follows:

INPUT: Order of the matrix n, value of k and the values, a_i, b_i, $i = 1, 2, \ldots, n-k$, d_i, $i = 1, 2, \ldots, n$.

OUTPUT: The determinant of $T_n^{(k)}$ in (7).

Step 1: **if** $n \leq 2k$ **then**
    for $j = 1, 2, ..., k$ **do**
      $f_j = d_j$
      if $j \leq n - k$ **then**
        $f_{j+k} = d_{j+k} f_j - b_j a_j$
      end if
    end do
  else
    compute $N_1, N_2, ..., N_k$
      for $j = 1, 2, ..., k$ **do**
        $f_j = d_j$
        $f_{j+k} = d_{j+k} f_j - b_j a_j$
        for $i = 3, 4, ..., N_j$ **do**
          $f_{k(i-1)+j} = d_{k(i-1)+j} f_{k(i-2)+j} - b_{k(i-2)+j} a_{k(i-2)+j} f_{k(i-3)+j}$
        end do
      end do
  end if

Step 2: **Compute** $\det(T_n^{(k)}) = \prod_{s=1}^{k} f_{n-k+s}$.

For the special case where $k = 1$, we have $n > 2k$ and $N_k = N_1 = n$. In this case the Algorithm 3.1 reduces to Algorithm 3 in Sogabe and Yilmaz (2014).

For the $n \times n$ matrix, $T_n^{(k)}(b, d, a)$ of the form (12), the Algorithm 3.1 takes the form:

Algorithm 3.2. (An algorithm for computing the determinant of $T_n^{(k)}(b, d, a)$ in (12)).

To compute the determinant of $T_n^{(k)}(b, d, a)$ in (12), we may proceed as follows:
INPUT: Order of the matrix n, value of k, the values, a, b and d.
OUTPUT: The determinant of $T_n^{(k)}(b, d, a)$ in (12).
Step 1: Compute $N_1, N_2, ..., N_k$
for $j = 1, 2, ..., k$ **do**
$f_j = d$
$f_{j+k} = d\, f_j - b\, a$
for $i = 3, 4, ..., N_j$ **do**
$f_{k(i-1)+j} = d\, f_{k(i-2)+j} - b\, a\, f_{k(i-3)+j}$
end do
end do
Step 2: if $n \leq 2k$ **then**
$\det(T_n^{(k)}(b, d, a) = d^{2k-n}\,(d^2 - a\, b)^{n-k}$.
else
$\det(T_n^{(k)}(b, d, a)) = \prod_{s=1}^{k} f_{n-k+s}$.
end if.

Algorithm 3.2 will be referred to as **DETkTOEP-II**. The MATLAB codes for the algorithms **DETkTOEP-I** and **DETkTOEP-II** are available from the authors upon request.

4. Numerical Tests and Illustrative Examples

In this section we are going to consider some numerical tests and illustrative examples. All experiments were carried out using **MATLAB** 7.10.0.499 (R2010a) on a PC with Intel(R) Core(TM) i7-3770 CPU processor.

Example 4.1. *Consider the symmetric k-tridiagonal matrix,* $T_n^{(k)} = [\hat{t}_{ij}]_{i,j=1}^{n}$ *where*

$$\hat{t}_{ij} = \begin{cases} 3, & \text{if } i = j \\ -1, & \text{if } |i-j| = k \\ 0, & \text{otherwise.} \end{cases}$$

Compute $\det(T_n^{(k)})$.

Solution. We have $d = 3$, $a = b = -1$ and $\Delta = d^2 - 4ab = 5 > 0$. We also have $\alpha = \frac{1}{2}(3+\sqrt{5}) = (\frac{1+\sqrt{5}}{2})^2$ and $\beta = \frac{1}{2}(3-\sqrt{5}) = (\frac{1-\sqrt{5}}{2})^2$. By applying (23), we get:

$$\begin{aligned}\det(T_n^{(k)}(b,d,a)) &= \left(\frac{1}{\sqrt{\Delta}}\right)^k \left(\alpha^{m+2} - \beta^{m+2}\right)^r \left(\alpha^{m+1} - \beta^{m+1}\right)^{k-r} \\ &= \left[\left(\frac{1}{\sqrt{\Delta}}\right)\left(\alpha^{m+2} - \beta^{m+2}\right)\right]^r \left[\left(\frac{1}{\sqrt{\Delta}}\right)\left(\alpha^{m+1} - \beta^{m+1}\right)\right]^{k-r} \\ &= \left(F_{2m+3}\right)^r \left(F_{2m+1}\right)^{k-r}\end{aligned} \tag{32}$$

where the Fibonacci numbers, F_j in (32) are given by:

$$F_j = \frac{1}{\sqrt{5}}\left((\frac{1+\sqrt{5}}{2})^{j+1} - (\frac{1-\sqrt{5}}{2})^{j+1}\right),\ j = 0, \pm 1, \pm 2, ... \tag{33}$$

It is worth mentioned that $\frac{1+\sqrt{5}}{2}$ is called the golden ratio Rosen (2000).

Example 4.2. *Consider the k-tridiagonal matrix,* $T_n^{(k)} = [\hat{t}_{ij}]_{i,j=1}^n$ *for* $n = 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000,$ *and* $k = 2$ *with*

$$\hat{t}_{ij} = \begin{cases} 1.5 + h^2, & \text{if } i = j \\ 1, & \text{if } |i-j| = k \\ 0, & \text{otherwise,} \end{cases}$$

where $h = \frac{1}{n}$.

The numerical results with **DETkTOEP-II**, **MATLAB** $'\det()'$ built-in function and **DETkTOEP-I** algorithm are given in Table 1. The mean value of the CPU times (after 100 tests) in the computation of the determinant are shown in Figure 1.

Example 4.3. *Consider the k-tridiagonal matrix,* $T_n^{(k)} = [\hat{t}_{ij}]_{i,j=1}^n$ *for* $n = 1000, 5000, 10000, 15000, 20000, 25000, 30000, 35000, 40000$ *and* $k = 3$ *given by*

$$\hat{t}_{ij} = \begin{cases} 2, & \text{if } i = j \\ 1, & \text{if } |i-j| = k \\ 0, & \text{otherwise.} \end{cases}$$

Table 1: Numerical results of the determinants for Example 4.2

n	**DETkTOEP-I** algorithm	**DETkTOEP-II** algorithm	**MATLAB** built-in function
20	2.282574412392193	2.282574412392195	2.282574412392195
50	0.010048080272980	0.010048080272980	0.010048080272980
100	1.275628131576196	1.275628131576192	1.275628131576192
200	1.033858646548643	1.033858646548663	1.033858646548665
500	1.191852582595496	1.191852582595514	1.191852582595517
1000	1.190676325073811	1.190676325073825	1.190676325073820
2000	1.381286295217381	1.381286295217223	1.381286295217232
5000	1.891450930105525	1.891450930104585	1.891450930104579
10000	2.285552570500651	2.285552570500804	2.285552570500802

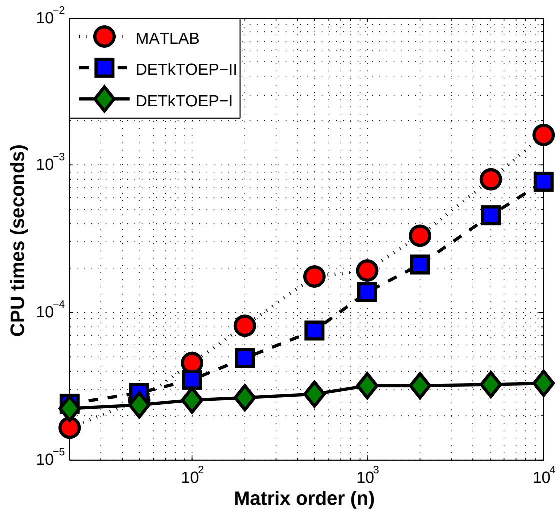

Figure 1: CPU times (after 100 tests), in Log scale, for Example 4.2.

Table 2: Numerical results of the determinants for Example 4.3

n	**DETkTOEP-I** algorithm	**DETkTOEP-II** algorithm	**MATLAB** built-in function
1000	37371260	37371260	3.737125999999501e+7
5000	4.637967408000e+9	4.637967408000e+9	4.637967407995532e+9
10000	3.707037926000e+10	3.707037926000e+10	3.707037925958355e+10
15000	1.250750150010e+11	1.250750150010e+11	1.250750149990213e+11
20000	2.964296474080e+11	2.964296474080e+11	2.964296474030406e+11
25000	5.789120592600e+11	5.789120592600e+11	5.789120592374078e+11
30000	1.000300030001e+12	1.000300030001e+12	1.000300029951314e+12
35000	1.588371327408e+12	1.588371327408e+12	1.588371327317226e+12
40000	2.370903739260e+12	2.370903739260e+12	2.370903739125952e+12

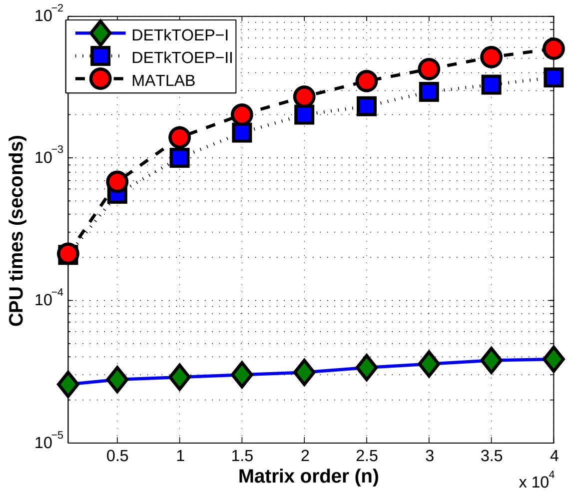

Figure 2: CPU times (after 100 tests), in Log scale, for Example 4.3

Table 2 gives the values of the determinant for different values of n. In each case $k = 3$. Also Figure 2, gives the mean value of the CPU times (after 100 tests) in the computation of the determinant.

Example 4.4. *Compute the determinant of the k-tridiagonal matrix $T_n^{(k)}$ given by:*

$$T_n^{(k)} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 3 \end{bmatrix}.$$

Solution. Here $k = 4$, $n = 6 < 2k$. We are going to solve this example by using two different methods:
Method 1: By using Algorithm 3.1, we obtain:
$f_1 = d_1 = 1$, $f_5 = d_5\, f_1 - b_1\, a_1 = 2$, $f_2 = d_2 = -2$, $f_6 = d_6\, f_2 - b_2\, a_2 = -2$, $f_3 = d_3 = 5$ and $f_4 = d_4 = 3$. Consequently,

$$\det(T_n^{(k)}) = \prod_{s=1}^{k} f_{n-k+s} = f_3 \times f_4 \times f_5 \times f_6 = (5)(3)(2)(-2) = -60.$$

Method 2: Applying the **k-DETGTRI** algorithm El-Mikkawy (2012), yields:

$$\det(T_n^{(k)}) = \prod_{s=1}^{6} c_s = (1)(-2)(5)(3)(2)(1) = -60.$$

5. Concluding Remarks

In this paper, we have considered the determinant evaluation of n-th order k-tridiagonal determinants having Toeplitz structure. Test results indicate the superiority of the new algorithm relative to the MATLAB built-in function and the **DETkTOEP-II** algorithm of the present paper.

Acknowledgement

The authors wish to thank anonymous referees and the editorial board of the MJMS for careful reading and useful comments that enhanced the quality of this paper.

References

Aiat, H. D. and Elouafi, M. (2008). A fast numerical algorithm for the inverse of a tridiagonal and pentadiagonal matrix. *Appl. Math. Comput.*, **202**:441–445.

Arfken, G. (1985). *Mathematical methods for physics, 3rd ed.* Orlando, Academic Press.

Asci, M., Tasci, D., and El-Mikkawy, M. (2012). On determinants and permanents of k-tridiagonal toeplitz matrices. *Util. Math.*, **89**:97–106.

Borowska, J., Lacinska, L., and Rychlewska, J. (2013). On determinant of certain pentadiagonal matrix. *J. Appl. Math. Comput. Mech.*, **12**(3):21–26.

Burden, R. and Faires, J. (2001). *Numerical analysis, 7th Edition.* Books & Cole Publishing, Pacific Grove, CA.

El-Mikkawy, M. (2003). A note on a three-term recurrence for a tridiagonal matrix. *Appl. Math. Comput.*, **139**:503–511.

El-Mikkawy, M. (2004a). A fast algorithm for evaluating nth order tri-diagonal determinants. *J. Comput. Appl. Math.*, **166**:581–584.

El-Mikkawy, M. (2004b). On the inverse of a general tridiagonal matrix. *Appl. Math. Comput.*, **150**:669–679.

El-Mikkawy, M. (2012). A generalized symbolic thomas algorithm. *Applied Mathematics*, **3**(4):342–345.

El-Mikkawy, M. and Atlan, F. (2014a). Algorithms for solving linear systems of equations of tridiagonal type via transformations. *Applied Mathematics*, **5**:413–422.

El-Mikkawy, M. and Atlan, F. (2014b). A novel algorithm for inverting a general k-tridiagonal matrix. *Appl. Math. Lett.*, **32**:41–47.

El-Mikkawy, M. and Karawia, A. (2006). Inversion of general tridiagonal matrices. *Appl. Math. Lett.*, **19**:712–720.

El-Mikkawy, M. and Rahmo, E. (2008). A new recursive algorithm for inverting tridiagonal and anti-tridiagonal matrices. *Appl. Math. Comput.*, **204**:368–372.

El-Mikkawy, M. and Sogabe, T. (2010). A new family of k-fibonacci numbers. *Appl. Math. Comput.*, **215**:4456–4461.

Hager, W. (1988). *Applied numerical linear algebra.* Prentice-Hall International Editions, Englewood Cliffs, NJ.

Huang, Y. and McColl, W. (1997). Analytic inversion of general tridiagonal matrices. *J. Phys. A.*, **30**:7919–7933.

Jia, J., Sogabe, T., and El-Mikkawy, M. (2013). Inversion of k-tridiagonal matrices with toeplitz structure. *Comput. Math. Appl.*, **65**:116–125.

Kilic, E. (2008a). Explicit formula for the inverse of a tridiagonal matrix by backward continued fractions. *Appl. Math. Comput.*, **197**:345–357.

Kilic, E. (2008b). On a constant-diagonals matrix. *Appl. Math. Comput.*, **204**:184–190.

Lewis, J. (1982). Inversion of the tridiagonal matrices. *Numer. Math.*, **38**:333–345.

Mallik, R. (2001). The inverse of a tridiagonal matrix. *Linear Algebr. Appl.*, **325**:109–139.

Ran, R.-S., Huang, T.-Z., Liu, X.-P., and Gu, T.-X. (2009). An inversion algorithm for general tridiagonal matrix. *Appl. Math. Mech.-Engl. Ed.*, **30**:247–253.

Rosen, K. (2000). *Handbook of discrete and combinatorial mathematics.* CRC Press, Boca Raton, FL.

Sogabe, T. and El-Mikkawy, M. (2011). Fast block diagonalization of k-tridiagonal matricess. *Appl. Math. Comput.*, **218**:2740–2743.

Sogabe, T. and Yilmaz, F. (2014). A note on a fast breakdown-free algorithm for computing the determinants and the permanents of k-tridiagonal matrices. *Appl. Math. Comput.*, **249**:98–102.

Yalciner, A. (2011). The lu factorization and determinants of the k-tridiagonal matrices. *Asian-European J. Math.*, **4**:187–197.

Yamamoto, T. (2001). Inversion formulas for tridiagonal matrices with applications to boundary value problems. *Numer. Funct. Anal. and Optimiz.*, **22**:357– 385.

Yilmaz, F. and Sogabe, T. (2014). A note on symmetric k-tridiagonal matrix family and the fibonacci. *Int. J. Pure and Appl. Math.*, **96**(2):289–298.