Applied Mathematics, 2015, 6, 967-978
Published Online June 2015 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2015.66089

Solving Doubly Bordered Tridiagonal Linear Systems via Partition

Moawwad El-Mikkawy[1*], Mohammed El-Shehawy[2], Nermeen Shehab[2]

[1]Mathematics Department, Faculty of Science, Mansoura University, Mansoura, Egypt
[2]Mathematics Department, Faculty of Science, Damietta University, Egypt
Email: [*]m_elmikkawy@yahoo.com, melshehawey@yahoo.com, nermeen_shehab87@yahoo.com

Received 27 April 2015; accepted 30 May 2015; published 2 June 2015

Abstract

This paper presents new numeric and symbolic algorithms for solving doubly bordered tridiagonal linear system. The proposed algorithms are derived using partition together with UL factorization. Inversion algorithm for doubly bordered tridiagonal matrix is also considered based on the Sherman-Morrison-Woodbury formula. The algorithms are implemented using the computer algebra system, MAPLE. Some illustrative examples are given.

Keywords

Doubly Bordered Tridiagonal Matrices, UL Factorization, Block Matrices, Computer Algebra Systems, Sherman-Morrison-Woodbury Formula

1. Introduction

Linear systems are among the most important and common problems encountered in scientific computing. The whole range of technical problems leads to the solution of systems of linear equations. The first step in numerical solution of many problems is a choice of an appropriate algorithm.

Throughout this paper the word, "simplify", means simplify the expression under consideration to its simplest rational form. The abbreviation DBT refers to doubly bordered tridiagonal.

The main objective of the current paper is to construct efficient algorithms for solving DBT linear system of the form:

$$Ax = \boldsymbol{f} \tag{1}$$

where the coefficient matrix A is a DBT matrix with the special structure of the partitioned form

[*]Corresponding author.

How to cite this paper: El-Mikkawy, M., El-Shehawy, M. and Shehab, N. (2015) Solving Doubly Bordered Tridiagonal Linear Systems via Partition. *Applied Mathematics*, **6**, 967-978. http://dx.doi.org/10.4236/am.2015.66089

$$A=\left[\begin{array}{c|cccccc} d_1 & a_1 & h_1 & h_2 & \cdots & h_{n-3} & h_{n-2} \\ \hline b_2 & d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ v_1 & b_3 & d_3 & a_3 & \ddots & & \vdots \\ v_2 & 0 & b_4 & d_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ v_{n-3} & \vdots & & \ddots & b_{n-1} & d_{n-1} & a_{n-1} \\ v_{n-2} & 0 & \cdots & \cdots & 0 & b_n & d_n \end{array}\right],$$

$\boldsymbol{x}=\left(x_1\ x_2 \cdots x_n\right)^t$ and $\boldsymbol{f}=\left(f_1\ f_2 \cdots f_n\right)^t$, $n \geq 3$.

Such systems arise in many scientific and engineering applications when considering certain partial differential equations and spline approximation. For more details see, for instance [1]-[5]. It is worth noting that in [6], the authors solved the system (1) via transformation method. In this paper, we are going to solve the system under consideration via partition.

The linear system (1) can also be written in block form as follows:

$$\left[\begin{array}{c|c} d_1 & \boldsymbol{h}^t \\ \hline \boldsymbol{v} & \hat{T} \end{array}\right]\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right]=\left[\begin{array}{c} \boldsymbol{f}_1 \\ \hline \boldsymbol{f}_2 \end{array}\right] \tag{3}$$

where

$$A=\left[\begin{array}{c|c} d_1 & \boldsymbol{h}^t \\ \hline \boldsymbol{v} & \hat{T} \end{array}\right] \tag{4}$$

is a 2×2 block matrix, $\boldsymbol{h}^t=\left(a_1 \quad h_1 \quad h_2 \quad \cdots \quad h_{n-3} \quad h_{n-2}\right)$, $\boldsymbol{v}=\left(b_2 \quad v_1 \quad v_2 \quad \cdots \quad v_{n-3} \quad v_{n-2}\right)^t$ $\boldsymbol{h}, \boldsymbol{v} \in \mathbb{R}^{(n-1)\times 1}$, $\boldsymbol{x}_1=\left(x_1\right)$, $\boldsymbol{x}_2=\left(x_2 \quad x_3 \quad \cdots \quad x_n\right)^t$, $\boldsymbol{f}_1=\left(f_1\right)$, $\boldsymbol{f}_2=\left(f_2 \quad f_3 \quad \cdots \quad f_n\right)^t$ and $\hat{T}$ is $(n-1)\times(n-1)$ tridiagonal matrix which is given by

$$\hat{T}=\begin{bmatrix} d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ b_3 & d_3 & a_3 & \ddots & & \vdots \\ 0 & b_4 & d_4 & a_4 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \cdots & \cdots & 0 & b_n & d_n \end{bmatrix}. \tag{5}$$

The present paper is organized as follows. In Section 2, we introduce some facts about tridiagonal matrices and a new recursive procedure for inverting this matrix is proposed. In Section 3, the UL factorization of DBT matrix is considered. Numeric and symbolic algorithms for evaluating DBT matrix determinant are constructed. Also a symbolic algorithm for computing the inverse of DBT matrix is developed using the Sherman-Morrison-Woodbury formula. Finally, the solution of the linear system whose coefficient matrix is of DBT matrix type is proposed. In Section 4, some illustrative examples are given.

2. Preliminaries and Basic Results

Tridiagonal matrices arise in many contexts in pure and applied mathematics. Also, they arise in many different theoretical fields, especially in applicative fields such as spline approximation, numerical analysis, ordinary and partial differential equations, solution of linear systems of equations, engineering, telecommunication system analysis, system identification, signal processing (e.g., speech decoding), partial differential equations and naturally linear algebra. See [1]-[4]. Research area on these types of matrices is very active and has recently attracted the attention of many researchers.

In many of these areas, inversions of tridiagonal matrices are necessary and different approaches are considered [7]-[10] in order to obtain such inverse. Some authors consider a general tridiagonal matrix of finite order and then describe the LU factorizations, determine the determinant and inverse of a tridiagonal matrix under certain conditions [1], [2] and sometimes without any restrictive conditions [8], [11]. The interested reader may refer to [12]-[23].

The general tridiagonal matrix $T = \left(t_{ij}\right) 1 \le i,\ j \le n$ takes the form

$$T = \begin{bmatrix} d_1 & a_1 & 0 & \cdots & \cdots & 0 \\ b_2 & d_2 & a_2 & \ddots & & \vdots \\ 0 & b_3 & d_3 & a_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \cdots & \cdots & 0 & b_n & d_n \end{bmatrix} \tag{6}$$

in which $t_{ij} = 0$ for $|i-j| \ge 2$.

When we consider the matrix *T* defined by (6), it is helpful to introduce the *n* quantities $e_i,\ i = 1,2,\cdots,n$ as follows:

$$e_i = \begin{cases} d_n & \text{if } i = n, \\ d_i - \dfrac{a_i}{e_{i+1}} b_{i+1} & \text{if } i = n-1, n-2, \cdots, 1. \end{cases} \tag{7}$$

Following [8], we have the following result whose proof will be omitted for the sake of space requirement.

Theorem 2.1. The *UL* factorization of the matrix *T* in (6) is possible if $e_i \ne 0$ for each $i = 1,2,\cdots,n$. In this case we have the following two *UL* factorizations:

$$T = U_1 L_1 = U_2 L_2 \tag{8}$$

where

$$U_1 = \begin{bmatrix} e_1 & a_1 & 0 & \cdots & \cdots & 0 \\ 0 & e_2 & a_2 & \ddots & & \vdots \\ 0 & 0 & e_3 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & a_{n-1} \\ 0 & \cdots & & \cdots & 0 & e_n \end{bmatrix}, \quad L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ \dfrac{b_2}{e_2} & 1 & 0 & & & \\ 0 & \dfrac{b_3}{e_3} & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \dfrac{b_n}{e_n} & 1 \end{bmatrix} \tag{9}$$

and

$$U_2 = \begin{bmatrix} 1 & \dfrac{a_1}{e_2} & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \dfrac{a_2}{e_3} & \ddots & & \vdots \\ 0 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \ddots & \dfrac{a_{n-1}}{e_n} \\ 0 & \cdots & & \cdots & 0 & 1 \end{bmatrix}, \quad L_2 = \begin{bmatrix} e_1 & 0 & 0 & \cdots & \cdots & 0 \\ b_2 & e_2 & 0 & & & \\ 0 & b_3 & e_3 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & b_n & e_n \end{bmatrix}. \tag{10}$$

The following are two useful remarks:

Remark 1: The matrices L_1, L_2, U_1, U_2 in (9) and (10) are related by

$$L_1 = D^{-1} L_2 \tag{11}$$

and

$$U_1 = U_2 D \tag{12}$$

where

$$D = Diag\left(e_1, e_2, \cdots, e_n\right) = \begin{bmatrix} e_1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & e_2 & 0 & \ddots & & \vdots \\ 0 & 0 & e_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & & \cdots & 0 & e_n \end{bmatrix}. \tag{13}$$

Remark 2: If the *UL* factorization of the matrix *T* in (6) is possible, then we have:

$$Det\left(T\right) = \prod_{k=1}^{n} e_k \tag{14}$$

where $e_1, e_2, \cdots, e_n$ are given by (7). In other words, the matrix *T* is invertible as long as

$$\prod_{i=1}^{n} e_i \neq 0. \tag{15}$$

Now we are going to state the following result without proof.

Theorem 2.2. Let *T* be a non-singular tridiagonal matrix, given by (6), and let $T^{-1} = \left(\alpha_{ij}\right)_{1 \le i,\, j \le n}$ then for $i, j = 1, 2, \cdots, n$ we have

$$\alpha_{i,j} = \begin{cases} \dfrac{1}{e_1} & \text{if } i = j = 1 \\ \dfrac{1}{e_i} + y_{i-1} z_i \alpha_{i-1,i-1} & \text{if } i = j \neq 1 \\ -z_i \alpha_{i-1,\,j} & \text{if } i > j \\ -y_{j-1} \alpha_{i,\,j-1} & \text{if } i < j \end{cases} \tag{16}$$

where $z_i = \dfrac{b_i}{e_i},\ y_{i-1} = \dfrac{a_{i-1}}{e_i},\ i = 2, \cdots, n.$

3. Doubly Bordered Triangular Matrix (DBT Matrix)

In this section, we are going to construct new numeric and symbolic algorithms to evaluate the determinant of a DBT matrix and to compute the inverse of such matrices. For these purposes it is advantageous to consider the *UL* factorization.

To factor the *n* × *n* DBT matrix *A* given in (2) into the product of an upper-triangular matrix $U = \left(u_{ij}\right)$ and a lower-triangular matrix $L = \left(l_{ij}\right)$ in the form *A* = *UL*, we may proceed as follows

$$A=\left[\begin{array}{c|cccccc} d_1 & a_1 & h_1 & h_2 & \cdots & h_{n-3} & h_{n-2} \\ \hline b_2 & d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ v_1 & b_3 & d_3 & a_3 & \ddots & & \vdots \\ v_2 & 0 & b_4 & d_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ v_{n-3} & \vdots & & \ddots & b_{n-1} & d_{n-1} & a_{n-1} \\ v_{n-2} & 0 & \cdots & \cdots & 0 & b_n & d_n \end{array}\right]$$

$$=\left[\begin{array}{c|cccccc} 1 & q_1 & q_2 & q_3 & \cdots & \cdots & q_{n-1} \\ \hline 0 & 1 & \frac{a_2}{e_3} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \frac{a_3}{e_4} & \ddots & & \vdots \\ 0 & 0 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & & & \ddots & \ddots & \frac{a_{n-1}}{e_n} \\ 0 & 0 & \cdots & & \cdots & 0 & 1 \end{array}\right]\times\left[\begin{array}{c|cccccc} e_1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \hline p_1 & e_2 & 0 & 0 & \cdots & \cdots & 0 \\ p_2 & b_3 & e_3 & 0 & & & \\ p_3 & 0 & b_4 & e_4 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 \\ p_{n-1} & 0 & \cdots & \cdots & 0 & b_n & e_n \end{array}\right]=UL, \tag{17}$$

where

$$U=\left[\begin{array}{c|cccccc} 1 & q_1 & q_2 & q_3 & \cdots & \cdots & q_{n-1} \\ \hline 0 & 1 & \frac{a_2}{e_3} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \frac{a_3}{e_4} & \ddots & & \vdots \\ 0 & 0 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & & & \ddots & \ddots & \frac{a_{n-1}}{e_n} \\ 0 & 0 & \cdots & & \cdots & 0 & 1 \end{array}\right] \tag{18}$$

and

$$L=\left[\begin{array}{c|cccccc} e_1 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ \hline p_1 & e_2 & 0 & 0 & \cdots & \cdots & 0 \\ p_2 & b_3 & e_3 & 0 & & & \\ p_3 & 0 & b_4 & e_4 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 \\ p_{n-1} & 0 & \cdots & \cdots & 0 & b_n & e_n \end{array}\right]. \tag{19}$$

Equation (17) can be written in block matrix form as follows:

$$\left[\begin{array}{c|c} d_1 & \boldsymbol{h}^t \\ \hline \boldsymbol{v} & \hat{T} \end{array}\right]=\left[\begin{array}{c|c} 1 & \boldsymbol{q}^t \\ \hline \boldsymbol{0} & \hat{U} \end{array}\right]\left[\begin{array}{c|c} e_1 & \boldsymbol{0} \\ \hline \boldsymbol{p} & \hat{L} \end{array}\right] \tag{20}$$

where $\boldsymbol{q}^t=\left(q_1 \quad q_2 \quad \cdots \quad q_{n-2} \quad q_{n-1}\right)$, $\boldsymbol{p}=\left(p_1 \quad p_2 \quad \cdots \quad p_{n-2} \quad p_{n-1}\right)^t$, $\hat{L}$ and $\hat{U}$ are the *lower* and *upper* triangular matrices of the tridiagonal matrix $\hat{T}$ given in (5), respectively.

From (20), we see that the following equations are satisfied:

$$e_1 + \boldsymbol{q}^t \boldsymbol{p} = d_1, \tag{21}$$

$$\boldsymbol{h}^t = \boldsymbol{q}^t \hat{L}, \tag{22}$$

$$\boldsymbol{v} = \hat{U}\boldsymbol{p}, \tag{23}$$

and

$$\hat{T} = \hat{U}\hat{L}. \tag{24}$$

Solving (21), (22) and (23) for e_1, $\boldsymbol{q}$ and $\boldsymbol{p}$ yields:

$$q_i = \begin{cases} h_{n-2} & \text{if } i = n-1, \\ h_{i-1} - z_{i+2}\, q_{i+1} & \text{if } i = n-2, n-3, \cdots, 2 \\ a_1 - z_3\, q_2 & \text{if } i = 1, \end{cases} \tag{25}$$

$$p_i = \begin{cases} \dfrac{v_{n-2}}{e_n} & \text{if } i = n-1, \\ \dfrac{v_{i-1}}{e_{i+1}} - y_{i+1}\, p_{i+1} & \text{if } i = n-2, n-3, \cdots, 2 \\ \dfrac{b_2}{e_2} - y_2\, p_2 & \text{if } i = 1. \end{cases} \tag{26}$$

and

$$e_1 = d_1 - \sum_{i=1}^{n-1} q_i\, p_i\,, \tag{27}$$

where

$$z_i = \frac{b_i}{e_{i-1}},\; y_{i-1} = \frac{a_{i-1}}{e_i}\;, i = 3, \cdots, n. \tag{28}$$

3.1. Algorithms for Evaluating the Determinant of a DBT Matrix

The determinant of the matrix A in (2) can be computed using the following numeric algorithm.

Algorithm 3.1. A numeric algorithm for computing the determinant of a DBT matrix.

To compute the determinant of the DBT matrix in (2), we may proceed as follows:

```
INPUT: Order of the matrix, $n$ and vectors $\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}, \boldsymbol{h}, \boldsymbol{v}$.
OUTPUT: The determinant of the matrix $A$ in (2).
Step 1: Set $e_n = d_n$.
Step 2: For $i = n-1, n-2, \cdots, 2$ do
        compute and simplify $e_i = d_i - \frac{a_i}{e_{i+1}} b_{i+1}$.
        End do.
Step 3: Compute $e_1$ using (27).
Step 4: $\mathrm{Det}(A) = \prod_{r=1}^{n} e_r$.
```

As can be easily seen, Algorithm (3.1) breaks down if any $e_i = 0$ for some $i \in \{1, 2, \cdots, n\}$. The following symbolic algorithm is developed in order to remove the case where the numeric algorithm fails.

Algorithm 3.2. A symbolic algorithm for computing the determinant of DBT matrix.

To compute the determinant of the DBT matrix in (2), we may proceed as follows:

```
INPUT: Order of the matrix, $n$ and vectors $\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}, \boldsymbol{h}, \boldsymbol{v}$.
```

```
OUTPUT: The determinant of the matrix A in (2)
Step 1: Set $e_n = d_n$. If $e_n = 0$ then $e_n = t$ End if. (t is just a symbolic name)
Step 2: For $i = n - 1, n - 2, \cdots, 2$ do
            compute and simplify:
                $e_i = d_i - \frac{a_i}{e_{i+1}} b_{i+1}$. If $e_i = 0$ then $e_i = t$ End if.
        End do.
Step 3: Compute and simplify $e_1$ using (27).
Step 4: Compute and simplify:
                $p(t) = \prod_{r=1}^{n} e_r$.
Step 5: Det (A) = p(0).
```

The Algorithm (3.2) will be referred to as **DETGDBTRI** algorithm.

3.2. Algorithm for Inverting DBT Matrix

In this sub-section, we are going to formulate a new symbolic algorithm for inverting the general DBT matrix based on the Sherman-Morrison-Woodbury formula.

A general DBT matrix can be written as following:

$$A = \begin{bmatrix} d_1 & a_1 & h_1 & h_2 & \cdots & h_{n-3} & h_{n-2} \\ b_2 & d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ v_1 & b_3 & d_3 & a_3 & \ddots & & \vdots \\ v_2 & 0 & b_4 & d_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ v_{n-3} & \vdots & & \ddots & b_{n-1} & d_{n-1} & a_{n-1} \\ v_{n-2} & 0 & \cdots & \cdots & 0 & b_n & d_n \end{bmatrix}$$

$$= \begin{bmatrix} d_1 & a_1 & 0 & \cdots & \cdots & 0 \\ b_2 & d_2 & a_2 & \ddots & & \vdots \\ 0 & b_3 & d_3 & a_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \cdots & \cdots & 0 & b_n & d_n \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ v_1 & 0 \\ v_2 & 0 \\ \vdots & \vdots \\ v_{n-2} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & h_1 & h_2 & \cdots & h_{n-2} \end{bmatrix} \tag{29}$$

$$= T + UV^t,$$

$$U = [\boldsymbol{u}_1, \boldsymbol{u}_2], V = [\boldsymbol{v}_1, \boldsymbol{v}_2]$$

Applying the Sherman-Morrison-Woodbury formula [13] to A gives

$$A^{-1} = \left(T + UV^t\right)^{-1} = T^{-1} - T^{-1}U\left(I + V^t T^{-1} U\right)^{-1} V^t T^{-1}. \tag{30}$$

Now we are ready to formulate the following symbolic algorithm for inverting the DBT matrix.

Algorithm 3.3. A symbolic algorithm to compute the inverse of a non-singular DBT matrix.

To invert a general DBT matrix (2), we may proceed as follows:

```
INPUT: The order of the matrix, n and the entries of the matrices T, U and V.
OUTPUT: The inverse of the DBT matrix A given in (2).
Step 1: Set $e_n = d_n$. If $e_n = 0$ then $e_n = t$. (t is just a symbolic name)
Step 2: For $i = n-1, n-2, \cdots, 1$ do
            compute and simplify:
                $e_i = d_i - \frac{a_i}{e_{i+1}} b_{i+1}$. If $e_i = 0$ then $e_i = t$ End if
```

End do.

Step 3: Compute and simplify the elements of the inverse matrix T^{-1} using (16).

Step 4: compute A^{-1} using (30).

Step 5: Substitute t = 0 in all expressions of the matrix A^{-1}.

3.3. Solving Linear System of Equations of DBT Matrix Type

In this sub-section, we introduce different approaches for solving doubly bordered tridiagonal linear systems of the form (1).

The inverse matrix of the DBT matrix A introduced in (2) could be written in the block form as follows:

$$B = A^{-1} = \begin{bmatrix} \tilde{d} & \tilde{\boldsymbol{h}} \\ \tilde{\boldsymbol{v}} & \tilde{T} \end{bmatrix} \tag{31}$$

where $\tilde{d} = \dfrac{1}{d_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{v}}$, $\tilde{\boldsymbol{h}} = \dfrac{-\boldsymbol{h}^t\hat{T}^{-1}}{d_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{v}}$, $\tilde{\boldsymbol{v}} = \dfrac{-\hat{T}^{-1}\boldsymbol{v}}{d_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{v}}$ and $\tilde{T} = \left(d_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{v}\right)\tilde{\boldsymbol{v}}\tilde{\boldsymbol{h}} + \hat{T}^{-1}$.

Now, we can formulate our first algorithm for solving the linear systems of the form (1).

Algorithm 3.4. A first numeric algorithm for solving linear systems of DBT matrix type.

To solve a general bordered tridiagonal linear system of the form (1), we may proceed as follows:

INPUT: The entries of the matrix $\hat{T}$, the value of d_1 and the vectors $\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{f}$.

OUTPUT: The solution vector $\boldsymbol{x} = \left[x_1 \;\; x_2\right]^t$.

Step 1: Compute $x_1 = \left(d_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{v}\right)^{-1}\left(\boldsymbol{f}_1 - \boldsymbol{h}^t\hat{T}^{-1}\boldsymbol{f}_2\right)$.

Step 2: Compute $\boldsymbol{x}_2 = \hat{T}^{-1}\left(\boldsymbol{f}_2 - \boldsymbol{v}\,\boldsymbol{x}_1\right)$.

Step 3: The solution vector is $\boldsymbol{x} = \left[x_1 \;\; x_2\right]^t$.

As we can see, the computational cost of Algorithm (3.4) is $O\left(n^2\right)$.

The following algorithm depends on the *UL* factorization of the coefficient matrix A in (1), so it is more convenient to rewrite the linear system (1) as:

$$\left[\begin{array}{c|c} 1 & \boldsymbol{q}^t \\ \hline \boldsymbol{0} & \hat{U} \end{array}\right]\left[\begin{array}{c|c} e_1 & \boldsymbol{0} \\ \hline \boldsymbol{p} & \hat{L} \end{array}\right]\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f}_1 \\ \hline \boldsymbol{f}_2 \end{array}\right]. \tag{32}$$

To solve the linear system (32), it is equivalent to solve the two standard linear systems:

$$\left[\begin{array}{c|c} 1 & \boldsymbol{q}^t \\ \hline \boldsymbol{0} & \hat{U} \end{array}\right]\left[\begin{array}{c} \boldsymbol{w}_1 \\ \hline \boldsymbol{w}_2 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{f}_1 \\ \hline \boldsymbol{f}_2 \end{array}\right] \tag{33}$$

and

$$\left[\begin{array}{c|c} e_1 & \boldsymbol{0} \\ \hline \boldsymbol{p} & \hat{L} \end{array}\right]\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{w}_1 \\ \hline \boldsymbol{w}_2 \end{array}\right] \tag{34}$$

where $\boldsymbol{w}_1 = \left[w_1\right]$ and $\boldsymbol{w}_2 = \left[w_2 \quad w_3 \quad \cdots \quad w_n\right]^t$.

Armed with the above results, we may formulate the following numeric algorithm:

Algorithm 3.5. A second numeric algorithm for solving linear systems of DBT matrix type.

To solve a general doubly bordered tridiagonal linear system of the form (1), we may proceed as follows:

INPUT: The components, a_i, b_i, d_i, h_i, v_i and f_i.

OUTPUT: The solution vector $\boldsymbol{x} = \left[x_1 \; x_2 \; \cdots \; x_n\right]^t$ of the linear system in (1).

Step1: Use the algorithm (3.1) to check the non-singularity of the coefficient matrix A of the system (1).

Step2: If Det (A) = 0; then OUTPUT (‘no inverse exists’); STOP.

Step 3: Set $w_n = f_n$.

Step 4: For $i = n-1, n-2, \cdots, 2$ do
    compute $w_i = f_i - y_i w_{i+1}$
    End do.

```
Step 5: Compute $w_1 = f_1 - \sum_{i=1}^{n-1} q_i w_{i+1}$.
Step 6: Set $x_1 = w_1 / e_1$ .
Step 7: Compute $x_2 = (w_2 - p_1 x_1) / e_2$ ,
        For $i = 3, 4, \cdots, n$ do
            compute $x_i = (w_i - p_{i-1} x_1 - b_i x_{i-1}) / e_i$
        End do.
Step 8: The solution vector $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^t$ .
```

To remove all cases in which Algorithm (3.5) fails, it is convenient to give the following symbolic version of the numeric algorithm described above.

Algorithm 3.6. A symbolic algorithm for solving linear systems of DBT matrix type.

To solve a general doubly bordered tridiagonal linear system of the form (1), we may proceed as follows:

INPUT: The components, a_i, b_i, d_i, h_i, v_i and f_i .

OUTPUT: The solution vector $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^t$ of the linear system in (1).

```
Step 1: Use the DETGDBTRI algorithm to compute $e_i,\ i = 1, \cdots, n$ .
Step 2: Use the DETGDBTRI algorithm to check the non-singularity of the coefficient matrix $A$ of the system (1).
Step 3: If Det($A$) = 0; then OUTPUT ('no inverse exists'); STOP.
Step 4: Compute and simplify $q_i,\ i = 1, \cdots, n-1$ using (25).
Step 5: Compute and simplify $p_i,\ i = 1, \cdots, n-1$ using (26).
Step 6: Set $w_n = f_n$ .
Step 7: Compute and simplify $y_i,\ i = 2, \cdots, n-1$ using (28).
Step 8: For $i = n-1, n-2, \cdots, 2$ do
            Compute and simplify:
                $w_i = f_i - y_i w_{i+1}$
        End do
Step 9: Compute and simplify:
            $w_1 = f_1 - \sum_{i=1}^{n-1} q_i w_{i+1}$.
Step 10: Set $x_1 = w_1 / e_1$ .
Step 11: Compute and simplify:
            $x_2 = (w_2 - p_1 x_1) / e_2$ .
         For $i = 3, 4, \cdots, n$ do
             Compute and simplify:
                 $x_i = (w_i - p_{i-1} x_1 - b_i x_{i-1}) / e_i$ .
         End do.
Step 12: Substitute $t$ = 0 in all expressions of the solution vector $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^t$ .
```

The computational cost of Algorithm (3.6) is $11n - 16$ multiplications/divisions and $8n - 13$ additions/subtractions. The Algorithm (3.6) will be referred to as **DBTLSys** algorithm.

4. Illustrative Examples

In this section we give four examples for the sake of illustration.

Example 4.1. Solve the following periodic tridiagonal linear system of size 12.

$$\left[\begin{array}{c|ccccc} 2 & -1 & 0 & \cdots & 0 & 1 \\ \hline -1 & 2 & -1 & \ddots & \ddots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & -1 & 2 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & -1 \\ 1 & 0 & \cdots & 0 & -1 & 2 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{12} \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 2 \end{bmatrix} \tag{35}$$

***Solution*:** we will solve this system as a DBT linear system where

$$\begin{aligned} &n=12,\ \boldsymbol{d}=\begin{bmatrix}2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2\end{bmatrix},\\ &\boldsymbol{h}^t=\begin{bmatrix}-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix},\\ &\boldsymbol{v}=\begin{bmatrix}-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^t,\\ &\boldsymbol{a}=\begin{bmatrix}-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\end{bmatrix},\\ &\boldsymbol{b}=\begin{bmatrix}-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\end{bmatrix}\\ &\text{and } \boldsymbol{f}=\begin{bmatrix}2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\end{bmatrix}. \end{aligned}$$

By applying DBTLSys algorithm, we have

- $\boldsymbol{e}=\begin{bmatrix}\frac{1}{3} & \frac{12}{11} & \frac{11}{10} & \frac{10}{9} & \frac{9}{8} & \frac{8}{7} & \frac{7}{6} & \frac{6}{5} & \frac{5}{4} & \frac{4}{3} & \frac{3}{2} & 2\end{bmatrix}.$
- $\operatorname{Det}(A)=\prod_{r=1}^{12} e_r = 4.$
- The solution vector is $\boldsymbol{x}=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}^t$.

This result is in complete agreement with the result in [14].

Example 4.2. Solve the following DBT linear system

$$\begin{bmatrix} 5 & 2 & 2 & 6 & 3 & 5 & 2 & 7 & 12 & 4\\ 2 & 1 & 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0\\ 5 & 2 & 5 & 2 & 0 & & & & & 0\\ 4 & 0 & 1 & 2 & 7 & 0 & & & & \vdots\\ 1 & 0 & 0 & 3 & 10 & 2 & \ddots & & & \\ -6 & \vdots & & 0 & 1 & 15 & 3 & 0 & & \vdots\\ 7 & \vdots & & & \ddots & 9 & 2 & 5 & 0 & 0\\ 2 & \vdots & & & & \ddots & 1 & 1 & 7 & 0\\ 2 & \vdots & & & & & 0 & 3 & 1 & 1\\ 3 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8\\ x_9\\ x_{10}\end{bmatrix} = \begin{bmatrix}34\\ 5\\ 12\\ 3\\ 0\\ 18\\ 32\\ 7\\ 9\\ 4\end{bmatrix} \tag{36}$$

***Solution*:** we have a DBT linear system with

$$\begin{aligned} &n=10,\ \boldsymbol{d}=\begin{bmatrix}5 & 1 & 5 & 2 & 10 & 15 & 2 & 1 & 1 & 1\end{bmatrix},\\ &\boldsymbol{a}=\begin{bmatrix}1 & 2 & 7 & 2 & 3 & 5 & 7 & 1\end{bmatrix},\\ &\boldsymbol{b}=\begin{bmatrix}2 & 1 & 3 & 1 & 9 & 1 & 3 & 1\end{bmatrix},\\ &\boldsymbol{h}=\begin{bmatrix}2 & 2 & 6 & 3 & 5 & 2 & 7 & 12 & 4\end{bmatrix},\\ &\boldsymbol{v}=\begin{bmatrix}2 & 5 & 4 & 1 & -6 & 7 & 2 & 2 & 3\end{bmatrix}^t\\ &\text{and } \boldsymbol{f}=\begin{bmatrix}34 & 5 & 12 & 3 & 0 & 18 & 32 & 7 & 9 & 4\end{bmatrix}^t \end{aligned}$$

By using DBTLSys algorithm we have

$\boldsymbol{e}=$

- $\begin{bmatrix}-\frac{15048t+163819}{105(16t+17)} & \frac{5(16t+17)}{88t+107} & \frac{(88t+107)}{4t+11} & \frac{-(4t+11)}{2(17t+13)} & \frac{14(17t+13)}{3(8t+7)} & \frac{3(8t+7)}{t+14} & \frac{-3(t+14)}{t-21} & \frac{t-21}{t} & t & 1\end{bmatrix}.$

  $\operatorname{Det}(A)=\prod_{r=1}^{10} e_r\Big|_{t=0} = -163819.$
- The solution vector is $\boldsymbol{x}=\begin{bmatrix}1 & 2 & 1 & -1 & 0 & 1 & 3 & 2 & 0 & 1\end{bmatrix}^t$.

This result is in complete agreement with the result in [6].

Example 4.3. Solve the following DBT linear system

$$\begin{bmatrix} 1 & 1 & 1 & 5 & 1 & 1 & 1 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ 2 & 1 & 3 & 3 & 0 & & & & & 0 \\ 3 & 0 & 4 & 6 & 2 & \ddots & & & & \vdots \\ 1 & 0 & 0 & 2 & 2 & 1 & \ddots & & & \vdots \\ 2 & \vdots & & \ddots & 3 & 1 & -2 & \ddots & & \vdots \\ 4 & \vdots & & & \ddots & 1 & 5 & 1 & 0 & 0 \\ 5 & \vdots & & & & \ddots & -3 & 2 & 4 & 0 \\ 1 & 0 & & & & & 0 & 2 & 1 & 1 \\ 6 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & 5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \\ \vdots \\ \\ \vdots \\ \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 10 \\ 7 \\ 4 \\ 3 \\ 4 \\ 4 \\ 18 \\ 5 \\ 6 \\ 13 \end{bmatrix} \tag{37}$$

***Solution*:** we have a DBT linear system with

$$\begin{aligned} & n = 10,\ \boldsymbol{d} = [1 \quad 1 \quad 3 \quad 6 \quad 2 \quad 1 \quad 5 \quad 2 \quad 1 \quad 1], \\ & \boldsymbol{a} = [3 \quad 3 \quad 2 \quad 1 \quad -2 \quad 1 \quad 4 \quad 1], \\ & \boldsymbol{b} = [1 \quad 4 \quad 2 \quad 3 \quad 1 \quad -3 \quad 2 \quad 5], \\ & \boldsymbol{h} = [1 \quad 1 \quad 5 \quad 1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 1], \\ & \boldsymbol{v} = [2 \quad 2 \quad 3 \quad 1 \quad 2 \quad 4 \quad 5 \quad 1 \quad 6]^t \\ & \text{and } \boldsymbol{f} = [10 \quad 7 \quad 4 \quad 3 \quad 4 \quad 4 \quad 18 \quad 5 \quad 6 \quad 13]^t \end{aligned}$$

By using DBTLSys algorithm we have

- $\boldsymbol{e} = \left[\frac{-127}{28} \quad \frac{-14}{69} \quad \frac{207}{83} \quad \frac{166}{7} \quad \frac{-7}{31} \quad \frac{31}{23} \quad \frac{23}{4} \quad 4 \quad -4 \quad 1 \right].$
- $\mathrm{Det}(A) = \prod_{r=1}^{10} e_r = 1524.$
- The solution vector is $\boldsymbol{x} = [1 \quad 2 \quad 1 \quad -1 \quad 1 \quad 3 \quad 2 \quad 1 \quad 1 \quad 2]^t$.

Example 4.4. Solve the following DBT linear system

$$\begin{bmatrix} 4 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 2 & 0 & 0 & 0 \\ 2 & 1 & 1 & -1 & 0 & 0 \\ 3 & 0 & 3 & 3 & 3 & 0 \\ 4 & 0 & 0 & 1 & 5 & 4 \\ 5 & 0 & 0 & 0 & 5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 12 \\ 14 \\ 12 \\ 18 \\ 4 \\ 7 \end{bmatrix} \tag{38}$$

***Solution*:** we have a DBT linear system with

$$\begin{aligned} & n = 6,\ \boldsymbol{d} = [4 \quad 2 \quad 1 \quad 3 \quad 5 \quad 4],\ \boldsymbol{a} = [2 \quad -1 \quad 3 \quad 4], \\ & \boldsymbol{b} = [1 \quad 3 \quad 1 \quad 5],\ \boldsymbol{h} = [1 \quad 2 \quad 3 \quad 4 \quad 5], \\ & \boldsymbol{v} = [1 \quad 2 \quad 3 \quad 4 \quad 5]^t \text{ and } \boldsymbol{f} = [12 \quad 14 \quad 12 \quad 18 \quad 4 \quad 7]^t. \end{aligned}$$

By using DBTLSys algorithm we have

- $\boldsymbol{e} = \left[-\frac{6t-13}{8t} \quad \frac{2t}{2t-1} \quad \frac{2t-1}{t-1} \quad \frac{3(t-1)}{t} \quad t \quad 4 \right].$
- $\mathrm{Det}(A) = \prod_{r=1}^{6} e_r \Big|_{t=0} = 39.$

- The solution vector is $\boldsymbol{x} = [4 \quad 3 \quad 2 \quad 1 \quad -1 \quad -2]^t$.

References

[1] Udala, A., Reedera, R., Velmrea, E. and Harrisonb, P. (2006) Comparison of Methods for Solving the Schrödinger Equation for Multiquantum Well Heterostructure Applications. *Proceedings of the Estonian Academy of Sciences*, *Engineering*, **12**, 246-261.

[2] Pajic, S. (2007) Power System State Estimation and Contingency Constrained Optimal Power Flow—A Numerically Robust Implementation. Polytechnic Ins. [S.l.], Worcester.

[3] Mazilu, I., Mazilu, A.D. and Williams, H.T. (2012) Applications of Tridiagonal Matrices in Non-Equilibrium Statistical Physics. *Electronic Journal of Linear Algebra*, **24**, 7-17.

[4] Borowska, J., Łacińska, L. and Rychlewska, J. (2012) Application of Difference Equation to Certain Tridiagonal Matrices. *Scientific Research of the Institute of Mathematics and Computer Science*, **3**, 15-20.

[5] Fischer, C.F. and Usmani, R.A. (1969) Properties of Some Tridiagonal Matrices and Their Application to Boundary Value Problems. *Society for Industrial and Applied Mathematics*, **6**, 127-142. http://dx.doi.org/10.1137/0706014

[6] El-Mikkawy, M. and Atlan, F. (2014) Algorithms for Solving Doubly Bordered Tridiagonal Linear Systems. *British Journal of Mathematics and Computer Science*, **4**, 1246-1267. http://dx.doi.org/10.9734/BJMCS/2014/8835

[7] Al-Hassan, Q. (2005) An Algorithm for Computing Inverses of Tridiagonal Matrices with Applications. *Soochow Journal of Mathematics*, **31**, 449-466.

[8] El-Mikkawy, M.E.A. (2004) On the Inverse of a General Tridiagonal Matrix. *Applied Mathematics and Computation*, **150**, 669-679. http://dx.doi.org/10.1016/S0096-3003(03)00298-4

[9] Ran, R.S., Huang, T.Z., Liu, X.P. and Gu, T.X. (2009) An Inversion Algorithm for General Tridiagonal Matrix. *Applied Mathematics and Mechanics* (*English Edition*), **30**, 247-253. http://dx.doi.org/10.1007/s10483-009-0212-x

[10] Hadj, A.D. and Elouafi, M. (2008) A Fast Numerical Algorithm for the Inverse of a Tridiagonal and Pentadiagonal Matrix. *Applied Mathematics and Computation*, **202**, 441-445. http://dx.doi.org/10.1016/j.amc.2008.02.026

[11] El-Mikkwy, M. and Rahmo, E.-D. (2008) A New Recursive Algorithm for Inverting General Tridiagonal and Anti-Tridiagonal Matrices. *Applied Mathematics and Computation*, **204**, 368-372. http://dx.doi.org/10.1016/j.amc.2008.06.053

[12] Davis, T.A. (2006) Direct Methods for Sparse Linear Systems. University of Florida, Gainesville. http://dx.doi.org/10.1137/1.9780898718881

[13] Golub, G.H. and Van Loan, C.F. (2013) Matrix Computations. 4rd Edition, Johns Hopkins Press, Baltimore.

[14] EL-Mikkawy, M. (2005) A New Computational Algorithm for Solving Periodic Tri-Diagonal Linear Systems. *Applied Mathematics and Computation*, **161**, 691-696. http://dx.doi.org/10.1016/j.amc.2003.12.114

[15] EL-Mikkawy, M. (2012) A Generalized Symbolic Thomas Algorithm. *Applied Mathematics*, **3**, 342-345. http://dx.doi.org/10.4236/am.2012.34052

[16] Alexandre, M. and Iain, D.B. (2010) Variant of the Thomas Algorithm for Opposite-Bordered Tridiagonal Systems of Equations. *Numerical Methods in Biomedical Engineering*, **26**, 752-759.

[17] Strang, G. (2005) Linear Algebra and Its Applications. 4th Edition, Cengage Learning, Boston.

[18] Reddy, B.R. (2012) Recursive Method for Inversion of Lower Triangular Matrix. *Journal of Experimental Sciences*, **3**, 2218-1768.

[19] Wang, X.-B. (2009) A New Algorithm with Its Scilab Implementation for Solution of Bordered Tridiagonal Linear Equations. *IEEE International Workshop on Open-Source Software for Scientific Computation* (*OSSC*), Guiyang, 18-20 September 2009, 11-14.

[20] Amodio, P., Gladwell, I. and Romanazzi, G. (2007) An Algorithm for the Solution of Bordered ABD Linear Systems Arising from Boundary Value Problems. *ECCOMAS Thematic Conference*, Milano, 25-28 June 2007, 25-28.

[21] Amodio, P., Gladwelly, I. and Romanazzi, G. (2006) Numerical Solution of General Bordered ABD Linear Systems by Cyclic Reduction. *Numerical Analysis*, *Industrial and Applied Mathematics*, **1**, 5-12.

[22] El-Mikkawy, M. (2003) A Note on a Three-Term Recurrence for a Tridiagonal Matrix. *Applied Mathematics and Computation*, **139**, 503-511. http://dx.doi.org/10.1016/S0096-3003(02)00212-6

[23] Stoer, J. and Bulirsch, R (1992) Introduction to Numerical Analysis. 2nd Edition, Springer Verlag, New York.