***Applied Mathematics*, 2014, 5, 413-422**
Published Online February 2014 (http://www.scirp.org/journal/am)
http://dx.doi.org/10.4236/am.2014.53042

Algorithms for Solving Linear Systems of Equations of Tridiagonal Type via Transformations

Moawwad El-Mikkawy, Faiz Atlan
Mathematics Department, Faculty of Science, Mansoura University, Mansoura, Egypt
Email: m_elmikkawy@yahoo.com, faizatlan11@yahoo.com

Received October 9, 2013; revised November 9, 2013; accepted November 16, 2013

ABSTRACT

Numeric algorithms for solving the linear systems of tridiagonal type have already existed. The well-known Thomas algorithm is an example of such algorithms. The current paper is mainly devoted to constructing symbolic algorithms for solving tridiagonal linear systems of equations via transformations. The new symbolic algorithms remove the cases where the numeric algorithms fail. The computational cost of these algorithms is given. MAPLE procedures based on these algorithms are presented. Some illustrative examples are given.

KEYWORDS

Tridiagonal Matrix; Permutation Matrix; Algorithm; MAPLE

1. Introduction

Linear systems of equations of tridiagonal type arise in solving problems in a wide variety of disciplines including physics [1,2], mathematics [3-8], engineering [9,10] and others. Many researchers have been devoted to dealing with such systems (see [11-27]). When a system of linear equations has a coefficient matrix of special structure, it is recommended to use a tailor-made algorithm for such systems of equations. The tailor-made algorithms are not only more efficient in terms of computational time and computer memory, but also accumulate smaller round-off errors. As a matter of fact, many problems arising in practice lead to the solution of linear system of equations with special coefficient matrices. The current paper is mainly devoted to developing new algorithms for solving linear system of equations of tridiagonal type of the form:

$$T\boldsymbol{x} = \boldsymbol{f}, \tag{1}$$

where

$$T = \begin{bmatrix} d_1 & a_1 & 0 & \cdots & 0 \\ b_2 & d_2 & a_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & \cdots & 0 & b_n & d_n \end{bmatrix}, \tag{2}$$

$$\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}} \text{ and } \boldsymbol{f} = [f_1, f_2, \cdots, f_n]^{\mathrm{T}}.$$

The coefficient matrix T in (2) can be stored in $3n$ memory locations by using three vectors: $\boldsymbol{a} = [a_1, a_2, \cdots, a_n]$, $\boldsymbol{b} = [b_1, b_2, \cdots, b_n]$, and $\boldsymbol{d} = [d_1, d_2, \cdots, d_n]$, with $a_n = b_1 = 0$. This is always a good habit in computation in order to save memory space.

Of course, the non-singularity of the coefficient matrix should be checked firstly to make sure that the system

(1) has a non-trivial solution. The **DETGTRI** algorithm [28] can be used efficiently for this purpose.

Definition 1.1 [29]. The symmetric matrix $A=\left(a_{ij}\right)_{i,j=1}^{n}$ is called positive definite if and only if

$$\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}>0,\ \text{for all } x\in\mathbb{R}^{n},\ x\neq 0.$$

Theorem 1.2 [29]. The symmetric matrix $A\in\mathbb{R}^{n\times n}$ is positive definite if and only if any of the following conditions is satisfied:

1) A has only positive eigenvalues.

2) $F_k=\begin{vmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{vmatrix}>0,\quad \text{for } k=1,2,\cdots,n.$

In particular, the author in [30] proved that for the tridiagonal matrix (2), it is true that $F_k=c_kF_{k-1},\ k=1,2,\cdots,n,$ provided that $F_0=1.$ Thus the tridiagonal matrix (2) is positive definite if and only if $c_i>0,\ i=1,2,\cdots,n.$ This is an easy way to check weather a tridiagonal matrix is positive definite or not.

3) A can be written as: $A=B^{\mathrm{T}}B$ for a non-singular matrix $B\in\mathbb{R}^{n\times n}$.

Definition 1.3 [29]. An $n\times n$ matrix A is called diagonally dominant if

$$\left|a_{ii}\right|\geq\sum_{\substack{j=1,\\ j\neq i}}^{n}\left|a_{ij}\right|,\quad 1\leq i\leq n,$$

and strictly diagonally dominant if

$$\left|a_{ii}\right|>\sum_{\substack{j=1,\\ j\neq i}}^{n}\left|a_{ij}\right|,\quad 1\leq i\leq n.$$

The current paper is organized as follows. In Section 2, new algorithms for solving linear systems of equations of tridiagonal type via transformations are given. In Section 3, concluding remarks are given. MAPLE procedures are given in Section 4. Illustrative examples are presented in Section 5.

Throughout this paper, the word “simplify” means simplifying the expression under consideration to its simplest rational form.

2. Main Results

In this Section, we are going to consider the derivation of new algorithms for solving linear systems of equations of tridiagonal type (1) via transformations. For this purpose it is convenient to introduce three vectors $\boldsymbol{c}=\left[c_1,c_2,\cdots,c_n\right]$, $\boldsymbol{y}=\left[y_1,y_2,\cdots,y_n\right]$ and $\boldsymbol{z}=\left[z_1,z_2,\cdots,z_n\right]$ where

$$c_1=d_1,\ y_1=\frac{a_1}{c_1},\ z_1=\frac{f_1}{c_1}\ \text{ and }\ \begin{cases} c_i=d_i-b_iy_{i-1}, \\ y_i=\dfrac{a_i}{c_i}, \\ z_i=\dfrac{1}{c_i}\left(f_i-b_iz_{i-1}\right). \end{cases}\qquad \text{for } i=2,3,\cdots,n \tag{3}$$

By using the vectors $\boldsymbol{c}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, together with the suitable elementary row operations (ERO’s), we see that the system (1) may be transformed to the equivalent linear system:

$$\begin{bmatrix} 1 & y_1 & 0 & \cdots & 0 \\ 0 & 1 & y_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & y_{n-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}=\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{bmatrix}. \tag{4}$$

The transformed system (4) is easy to solve by backward substitution. Consequently, the linear system (1) can be solved using the following algorithm:

Algorithm 2.1. Numeric algorithm for solving tridiagonal linear system.

```
To solve the linear system of the form (1), we may proceed as follows:
INPUT: Order of the matrix $n$ and the components, $a_i, d_i, b_i, f_i, i = 1, 2, \cdots, n,$ $(a_n = b_1 = 0)$.
OUTPUT: The solution vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$.
Step 1: Use the DETGTRI algorithm [28] to check the non-singularity of the coefficient matrix of the system (1).
Step 2: If $\det(T) = 0$, then Exiterror (‘”No solutions”) end if.
Step 3: Set $c_1 = d_1,$ $y_1 = \frac{a_1}{c_1}$ and $z_1 = \frac{f_1}{c_1}$.
Step 4: For $i = 2, 3, \cdots, n-1$ do
            Compute and simplify:
            $c_i = d_i - b_i y_{i-1},$
            $y_i = \frac{a_i}{c_i},$
            $z_i = \frac{1}{c_i}(f_i - b_i z_{i-1}).$
        End do.
        $c_n = d_n - b_n y_{n-1}.$
        $z_n = \frac{1}{c_n}(f_n - b_n z_{n-1}).$
Step 5: Compute the solution vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$ using
        $x_n = z_n,$
        For $i = n-1, n-2, \cdots, 1$ do
            $x_i = z_i - y_i x_{i+1}$
        End do.
```

The **Algorithm 2.1**, will be referred to as **TRANSTRI-I** algorithm. The cost of the algorithm is $5n-4$ multiplications/divisions and $3n-3$ additions/subtractions.

Note that the algorithm **TRANSTRI-I** works properly only if $c_i \neq 0$ for all $i \in \{1, 2, \cdots, n\}$.

At this point, it should be mentioned that if the coefficient matrix, T of the system (1) is positive definite or diagonally dominant, then the numeric algorithm **TRANSTRI-I** will never fail.

The following symbolic version algorithm is developed in order to remove the cases where the numeric algorithm **TRANSTRI-I** fails. The parameter "*s*" in the algorithm is just a symbolic name. It is a dummy argument and its actual value is zero.

Algorithm 2.2. Symbolic version algorithm for TRANSTRI-I algorithm.

```
To solve the linear system of the form (1), we may proceed as follows:
INPUT: Order of the matrix $n$ and the components, $a_i, d_i, b_i, f_i, i = 1, 2, \cdots, n,$ $(a_n = b_1 = 0)$.
OUTPUT: The solution vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$.
Step 1: Use the DETGTRI algorithm [28] to check the non-singularity of the coefficient matrix of the system (1).
Step 2: If $\det(T) = 0$, then Exiterror(“No solutions”) end if.
Step 3: Set $c_1 = d_1$. If $c_1 = 0$ then $c_1 = s$ end if.
        Set $y_1 = \frac{a_1}{c_1}$ and $z_1 = \frac{f_1}{c_1}$.
Step 4: For $i = 2, 3, \cdots, n-1$ do
            Compute and simplify:
            $c_i = d_i - b_i y_{i-1}$. If $c_i = 0$ then $c_i = s$ end if.
            $y_i = \frac{a_i}{c_i},$
        $z_i = \frac{1}{c_i}\left( f_i - b_i z_{i-1}\right).$
    End do.
    $c_n = d_n - b_n y_{n-1}$.  If  $c_n = 0$  then  $c_n = s$  end if.
    $z_n = \frac{1}{c_n}\left( f_n - b_n z_{n-1}\right).$
Step 5: Compute the solution vector  $\boldsymbol{x} = \left[x_1, x_2, \cdots, x_n\right]^{\mathrm{T}}$  using
    $x_n = z_n,$
    For  $i = n-1, n-2, \cdots, 1$  do
        $x_i = z_i - y_i x_{i+1}$
    End do.
Step 6: Substitute  $s = 0$  in all expressions of the solution vector  $x_i, i = 1, 2, \cdots, n.$
```

The **Algorithm 2.2**, will be referred to as **TRANSTRI-II** algorithm.

In a similar manner, we may consider three vectors $\boldsymbol{e} = \left[e_1, e_2, \cdots, e_n\right]$, $\boldsymbol{Y} = \left[Y_1, Y_2, \cdots, Y_n\right]$ and $\boldsymbol{Z} = \left[Z_1, Z_2, \cdots, Z_n\right]$ where

$$e_n = d_n,\ Y_n = \frac{b_n}{e_n},\ Z_n = \frac{f_n}{e_n} \text{ and } \begin{cases} e_i = d_i - a_i Y_{i+1}, \\ Y_i = \dfrac{b_i}{e_i}, \\ Z_i = \dfrac{1}{e_i}\left(f_i - a_i Z_{i+1}\right). \end{cases} \quad \text{for } i = n-1, n-2, \cdots, 1 \tag{5}$$

in order to develop a new algorithm.

We are going to focus on the symbolic version only. As in **Algorithm 2.1**, by using the vectors ***e***, ***Y*** and ***Z***, together with the suitable ERO's, we see that the system (1) may be transformed to the equivalent linear system:

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ Y_2 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & Y_n & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ \vdots \\ Z_n \end{bmatrix}. \tag{6}$$

The transformed system (6) is easy to solve using forward substitution. Therefore the linear system (1) can be solved using the following algorithm:

Algorithm 2.3. Symbolic version algorithm for solving tridiagonal linear system.

```
To solve the linear system of the form (1), we may proceed as follows:
INPUT: Order of the matrix  $n$  and the components,  $a_i, d_i, b_i, f_i, i = 1, 2, \cdots, n$,  $\left(a_n = b_1 = 0\right)$.
OUTPUT: The solution vector  $\mathbf{x} = \left[x_1, x_2, \cdots, x_n\right]^{\mathrm{T}}$ .
Step 1: Use the DETGTRI algorithm [28] to check the non-singularity of the coefficient matrix of the system (1).
Step 2: If  $\det\left(T\right) = 0$ , then Exiterror("No solutions") end if.
Step 3: Set  $e_n = d_n$.  If  $e_n = 0$  then  $e_n = s$  end if.
    $Y_n = \frac{b_n}{e_n}$  and  $Z_n = \frac{f_n}{e_n}$.
Step 4: For  $i = n-1, n-2, \cdots, 2$  do
        Compute and simplify:
        $e_i = d_i - a_i Y_{i+1}$.  If  $e_i = 0$  then  $e_i = s$  end if.
        $Y_i = \frac{b_i}{e_i},$
        $Z_i = \frac{1}{e_i}\left( f_i - a_i Z_{i+1} \right).$
    End do.
    $e_1 = d_1 - a_1 Y_2$.  If  $e_1 = 0$  then  $e_1 = s$  end if.
    $Z_1 = \frac{1}{e_1}\left( f_1 - a_1 Z_2 \right).$
Step 5: Compute the solution vector  $\boldsymbol{x} = \left[ x_1, x_2, \cdots, x_n \right]^{\mathrm{T}}$  using
    $x_1 = Z_1,$
    For  $i = 2, 3, \cdots, n$  do
        $x_i = Z_i - Y_i x_{i-1}$
    End do.
Step 6: Substitute  $s = 0$  in all expressions of the solution vector  $x_i, i = 1, 2, \cdots, n.$
```

The **Algorithm 2.3**, will be referred to as **TRANSTRI-III** algorithm.

Corollary 2.1. Let $\hat{T}$ be the backward matrix of the tridiagonal matrix T in (2), and given by:

$$\hat{T} = \begin{bmatrix} 0 & \cdots & \cdots & 0 & a_1 & d_1 \\ \vdots & & \iddots & a_2 & d_2 & b_2 \\ \vdots & \iddots & a_3 & d_3 & b_3 & 0 \\ 0 & \iddots & \iddots & \iddots & \iddots & \vdots \\ a_{n-1} & d_{n-1} & b_{n-1} & \iddots & & \vdots \\ d_n & b_n & 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{7}$$

Then the backward tridiagonal linear system

$$\hat{T}\left[u_1, u_2, \cdots, u_n \right]^{\mathrm{T}} = \boldsymbol{f} \tag{8}$$

has the solution: $u_i = x_{n+1-i}, i = 1, 2, \cdots, \lfloor n \rfloor$, where $\lfloor k \rfloor$ is the floor function of k and $\left[x_1, x_2, \cdots, x_n \right]^{\mathrm{T}}$ is the solution vector of the linear system (1).

Proof. Consider the $n \times n$ permutation matrix P defined by:

$$P = \begin{bmatrix} 0 & \cdots & \cdots & 0 & 1 \\ \vdots & & \iddots & 1 & 0 \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ 0 & 1 & \iddots & & \vdots \\ 1 & 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{9}$$

For this matrix, we have:

$$P^{-1} = P^{\mathrm{T}} = P. \tag{10}$$

Since

$$\hat{T} = TP \tag{11}$$

Then using (10) and (11), the result follows.

Corollary 2.2. The determinants of the coefficient matrices T and $\hat{T}$ in (2) and (7) are given respectively by:

$$\det\left(T \right) = \prod_{r=1}^{n} c_r = \prod_{r=1}^{n} e_r \tag{12}$$

and

$$\det\left(\hat{T} \right) = \left(-1 \right)^{\frac{n(n-1)}{2}} \prod_{r=1}^{n} c_r = \left(-1 \right)^{\frac{n(n-1)}{2}} \prod_{r=1}^{n} e_r, \tag{13}$$

where $c_1, c_2, \cdots, c_n$ and $e_1, e_2, \cdots, e_n$ satisfy (3) and (5).

3. Conclusions

There are many numeric algorithms in current use for solving linear systems of tridiagonal type. The Thomas algorithm is the well known numeric algorithm for solving such systems. However, all Thomas and Thomas-like numeric algorithms including the **TRANSTRI-I** algorithm of the current paper, fail to solve the tridiagonal linear system if $c_i = 0$ for any $i \in \{1, 2, \cdots, n\}$. For example, all these numeric algorithms fail to solve the linear system:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 18 \end{bmatrix},$$

since $c_2 = 0$, although its coefficient matrix is invertible and its inverse is the following matrix

$$\begin{bmatrix} \frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \\ -\frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

The symbolic algorithms **TRANSTRI-II** and **TRANSTRI-III** of the current paper are constructed in order to remove the cases where the numeric algorithms fail. These are the only symbolic algorithms for solving linear systems of tridiagonal type. Consequently, we are not going to compare them with numeric algorithms.

4. Computer Programs

In this Section, we are going to introduce MAPLE procedures for solving linear system of tridiagonal type (1). These procedures are based on the algorithms **DETGTRI**, **TRANSTRI-II** and **TRANSTRI-III**. The procedure of **Program 1**, alters the contents of the vectors $\boldsymbol{d}$, $\boldsymbol{a}$ and $\boldsymbol{f}$. Eventually, the contents of the vectors $\boldsymbol{c}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ are stored in $\boldsymbol{d}$, $\boldsymbol{a}$ and $\boldsymbol{f}$, respectively. The procedure of **Program 2**, alters the contents of the vectors $\boldsymbol{d}$, $\boldsymbol{a}$ and $\boldsymbol{f}$. Eventually, the contents of the vectors $\boldsymbol{e}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are stored in $\boldsymbol{d}$, $\boldsymbol{b}$ and $\boldsymbol{f}$, respectively.

Program 1. A MAPLE procedure for solving linear system of tridiagonal type.

```
> restart:
tritrans := proc(d::vector, a::vector, b::vector,f::vector,n::posint)
     local i:
     global x,T:
     x:= vector(n):
     if d[1] = 0 then d[1]:=s fi:
     a[1]:=simplify(a[1]/d[1]): f[1]:=simplify(f[1]/d[1]):
     for i from 2 to n-1 do
        d[i] := simplify(d[i]-a[i-1]*b[i]);
        if d[i] = 0 then d[i] := s; fi:
        a[i] := simplify(a[i]/d[i]);
        f[i] := simplify((f[i]-f[i-1]*b[i])/d[i]);
     od:
        d[n] := simplify(d[n]-a[n-1]*b[n]);
        if d[n] = 0 then d[n] := s; fi:
        f[n] := simplify((f[n]-f[n-1]*b[n])/d[n]);
     #To compute the determinant of the tridiagonal matrix#
     T := simplify(subs(s =0,simplify(product(d[r],r= 1..n)))):
     if T = 0 then
        error("Singular Matrix")
     else
     # To compute the Solution of the system X. #
```

```
        x[n]:=simplify(f[n]);
        for i from n-1 by -1 to 1 do
        x[i]:=simplify((f[i]-a[i]*x[i+1]));
        od;
        eval(x);
      fi:
end proc :
```

Program 2. A MAPLE procedure for solving linear system of tridiagonal type.

Based on the algorithm **TRANSTRI-III**, a MAPLE procedure for solving the linear system of tridiagonal type (1) is given below.

```
> restart:
   tritrans := proc(d::vector, a::vector, b::vector,f::vector,n::posint)
        local i:
        global x,T:
        x:= vector(n):
        if d[n] = 0 then d[n]:=s fi:
        b[n]:=simplify(b[n]/d[n]): f[n]:=simplify(f[n]/d[n]):
        for i from n-1 by -1 to 2 do
           d[i] := simplify(d[i]-b[i+1]*a[i]);
           if d[i] = 0 then d[i] := s; fi:
           b[i] := simplify(b[i]/d[i]);
           f[i] := simplify((f[i]-f[i+1]*a[i])/d[i]);
        od:
           d[1] := simplify(d[1]-b[2]*a[1]);
           if d[1] = 0 then d[1] := s; fi:
           f[1] := simplify((f[1]-f[2]*a[1])/d[1]);
        #To compute the determinant of the tridiagonal matrix#
        T := simplify(subs(s =0,simplify(product(d[r],r= 1..n)))):
        if T = 0 then
           error(“Singular Matrix”)
        else
        # To compute the Solution of the system X. #
           x[1]:=simplify(f[1]);
           for i from 2 to n do
           x[i]:=simplify((f[i]-b[i]*x[i-1]));
           od;
           eval(x);
        fi:
end proc :
```

5. Illustrative Examples

All results in this section are obtained by executing the MAPLE procedures of Program 1 and Program 2 presented in the previous section.

Example 5.1. Solve the tridiagonal linear system

$$\begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ -6 \\ 2 \\ 1 \\ 0 \end{bmatrix} \tag{14}$$

Solution: We have

$$n = 7, \quad \boldsymbol{a} = [0,1,1,1,1,1,0]^{\mathrm{T}}, \quad \boldsymbol{d} = [6,4,4,4,4,4,6]^{\mathrm{T}}, \quad \boldsymbol{b} = [0,1,1,1,1,1,0]^{\mathrm{T}},$$

and $\boldsymbol{f} = [0,1,2,-6,2,1,0]^{\mathrm{T}}$.

By applying the **TRANSTRI-I** algorithm, we get

- $\boldsymbol{c} = \left[6,4,\frac{15}{4},\frac{56}{15},\frac{209}{56},\frac{780}{209},6\right]$.
- $\det(T) = \prod_{i=1}^{7} c_i = 28080.$
- The solution vector is given by: $\boldsymbol{x} = [0,0,1,-2,1,0,0]^{\mathrm{T}}$.

Note that the coefficient matrix T in (14) is positive definite.

By applying the algorithms **TRANSTRI-II** and **TRANSTRI-III**, we obtain the same solution vector.

Example 5.2. Solve the tridiagonal linear system

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 2 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 4 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 1 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 2 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 4 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 6 \\ 34 \\ 10 \\ 1 \\ 4 \\ 22 \\ 25 \\ 3 \end{bmatrix}$$

Solution: Here, we have

$$n = 10, \quad \boldsymbol{a} = [1,2,1,6,1,3,5,7,3,0]^{\mathrm{T}}, \quad \boldsymbol{d} = [2,3,3,2,2,4,1,2,4,5]^{\mathrm{T}},$$

$$\boldsymbol{b} = [0,1,1,7,6,3,8,6,5,4]^{\mathrm{T}}, \quad \text{and} \quad \boldsymbol{f} = [1,2,6,34,10,1,4,22,25,3]^{\mathrm{T}}.$$

By applying the **TRANSTRI-I** algorithm, we get

- $\boldsymbol{c} = \left[2,\frac{5}{2},\frac{11}{5},\frac{-13}{11},\frac{422}{13},\frac{1649}{422},\frac{-8479}{1649},\frac{66428}{8479},\frac{-31053}{66428},\frac{317467}{10351}\right]$.
- $\det(T) = \prod_{i=1}^{10} c_i = -952401.$
- The solution vector is given by: $\boldsymbol{x} = [1,-1,2,1,3,-2,0,4,2,-1]^{\mathrm{T}}$.

By using the algorithms **TRANSTRI-II** and **TRANSTRI-III**, we obtain the same solution vector.

Example 5.3. Solve the tridiagonal linear system

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 11 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 3 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 4 \\ 14 \\ 26 \\ 25 \\ 0 \\ 2 \\ 1 \\ 3 \\ 10 \\ 8 \end{bmatrix} \tag{15}$$

Solution: Here, we have

$$n=10,\quad \boldsymbol{a}=[1,10,2,1,7,2,2,1,4,0]^{\mathrm{T}},\quad \boldsymbol{d}=[1,1,1,11,3,1,2,1,2,5]^{\mathrm{T}},$$

$$\boldsymbol{b}=[0,1,7,2,2,3,-1,2,5,1]^{\mathrm{T}},\ \text{and}\quad \boldsymbol{f}=[4,14,26,25,0,2,1,3,10,8]^{\mathrm{T}}.$$

The numeric algorithm **TRANSTRI-I** fails to solve the linear system (15) since $c_2=0$. Applying the **TRANSTRI-II** algorithm, it gives:

- $$\boldsymbol{c}=\Bigg[1,s,\frac{(s-70)}{s},\frac{7\times(s-110)}{(s-70)},\frac{1}{7}\times\frac{(19\times s-2170)}{(s-110)},\frac{-16\times(8\times s-875)}{(19\times s-2170)},\frac{1}{8}\times\frac{(109\times s-11830)}{(8\times s-875)},$$
$$\frac{-147\times(s-110)}{(109\times s-11830)},\frac{1}{147}\times\frac{(839\times s-91490)}{(s-110)},\frac{(3607\times s-392770)}{(839\times s-91490)}\Bigg].$$
- $\det(T)=\left(\prod_{i=1}^{10}c_i\right)_{s=0}=(7214\times s-785540)_{s=0}=-785540.$
- The solution vector is given by:

$$\boldsymbol{x}=\Bigg[\frac{2\times(7214\times s-196385)}{(3607\times s-392770)},\frac{-1178310}{(3607\times s-392770)},\frac{2\times(60719\times s-196385)}{(3607\times s-392770)},\frac{-4\times(3457\times s+196385)}{(3607\times s-392770)},$$
$$\frac{-(593\times s+392770)}{(3607\times s-392770)},\frac{5\times(841\times s+78554)}{(3607\times s-392770)},\frac{2394\times s}{(3607\times s-392770)},\frac{1512\times s}{(3607\times s-392770)},$$
$$\frac{3\times(1507\times s-392770)}{(3607\times s-392770)},\frac{31\times(157\times s-12670)}{(3607\times s-392770)}\Bigg]^{\mathrm{T}}_{s=0}$$
$$=[1,3,1,2,1,-1,0,0,3,1]^{\mathrm{T}}.$$

Using the **TRANSTRI-III** algorithm, it gives the same solution vector.

Acknowledgements

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

REFERENCES

[1] G. Y. Hu and R. F. O'Connell, "Analytical Inversion of Symmetric Tridiagonal Matrices," *Journal of Physics A*, Vol. 29, No. 7, 1996, pp. 1511-1513.

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

[3] J. A. Marrero, M. Rachidi and V. Tomeo, "Non-Symbolic Algorithms for the Inversion of Tridiagonal Matrices," *Journal of Computational and Applied Mathematics*, Vol. 252, 2013, pp. 3-11. http://dx.doi.org/10.1016/j.cam.2012.05.003

[4] Q. Al-Hassan, "On Powers of Tridiagonal Matrices with Nonnegative Entries," *Journal of Applied Mathematical Sciences*, Vol. 6, No. 48, 2012, pp. 2357-2368.

[5] C. M. da Fonseca and J. Petronilho, "Explicit Inverses of Some Tridiagonal Matrices," *Linear Algebra and Its Applications*, Vol. 325, No. 1-3, 2001, pp. 7-21. http://dx.doi.org/10.1016/S0024-3795(00)00289-5

[6] Y. Huang and W. F. McColl, "Analytic Inversion of General Tridiagonal Matrices," *Journal of Physics A*, Vol. 30, No. 22, 1997, pp. 7919-7933. http://dx.doi.org/10.1088/0305-4470/30/22/026

[7] A. Kavcic and J. M. F. Moura, "Matrices with Banded Inverses: Inversion Algorithms and Factorization of Gauss-Markov Processes," *IEEE Transactions on Information Theory*, Vol. 46, No. 4, 2000, pp. 1495-1509. http://dx.doi.org/10.1109/18.954748

[8] H.-B. Li, T.-Z. Huang, X.-P. Liu and H. Li, "On the Inverses of General Tridiagonal Matrices," *Linear Algebra and Its Applications*, Vol. 433, No. 5, 2010, pp. 965-983. http://dx.doi.org/10.1016/j.laa.2010.04.042

[9] A. Martin and I. D. Boyd, "Variant of the Thomas Algorithm for Opposite-Bordered Tridiagonal Systems of Equations," *International Journal for Numerical Methods in Biomedical Engineering*, Vol. 26, No. 6, 2010, pp. 752-759.

[10] E. Olcayto, "Recursive Formulae for Ladder Network Optimization," *Electronics Letters*, Vol. 15, No. 9, 1979, pp. 249-250. http://dx.doi.org/10.1049/el:19790176

[11] T. M. Austin, M. Berndt and J. D. Moulton, "A Memory Efficient Parallel Line Solver," Submitted to SISC, 2004.

[12] J.-J. Climent, L. Tortosa and A. Zamora, "A Note on the Recursive Decoupling Method for Solving Tridiagonal Linear Systems," *Applied Mathematics and Computation*, Vol. 140, No. 1, 2003, pp. 159-164. http://dx.doi.org/10.1016/S0096-3003(02)00218-7

[13] O. Egecioglu, C. K. Koc and A. J. Laub, "A Recursive Doubling Algorithm for Solution of Tridiagonal Systems on Hypercube Multiprocessors," *Journal of Computational and Applied Mathematics*, Vol. 27, No. 1-2, 1989, pp. 95-108. http://dx.doi.org/10.1016/0377-0427(89)90362-2

[14] M. E. A. El-Mikkawy, "An Algorithm for Solving Tridiagonal Systems," *Journal of Institute of Mathematics and Computer Science Computer Science Series*, Vol. 4, No. 2, 1991, pp. 205-210.

[15] M. E. A. El-Mikkawy, "On the Inverse of a General Tridiagonal Matrix," *Applied Mathematics and Computation*, Vol. 150, No. 3, 2004, pp. 669-679. http://dx.doi.org/10.1016/S0096-3003(03)00298-4

[16] M. E. A. El-Mikkawy, "A New Computational Algorithm for Solving Periodic Tri-Diagonal Linear Systems," *Applied Mathematics and Computation*, Vol. 161, No. 2, 2005, pp. 691-696. http://dx.doi.org/10.1016/j.amc.2003.12.114

[17] M. E. A. El-Mikkawy and A. Karawia, "Inversion of General Tridiagonal Matrices," *Applied Mathematics Letters*, Vol. 19, No. 8, 2006, pp. 712-720. http://dx.doi.org/10.1016/j.aml.2005.11.012

[18] M. E. A. El-Mikkawy and E.-D. Rahmo, "A New Recursive Algorithm for Inverting General Tridiagonal and Anti-Tridiagonal Matrices," *Applied Mathematics and Computation*, Vol. 204, No. 1, 2008, pp. 368-372. http://dx.doi.org/10.1016/j.amc.2008.06.053

[19] M. E. A. El-Mikkawy, "A Generalized Symbolic Thomas Algorithm," *Applied Mathematics*, Vol. 3, No. 4, 2012, pp. 342-345. http://dx.doi.org/10.4236/am.2012.34052

[20] D. Fanache, "A Parallel Solution of Tridiagonal Linear Systems by Continued Fractions," *Journal of Arts & Sciences*, Vol. 11, No. 1, 2011, pp. 21-30.

[21] R. K. Mallik, "The Inverse of a Tridiagonal Matrix," *Linear Algebra and Its Applications*, Vol. 325, No. 1-3, 2001, pp. 109-139. http://dx.doi.org/10.1016/S0024-3795(00)00262-7

[22] B. V. Minchev, "Some Algorithms for Solving Special Tridiagonal Block Toeplitz Linear Systems," *Journal of Computational and Applied Mathematics*, Vol. 156, No. 1, 2003, pp. 179-200. http://dx.doi.org/10.1016/S0377-0427(02)00911-1

[23] T. Sogabe, "On a Two-Term Recurrence for the Determinant of a General Matrix," *Applied Mathematics and Computation*, Vol. 187, No. 2, 2007, pp. 785-788. http://dx.doi.org/10.1016/j.amc.2006.08.156

[24] T. Sugimoto, "On an Inverse Formula of a Tridiagonal Matrix," *Operators and Matrices*, Vol. 6, No. 3, 2012, pp. 465-480. http://dx.doi.org/10.7153/oam-06-30

[25] R. Usmani, "Inversion of a Tridiagonal Jacobi Matrix," *Linear Algebra and Its Applications*, Vol. 212-213, 1994, pp. 413-414. http://dx.doi.org/10.1016/0024-3795(94)90414-6

[26] T. Yamamoto and Y. Ikebe, "Inversion of Band Matrices," *Linear Algebra and Its Applications*, Vol. 24, 1979, pp. 105-111. http://dx.doi.org/10.1016/0024-3795(79)90151-4

[27] Y. Zhang, J. Cohen and J. D. Owens, "Fast Tridiagonal Solvers on the GPU," *Proceedings of the* 15*th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP* 2010), Bangalore, 9-14 January 2010, pp. 127-136.

[28] M. E. A. El-Mikkawy, "A Fast Algorithm for Evaluating *n*th Order Tri-Diagonal Determinants," *Applied Mathematics and Computation*, Vol. 202, No. 1, 2008, pp. 210-215. http://dx.doi.org/10.1016/j.amc.2008.01.032

[29] J. W. Demmel, "Applied Numerical Linear Algebra," Society for Industrial and Applied Mathematics, Philadelphia, 1997. http://dx.doi.org/10.1137/1.9781611971446

[30] M. E. A. El-Mikkawy, "A Note on a Three-Term Recurrence for a Tridiagonal Matrix," *Applied Mathematics and Computation*, Vol. 139, No. 2-3, 2003, pp. 503-511. http://dx.doi.org/10.1016/S0096-3003(02)00212-6