

## Research Article

# On Solving Pentadiagonal Linear Systems via Transformations

S. S. Askar<sup>1,2</sup> and A. A. Karawia<sup>2,3</sup>

<sup>1</sup>Department of Statistics and Operations Researches, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia

<sup>2</sup>Department of Mathematics, Faculty of Science, Mansoura University, Mansoura 35516, Egypt

<sup>3</sup>Computer Science Unit, Deanship of Educational Services, Qassim University, P.O. Box 6595, Buraidah 51452, Saudi Arabia

Correspondence should be addressed to S. S. Askar; s.e.a.askar@hotmail.co.uk

Received 4 January 2015; Revised 18 March 2015; Accepted 18 March 2015

Academic Editor: George S. Dulikravich

Copyright © 2015 S. S. Askar and A. A. Karawia. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many authors have studied numerical algorithms for solving the linear systems of pentadiagonal type. The well-known fast pentadiagonal system solver algorithm is an example of such algorithms. The current paper describes new numerical and symbolic algorithms for solving pentadiagonal linear systems via transformations. The proposed algorithms generalize the algorithms presented in El-Mikkawy and Atlan, 2014. Our symbolic algorithms remove the cases where the numerical algorithms fail. The computational cost of our algorithms is better than those algorithms in literature. Some examples are given in order to illustrate the effectiveness of the proposed algorithms. All experiments are carried out on a computer with the aid of programs written in MATLAB.

## 1. Introduction

The pentadiagonal linear systems, denoted by (PLS), take the following forms:

$$PX = Y, \quad (1)$$

where  $P$  is  $n$  – by –  $n$  pentadiagonal matrix given by

$$P = \begin{pmatrix} d_1 & a_1 & b_1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ c_2 & d_2 & a_2 & b_2 & 0 & \dots & \dots & \dots & \dots & 0 \\ e_3 & c_3 & d_3 & a_3 & b_3 & 0 & \dots & \dots & \dots & 0 \\ 0 & e_4 & c_4 & d_4 & a_4 & b_4 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & e_{n-2} & c_{n-2} & d_{n-2} & a_{n-2} & b_{n-2} \\ 0 & \dots & \dots & \dots & \dots & 0 & e_{n-1} & c_{n-1} & d_{n-1} & a_{n-1} \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & e_n & c_n & d_n \end{pmatrix}, \quad n \geq 4, \quad (2)$$

where  $X = (x_1, x_2, \dots, x_n)^t$  and  $Y = (y_1, y_2, \dots, y_n)^t$  are vectors of length  $n$ .

This kind of linear systems is well known in the literature [1–11] and often these types of linear systems are widely used in areas of science and engineering, for example, in numerical solution of ordinary and partial differential equations (ODE and PDE), interpolation problems, boundary value problems (BVP), parallel computing, physics, and matrix algebra [4–14]. The authors in [7] have developed an efficient algorithm to find the inverse of a general pentadiagonal matrix. In [8], the author presented an efficient computational algorithm for solving periodic pentadiagonal linear systems. The algorithm depends on the LU factorization of the periodic pentadiagonal matrix. New algorithms are used for solving periodic pentadiagonal linear systems based on the use of any pentadiagonal linear solver. The author described a symbolic algorithm for solving pentadiagonal linear systems [9]. In [10], the authors discussed the general nonsymmetric problem and proposed an algorithm for solving nonsymmetric pentadiagonal Toeplitz linear systems. A fast algorithm for solving a large system with a symmetric Toeplitz pentadiagonal coefficient matrix has been presented [11]. This efficient method depends on the idea of a system perturbation followed by corrections and is

competitive with standard methods. In [12], the authors described an efficient computational algorithm and symbolic algorithm for solving nearly pentadiagonal linear systems based on the LU factorization of the nearly pentadiagonal matrix.

In this paper, we introduce more efficient algorithms based on transformations which can be described as a natural generalization of the efficient algorithms in [15].

The current paper is organized as follows. In Section 2, new numerical algorithms for solving a pentadiagonal linear system are presented. New symbolic algorithms for solving a pentadiagonal linear system are constructed in Section 3. In Section 4, three illustrative examples are presented. Conclusions of the work are given in Section 5.

## 2. Numerical Algorithms for Solving PLS

In this section, we will focus on the construction of new numerical algorithms for computing the solution of pentadiagonal linear system. For this purpose it is convenient to give five vectors  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{n-1})$ ,  $\beta = (\beta_1, \beta_2, \dots, \beta_{n-2})$ ,  $\mathbf{Z} = (z_1, z_2, \dots, z_n)$ ,  $\gamma = (\gamma_2, \gamma_3, \dots, \gamma_n)$ , and  $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ , where

$$\alpha_i = \begin{cases} \frac{a_1}{\mu_1}, & i = 1, \\ \frac{a_i - \beta_{i-1}\gamma_i}{\mu_i}, & i = 2, 3, \dots, n-1, \end{cases} \quad (3)$$

$$\beta_i = \frac{b_i}{\mu_i}, \quad i = 2, 3, \dots, n-1, \quad (4)$$

$$z_i = \begin{cases} \frac{y_1}{\mu_1}, & i = 1, \\ \frac{y_2 - z_1\gamma_2}{\mu_2}, & i = 2, \\ \frac{y_i - z_{i-2}e_i - z_{i-1}\gamma_i}{\mu_i}, & i = 3, 4, \dots, n, \end{cases} \quad (5)$$

$$\gamma_i = \begin{cases} c_2, & i = 2, \\ c_i - \alpha_{i-2}e_i, & i = 3, 4, \dots, n, \end{cases} \quad (6)$$

$$\mu_i = \begin{cases} d_1, & i = 1, \\ d_2 - \alpha_1\gamma_2, & i = 2, \\ d_i - \beta_{i-2}e_i - \alpha_{i-1}\gamma_i, & i = 3, 4, \dots, n. \end{cases} \quad (7)$$

By using those vectors together with the suitable elementary row operations, one can see that system (1)

may be transformed to the equivalent linear system as follows:

$$\begin{pmatrix} 1 & \alpha_1 & \beta_1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \alpha_2 & \beta_2 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 1 & \alpha_3 & \beta_3 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & 1 & \alpha_4 & \beta_4 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 & 0 & 1 & \alpha_{n-2} & \beta_{n-2} \\ 0 & \dots & \dots & \dots & \dots & 0 & 0 & 0 & 1 & \alpha_{n-1} \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ z_{n-2} \\ z_{n-1} \\ z_n \end{pmatrix}. \quad (8)$$

The transformed system (8) is easy to solve by a backward substitution. Consequently, the PLS (1) can be solved using Algorithm 1.

The numerical Algorithm 1 will be referred to as PTRANS-I algorithm in the sequel. The computational cost of PTRANS-I algorithm is  $19n - 29$  operations. The conditions  $\mu_i \neq 0$ ,  $i = 1, 2, \dots, n$ , are sufficient for its validity.

In a similar manner, we may consider five vectors  $\sigma = (\sigma_2, \sigma_3, \dots, \sigma_n)$ ,  $\phi = (\phi_3, \phi_4, \dots, \phi_n)$ ,  $\mathbf{W} = (w_1, w_2, \dots, w_n)$ ,  $\rho = (\rho_1, \rho_2, \dots, \rho_{n-1})$ , and  $\psi = (\psi_1, \psi_2, \dots, \psi_n)$ , where

$$\sigma_i = \begin{cases} \frac{c_n}{\psi_n}, & i = n, \\ \frac{c_i - \phi_{i+1}\rho_i}{\psi_i}, & i = n-1, n-2, \dots, 2, \end{cases} \quad (9)$$

$$\phi_i = \frac{e_i}{\psi_i}, \quad i = n, n-1, \dots, 3, \quad (10)$$

To find the solution of PLS (1) using the transformed system (8), we proceed as follows:

**INPUT** Order of the matrix  $n$  and the components  $d_i, a_i, b_i, c_i, e_i, f_i, i = 1, 2, \dots, n, (a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0)$ .

**OUTPUT** The solution vector  $X = (x_1, x_2, \dots, x_n)^t$ .

*Step 1.* Use DETGPENTA algorithm [14] or DETGPENTA algorithm [16] to check the non-singularity of the coefficient matrix of the system (1).

*Step 2.* If  $\det(P) = 0$ , then Exit and Print Message ("No solutions") end if.

*Step 3.* Set  $\mu_1 = d_1, \alpha_1 = a_1/\mu_1, \beta_1 = b_1/\mu_1$ , and  $z_1 = y_1/\mu_1$ .

*Step 4.* Set  $\gamma_2 = c_2, \mu_2 = d_2 - \alpha_1\gamma_2, \alpha_2 = (a_2 - \beta_1\gamma_2)/\mu_2, \beta_2 = b_2/\mu_2$ , and  $z_2 = (y_2 - z_1\gamma_2)/\mu_2$ .

*Step 5.* For  $i = 3, 4, \dots, n-2$  do

    Compute and simplify:

$$\gamma_i = c_i - \alpha_{i-2}e_i,$$

$$\mu_i = d_i - \beta_{i-2}e_i - \alpha_{i-1}\gamma_i,$$

$$\alpha_i = \frac{a_i - \beta_{i-1}\gamma_i}{\mu_i},$$

$$\beta_i = \frac{b_i}{\mu_i},$$

$$z_i = \frac{y_i - z_{i-2}e_i - z_{i-1}\gamma_i}{\mu_i},$$

End do.

$$\gamma_{n-1} = c_{n-1} - \alpha_{n-3}e_{n-1},$$

$$\mu_{n-1} = d_{n-1} - \beta_{n-3}e_{n-1} - \alpha_{n-2}\gamma_{n-1},$$

$$\alpha_{n-1} = \frac{a_{n-1} - \beta_{n-2}\gamma_{n-1}}{\mu_{n-1}},$$

$$\gamma_n = c_n - \alpha_{n-2}e_n,$$

$$\mu_n = d_n - \beta_{n-2}e_n - \alpha_{n-1}\gamma_n,$$

$$z_{n-1} = \frac{y_{n-1} - z_{n-2}e_{n-1} - z_{n-2}\gamma_{n-1}}{\mu_{n-1}},$$

$$z_n = \frac{y_n - z_{n-1}e_n - \mu_{n-1}\gamma_n}{\mu_n},$$

*Step 6.* Compute the solution vector  $X = (x_1, x_2, \dots, x_n)^t$  using  $x_n = z_n, x_{n-1} = z_{n-1} - \alpha_{n-1}x_n$ .

For  $i = n-2, n-3, \dots, 1$  do

    Compute and simplify:

$$x_i = z_i - \alpha_i x_{i+1} - \beta_i x_{i+2}$$

End do.

ALGORITHM 1: First numerical algorithm for solving pentadiagonal linear system.

$$w_i = \begin{cases} \frac{y_n}{\psi_n}, & i = n, \\ \frac{y_{n-1} - w_n \rho_{n-1}}{\psi_{n-1}}, & i = n-1, \\ \frac{y_i - w_{i+2}b_i - w_{i+1}\rho_i}{\psi_i}, & i = n-2, n-3, \dots, 1, \end{cases} \quad (11)$$

$$\rho_i = \begin{cases} a_{n-1}, & i = n-1, \\ a_i - \sigma_{i+2}b_i, & i = n-2, n-3, \dots, 1, \end{cases} \quad (12)$$

$$\psi_i = \begin{cases} d_n, & i = n, \\ d_{n-1} - \sigma_n \rho_{n-1}, & i = n-1, \\ d_i - \phi_{i+2}b_i - \sigma_{i+1}\rho_i, & i = n-2, n-3, \dots, 1. \end{cases} \quad (13)$$

together with the suitable elementary row operations, we see that system (1) may be transformed to the equivalent linear system as follows:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \sigma_2 & 1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & 0 \\ \phi_3 & \sigma_3 & 1 & 0 & 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & \phi_4 & \sigma_4 & 1 & 0 & 0 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & \phi_{n-2} & \sigma_{n-2} & 1 & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & \phi_{n-1} & \sigma_{n-1} & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & \phi_n & \sigma_n & 1 \end{pmatrix}$$

Now we will present another algorithm for solving PLS. As in PTRANS-I algorithm, by using the vectors  $\sigma, \phi, W, \rho$ , and  $\psi$ ,

To find the solution of PLS (1) using the transformed system (14), we proceed as follows:

**INPUT** Order of the matrix  $n$  and the components  $d_i, a_i, b_i, c_i, e_i, f_i, i = 1, 2, \dots, n, (a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0)$ .

**OUTPUT** The solution vector  $X = (x_1, x_2, \dots, x_n)^t$ .

*Step 1.* Use DETGPENTA algorithm [14] or DETGPENTA algorithm [16] to check the non-singularity of the coefficient matrix of the system (1).

*Step 2.* If  $\det(P) = 0$ , then Exit and Print Message ("No solutions") end if.

*Step 3.* Set  $\psi_n = d_n, \sigma_n = c_n/\psi_n, \phi_n = e_n/\psi_n$ , and  $w_n = y_n/\psi_n$ .

*Step 4.* Set  $\rho_{n-1} = a_{n-1}, \psi_{n-1} = d_{n-1} - \sigma_n \rho_{n-1}, \sigma_{n-1} = (c_{n-1} - \phi_n \rho_{n-1})/\psi_{n-1}, \phi_{n-1} = e_{n-1}/\psi_{n-1}$ , and  $w_{n-1} = (y_{n-1} - w_n \rho_{n-1})/\psi_{n-1}$ .

*Step 5.* For  $i = n-2, n-3, \dots, 3$  do

    Compute and simplify:

$$\rho_i = a_i - \sigma_{i+2} b_i,$$

$$\psi_i = d_i - \phi_{i+2} b_i - \sigma_{i+1} \rho_i,$$

$$\sigma_i = \frac{c_i - \phi_{i+1} \rho_i}{\psi_i},$$

$$\phi_i = \frac{e_i}{\psi_i},$$

$$w_i = \frac{y_i - w_{i+2} b_i - w_{i+1} \rho_i}{\psi_i},$$

End do.

$$\rho_2 = a_2 - \sigma_4 b_2,$$

$$\psi_2 = d_2 - \phi_4 b_2 - \sigma_3 \rho_2,$$

$$\sigma_2 = \frac{c_2 - \phi_3 \rho_2}{\psi_2},$$

$$\rho_1 = a_1 - \sigma_3 b_1,$$

$$\psi_1 = d_1 - \phi_3 b_1 - \sigma_2 \rho_1,$$

$$w_2 = \frac{y_2 - w_4 b_2 - w_3 \rho_2}{\psi_2},$$

$$w_1 = \frac{y_1 - w_3 b_1 - w_2 \rho_1}{\psi_1},$$

*Step 6.* Compute the solution vector  $X = (x_1, x_2, \dots, x_n)^t$  using  $x = w_1, x_2 = w_2 - \sigma_2 x_1$ .

For  $i = 3, 4, \dots, n$  do

    Compute and simplify:

$$x_i = w_i - \sigma_i x_{i-1} - \phi_i x_{i-2}$$

End do.

**ALGORITHM 2:** Second numerical algorithm for solving pentadiagonal linear system.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \vdots \\ \vdots \\ \vdots \\ w_{n-2} \\ w_{n-1} \\ w_n \end{pmatrix}.$$

(14)

The transformed system (14) is easy to solve by a forward substitution. Consequently, the PLS (1) can be solved using Algorithm 2.

The numerical Algorithm 2 will be referred to as PTRANS-II algorithm in the sequel. The computational cost

of PTRANS-II algorithm is  $19n - 29$  operations. Also, the conditions  $\psi_i \neq 0, i = 1, 2, \dots, n$ , are sufficient for its validity.

If  $\mu_i = 0$  or  $\psi_i = 0$  for any  $i \in \{1, 2, \dots, n\}$  then PTRANS-I and PTRANS-II algorithms fail to solve pentadiagonal linear systems, respectively. So, in the next section, we develop two symbolic algorithms in order to remove the cases where the numerical algorithms fail. The parameter " $p$ " in the following symbolic algorithms is just a symbolic character. It is a dummy argument and its actual value is zero.

### 3. Symbolic Algorithms for Solving PLS

In this section, we will focus on the construction of new symbolic algorithms for computing the solution of pentadiagonal linear systems. Algorithm 3 is a symbolic version of PTRANS-I algorithm.

The symbolic Algorithm 3 will be referred to as SPTRANS-I algorithm in the sequel.

Algorithm 4 gives the symbolic version of PTRANS-II algorithm.

The symbolic Algorithm 4 will be referred to as SPTRANS-II algorithm in the sequel.

To find the solution of PLS (1) using the transformed system (8), we proceed as follows:

**INPUT** Order of the matrix  $n$  and the components  $d_i, a_i, b_i, c_i, e_i, f_i, i = 1, 2, \dots, n, (a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0)$ .

**OUTPUT** The solution vector  $X = (x_1, x_2, \dots, x_n)^t$ .

**Step 1.** Use DETGPENTA algorithm [14] or DETGPENTA algorithm [16] to check the non-singularity of the coefficient matrix of the system (1).

**Step 2.** If  $\det(P) = 0$ , then Exit and Print Message ("No solutions") end if.

**Step 3.** Set  $\mu_1 = d_1$ . If  $\mu_1 = 0$  then  $\mu_1 = p$  end if.

**Step 4.** Set  $\alpha_1 = a_1/\mu_1, \beta_1 = b_1/\mu_1, z_1 = \gamma_1/\mu_1$  and  $\gamma_2 = c_2$ .

**Step 5.** Set  $\mu_2 = d_2 - \alpha_1\gamma_2$ . If  $\mu_2 = 0$  then  $\mu_2 = p$  end if.

**Step 6.** Set  $\alpha_2 = (a_2 - \beta_1\gamma_2)/\mu_2, \beta_2 = b_2/\mu_2$ , and  $z_2 = (\gamma_2 - z_1\gamma_2)/\mu_2$ .

**Step 7.** For  $i = 3, 4, \dots, n-2$  do

    Compute and simplify:

$$\gamma_i = c_i - \alpha_{i-2}e_i,$$

$$\mu_i = d_i - \beta_{i-2}e_i - \alpha_{i-1}\gamma_i,$$

    If  $\mu_i = 0$  then  $\mu_i = p$  end if.

$$\alpha_i = \frac{a_i - \beta_{i-1}\gamma_i}{\mu_i},$$

$$\beta_i = \frac{b_i}{\mu_i},$$

$$z_i = \frac{\gamma_i - z_{i-2}e_i - z_{i-1}\gamma_i}{\mu_i},$$

End do.

$$\gamma_{n-1} = c_{n-1} - \alpha_{n-3}e_{n-1},$$

$\mu_{n-1} = d_{n-1} - \beta_{n-3}e_{n-1} - \alpha_{n-2}\gamma_{n-1}$ . If  $\mu_{n-1} = 0$  then  $\mu_{n-1} = p$  end if.

$$\alpha_{n-1} = \frac{a_{n-1} - \beta_{n-2}\gamma_{n-1}}{\mu_{n-1}},$$

$$\gamma_n = c_n - \alpha_{n-2}e_n,$$

$\mu_n = d_n - \beta_{n-2}e_n - \alpha_{n-1}\gamma_n$ . If  $\mu_n = 0$  then  $\mu_n = p$  end if.

$$z_{n-1} = \frac{\gamma_{n-1} - z_{n-2}e_{n-1} - z_{n-2}\gamma_{n-1}}{\mu_{n-1}},$$

$$z_n = \frac{\gamma_n - z_{n-1}e_n - z_{n-1}\gamma_n}{\mu_n},$$

**Step 8.** Compute the solution vector  $X = (x_1, x_2, \dots, x_n)^t$  using  $x_n = z_n, x_{n-1} = z_{n-1} - \alpha_{n-1}x_n$ .

For  $i = n-2, n-3, \dots, 1$  do

    Compute and simplify:

$$x_i = z_i - \alpha_i x_{i+1} - \beta_i x_{i+2}$$

End do.

**Step 9.** Substitute  $p = 0$  in all expressions of the solution vector  $x_i, i = 1, 2, \dots, n$ .

ALGORITHM 3: First symbolic algorithm for solving pentadiagonal linear system.

**Corollary 1** (generalization version of Corollary 2.1 in [15]).

Let  $\hat{P}$  be the backward matrix of the pentadiagonal matrix  $P$  in (2) and be given by

Then the backward pentadiagonal linear system

$$\hat{P}V = Y, \quad V = (v_1, v_2, \dots, v_n)^t \quad (16)$$

has the solution  $v_i = x_{n-i+1}, i = 1, 2, \dots, [n]$ , where  $[j]$  is the floor function of  $j$  and  $X = (x_1, x_2, \dots, x_n)^t$  is the solution vector of the linear system (1).

*Proof.* Consider the  $n \times n$  permutation matrix  $M$  defined by

$$M = \begin{pmatrix} 0 & \dots & \dots & 0 & 1 \\ \vdots & & & 1 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 1 & & \vdots & \\ 1 & 0 & \dots & \dots & 0 \end{pmatrix}. \quad (17)$$

$$\hat{P} = \begin{pmatrix} 0 & \dots & \dots & \dots & \dots & \dots & 0 & b_1 & a_1 & d_1 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & b_2 & a_2 & d_2 & c_2 \\ 0 & \dots & \dots & \dots & 0 & b_3 & a_3 & d_3 & c_3 & e_3 \\ 0 & \dots & \dots & 0 & b_4 & a_4 & d_4 & c_4 & e_4 & 0 \\ \vdots & \dots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ b_{n-2} & a_{n-2} & d_{n-2} & c_{n-2} & e_{n-2} & 0 & \dots & \dots & \dots & 0 \\ a_{n-1} & d_{n-1} & c_{n-1} & e_{n-1} & 0 & \dots & \dots & \dots & \dots & 0 \\ d_n & c_n & e_n & 0 & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix},$$

$n \geq 4.$   
(15)

For this matrix, we have

$$M^{-1} = M. \quad (18)$$

Since

$$\hat{P} = PM \quad (19)$$

then, using (18) and (19), the result follows.  $\square$

**Corollary 2** (generalization version of Corollary 2.2 in [15]).  
The determinants of the coefficient matrices  $P$  and  $\hat{P}$  in (2) and (15) are given, respectively, by

$$\det(P) = \prod_{i=1}^n \mu_i = \prod_{i=1}^n \psi_i, \quad (20)$$

$$\det(\hat{P}) = (-1)^{n(n-1)/2} \prod_{i=1}^n \mu_i = (-1)^{n(n-1)/2} \prod_{i=1}^n \psi_i,$$

where  $\mu_1, \mu_2, \dots, \mu_n$  and  $\psi_1, \psi_2, \dots, \psi_n$  satisfy (7) and (13), respectively.

*Proof.* Using (8), (14) and (19) complete the proof.  $\square$

#### 4. Illustrative Examples

In this section, we give three examples for the sake of illustration. All experiments were performed in MATLAB R2014a with an Intel Core i7-4700MQ CPU@2.40 GHz 2.40 GHz.

**Example 1** (Case I:  $\mu_i \neq 0$  and  $\psi_i \neq 0$  for all  $i$ ). Find the solution of the following pentadiagonal linear system of size 10:

$$\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 1 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & -4 & 5 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 5 & -7 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 1 & 6 & 3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 7 & -1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 1 & -1 & 4 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{pmatrix} = \begin{pmatrix} 8 \\ 33 \\ 8 \\ 24 \\ 29 \\ 82 \\ 71 \\ 17 \\ 57 \\ 108 \end{pmatrix}.$$

*Solution.* We have

$$n = 10, d = (1, 2, 3, -4, 5, 6, 7, -1, 1, 8)^t, a = (2, 2, 1, 5, -7, 3, -1, 4, 5)^t, b = (1, 5, -2, 1, 5, 2, 4, -3)^t, c = (0, 3, 2, 1, 2, 1, 2, 1, -2, 4)^t, e = (0, 0, 1, 3, 1, 5, 2, 2, 2, -1)^t, \text{ and } y = (8, 33, 8, 24, 29, 98, 99, 17, 57, 108)^t.$$

(i) Applying the PTRANS-I algorithm, it yields

$$\begin{aligned} \text{(a) } \mu &= (1, -4, 2, -3/8, 27, 245/9, 3289/441, -335/383, -2897/484, 3439/279)^t, \text{ and } \det(P) = \prod_{i=1}^{10} \mu_i = 4989610795975/4701708; \\ \text{(b) PTRANS-I}(n, d, a, b, c, e, y) &= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)^t. \end{aligned}$$

(ii) Applying the PTRANS-II algorithm, it yields

$$\begin{aligned} \text{(a) } \psi &= (-6213/3613, 1603/1405, 1487/433, -5173/239, 383/156, 988/161, 69/11, -77/12, -3/2, 8)^t, \text{ and } \det(P) = \prod_{i=1}^{10} \psi_i = 557494642026514353/525327436055; \\ \text{(b) PTRANS-II}(n, d, a, b, c, e, y) &= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)^t. \end{aligned}$$

**Example 2** (Case II:  $\mu_i = 0$  and  $\psi_i = 0$  for some  $i$ ). Find the solution of the following pentadiagonal linear system of size 4:

$$\begin{pmatrix} 3 & 2 & 1 & 0 \\ -3 & -2 & 7 & 1 \\ 3 & 2 & -1 & 5 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ 9 \\ 6 \end{pmatrix}. \quad (22)$$

*Solution.* We have

$$n = 4, d = (3, -2, -1, 3)^t, a = (2, 7, 5)^t, b = (1, 1)^t, c = (0, -3, 2, 2)^t, e = (0, 0, 3, 1)^t, \text{ and } y = (6, 3, 9, 6)^t.$$

The numerical algorithms PTRANS-I and PTRANS-II fail to solve the pentadiagonal linear system (22) since  $\mu_2 = 0$ .

To find the solution of PLS (1) using the transformed system (14), we proceed as follows:

**INPUT** Order of the matrix  $n$  and the components  $d_i, a_i, b_i, c_i, e_i, f_i, i = 1, 2, \dots, n$ , ( $a_n = b_n = b_{n-1} = c_1 = e_1 = e_2 = 0$ ).

**OUTPUT** The solution vector  $X = (x_1, x_2, \dots, x_n)^t$ .

*Step 1.* Use DETGPENTA algorithm [14] or DETGPENTA algorithm [16] to check the non-singularity of the coefficient matrix of the system (1).

*Step 2.* If  $\det(P) = 0$ , then Exit and Print Message ("No solutions") end if.

*Step 3.* Set  $\psi_n = d_n$ . If  $\psi_n = 0$  then  $\psi_n = p$  end if.

*Step 4.*  $\sigma_n = c_n/\psi_n$ ,  $\phi_n = e_n/\psi_n$ ,  $w_n = y_n/\psi_n$  and  $\rho_{n-1} = a_{n-1}$ .

*Step 5.* Set  $\psi_{n-1} = d_{n-1} - \sigma_n \rho_{n-1}$ . If  $\psi_{n-1} = 0$  then  $\psi_{n-1} = p$  end if.

*Step 6.*  $\sigma_{n-1} = (c_{n-1} - \phi_n \rho_{n-1})/\psi_{n-1}$ ,  $\phi_{n-1} = e_{n-1}/\psi_{n-1}$ , and  $w_{n-1} = (y_{n-1} - w_n \rho_{n-1})/\psi_{n-1}$ .

*Step 7.* For  $i = n-2, n-3, \dots, 3$  do

    Compute and simplify:

$$\rho_i = a_i - \sigma_{i+2} b_i,$$

$$\psi_i = d_i - \phi_{i+2} b_i - \sigma_{i+1} \rho_i,$$

    If  $\psi_i = 0$  then  $\psi_i = p$  end if.

$$\sigma_i = \frac{c_i - \phi_{i+1} \rho_i}{\psi_i},$$

$$\phi_i = \frac{e_i}{\psi_i},$$

$$w_i = \frac{y_i - w_{i+2} b_i - w_{i+1} \rho_i}{\psi_i},$$

End do.

$$\rho_2 = a_2 - \sigma_4 b_2,$$

$\psi_2 = d_2 - \phi_4 b_2 - \sigma_3 \rho_2$ . If  $\psi_2 = 0$  then  $\psi_2 = p$  end if.

$$\sigma_2 = \frac{c_2 - \phi_4 \rho_2}{\psi_2},$$

$$\rho_1 = a_1 - \sigma_3 b_1,$$

$\psi_1 = d_1 - \phi_3 b_1 - \sigma_2 \rho_1$ . If  $\psi_1 = 0$  then  $\psi_1 = p$  end if.

$$w_2 = \frac{y_2 - w_4 b_2 - w_3 \rho_2}{\psi_2},$$

$$w_1 = \frac{y_1 - w_3 b_1 - w_2 \rho_1}{\psi_1},$$

*Step 8.* Compute the solution vector  $X = (x_1, x_2, \dots, x_n)^t$  using  $x = w_1$ ,  $x_2 = w_2 - \sigma_2 x_1$ .

For  $i = 3, 4, \dots, n$  do

    Compute and simplify:

$$x_i = w_i - \sigma_i x_{i-1} - \phi_i x_{i-2}$$

End do.

*Step 9.* Substitute  $p = 0$  in all expressions of the solution vector  $x_i, i = 1, 2, \dots, n$ .

ALGORITHM 4: Second symbolic algorithm for solving pentadiagonal linear system.

(i) Applying the SPTRANS-I algorithm, it yields

$$(a) \mu = (3, p, -2, (8p - 21)/p)^t, \det(P) = (\prod_{i=1}^4 \mu_i)_{p=0} = 126;$$

$$(b) \text{SPTRANS-I}(n, d, a, b, c, e, y) = (((25p - 42)/(16p - 42), -21/(8p - 21), 21(p - 2)/(8p - 21), \text{and } (9p - 21)/(8p - 21))^t)_{p=0} = (1, 1, 1, 1)^t.$$

(ii) Applying the SPTRANS-II algorithm, it yields

$$(a) \psi = (21/4, -24/13, -13/3, 3)^t, \det(P) = \prod_{i=1}^4 \psi_i = 126;$$

$$(b) \text{SPTRANS-II}(n, d, a, b, c, e, y) = (1, 1, 1, 1)^t.$$

**Example 3.** We consider the following  $n \times n$  pentadiagonal linear system in order to demonstrate the efficiency of Algorithms 3 and 4:

$$\begin{pmatrix} 9 & -4 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ -4 & 6 & -4 & 1 & 0 & & & & & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & & & & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & \ddots & & & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & 0 & 1 & -4 & 6 & -4 & 1 \\ \vdots & & & & & & 0 & 1 & -4 & 5 & -2 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 & -2 & 1 \end{pmatrix}$$



TABLE 1: Comparisons between our proposed algorithms and state-of-the-art algorithms in literature.

$n$	$\ \mathbf{x} - \bar{\mathbf{x}}\ _\infty$ and CPU time (S)									
	PTRANS-I		PTRANS-II		Algorithm 3 [9]		A \ b (MATLAB)		SYMPENTAINV [7]	
500	$1.5856 \times 10^{-7}$	0.0069	0	0.0086	$6.8579 \times 10^{-8}$	0.0048	$9.98 \times 10^{-8}$	0.0023	$1.5881 \times 10^{-7}$	0.3317
5000	$8.3674 \times 10^{-4}$	0.0062	0	0.0391	$3.0253 \times 10^{-4}$	0.0057	$2.50 \times 10^{-4}$	0.7548	$8.3654 \times 10^{-4}$	108.0919
10000	0.0058	0.0114	0	0.0511	0.0052	0.0101	0.0106	4.5464	0.0058	812.9783
50000	2.1415	0.0308	0	0.1687	7.9056	0.0119	0.0159	655.51	—	—

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 6 \\ -1 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (23)$$

Analytically, one can see that the exact solution of the above system is  $\mathbf{x} = (1, 1, \dots, 1)^T$ . In Table 1, we give some comparisons between our proposed algorithm and the state-of-the-art algorithms in literature. Table 1 shows solutions obtained by our proposed algorithms and other algorithms in literature for different sizes. Our obtained results show that our algorithm PTRANS-II gives better absolute error than those algorithms used in comparisons for large values of  $n$ . Moreover the obtained results indicate that the value of the running time for Algorithm 3 [9] is small in comparison with other algorithms for large sizes.

## 5. Conclusion

There are many numerical algorithms that have been used for solving linear systems of pentadiagonal type. All numerical algorithms including the PTRANS-I and PTRANS-II algorithms of the current paper fail to solve the pentadiagonal linear system if  $\mu_i = 0$  and  $\psi_i = 0$  for any  $i \in \{1, 2, \dots, n\}$ . The symbolic algorithms SPTRANS-I and SPTRANS-II of the current paper are constructed in order to remove the cases where the numerical algorithms fail. Using numerical examples we have obtained that SPTRANS-II algorithm works as well as Algorithm 3 [9] and (A\b) MATLAB algorithms. Hence, it may become a useful tool for solving linear systems of pentadiagonal type.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors would like to express their extremely gratefulness to the referees and editor for their useful comments and suggestions that improved this paper. This project was supported by King Saud University, Deanship of Scientific Research, College of Science Research Center.

## References

- [1] T. A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, Pa, USA, 2006.
- [2] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, Pa, USA, 1997.
- [3] G. H. Golub and C. F. van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [4] J. P. Killingbeck and G. Jolicard, "The folding algorithm for pentadiagonal matrices," *Physics Letters A*, vol. 166, no. 3-4, pp. 159-162, 1992.
- [5] J. J. Dongarra and A. H. Sameh, "On some parallel banded system solvers," *Parallel Computing*, vol. 1, no. 3-4, pp. 223-235, 1984.
- [6] U. Meier, "A parallel partition method for solving banded systems of linear equations," *Parallel Computing*, vol. 2, no. 1, pp. 33-43, 1985.
- [7] X.-L. Zhao and T.-Z. Huang, "On the inverse of a general pentadiagonal matrix," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 639-646, 2008.
- [8] A. A. Karawia, "A computational algorithm for solving periodic penta-diagonal linear systems," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 613-618, 2006.
- [9] T. Sogabe, "New algorithms for solving periodic tridiagonal and periodic pentadiagonal linear systems," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 850-856, 2008.
- [10] L. Du, T. Sogabe, and S.-L. Zhang, "An algorithm for solving nonsymmetric penta-diagonal toeplitz linear systems," *Applied Mathematics and Computation*, vol. 244, pp. 10-15, 2014.
- [11] S. S. Nemani, "A fast algorithm for solving Toeplitz penta-diagonal systems," *Applied Mathematics and Computation*, vol. 215, no. 11, pp. 3830-3838, 2010.
- [12] A. A. Karawia and S. A. El-Shehawy, "Nearly pentadiagonal linear systems," *Journal of Natural Sciences and Mathematics, Qassim University*, vol. 5, pp. 89-102, 2012.
- [13] J. M. McNally, "A fast algorithm for solving diagonally dominant symmetric pentadiagonal Toeplitz systems," *Journal of Computational and Applied Mathematics*, vol. 234, no. 4, pp. 995-1005, 2010.
- [14] T. Sogabe, "A fast numerical algorithm for the determinant of a pentadiagonal matrix," *Applied Mathematics and Computation*, vol. 196, no. 2, pp. 835-841, 2008.



- [15] M. El-Mikkawy and F. Atlan, "Algorithms for solving linear systems of equations of tridiagonal type via transformations," *Applied Mathematics*, vol. 5, no. 3, pp. 413–422, 2014.
- [16] M. E. A. El-Mikkawy, "A fast and reliable algorithm for evaluating  $n$ th order pentadiagonal determinants," *Applied Mathematics and Computation*, vol. 202, no. 1, pp. 210–215, 2008.

