J Math Chem (2014) 52:2222–2233
DOI 10.1007/s10910-014-0378-1

ORIGINAL PAPER

A symbolic algorithm for periodic tridiagonal systems of equations

Ji-Teng Jia · Qiong-Xiang Kong

Received: 20 February 2014 / Accepted: 19 June 2014 / Published online: 4 July 2014

Abstract In the current paper, we present a novel symbolic algorithm for solving periodic tridiagonal linear systems without imposing any restrictive conditions. The computational cost of the algorithm is less than or almost equal to those of three well-known algorithms given by Chawla and Khazal (Int. J. Comput. Math. 79(4):473–484, 2002) and by El-Mikkawy (Appl. Math. Comput. 161:691–696, 2005), respectively. In addition, the solution of periodic anti-tridiagonal linear systems is also discussed. Two numerical experiments are provided in order to illustrate the performance and effectiveness of our algorithm. All of the experiments were performed on a computer with aid of programs written in MATLAB.

Keywords Periodic tridiagonal matrices · Periodic anti-tridiagonal matrices · Matrix decomposition · Linear systems · Computational cost · Computer Algebra Systems (CASs)

Mathematics Subject Classification 15A09 · 15F15 · 33F30 · 65F30

1 Introduction and main objectives

Tridiagonal matrices and periodic (or cyclic) tridiagonal matrices frequently appear in mathematical chemistry and computational physics as well as scientific and engi-

J.-T. Jia (✉)
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China
e-mail: lavenderjjt@163.com

Q.-X. Kong
Department of Building Environment and Services Engineering, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China
e-mail: qxkong@mail.xjtu.edu.cn

neering investigations. Several examples of this can be found in quantum chemistry, Hückel theory, boundary value problems (BVPs), fluid mechanics, spline approximation, parallel computing, and vision, image and signal processing (VISP), etc. [1–7].

In this paper we mainly consider symbolic algorithm for the solution of the linear system

$$A\mathbf{x} = \mathbf{f}, \tag{1.1}$$

where A is an n-by-n periodic tridiagonal matrix of the form

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 & a_1 \\ a_2 & b_2 & c_2 & 0 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & 0 & \cdots & 0 & a_n & b_n \end{pmatrix}, \tag{1.2}$$

while $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the vector of unknowns and $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$ is the right-hand side vector. Throughout the paper, the superscript symbol T corresponds to the transpose operation of vector or matrix.

In the case where $a_1 = c_n = 0$, the coefficient matrix A is strictly tridiagonal, and algorithms for the solution of such systems are widely known [8–11]. PT linear systems with $a_1 \neq 0$, $c_n \neq 0$ arise not only in a variety of theoretical areas (linear algebra and numerical analysis), but also in different areas of applications [12–15].

It often happens that the coefficient matrix has some nice properties, such that a general-purpose solution algorithm results too expensive in comparison with more "clever" ones, which use the favorable characteristics of the matrix itself. In recent years, some researchers have developed fast numerical (or symbolic) algorithms for solving PT linear systems. For example, Chawla and Khazal [16], El-Mikkawy [17]. According to these results, our main objective is to devise a more efficient algorithm for solving system (1.1) without imposing any restrictive conditions.

An outline of this paper is as follows: in the next section, we review three well-known algorithms and construct a novel symbolic algorithm to solve a PT linear system in linear time. In addition, we propose a method for solving periodic anti-tridiagonal (PAT) linear systems. In Sect. 3, we give the results of some numerical examples to show the performance of the proposed algorithm and its competitiveness with other existing algorithms. Finally, we make some conclusions of the work in Sect. 4.

2 Main results

In this section, we are going to establish a symbolic algorithm for solving a PT linear system of the form (1.1). To do this, we begin by reviewing three existing algorithms: Classical elimination algorithm, Sherman-Morrison algorithm [16], and PERTRI algorithm [17].

2.1 Numerical algorithms for solving PT linear systems

We first describe adaptation of classical elimination process for the solution of system (1.1) below.

Algorithm 2.1 Classical elimination algorithm

Step 1. Elimination stage:
For $k = 1, 2, \ldots, n-3$,
$m_k = \frac{a_{k+1}}{b_k}$, $b_{k+1} = b_{k+1} - m_k c_k$, $d_{k+1} = d_{k+1} - m_k d_k$, $\mu_k = \frac{s_k}{b_k}$,
$t_{k+1} = -m_k t_k$, $s_{k+1} = -\mu_k c_k$, $b_n = b_n - \mu_k t_k$, $d_n = d_n - \mu_k d_k$,
End
For $k = n-2$,
$m_k = \frac{a_{k+1}}{b_k}$, $b_{k+1} = b_{k+1} - m_k c_k$, $d_{k+1} = d_{k+1} - m_k d_k$, $\mu_k = \frac{s_k}{b_k}$,
$c_{n-1} = c_{n-1} - m_k t_k$, $a_n = a_n - \mu_k c_k$, $b_n = b_n - \mu_k t_k$, $d_n = d_n - \mu_k d_k$,
End
For $k = n-1$,
$m_k = \frac{a_{k+1}}{b_k}$, $b_{k+1} = b_{k+1} - m_k c_k$, $d_{k+1} = d_{k+1} - m_k d_k$,
End

Step 2. Solution stage:
$x_n = \frac{d_n}{b_n}$, $x_{n-1} = \frac{d_{n-1} - c_{n-1} x_n}{b_{n-1}}$,
For $k = n-2, n-3, \ldots, 1$,
$x_k = \frac{d_k - c_k x_{k+1} - t_k x_n}{b_k}$,
End

Here, it is easy to see that the elimination stage involves $12n - 17$ operations while the solution stage involves $5n - 6$ operations. Thus, classical elimination for the system (1.1) involves a total of $17n - 23$ arithmetic operations.

In their paper [16], Chawla and Khazal proposed a computational algorithm based on the Sherman–Morrison–Woodbury formula [18]. The resulting so-called Sherman–Morrison algorithm for PT linear systems is described below.

Suppose two n-by-n matrices X and Y are related by

$$X = Y - \mathbf{u}\mathbf{v}^T,$$

where $\mathbf{u}$ and $\mathbf{v}$ are vectors of length n. If Y is nonsingular and $\mathbf{v}^T Y^{-1}\mathbf{u} \neq 1$, then the inverse of the matrix X can be explicitly given by

$$X^{-1} = Y^{-1} + \delta \cdot Y^{-1}\mathbf{u}\mathbf{v}^T Y^{-1},$$

where

$$\delta = \frac{1}{1 - \mathbf{v}^T Y^{-1}\mathbf{u}}.$$

For the case of PT linear systems (1.1), $X = A$ and we select $\mathbf{u} = (1, 0, \ldots, 0, -1)^T$ and $\mathbf{v} = (c_n, 0, \ldots, 0, -a_1)^T$. Then, the algorithm is as follows.

Algorithm 2.2 Sherman-Morrison algorithm

Step 1. Set $\mathbf{u} = (1, 0, \ldots, 0, -1)^T$ and $\mathbf{v} = (c_n, 0, \ldots, 0, -a_1)^T$.
Step 2. Form the tridiagonal matrix

$$Y = \begin{pmatrix} b_1 + c_n & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & \cdots & 0 & a_n & b_n + a_1 \end{pmatrix}.$$

Step 3. Solve for $\mathbf{z}$ and $\mathbf{w}$ the two tridiagonal linear systems:

$$Y\mathbf{z} = \mathbf{f},\ Y\mathbf{w} = \mathbf{u}.$$

Step 4. Compute

$$\alpha_1 = \mathbf{v}^T\mathbf{z},\ \ \alpha_2 = \mathbf{v}^T\mathbf{w},\ \ \beta = \frac{\alpha_1}{1 - \alpha_2}.$$

Step 5. Compute the solution

$$\mathbf{x} = \mathbf{z} + \beta \cdot \mathbf{w}.$$

From the above algorithm, we can see that Chawla and Khazal's algorithm requires $14n + 2$ operations for solving the PT system (1.1), and this leads to the result that Chawla and Khazal's algorithm is about 18 % faster than the classical elimination algorithm when the system size n is large enough. However, for the validity of the algorithm, the conditions are that Y is nonsingular and $\mathbf{v}^T Y^{-1}\mathbf{u} \neq 1$.

More recently, El-Mikkawy presented a recursive algorithm (PERTRI algorithm) for PT linear systems. The algorithm is based on the LU decomposition that represents the coefficient matrix A as a product of lower and upper triangular matrices, i.e., $A = LU$. Consequently, solving $A\mathbf{x} = \mathbf{f}$ can be achieved by solutions of two triangular linear systems $L\mathbf{y} = \mathbf{f}$ and $U\mathbf{x} = \mathbf{y}$. The corresponding algorithm is given below.

Algorithm 2.3 PERTRI algorithm

Step 1. Set $d_1 = b_1, v_1 = a_1, h_1 = \frac{c_n}{d_1}$.
Step 2. For $i = 2, 3, \dots, n-2$,

$d_i = b_i - \frac{a_i}{d_{i-1}} c_{i-1}, v_i = -\frac{a_i}{d_{i-1}} v_{i-1}, h_i = -\frac{c_{i-1}}{d_i} h_{i-1},$

End

$d_{n-1} = b_{n-1} - \frac{a_{n-1}}{d_{n-2}}, v_{n-1} = c_{n-1} - \frac{a_{n-1}}{d_{n-2}} v_{n-2},$

$h_{n-1} = \frac{a_n - h_{n-2} c_{n-2}}{d_{n-1}}, d_n = b_n - \sum_{i=1}^{n-1} h_i v_i.$

Step 3. Set $r_1 = f_1$,

For $i = 2, 3, \dots, n$,

$r_i = f_i - \frac{a_i}{d_{i-1}} r_{i-1},$

End

$r_n = f_n - \sum_{i=1}^{n-1} h_i r_i.$

Step 4. Compute the solution vector $\mathbf{x}$ using

$x_n = \frac{r_n}{d_n}, x_{n-1} = \frac{r_{n-1} - v_{n-1} x_n}{d_{n-1}},$

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

$x_i = \frac{r_i - c_i x_{i+1} - v_i x_n}{d_i},$

End

From the above algorithm, we can see that El-Mikkawy's algorithm requires $19n - 27$ operations for solving the PT linear system (1.1). In fact, the PERTRI algorithm is a generalization of the well-known algorithm due to Thomas [19,20] for solving PT linear systems. For validity of the algorithm, the conditions are that $d_i \neq 0$ for all $i = 1, 2, \dots, n$.

2.2 A more efficient symbolic algorithm for PT linear systems

In this subsection, we will develop a symbolic algorithm for solving PT linear systems based on the use of a special matrix decomposition. Let us consider the lower triangular matrix L, the nearly upper unitriangular matrix M and the upper unitriangular matrix U defined by

$$L = \begin{pmatrix} d_1 & 0 & \cdots & \cdots & \cdots & 0 \\ a_2 & d_2 & \ddots & \ddots & \ddots & \vdots \\ 0 & a_3 & d_3 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1} & d_{n-1} & 0 \\ 0 & \cdots & \cdots & 0 & a_n & d_n \end{pmatrix} \in \mathbb{R}^{n \times n}, \tag{2.1}$$

$$M = \begin{pmatrix} 1 & e_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & e_{n-3} \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & e_{n-2} & \vdots \\ 0 & \ddots & \ddots & \ddots & 1 & 0 \\ g_n & 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n\times n}, \tag{2.2}$$

and

$$U = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & g_1 \\ 0 & 1 & \ddots & \ddots & \ddots & g_2 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & g_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n\times n}. \tag{2.3}$$

Proposition 2.1 *The periodic tridiagonal matrix A given in* (1.2) *can be decomposed into three matrices as*

$$A = L(\tau)\cdot M(\tau)\cdot U(\tau), \tag{2.4}$$

where $L(\tau)$, $M(\tau)$, $U(\tau)$ are matrices defined in (2.1)-(2.3), and the elements in these matrices satisfy

$$d_i = \begin{cases} b_1 & \textit{if } i = 1 \\ b_i - e_{i-1}a_i & \textit{if } i = 3, 4, \ldots, n-1 \\ b_n - c_n g_1 - a_n g_{n-1} & \textit{if } i = n, \end{cases} \tag{2.5}$$

(If $d_i = 0$, then set $d_i = \tau$, τ is just a symbolic name),

$$e_i = \frac{c_i}{d_i},\ i = 1, 2, \ldots, n-2, \tag{2.6}$$

$$g'_i = \begin{cases} \frac{a_1}{d_1} & \textit{if } i = 1 \\ -\frac{a_i g'_{i-1}}{d_i} & \textit{if } i = 2, 3, \ldots, n-2 \\ \frac{c_{n-1} - a_{n-1} g'_{n-2}}{d_{n-1}} & \textit{if } i = n-1, \end{cases} \tag{2.7}$$

$$g_i = \begin{cases} \frac{c_n}{d_n} & \textit{if } i = n \\ g'_{n-1} & \textit{if } i = n-1 \\ g'_i - e_i g_{i+1} & \textit{if } i = n-2, n-3, \ldots, 1. \end{cases} \tag{2.8}$$

Proof By the identity of (2.4) and simple algebraic manipulations, the conclusion holds. □

Remark 2.2 It should be mentioned that the LMU decomposition for the periodic tridiagonal matrix A always exists even if A is singular. In fact, such decomposition depends on at most one formal parameter τ which can be regarded as a symbolic name whose actual value is 0.

In addition, it follows from (2.4) that we have

$$\det(A) = \det(L(\tau)) \cdot \det(M(\tau)) \cdot \det(U(\tau)) = \left.\left(\prod_{i=1}^{n} d_i\right)\right|_{\tau=0}, \tag{2.9}$$

since $\det(M(\tau)) = \det(U(\tau)) = 1$. In general, it is important to compute the determinant of the coefficient matrix since the value tells whether the corresponding system is ill-conditioned or not and the system has unique solution or not [21,22].

Remark 2.3 As a direct consequence of the above results, we can conclude that if the periodic tridiagonal matrix A is symmetric, then it is positive definite if and only if $d_i > 0$ for all $i = 1, 2, \ldots, n$.

Based on the decomposition in Proposition 2.1, the PT linear system $A\mathbf{x} = \mathbf{f}$ can be directly transformed into three linear systems as follows:

$$L(\tau)\mathbf{z} = \mathbf{f},\ \ M(\tau)\mathbf{y} = \mathbf{z},\ \ U(\tau)\mathbf{x} = \mathbf{y},$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_n)^T$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$. Moreover, one may notice that these systems can be efficiently solved by forward substitution and back substitution. The resulting algorithm is summarized as follows:

Algorithm 2.4

Step 1. Input a_i, b_i, c_i, f_i and n.
Step 2. Compute the LMU decomposition of matrix A by Proposition 2.1.
Step 3. Set $z_1 = \frac{f_1}{d_1}$,
For $i = 2, 3, \ldots, n$ compute
$z_i = \frac{f_i - a_i z_{i-1}}{d_i}$,
End.
Step 4. Set $y_{n-1} = z_{n-1}$,
For $i = n-2, n-3, \ldots, 1$ compute
$y_i = z_i - e_i y_{i+1}$,
End.
Step 5. Set $x_n = z_n - g_n y_1$,
For $i = n-1, n-2, \ldots, 1$ compute
$x_i = y_i - g_i x_n$,
End.
Step 6. Output the solution of system: $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$.

This new symbolic algorithm will be referred to as the SPT algorithm. The computational cost (i.e., the number of basic arithmetic operations) for Algorithm 2.4 are

Table 1 Comparisons of the computational cost for different algorithms

Algorithms	Number of basic arithmetic operations
Classical elimination algorithm	$17n - 23$
Sherman-Morrison algorithm	$14n + 2$
PERTRI algorithm	$19n - 27$
Our algorithm	$14n - 13$

$14n - 13$, since costs for the steps 2, 3, 4, and 5 are $7n - 7$, $3n - 2$, $2n - 2$, and $2n - 2$, respectively.

In the following, we compare the computational cost among Classical elimination algorithm (Algorithm 2.1), Sherman-Morrison algorithm (Algorithm 2.2), PERTRI algorithm (Algorithm 2.3) and our algorithm (Algorithm 2.4) in Table 1.

Comparing the results in above table, we can see that the computational cost of our algorithm is less than or almost equal to those of other existing algorithms.

2.3 A method for solving PAT linear systems

In this subsection, we consider the solution of PAT linear systems of the form $\hat{A}\mathbf{x} = \mathbf{f}$, where $\hat{A}$ is an n-by-n PAT matrix given by

$$\hat{A} = \begin{pmatrix} a_1 & 0 & \cdots & 0 & c_1 & b_1 \\ 0 & \iddots & 0 & c_2 & b_2 & a_2 \\ \vdots & \iddots & \iddots & \iddots & \iddots & 0 \\ 0 & \iddots & \iddots & \iddots & \iddots & \vdots \\ c_{n-1} & b_{n-1} & a_{n-1} & 0 & \iddots & 0 \\ b_n & a_n & 0 & \cdots & 0 & c_n \end{pmatrix}. \tag{2.10}$$

To do this, we take into account the fact that for the n-by-n matrix Q of the form

$$Q = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 & 1 \\ \vdots & \iddots & \iddots & \iddots & 1 & 0 \\ \vdots & \iddots & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \iddots & \vdots \\ 0 & 1 & \iddots & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix},$$

it is true that $Q = Q^T = Q^{-1}$. By noticing the following relationship between the PAT matrix $\hat{A}$ in (2.10) and the original periodic tridiagonal matrix A in (1.2)

$$\hat{A} = AQ,$$

we can readily obtain $\hat{A}^{-1} = Q^{-1}A^{-1} = QA^{-1}$. Therefore, we can conclude that the solution of the PAT linear systems may be obtained from the solution of a PT linear system $A\mathbf{x} = \mathbf{f}$ by interchanging x_i with x_{n-i+1} for all $i = 1, 2, \ldots, n$.

3 Numerical experiments

In this section, we study the performance and effectiveness of our algorithm for two different numerical experiments. First, symbolic computations are performed in Example 3.1 by using MATLAB with Symbolic Math Toolbox. Then, we compare the proposed algorithm with three existing algorithms by means of execution times and accuracy of the solutions in Example 3.2. All experiments were carried out using MATLAB 7.12.0.635 (R2011a) on a HP Compaq 6280 Pro Microtower PC with Inter (R) Core (TM) i5-2400 processor that has 4 cores.

Example 3.1 First, we consider the following 8-by-8 PT linear system

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 11 \\ 6 \\ 12 \\ 16 \\ 20 \\ 24 \\ 28 \\ 22 \end{pmatrix}.$$

We apply the SPT algorithm to solve the above system. The results using MATLAB with Symbolic Math Toolbox are as follows.

Step 1–Step 2. By using Proposition 2.1, we have

$$L(\tau) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \tau & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{2\tau-1}{\tau} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{3\tau-2}{2\tau-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{4\tau-3}{3\tau-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{5\tau-4}{4\tau-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{6\tau-5}{5\tau-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{13\tau-5}{6\tau-5} \end{pmatrix},$$

$$M(\tau) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{\tau} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{\tau}{2\tau-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{2\tau-1}{3\tau-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{3\tau-2}{4\tau-3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \frac{4\tau-3}{5\tau-4} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -\frac{6\tau-5}{13\tau-5} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$U(\tau) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{6\tau+2}{6\tau-5} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\frac{7}{6\tau-5} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{\tau+5}{6\tau-5} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\frac{2\tau+3}{6\tau-5} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{3\tau+1}{6\tau-5} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -\frac{4\tau-1}{6\tau-5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{5\tau-3}{6\tau-5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Moreover, it follows from (2.9) that

$$\det(A) = \left(\prod_{i=1}^{8} d_i\right)\bigg|_{\tau=0} = (13\tau - 5)|_{\tau=0} = -5.$$

Step 3–Step 6. By using forward substitution and back substitution, solve the equations:

$$L(\tau)\mathbf{z} = \mathbf{f},\ M(\tau)\mathbf{y} = \mathbf{z},\ U(\tau)\mathbf{x} = \mathbf{y},$$

then we have

$$\mathbf{z} = \left(11, -\frac{5}{\tau}, \frac{12\tau+5}{2\tau-1}, \frac{20\tau-21}{3\tau-2}, \frac{40\tau-19}{4\tau-3}, \frac{56\tau-53}{5\tau-4}, \frac{84\tau-59}{6\tau-5}, \frac{48\tau-51}{13\tau-5}\right)^T,$$

$$\mathbf{y} = \left(\frac{66\tau+11}{6\tau-5}, -\frac{66}{6\tau-5}, \frac{36\tau+25}{6\tau-5}, -\frac{44}{6\tau-5}, \frac{60\tau-17}{6\tau-5}, -\frac{22}{6\tau-5}, \frac{84\tau-59}{6\tau-5}, \frac{114\tau-40}{13\tau-5}\right)^T,$$

and

$$\mathbf{x} = \left(\frac{29\tau-5}{13\tau-5}, -\frac{10}{13\tau-5}, \frac{59\tau-15}{13\tau-5}, \frac{38\tau-20}{13\tau-5}, \frac{73\tau-25}{13\tau-5}, \frac{76\tau-30}{13\tau-5}, \frac{87\tau-35}{13\tau-5}, \frac{114\tau-40}{13\tau-5}\right)^T.$$

Table 2 Numerical results of Example 3.2

Algorithms	n	100	500	1000	6000
Classical elimination algorithm	$\|\mathbf{x}-\tilde{\mathbf{x}}\|$	9.8638e−014	3.5403e−012	7.5261e−012	2.3599e−010
	Elapsed time (s)	0.0002	0.0028	0.0037	0.0093
Sherman-Morrison algorithm	$\|\mathbf{x}-\tilde{\mathbf{x}}\|$	8.9780e−014	3.8331e−012	1.7673e−011	3.5892e−010
	Elapsed time (s)	0.0002	0.0010	0.0026	0.0085
PERTRI algorithm	$\|\mathbf{x}-\tilde{\mathbf{x}}\|$	8.4695e−014	1.8187e−012	6.0934e−012	3.6330e−010
	Elapsed time (s)	0.0003	0.0012	0.0039	0.0103
Our algorithm	$\|\mathbf{x}-\tilde{\mathbf{x}}\|$	8.4371e−014	1.1544e−012	5.6300e−012	2.2561e−010
	Elapsed time (s)	0.0002	0.0008	0.0014	0.0071

Finally, setting $\tau = 0$, we obtain the solution

$$\mathbf{x} = (1, 2, 3, 4, 5, 6, 7, 8)^T.$$

Example 3.2 In order to show the efficiency and competitiveness of the symbolic algorithm described in this paper, we now consider an n-by-n PT linear system originating from [16] and is given by

$$\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 1 \\ -1 & 2 & -1 & 0 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & -1 & 2 & -1 \\ 1 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 2 \end{pmatrix}.$$

It can be readily verified that the exact solution of the system is $\tilde{\mathbf{x}} = (1, 1, \ldots, 1)^T$. For $n = 100, 500, 1000, 6000$, we used Classical elimination algorithm, Sherman-Morrison algorithm, PERTRI algorithm, and our algorithm to compute $\mathbf{x}$. The absolute errors $\|\mathbf{x}-\tilde{\mathbf{x}}\|$ and elapsed time are provided in Table 2. Here, $\|\cdot\|$ denotes the Euclidean vector norm.

We note from Table 2 that our algorithm slightly outperforms other existing algorithms for every dimension.

4 Conclusions

In this paper, we considered the solution of PT linear systems. First, we reviewed three well-known algorithms for solving the linear system (1.1). Then, in Sect. 2.2, we derived a symbolic algorithm (Algorithm 2.4) for robust computation. The algorithm is based on a special matrix decomposition which never suffers from breakdown. Finally, two numerical examples were given to demonstrate the effectiveness of our algorithm and its competitiveness with other existing algorithms.

Since the implementation of the proposed algorithm using Computer Algebra Systems (CASs) is straightforward, we believe that it will become a useful tool for solving PT linear systems.

Acknowledgments This work was supported by the Natural Science Foundation of China (NSFC) under Grant 11371287 and the International Science and Technology Cooperation Program of China under Grant 2010DFA14700.

References

1. M. Znojil, J. Math. Chem. **28**, 140 (2000)
2. A. Bjorck, G.H. Golub, SIAM Rev. **19**, 5 (1977)
3. F. Iachello, A. Del Sol Mesa, J. Math. Chem. **25**, 345 (1999)
4. Y.J. Ye, J. Math. Chem. **14**, 121 (1993)
5. J. Verkaik, H.X. Lin, Parallel Comput. **31**, 563 (2005)
6. M.M. Chawla, K. Passi, P.N. Shivakumar, Int. J. Comput. Math. **45**, 113 (1992)
7. M. Alfaro, J.M. Montaner, Numer. Algor. **10**, 137 (1995)
8. C.F. Fischer, R.A. Usmani, SIAM J. Numer. Anal. **6**, 127 (1969)
9. J.T. Jia, T. Sogabe, J. Math. Chem. **51**, 881 (2013)
10. J.T. Jia, T. Sogabe, M. El-Mikkawy, Comput. Math. Appl. **65**, 116 (2013)
11. A.A. Karawia, Appl. Math. Comput. **194**, 534 (2007)
12. D.J. Evans, Int. J. Comput. Math. **72**, 375 (1999)
13. C. Temperton, J. Comput. Phys. **19**, 317 (1975)
14. M.A.L. Marques, A. Castro, G.F. Bertsch, A. Rubio, Comput. Phys. Commun. **151**, 60 (2003)
15. T.A. Davis, *Direct Methods for Sparse Linear Systems* (SIAM, Philadelphia, 2006)
16. M.M. Chawla, R.R. Khazal, Int. J. Comput. Math. **79**, 473 (2002)
17. M. El-Mikkawy, Appl. Math. Comput. **161**, 691 (2005)
18. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins University Press, Baltimore and London, 1996)
19. M. El-Mikkawy, Appl. Math. **3**, 342 (2012)
20. M.B. Allen III, E.L. Isaacson, *Numerical Analysis for Applied Science* (Wiley-Interscience, Wiley, New York, 1997)
21. Y.L. Jiang, J.T. Jia, J. Math. Chem. **51**, 2503 (2013)
22. M. El-Mikkawy, J. Comput. Appl. Math. **166**, 581 (2004)