

A Generalized Symbolic Thomas Algorithm

Moawwad El-Mikkawy

Mathematics Department, Faculty of Science, Mansoura University, Mansoura, Egypt

Email: m_elmikkawy@yahoo.com

Received February 4, 2012; revised March 1, 2012; accepted March 8, 2012

ABSTRACT

The current paper is mainly devoted to construct a generalized symbolic Thomas algorithm that will never fail. Two new efficient and reliable computational algorithms are given. The algorithms are suited for implementation using computer algebra systems (CAS) such as Mathematica, Macsyma and Maple. Some illustrative examples are given.

Keywords: Tridiagonal Matrices; LU Factorization; Linear Systems; DETGTRI Algorithm; Thomas Algorithm

1. Introduction

The $n \times n$ general tridiagonal matrix T takes the form:

$$T = \begin{bmatrix} d_1 & a_1 & 0 & \cdots & 0 \\ b_2 & d_2 & a_2 & \ddots & \vdots \\ 0 & b_3 & d_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \cdots & 0 & b_n & d_n \end{bmatrix} \quad (1.1)$$

This type of matrices frequently appear in many scientific and engineering applications. This area of research is still very active and has been considered by many authors. The interested reader may refer to, for instance to [1-3] and the references therein. The $n \times n$ general tridiagonal matrix T in (1.1) can be stored using only three n -dimensional vectors since there is no need to store the zero elements. In this paper we shall focus on k -tridiagonal matrices [4] of order $n \times n$ of the form:

$$T_n^{(k)} = \begin{bmatrix} d_1 & 0 & \cdots & 0 & a_1 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 & a_2 & \ddots & \vdots \\ \cdots & 0 & \ddots & 0 & \cdots & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & d_{n-k} & \ddots & \cdots & \ddots & a_{n-k} \\ b_{k+1} & 0 & \cdots & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & b_{k+2} & \ddots & \cdots & 0 & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & 0 & \cdots & 0 & d_{n-1} & 0 \\ 0 & \cdots & 0 & b_n & 0 & \cdots & 0 & d_n \end{bmatrix} \quad (1.2)$$

where $1 \leq k < n$. For $k \geq n$, the matrix in (1.2) is a diagonal matrix and the case $k = 1$ gives the ordinary tridiagonal matrix in (1.1). As pointed out in [4,5], the matrix $T_n^{(k)}$ plays an important role in describing generalized k -Fibonacci numbers. Furthermore, the matrix $T_n^{(k)}$ has very recently received attention by some authors. In [5], using direct sum of matrices, it is proved that $\det(T_n^{(k)})$ may be obtained as the product of exactly k ordinary

tridiagonal matrices of the form (1.1) and this can be done using the efficient symbolic algorithm DETGTRI in [6]. The motivation of the present paper is to continue the study of the LU factorization [7] of the matrix $T_n^{(k)}$ in (1.2) using a new approach, different from the recent approach given in [8], in order to obtain a generalized symbolic Thomas algorithm that will never fail.

The organization of the paper is as follows. In Section 2, generalization of the DETGTRI algorithm in [6] is investigated. The generalization of the Thomas algorithm [7] is considered in Section 3. Some illustrative examples are given in Section 4 for the sake of illustration. A conclusion is given in Section 5.

2. Generalization of the DETGTRI Algorithm

We begin this section by giving the following result whose proof will be omitted.

Theorem 2.1. For $k = 1, 2, \dots, n-1$, the Doolittle LU factorization [7] of the k -tridiagonal matrix $T_n^{(k)}$ in (1.2) is given by:

$$T_n^{(k)} = L_n^{(k)} U_n^{(k)}, \quad (2.1)$$

where

$$L_n^{(k)} = \begin{bmatrix} 1 & & & & & & & & \\ 0 & 1 & & & & & & & \\ \cdots & 0 & \ddots & & & & & & \\ 0 & \cdots & \ddots & 1 & & & & & \\ \frac{b_{k+1}}{c_1} & 0 & \cdots & \ddots & \ddots & & & & \\ 0 & \frac{b_{k+2}}{c_2} & \ddots & \cdots & 0 & \ddots & & & \\ \vdots & \ddots & \ddots & 0 & \cdots & 0 & 1 & & \\ 0 & \cdots & 0 & \frac{b_n}{c_{n-k}} & 0 & \cdots & 0 & 1 & \end{bmatrix} \quad (2.2)$$

and

$$U_n^{(k)} = \begin{bmatrix} c_1 & 0 & \cdots & 0 & a_1 & 0 & \cdots & 0 \\ & c_2 & 0 & \cdots & 0 & a_2 & \ddots & \vdots \\ & & \ddots & 0 & \cdots & \ddots & \ddots & 0 \\ & & & c_{n-k} & \ddots & \cdots & \ddots & a_{n-k} \\ & & & & \ddots & \ddots & \cdots & 0 \\ & & & & & \ddots & 0 & \cdots \\ & & & & & & c_{n-1} & 0 \\ & & & & & & & c_n \end{bmatrix} \quad (2.3)$$

with

$$c_i = \begin{cases} d_i & i = 1, 2, \dots, k \\ d_i - \frac{b_i a_{i-k}}{c_{i-k}} & i = k + 1, k + 2, \dots, n \end{cases} \quad (2.4)$$

Now we can formulate our first result.

Algorithm 2.1. (Generalization of the DETGTRI algorithm [6])

To compute the determinant of the k -tridiagonal matrix $T_n^{(k)}$ in (1.2) we may proceed as follows:

Step 1: Compute c_1, c_2, \dots, c_n to its simplest rational forms according to (2.4), setting $c_r = \lambda$ (λ is just a symbolic name) whenever $c_r = 0, r \in \{1, 2, \dots, n\}$.

Step 2: Compute the polynomial $p(\lambda) = \prod_{r=1}^n c_r$ to its simplest rational form. Meanwhile, $\det(T_n^{(k)}) = p(0)$.

The Algorithm 2.1 will be referred to as k -DETGTRI algorithm. The DETGTRI algorithm in [6] is a special case of the k -DETGTRI algorithm when $k = 1$. Note that the k -DETGTRI algorithm computes $\det(T_n^{(k)})$ in linear time.

3. Generalization of the Thomas Numeric Algorithm

In this section, we are going to generalize the well-known Thomas numeric algorithm [7]. The purpose of the generalized algorithm is to solve the following linear system for any $k \in \{1, 2, \dots, n-1\}$.

The new algorithm can be stated as follows:

Algorithm 3.1. (Generalization of the Thomas numeric algorithm [7])

To solve the linear system of the form (3.1), we may proceed as follows:

Step 1: Compute c_1, c_2, \dots, c_n using (2.4).

Step 2: Compute z_1, z_2, \dots, z_n using

$$z_i = \begin{cases} y_i & i = 1, 2, \dots, k \\ y_i - \frac{b_i z_{i-k}}{c_{i-k}} & i = k + 1, k + 2, \dots, n \end{cases}$$

Step 3: The solution of the system (3.1) is

$$x_i = \begin{cases} \frac{z_i}{c_i} & i = n, n-1, \dots, n-k+1 \\ \frac{z_i - a_i x_{i+k}}{c_i} & i = n-k, n-k-1, \dots, 1 \end{cases}$$

The Algorithm 3.1 will be referred to as k -Thomas algorithm. The Thomas algorithm is a special case of the k -Thomas algorithm when $k = 1$.

Note 3.1. Since the k -Thomas algorithm is based on the LU factorization of the matrix, $T_n^{(k)}$, then the k -Thomas numeric algorithm may fail if $c_r = 0$,

$r \in \{1, 2, \dots, n-k\}$ as can be seen from (2.1)-(2.4). To remove all cases in which the k -Thomas numeric algorithm fails, it is convenient to give the following symbolic version of the k -Thomas numeric algorithm described above. The k -Thomas symbolic algorithm can be stated as follows:

Algorithm 3.2. (Generalized symbolic Thomas algorithm)

To solve the linear system of the form (3.1), we may proceed as follows:

Step 1: Compute c_1, c_2, \dots, c_n to its simplest rational forms using (2.4), setting $c_r = \lambda$ (λ is just a symbolic name) whenever $c_r = 0, r \in \{1, 2, \dots, n\}$.

Step 2: Compute z_1, z_2, \dots, z_n to its simplest rational forms using:

$$\begin{bmatrix} d_1 & 0 & \cdots & 0 & a_1 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 & a_2 & \ddots & \vdots \\ \cdots & 0 & \ddots & 0 & \cdots & \ddots & \ddots & 0 \\ 0 & \cdots & \ddots & d_{n-k} & \ddots & \cdots & \ddots & a_{n-k} \\ b_{k+1} & 0 & \cdots & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & b_{k+2} & \ddots & \cdots & 0 & \ddots & 0 & \cdots \\ \vdots & \ddots & \ddots & 0 & \cdots & 0 & d_{n-1} & 0 \\ 0 & \cdots & 0 & b_n & 0 & \cdots & 0 & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix} \quad (3.1)$$

$$z_i = \begin{cases} y_i & i = 1, 2, \dots, k \\ y_i - \frac{b_i z_{i-k}}{c_{i-k}} & i = k + 1, k + 2, \dots, n \end{cases}$$

Step 3: The solution of the system (3.1) is

$$x_i = \begin{cases} \frac{z_i}{c_i} & i = n, n - 1, \dots, n - k + 1 \\ \frac{z_i - a_i x_{i+k}}{c_i} & i = n - k, n - k - 1, \dots, 1 \end{cases}$$

evaluated at $\lambda = 0$.

Before we consider the solution of the system (3.1) by using the k -Thomas algorithm, it is recommended to use the k -DETGTTRI algorithm to check the nonsingularity of the coefficient matrix $T_n^{(k)}$ firstly.

4. Some Illustrative Examples

In this section we are going to give some illustrative examples.

Example 4.1. Consider the following k -tridiagonal system with $n = 10$ and $k = 4$:

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 3 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_3 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \\ 13 \\ 12 \\ 10 \\ 0 \\ 9 \\ 0 \\ 4 \end{bmatrix} \tag{4.1}$$

Applying the numeric or the symbolic version of the k -Thomas algorithms gives:

$$c_1 = 2, c_2 = 1, c_3 = -1, c_4 = 3, c_5 = 3, c_6 = -3, c_7 = 11,$$

$$c_8 = \frac{1}{3}, c_9 = -\frac{4}{3}, c_{10} = 8 \text{ (Step 1);}$$

Hence $\det(T_n^{(k)}) = -2112$. Thus $T_n^{(k)}$ is nonsingular.

- $z_1 = 4, z_2 = 2, z_3 = 0, z_4 = 13, z_5 = 8, z_6 = 12, z_7 = 0,$
 $z_8 = \frac{1}{3}, z_9 = -\frac{8}{3}, z_{10} = 24$ (Step 2);
- $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 3, x_5 = 2, x_6 = -1, x_7 = 0, x_8 = 1,$
 $x_9 = 2, x_{10} = 3$ (Step 3).

Example 4.2. Consider the following k -tridiagonal system with $n = 10$ and $k = 4$ (this example is Example 4.1 with slight modifications in some positions. The

modified positions are written in bold):

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 3 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_3 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \\ 13 \\ \mathbf{6} \\ 10 \\ 0 \\ 9 \\ 0 \\ 4 \end{bmatrix} \tag{4.2}$$

In this case the k -Thomas numeric algorithm fails since $c_5 = 0$ as can be easily checked. However, the k -Thomas symbolic algorithm yields:

- $c_1 = 2, c_2 = 1, c_3 = -1, c_4 = 3, c_5 = \lambda, c_6 = -3, c_7 = 11,$
 $c_8 = \frac{1}{3}, c_9 = -\frac{(\lambda + 1)}{\lambda}, c_{10} = 8$ (Step 1);

Hence $\det(T_n^{(k)}) = -528(\lambda + 1)|_{\lambda=0} = -528$. Consequently

the matrix $T_n^{(k)}$ is nonsingular.

- $z_1 = 4, z_2 = 2, z_3 = 0, z_4 = 13, z_5 = 8, z_6 = 12, z_7 = 0,$
 $z_8 = \frac{1}{3}, z_9 = -\frac{2}{\lambda}, z_{10} = 24$ (Step 2);
- $x_1 = \frac{2\lambda + 1}{\lambda + 1}|_{\lambda=0} = 1, x_2 = 1, x_3 = 0, x_4 = 3,$
 $x_5 = \frac{2}{\lambda + 1}|_{\lambda=0} = 2, x_6 = -1, x_7 = 0, x_8 = 1, x_9 = \frac{2}{\lambda + 1}|_{\lambda=0} = 2,$
 $x_{10} = 3$ (Step 3).

Example 4.3. Consider the k -tridiagonal matrix

$$T_n^{(k)} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 3 \\ 0 & 0 & 3 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & -5 \end{bmatrix} \tag{4.3}$$

For this matrix, the k -DETGTTRI algorithm gives:

$$c_1 = 2, c_2 = 1, c_3 = -1, c_4 = 3, c_5 = \lambda, c_6 = -3, c_7 = 11, c_8 = \frac{1}{3},$$

$$c_9 = -\frac{(\lambda + 1)}{\lambda}, c_{10} = \lambda \text{ (Step 1)}$$

Hence $\det(T_n^{(k)}) = -66\lambda(\lambda+1)|_{\lambda=0} = 0$. Therefore the matrix $T_n^{(k)}$ is singular.

5. Conclusion

A generalized symbolic Thomas algorithm, that will never fail, is developed, for the first time, in this paper. The algorithm is suited for implementation using the Computer Algebra Systems (CAS) such as Macsyma, Maple and Mathematica.

REFERENCES

- [1] 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. [doi:10.1016/S0096-3003\(02\)00212-6](https://doi.org/10.1016/S0096-3003(02)00212-6)
- [2] 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. [doi:10.1016/S0096-3003\(03\)00298-4](https://doi.org/10.1016/S0096-3003(03)00298-4)
- [3] M. E. A. El-Mikkawy and A. Karawia, "Inversion of General Tridiagonal Matrices," *Applied Mathematics Letters*, Vol. 19, No. 8, 2006, pp. 712-720. [doi:10.1016/j.aml.2005.11.012](https://doi.org/10.1016/j.aml.2005.11.012)
- [4] M. El-Mikkawy and T. Sogabe, "A New Family of k-Fibonacci Numbers," *Applied Mathematics and Computation*, Vol. 215, No. 12, 2010, pp. 4456-4461. [doi:10.1016/j.amc.2009.12.069](https://doi.org/10.1016/j.amc.2009.12.069)
- [5] T. Sogabe and M. El-Mikkawy, "Fast Block Diagonalization of k-Tridiagonal Matrices," *Applied Mathematics and Computation*, Vol. 218, No. 6, 2011, pp. 2740-2743. [doi:10.1016/j.amc.2011.08.014](https://doi.org/10.1016/j.amc.2011.08.014)
- [6] M. E. A. El-Mikkawy, "A Fast Algorithm for Evaluating nth Order Tridiagonal Determinants," *Journal of Computational and Applied Mathematics*, Vol. 166, No. 2, 2004, pp. 581-584. [doi:10.1016/j.cam.2003.08.044](https://doi.org/10.1016/j.cam.2003.08.044)
- [7] R. L. Burden and J. D. Faires, "Numerical Analysis," 7th Edition, Books & Cole Publishing, Pacific Grove, 2001.
- [8] A. Yalciner, "The LU Factorization and Determinants of the k-Tridiagonal Matrices," *Asian-European Journal of Mathematics*, Vol. 4, No. 1, 2011, pp. 187-197. [doi:10.1142/S1793557111000162](https://doi.org/10.1142/S1793557111000162)