JAMCM Journal of Applied Mathematics and Computational Mechanics 2018, 17(3), 83-95
www.amcm.pcz.pl p-ISSN 2299-9965
DOI: 10.17512/jamcm.2018.3.08 e-ISSN 2353-0588

A NUMERICAL ALGORITHM FOR COMPUTING THE INVERSE OF A TOEPLITZ PENTADIAGONAL MATRIX

***Boutaina Talibi*[1]*, Ahmed Driss Aiat Hadj*[2]*, Driss Sarsri*[1]**

[1] *National School of Applied Sciences, Abdelmalek Essaadi University*
Tangier, Morocco
[2] *Regional Center of the Trades of Education and Training (CRMEF)-Tangier, Avenue My Abdelaziz Souani, BP: 3117, Tangier, Morocco*
talibi.boutaina@gmail.com, ait_hadj@yahoo.com, dsarsri@yahoo.fr

Received: 2 August 2018; Accepted: 27 November 2018

Abstract. In the current paper, we present a computationally efficient algorithm for obtaining the inverse of a pentadiogonal toeplitz matrix. Few conditions are required, and the algorithm is suited for implementation using computer algebra systems.

MSC 2010: *15B05 65F40 33C45*
Keywords: *pentadiagonal matrix, Toeplitz matrix, K-Hessenberg matrix, inverse*

1. Introduction

Pentadiagonal Toeplitz matrices often occur when solving partial differential equations numerically using the finite difference method, the finite element method and the spectral method, and could be applied to the mathematical representation of high dimensional, nonlinear electromagnetic interference signals [1-7]. Pentadiagonal matrices are a certain class of special matrices, and other common types of special matrices are Jordan, Frobenius, generalized Vandermonde, Hermite, centrosymmetric, and arrowhead matrices [8-12].

Typically, after the original partial differential equations are processed with these numerical methods, we need to solve the pentadiagonal Toeplitz systems of linear equations for obtaining the numerical solutions of the original partial differential equations. Thus, the inversion of pentadiagonal Toeplitz matrices is necessary to solve the partial differential equations in these cases.

J. Jia, T. Sogabe and M. El-Mikkawy recently proposed an explicit inverse of the tridiagonal Toeplitz matrix which is strictly row diagonally dominant [13]. However, this result cannot be generalized to the case of the pentadiagonal Toeplitz matrix directly. This motivates us to investigate whether an explicit inverse of the pentadiagonal Toeplitz matrix also exists.

In this paper, we present a new algorithm with a totally different approach for numerically inversing a pentadiagonal matrix. Few conditions are required, and the algorithm is suited for implementation using computer algebra systems.

2. Inverse of a Toeplitz pentadiagonal matrix algoritm 1

Definition: We consider the pentadiagonal Toeplitz matrix:

$$\mathbf{P}=\begin{bmatrix} a & b & c & 0 & \cdots & 0 \\ \alpha & a & b & c & \ddots & \vdots \\ \beta & \alpha & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & c \\ \vdots & \ddots & \ddots & \ddots & \ddots & b \\ 0 & \cdots & 0 & \beta & \alpha & a \end{bmatrix}$$

Where: $\mathbf{P}\in \mathrm{M}_{n,n}(\mathrm{K})$. Also a,b,c,α and β are an arbitrary numbers.
Assume that $\mathbf{P}$ is non-singular and denote:

$$\mathbf{P}^{-1}=(C_1,C_2,...,C_n)$$

Where $(C_i)_{1\le i\le n}$ are the columns of the inverse $\mathbf{P}^{-1}$.

From the relation $\mathbf{P}^{-1}\mathbf{P}=I_n$. Where I_n denotes the identity matrix of order n. We deduce the relations:

$$\begin{aligned} C_{n-2} &= \frac{1}{c}(E_n - bC_{n-1} - aC_n) \\ C_{n-3} &= \frac{1}{c}(E_{n-1} - bC_{n-2} - aC_{n-1} - \alpha C_{n-2}) \\ C_{j-2} &= \frac{1}{c}(E_j - bC_{j-1} - aC_j - \alpha C_{j+1} - \beta C_{j+2}) \text{ for } j=n-2,\dots,3 \end{aligned} \tag{1}$$

Where $E_j=[(\delta_{i,j})_{1\le i\le n}]^t\in \mathbb{k}^n$ is the vector of order j of the canonical basis of $\mathbb{k}^n$.
Consider the sequence of numbers $(A_i)_{1\le i\le n}$ and $(B_i)_{1\le i\le n}$ characterized by a term recurrence relation:

$$\begin{aligned} &A_0 = 0 \\ &A_1 = 1 \\ &aA_0 + bA_1 + cA_2 = 0 \\ &\alpha A_0 + aA_1 + bA_2 + cA_3 = 0 \end{aligned} \tag{2}$$

And

$$\beta A_{i-3} + \alpha A_{i-2} + aA_{i-1} + bA_i + cA_{i+1} = 0 \text{ for } i \geq 3$$

Also we have:

$$\begin{aligned} &B_0 = 0 \\ &B_1 = 1 \\ &aB_0 + bB_1 + cB_2 = 0 \\ &\alpha B_0 + aB_1 + bB_2 + cB_3 = 0 \end{aligned} \tag{3}$$

And

$$\beta B_{i-3} + \alpha B_{i-2} + aB_{i-1} + bB_i + cB_{i+1} = 0 \text{ for } i \geq 3$$

We can give a matrix form to this term recurrence:

$$\mathbf{PA} = -A_n E_{n-1} - A_{n+1} E_n \tag{4}$$

$$\mathbf{PB} = -B_n E_{n-1} - B_{n+1} E_n \tag{5}$$

Where $\mathbf{A} = [A_0, A_1, \ldots, A_{n-1}]^t$ and $\mathbf{B} = [B_0, B_1, \ldots, B_{n-1}]^t$.
Let's define for each $0 \leq i \leq n+1$

$$X_i = \det\begin{bmatrix} A_{n+1} & A_i \\ B_{n+1} & B_i \end{bmatrix} \text{ and } Y_i = \det\begin{bmatrix} A_n & A_i \\ B_n & B_i \end{bmatrix} \tag{6}$$

Immediately we have:

$$\mathbf{PX} = -X_n E_{n-1} \text{ and } \mathbf{PY} = -Y_{n+1} E_n \tag{7}$$

Where $\mathbf{X} = [X_0, X_1, \ldots, X_{n-1}]^t$ and $\mathbf{Y} = [Y_0, Y_1, \ldots, Y_{n-1}]^t$.
Lemma: If $X_n = 0$ the matrix $\mathbf{P}$ is singular.
Proof: If $X_n = 0$. The $\mathbf{X}$ is non-null and by (7) we have $\mathbf{PX} = 0$. We conclude that $\mathbf{P}$ is singular.
THEOREM 2.1: We supposed that $X_n \neq 0$ then $\mathbf{P}$ is invertible. And:

$$\begin{aligned} C_n &= \left[\frac{-Y_0}{Y_{n+1}}, \frac{-Y_1}{Y_{n+1}}, \ldots, \frac{-Y_{n-1}}{Y_{n+1}}\right]^t \\ C_{n-1} &= \left[\frac{-X_0}{X_n}, \frac{-X_1}{X_n}, \ldots, \frac{-X_{n-1}}{X_n}\right]^t \end{aligned} \tag{8}$$

Proof: We have $\det(\mathbf{P}) = c^{n-2}X_n \neq 0$ [14]. So $\mathbf{P}$ is invertible from the relations Equations (7) and (8) we have $\mathbf{P}C_n = E_n$ and $\mathbf{P}C_{n-1} = E_{n-1}$. The proof is completed.

Algorithm: New efficient computational algorithm for computing the inverse of the pentadiagonal matrix.

INPUT:
The dimension n.
The arbitrary numbers: β, α, a, b and c.
OUTPUT:
The inverse matrix $\mathbf{P}^{-1} = (C_1, C_2, \ldots, C_n)$.
Step 1:

$$C_n = \left[\frac{-Y_0}{Y_{n+1}}, \frac{-Y_1}{Y_{n+1}}, \ldots, \frac{-Y_{n-1}}{Y_{n+1}}\right]^t$$

$$C_{n-1} = \left[\frac{-X_0}{X_n}, \frac{-X_1}{X_n}, \ldots, \frac{-X_{n-1}}{X_n}\right]^t$$

Step 2:

$$C_{n-2} = \frac{1}{c}(E_n - bC_{n-1} - aC_n),$$

$$C_{n-3} = \frac{1}{c}(E_{n-1} - bC_{n-2} - aC_{n-1} - \alpha C_n),$$

Step 3:

$$C_{j-2} = \frac{1}{c}(E_j - bC_{j-1} - aC_j - \alpha C_{j+1} - \beta C_{j+2}) \text{ for } j = n-2, \ldots, 3$$

3. Inverse of a Toeplitz pentadiagonal matrix algoritm 2

Definition: Hessenberg (or lower Hessenberg) matrices are the matrices $H = [h_{ij}]$ satisfying the condition $h_{ij} = 0$ for $j - i > 1$. More generally, the matrix is k-Hessenberg if and only if $H_{ij} = 0$ for $j - i > k$.

THEOREM [15]: Let H be a strict-Hessenberg matrix with the block decomposition:

$$H = \begin{bmatrix} B & A \\ D & C \end{bmatrix}, \ A \in \mathrm{M}_{n-k}(F), \ B \in \mathrm{M}_{(n-k)\times k}(F), \ C \in \mathrm{M}_{k\times(n-k)}(F), \ D \in \mathrm{M}_k(F). \quad (9)$$

The H is invertible if and only if $CA^{-1}B-D$ is invertible and if H is invertible we have:

$$H^{-1}=\begin{bmatrix}0 & 0\\ A^{-1} & 0\end{bmatrix}-\begin{bmatrix}I_k\\ -A^{-1}B\end{bmatrix}(CA^{-1}B-D)^{-1}\begin{bmatrix}-CA^{-1} & I_k\end{bmatrix} \tag{10}$$

R. Slowik [16] worked in the particular case of the Toeplitz-Hessenberg matrix.
Our work is applied on pentdiagonal matrix **P** of the form:

$$\mathbf{P}=\begin{bmatrix}a & b & c & 0 & \cdots & 0\\ \alpha & a & b & c & \ddots & \vdots\\ \beta & \alpha & \ddots & \ddots & \ddots & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & c\\ \vdots & \ddots & \ddots & \ddots & \ddots & b\\ 0 & \cdots & 0 & \beta & \alpha & a\end{bmatrix} \tag{11}$$

Then we find:

$$\mathbf{P}^{-1}=\begin{bmatrix}0 & 0\\ A^{-1} & 0\end{bmatrix}-\begin{bmatrix}I_2\\ -A^{-1}B\end{bmatrix}\left(CA^{-1}B-D\right)^{-1}\begin{bmatrix}-CA^{-1} & I_2\end{bmatrix} \tag{12}$$

The blocks B, C, D and A are given as:

$$B=\begin{bmatrix}a & b\\ \alpha & a\\ \beta & \alpha\\ 0 & \beta\\ 0 & 0\\ \vdots & \vdots\\ 0 & 0\end{bmatrix} \tag{13}$$

Also:

$$C=\begin{bmatrix}0 & \cdots & 0 & \beta & \alpha & a & b\\ 0 & \cdots & \cdots & 0 & \beta & \alpha & a\end{bmatrix} \tag{14}$$

$$D=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} \tag{15}$$

And:

$$A=\begin{bmatrix} c & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ b & c & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ a & b & c & 0 & \cdots & \cdots & \cdots & 0 \\ \alpha & a & b & c & 0 & \cdots & \cdots & 0 \\ \beta & \alpha & a & b & c & 0 & \cdots & \vdots \\ 0 & \beta & \alpha & a & b & c & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta & \alpha & a & b & c \end{bmatrix} \tag{16}$$

Then the inverse of the matrix A will be:

$$A^{-1}=\frac{1}{c}\sum_{k=0}^{n-2}\sum_{r=1}^{n-1-k} l_k E_{r+k,r} \tag{17}$$

Where $l_0=1$ and $l_k=-\frac{1}{c}(bl_{k-1}+al_{k-2}+\alpha l_{k-3}+\beta l_{k-4})$ for $k\geq 1$.

Proof: Since A is a triangular matrix, A^{-1} as well. We prove (17) inductively on n. We have:

$$\begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & \frac{-b}{c} & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ \frac{-b}{c} & 1 & \\ \frac{-a}{c} & 0 & 1 \end{bmatrix}=\begin{bmatrix} 1 & & \\ -l_1 & 1 & \\ l_2 & l_1 & 1 \end{bmatrix} \tag{18}$$

With:

$$l_1=-\frac{b}{c};\ l_2=\left(\frac{b}{c}\right)^2-\frac{a}{c}=-\frac{1}{c}(bl_1+al_0) \tag{19}$$

The first step of induction holds.
Consider now $n>4$. We have

$$\begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & l_1 & 1 & & \\ \vdots & \vdots & \ddots & & \\ 0 & l_{n-3} & \cdots & l_1 & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ -\frac{b}{c} & 1 & & & \\ -\frac{a}{c} & & 1 & & \\ 0 & & & 1 & \\ \vdots & & & & \ddots \\ 0 & & & & & 1 \end{bmatrix}=\begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ l_2 & l_1 & 1 & & \\ \vdots & \vdots & \ddots & & \\ l_{n-2} & l_{n-3} & \cdots & l_1 & 1 \end{bmatrix} \tag{20}$$

Then: $l_{n-2} = -\frac{1}{c}(bl_{n-3} + al_{n-4} + \alpha l_{n-5} + \beta l_{n-6})$

4. Example

In this section, we give a numerical example to illustrate the effectiveness of our algorithm. Our algorithm is tested by MATLAB R2014a.
Consider the following 7-by-7 pentadiagonal matrix

$$\mathbf{P} = \begin{bmatrix} 2 & 3 & 4 & 0 & 0 & 0 & 0 \\ 5 & 2 & 3 & 4 & 0 & 0 & 0 \\ 6 & 5 & 2 & 3 & 4 & 0 & 0 \\ 0 & 6 & 5 & 2 & 3 & 4 & 0 \\ 0 & 0 & 6 & 5 & 2 & 3 & 4 \\ 0 & 0 & 0 & 6 & 5 & 2 & 3 \\ 0 & 0 & 0 & 0 & 6 & 5 & 2 \end{bmatrix} \tag{21}$$

The columns of the inverse $\mathbf{P}^{-1}$ are:

$$C_1 = \begin{bmatrix} -1.0787 \\ -3.4400 \\ 3.3693 \\ 0.5414 \\ 3.8273 \\ -2.1929 \\ -5.9997 \end{bmatrix}, \ C_2 = \begin{bmatrix} -0.1948 \\ -1.2309 \\ 1.0205 \\ 0.3435 \\ 1.0629 \\ -0.3983 \\ -2.1929 \end{bmatrix}, \ C_3 = \begin{bmatrix} 0.6886 \\ 2.1724 \\ -1.9736 \\ -0.4667 \\ -2.1615 \\ 1.0629 \\ 3.8273 \end{bmatrix}$$

$$C_4 = \begin{bmatrix} 0.0305 \\ 0.4866 \\ -0.3802 \\ 0.0037 \\ -0.4667 \\ 0.3435 \\ 0.5414 \end{bmatrix}, C_5 = \begin{bmatrix} 0.5616 \\ 1.7791 \\ -1.6151 \\ -0.3802 \\ -1.9736 \\ 1.0205 \\ 3.3693 \end{bmatrix}, \ C_6 = \begin{bmatrix} -0.6926 \\ -1.9104 \\ 1.7791 \\ 0.4866 \\ 2.1724 \\ -1.2309 \\ -3.4400 \end{bmatrix}, \ C_7 = \begin{bmatrix} -0.0843 \\ -0.6926 \\ 0.5616 \\ 0.0305 \\ 0.6886 \\ -0.1948 \\ -1.0787 \end{bmatrix}$$

Let's employ the same example and give the blocks A, B, C, and D of the matrix **P**

$$A=\begin{bmatrix}4&0&0&0&0\\3&4&0&0&0\\2&3&4&0&0\\5&2&3&4&0\\6&5&2&3&4\end{bmatrix} \tag{22}$$

$$B=\begin{bmatrix}2&3\\5&2\\6&5\\0&6\\0&0\end{bmatrix} \tag{23}$$

$$C=\begin{bmatrix}0&6&5&2&3\\0&0&6&5&2\end{bmatrix} \tag{24}$$

$$D=\begin{bmatrix}0&0\\0&0\end{bmatrix} \tag{25}$$

Therefore the inverse $\mathbf{P}^{-1}$ is:

$$\mathbf{P}^{-1}=\begin{bmatrix}-1.0787&-0.1948&0.6886&0.0305&0.5616&-0.6926&-0.0843\\-3.4400&-1.2309&2.1724&0.4866&1.7791&-1.9104&-0.6926\\3.3693&1.0205&-1.9736&-0.3802&-1.6151&1.7791&0.5616\\0.5414&0.3435&-0.4667&0.0037&-0.3802&0.4866&0.0305\\3.8273&1.0629&-2.1615&-0.4667&-1.9736&2.1724&0.6886\\-2.1929&-0.3983&1.0629&0.3435&1.0205&-1.2309&-0.1948\\-5.9997&-2.1929&3.8273&0.5414&3.3693&-3.4400&-1.0787\end{bmatrix} \tag{26}$$

As we can see, the two methods give the same result.

In Table 1 we give a comparison of the running time between the 'Toeplitz--Hessenberg' algorithm, our algorithm in MATLAB R2014a and the famous method LU.

The running time (in seconds) of two algorithms in MATLAB R2014a.

Table 1

The running time

Size of the matrix (n)	Toepltiz-Hessenberg algorithm	The fast algorithm	LU method
100	0.132870	0.020556	0.807013
200	0.514653	0.035023	4.424515
300	1.157204	0.050520	10.066555
500	3.212367	0.083163	34.979610
1000	12.846616	0.179746	418.104899

5. Conclusions

In this paper, numerical algorithms for computing the inverse of a pentadiagonal Toeplitz matrix are presented. We have showed that the computational cost of our algorithm for finding the inverse of pentadiagonal Toeplitz matrix is much less than those of well-known algorithms. From some numerical examples we have learned that our algorithms work as well as some other well-known algorithms.

Algorithm for Inverse of a Toeplitz pentadiagonal matrix algorithm 1

```
clear all;
clc;
close all;
s=10;
d = ones(s,1);
P = spdiags([6*d 5*d 2*d 3*d 4*d], -2:2,s ,s );
tic

n=length(P);

b(n)=0;
c(n)=1;c(n-1)=1;
a=diag(P);
b=[diag(P,1); b(n)];
c=[diag(P,2); c(n-1); c(n)];
fa=[0 ;diag(P,-1)];
ta=[0; 0 ;diag(P,-2)];

e1=eye(n);

A0=0;
```

```
A(1)=1;
A(2)=(-1/c(1))*(a(1)*A0+b(1)*A(1));
A(3)=(-1/c(2))*(a(2)*A(1)+b(2)*A(2)+fa(2)*A0);
A(4)=(-1/c(3))*(a(3)*A(2)+b(3)*A(3)+fa(3)*A(1)+ta(3)*A0);

for i=4:n
    A(i+1)=(-1/c(i))*(a(i)*A(i-1)+b(i)*A(i)+fa(i)*A(i-2)+ta(i)*A(i-3));
end
AA=zeros(n,1);
AA(1)=A0;

for k=2:n
AA(k)=A(k-1);
end

B0=1;
B(1)=0;
B(2)=(-1/c(1))*(a(1)*B0+b(1)*B(1));
B(3)=(-1/c(2))*(a(2)*B(1)+b(2)*B(2)+fa(2)*B0);
B(4)=(-1/c(3))*(a(3)*B(2)+b(3)*B(3)+fa(3)*B(1)+ta(3)*B0);

for i=4:n
    B(i+1)=(-1/c(i))*(a(i)*B(i-1)+b(i)*B(i)+fa(i)*B(i-2)+ta(i)*B(i-3));
end

BB=zeros(n,1);
BB(1)=B0;
for k=2:n
BB(k)=B(k-1);
end

Q0=A(n);
for i=1:n-1
   QQ(i)=det([ A(n) A(i);B(n) B(i)]) ;
end
Q=[Q0 QQ];
QN=det([ A(n) A(n+1);B(n) B(n+1)]);


R0=A(n+1)*B0;
for i=1:n-1
    RR(i)=det([ A(n+1) A(i);B(n+1) B(i)]) ;
 end
 R=[R0 RR];
RN=det([ A(n+1) A(n);B(n+1) B(n)]) ;
```

```
C=zeros(n,n);
C(:,n)=(-1/QN)*Q;
C(:,n-1)=(-1/RN)*R;

C(:,n-2)=(1/c(n-2))*(e1(:,n)-a(n)*C(:,n)-b(n-1)*C(:,n-1));
C(:,n-3)=(1/c(n-3))*(e1(:,n-1)-a(n-1)*C(:,n-1)-b(n-2)*C(:,n-2)-
fa(n)*C(:,n));
C(:,n-4)=(1/c(n-4))*(e1(:,n-2)-a(n-2)*C(:,n-2)-b(n-3)*C(:,n-3)-fa(n-
1)*C(:,n-1)-ta(n)*C(:,n));
for j=n-3:-1:3
C(:,j-2)=(1/c(j-2))*(e1(:,j)-a(j)*C(:,j)-b(j-1)*C(:,j-1)-fa(j+1)*C(:,j+1)-
ta(j+2)*C(:,j+2));
end
%C
%I=C*P
toc
```

Algorithm for the inverse of a Toeplitz pentadiagonal matrix algorithm 2

```
close all
clear all
clc
n=7;

d = ones(n,1);
P = spdiags([6*d 5*d 2*d 3*d 4*d], -2:2,n ,n );
%P=[2 4 5 0 0 0 0; 3 6 5 8 0 0 0; 1 1 2 3 7 0 0; 0 8 9 1 2 3 0;0 0 7 9 1 2
5; 0 0 0 2 3 4 5;0 0 0 0 2 5 6];
tic

B=P(1:end-2,1:2);

C=P(end-1:end,3:end);

D=zeros(2,2);

M=fct_calclul_inverse_A(P);
H=[zeros(2,n-2) zeros(2,2) ; M zeros(n-2,2)]-[eye(2);-M*B]*(C*M*B-D)^(-1)*[-
C*M eye(2)];
%I=H*P
toc
```

```
function A=fct_calclul_inverse_A(P)

m=size(P);
n=m(:,1);
ar=P(:,3);
% calcul b_k
l0=1;
l=zeros(n-2,1);
l(1)=(-ar(2)/ar(1))*l0;
l(2)=-(1/ar(1))*(ar(2)*l(1)+ar(3)*l0);
for k=3:n-2
    for r=0:k-2
     l(k)=l(k)-(1/ar(1))*(ar(r+2)*l(k-r-1));
      r=r+1;
    end
        l(k)=l(k)-(1/ar(1))*ar(k+1)*l0;
end

E=ones(n-2);
A=zeros(n-2,n-2);
b=[l0 ;l];

for k=0:n-2
   for r=1:n-2-k
       A(r+k,r)=A(r+k,r)+1/ar(1)*(b(k+1)*E(r+k,r));
   end
end
end
```

References

[1] Aceto, L., Ghelardoni, P., & Magherini, C. (2012). PGSCM: A family of P-stable boundary value methods for second-order initial value problems. *J. Comput. Appl. Math.*, *236*, 3857-3868; El-Mikkawy, M., & Karawia, A. (2006). Inversion of general tridiagonal matrices. *Appl. Math. Lett.*, *19*, 712720.

[2] Aceto, L., Ghelardoni, P., & Magherini, C. (2012). Boundary value methods for the reconstruction of SturmLiouville potentials. *Appl. Math. Comput.*, *219*, 2960-2974; Losiak, J., Neuman, E., & Nowak, J. (1988).The inversion of cyclic tridiagonal matrices. *Zastos. Mat.*, *20*(1), 93102.

[3] Kaya, D. (2003). An explicit and numerical solutions of some fifth-order Kdv equation by decomposition method. *Appl. Math. Comput.*, 144, 353-363.

[4] El-Sayed, S.M., & Kaya, D. (2004). An application of the ADM to seven order Sawada Kotara equations. *Appl. Math. Comput.*, *157*, 93-101.

[5] Shen, Jie, & Tang, Tao (2006). *Spectral and High-Order Methods with Applications*. Beijing: Science Press.

[6] Monterde, J., & Ugail, H. (2006). A general 4th-order PDE method to generate Bezier surfaces from the boundary. Comput. *Aided Geom. Design*, *23*, 208-225.

[7] Patil, P.G., & Swamy, Y.S. (2008). An eficient model for vibration control by piezoelectric smart structure using finite element method. *Eur. J. Comput. Sci. Netw. Secu.*, *8*, 258-264.

[8] Respondek, J.S. (2011). Numerical recipes for the high efficient inverse of the confluent Vandermonde matrices. *Appl. Math. Comput.*, *218*, 2044-2054.

[9] Respondek, J.S. (2013). Recursive numerical recipes for the high efficient inversion of the confluent Vandermonde matrices. *Appl. Math. Comput.*, *225*, 718-730.

[10] Li, H., & Zhao, D. (2014). An extension of the Golden-Thompson theorem. *J. Inequal. Appl.*

[11] Li, H.Y., Gong, Z.G., & Zhao, D. (2014). Least squares solutions of the matrix equation AXB+CYD=E with the least norm for symmetric arrowhead matrices. *Appl. Math. Comput.*, *226*, 719-724.

[12] Wang, Chaojie, Li, Hongyi, & Zhao, Di (2015). An explicit formula for the inverse of a pentadiagonal Toeplitz matrix. *Journal of Computational and Applied Mathematics*, *278*, 12-18.

[13] Hadj, A., & Elouafi, M. (2008). A fast numerical algorithm for the inverse of a tridiagonal and pentadiagonal matrix. *Appl. Math. Comput.*, 202, 441-445.

[14] Hadj, D.A., & Elouafi, M. (2008). The characteristic polynomial, eigenvectors and determinant of a pentadiagonal matrix. *Appl. Math. Comput.*, *198*(2), 634-642.

[15] Verde-Star, L. (2015). Elementary triangular matrices and inverses of k-Hessenberg and triangular matrices. *Spec. Matrices*, *3*, 250256.

[16] Slowik, R. (2018). Inverses and determinants of Toeplitz-Hessenberg matrices. *Taiwanese Journal of Mathematics*, *22*, 4, 901-908.