**Journal of Mathematical Modeling**
Vol. 10, No. 4, 2022, pp. 453-461. Research Article

JMM

# A direct solver for solving systems of linear equations with banded ill-conditioned Toeplitz matrices

**Nasser Akhoundi***

*School of Mathematics and Computer Science, Damghan University, Damghan, Iran*
*Email(s): akhoundi@du.ac.ir*

**Abstract.** In this paper, the banded Toeplitz matrices generated by $f(\theta) = (2(1-\cos(\theta-\tilde{\theta})))^d$ are studied. The function $f$ is a real non-negative function with a zero of order $2d$ at $\tilde{\theta}$ and the generated matrices are ill-conditioned Hermitian positive definite. We show that these banded Toeplitz matrices are similar to the banded real symmetric positive definite Toeplitz matrices that are generated by $f(\theta) = (2(1-\cos(\theta)))^d$. A fast direct solver is proposed to compute the inverse of these real matrices. Numerical experiments show that our proposed method is faster and more stable than the stable Levinson algorithm.

*Keywords*: Toeplitz matrices, fast Toeplitz solver, Levinson algorithm.
*AMS Subject Classification 2010*: 65F05, 15B05, 65F15

## 1 Introduction

An $n \times n$ banded Hermitian Toeplitz matrices with bandwidth $2d-1$ can be defined as follows

$$T_n^{(d)} = \begin{pmatrix} t_0 & \bar{t}_1 & \cdots & \bar{t}_{d-1} & \cdots & 0 \\ t_1 & t_0 & \bar{t}_1 & \cdots & \bar{t}_{d-1} & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ t_{d-1} & \cdots & t_1 & t_0 & \cdots & 0 \\ \vdots & \ddots & & & \ddots & \vdots \\ 0 & \cdots & t_{d-1} & \cdots & t_1 & t_0 \end{pmatrix}. \tag{1}$$

*Corresponding author.
Received: 11 May 2022 / Revised: 14 July 2022 / Accepted: 7 August 2022
DOI: 10.22124/JMM.2022.22278.1965

 

We see that $T_n^{(d)} = \left(t_{i-j}\right)_{i,j=1,2,\ldots,n}$, where $t_{-k} = \bar{t}_k$ and $t_k = 0$ for $|k| \geq d$. As known, the Toeplitz matrices $T_n = \left(t_{i-j}\right)_{i,j=1,\ldots,n}$ can be interpreted as the Fourier coefficients of the generating function

$$f(\theta) = \sum_{-\infty}^{\infty} t_k e^{ik\theta}, \tag{2}$$

defined on $[-\pi, \pi]$, i.e.,

$$t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \qquad k = 0, \pm 1, \pm 2, \ldots. \tag{3}$$

If the generating function $f(\theta)$ is non-negative real, then $T_n$ is Hermitian positive definite (HPD) matrix. Furthermore for the even function $f$, the Toeplitz matrix $T_n$ is a real symmetric matrix.

In this paper we assume that

$$f(\theta) = (2(1 - \cos(\theta - \tilde{\theta})))^d, \tag{4}$$

where $\tilde{\theta} \in [-\pi, \pi]$ is a constant number. In this case we denote the generated Toeplitz matrix by $T_{\tilde{\theta},n}^{(d)}$. $T_{\tilde{\theta},n}^{(d)}$ is banded HPD, with bandwidth $2d-1$ (as we show in (1)). For the special case $\tilde{\theta} = 0$, we omit the subscript $\tilde{\theta}$ in $T_{\tilde{\theta},n}^{(d)}$ and we name it $T_n^{(d)}$. These linear systems arise in the discretization of differential equations. In [4] and [5] the authors showed that these banded Toeplitz matrices can be performed as a good preconditioner.

Function $f$ in (4) has zero of order $2d$ at $\tilde{\theta}$, hence the condition number of these matrices can be very large. As an example for $d = 2$, in [1], the authors showed that the condition number of these matrices is about $16^2(\frac{n+2}{3\pi})^4$.

Some of the terminology used in this paper will be given for convenience. $T_f$ is the infinity Toeplitz matrix that is generated by the function $f$, and defined by

$$T_f = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & t_{-3} & \cdots \\ t_1 & t_0 & t_{-1} & t_{-2} & \cdots \\ t_2 & t_{-1} & t_0 & t_{-1} & \cdots \\ \vdots & & \ddots & & \vdots \end{pmatrix}, \tag{5}$$

and $T_{f,n}$ denotes its $n$-by-$n$ leading principal submatrix. We define the Hankle matrix generated by $f$ as follows

$$H_f = \begin{pmatrix} t_1 & t_2 & t_3 & \cdots \\ t_2 & t_3 & t_4 & \cdots \\ t_3 & t_4 & t_5 & \cdots \\ \vdots & & \ddots & \vdots \end{pmatrix}, \tag{6}$$

and

$$H_{\tilde{f}} = \begin{pmatrix} t_{-1} & t_{-2} & t_{-3} & \cdots \\ t_{-2} & t_{-3} & t_{-4} & \cdots \\ t_{-3} & t_{-4} & t_{-5} & \cdots \\ \vdots & & \ddots & \vdots \end{pmatrix}. \tag{7}$$

some MATLAB notations are used; for instance, $A(m_1 : m_2, n_1 : n_2)$ denotes the portion of $A$ with rows from $m_1$ to $m_2$ and columns from $n_1$ to $n_2$.

If $A$ is an infinity matrix, we use the notation $A(m_1 : m_2, :)$ as the selection of rows from $m_1$ to $m_2$ of $A$, the same definition can be used for columns of $A$. The vectors will be shown with bold case letters, and we use upper case to show matrices.

This paper is organized as follows. In Section 2, we review preliminary and main results on banded Toeplitz matrices. The algorithm and some computational notes are discussed in Section 3. Numerical tests are given in Section 4 to show the efficiency of our algorithm.

## 2 Main results

In (4), let $w = e^{i\tilde{\theta}}$ and $d = 1$. Then the Toeplitz matrix $T^{(1)}_{\tilde{\theta},n}$ is tridiagonal of the form $T^{(1)}_{\tilde{\theta},n} = \text{tridiag}(t_1, t_0, \bar{t}_1)$, where

$$t_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} 2(1-\cos(\theta-\tilde{\theta}))d\theta = 2,$$

and

$$t_1 = \frac{1}{2\pi}\int_{-\pi}^{\pi} 2(1-\cos(\theta-\tilde{\theta}))e^{i\theta}d\theta = -\bar{w}.$$

If $\tilde{\theta} = 0$, then we have $T^{(1)}_n = \text{tridiag}(-1, 2, -1)$. The eigenvalues of $T^{(1)}_n$ are known:

$$\lambda_k = 2(1-\cos(\frac{k\pi}{n+1})), \quad \text{for} \quad k = 1, 2, \ldots, n.$$

If we define the diagonal matrix $\Omega = \text{diag}(1, w, w^2, \ldots, w^{n-1})$, then $\Omega$ is unitary and we can easily infer that $T^{(1)}_{\tilde{\theta},n} = \Omega^H T^{(1)}_n \Omega$. Hence, two matrices $T^{(1)}_{\tilde{\theta},n}$ and $T^{(1)}_n$ are similar and have the same eigenvalues. The same results will be established for the matrices $T^{(d)}_{\tilde{\theta},n}$ and $T^{(d)}_n$. For this end, we use the following proposition (Proposition 1.3, [2]).

**Proposition 1.** *Let $T_f$ and $T_g$ be the Toeplitz matrices generated by $f(\theta)$ and $g(\theta)$ respectively, then $T_{fg} = T_f T_g + H_f H_{\tilde{g}}$.*

Theorem 1 satisfies for infinite Toeplitz matrices that generated by $f$ and $g$. We use (4) in Theorem 1 and derive the following result.

**Theorem 1.** *Let $T^{(d)}_{\tilde{\theta},n}$ is an $n$ by $n$ banded Toeplitz matrix with bandwidth $2d-1$ and generated by $f(\theta) = (2(1-\cos(\theta-\tilde{\theta})))^d$, then*

$$T^{(d)}_{\tilde{\theta},n} = T^{(d-1)}_{\tilde{\theta},n} T^{(1)}_{\tilde{\theta},n} - w\mathbf{t}^{(d-1)}_{\tilde{\theta}}\mathbf{e}_1^T - \bar{w}J\bar{\mathbf{t}}^{(d-1)}_{\tilde{\theta}}\mathbf{e}_n^T, \tag{8}$$

*where*

$$\mathbf{t}^{(d-1)}_{\tilde{\theta}} = Z^T T^{(d-1)}_{\tilde{\theta},n}(1:n, 1).$$

*Proof.* We partition $T_{\tilde{\theta}}^{(d-1)}$ into

$$T_{\tilde{\theta}}^{(d-1)} = \begin{pmatrix} T_{\tilde{\theta},n}^{(d-1)} & T_{1,2}^{(d-1)} \\ T_{1,2}^{(d-1)H} & T_{\tilde{\theta}}^{(d-1)} \end{pmatrix}, \tag{9}$$

where $T_{1,2}^{(d-1)} = T_{\tilde{\theta}}^{(d-1)}(1:n,:)$. Similarly $T_{\tilde{\theta}}^{(1)}$ can be partitioned into

$$T_{\tilde{\theta}}^{(1)} = \begin{pmatrix} T_{\tilde{\theta},n}^{(1)} & -w\mathbf{e}_n\mathbf{e}_1^T \\ -\bar{w}\mathbf{e}_1\mathbf{e}_n^T & T_{\tilde{\theta}}^{(1)} \end{pmatrix}. \tag{10}$$

In Proposition 1, let $f(\theta) = (2(1-\cos(\theta-\tilde{\theta})))^{d-1}$ and $g(\theta) = 2(1-\cos(\theta-\tilde{\theta}))$, then we have

$$T_{\tilde{\theta}}^{(d)} = T_{\tilde{\theta}}^{(d-1)}T_{\tilde{\theta}}^{(1)} + H_f H_{\tilde{g}}. \tag{11}$$

By using (9) and (10), we infer that the $n\times n$ leading principal submatrix of $T_{\tilde{\theta}}^{(d-1)}T_{\tilde{\theta}}^{(1)}$ can be written as follows

$$(T_{\tilde{\theta}}^{(d-1)}T_{\tilde{\theta}}^{(1)})(1:n,1:n) = T_{\tilde{\theta},n}^{(d-1)}T_{\tilde{\theta},n}^{(1)} - \bar{w}T_{1,2}^{(d-1)}\mathbf{e}_1\mathbf{e}_n^T. \tag{12}$$

The result can be obtained by using the relation $T_{1,2}^{(d-1)}\mathbf{e}_1 = J\bar{\mathbf{t}}_{\tilde{\theta}}^{(d-1)}$, the fact that $H(\tilde{g}) = -w\mathbf{e}_1\mathbf{e}_1^T$, and $(H_f H_{\tilde{g}})(1:n,1:n) = -w\mathbf{t}_{\tilde{\theta}}^{(d-1)}\mathbf{e}_1^T$. □

As a special case, by letting $\tilde{\theta} = 0$, the relation (8) reduces to the form

$$T_n^{(d)} = T_n^{(d-1)}T_n^{(1)} - \mathbf{t}^{(d-1)}\mathbf{e}_1^T - J\mathbf{t}^{(d-1)}\mathbf{e}_n^T, \tag{13}$$

where $\mathbf{t}^{(d-1)} = Z^T T_n^{(d-1)}(1:n,1)$. As a corollary, from Theorem 1 we can prove that $T_{\tilde{\theta}}^{(d)}$ and $T_n^{(d)}$ are similar.

**Remark 1.** $T_{\tilde{\theta},n}^{(d)} = \Omega_n^H T_n^{(d)}\Omega$.

*Proof.* The proof can be done by induction on $d$. For $d=1$ the proof is obvious. By using the definition of $\mathbf{t}_{\tilde{\theta}}^{(d-1)}$ and induction hypothesis for $d-1$, we can simply conclude that

$$\begin{aligned} w\Omega\mathbf{t}_{\tilde{\theta}}^{(d-1)} &= Z^T\Omega T_{\tilde{\theta},n}^{(d-1)}(:,1) \\ &= Z^T\Omega T_{\tilde{\theta},n}^{(d-1)}\Omega\mathbf{e}_1 \\ &= Z^T T_n^{(d-1)}(:,1) = \mathbf{t}^{(d-1)}. \end{aligned}$$

In the similar way it is easily to prove that $\bar{w}^n\Omega J\bar{\mathbf{t}}_{\tilde{\theta}}^{(d-1)} = J\mathbf{t}^{(d-1)}$.

If we multiply the relation of (8) from the left by $\Omega$ and from the right by $\Omega^H$, then we have

$$\begin{aligned} \Omega T_{\tilde{\theta},n}^{(d)}\Omega^H &= \Omega T_{\tilde{\theta},n}^{(d-1)}\Omega^H\Omega T_{\tilde{\theta},n}^{(1)}\Omega^H - w\Omega\mathbf{t}_{\tilde{\theta}}^{(d-1)}\mathbf{e}_1^T - \bar{w}^n\Omega J\bar{\mathbf{t}}_{\tilde{\theta}}^{(d-1)}\mathbf{e}_n^T \\ &= T_n^{(d-1)}T_n^{(1)} - \mathbf{t}^{(d-1)}\mathbf{e}_1^T - J\mathbf{t}^{(d-1)}\mathbf{e}_n^T \\ &= T_n^{(d)}. \end{aligned}$$

Hence,

$$\Omega T^{(d)}_{\tilde{\theta},n}\Omega^H = T^{(d)}_n, \tag{14}$$

which completes the proof. □

As a consequence, we can focus on $T^{(d)}_n$ instead of computing the inverse of $T_{\tilde{\theta},n}$.

## 3 Some Computational Notes

For a given vector $\mathbf{b}$, we aim to solve the linear system $T^{(d)}_n \mathbf{x}^{(d)} = \mathbf{b}$. Letting $\mathbf{a}^{(d-1)} = T_n^{-(d-1)}\mathbf{t}^{(d-1)}$, we can rewrite (8) as

$$T_n^{-(d-1)} T^{(d)}_n = T^{(1)}_n - \mathbf{a}^{(d-1)} e_1^T - J\mathbf{a}^{(d-1)} e_n^T. \tag{15}$$

If we define $A_n = T^{(1)}_n - \mathbf{a}^{(d-1)} e_1^T - J\mathbf{a}^{(d-1)} e_n^T$, then by using (15), $\mathbf{x}^d_n$ satisfies the recursive relation

$$A_n \mathbf{x}^{(d)}_n = \mathbf{x}^{(d-1)}_n. \tag{16}$$

If $\mathbf{x}^{(d-1)}_n$ is known, by solving this linear system then we get $\mathbf{x}^{(d)}_n$. The coefficient matrix in (16) has the following block structure

$$A_n = \begin{pmatrix} -b_1 - 2 & -\mathbf{e}_1^T & -b_n \\ \mathbf{c} & T^{(1)}_{n-2} & J\mathbf{c} \\ -b_{n-1} & -\mathbf{e}_{n-2}^T & -b_1 - 2 \end{pmatrix}, \tag{17}$$

where $\mathbf{c} = -\mathbf{e}_1 - \mathbf{a}(2 : n-2)$. If we define the permutations matrix

$$E_n = \begin{pmatrix} 0_{n-2} & I_{n-2} & 0_{n-2} \\ 1 & 0_{n-1}^T & 0 \\ 0 & 0_{n-1}^T & 1 \end{pmatrix}, \tag{18}$$

and $\tilde{A} = E_n A_n E_n^T$, then $\tilde{A}$ can be partitioned as

$$\tilde{A}_n = \begin{pmatrix} T^{(1)}_{n-2} & \mathscr{R} \\ \mathscr{S} & \mathscr{T} \end{pmatrix}, \tag{19}$$

where $\mathscr{R} = \begin{pmatrix} \mathbf{c} & J\mathbf{c} \end{pmatrix}$, $\mathscr{S} = \begin{pmatrix} -\mathbf{e}_1 & -\mathbf{e}_{n-2} \end{pmatrix}^T$, and

$$\mathscr{T} = \begin{pmatrix} -b_1 - 2 & -b_n \\ -b_n & -b_1 - 2 \end{pmatrix}. \tag{20}$$

By defining $\tilde{\mathbf{x}}^{(d)}_n = E_n \mathbf{x}^{(d)}_n$, the relation (16) can be rewritten as

$$\tilde{A}_n \tilde{\mathbf{x}}^{(d)}_n = \tilde{\mathbf{x}}^{(d-1)}_n \tag{21}$$

In order to solve this equation, we decompose $\tilde{A}_n$ as $\tilde{A}_n = P_n Q_n$, where

$$P_n = \begin{pmatrix} I_{n-2} & \\ \mathscr{S} T^{-(1)}_{n-2} & I_2 \end{pmatrix}, \tag{22}$$

$$Q_n = \begin{pmatrix} T_{n-2}^{(1)} & \mathscr{R} \\ 0 & \mathscr{Z} \end{pmatrix}, \tag{23}$$

with

$$\begin{aligned} \mathscr{Z} &= \mathscr{T} - \mathscr{S} T_{n-2}^{-(1)} \mathscr{R} \\ &= \mathscr{T} + \begin{pmatrix} \mathbf{x}_{n-2}^{(1)} & J\mathbf{x}_{n-2}^{(1)} \end{pmatrix}^T \mathscr{R}. \end{aligned} \tag{24}$$

So the linear system $\tilde{A}_n \tilde{\mathbf{x}}_n^{(d)} = \tilde{\mathbf{x}}_n^{(d-1)}$ can be solved by $P_n \mathbf{u}_n = \tilde{\mathbf{x}}_n^{(d-1)}$ and $Q_n \tilde{\mathbf{x}}_n^{(d)} = u_n$. Hence we, have

$$\mathbf{u}_n(1:n-2) = \tilde{\mathbf{x}}_{n-2}^{(d-1)}, \tag{25}$$

$$\begin{aligned} \mathbf{u}_n(n-1:n) &= \tilde{\mathbf{x}}_n^{(d)}(n-1:n) - \mathscr{S} T_{n-2}^{-(1)} \mathbf{u}_n(1:n-2)) \\ &= \tilde{\mathbf{x}}_n^{(d)}(n-1:n) + \begin{pmatrix} \mathbf{x}_{n-2}^{(1)} & J\mathbf{x}_{n-2}^{(1)} \end{pmatrix}^T \mathbf{u}_n(1:n-2)), \end{aligned} \tag{26}$$

and

$$\tilde{\mathbf{x}}_n^{(d)}(n-1:n) = \mathscr{Z}^{-1} \mathbf{u}_n(n-1:n), \tag{27}$$

$$\tilde{\mathbf{x}}_n^{(d)}(1:n-2) = T_{n-2}^{-(1)} \left( \mathbf{u}_n(1:n-2) - \mathscr{R} \mathbf{u}_n(n-1:n) \right). \tag{28}$$

The main step of the recursive algorithm to compute $\mathbf{x}_n^{(d)}$ may be summarized as follows

**Algorithm 1** Recursive algorithm

```
1:  procedure RECURSIVE(b, d)
2:      if d == 1 then
3:          return x_n^1 = T_n^{-(1)} b.
4:      else
5:          x_n^{(d-1)} = Recursive(b, d-1).
6:          a_n^{(d-1)} = Recursive(t^{(d-1)}, d-1).
7:          Define the permutation matrix E_n as in (18). and set x̃_n^{(d-1)} = E_n x_n^{(d-1)}
8:          Compute u from (25) and (26).
9:          Compute 𝒵 as defined in (24).
10:         x̃_n^{(d)}(n-1:n) = 𝒵^{-1} u_n(n-1:n).
11:         x̃_n^{(d)}(1:n-2) = T_{n-2}^{-(1)} (u_n(1:n-2) - ℛ u_n(n-1:n)).
12:         return x_n^{(d)} = E_n^T x̃_n^{(d)}
13:     end if
14: end procedure
```

We note that the vector $T_n^{-(1)} b$ can be computed by the Gaussian elimination of order $O(n)$. If we define the sequence $\{\alpha_k\}_{k\geq 1}$ as follows

$$\alpha_1 = 2, \ \alpha_k = 2 - \frac{1}{\alpha_{k-1}}, \qquad k = 2, 3, \ldots, \tag{29}$$

then the $LU$-factorization of $T_n^1 = \text{tridiag}\{-1,2,-1\}$ can be computed as follows

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\frac{1}{\alpha_1} & 1 & 0 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & -\frac{1}{\alpha_{n-1}} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} \alpha_1 & -1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & -1 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n \end{pmatrix}. \tag{30}$$

Hence, the computational complexity for solving the linear system $T_n^1 \mathbf{x}^{(1)} = \mathbf{b}$ is $O(n)$, and it uses $O(n)$ storage. We note that $\alpha_k \geq 1$ for $k = 1,2,3,\ldots$. Hence, the Gaussian elimination is stable and fast.

### 3.1 Stability notes

Let $\tilde{\mathbf{x}}_n^{(d-1)*}$ denotes as an approximation of $\tilde{\mathbf{x}}_n^{(d-1)}$, i.e.,

$$\tilde{\mathbf{x}}_n^{(d-1)*} = \tilde{\mathbf{x}}_n^{(d-1)} + \delta_n^{(d)}.$$

As we assume that the other terms of (25) to (28) are exact, then we have

$$\tilde{\mathbf{x}}_n^{(d)*}(1:n-2) = \tilde{\mathbf{x}}_n^{(d-1)*}(1:n-2) + T_{n-2}^{-(1)} \delta_n^{(d-1)}(1:n-2).$$

By simple induction, we can conclude that

$$\begin{aligned} \tilde{\mathbf{x}}_n^{(d)*}(1:n-2) &= \tilde{\mathbf{x}}_n^{(d)}(1:n-2) + \delta_n^{(d)}(1:n-2) \\ &= \tilde{\mathbf{x}}_n^{(d)}(1:n-2) + (T_{n-2}^{-(1)})^{d-1} \delta_n^{(1)}(1:n-2). \end{aligned} \tag{31}$$

It is known that $T_{n-2}$ is symmetric positive definite, and its eigenvalues are $2(1-\cos(\frac{j\pi}{n-1}))$ for $j = 1,\ldots,n-2$. Hence

$$\|T_{n-2}\|_2 = 2(1-\cos(\frac{(n-2)\pi}{n-1})), \quad \text{and} \quad \|T_{n-2}^{-1}\|_2 = \frac{1}{2(1-\cos(\frac{\pi}{n-1}))}.$$

When $n$ is large enough we can conclude that $\|T_{n-2}^{-1}\|_2 \approx \frac{(n-1)^2}{2\pi^2}$. From (31), we see that

$$\begin{aligned} \|\delta_n^{(d)}(1:n-2)\|_2 &\leq \|(T_{n-2}^{-(1)})^{d-1}\|_2 \|\delta_n^{(1)}(1:n-2)\|_2 \\ &\approx (\frac{(n-1)^2}{2\pi^2})^{(d-1)} \|\delta_n^{(1)}(1:n-2)\|_2. \end{aligned}$$

For example, for $n = 1000$ and $\|\delta_n^{(1)}(1:n-2)\|_2 = \varepsilon$, and assuming that all other computations are exactly done or about $\varepsilon$, then we have $\|\delta_n^{(5)}(1:n-2)\|_2 \leq 3.1250 \times 10^8 \varepsilon$. In Section 4, the numerical experiments confirm this fact.

The complexity of Algorithm 1, depends on the complexity of $T_n^{-(1)}\mathbf{a}$. As we see in Algorithm 1 the complexity computation of $T_n^{-(1)}\mathbf{a}$ is $O(n)$. So the complexity of Algorithm 1 is $O(n)$.

Table 1: Comparison the results of the RS algorithm with the GLev Algorithm for $T_n^{(2)}$.

| | **RS** | | **Glev** | |
|---|---|---|---|---|
| $n$ | $\|\mathbf{r}_n\|$ | CPU | $\|\mathbf{r}_n\|$ | CPU |
| $2^9$ | $1.3267 \times 10^{-12}$ | 0.00127 | $4.6825 \times 10^{-12}$ | 0.01011 |
| $2^{10}$ | $1.5947 \times 10^{-11}$ | 0.0018 | $2.0252 \times 10^{-11}$ | 0.0221 |
| $2^{11}$ | $6.5652 \times 10^{-11}$ | 0.0054 | $7.8021 \times 10^{-11}$ | 0.049430 |
| $2^{12}$ | $2.4837 \times 10^{-10}$ | 0.0152 | $3.1374 \times 10^{-10}$ | 0.1452 |
| $2^{13}$ | $1.0242 \times 10^{-9}$ | 0.0325 | $1.2520 \times 10^{-9}$ | 0.7443 |
| $2^{14}$ | $4.0362 \times 10^{-9}$ | 0.1253 | $5.1109 \times 10^{-9}$ | 2.8594 |
| $2^{15}$ | $1.5964 \times 10^{-8}$ | 0.42035 | $1.7007 \times 10^{-8}$ | 11.3476 |
| $2^{16}$ | $6.3628 \times 10^{-8}$ | 1.3627 | $4.2896 \times 10^{-8}$ | 53.8154 |

Table 2: Comparison the results of the RS algorithm with the GLev Algorithm for $T_n^{(3)}$.

| | **RS** | | **Glev** | |
|---|---|---|---|---|
| $n$ | $\|\mathbf{r}_n\|$ | CPU | $\|\mathbf{r}_n\|$ | CPU |
| $2^9$ | $2.8586 \times 10^{-10}$ | 0.0053 | $9.4174 \times 10^{-10}$ | 0.01011 |
| $2^{10}$ | $7.3404 \times 10^{-9}$ | 0.0038 | $6.2314 \times 10^{-9}$ | 0.0221 |
| $2^{11}$ | $5.9609 \times 10^{-8}$ | 0.006566 | $6.8530 \times 10^{-8}$ | 0.049430 |
| $2^{12}$ | $4.9623 \times 10^{-7}$ | 0.01612 | $4.6794 \times 10^{-4}$ | 0.1452 |
| $2^{13}$ | $4.0173 \times 10^{-6}$ | 0.07099 | $5.1728 \times 10^{-4}$ | 0.7443 |
| $2^{14}$ | $3.1982 \times 10^{-5}$ | 0.2238 | $6.2078 \times 10^{-4}$ | 2.8594 |
| $2^{15}$ | $2.5724 \times 10^{-4}$ | 0.7596 | 0.0084 | 11.3476 |
| $2^{16}$ | 0.0021 | 2.8229 | 0.0338 | 53.8154 |

## 4 Numerical Experiments

In this section, we compare our recursive solver (RS) as described in Algorithm 1 with general Levinson algorithm (GLev) [3] for linear systems with coefficient matrices $T_n^{(2)}$, $T_n^{(3)}$, and $T_n^{(4)}$, respectively.We used a random vector scaled to have the unit length for the right-hand side vector $\mathbf{b}$. All tests are performed in MATLAB with double precision. Our comparisons are done for the norm of residual vector $\|\mathbf{r}_n\|$ and the elapsed CPU time (denoted by "CPU"). All the numerical results are performed for $n = 2^k$ for $k = 9, \ldots, 16$. The corresponding numerical results are listed in Tables 1-3.

In the following, we summarize the observation from Tables 1-3. In all cases, in terms of the CPU time needed for solving the linear systems, the RS is faster than the GLev method. CPU time shows that the complexity of the RS algorithm is $O(n)$. Despite the results of Table 1 that indicate the RS and GLev have the same rate of the residual norm, Tables 2 and 3 show that the norm of residual vector for the RS algorithm is much better than GLev algorithm. These results imply that the computational efficiency of the RS method is higher than that of the GLev algorithm.

Table 3: Comparison the results of the RS algorithm with the GLev Algorithm for $T_n^{(4)}$.

| | **RS** | | **Glev** | |
|---|---|---|---|---|
| $n$ | $\|\mathbf{r}_n\|$ | CPU | $\|\mathbf{r}_n\|$ | CPU |
| $2^9$ | $4.8009 \times 10^{-8}$ | 0.0067 | $1.2512 \times 10^{-7}$ | 0.01011 |
| $2^{10}$ | $2.4612 \times 10^{-6}$ | 0.0046 | $5.3260 \times 10^{-5}$ | 0.0221 |
| $2^{11}$ | $4.1866 \times 10^{-5}$ | 0.0110 | $8.6481 \times 10^{-5}$ | 0.049430 |
| $2^{12}$ | $6.538 \times 10^{-4}$ | 0.0265 | 0.0935 | 0.1452 |
| $2^{13}$ | 0.0103 | 0.0928 | 1.5059 | 0.7443 |
| $2^{14}$ | 0.1638 | 0.2572 | $> 10^3$ | 2.8594 |
| $2^{15}$ | 2.6122 | 1.0627 | $> 10^3$ | 11.3476 |
| $2^{16}$ | 41.3931 | 4.0536 | $> 10^3$ | 53.8154 |

## References

[1] M. Barrera, A. Bttcher, S.M. Grudsky, E.A. Maximenko, *Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic*, In: Bttcher, A., Potts, D., Stollmann, P., Wenzel, D. (eds) The Diversity and Beauty of Applied Operator Theory. Operator Theory: Advances and Applications, vol 268. Birkhuser, Cham, https://doi.org/10.1007/978-3-319-75996-8_2

[2] A. Bttcher, S.M. Grudsky. *Spectral properties of banded Toeplitz matrices*, SIAM, 2005.

[3] P. Favati, L. Grazia, M. Ornella, *Stability of the Levinson algorithm for Toeplitz-like systems*, SIAM J. Matrix Anal. Appl. **31** (2010) 2531–2552.

[4] I. Gohberg, S. Arkadii, *On the inversion of finite Toeplitz matrices and their continuous analogs*, Mat. Issued **2** (1972) 201–233.

[5] S. Hon, S. Serra-Capizzano, W. Andy, *Band-Toeplitz preconditioners for ill-conditioned Toeplitz systems*, BIT Numer. Math. (2021) 1–27.