



A matrix-less method to approximate the spectrum and the spectral function of Toeplitz matrices with real eigenvalues

Sven-Erik Ekström¹ · Paris Vassalos²

Received: 10 January 2021 / Accepted: 2 May 2021 / Published online: 24 June 2021
© The Author(s) 2021

Abstract

It is known that the generating function f of a sequence of Toeplitz matrices $\{T_n(f)\}_n$ may not describe the asymptotic distribution of the eigenvalues of $T_n(f)$ if f is not real. In this paper, we assume as a working hypothesis that, if the eigenvalues of $T_n(f)$ are real for all n , then they admit an asymptotic expansion of the same type as considered in previous works, where the first function, called the eigenvalue symbol \mathfrak{f} , appearing in this expansion is real and describes the asymptotic distribution of the eigenvalues of $T_n(f)$. This eigenvalue symbol \mathfrak{f} is in general not known in closed form. After validating this working hypothesis through a number of numerical experiments, we propose a matrix-less algorithm in order to approximate the eigenvalue distribution function \mathfrak{f} . The proposed algorithm, which opposed to previous versions, does not need any information about neither f nor \mathfrak{f} is tested on a wide range of numerical examples; in some cases, we are even able to find the analytical expression of \mathfrak{f} . Future research directions are outlined at the end of the paper.

Keywords Spectral analysis · Structured matrices · Asymptotic expansion · Toeplitz matrices

✉ Sven-Erik Ekström
sven-erik.ekstrom@it.uu.se

Paris Vassalos
pvassal@aub.gr

- ¹ Division of Scientific Computing, Department of Information Technology, Uppsala University, Uppsala, Sweden
- ² School of Information Sciences & Technology, Department of Informatics, Athens University of Economics and Business, Athens, Greece

1 Introduction

Given a function $f \in L^1([-\pi, \pi])$, the $n \times n$ Toeplitz matrix generated by f is defined as

$$T_n(f) = \left[\hat{f}_{i-j} \right]_{i,j=1}^n = \begin{bmatrix} \hat{f}_0 & \hat{f}_{-1} & \cdots & \hat{f}_{1-n} \\ \hat{f}_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{f}_{-1} \\ \hat{f}_{n-1} & \cdots & \hat{f}_1 & \hat{f}_0 \end{bmatrix},$$

where the numbers \hat{f}_k are the Fourier coefficients of f , that is,

$$\hat{f}_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \quad f(\theta) = \sum_{k=-\infty}^{\infty} \hat{f}_k e^{ik\theta}, \quad i^2 = -1, \quad k \in \mathbb{Z}. \tag{1}$$

It is known that the generating function f , also known as the symbol of the sequence $\{T_n(f)\}_n$, describes the asymptotic distribution of the singular values of $T_n(f)$; if f is real or if $f \in L^\infty([-\pi, \pi])$ and its essential range has empty interior and does not disconnect the complex plane, then f also describes the asymptotic distribution of the eigenvalues of f ; see [12, 20, 25] for details and [20, Section 3.1] for the notion of asymptotic singular value and eigenvalue distribution of a sequence of matrices. We will write $\{T_n(f)\}_n \sim_\lambda f$ to indicate that $\{T_n(f)\}_n$ has an asymptotic eigenvalue distribution described by f . The precise definition is the following [20, Section 3.1].

Definition 1 (eigenvalue distribution of a matrix sequence) Let $\{A_n\}_n$ be a matrix sequence, with A_n of size n , and let $f : D \subset \mathbb{R}^k \rightarrow \mathbb{C}$ be a measurable function defined on a set D with $0 < \mu_k(D) < \infty$. We say that it has an asymptotic eigenvalue (or spectral) distribution described by f and we write $\{A_n\}_n \sim_\lambda f$, if

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{j=1}^n F(\lambda_j(A_n)) = \frac{1}{\mu_k(D)} \int_D F(f(x_1, \dots, x_k)) dx_1 \cdots dx_k, \quad \forall F \in C_c(\mathbb{C}).$$

In this case, the function f is referred to as the eigenvalue (or spectral) symbol of the matrix-sequence $\{A_n\}_n$.

Informally, this means that if we have a continuous a.e. and real f , an equispaced grid $\theta_{j,n}$ in its domain, and the eigenvalues suitably ordered, then, $\lambda_j(A_n) \approx f(\theta_{j,n})$ for all $j = 1, \dots, n$, when n is large enough.

The cases of interest in this paper are those in which $A_n = T_n(f)$ and $\{T_n(f)\}_n \not\sim_\lambda f$ (except for Examples 2 and 6, where we show that the same approach works for $\{T_n(f)\}_n \sim_\lambda f$) where f is complex-valued (and $T_n(f)$ is non-Hermitian), but the eigenvalues of $T_n(f)$ are real for all n . We believe that in these cases there exist a real function f such that $\{T_n(f)\}_n \sim_\lambda f$ and the eigenvalues of $T_n(f)$ admit an asymptotic

expansion of the same type as considered in previous works [2–4, 7, 8, 15–18]. We therefore formulate the following working hypothesis.

Working Hypothesis Suppose that the eigenvalues of $T_n(f)$ are real for all n . Then, for every integer $\alpha \geq 0$, every n and every $j = 1, \dots, n$, the following asymptotic expansion holds:

$$\begin{aligned} \lambda_j(T_n(f)) &= f(\theta_{j,n}) + \sum_{k=1}^{\alpha} c_k(\theta_{j,n})h^k + E_{j,n,\alpha}, \\ &= \sum_{k=0}^{\alpha} c_k(\theta_{j,n})h^k + E_{j,n,\alpha}, \end{aligned} \tag{2}$$

where:

- the eigenvalues of $T_n(f)$ are arranged in non-decreasing order, $\lambda_1(T_n(f)) \leq \dots \leq \lambda_n(T_n(f))$;
- $\{c_0, c_1, c_2, c_3, \dots\}$ is a sequence of functions from $(0, \pi)$ to \mathbb{R} which depends only on f , and $f := c_0$;
- $h = \frac{1}{n+1}$ and $\theta_{j,n} = \frac{j\pi}{n+1} = j\pi h$;
- $E_{j,n,\alpha} = O(h^{\alpha+1})$ is the remainder (the error), which satisfies the inequality $|E_{j,n,\alpha}| \leq C_\alpha h^{\alpha+1}$ for some constant C_α depending only on α, f .

Remark 1 In item three of the working hypothesis, the grid $\theta_{j,n}$ is defined, but for some matrices, it is advantageous to define a different equispaced grid, e.g., by choosing $h = 1/n$; see, e.g., [15] where the grid $\theta_{j,n} = j\pi/n$ is almost the “perfect grid” $\xi_{j,n}$ (defined, such that, $\lambda_j(T_n(f)) = f(\xi_{j,n})$ for $j = 1, \dots, n$). The typically not equispaced “perfect grid” $\xi_{j,n}$ is further discussed in [14]. When choosing a different grid $\theta_{j,n}$ in item three, the functions c_1, c_2, c_3, \dots in item two will differ, but $c_0 = f$ remains the same; when choosing the (typically unknown) grid $\xi_{j,n}$, then $c_k = 0$ for $k > 0$. The functions c_k can also be defined on any interval, not just $(0, \pi)$, but then h and $\theta_{j,n}$ in item three should be modified accordingly. Also, in the working hypothesis, we arrange the eigenvalues of $T_n(f)$ in non-decreasing order; however, using a non-increasing order would result in another function f . The case where the eigenvalues of $T_n(f)$ can be described by a complex-valued or non-monotone function f is out of the scope of this article and warrants further research.

2 Motivation and illustrative examples

In this section, we present four examples in support of our working hypothesis. We also discuss the fact that standard double precision eigenvalue solvers (such as LAPACK, eig in MATLAB, and eigvals in JULIA [6]) fail to give accurate eigenvalues of certain non-Hermitian matrices $T_n(f)$ (both with real and complex

eigenvalues); see, e.g., [5, Section 7], [12, Sections 3 and 5.8], [22], and [26, Chapter II]. A simple illustrative example is the Toeplitz matrix $T_n(f)$ generated by the symbol $f(\theta) = e^{i\theta} + e^{-2i\theta}$ presented in Fig. 1; see, e.g., [5, Section 5.1]. The numerically computed eigenvalues $\Psi_j(T_n(f))$ (beige) and $\Psi_j(T_n^T(f))$ (blue), for $n = 999$ and using MATLAB (double precision), differ and neither is correct. A high-precision computation (256 bit) [1] of the eigenvalues $\lambda_j(T_n(f))$ (green) agrees well with the theoretical prediction; see, e.g., [23, Section 7] and [10, Example 11.18]. Along the first arm of the “star,” the eigenvalues are real, and the other arms are simply rotations of these real eigenvalues.

High-precision computations, by using packages such as `GENERICLINEARALGEBRA.JL` [1] in `JULIA` can compute the true eigenvalues, but they are very expensive from the computational point of view. Therefore, approximating f on the grid $\theta_{j,n}$ and using matrix-less methods, described in Section 3, to compute the spectrum of $T_n(f)$ can be computationally very advantageous. Also, the presented approaches can be a valuable tool for the analysis of the spectra of non-normal Toeplitz matrices having real eigenvalues.

Here is a short description of the four examples we are going to consider, where $\xi_{j,n}$ is the “perfect grid” defined in Remark 1.

- Example 1: $T_n(f)$ is non-symmetric tridiagonal, f is known, and the eigenvalues $\lambda_j(T_n(f)) = f(\theta_{j,n})$ are known explicitly;
- Example 2: $T_n(f)$ is symmetric pentadiagonal, $f = f$, and the eigenvalues $\lambda_j(T_n(f)) = f(\xi_{j,n})$ are not known explicitly;
- Example 3: $T_n(f)$ is non-symmetric, f is known, and the eigenvalues $\lambda_j(T_n(f)) = f(\xi_{j,n})$ are not known explicitly;
- Example 4: $T_n(f)$ is non-symmetric, f is not known, and the eigenvalues $\lambda_j(T_n(f)) = f(\xi_{j,n})$ are not known explicitly.

Example 1 Consider the symbol

$$f(\theta) = \hat{f}_{-1}e^{-i\theta} + \hat{f}_0 + \hat{f}_1e^{i\theta}. \tag{3}$$

The matrix $T_n(f)$ is tridiagonal, and there exists a function

$$\mathfrak{f}(\theta) = \hat{f}_0 + 2\sqrt{\hat{f}_{-1}\hat{f}_1}\sqrt{\cos(\theta)}, \tag{4}$$

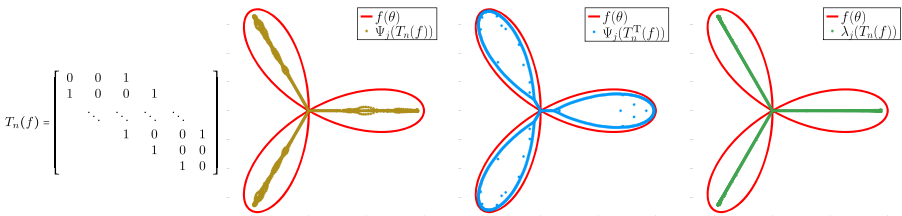


Fig. 1 [Symbol $f(\theta) = e^{i\theta} + e^{-2i\theta}$, $n = 999$] The matrix $T_n(f)$ and the numerically (incorrect) computed eigenvalues $\Psi_j(T_n(f))$ (beige) and $\Psi_j(T_n^T(f))$ (blue), using a double precision solver (MATLAB). To the right is shown the eigenvalues $\lambda_j(T_n(f))$ (green), computed using a high-precision solver [1], agreeing well with the theory

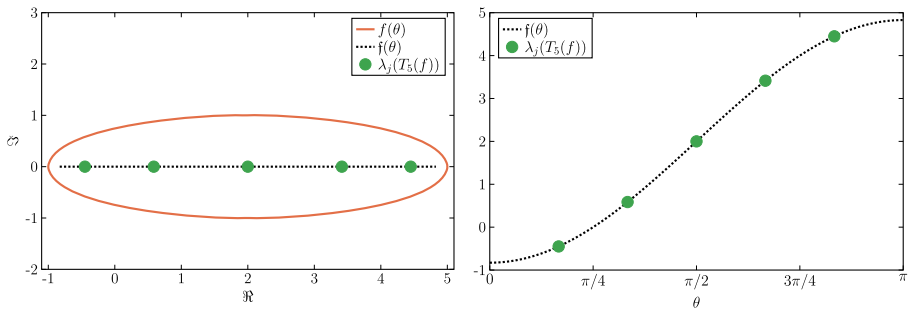


Fig. 2 [Example 1: Symbol $f(\theta) = -2e^{-i\theta} + 2 - e^{i\theta}$] Left: Plots in the complex plane of $f(\theta)$ (red line), $\hat{f}(\theta) = 2 - 2\sqrt{2}\cos(\theta)$ (dashed black line), and $\lambda_j(T_n(f)) = \lambda_j(T_n(\hat{f}))$ for $n = 5$ (green dots). Right: Plots of \hat{f} and $\lambda_j(T_5(f)) = \lambda_j(T_5(\hat{f})) = \hat{f}(\theta_{j,5})$

such that $T_n(f) \sim T_n(\hat{f})$, that is, they are similar and hence have the same eigenvalues. The eigenvalues are given explicitly by

$$\lambda_j(T_n(f)) = \hat{f}(\theta_{j,n}), \tag{5}$$

where $\theta_{j,n}$ is defined in the working hypothesis. Now, choose the Fourier coefficients $\hat{f}_{-1} = -2$, $\hat{f}_0 = 2$, and $\hat{f}_1 = -1$. In this case, we have

$$f(\theta) = -2e^{-i\theta} + 2 - e^{i\theta},$$

and thus,

$$\hat{f}(\theta) = 2 - 2\sqrt{2}\cos(\theta), \tag{6}$$

where the spectrum of $T_n(f)$ is real and is described by $\hat{f}(\theta)$, and the symbol f is complex-valued. The Toeplitz matrices generated by f and \hat{f} are given by

$$T_n(f) = \begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -2 \\ & & & -1 & 2 \end{bmatrix}, \quad T_n(\hat{f}) = \begin{bmatrix} 2 & -\sqrt{2} & & & \\ -\sqrt{2} & 2 & -\sqrt{2} & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -\sqrt{2} \\ & & & -\sqrt{2} & 2 \end{bmatrix}.$$

We also note that $T_n(\hat{f})$ is a symmetrized version of $T_n(f)$, in the sense that there exists a transformation $T_n(\hat{f}) = DT_n(f)D^{-1}$ where D is a diagonal matrix with elements $(D)_{i,i} = \gamma^{i-1}$, and $\gamma = \sqrt{\hat{f}_{-1}}/\sqrt{\hat{f}_1}$; see, e.g., [21].

In the left panel of Fig. 2, we represent the function f (red line), \hat{f} (dashed black line), and the eigenvalues $\lambda_j(T_n(f)) = \lambda_j(T_n(\hat{f}))$ (green dots) for $n = 5$. In the right panel of Fig. 2, we show the function \hat{f} (dashed black line) on the interval $[0, \pi]$ only (since it is even on $[-\pi, \pi]$) and the eigenvalues $\lambda_j(T_5(f)) = \lambda_j(T_5(\hat{f})) = \hat{f}(\theta_{j,5})$ (green dots).

In Fig. 3, we present the symbol $f(\theta)$ (red line), the numerically computed spectra $\Psi_j(T_n(f))$ (beige dots), and $\Psi_j(T_n^T(f))$ (blue dots), for $n = 1000$, using a standard double precision eigenvalue solver. The analytical spectrum, defined by (5) and (6),

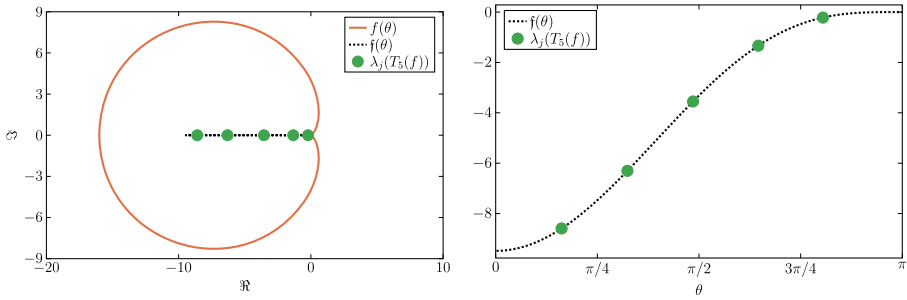


Fig. 5 [Example 3: Symbol $f(\theta) = e^{-i\theta} (6 - 8 \cos(\theta) + 2 \cos(2\theta))$] Left: Plots in the complex plane of $f(\theta)$ (red line), $\hat{f}(\theta) = -\sin^4(\theta)/(\sin(\theta/4) \sin^3(3\theta/4))$ (dashed black line), and $\lambda_j(T_n(f))$ for $n = 5$ (green dots). Right: Plot of \hat{f} and $\lambda_j(T_5(f)) = \hat{f}(\xi_{j,5})$

which is equivalent to (41) in [24, Example 3.] with $z = e^{i\theta}$, $a = -1$, $r = 3$, and $s = 1$. Hence, by (43) in [24], we have,

$$f(\theta) = -\frac{\sin^4(\theta)}{\sin(\theta/4) \sin^3(3\theta/4)}, \tag{7}$$

and the matrix $T_n(f)$ would be full with $\lambda_j(T_n(f)) \approx \lambda_j(T_n(f)) \in (-\frac{(r+s)^{r+s}}{r^r s^s}, 0) = (-\frac{256}{27}, 0)$ for all j .

In the left panel of Fig. 5, we represent the functions f (red line), \hat{f} (dashed black line) and the eigenvalues $\lambda_j(T_n(f))$ (green dots) for $n = 5$. In the right panel of Fig. 5, we show the function \hat{f} (dashed black line) on the interval $[0, \pi]$ only (since it is even on $[-\pi, \pi]$) and the eigenvalues $\lambda_j(T_5(f)) = \hat{f}(\xi_{j,5})$ (green dots).

In Fig. 6, we present the symbol f (red line), the numerically computed spectra $\Psi_j(T_n(f))$ (beige dots) and $\Psi_j(T_n^T(f))$ (blue dots), for $n = 1000$, using a

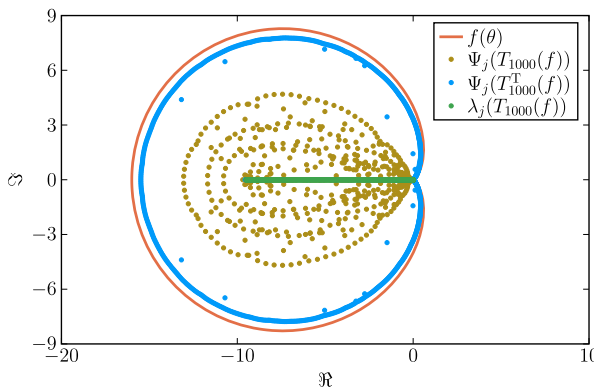


Fig. 6 [Example 3: Symbol $f(\theta) = e^{-i\theta} (6 - 8 \cos(\theta) + 2 \cos(2\theta))$] Symbol $f(\theta)$ (red line), the numerically computed spectra (using a standard double precision eigenvalue solver in JULIA) $\Psi_j(T_{1000}(f))$ (beige dots), $\Psi_j(T_{1000}^T(f))$ (blue dots), and approximated spectrum $\lambda_j(T_{1000}(f)) = \hat{f}(\xi_{j,1000})$ (green dots)

In the left panel of Fig. 7, we represent the symbol f (red line) and the eigenvalues $\lambda_{1000}(T_n(f))$, computed with 256 bit precision `BigFloat` (black dots) since f is not known. The eigenvalues $\lambda_j(T_n(f))$ (green dots) for $n = 5$ are also shown. In the right panel of Fig. 7, we again show the eigenvalues $\lambda_{1000}(T_n(f))$ arranged in non-decreasing order (black dots) since f is not known on the interval $[0, \pi]$, and it is even on $[-\pi, \pi]$. Also the eigenvalues $\lambda_j(T_5(f)) = f(\xi_{j,5})$ (green dots) are shown. The “perfect” grid $\xi_{j,n}$ is computed using data from Example 8. Numerically, we have $\lambda_j(T_{1000}(f)) \in [-22.0912, 14.9641]$ in agreement with [24]. In Fig. 8, we present the numerically computed spectra $\Psi_j(T_n(f))$ (beige dots) and $\Psi_j(T_n^T(f))$ (blue dots), for $n = 1000$, using a standard double precision eigenvalue solver. The true eigenvalues $\lambda_j(T_{1000}(f)) = f(\xi_{j,1000})$ (green dots) are approximated using a 256 bit precision computation.

3 Describing the real-valued eigenvalue distribution

Assuming that f is a real cosine trigonometric polynomial (RCTP) symbol associated with a symbol f as in the working hypothesis, we introduce in Section 3.1 a new *matrix-less* method to accurately compute the expansion functions $c_k, k = 0, \dots, \alpha$, where we recall from the working hypothesis that $c_0 = f$. Subsequently, in Section 3.2, we present procedures to obtain an approximation or even the analytical expression of f .

3.1 Approximating the expansion functions c_k in grid points θ_{j,n_0}

An asymptotic expansion of the eigenvalue errors $E_{j,n,0} := E_{j,n}$ (when sampling the symbol f with the grid $\theta_{j,n}$ defined in the working hypothesis, under certain

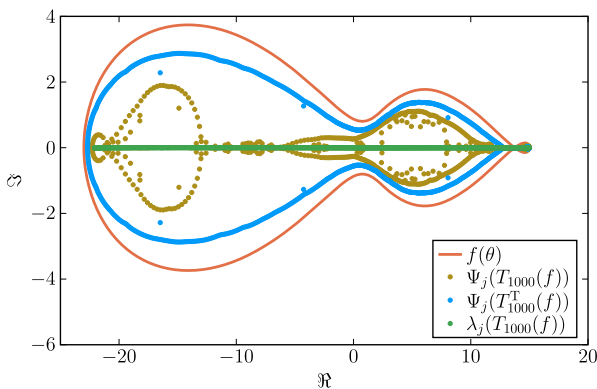


Fig. 8 [Example 4: Symbol $f(\theta) = -e^{-4i\theta} + 2e^{-3i\theta} - 2e^{-2i\theta} + 9e^{-i\theta} + 7e^{i\theta} - e^{2i\theta} + e^{3i\theta}$] Symbol $f(\theta)$ (red line), the numerically computed spectra $\Psi_j(T_{1000}(f))$ (beige dots), $\Psi_j(T_{1000}^T(f))$ (blue dots), and $\lambda_j(T_{1000}(f)) = f(\xi_{j,n})$ (green dots)

assumptions on f implying that $\mathfrak{f} = f$) was previously discussed in a series of papers [7, 8, 11]; such expansions can be deduced from

$$\lambda_j(T_n(f)) = f(\theta_{j,n}) + \underbrace{\sum_{k=1}^{\alpha} c_k(\theta_{j,n})h^k}_{E_{j,n,\alpha}} + E_{j,n,\alpha}, \tag{8}$$

where $\theta_{j,n}$, h , and $E_{j,n,\alpha}$ are defined in the working hypothesis.

In [7, pp. 1329], [8], and then [18], an algorithm was proposed to approximate the functions $c_k(\theta)$, which was subsequently extended and studied, see [2–4, 15–17], to cover also other types of Toeplitz-like matrices A_n , possessing an asymptotic expansion such as (8). We call this type of methods *matrix-less*, since they do not need to construct the large matrix A_n to approximate its eigenvalues; indeed, they approximate the functions $c_k(\theta)$ from α small matrices $A_{n_1}, \dots, A_{n_\alpha}$ and then they use this approximations to compute the approximate spectrum of A_n through the formula

$$\lambda_j(A_n) \approx \tilde{\lambda}_j(A_n) = f(\theta_{j,n}) + \sum_{k=1}^{\alpha} \tilde{c}_k(\theta_{j,n})h^k. \tag{9}$$

Assuming that the eigenvalues of $T_n(f)$ admit an asymptotic expansion in terms of a possibly unknown function \mathfrak{f} instead of f , as in our working hypothesis, we can modify Algorithm 1 in [13, Section 2.1] in order to find approximations of both \mathfrak{f} and the eigenvalues of $T_n(f)$ through the following formula, analogous to (9):

$$\begin{aligned} \lambda_j(T_n(f)) \approx \tilde{\lambda}_j(T_n(f)) &= \sum_{k=0}^{\alpha} \tilde{c}_k(\theta_{j,n})h^k \\ &= \tilde{\mathfrak{f}}(\theta_{j,n}) + \sum_{k=1}^{\alpha} \tilde{c}_k(\theta_{j,n})h^k, \end{aligned} \tag{10}$$

where the approximation $\tilde{\mathfrak{f}}(\theta) := \tilde{c}_0(\theta)$ of $\mathfrak{f}(\theta) := c_0(\theta)$ is obtained from $\alpha + 1$ small matrices $T_{n_0}(f), \dots, T_{n_\alpha}(f)$ as mentioned above.

Here follows an implementation in `JULIA` of the algorithm that computes the approximations $\tilde{c}_k(\theta)$ for $k = 0, \dots, \alpha$; the algorithm is written for clarity and not performance. All computations in this article are made with `JULIA` 1.6.0 [6], using `Float64` and `BigFloat` data types, and the `GENERICLINEARALGEBRA.JL` package [1].

Algorithm 1 Approximate expansion functions $c_k(\theta)$ for $k = 0, \dots, \alpha$ on the grid θ_{j,n_0} .

```

using LinearAlgebra, GenericLinearAlgebra
setprecision(BigFloat,128)

# Example: compute_C(100, 4, BigFloat[2, -1], BigFloat[2, -2])
function compute_C(n0      :: Integer,
                  # Number of grid points in grid theta_{j,n0}
                  alpha    :: Integer,
                  # Number of c_k to approximate, k=0,...,alpha
                  vc       :: Array{T,1}, # First column of T_n(f)
                  vr       :: Array{T,1}, # First row of T_n(f)
                  revorder :: Bool=false
                  # Reverse ordering of eigenvalues of T_n(f)
                  ) where T

    j0 = 1:n0
    E = zeros(T,alpha+1,n0)
    hs = zeros(T,alpha+1)
    for kk = 1:alpha+1
        nk = 2^(kk-1)*(n0+1)-1
        jk = 2^(kk-1)*j0
        hs[kk] = convert(T,1)/(nk+1)
        Tnk = toeplitz(nk,vc,vr)
        eTnk = eigvals(Tnk)
        if !isreal(eTnk)
            error("Spectrum not real. Decrease n0 or alpha, or use BigFloat
                with higher precision.")
        end
        eTnk = sort(real.(eTnk),rev=revorder)
        E[kk,:] = eTnk[jk]
    end
    V = zeros(T,alpha+1,alpha+1)
    for ii = 1:alpha+1, jj = 1:alpha+1
        V[ii,jj] = hs[ii]^(jj-1)
    end
    return C=V\E
end

# Example: toeplitz(100, Float64[2, -1], Float64[2, -2])
function toeplitz(n :: Integer, # Order of Toeplitz matrix T_n(f)
                 vc :: Array{T,1}, # First column of T_n(f)
                 vr :: Array{T,1} # First row of T_n(f)
                 ) where T
    Tn = zeros(T,n,n)
    for ii = 1:length(vc)
        Tn = Tn + diagm(-ii+1 => vc[ii]*ones(T,n-ii+1))
    end
    for jj = 2:length(vr)
        Tn = Tn + diagm( jj-1 => vr[jj]*ones(T,n-jj+1))
    end
    return Tn
end

```

Using the output $\tilde{c}_k(\theta_{j,n_0})$, we can employ the interpolation–extrapolation technique described in [17] to efficiently compute very accurate approximations of $\tilde{c}_k(\theta)$ and, through (10), the eigenvalues of $T_n(f)$ for an arbitrarily large order n . In the next section, we focus on the use of the approximations $\tilde{c}_0 = \mathfrak{f}$ to describe \mathfrak{f} .

3.2 Constructing a function f from approximations $\tilde{f}(\theta_{j,n_0}) = \tilde{c}_0(\theta_{j,n_0})$

We here assume, for the sake of simplicity, that the sought function f is real and even, so that it admits a cosine Fourier series of the form

$$f(\theta) = \hat{f}_0 + 2 \sum_{k=1}^{\infty} \hat{f}_k \cos(k\theta), \quad \hat{f}_k \in \mathbb{R}. \tag{11}$$

As we shall see, if f is a real cosine trigonometric polynomial (RCTP), that is, a function of the form

$$f(\theta) = \hat{f}_0 + 2 \sum_{k=1}^m \hat{f}_k \cos(k\theta), \quad \hat{f}_k \in \mathbb{R}, \tag{12}$$

then we will be able to recover the exact expression of f (see Examples 5 and 6); otherwise, we will get a truncated representation of the Fourier expansion of f in (11) (see Examples 7 and 8). More specifically, what we do is the following: we consider the approximations $\tilde{c}_0(\theta_{j,n_0})$ provided by Algorithm 1 and we approximate the first n_0 Fourier coefficients $\hat{f}_0, \dots, \hat{f}_{n_0}$ with the numbers $\tilde{f}_0, \dots, \tilde{f}_{n_0}$ obtained by solving the linear system

$$\tilde{f}_0 + 2 \sum_{k=1}^{n_0} \tilde{f}_k \cos(k\theta_{j,n_0}) = \tilde{c}_0(\theta_{j,n_0}), \quad j = 1, \dots, n_0. \tag{13}$$

Algorithm 2 Compute approximations \tilde{f}_k of the Fourier coefficients \hat{f}_k of $f(\theta)$.

```
# Example: compute_fourier_coefficients(C[1,:])
function compute_fourier_coefficients(c0 :: Array{T,1}) where T
    n0 = length(c0)
    t = LinRange(convert(T,pi)/(n0+1), n0*convert(T,pi)/(n0+1), n0)
    F = zeros(T,n0,n0)
    F[:,1] = ones(T,n0)
    for jj = 2:n0
        F[:,jj] = 2*cos.((jj-1)*t)
    end
    return fourier_coefficients = F \ c0
end
```

4 Numerical examples

We now employ the proposed Algorithms 1 and 2 on matrices generated by the symbols f discussed in Examples 1–4 to highlight the applicability of the approach, in the respective Examples 5–8.

- Example 5: Only \tilde{c}_0 is non-zero, since $\theta_{j,n}$ gives exact eigenvalues, and the function f is constructed.

- Example 6: Symbol $f = c_0$, and $c_k, k = 1, \dots, 4$, are recovered accurately, and the function $\tilde{f} = f$ is constructed.
- Example 7: Symbol $\tilde{f} = c_0$, and $c_k, k = 1, \dots, 4$, are recovered accurately, and a truncated RCTP representation of \tilde{f} is constructed.
- Example 8: Symbol $\tilde{f} = c_0$, and $c_k, k = 1, \dots, 4$, are constructed, and a truncated RCTP representation of \tilde{f} is constructed.

Example 5 We return to the non-symmetric symbol $f(\theta) = -2e^{-i\theta} + 2 - e^{i\theta}$ of Example 1, and first use the proposed Algorithm 1. Note that care has to be taken when using for example standard MATLAB `eig` command, since already for $n = 160$ the computed eigenvalues are wrong and complex-valued. In such a circumstance, a choice of n_0 and α needs to be such that $n_\alpha = 2^\alpha(n_0 + 1) - 1 < 160$. However, we here also use an arbitrary precision solver, `GENERICLINEARALGEBRA.JL` in `JULIA`, so we can increase precision such that theoretically any combination of n_0 and α can be chosen. The performance however decreases fast as we increase the computational precision, making obvious the applicability of the current proposed algorithms.

In Fig. 9, we present the computation of $\tilde{c}_k(\theta_{j,n_0})$, for $n_0 = 31$, and different precision and α . In the left panel, we show the approximated expansion functions \tilde{c}_k , $k = 0, \dots, \alpha$, where $\alpha = 4$, and using 128 bit precision computation. As is seen, the only non-zero \tilde{c}_k is \tilde{c}_0 , which is expected since the exact eigenvalues are given by $\tilde{f}(\theta_{j,n_0}) = c_0(\theta_{j,n})$. In the right panel of Fig. 9, we show the absolute error in the approximation of $c_0(\theta_{j,n_0})$ for double precision computation, with $\alpha = 2$, and 128 bit computation, with $\alpha = 4$.

Now we employ Algorithm 2 to compute the Fourier coefficients of \tilde{f} . For illustrative purposes, we only show a subset of the system (13). If we choose $n_0 = 2^\beta - 1$, we have three grid points θ_{j_β, n_0} equal to $\pi/4, \pi/2$ and $3\pi/4$, corresponding to indices $j_\beta = 2^{\beta-2}\{1, 2, 3\}$. We then have

$$\begin{bmatrix} 1 & \sqrt{2} & 0 \\ 1 & 0 & -1 \\ 1 & -\sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \end{bmatrix} \approx \begin{bmatrix} \tilde{c}_0(\pi/4) \\ \tilde{c}_0(\pi/2) \\ \tilde{c}_0(3\pi/4) \end{bmatrix}.$$

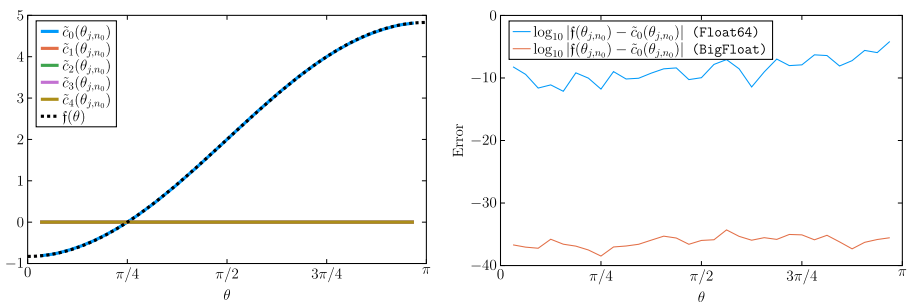


Fig. 9 [Example 5: Symbol $f(\theta) = -2e^{-i\theta} + 2 - e^{i\theta}$] The computed $\tilde{c}_k(\theta_{j,n_0}), k = 0, \dots, \alpha, n_0 = 31$ using Algorithm 1. Left: Computation using 128 bit precision with $\alpha = 4$. Only $\tilde{c}_0(\theta_{j,n_0}) = \tilde{f}(\theta_{j,n_0})$ are non-zero. Right: The absolute errors of $\tilde{c}_0(\theta_{j,n_0})$ compared with $\tilde{f}(\theta_{j,n})$ for double precision (53 bits `Float64`) computation, with $\alpha = 2$, and 128 bit precision `BigFloat`, with $\alpha = 4$

We construct the following system with \tilde{c}_0 computed with $n_0 = 31$ (that is, $\beta = 5$ and $\alpha = 2$ above) using double precision, and subsequently we compute $[\tilde{f}_0, \tilde{f}_1, \tilde{f}_2]^T$,

$$\begin{bmatrix} 1 & \sqrt{2} & 0 \\ 1 & 0 & -1 \\ 1 & -\sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \end{bmatrix} \approx \begin{bmatrix} 0.00000000006924 \\ 1.999999999889880 \\ 3.999999989588136 \end{bmatrix}, \quad \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \end{bmatrix} \approx \begin{bmatrix} 1.999999994797530 \\ -1.414213558689498 \\ -0.00000005092350 \end{bmatrix}.$$

We conclude from this computation that $f(\theta) = \hat{f}_0 + 2\hat{f}_1 \cos(\theta) + 2\hat{f}_2 \cos(2\theta) = 2 - 2\sqrt{2} \cos(\theta)$, which is the already known analytical expression; see (5) and (6). Note also that, in this simple example, the vector containing $\tilde{c}_0(\theta_{j\beta, n_0})$ can be assumed to be equal to $[0, 2, 4]^T$, which would yield the exact solution (to machine precision). Using the full system (13) in Algorithm 2 yields the same result. If we now would approximate the monotonically non-increasing f (instead of the non-decreasing) in Algorithm 1, the vector containing $\tilde{c}(\theta_{j\beta, n_0})$ would be $[4, 2, 0]^T$ and would yield the symbol $f(\theta) = 2 + 2\sqrt{2} \cos(\theta)$. Obviously, the eigenvalues of $T_n(f)$ are the same for both versions of f .

Example 6 We here return to the Hermitian symbol $f(\theta) = (2 - 2 \cos(\theta))^2 = 6 - 8 \cos(\theta) + 2 \cos(2\theta)$, as in Example 2. Since we know $\{T_n(f)\}_n \sim_\lambda f = f$, employing Algorithm 1 will return as \tilde{c}_0 an approximation of f , and as $\tilde{c}_k, k > 0$ the expansion functions previously obtained and studied in [3, 13].

In Fig. 10, we show in the left panel the approximated expansion functions $\tilde{c}_k(\theta_{j, n_0})$ for $k = 0, \dots, \alpha$, computed using $n_0 = 100, \alpha = 4$. Computations are made with double precision. In the right panel of Fig. 10 we show the absolute error of the approximation of f , that is, $\log_{10} |f(\theta_{j, n_0}) - \tilde{c}_0(\theta_{j, n_0})|$, for different combinations of n_0 and α , both with high precision (128 bit `BigFloat`) and standard double precision (`Float64`). Increasing the precision does not reduce the error, unless the error is close to machine epsilon (black line for `Float64`). We note that increasing α has larger impact than increasing n_0 to reduce the errors.

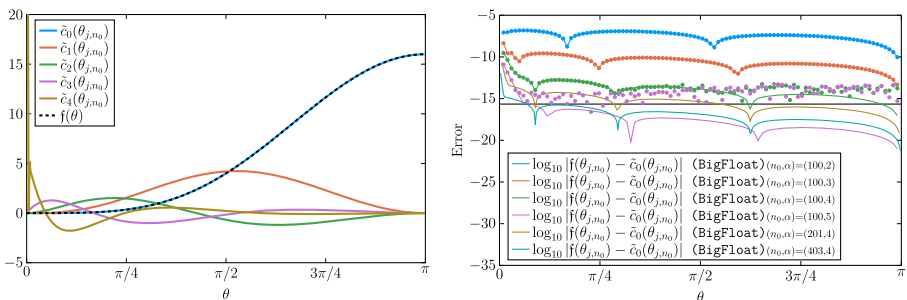


Fig. 10 [Example 6: Symbol $f(\theta) = 6 - 8 \cos(\theta) + 2 \cos(2\theta)$] Left: The approximated expansion functions $\tilde{c}_k(\theta_{j, n_0}), k = 0, \dots, \alpha$ for $n_0 = 100$ and $\alpha = 4$. Right: The absolute error $\log_{10} |f(\theta_{j, n_1}) - \tilde{c}_0(\theta_{j, n_1})|$, for different combinations of n_0 and α . All computations made with 128 bit `BigFloat` (lines) and `Float64` (dots)

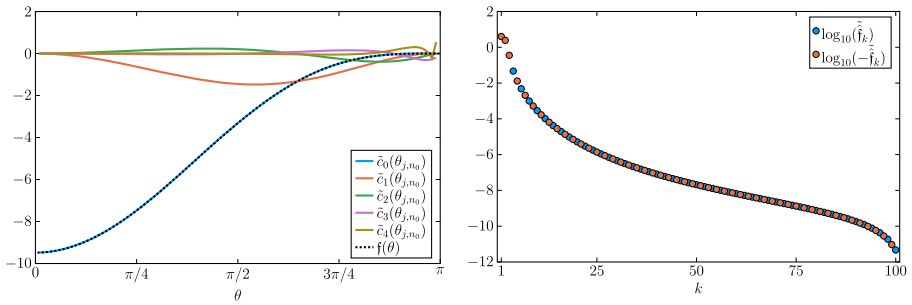


Fig. 11 [Example 7: Symbol $f(\theta) = e^{-i\theta}(6 - 8 \cos(\theta) + 2 \cos(2\theta))$] Left: The approximated expansion functions $\tilde{c}_k(\theta_{j,n_0})$, $k = 0, \dots, \alpha$ for $n_0 = 100$ and $\alpha = 4$. The approximation $\tilde{c}_0(\theta_{j,n_0})$ overlaps well with $\tilde{f}(\theta) = -\sin^4(\theta)/(\sin(\theta/4) \sin^3(3\theta/4))$. Note the erratic behavior of \tilde{c}_4 close to $\theta = \pi$. Right: The absolute value of the approximated first one hundred Fourier coefficients, $\log_{10} |\tilde{f}_k|$

The erratic behavior of $\tilde{c}_4(\theta)$ close to $\theta = 0$ in the left panel and the increased error close to $\theta = 0$ in the right panel are due to the fact that the symbol f violates the so-called simple-loop conditions, discussed in [3, 13].

Using Algorithm 2, we compute approximations of the Fourier coefficients of the monotonically increasing \tilde{f} to be $\tilde{f}_0 = 6, \tilde{f}_1 = -4, \tilde{f}_2 = 1$, and $\tilde{f}_k = 0$ for $k > 2$. We thus recover the true symbol $\tilde{f}(\theta) = f(\theta) = \hat{f}_0 + 2\hat{f}_1 \cos(\theta) + 2\hat{f}_2 \cos(2\theta)$. If Algorithm 1 was used to compute \tilde{c}_k for the monotonically decreasing \tilde{f} instead, the computed Fourier coefficients would be $\tilde{f}_0 = 6, \tilde{f}_1 = 4$, and $\tilde{f}_2 = 1$. In fact, for $\tilde{f}(\theta) = 6 \pm 8 \cos(\theta) + 2 \cos(2\theta)$, we have the same eigenvalues for $T_n(\tilde{f})$ and $T_n(f)$.

Example 7 In this example, we continue the investigation of $f(\theta) = e^{-i\theta}(6 - 8 \cos(\theta) + 2 \cos(2\theta))$ from Example 3. In Fig. 11, we show in the left panel the approximated expansion functions $\tilde{c}_k(\theta_{j,n_0})$ for $n_0 = 100$ and $\alpha = 4$. Computations

Table 1 [Example 7: Symbol $f(\theta) = e^{-i\theta}(6 - 8 \cos(\theta) + 2 \cos(2\theta))$] First ten true (\hat{f}_k) and computed (\tilde{f}_k) Fourier coefficients of \tilde{f} . Approximations computed using $n_0 = 100$ and $\alpha = 4$, and 256 bit precision

k	\hat{f}_k	\tilde{f}_k
0	-4.000000000000000	-3.999999999436239
1	-2.423215805461417	-2.423215806024005
2	-0.354481702999765	-0.354481702436023
3	0.046583829909932	0.046583829347381
4	-0.013008232443064	-0.013008231879376
5	0.004790313798591	0.004790313236114
6	-0.002068441503570	-0.002068440939976
7	0.000995276400689	0.000995275838326
8	-0.000518988396995	-0.000518987833535
9	0.000288215823752	0.000288215261541

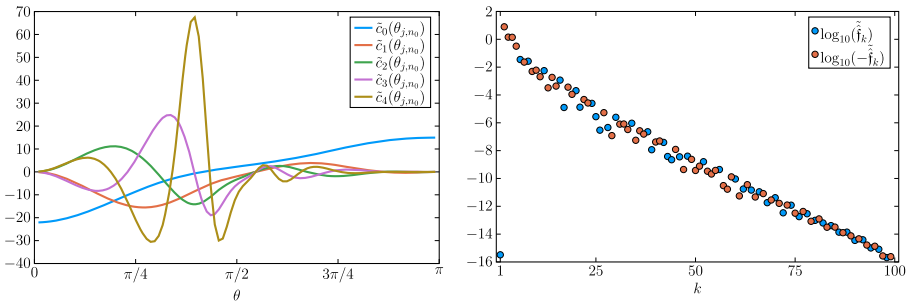


Fig. 12 [Example 8: Symbol $f(\theta) = -e^{-4i\theta} + 2e^{-3i\theta} - 2e^{-2i\theta} + 9e^{-i\theta} + 7e^{i\theta} - e^{2i\theta} + e^{3i\theta}$] Left: The approximated expansion functions $\tilde{c}_k(\theta_{j,n_0})$, $k = 0, \dots, \alpha$ for $n_0 = 100$ and $\alpha = 4$. The approximation $\tilde{c}_0(\theta_{j,n_0})$ corresponds well with $\lambda_j(T_{1000}(f))$ in the right panel of Fig. 7 (since f is unknown). Right: The absolute value of the approximated first one hundred Fourier coefficients, $\log_{10} |\tilde{f}_k|$

are made with 256 bit precision and the approximation $\tilde{c}_0(\theta_{j,n_0})$ overlaps well with f , defined in (7). Note the erratic behavior of \tilde{c}_4 close to $\theta = \pi$. In the right panel of Fig. 11, we show the absolute values of the first one hundred approximated Fourier coefficients \tilde{f}_k , given by Algorithm 2. In Table 1, we present the first ten true Fourier coefficients, \hat{f}_k , computed with f defined in (7) and (1), and the approximations \tilde{f}_k from Algorithm 2. Since f is not an RCTP, we can not recover the original simple expression of the symbol (7), but we can anyway obtain an approximated expression of f through our Algorithm 2.

Example 8 Finally, we return to the non-symmetric symbol discussed in Example 4, that is, $f(\theta) = -e^{-4i\theta} + 2e^{-3i\theta} - 2e^{-2i\theta} + 9e^{-i\theta} + 7e^{i\theta} - e^{2i\theta} + e^{3i\theta}$. Again, we employ Algorithms 1 and 2 to study the symbols f and \hat{f} . In the left panel of

Table 2 [Example 8: Symbol $f(\theta) = -e^{-4i\theta} + 2e^{-3i\theta} - 2e^{-2i\theta} + 9e^{-i\theta} + 7e^{i\theta} - e^{2i\theta} + e^{3i\theta}$] First ten computed (\tilde{f}_k) Fourier coefficients of the unknown f . Approximations computed using $n_0 = 100$ and $\alpha = 4$, and 512 bit precision

k	\tilde{f}_k
0	-0.0000000000000003
1	-7.931536795875190
2	-1.429849731406187
3	-1.393034471115375
4	-0.321121280053002
5	0.035288447846840
6	-0.023038821632295
7	0.026692519463291
8	-0.004916206049977
9	-0.006047350374789

Fig. 12, we present the approximated expansion functions in the working hypothesis, for $n_0 = 100$ and $\alpha = 4$. Computations are made with 512 bit precision. The blue line, $\tilde{c}_0(\theta_{j,n_0})$, corresponds to the approximation of the unknown symbol \mathfrak{f} . Recall the curve of $\lambda_j(T_{1000}(f))$ in the right panel of Fig. 7, which in principal matches the current \tilde{c}_0 . Note how all \tilde{c}_k , for $k > 0$, are zero in apparently the same point $\theta_0 \in [\frac{55\pi}{101}, \frac{56\pi}{101}]$. In the right panel of Fig. 12, we see the first one hundred approximated Fourier coefficients of \mathfrak{f} , by using Algorithm 2. Table 2 presents the first ten approximated Fourier coefficients, $\tilde{\mathfrak{f}}_k$.

5 Conclusions

The working hypothesis in this article concerns the existence of an asymptotic expansion such that there exists a function \mathfrak{f} describing the eigenvalue distribution of the Toeplitz matrices $T_n(f)$ generated by a symbol f . As opposed to previous versions of matrix-less methods no information of f or \mathfrak{f} is needed to approximate the expansion functions $\{\mathfrak{f} := c_0, c_1, c_2, \dots\}$. We have shown numerically that we can recover an approximation of the function \mathfrak{f} . This is done by a matrix-less method described in Algorithm 1, which in principle can be modified so as to work without any information on f or the way in which the eigenvalues of the smaller versions of $T_n(f)$ are computed. Algorithm 1 can also be used to fast and accurately compute the eigenvalues of $T_n(f)$ for an arbitrarily large order n , as highlighted in (10). However, in this article, we have focused only on using the obtained approximation of \mathfrak{f} to find an approximation of its truncated Fourier series; and if \mathfrak{f} is an RCTP, we have shown that we are able to recover the original function \mathfrak{f} analytically. These approaches can be a valuable tool for the exploration of the spectrum of Toeplitz and Toeplitz-like matrices previously not easily understood because of the high computational cost. For future research, we propose the extension to complex-valued functions \mathfrak{f} of the results presented herein, see, e.g., [19] and the recent [9]. Also, the study of non-banded Toeplitz matrices and matrices more general than $T_n(f)$ is of interest.

Acknowledgements The authors would like to thank Carlo Garoni, Stefano Serra-Capizzano, and the anonymous reviewers for valuable insights, comments, and suggestions during the preparation of this work.

Funding Open access funding provided by Uppsala University. The first author has been supported by the research grant “DRASI II” of Athens University of Economics and Business, and the International Postdoc Grant of the Swedish Research Council (Registration Number 2019-00495).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Generic Linear Algebra.jl. <https://github.com/JuliaLinearAlgebra/GenericLinearAlgebra.jl>
2. Ahmad, F., Al-Aidarous, E.S., Alrehailli, D.A., Ekström, S.-E., Furci, I., Serra-Capizzano, S.: Are the eigenvalues of preconditioned banded symmetric Toeplitz matrices known in almost closed form? *Numer. Algorithms* **78**, 867–893 (2017)
3. Barrera, M., Böttcher, A., Grudsky, S.M., Maximenko, E.A.: Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic. In: *The Diversity and Beauty of Applied Operator Theory*, pp. 51–77. Springer International Publishing (2018)
4. Batalshchikov, A., Grudsky, S., Malisheva, I., Mihalkovich, S., de Arellano, E.R., Stukopin, V.: Asymptotics of eigenvalues of large symmetric Toeplitz matrices with smooth simple-loop symbols. *Linear Algebra Appl.* **580**, 292–335 (2019)
5. Beam, R.M., Warming, R.F.: The asymptotic spectra of banded Toeplitz and quasi-Toeplitz matrices. *SIAM J. Sci. Comput.* **14**, 971–1006 (1993)
6. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. *SIAM Rev.* **59**, 65–98 (2017)
7. Bogoya, J.M., Böttcher, A., Grudsky, S.M., Maximenko, E.A.: Eigenvalues of Hermitian Toeplitz matrices with smooth simple-loop symbols. *J. Math. Anal. Appl.* **422**, 1308–1334 (2015)
8. Bogoya, J.M., Grudsky, S.M., Maximenko, E.A.: Eigenvalues of Hermitian Toeplitz matrices generated by simple-loop symbols with relaxed smoothness. In: *Large Truncated Toeplitz Matrices, Toeplitz Operators, and Related Topics*, pp. 179–212. Springer International Publishing (2017)
9. Böttcher, A., Gasca, J., Grudsky, S.M., Kozak, A.V.: Eigenvalue clusters of large tetradiagonal toeplitz matrices. *Integr. Equ. Oper. Theory*, 93 (2021)
10. Böttcher, A., Grudsky, S.M.: Spectral properties of banded Toeplitz matrices, Society for industrial and applied mathematics (2005)
11. Böttcher, A., Grudsky, S.M., Maximenko, E.A.: Inside the eigenvalues of certain Hermitian Toeplitz band matrices. *J. Comput. Appl. Math.* **233**, 2245–2264 (2010)
12. Böttcher, A., Silbermann, B.: *Introduction to Large Truncated Toeplitz Matrices*. Springer, New York (1999)
13. Ekström, S.-E.: Matrix-less methods for computing eigenvalues of large structured matrices, PhD thesis, Uppsala University, Uppsala, Acta Universitatis Upsaliensis (2018)
14. Ekström, S.-E.: Approximating the perfect sampling grids for computing the eigenvalues of Toeplitz-like Matrices Using the Spectral Symbol. arXiv:1901.06917 (2019)
15. Ekström, S.-E., Furci, I., Garoni, C., Manni, C., Serra-Capizzano, S., Speleers, H.: Are the eigenvalues of the λ -spline isogeometric analysis approximation of $-\Delta u = \lambda u$ known in almost closed form? *Numer. Linear Algebra Appl.* **25**, e2198 (2018)
16. Ekström, S.-E., Furci, I., Serra-Capizzano, S.: Exact formulae and matrix-less eigensolvers for block banded symmetric Toeplitz matrices. *BIT Numer. Math.* **58**, 937–968 (2018)
17. Ekström, S.-E., Garoni, C.: A matrix-less and parallel interpolation–extrapolation algorithm for computing the eigenvalues of preconditioned banded symmetric Toeplitz matrices. *Numer. Algorithms* **80**, 819–848 (2018)
18. Ekström, S.-E., Garoni, C., Serra-Capizzano, S.: Are the eigenvalues of banded symmetric Toeplitz matrices known in almost closed form? *Exp. Math.* **27**, 478–487 (2017)
19. Ekström, S.-E., Vassalos, P.: A Matrix-Less Method to Approximate the spectrum and the spectral function of Toeplitz matrices with complex eigenvalues, arXiv:1910.13810 (2019)
20. Garoni, C., Serra-Capizzano, S.: *Generalized locally Toeplitz sequences: theory and applications (Volume 1)*, Springer International Publishing (2017)
21. Parter, S.V., Youngs, J.: The symmetrization of matrices by diagonal matrices. *J. Math. Anal. Appl.* **4**, 102–110 (1962)
22. Reichel, L., Trefethen, L.N.: Eigenvalues and pseudo-eigenvalues of Toeplitz matrices. *Linear Algebra Appl.* **162–164**, 153–185 (1992)
23. Schmidt, P., Spitzer, F.: The Toeplitz matrices of an arbitrary Laurent polynomial. *Math. Scand.* **8**, 15–38 (1960)
24. Shapero, B., Stampach, F.: Non-self-adjoint Toeplitz matrices whose principal submatrices have real spectrum. *Constr. Approx.* **49**, 191–226 (2019)

25. Tilli, P.: Some results on complex Toeplitz eigenvalues. *J. Math. Anal. Appl.* **239**, 390–401 (1999)
26. Trefethen, L.N., Embree, M.: *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, Princeton (2005)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.