Numerical Algorithms
https://doi.org/10.1007/s11075-022-01318-7

ORIGINAL PAPER

Fast Toeplitz eigenvalue computations, joining interpolation-extrapolation matrix-less algorithms and simple-loop theory

Manuel Bogoya[1,2] · **Sven-Erik Ekström**[3] · **Stefano Serra-Capizzano**[2,3]

Received: 7 January 2022 / Accepted: 14 April 2022

Abstract
Under appropriate technical assumptions, the simple-loop theory allows to derive various types of asymptotic expansions for the eigenvalues of Toeplitz matrices generated by a function f. Independently and under the milder hypothesis that f is even and monotone over $[0, \pi]$, matrix-less algorithms have been developed for the fast eigenvalue computation of large Toeplitz matrices, within a linear complexity in the matrix order: behind the high efficiency of such algorithms there are the expansions predicted by the simple-loop theory, combined with the extrapolation idea. Here we focus our attention on a change of variable, followed by the asymptotic expansion of the new variable, and we adapt the matrix-less algorithm to the considered new setting. Numerical experiments show a higher precision (till machine precision) and the same linear computation cost, when compared with the matrix-less procedures already presented in the relevant literature. Among the advantages, we concisely mention the following: (a) when the coefficients of the simple-loop function are analytically known, the algorithm computes them perfectly; (b) while the proposed algorithm is better or at worst comparable to the previous ones for computing the inner eigenvalues, it is vastly better for the computation of the extreme eigenvalues; a mild deterioration in the quality of the numerical experiments is observed when dense Toeplitz matrices are considered, having generating function of low smoothness and not satisfying the simple-loop assumptions.

Keywords Eigenvalue computation · Toeplitz matrix · Matrix-less method · Asymptotic expansion

Mathematics Subject Classification (2010) 15B05 · 65F15 · 65D05 · 47B35. Secondary 15A18 · 47A38

✉ Manuel Bogoya
johan.bogoya@correounivalle.edu.co; johanmanuel.bogoya@uninsubria.it

Extended author information available on the last page of the article.

Published online: 16 May 2022

1 Introduction

The target of this article is to design fast procedures for the computation of all the spectra of large Toeplitz matrices having an even generating function which is monotone in the interval $[0, \pi]$. For the formal definition of Toeplitz matrix generated by a Lebesgue integrable function over the basic interval $Q \equiv (-\pi, \pi)$ (see the first lines of Section 2).

This topic has been studied in the recent years by several researchers. Indeed, taking into account a clear numerical evidence developed in a systematic series of the numerical tests, in [1], the second and third authors formulated a variation of the following conjecture.

Conjecture 1.1 *Let l, g be two real-valued even functions with $g > 0$ on $(0, \pi)$, and suppose that $f \equiv \frac{l}{g}$ is monotone increasing over $(0, \pi)$. Set $X_n \equiv T_n^{-1}(g)T_n(l)$ for all n. Then, for some integer $K \geqslant 0$, every n, and every $j = 1, \ldots, n$, the following asymptotic expansion holds:*

$$\lambda_j(X_n) = f(\theta_{j,n}) + \sum_{k=1}^{K} c_k(\theta_{j,n})h^k + E_{j,n,K},$$

where:

- *the eigenvalues of X_n are arranged in non-decreasing order, that is $\lambda_1(X_n) \leqslant \cdots \leqslant \lambda_n(X_n)$;* [1]
- *$\{c_k\}_{k=1}^{K}$ are functions from $(0, \pi)$ to $\mathbb{R}$ which depends only on l, g;*
- *$h \equiv \frac{1}{n+1}$ and $\theta_{j,n} \equiv \frac{j\pi}{n+1} = j\pi h$;*
- *$E_{j,n,K} = O(h^{K+1})$ is the remainder (the error), which satisfies the inequality $|E_{j,n,K}| \leqslant ch^{K+1}$ for some constant c depending only on l and g.*

In the case where $g = 1$ identically, a variation of Conjecture 1.1 was originally formulated and supported through numerical experiments in [1]. Then, the algorithmic proposal was extended and refined in [2–5]. When $g \equiv 1$ and l satisfies further technical additional assumptions, those of the simple-loop method, Conjecture 1.1 was formally proved by Bogoya, Böttcher, Grudsky, and Maximenko in a series of papers [6–8]. In the paper [9], the authors considered the function $g(\theta) = (2 - 2\cos\theta)^2$, which produces a pentadiagonal Toeplitz matrix, and formally proved that Conjecture 1.1 is true for $K \leqslant 3$ and false if $K \geqslant 4$. Hence, the selection of the constant K is a key part in our work and can not be taken for granted.

However, again in the case of $g \equiv 1$, the power of the simple-loop method has been not exploited completely, since in the case of a continuous generating function,

[1] Note that the eigenvalues of X_n are real, because $T_n(g)$ is symmetric positive definite and X_n is similar to the symmetric matrix $T_n^{-\frac{1}{2}}(g)T_n(l)T_n^{-\frac{1}{2}}(g)$.

the distribution results reported in Theorem 2.1 imply that $\lambda_j(T_n(f)) = f(s_{j,n})$ with $s_{j,n}$ belonging to $(0, \pi)$ and distributed as

$$s_{j,n} = \frac{\pi j}{n+1} + o(1),$$

that is the identity function. However, more is known and indeed also in the independent variable θ there exists an asymptotic expansion regarding exactly the points $s_{j,n}$ (see, for example, [8, Th.2.2]).

This article deals with the adaptation of the interpolation-extrapolation algorithms to the previous change of variable, that is θ instead of λ, joined with a trick at the end points introduced in [10].

The numerical results are extremely precise, even compared with the already good performances described in [1–5], since it is not difficult to reach machine precision, and the complexity is still linear.

The present paper is organized as follows. Preliminary definitions, tools, and results are concisely reported in Section 2. Section 3 presents the new adapted algorithm for computing the Toeplitz eigenvalues: as in [5], our technique combines the extrapolation procedure proposed in [1, 2] – which allows the computation of *some* of the eigenvalues of X_n – with an appropriate interpolation process, designed for the simultaneous computation of *all* the eigenvalues of X_n, with the additional end point trick in [10]. In Section 4 we present the numerical experiments, while in Section 5 we draw conclusions and we list few open problems for research lines to be investigated in the next future.

2 Preliminaries and tools

For a real or complex valued function f in $L^1(-\pi, \pi)$, let $\mathfrak{a}_j(f)$ be its jth Fourier coefficient, i.e.,

$$\mathfrak{a}_j(f) \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) \mathrm{e}^{-\mathrm{i}j\theta} \mathrm{d}\theta, \quad j \in \mathbb{Z},$$

and consider the sequence $\{T_n(f)\}_{n=1}^{\infty}$ of the $n \times n$ Toeplitz matrices defined by $T_n(f) \equiv (\mathfrak{a}_{j-k}(f))_{j,k=0}^{n-1}$. The function f is customarily referred to as the generating function of this sequence.

As a second step, we introduce some notations and definitions concerning general sequences of matrices. For any function F defined on the complex field and for any matrix A_n of size d_n, by the symbol $\Sigma_\lambda(F, A_n)$, we denote the mean

$$\Sigma_\lambda(F, A_n) \equiv \frac{1}{d_n} \sum_{j=1}^{d_n} F[\lambda_j(A_n)],$$

while by the symbol $\Sigma_\sigma(F, A_n)$, we denote the mean

$$\Sigma_\sigma(F, A_n) \equiv \frac{1}{d_n} \sum_{j=1}^{d_n} F[\sigma_j(A_n)].$$

Here, $\lambda_j(A_n)$ and $\sigma_j(A_n)$ are the eigenvalues and singular values of the matrix A_n, respectively.

Definition 2.1 Given a sequence $\{A_n\}$ of matrices of size d_n with $d_n < d_{n+1}$ and given a Lebesgue-measurable function ψ defined over a measurable set $K \subset \mathbb{R}^\nu$ of finite and positive Lebesgue measure $\mu(K)$, we say that $\{A_n\}$ is distributed as (ψ, K) in the sense of the eigenvalues if for every continuous function F with bounded support, the following limit relation holds:

$$\lim_{n\to\infty} \Sigma_\lambda(F, A_n) = \frac{1}{\mu(K)} \int_K F(\psi)\, \mathrm{d}\mu.$$

In this case, we write in short $\{A_n\} \sim_\lambda (\psi, K)$ and we say that ψ is the (spectral) symbol of $\{A_n\}$. Furthermore, we say that $\{A_n\}$ is distributed as (ψ, K) in the sense of the singular values if for every continuous function F with bounded support, the following limit relation holds:

$$\lim_{n\to\infty} \Sigma_\sigma(F, A_n) = \frac{1}{\mu(K)} \int_K F(|\psi|)\, \mathrm{d}\mu.$$

In this case, we write in short $\{A_n\} \sim_\sigma (\psi, K)$, which is equivalent to saying that $\{A_n^* A_n\} \sim_\lambda (|\psi|^2, K)$, and we say that ψ is the singular value symbol of $\{A_n\}$.

In Remark 2.1, we provide an informal meaning of the notion of eigenvalue distribution. For the singular value distribution similar statements can be made.

Remark 2.1 The informal meaning behind the above definition is the following. If ψ is continuous, n is large enough, and

$$\left\{\mathbf{x}_j^{(d_n)},\ j = 1, \ldots, d_n\right\},$$

is an equispaced grid on K, then a suitable ordering $\lambda_j(A_n)$, $j = 1, \ldots, d_n$, of the eigenvalues of A_n is such that the pairs $\left\{\left(\mathbf{x}_j^{(d_n)}, \lambda_j(A_n)\right),\ j = 1, \ldots, d_n\right\}$ reconstruct approximately the hypersurface

$$\left\{(\mathbf{x}, \psi(\mathbf{x})),\ \mathbf{x} \in K\right\}.$$

In other words, the spectrum of A_n 'behaves' like a uniform sampling of ψ over K. For instance, if $\nu = 1$, $d_n = n$, and $K = [a, b]$, then the eigenvalues of A_n are approximately equal to $\psi\left(a + \frac{j}{n}(b-a)\right)$, $j = 1, \ldots, n$, for n large enough and up to at most $o(n)$ outliers. Analogously, if $\nu = 2$, $d_n = n^2$, and $K = [a_1, b_1] \times [a_2, b_2]$, then the eigenvalues of A_n are approximately equal to $\psi\left(a_1 + \frac{j}{n}(b_1 - a_1), a_2 + \frac{k}{n}(b_2 - a_2)\right)$, $j, k = 1, \ldots, n$, for n large enough and up to at most $o(n^2)$ outliers.

The asymptotic distribution of eigenvalues and singular values of Toeplitz matrix sequences has been studied deeply and continuously in the last century (for example, see [8, 11–15] and references therein). The starting point of this theory, which contains many extensions and other results, is a famous theorem of Szegő [16], which we report in the version due to Tyrtyshnikov and Zamarashkin [17].

Theorem 2.1 *If f is integrable over $Q \equiv (-\pi, \pi)$, and if $\{T_n(f)\}$ is the sequence of Toeplitz matrices generated by f, then f is the singular value symbol of $\{T_n(f)\}$*

$$\{T_n(f)\} \sim_\sigma (f, Q).$$

Moreover, if f is also real-valued, then each matrix $T_n(f)$ is Hermitian and f is the (spectral) symbol of $\{T_n(f)\}$ that is

$$\{T_n(f)\} \sim_\lambda (f, Q).$$

Furthermore, strong localization results are known in the case where the generating function is real-valued, as stated in [18, Th.2.2] which we partly report below.

Theorem 2.2 *If f is integrable and real-valued a.e. over $Q \equiv (-\pi, \pi)$, m is the essential infimum of f, M is the essential supremum of f, and if $\{T_n(f)\}$ is the sequence of Toeplitz matrices generated by f, then*

$$\lambda_j(T_n(f)) \in (m, M)$$

for every $j = 1, \ldots, n$, for every positive integer n, under the assumption $m < M$. Moreover, if $m = M$ then the generating function f is constant almost everywhere and trivially we conclude that $T_n(f)$ coincides with m times the identity matrix.

From now on, in the light of Theorem 2.1, we will use with no distinction the expressions "(spectral) symbol" and "generating function," when the Toeplitz Hermitian setting is considered.

2.1 The simple-loop case

For $\alpha > 0$, the well-known weighted Wiener algebra W^α is the collection of all functions $f : \mathbb{T} \to \mathbb{C}$ whose Fourier coefficients satisfy

$$\|f\|_\alpha \equiv \sum_{k=-\infty}^{\infty} |\mathfrak{a}_k(f)|(|k| + 1)^\alpha < \infty.$$

It is easy to see that if $f \in W^\alpha$ then $f \in C^{\lfloor \alpha \rfloor}(-\pi, \pi)$, hence the constant α is measuring the smoothness of the symbol f. In what follows we extend every symbol f to the whole real line in the natural way turning it into a 2π-periodic function, and we denote this extension by f as well.

The simple-loop class, denoted by SL^α, is the collection of all the real-valued symbols in W^α tracing out a simple-loop over the interval $[-\pi, \pi]$ with the following properties:

(i) the range of f is a segment $[0, \mu]$ with $\mu > 0$;
(ii) $f(0) = f(2\pi) = 0$, $f''(0) = f''(2\pi) > 0$;
(iii) there is a unique $\theta_0 \in [0, 2\pi]$ such that $f(\theta_0) = \mu$, $f'(\theta) > 0$ for $\theta \in (0, \theta_0)$ and $f'(\theta) < 0$ for $\theta \in (\theta_0, 2\pi)$.

Note that if $f \in SL^\alpha$ then $f'(0) = f'(2\pi) = 0$, and that $\theta_0 = \pi$ for every even symbol, i.e., a symbol satisfying $f(\theta) = f(-\theta)$ for $\theta \in \mathbb{R}$.

Consider an even symbol $f \in SL^\alpha$ with $\alpha > 2$. For a $n \times n$ matrix A let $\lambda_j(A)$ $(j = 1, \ldots, n)$ be its eigenvalues. In the works [6, 7, 10, 19], for example, the authors state that

(i) the eigenvalues of $T_n(f)$ are all distinct, i.e.,

$$\lambda_1(T_n(f)) < \lambda_2(T_n(f)) < \cdots < \lambda_n(T_n(f));$$

(ii) let g be the inverse function of f when restricted to the interval $[0, \pi]$. The numbers $s_{j,n} \equiv g(\lambda_j(T_n(f)))$ for $j = 1, \ldots, n$, satisfy

$$(n+1)s_{j,n} + \eta(s_{j,n}) = \pi j + E_{j,n,\alpha}, \tag{2.1}$$

where $E_{j,n,\alpha}$ is an error term satisfying certain bounding condition, and η is a function depending only on f with certain smoothness depending on α;

(iii) the previous (2.1) has exactly one solution $s_{j,n} \in [0, \pi]$ for each $j = 1, \ldots, n$.

In the bulk of the so-called *simple-loop method*, the authors use the Banach fixed-point theorem to iterate over (2.1) and solve it for $s_{j,n}$, obtaining an expansion of the kind (see [6, Th.2.2] for example)

$$\lambda_j(T_n(f)) = f(s_{j,n}), \qquad s_{j,n} = \theta_{j,n} + \sum_{k=1}^{\lfloor \alpha \rfloor} r_k(\theta_{j,n}) h^k + E_{j,n,\alpha}, \tag{2.2}$$

where

- the numbers $s_{j,n}$ are arranged in nondecreasing order;
- $h \equiv \frac{1}{n+1}$ and $\theta_{j,n} \equiv \pi j h$;
- the coefficients r_k depend only on f and can be found explicitly, for example,

$$r_1 = -\eta, \quad r_3 = -\eta(\eta')^2 - \frac{1}{2}\eta^2\eta'',$$
$$r_2 = \eta\eta', \quad r_4 = \eta(\eta')^3 + \frac{3}{2}\eta^2\eta'\eta'' + \frac{1}{6}\eta^3\eta'''.$$

- $E_{j,n,\alpha} = O(h^\alpha)$ is the remainder (error) term, which satisfies the bounding $|E_{j,n,\alpha}| \leqslant \kappa_\alpha h^\alpha$ for some constant κ_α depending only on α and f.

Using the expansion (2.2) and the smoothness of the symbol f, the authors apply f to both sides obtaining an expansion of the kind

$$\lambda_j(T_n(f)) = f(\theta_{j,n}) + \sum_{k=1}^{\lfloor \alpha \rfloor} c_k(\theta_{j,n}) h^k + E_{j,n,\alpha}, \tag{2.3}$$

with similar characteristics but where the coefficients c_k involve the symbol and its derivatives, for instance $c_1 = -f'\eta$ and $c_2 = f'\eta\eta' + \frac{1}{2}f''\eta^2$. The previous works [1, 4, 5, 10] used (2.3) as the basic expansion. For a simple-loop symbol f, we have $f'(0) = f'(\pi) = 0$, and hence f is 'barely' increasing at $\{0, \pi\}$, which

produces inaccuracies in the numerical experiments. Hence, we decided to work with the following expansion, which is a variation of (2.2), instead.

Conjecture 2.1 *Let f be a real-valued even function with $f > 0$ on $(0, \pi)$, and suppose that f is monotone increasing over $(0, \pi)$. Then, for some integer $K \geqslant 0$, every n, and every $j = 1, \ldots, n$, the following asymptotic expansion holds:*

$$\lambda_j(T_n(f)) = f(s_{j,n}), \quad s_{j,n} = \theta_{j,n} + \sum_{k=1}^{K} r_k(\theta_{j,n}) h^k + E_{j,n,K},$$

where:

- *the eigenvalues of $T_n(f)$ are arranged in non-decreasing order, that is $\lambda_1(T_n(f)) \leqslant \cdots \leqslant \lambda_n(T_n(f))$;*
- *$\{r_k\}_{k=1}^{K}$ are functions from $(0, \pi)$ to $\mathbb{R}$ which depends only on f;*
- *$h \equiv \frac{1}{n+1}$ and $\theta_{j,n} \equiv \frac{j\pi}{n+1} = j\pi h$;*
- *$E_{j,n,K} = O(h^{K+1})$ is the remainder (the error), which satisfies the inequality $|E_{j,n,K}| \leqslant c h^{K+1}$ for some constant c depending only on f.*

In the light of Theorem 2.1, since f is an even and real-valued function, we find $\{T_n(f)\} \sim_\lambda (f, [0, \pi])$. Consequently we deduce

$$\{\mathrm{diag}_{j=1,\ldots,n}(s_{j,n})\} \sim_\lambda (\mathrm{id}, [0, \pi]), \quad \mathrm{id}(\theta) \equiv \theta,$$

with $s_{j,n} \in (0, \pi)$, by virtue of Theorem 2.2. Notice that the function $\mathrm{id}(\cdot)$ is very basic, and, as already claimed, when compared with the study and the proposals in the works [1, 3–8, 10], the troubles that the derivatives of f produce are completely removed: these nice features are clearly evident by looking at the high precision of the numerical computations, reported in Section 4, containing the numerical experiments.

3 The algorithm

Our algorithm is based on the expansion in Conjecture 2.1 and is an evolution of the algorithms proposed in [1, 4, 5, 10]. As in the mentioned works it is suited for parallel implementation and can be called matrix-less since it does not require to calculate or even to store the matrix entries. For every $n \in \mathbb{N}$ let $h \equiv \frac{1}{n+1}$ and $\theta_{j,n} \equiv \pi j h$, thus the collection $\{\theta_{j,n}\}_{j=0}^{n+1}$ is a regular grid for the interval $[0, \pi]$ with step size πh. We will mirror the notations for n and h, for example, h_k means $\frac{1}{n_k+1}$, and so on. We assume that

- the symbol f is even and real-valued, strictly increasing in the interval $[0, \pi]$, and $f(0) = 0$;
- n_1 and K are fixed natural numbers and $n \gg n_1$;
- for $k = 1, \ldots, K$ let $n_k \equiv 2^{k-1}(n_1 + 1) - 1$;
- for $j_1 = 1, \ldots, n$ and $k = 1, \ldots, K$, let $j_k \equiv 2^{k-1} j_1$.

Note that j_k depends on j_1, and similarly, n_k depends on n_1, but for notation simplicity, we suppressed those dependencies. The numbers j_k and the matrix sizes n_k were calculated in such a way that

$$\theta_{j_1,n_1} = \theta_{j_2,n_2} = \cdots = \theta_{j_K,n_K},$$

which is the key idea of the following extrapolation phase (see Fig. 1). We also want to emphasize that f is not necessarily simple-loop.

As in [1, 4, 5, 10], our algorithm is designed to calculate eigenvalues for "big" matrix sizes n with respect to $n_1, \ldots, n_K$, meaning that, from a computational viewpoint, the calculation of the eigenvalues of $T_n(f)$ is hard while for $T_{n_k}(f)$ it can be easily done with any standard eigensolver (i.e., `Eigenvalues` in MATHEMATICA, `eig` in MATLAB, or `eigvals` in JULIA). But our proposal is able to reach machine precision accuracy easily. The algorithm has two phases; the first one involves an extrapolation procedure, and the second one consists in a local interpolation technique. As a precomputing phase we need to calculate the eigenvalues of $T_n(f)$ for $n = n_1, \ldots, n_K$.

Extrapolation For each fixed $j_1 = 1, \ldots, n_1$ let $\sigma_{j_1} \equiv \theta_{j_1,n_1} = \cdots = \theta_{j_K,n_K}$ (that is, a vertical line of orange dots in Fig. 1), and apply K times the expansion in Conjecture 2.1 obtaining,

$$\begin{aligned}
s_{j_1,n_1} - \sigma_{j_1} &= r_1(\sigma_{j_1})h_1 + r_2(\sigma_{j_1})h_1^2 + \cdots + r_K(\sigma_{j_1})h_1^K + E_{j_1,n_1,K},\\
s_{j_2,n_2} - \sigma_{j_1} &= r_1(\sigma_{j_1})h_2 + r_2(\sigma_{j_1})h_2^2 + \cdots + r_K(\sigma_{j_1})h_2^K + E_{j_2,n_2,K},\\
&\vdots\\
s_{j_K,n_K} - \sigma_{j_1} &= r_1(\sigma_{j_1})h_K + r_2(\sigma_{j_1})h_K^2 + \cdots + r_K(\sigma_{j_1})h_K^K + E_{j_K,n_K,K}.
\end{aligned}$$

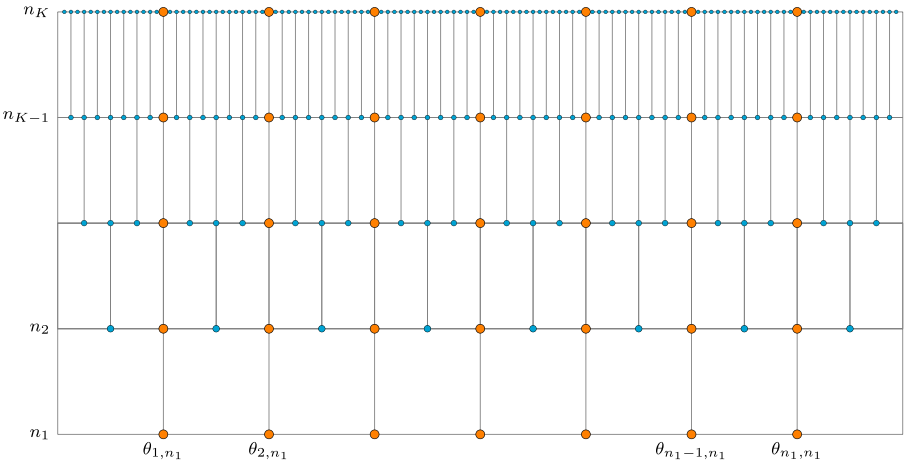

Fig. 1 The regular grids $\{\theta_{j,n_k}\}$ for $j = 1, \ldots, n_k$ and $k = 1, \ldots, K$. In the precomputing phase we will need to calculate the eigenvalues of $T_{n_k}(f)$ for $k = 1, \ldots, K$ corresponding to the blue and orange dots combined, but in the interpolation phase, we will use only the eigenvalues corresponding to the orange dots, that is $\lambda_{j_k}(T_{n_k}(f))$ for $j_1 = 1, \ldots, n_1$ and $k = 1, \ldots, K$

Let $\hat{r}_k(\sigma_{j_1})$ be the approximation of $r_k(\sigma_{j_1})$ obtained by removing all the error terms $E_{j_k,n_k,K}$ and solving the resulting linear system:

$$\begin{bmatrix} h_1 & h_1^2 & \cdots & h_1^K \\ h_2 & h_2^2 & \cdots & h_2^K \\ \vdots & \vdots & \ddots & \vdots \\ h_K & h_K^2 & \cdots & h_K^K \end{bmatrix} \begin{bmatrix} \hat{r}_1(\sigma_{j_1}) \\ \hat{r}_2(\sigma_{j_1}) \\ \vdots \\ \hat{r}_K(\sigma_{j_1}) \end{bmatrix} = \begin{bmatrix} s_{j_1,n_1} \\ s_{j_2,n_2} \\ \vdots \\ s_{j_K,n_K} \end{bmatrix} - \sigma_{j_1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{3.1}$$

As mentioned in previous works, a variant of this extrapolation strategy was first suggested by Albrecht Böttcher in [6, Sc.7] and is analogous to the Richardson extrapolation employed in the context of Romberg integration [20, Sc.3.4].

Interpolation For any $j \in \{1, \ldots, n\}$ we will estimate $r_k(\theta_{j,n})$. If $\theta_{j,n}$ coincides with one of the points in the grid $\{\theta_{0,n_1}, \ldots, \theta_{n_1+1,n_1}\}$, then we have the approximations $\hat{r}_k(\theta_{j,n})$ from the extrapolation phase for free. In any other case, we will do it by interpolating the data

$$(\theta_{0,n_k}, \hat{r}_k(\theta_{0,n_k})), (\theta_{1,n_k}, \hat{r}_k(\theta_{1,n_k})), \ldots, (\theta_{n_k+1,n_k}, \hat{r}_k(\theta_{n_k+1,n_k})),$$

for $k = 1, \ldots, K$, and then evaluating the resulting polynomial at $\theta_{j,n}$. This interpolation can be done in many ways, but to avoid spurious oscillations explained by the Runge phenomenon [21, p.78], and following the strategy of the previous works, we decided to do it considering only the $K-k+5$ points in the grid $\{\theta_{0,n_1}, \ldots, \theta_{n_1+1,n_1}\}$ which are closest to $\theta_{j,n}$. Those points can be determined uniquely unless $\theta_{j,n}$ is in the middle of two consecutive points in the grid, in which case we can take any of the possible two choices.

Finally, our eigenvalue approximation with k terms is given by

$$\hat{\lambda}_{j,k}^{\text{NAS}}(T_n(f)) \equiv f\Big(\theta_{j,n} + \sum_{\ell=0}^{k-1} \hat{r}_\ell(\theta_{j,n}) h^\ell\Big), \tag{3.2}$$

where $k = 1, \ldots, K$, and NAS stands for "Numerical Algorithm in the variable $s_{j,k}$".

Remark 3.1 To get the best result possible, in the precomputing phase, we advise to use the proposed algorithm (3.2), calculating the eigenvalues of the matrices $T_n(f)$ for $n = n_1, \ldots, n_K$ with a significant number of precision digits, let us say 60. For $K = 5$ and $n_1 = 100$, for instance, this can be done in a standard computer in a few minutes, and it only needs to be done once. While for the extrapolation phase (3.1), we advice to do it with the largest K possible, even if less terms will be used.

4 Numerical experiments

Let $\lambda_{j,k}^{\text{SL}}(T_n(f))$ be the kth term approximation of $\lambda_j(T_n(f))$ obtained with the simple-loop method in the variable $s_{j,n}$, that is

$$\lambda_{j,k}^{\text{SL}}(T_n(f)) \equiv f\Big(\theta_{j,n} + \sum_{\ell=1}^{k-1} r_\ell(\theta_{j,n}) h^\ell\Big),$$

where $\theta_{j,n} = \pi jh$. Let also $\lambda_{j,k}^{\mathrm{NA}}(T_n(f))$ be the kth term approximation of $\lambda_j(T_n(f))$ given by the Numerical Algorithm in [5] and finally, let $\lambda_{j,k}^{\mathrm{MNA}}(T_n(f))$ be the respective approximation given by the Modified Numerical Algorithm [10, Sc.4]. In order to compare the results of the different methods we use the following notation for the absolute individual errors:

$$\varepsilon_{j,n,k}^{\mathrm{SL}} \equiv |\lambda_j(T_n(f)) - \lambda_{j,k}^{\mathrm{SL}}(T_n(f))|, \qquad \varepsilon_{j,n,k}^{\mathrm{NA}} \equiv |\lambda_j(T_n(f)) - \lambda_{j,k}^{\mathrm{NA}}(T_n(f))|,$$
$$\varepsilon_{j,n,k}^{\mathrm{MNA}} \equiv |\lambda_j(T_n(f)) - \lambda_{j,k}^{\mathrm{MNA}}(T_n(f))|, \qquad \varepsilon_{j,n,k}^{\mathrm{NAS}} \equiv |\lambda_j(T_n(f)) - \lambda_{j,k}^{\mathrm{NAS}}(T_n(f))|,$$

and the respective maximum absolute errors:

$$\varepsilon_{n,k}^{\mathrm{SL}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{SL}} : j = 1, \ldots, n\}, \qquad \varepsilon_{n,k}^{\mathrm{NA}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{NA}} : j = 1, \ldots, n\},$$
$$\varepsilon_{n,k}^{\mathrm{MNA}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{MNA}} : j = 1, \ldots, n\}, \qquad \varepsilon_{n,k}^{\mathrm{NAS}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{NAS}} : j = 1, \ldots, n\}.$$

The numerical computations for the "exact" eigenvalues were conducted with the `BigFloat` data type in JULIA, with 1024 bit precision which approximately corresponds to 300 decimal digits of accuracy. We start with an example involving a well-known simple-loop symbol for which we can exactly calculate the coefficients r_k of the expansion in Conjecture 2.1 easily. Thus, we will be able to compare the accuracy of the different eigenvalue approximations.

Example 4.1 (A simple-loop symbol) Consider the even simple-loop symbol given by

$$f(\theta) \equiv \frac{(1+\rho)^2}{2} \cdot \frac{1-\cos\theta}{1-2\rho\cos\theta+\rho^2}, \qquad \theta \in [0, 2\pi], \tag{4.1}$$

for a constant $0 < \rho < 1$ (see Fig. 2). This symbol was inspired by the Kac–Murdock–Szegő Toeplitz matrices introduced in [22] and subsequently studied in [23, 24], which usually are present in important physics models.

The respective Fourier coefficients can be calculated as $\mathfrak{a}_k(f) = \frac{1}{4}(\rho^2-1)\rho^{|k|-1}$ for $k \neq 0$ and $\mathfrak{a}_k(f) = \frac{1}{2}(1+\rho)$ for $k = 0$. Then, we have,

$$\|f\|_\alpha = \frac{1+\rho}{2} + \frac{\rho^2-1}{2\rho}\sum_{k=1}^{\infty}\rho^k(k+1)^\alpha,$$

which is finite for every $\alpha > 0$, the remaining simple-loop conditions are easily verified in Fig. 2. Then, $f \in SL^\alpha$ for any $\alpha > 0$. According to [10, Sc.4], the function η in (2.1) is nicely given by

$$\eta(s) = 2\arctan\left(\frac{\rho\sin(s)}{1-\rho\cos(s)}\right).$$

Figure 3 and Table 1 show that, in this case, the approximation $\lambda_{j,k}^{\mathrm{MNA}}(T_n(f))$ can produce good results until the level $k = 2$ but it becomes unstable from this point on. While our algorithm (3.2) is still producing fine results in the 4th level, for a matrix of size 4096, it reaches machine-precision from level 3. Figure 3 reveals also that for the symbol (4.1), our proposed algorithm (3.2) can almost match the exact asymptotic simple-loop expansion until level 4.

In this example, Conjecture 2.1 is true for any number of levels K but the level 5 produced an interesting result, our algorithm (3.2) lowered the errors of the level 4

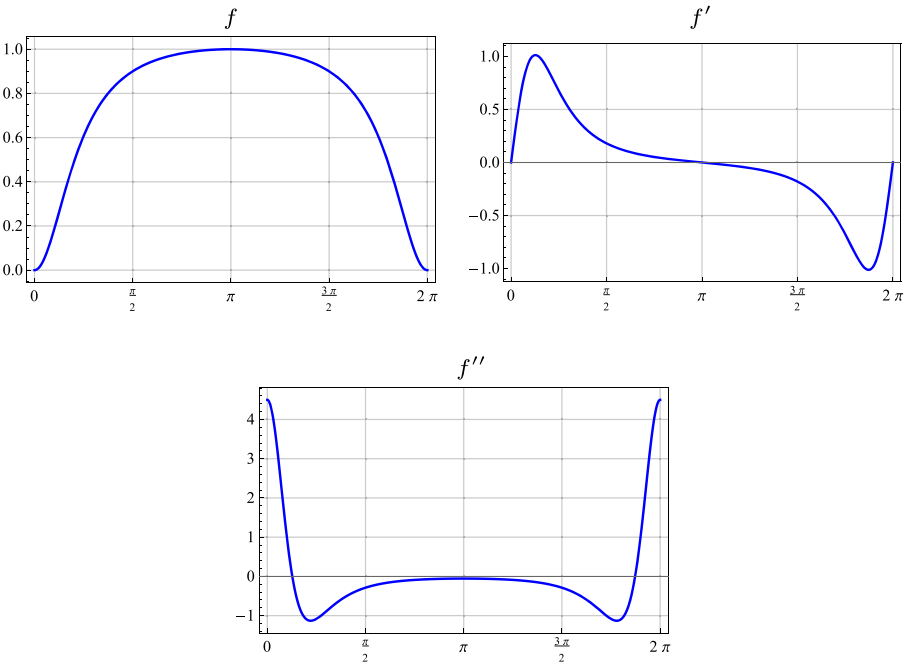

Fig. 2 The symbol f in (4.1) and its first two derivatives for $\rho = \frac{1}{2}$

but it was not able to match the exact asymptotic simple-loop expansion, proving that there is room for improving the numerical algorithm.

Example 4.2 (A non-simple-loop symbol) We now test our algorithm with a Real Cosine Trigonometric Polynomial (RCTP) (see [1, Sc.1]). For $\ell \in \mathbb{Z}_+$, consider the symbol

$$f_\ell(\theta) \equiv (2 - 2\cos\theta)^\ell, \quad \theta \in Q. \tag{4.2}$$

The respective Fourier coefficients can be exactly calculated as $\mathfrak{a}_k(f_\ell) = (-1)^k \binom{2\ell}{\ell+k}$ for $|k| \leqslant \ell$ and $\mathfrak{a}_k(f_\ell) = 0$ in any other case. Then the respective Toeplitz matrices $T_n(f_\ell)$ are banded with a band of size $2\ell + 1$. The case $\ell = 2$ was carefully studied by Barrera and Grudsky in [25], where they rigorously proved that

$$s_{j,n} = \frac{\pi(j+1)}{n+2} + \frac{u_{1,j}}{n+2} + \frac{u_{2,j}}{(n+2)^2} + O(h^3),$$

with some bounded and continuous coefficients $u_{1,j}, u_{2,j}$ (see Theorem 2.5 there). The previous expansion is slightly different from the one in Conjecture 2.1 but we were able to show that our algorithm is producing fine results in this case.

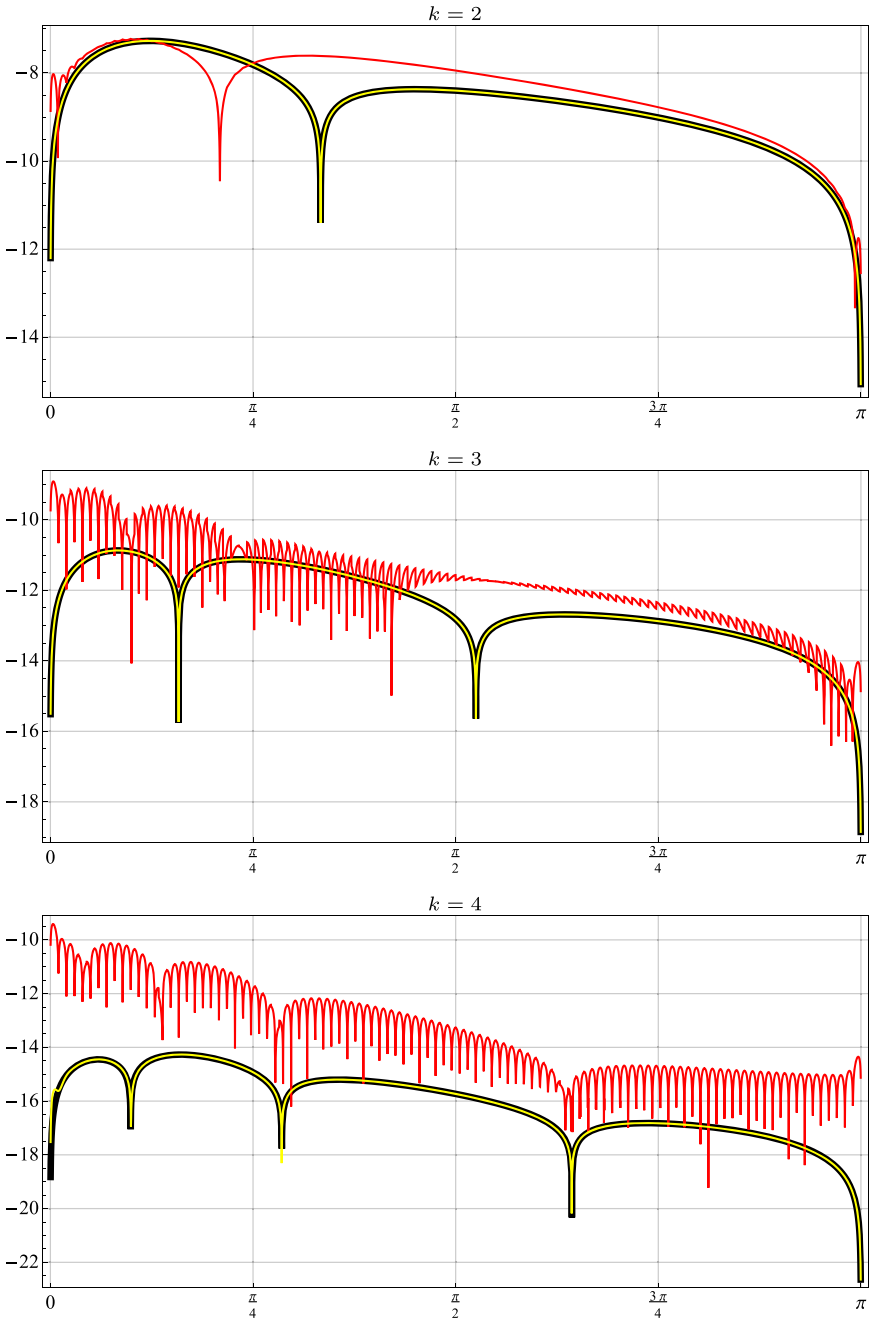

Fig. 3 Example 4.1: The base-10 logarithm for the individual errors $\varepsilon^{\mathrm{SL}}_{j,n,k}$ (thick black), $\varepsilon^{\mathrm{MNA}}_{j,n,k}$ (red), and $\varepsilon^{\mathrm{NAS}}_{j,n,k}$ (yellow) for the symbol f in (4.1) with $\rho = \frac{1}{2}$, a matrix size $n = 4096$, a grid size $n_1 = 100$, and different levels k. The proposed numerical algorithm (3.2) is represented by the yellow curve, while the exact asymptotic simple-loop expansion, by the black curve

Table 1 Example 4.1: The maximum errors $\varepsilon_{n,k}^{\mathrm{MNA}}$, $\varepsilon_{n,k}^{\mathrm{NAS}}$, and maximum normalized errors $(n+1)^k\varepsilon_{n,k}^{\mathrm{NAS}}$ for the levels $k = 1, 2, 3, 4$ and different matrix sizes n, corresponding to the symbol (4.1) with $\rho = \frac{1}{2}$. We used a grid of size $n_1 = 100$

n	256	512	1024	2048	4096
$\varepsilon_{n,1}^{\mathrm{MNA}}$	3.0897×10^{-3}	1.5494×10^{-3}	7.7577×10^{-4}	3.8816×10^{-4}	1.9415×10^{-4}
$\varepsilon_{n,1}^{\mathrm{NAS}}$	3.0897×10^{-3}	1.5494×10^{-3}	7.7577×10^{-4}	3.8816×10^{-4}	1.9415×10^{-4}
$(n+1)\varepsilon_{n,1}^{\mathrm{NAS}}$	7.9405×10^{-1}	7.9482×10^{-1}	7.9517×10^{-1}	7.9534×10^{-1}	7.9542×10^{-1}
$\varepsilon_{n,2}^{\mathrm{MNA}}$	1.5098×10^{-5}	3.7949×10^{-6}	9.5024×10^{-7}	2.3819×10^{-7}	5.9529×10^{-8}
$\varepsilon_{n,2}^{\mathrm{NAS}}$	1.3575×10^{-5}	3.4113×10^{-5}	8.5515×10^{-7}	2.1407×10^{-7}	5.3553×10^{-8}
$(n+1)^2\varepsilon_{n,2}^{\mathrm{NAS}}$	8.9661×10^{-1}	8.9775×10^{-1}	8.9844×10^{-1}	8.9875×10^{-1}	8.9890×10^{-1}
$\varepsilon_{n,3}^{\mathrm{MNA}}$	8.8167×10^{-8}	1.4780×10^{-8}	4.7324×10^{-9}	2.4238×10^{-9}	1.2270×10^{-9}
$\varepsilon_{n,3}^{\mathrm{NAS}}$	5.4356×10^{-8}	6.8619×10^{-9}	8.6153×10^{-10}	1.0794×10^{-10}	1.3507×10^{-11}
$(n+1)^3\varepsilon_{n,3}^{\mathrm{NAS}}$	9.2267×10^{-1}	9.2640×10^{-1}	9.2778×10^{-1}	9.2852×10^{-1}	9.2887×10^{-1}
$\varepsilon_{n,4}^{\mathrm{MNA}}$	6.6888×10^{-9}	3.1948×10^{-9}	1.5778×10^{-9}	7.8461×10^{-10}	3.8983×10^{-10}
$\varepsilon_{n,4}^{\mathrm{NAS}}$	3.4700×10^{-10}	2.1887×10^{-11}	1.3740×10^{-12}	8.6077×10^{-14}	5.4131×10^{-15}
$(n+1)^4\varepsilon_{n,4}^{\mathrm{NAS}}$	1.5138×10^{0}	1.5158×10^{0}	1.5166×10^{0}	1.5172×10^{0}	1.5252×10^{0}

It is clear that $f_\ell \in W^\alpha$ for any $\ell \in \mathbb{Z}_+$ and any $\alpha > 0$, but f_ℓ is simple-loop only when $\ell = 1$ because $f_\ell''(0) = 0$ for $\ell \neq 1$. Figures 4 and 5, and Tables 2 and 3, show the data for the cases $\ell = 2, 3$. Since our method is based on the simple-loop expansion (2.2), we expected difficulties for the very first eigenvalues, corresponding to the point $\theta = 0$. Nevertheless, the numerical approximations are good enough for machine precision purposes.

Figure 6 shows a comparison between the individual errors $\varepsilon_{j,n,k}^{\mathrm{NA}}$, given by the eigenvalue approximation of the numerical algorithm [1], and $\varepsilon_{j,n,k}^{\mathrm{MNA}}$, which corresponds to its boundary modification given by [10, Sc.4]. Then, it is clear that the modified version works better.

Example 4.3 (A matrix order dependent symbol) We now test our algorithm with a symbol which is a linear combination of RCTPs with coefficients depending on the matrix order n. Consider the symbol

$$F_n(\theta) \equiv f_2(\theta) + \alpha_1 f_1(\theta)h^2 + \alpha_0 f_0(\theta)h^4, \tag{4.3}$$

where f_ℓ is given by (4.2) and α_ℓ are real constants. We previously studied this symbol in [10, Sc.4.1] where we proposed an improvement to the numerical algorithm [5]. This symbol commonly arises when discretizing differential equations with a finite difference method.

For the respective boundary modification proposed in [10], we need to note that

$$\begin{aligned} c_2(0) &= \alpha_1 f_1(0) = 0, \quad & c_2(\pi) &= \alpha_1 f_1(\pi) = 4\alpha_1, \\ c_4(0) &= \alpha_0 f_0(0) = 0, \quad & c_4(\pi) &= \alpha_0 f_0(\pi) = \alpha_0, \end{aligned}$$

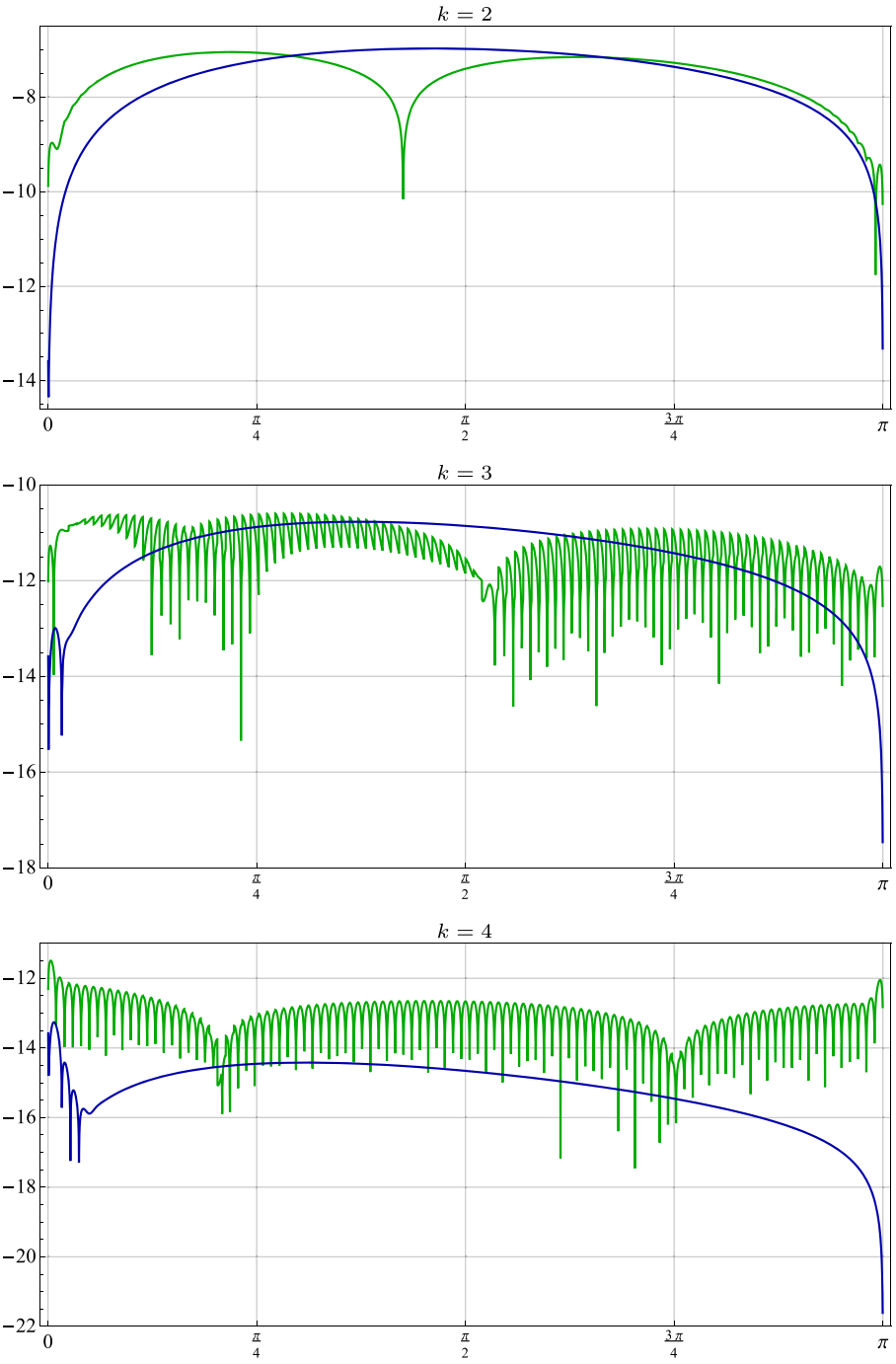

Fig. 4 Example 4.2: The base-10 logarithm for the individual errors $\varepsilon_{j,n,k}^{\mathrm{MNA}}$ (green) and $\varepsilon_{j,n,k}^{\mathrm{NAS}}$ (blue) for the RCTP symbol f_ℓ in (4.2) with $\ell = 2$, a matrix size $n = 4096$, a grid size $n_1 = 100$, and different levels k

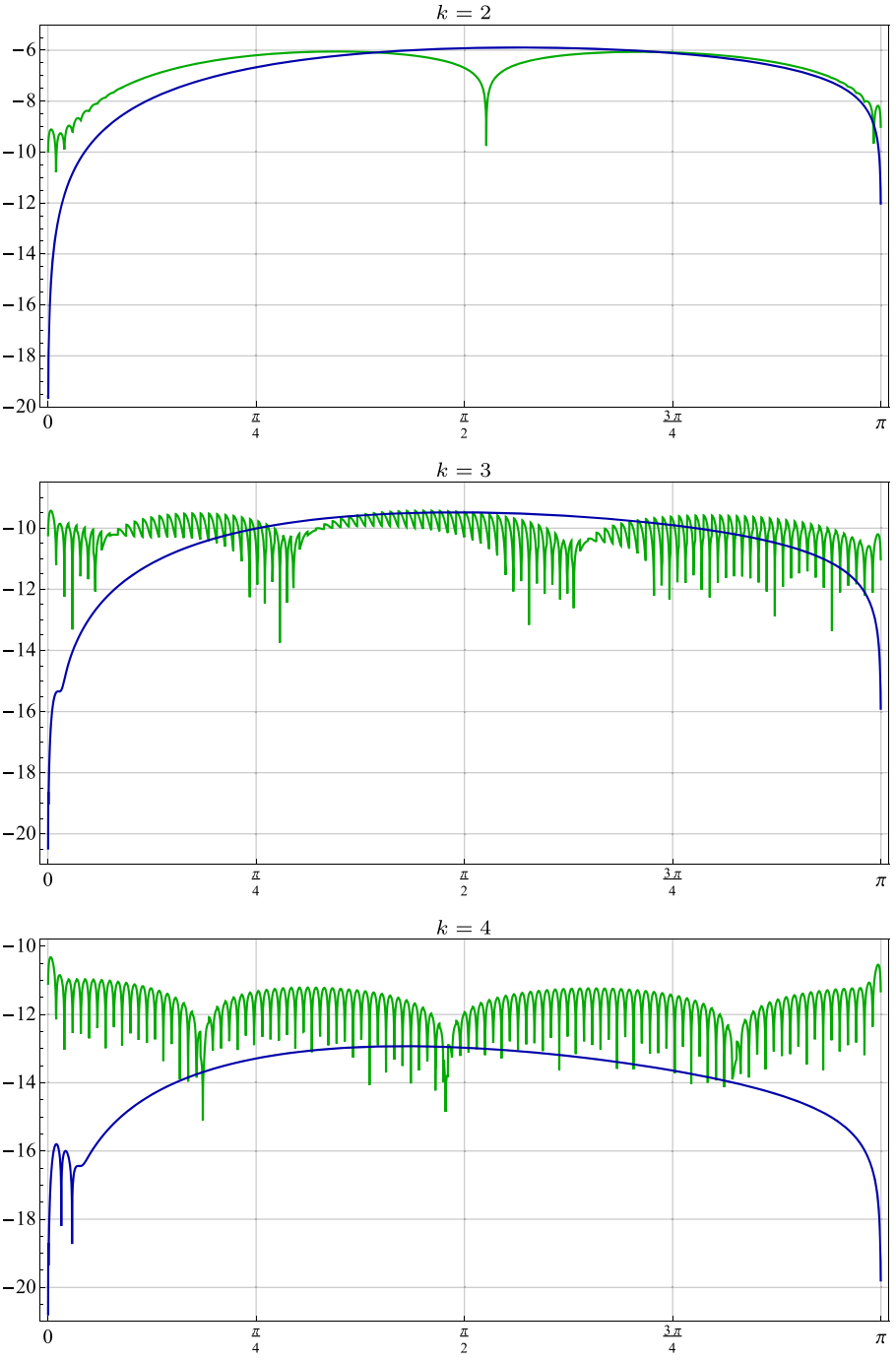

Fig. 5 Example 4.2: The same as Fig. 4 but this time with $\ell = 3$

Table 2 Example 4.2: The maximum errors $\varepsilon_{n,k}^{\mathrm{MNA}}$, $\varepsilon_{n,k}^{\mathrm{NAS}}$, and maximum normalized errors $(n+1)^k\varepsilon_{n,k}^{\mathrm{NAS}}$ for the levels $k = 1, 2, 3, 4$ and different matrix sizes n, corresponding to the symbol (4.2) with $\ell = 2$. We used a grid of size $n_1 = 100$

n	256	512	1024	2048	4096
$\varepsilon_{n,1}^{\mathrm{MNA}}$	1.6269×10^{-2}	8.1578×10^{-3}	4.0848×10^{-3}	2.0439×10^{-3}	1.0223×10^{-3}
$\varepsilon_{n,1}^{\mathrm{NAS}}$	1.6269×10^{-2}	8.1578×10^{-3}	4.0848×10^{-3}	2.0439×10^{-3}	1.0223×10^{-3}
$(n+1)\varepsilon_{n,1}^{\mathrm{NAS}}$	4.1811×10^{0}	4.1850×10^{0}	4.1869×10^{0}	4.1878×10^{0}	4.1883×10^{0}
$\varepsilon_{n,2}^{\mathrm{MNA}}$	2.2934×10^{-5}	5.7602×10^{-6}	1.4434×10^{-6}	3.6126×10^{-7}	9.0367×10^{-8}
$\varepsilon_{n,2}^{\mathrm{NAS}}$	2.7270×10^{-5}	6.8421×10^{-6}	1.7136×10^{-6}	4.2880×10^{-7}	1.0725×10^{-7}
$(n+1)^2\varepsilon_{n,2}^{\mathrm{NAS}}$	1.8011×10^{0}	1.8006×10^{0}	1.8004×10^{0}	1.8003×10^{0}	1.8002×10^{0}
$\varepsilon_{n,3}^{\mathrm{MNA}}$	7.5158×10^{-8}	9.4903×10^{-9}	1.2050×10^{-9}	1.5758×10^{-10}	2.5206×10^{-11}
$\varepsilon_{n,3}^{\mathrm{NAS}}$	6.9024×10^{-8}	8.6696×10^{-9}	1.0863×10^{-9}	1.3595×10^{-10}	1.7004×10^{-11}
$(n+1)^3\varepsilon_{n,3}^{\mathrm{NAS}}$	1.1717×10^{0}	1.1704×10^{0}	1.1698×10^{0}	1.1695×10^{0}	1.1694×10^{0}
$\varepsilon_{n,4}^{\mathrm{MNA}}$	3.7838×10^{-9}	2.2540×10^{-10}	1.6954×10^{-11}	5.9426×10^{-12}	3.2321×10^{-12}
$\varepsilon_{n,4}^{\mathrm{NAS}}$	2.7800×10^{-9}	1.3631×10^{-10}	7.4328×10^{-12}	4.5503×10^{-13}	5.4968×10^{-14}
$(n+1)^4\varepsilon_{n,4}^{\mathrm{NAS}}$	1.2128×10^{1}	9.4408×10^{0}	8.2044×10^{0}	8.2044×10^{0}	1.5487×10^{1}

while $c_\ell(0) = c_\ell(\pi) = 0$ in any other case. Figure 7 and Table 4 show the data.

The respective Fourier coefficients can be exactly calculated using the previous example and the linearity of the Fourier transform, as $\mathfrak{a}_k(F_n) = \alpha_2\mathfrak{a}_k(f_2) + \alpha_1\mathfrak{a}_k(f_1)h^2 + \alpha_0\mathfrak{a}_k(f_0)h^4$. Therefore, the Toeplitz matrices $T_n(F_n)$ are banded and

Table 3 Example 4.2: The same as Table 2 but this time with $\ell = 3$

n	256	512	1024	2048	4096
$\varepsilon_{n,1}^{\mathrm{MNA}}$	9.1868×10^{-2}	4.6172×10^{-2}	2.3146×10^{-2}	1.1588×10^{-2}	5.7978×10^{-3}
$\varepsilon_{n,1}^{\mathrm{NAS}}$	9.1868×10^{-2}	4.6172×10^{-2}	2.3146×10^{-2}	1.1588×10^{-2}	5.7978×10^{-3}
$(n+1)\varepsilon_{n,1}^{\mathrm{NAS}}$	2.3610×10^{1}	2.3686×10^{1}	2.3725×10^{1}	2.3744×10^{1}	2.3753×10^{1}
$\varepsilon_{n,2}^{\mathrm{MNA}}$	2.2588×10^{-4}	5.6759×10^{-5}	1.4227×10^{-5}	3.5615×10^{-6}	8.9091×10^{-7}
$\varepsilon_{n,2}^{\mathrm{NAS}}$	3.0497×10^{-4}	7.6550×10^{-5}	1.9176×10^{-5}	4.7989×10^{-6}	1.2003×10^{-6}
$(n+1)^2\varepsilon_{n,2}^{\mathrm{NAS}}$	2.0143×10^{1}	2.0146×10^{1}	2.0147×10^{1}	2.0148×10^{1}	2.0148×10^{1}
$\varepsilon_{n,3}^{\mathrm{MNA}}$	9.5995×10^{-7}	1.2158×10^{-7}	1.5664×10^{-8}	2.1683×10^{-9}	3.7673×10^{-10}
$\varepsilon_{n,3}^{\mathrm{NAS}}$	1.3355×10^{-6}	1.6765×10^{-7}	2.1002×10^{-8}	2.6281×10^{-9}	3.2868×10^{-10}
$(n+1)^3\varepsilon_{n,3}^{\mathrm{NAS}}$	2.2669×10^{1}	2.2634×10^{1}	2.2617×10^{1}	2.2608×10^{1}	2.2604×10^{1}
$\varepsilon_{n,4}^{\mathrm{MNA}}$	5.7200×10^{-9}	3.9578×10^{-10}	1.7663×10^{-10}	9.3710×10^{-11}	4.8060×10^{-11}
$\varepsilon_{n,4}^{\mathrm{NAS}}$	7.6467×10^{-9}	4.8020×10^{-10}	3.0083×10^{-11}	1.8824×10^{-12}	1.1772×10^{-13}
$(n+1)^4\varepsilon_{n,4}^{\mathrm{NAS}}$	3.3358×10^{1}	3.3258×10^{1}	3.3206×10^{1}	3.3181×10^{1}	3.3168×10^{1}

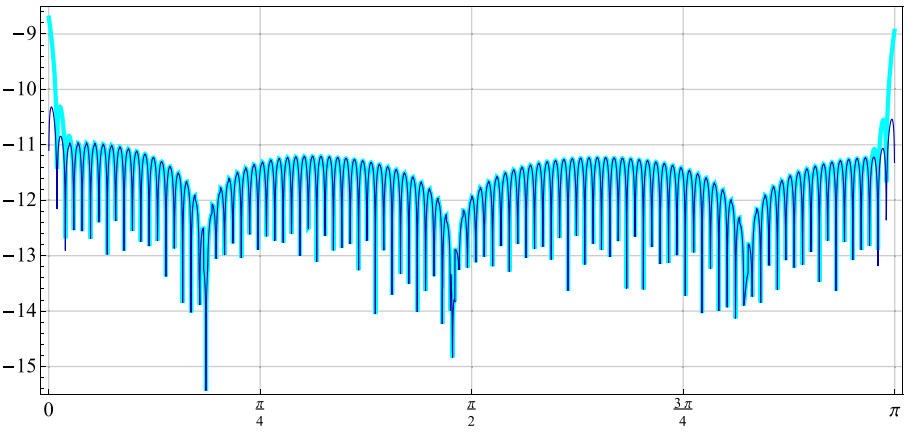

Fig. 6 Example 4.2: The base-10 logarithm for the individual errors $\varepsilon^{\mathrm{NA}}_{j,n,k}$ (thick cyan curve) and $\varepsilon^{\mathrm{MNA}}_{j,n,k}$ (thin blue curve), corresponding to the algorithms [1] and [10, Sc.4], respectively. We worked with the symbol (4.2) where $\ell = 3$, level $k = 4$, a matrix size $n = 4096$, and a grid size $n_1 = 100$

pentadiagonal. For implementing the numerical algorithm in [5], we need to assume an eigenvalue expansion with the form

$$\lambda_j(T_n(F_n)) = f_2(\theta_{j,n}) + \sum_{\ell=1}^{K} c_\ell(\theta_{j,n})h^\ell + E_{j,n,K},$$

where the coefficients c_ℓ are continuous and bounded functions from $[0, \pi]$ to $\mathbb{R}$, and the remainder (error) term $E_{j,n,K}$ satisfies the inequality $|E_{j,n,K}| \leqslant ch^{K+1}$.

Example 4.4 (Dense Toeplitz matrices) In the previous examples, the considered symbols produced banded Toeplitz matrices, that is, Toeplitz matrices with a finite number of non-zero diagonals, where this number is fixed and independent of the matrix order n. We now test our algorithm with a function generating a dense (non banded) Toeplitz matrix. Consider the symbol

$$f(\theta) \equiv |\theta|^\gamma, \quad \theta \in [-\pi, \pi], \tag{4.4}$$

where γ is a positive constant. We decided to study the cases $\gamma = 2$ and $\gamma = \frac{3}{2}$.

When $\gamma = 2$ the respective Fourier coefficients can be exactly calculated as $\mathfrak{a}_k(f) = \frac{1}{3}\pi^2$ if $k = 0$ and $\mathfrak{a}_k(f) = (-1)^k \frac{2}{k^2}$ in any other case. The function f does not belong to the simple-loop class because $f'(\pi) \neq 0$.

When $\gamma = \frac{3}{2}$, the respective Fourier coefficients can be exactly calculated as $\mathfrak{a}_k(f) = \frac{2}{5}\pi^{\frac{3}{2}}$ if $k = 0$ and $\mathfrak{a}_k(f) = \frac{3\sqrt{2}}{4\sqrt{\pi}\, j^{\frac{5}{2}}}((-1)^j\sqrt{2j} - C(\sqrt{2j}))$ where C is the well-known Fresnel integral given by

$$C(z) \equiv \int_0^z \cos\left(\frac{\pi}{2}t^2\right)\mathrm{d}t,$$

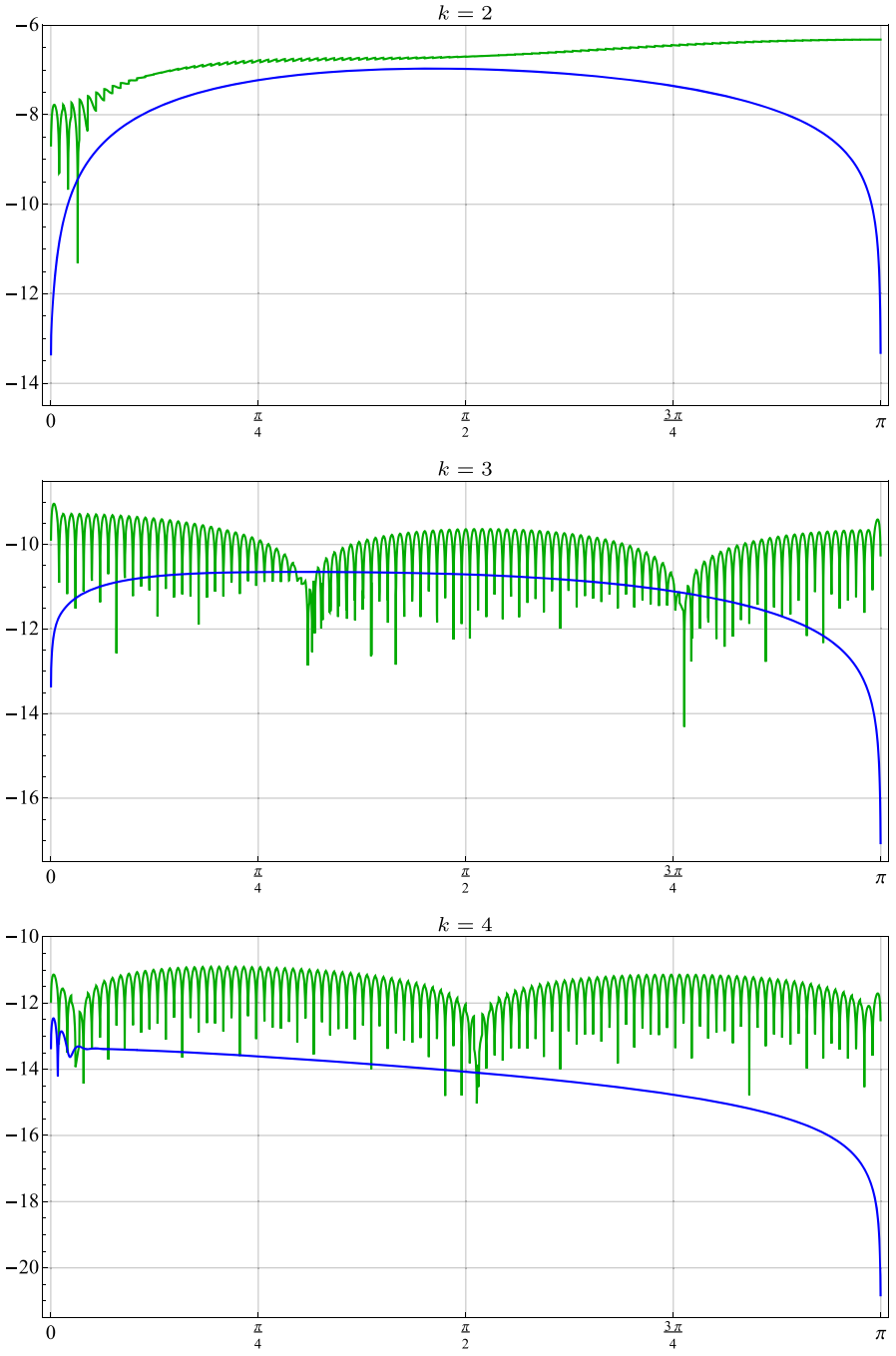

Fig. 7 Example 4.3: The base-10 logarithm for the individual errors $\varepsilon_{j,n,k}^{\mathrm{MNA}}$ (green) and $\varepsilon_{j,n,k}^{\mathrm{NAS}}$ (blue) for the matrix order dependent symbol F_n in (4.3), with $\alpha_0 = 3$, $\alpha_1 = 2$, a matrix size $n = 4096$, a grid size $n_1 = 100$, and different levels k

Table 4 Example 4.3: The maximum errors $\varepsilon_{n,k}^{\mathrm{MNA}}$, $\varepsilon_{n,k}^{\mathrm{NAS}}$, and maximum normalized errors $(n+1)^k\varepsilon_{n,k}^{\mathrm{NAS}}$ for the levels $k = 1, 2, 3, 4$ and different matrix sizes n, corresponding to the symbol (4.3) with $\alpha_0 = 3$ and $\alpha_1 = 2$. We used a grid of size $n_1 = 100$

n	256	512	1024	2048	4096
$\varepsilon_{n,1}^{\mathrm{MNA}}$	6.0007×10^{-3}	3.0088×10^{-3}	1.5065×10^{-3}	7.5377×10^{-4}	3.7702×10^{-4}
$\varepsilon_{n,1}^{\mathrm{NAS}}$	6.0007×10^{-3}	3.0088×10^{-3}	1.5065×10^{-3}	7.5377×10^{-4}	3.7702×10^{-4}
$(n+1)\varepsilon_{n,1}^{\mathrm{NAS}}$	1.5422×10^{0}	1.5435×10^{0}	1.5442×10^{0}	1.5445×10^{0}	1.5446×10^{0}
$\varepsilon_{n,2}^{\mathrm{MNA}}$	3.9869×10^{-5}	1.0043×10^{-5}	2.5287×10^{-6}	6.3851×10^{-7}	1.6273×10^{-7}
$\varepsilon_{n,2}^{\mathrm{NAS}}$	1.5208×10^{-5}	3.7944×10^{-6}	9.4766×10^{-7}	2.3679×10^{-7}	5.9184×10^{-8}
$(n+1)^2\varepsilon_{n,2}^{\mathrm{NAS}}$	1.0045×10^{0}	9.9858×10^{-1}	9.9563×10^{-1}	9.9416×10^{-1}	9.9342×10^{-1}
$\varepsilon_{n,3}^{\mathrm{MNA}}$	9.2045×10^{-8}	1.0710×10^{-8}	3.6629×10^{-9}	1.8241×10^{-9}	9.1977×10^{-10}
$\varepsilon_{n,3}^{\mathrm{NAS}}$	8.9731×10^{-8}	1.1313×10^{-8}	1.4203×10^{-9}	1.7792×10^{-10}	2.2264×10^{-11}
$(n+1)^3\varepsilon_{n,3}^{\mathrm{NAS}}$	1.5231×10^{0}	1.5274×10^{0}	1.5295×10^{0}	1.5306×10^{0}	1.5311×10^{0}
$\varepsilon_{n,4}^{\mathrm{MNA}}$	3.4333×10^{-9}	2.1389×10^{-10}	5.3710×10^{-11}	2.5589×10^{-11}	1.2650×10^{-11}
$\varepsilon_{n,4}^{\mathrm{NAS}}$	4.3281×10^{-9}	2.7008×10^{-10}	1.8110×10^{-11}	2.3324×10^{-12}	2.9853×10^{-13}
$(n+1)^4\varepsilon_{n,4}^{\mathrm{NAS}}$	1.8881×10^{1}	1.8705×10^{1}	1.9990×10^{1}	4.1112×10^{1}	8.4112×10^{1}

and included as an inner function in software programs like MATHEMATICA. In this case, the function f fails to belong to the simple-loop class also because $f''(0)$ is undefined and $f'(\pi) \neq 0$.

Hence, for $\gamma = 2$ we expect troubles for the eigenvalues with $\theta_{j,n}$ near to $\theta = \pi$ while for $\gamma = \frac{3}{2}$ we expect troubles for the eigenvalues with $\theta_{j,n}$ near to both $\theta = 0$ and $\theta = \pi$. To show the algorithm accuracy for the inner eigenvalues only, take an small $\delta > 0$ and consider the eigenvalues such that $\delta < \theta_{j,n} < \pi - \delta$ (or equivalently, $\frac{\delta(n+1)}{\pi} < j < \frac{(\pi-\delta)(n+1)}{\pi}$). Then, the inner errors are given by:

$$\hat{\varepsilon}_{n,k}^{\mathrm{SL}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{SL}} : \delta < \theta_{j,n} < \pi - \delta\}, \hat{\varepsilon}_{n,k}^{\mathrm{NA}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{NA}} : \delta < \theta_{j,n} < \pi - \delta\},$$
$$\hat{\varepsilon}_{n,k}^{\mathrm{MNA}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{MNA}} : \delta < \theta_{j,n} < \pi - \delta\}, \hat{\varepsilon}_{n,k}^{\mathrm{NAS}} \equiv \max\{\varepsilon_{j,n,k}^{\mathrm{NAS}} : \delta < \theta_{j,n} < \pi - \delta\}.$$

Figures 8 and 9, and Tables 5 and 6 show the data. In this case, the numerical algorithms [10, Sc.4] and (3.2) produced similar results. For a matrix of size $n = 4096$ a standard eigensolver can produce 11 precision digits while our algorithm was able to produce up to 10 precision digits for the inner eigenvalues and up to 6 precision digits for the extreme ones, in the level 3, and always with linear

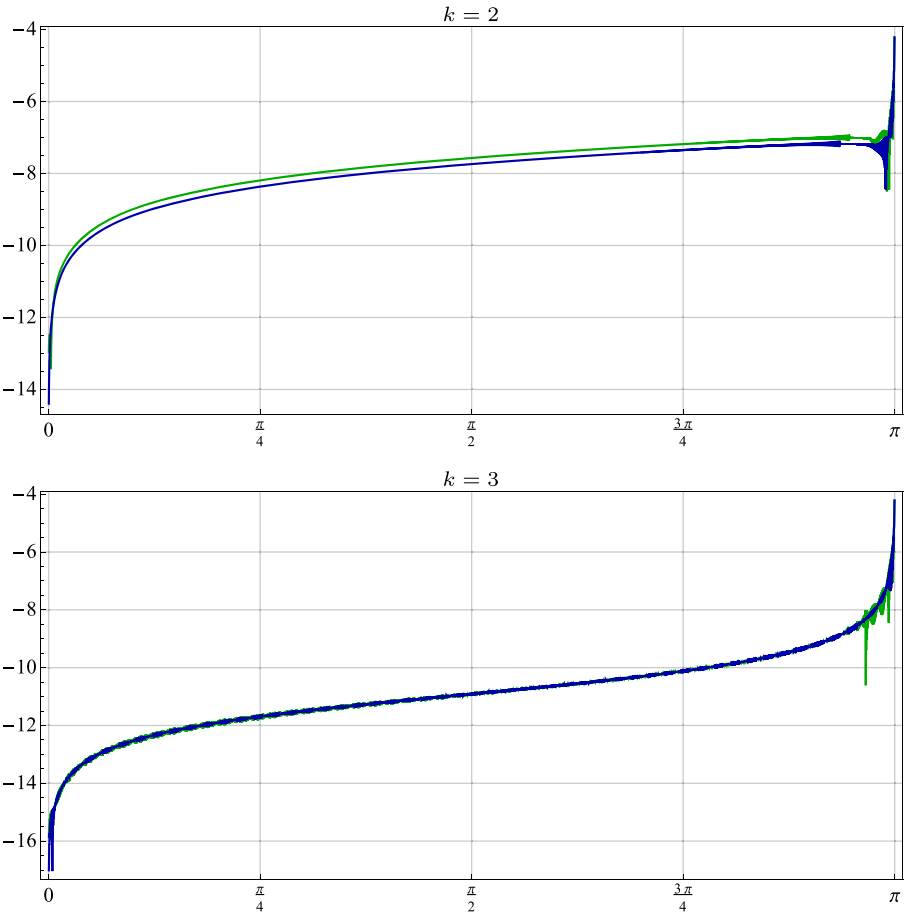

Fig. 8 Example 4.4: The base-10 logarithm for the individual errors $\varepsilon_{j,n,k}^{\mathrm{MNA}}$ (green) and $\varepsilon_{j,n,k}^{\mathrm{NAS}}$ (blue) for the symbol f in (4.4) with $\gamma = 2$, a matrix size $n = 4096$, a grid size $n_1 = 100$, and different levels k

computational cost. At first sight it can look unsatisfying but for a matrix of a serious size, let us say $n = 10^{10}$, a simple calculation shows that our algorithm will produce 30 precision digits for the inner eigenvalues and 26 for the extreme ones, while the standard calculation can only be done by the best super-computers in the world.

On the other hand, Tables 5 and 6 show that the inner eigenvalue errors have a nice behavior until level 3. But the level 4 produced no better results, then we conjecture that, for the symbol (4.4), the asymptotic expansion (2.2) is true until level 3 only.

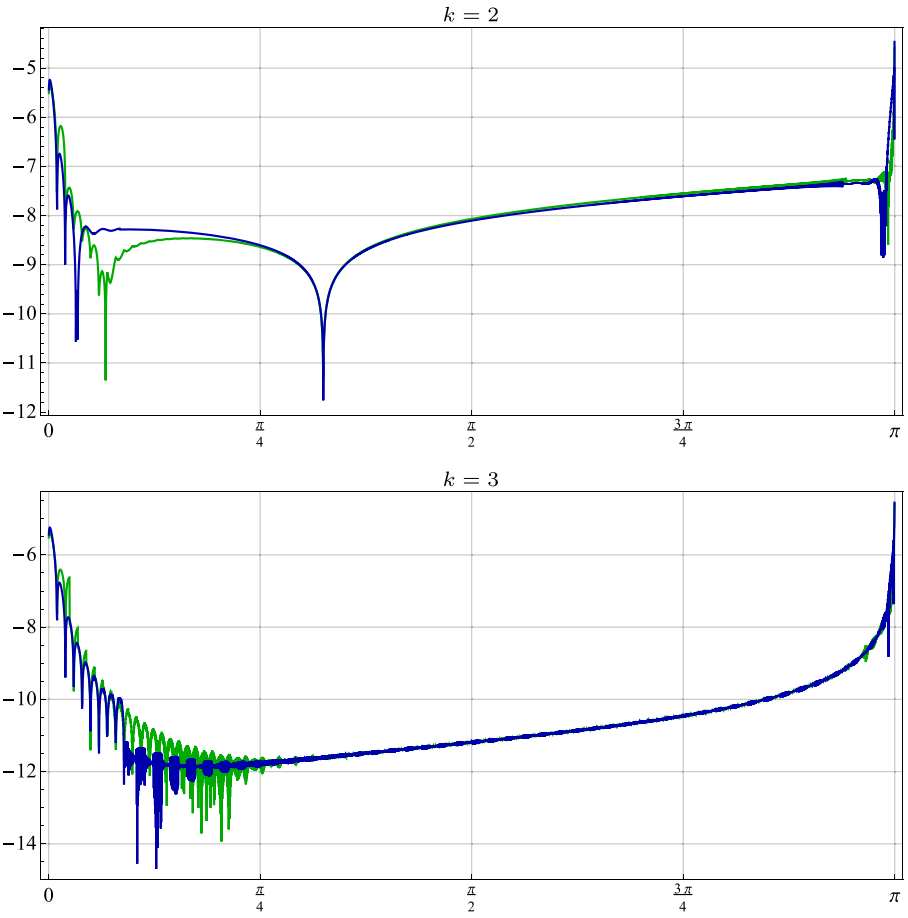

Fig. 9 Example 4.4: The same as Fig. 8 but with $\gamma = \frac{3}{2}$

Table 5 Example 4.4: The maximum inner errors $\hat{\varepsilon}_{n,k}^{\mathrm{MNA}}$, $\hat{\varepsilon}_{n,k}^{\mathrm{NAS}}$, for $\delta = \frac{1}{100}$, and maximum normalized inner errors $(n+1)^k \hat{\varepsilon}_{n,k}^{\mathrm{NAS}}$ for the levels $k = 1, 2, 3$ and different matrix sizes n, corresponding to the symbol (4.4) with $\gamma = 2$. We used a grid of size $n_1 = 100$

n	256	512	1024	2048	4096
$\hat{\varepsilon}_{n,1}^{\mathrm{MNA}}$	4.6648×10^{-3}	2.3445×10^{-3}	1.1753×10^{-3}	5.8841×10^{-4}	2.9439×10^{-4}
$\hat{\varepsilon}_{n,1}^{\mathrm{MNA}}$	4.6648×10^{-3}	2.3445×10^{-3}	1.1753×10^{-3}	5.8841×10^{-4}	2.9439×10^{-4}
$(n+1)\hat{\varepsilon}_{n,1}^{\mathrm{NAS}}$	1.1988×10^{0}	1.2027×10^{0}	1.2047×10^{0}	1.2056×10^{0}	1.2061×10^{0}
$\hat{\varepsilon}_{n,2}^{\mathrm{MNA}}$	6.8707×10^{-6}	1.7243×10^{-6}	4.3191×10^{-7}	1.0808×10^{-7}	2.7034×10^{-8}
$\hat{\varepsilon}_{n,2}^{\mathrm{NAS}}$	4.6552×10^{-6}	1.1661×10^{-6}	2.9179×10^{-7}	7.2981×10^{-8}	1.8251×10^{-8}
$(n+1)^2\hat{\varepsilon}_{n,2}^{\mathrm{NAS}}$	3.0747×10^{-1}	3.0687×10^{-1}	3.0656×10^{-1}	3.0640×10^{-1}	3.0634×10^{-1}
$\hat{\varepsilon}_{n,3}^{\mathrm{MNA}}$	5.6644×10^{-8}	7.1360×10^{-9}	8.8605×10^{-10}	1.0960×10^{-10}	1.4282×10^{-11}
$\hat{\varepsilon}_{n,3}^{\mathrm{NAS}}$	5.2043×10^{-8}	6.5179×10^{-9}	7.9855×10^{-10}	1.0260×10^{-10}	1.3227×10^{-11}
$(n+1)^3\hat{\varepsilon}_{n,3}^{\mathrm{NAS}}$	8.8340×10^{-1}	8.7995×10^{-1}	8.5995×10^{-1}	8.8258×10^{-1}	9.0963×10^{-1}

Table 6 Example 4.4: The same as Table 5 but with $\gamma = \frac{3}{2}$

n	256	512	1024	2048	4096
$\hat{\varepsilon}_{n,1}^{\mathrm{MNA}}$	5.1538×10^{-3}	2.5910×10^{-3}	1.2991×10^{-3}	6.5043×10^{-4}	3.2544×10^{-4}
$\hat{\varepsilon}_{n,1}^{\mathrm{MNA}}$	5.1538×10^{-3}	2.5910×10^{-3}	1.2991×10^{-3}	6.5043×10^{-4}	3.2544×10^{-4}
$(n+1)\hat{\varepsilon}_{n,1}^{\mathrm{NAS}}$	1.3245×10^{0}	1.3292×10^{0}	1.3315×10^{0}	1.3327×10^{0}	1.3333×10^{0}
$\hat{\varepsilon}_{n,2}^{\mathrm{MNA}}$	9.0160×10^{-6}	2.2422×10^{-6}	5.5893×10^{-7}	1.3953×10^{-7}	3.4863×10^{-8}
$\hat{\varepsilon}_{n,2}^{\mathrm{NAS}}$	7.9282×10^{-6}	1.9676×10^{-6}	4.8992×10^{-7}	1.2224×10^{-7}	3.0535×10^{-8}
$(n+1)^2\hat{\varepsilon}_{n,2}^{\mathrm{NAS}}$	5.2365×10^{-1}	5.1780×10^{-1}	5.1472×10^{-1}	5.1319×10^{-1}	5.1254×10^{-1}
$\hat{\varepsilon}_{n,3}^{\mathrm{MNA}}$	2.5715×10^{-7}	3.2350×10^{-8}	4.1645×10^{-9}	5.6318×10^{-10}	7.1117×10^{-11}
$\hat{\varepsilon}_{n,3}^{\mathrm{NAS}}$	2.5294×10^{-7}	3.1820×10^{-8}	4.2554×10^{-9}	5.7209×10^{-10}	7.2327×10^{-11}
$(n+1)^3\hat{\varepsilon}_{n,3}^{\mathrm{NAS}}$	4.2936×10^{0}	4.2958×10^{0}	4.5826×10^{0}	4.9214×10^{0}	4.9739×10^{0}

5 Conclusions

Under appropriate technical assumptions, the simple-loop theory allows to derive various types of asymptotic expansions for the eigenvalues of Toeplitz matrices $T_n(f)$ generated by a function f. Independently and under the milder hypothesis that f is even and monotonic over $[0, \pi]$, matrix-less algorithms have been developed for the fast eigenvalue computation of large Toeplitz matrices. These procedures work with a linear complexity in the matrix order n and behind the high efficiency of such algorithms there are the expansions predicted by the simple-loop theory, combined with the extrapolation idea.

In this article, we have focused our attention on a change of variable, followed by the asymptotic expansion of the new variable, that is,

$$\lambda_j(T_n(f)) \equiv f(s_{j,n}), \qquad s_{j,n} = \theta_{j,n} + \sum_{k=1}^{K} r_k(\theta_{j,n})h^k + E_{j,n,K},$$

and then we adapted the matrix-less to the resulting new setting. Numerical experiments have shown in a clear way a higher precision (till machine precision) and the same linear computation cost, when compared with the matrix-less procedures already presented in the relevant literature.

More specifically, among the advantages, we concisely mention the following:

1. when the coefficients of the simple-loop function are analytically known, the algorithm computes them perfectly;
2. while the proposed algorithm is better or at worst comparable to the previous ones for the computation of the inner eigenvalues, it is extremely better for the computation of the extreme eigenvalues, which are essential for determining important quantities, like the conditioning in the positive definite case.

As next steps, the following questions remain to be investigated:

- a fine error analysis for having a theoretical explanation of the reason why the new expansion leads to a much smaller errors, when compared with the numerical results in [1, 3–5];
- applications to the block cases (see [11, 12] for the theory in the block case) and related applications [3, 4] to differential problems;
- taking inspiration from [2], extension of the technique to the preconditioned case $X_n = T_n^{-1}(g)T_n(l)$ in which g is positive over $(0, \pi)$ and is not identically constant, making use of the ergodic theorems given in [26], where Theorem 2.1 is extended to the preconditioned case in a very general setting.

Acknowledgements We acknowledge the reviewers for their careful reading and insightful suggestions that improved our article. Part of the numerical experiments were calculated in the computer center Jürgen Tischer of the mathematics department at Universidad del Valle.

Funding The first author was partially supported by Universidad del Valle. The third author is partially supported by INdAM.GNCS.

Data availability Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Ekström, S.-E., Garoni, C., Serra-Capizzano, S.: Are the eigenvalues of banded symmetric Toeplitz matrices known in almost closed form? Exper. Math. **27**(4), 478–487 (2018)
2. Ahmad, F., Al–Aidarous, E.S., Alrehaili, 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. Algo. **78**(3), 867–893 (2018)
3. Ekström, S.-E., Furci, I., Garoni, C., Manni, C., Serra-Capizzano, S., Speleers, H.: Are the eigenvalues of the B-spline isogeometric analysis approximation of $-\Delta u = \lambda u$ known in almost closed form? Numer. Linear Algebra Appl. **25**(5), 2198–34 (2018)
4. Ekström, S.-E., Furci, I., Serra-Capizzano, S.: Exact formulae and matrix-less eigensolvers for block banded Toeplitz-like matrices. BIT **58**(4), 937–968 (2018)
5. Ekström, S.-E., Garoni, C.: A matrix-less and parallel interpolation-extrapolation algorithm for computing the eigenvalues of preconditioned banded symmetric Toeplitz matrices. Numer. Algor. **80**, 819–848 (2019)
6. Bogoya, 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)
7. Bogoya, M., Böttcher, A., Grudsky, S.M., Maximenko, E.A.: Eigenvectors of Hermitian Toeplitz matrices with smooth simple-loop symbols. Linear Algebra Appl. **493**, 606–637 (2016)
8. Böttcher, A., Bogoya, M., Grudsky, S.M., Maksimenko, E.A.: Asymptotics of the eigenvalues and eigenvectors of Toeplitz matrices. Mat. Sb. **208**(11), 4–28 (2017)
9. Barrera, M., Böttcher, A., Grudsky, S.M., Maximenko, E.A.: Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic. Oper. Theory Adv. Appl. **268**, 51–77 (2018)

10. Bogoya, M., Serra-Capizzano, S.: Eigenvalue superposition expansion for Toeplitz matrix-sequences, generated by linear combinations of matrix-order dependent symbols, and applications to fast eigenvalue computations. arXiv:2112.11794 (2022)
11. Barbarino, G., Garoni, C., Serra-Capizzano, S.: Block generalized locally Toeplitz sequences: Theory and applications in the multidimensional case. Electron. Trans. Numer. Anal. **53**, 113–216 (2020)
12. Barbarino, G., Garoni, C., Serra-Capizzano, S.: Block generalized locally Toeplitz sequences: theory and applications in the unidimensional case. Electron. Trans. Numer. Anal. **53**, 28–112 (2020)
13. Böttcher, A., Silbermann, B.: Introduction to large truncated Toeplitz matrices. Universitext, p. 258. Springer, Berlin (1999)
14. Garoni, C., Serra-Capizzano, S.: Generalized locally Toeplitz sequences: theory and applications, vol. I. Springer, Berlin (2017)
15. Garoni, C., Serra-Capizzano, S.: Generalized locally Toeplitz sequences: theory and applications, vol. II. Springer, Berlin (2018)
16. Grenander, U., Szegő, G. Toeplitz forms and their applications. California Monographs in Mathematical Sciences, 2nd edn. Chelsea Publishing Co, New York (1984)
17. Tyrtyshnikov, E.E., Zamarashkin, N.L.: Spectra of multilevel Toeplitz matrices: advanced theory via simple matrix relationships. Linear Algebra Appl. **270**, 15–27 (1998)
18. Serra-Capizzano, S.: The extension of the concept of the generating function to a class of preconditioned Toeplitz matrices. Linear Algebra Appl. **267**, 139–161 (1997)
19. Bogoya, M., Grudsky, S.M., Maximenko, E.A.: Eigenvalues of Hermitian Toeplitz matrices generated by simple-loop symbols with relaxed smoothness. Oper. Theory Adv. Appl. **259**, 179–212 (2017)
20. Stoer, J., Bulirsch, R. Introduction to Numerical Analysis, 3rd edn. Springer, Berlin (2010)
21. Davis, P.J.: Interpolation and approximation. Dover, New York (1975)
22. Kac, M., Murdock, W.L., Szegő, G.: On the eigenvalues of certain Hermitian forms. J. Rational Mech. Anal. **2**, 767–800 (1953)
23. Trench, W.F.: Asymptotic distribution of the spectra of a class of generalized kac–Murdock–Szegő matrices. Linear Algebra Appl. **294**, 181–192 (1999)
24. Trench, W.F.: Spectral decomposition of Kac–Murdock–Szegő matrices. The selected works of William F. Trench. http://works.bepress.com/william_trench/133 (2010)
25. Barrera, M., Grudsky, S.M.: Asymptotics of eigenvalues for pentadiagonal symmetric Toeplitz matrices. Oper. Theory Adv. Appl. **259**, 51–77 (2017)
26. Serra-Capizzano, S.: An ergodic theorem for classes of preconditioned matrices. Linear Algebra Appl. **282**(1-3), 161–183 (1998)

Affiliations

Manuel Bogoya[1,2] · Sven-Erik Ekström[3] · Stefano Serra-Capizzano[2,3]

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

Stefano Serra-Capizzano
s.serracapizzano@uninsubria.it

1 Departamento de Matemáticas, Universidad del Valle, Calle 13 No.100-00, Cali, Colombia

2 Dipartimento di Scienza e Alta Tecnologia, University of Insubria, via Valleggio, 11, Como, 22100, Italy

3 Department of Information Technology, Division of Scientific Computing, Uppsala University, Box 337, SE-751 05, Uppsala, Sweden