UPPSALA
UNIVERSITET

Symbol-based Spectral Analysis of Discretisations of the Variable Coefficient Diffusion Equation

Authors: Melker Claesson, David Meadon
Supervisor: Sven-Erik Ekström

Project in Computational Science: Report

January 2021

PROJECT REPORT

Institutionen för informationsteknologi

Abstract

The matrix-less method can be used to efficiently approximate the eigenvalues of certain classes of matrices. Specifically, the method has thus far been used to approximate the eigenvalues of Toeplitz and Toeplitz-like matrices where it uses the fact that a function, the so-called symbol $f(\theta)$ (and higher order symbol), of these matrices contains information about their eigenvalues.

In this project, we numerically investigate whether this method also works for matrices which result from discretising problems with variable coefficients, where we do not have constant, or almost constant, diagonals in the matrices but rather the diagonals are determined by the sampling of a function $a(x)$; the spectral symbol is then $f(x, \theta) = a(x)g(\theta)$. Two other matrix sequences, which are spectrally related to the original discretisation matrix, are also studied; these matrices share the same symbol as the original matrix but have different higher order symbols.

The numerical results show that the matrix-less method is able to well-approximate the eigenvalues for matrices generated with several different functions $a(x)$. Also, we identify examples where parts, or the whole spectrum, behaves as if the expansion used for the matrix-less method should be modified, but that it then probably will work. Furthermore, we find surprising similarity of the spectrum, even in higher order symbols, of the original discretisation matrix and one of the two alternative matrices. We conclude the report with suggestions for future avenues of research.

Contents

1 Introduction **2**

2 Background and Problem Setting **3**
2.1 Toeplitz Matrices 3
2.2 The Matrix-less Method 5
2.3 Problem Setting 7

3 Numerical Experiments **10**
3.1 Constant and Linear Examples 11
3.2 Laplace-like Example 16
3.3 Non-Monotonic Example 18
3.4 Non-Smooth Example 20
3.5 Discontinuous Example 23

4 Conclusions **26**

References **28**

Appendix A Code **30**

Appendix B Matrix Symmetrisation **32**

1 Introduction

Eigenvalue problems are an important topic in many scientific fields such as engineering, physics and mathematics. In many applications, the arising matrices may be very large, meaning that it would take traditional methods a considerable amount of time to calculate their spectrum. The matrix-less method, however, give us an efficient way to approximate these eigenvalues but only for certain classes of structured matrices. When discretising the diffusion equation, where a variable diffusion coefficient is used, the resulting matrix is not of the currently required form for using the matrix-less method. We are interested in the spectrum of these matrices since the spectral properties of this discretisation matrix affect both the stability, accuracy and convergence speed of solving the system. For stability, it is required that all the eigenvalues of the discretisation matrix, A_n, have non-positive real part, where a negative real part indicates diffusion in the solution. Moreover we can look to the condition number of the matrix, $\kappa(A_n) = \frac{|\lambda_{\max}(A_n)|}{|\lambda_{\min}(A_n)|}$, to evaluate the convergence rate of an iterative method as well as to construct preconditioners which lower the condition number of the matrix. Thus it would be beneficial to be able to efficiently approximate the spectrum of the discretisation matrix using the matrix-less method, not only for the current case with Toeplitz matrices, but slightly more generally for discretisations of differential equations with variable coefficients.

Our goal in this report is to numerically investigate how the matrix-less method can be applied to matrices which result from variable coefficients where the function $a(x)$ defining the variable coefficients is of a number of different forms.

In Section 2, we begin by first defining some basics on Toeplitz matrices and the basics of the matrix-less method. We then define the differential equation as well as the resulting matrix whose spectrum will be approximated in this paper. We also define two other matrices which are spectrally related to the main discretisation matrix, and which should yield identical results in an infinite case but differ in the finite one. Section 3 then contains the bulk of the report, where a number of different functions $a(x)$ defining the variable coefficients are tested. We end the report in Section 4 with our conclusions and suggestions of topics for future research.

2 Background and Problem Setting

In this section we will detail the differential equation and specifically the resulting matrices which will be of main interest for this report. We will first begin with a brief discussion on Toeplitz matrices, and then introduce the matrix-less method, before the main problem of interest of the report.

2.1 Toeplitz Matrices

A Toeplitz matrix $A_n \in \mathbb{C}^{n\times n}$ is a matrix with constant diagonals [5, pp. 95–96]:

$$A_n = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-(n-1)} \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{-1} \\ a_{n-1} & \cdots & a_1 & a_0 \end{bmatrix}. \tag{2.1}$$

For each Toeplitz matrix A_n we can associate a function $f(\theta)$, called a symbol. The symbol $f \in L^1(-\pi, \pi)$ generates a sequence of matrices $\{A_n\}_n$ of increasing size n [5, p. 4]. In particular, we have the generated matrix

$$T_n(f) = \left[\hat{f}_{i-j}\right]_{i,j=1}^{n} = \begin{bmatrix} \hat{f}_0 & \hat{f}_{-1} & \cdots & \hat{f}_{-(n-1)} \\ \hat{f}_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \hat{f}_{-1} \\ \hat{f}_{n-1} & \cdots & \hat{f}_1 & \hat{f}_0 \end{bmatrix}, \tag{2.2}$$

with the elements $\hat{f}_k \in \mathbb{C}$ being the Fourier-coefficients

$$\hat{f}_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)e^{-\mathrm{i}k\theta}\mathrm{d}\theta,\ k \in \mathbb{Z}. \tag{2.3}$$

The spectral behaviour of these matrix sequences can be described by the theory of GLT sequences; see [5]. If $\{T_n(f)\}_n$ is a GLT sequence, denoted

by $\{T_n(f)\}_n \sim_{\text{GLT}} f$, then the singular values (except possibly $o(n)$ outliers) $\sigma_j(T_n(f))$, can be approximated by $|f(\theta_{j,n})|$ where $\theta_{j,n}$ is an equispaced grid in $[-\pi, \pi]$. That is,

$$\sigma_j(T_n) = |f(\theta_{j,n}| + E_{j,n,0}, \tag{2.4}$$

where $E_{j,n,0} = \mathcal{O}(h)$ is an error term.

If f is real-valued, which means that $T_n(f)$ is a Hermitian Toeplitz matrix, then we say that $\{T_n(f)\}_n \sim_\lambda f$ and the eigenvalues $\lambda_j(T_n(f))$ can be approximated by $f(\theta_{j,n})$. If f is even, then we can choose a grid defined on $[0, \pi]$. Hence,

$$\lambda_j(T_n(f)) = f(\theta_{j,n}) + E_{j,n,0}, \tag{2.5}$$

again where the error term is $E_{j,n,0} = \mathcal{O}(h)$. A very useful standard grid used throughout this report is

$$\theta_{j,n} = \frac{j\pi}{n+1}. \tag{2.6}$$

A particularly useful Toeplitz matrix is the discrete Laplacian matrix,

$$T_n(f) = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad f(\theta) = 2 - 2\cos(\theta) \tag{2.7}$$

that is seen when discretising a differential equation which includes the Laplace operator using second order central finite differences [1, p. 572].

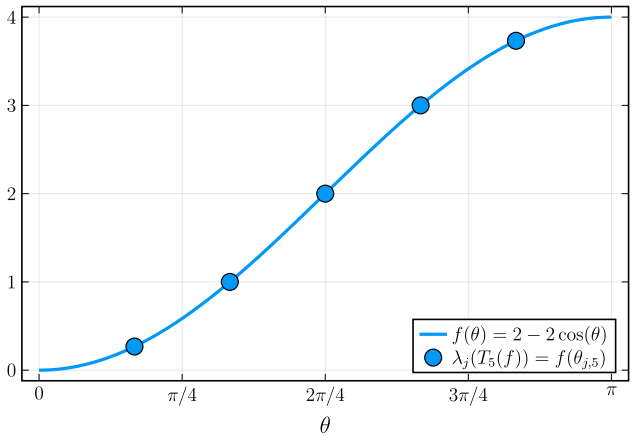

Figure 2.1: Symbol $f(\theta) = 2 - 2\cos(\theta)$ (blue line) of the discrete Laplacian matrix. The blue dots represent the eigenvalues of $T_n(f)$ for $n = 5$, i.e., $\lambda_j(T_5(f)) = f(\theta_{j,5})$ where $\theta_{j,5} = j\pi/6$, $j = 1, \ldots, 5$.

Using the symbol in (2.7) and sampling it using the grid $\theta_{j,n} = \frac{j\pi}{n+1}$ will yield the exact eigenvalues $\lambda_j(T_n(f)) = f(\theta_{j,n})$, as in Figure 2.1, of its respective

matrix $T_n(f)$; see, e.g., [5]. In general, this grid $\theta_{j,n}$ can be used for all symbols to approximate the eigenvalues, but sometimes other grids are preferred since they give smaller approximation errors; see, e.g., [9].

2.2 The Matrix-less Method

A class of methods, denoted matrix-less, were first introduced in [4] and has been extended to a big class of Hermitian matrices, see e.g. [7] and lately also to non-Hermitian matrices [11, 13]. The GLT eigenvalue approximation

$$\lambda_j(A_n) = f(\theta_{j,n}) + E_{j,n,0} \approx f(\theta_{j,n}) \tag{2.8}$$

works when we can find or define the symbol $f(\theta)$ for the sequence $\{A_n\}_n$, but the error $E_{j,n,0} = \mathcal{O}(h)$ might be prohibitively large for the application.

In the matrix-less method we exploit an asymptotic expansion of the form

$$\begin{aligned} \lambda_j(A_n) &= f(\theta_{j,n}) + E_{j,n,0} \\ &= f(\theta_{j,n}) + \sum_{k=1}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha} \\ &= \sum_{k=0}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha}, \end{aligned} \tag{2.9}$$

where $f(\theta) = c_0(\theta)$ and $\alpha \in \mathbb{Z}_+$ is chosen by the user. The resulting error is $E_{j,n,\alpha} = \mathcal{O}(h^{\alpha+1})$.

The matrix-less method approximates the functions $c_k(\theta)$ by samplings $\tilde{c}_k(\theta_{j,n_0})$, where $n_0 \in \mathbb{Z}_+$ is chosen by the user.

The idea of the method is to calculate the spectrum for $\alpha + 1$ smaller matrices using standard numerical solvers, approximate the functions c_k for $k = 0, \ldots, \alpha$, and then use these to approximate the spectrum for a matrix A_n where $n \gg n_0$. In practice this is done by computing the matrix

$$\widetilde{C} = \begin{bmatrix} \tilde{c}_0(\theta_{1,n_0}) & \tilde{c}_0(\theta_{2,n_0}) & \cdots & \tilde{c}_0(\theta_{n_0,n_0}) \\ \tilde{c}_1(\theta_{1,n_0}) & \tilde{c}_1(\theta_{2,n_0}) & \cdots & \tilde{c}_1(\theta_{n_0,n_0}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_\alpha(\theta_{1,n_0}) & \tilde{c}_\alpha(\theta_{2,n_0}) & \cdots & \tilde{c}_\alpha(\theta_{n_0,n_0}) \end{bmatrix}, \tag{2.10}$$

and then, by an interpolation–extrapolation scheme [12], approximate the corresponding matrix for $\theta_{j,n}$ instead of θ_{j,n_0}, and then compute $\lambda_j(A_n) \approx \tilde{\lambda}_{j,n} = \sum_{k=0}^{n} h^k \tilde{c}_k(\theta_{j,n})$.

First the matrices A_{n_k} of sizes $n_k = (n_0 + 1)2^k - 1$ for $k = 0, \ldots, \alpha$ are constructed and their eigenvalues are computed using a standard numerical solver; we use Julia's `eigvals` [15] mainly due to the support of high precision floats. After sorting the spectrum for each level k, we choose every 2^k-th eigenvalue to

construct a matrix $E \in \mathbb{C}^{\alpha+1\times n_0}$

$$E = \begin{bmatrix} \lambda_1(A_{n_0}) & \lambda_2(A_{n_0}) & \lambda_3(A_{n_0}) & \dots & \lambda_{n_0}(A_{n_0}) \\ \lambda_2(A_{n_1}) & \lambda_4(A_{n_1}) & \lambda_6(A_{n_1}) & \dots & \lambda_{2n_0}(A_{n_1}) \\ \vdots & \vdots & \vdots & \ddots & \\ \lambda_{2^\alpha}(A_{n_\alpha}) & \lambda_{2\cdot 2^\alpha}(A_{n_\alpha}) & \lambda_{3\cdot 2^\alpha}(A_{n_\alpha}) & \dots & \lambda_{n_0\cdot 2^\alpha}(A_{n_\alpha}) \end{bmatrix}. \tag{2.11}$$

This choice of grids and the corresponding subsets of eigenvalues for each level is presented in Figure 2.2.

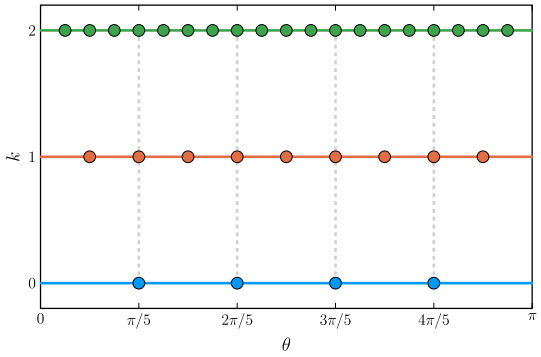

Figure 2.2: Grids for $\alpha = 2$ and $n_0 = 4$.

Once we have calculated the eigenvalues that the estimation is based on, we then use a Vandermonde matrix:

$$V = \begin{bmatrix} 1 & h_0 & h_0^2 & h_0^3 & \dots & h_0^\alpha \\ 1 & h_1 & h_1^2 & h_1^3 & \dots & h_1^\alpha \\ 1 & h_2 & h_2^2 & h_2^3 & \dots & h_2^\alpha \\ 1 & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & h_\alpha & h_\alpha^2 & h_\alpha^3 & \dots & h_\alpha^\alpha \end{bmatrix}, \quad h_k = \frac{1}{1+n_k}, \tag{2.12}$$

and solve the linear system $E = VC$ to find our approximation $\widetilde{C}$ in (2.10).

In Figure 2.3 is presented the approximated $\tilde{c}_k(\theta_{j,n_0})$ for the symbol $f(\theta) = 6 - 8\cos(\theta) + 2\cos(2\theta)$ with $(n_0, \alpha) = (200, 4)$. Note the erratic behaviour of $\tilde{c}_4$ visible close to $\theta = 0$; see detailed discussion in [6, 7].

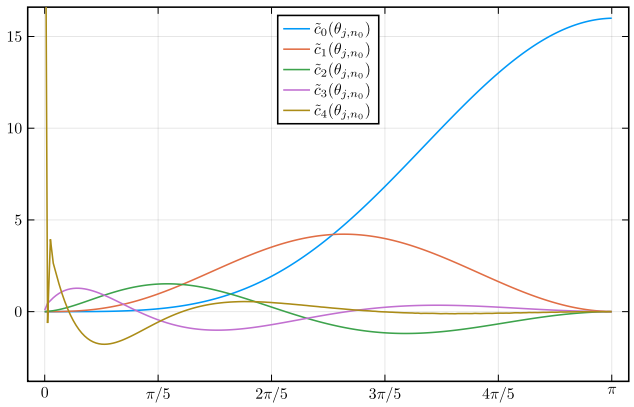

Figure 2.3: The approximations for $c_k(\theta)$ for symbol $f(\theta) = 6 - 8\cos(\theta) + 2\cos(2\theta)$ which corresponds to the finite difference approximation of the bi-Laplacian, computed with $(n_0, \alpha) = (200, 4)$.

So far, matrix-less methods have successfully been employed for a wide class of matrices A_n, most importantly

- $A_n = T_n(f)$, $\{A_n\}_n \sim_{\text{GLT}} f$, f Hermitian [4],
- $A_n = T_n(g)^{-1}T_n(f)$, $\{A_n\}_n \sim_{\text{GLT}} f/g$ [2],
- $A_n = T_n(f)$, $\{A_n\}_n \not\sim_{\text{GLT}} f$, $\{A_n\}_n \sim_{\text{GLT}} c_0$, f non-Hermitian [11, 13],
- $A_n = T_n(f) + R_n$, where R_n is a low-rank matrix [9],
- $A_{\mathbf{n}} = T_{\mathbf{n}}(\mathbf{f})$, where $\mathbf{f}$ is matrix-valued [8].

We shall now numerically investigate if we can use the matrix-less method to approximate the higher order symbols c_k for matrices A_n coming from discretisations of problems with variable coefficients.

2.3 Problem Setting

So far we have discussed a method for efficiently approximating the spectrum of Toeplitz matrices, however in this paper we would like to numerically test if this method can be used more generally. Consider the following 1D diffusion equation:

$$\begin{cases} -\left(a(x)u'(x)\right)' = b(x), & x \in (0,1), \\ u(0) = \gamma_1, \quad u(1) = \gamma_2, \end{cases} \tag{2.13}$$

where $a(x)$ is some given function, $b(x)$ is a source term and γ_1, γ_2 are the values of the solution at the boundaries.

This ODE can be solved using various numerical methods, see e.g., [1]. These often involve converting the problem into a linear system which is solved either explicitly or using an iterative method. For example, a second order Finite Difference approximation of (2.13) using n unknowns and a constant stepsize

$h = \frac{1}{n+1}$ could be [5, p. 192]:

$$-a(x_{j+\frac{1}{2}})u_{j+1}+(a(x_{j+\frac{1}{2}})+a(x_{j-\frac{1}{2}}))u_j-a(x_{j-\frac{1}{2}})u_{j-1} = h^2 b(x_j), \quad j = 1, ..., n. \tag{2.14}$$

where $x_j = jh = \frac{j}{1+n}$, $x_0 = 0, x_{n+1} = 1$ and $u_j = u(x_j)$. Assuming $\gamma_1 = \gamma_2 = 0$, this would lead to the linear system, $A_n \mathbf{u}_n = \mathbf{b}_n$, where $\mathbf{u}_n$ is the vector of unknowns, $b_i = b(x_i)$ is a source term, and the discretisation matrix, A_n, is given by:

$$A_n = \begin{bmatrix} a_{\frac{1}{2}} + a_{\frac{3}{2}} & -a_{\frac{3}{2}} & & & & \\ -a_{\frac{3}{2}} & a_{\frac{3}{2}} + a_{\frac{5}{2}} & -a_{\frac{5}{2}} & & & \\ & -a_{\frac{5}{2}} & \ddots & \ddots & & \\ & & \ddots & \ddots & -a_{n-\frac{1}{2}} & \\ & & & -a_{n-\frac{1}{2}} & a_{n-\frac{1}{2}} + a_{n+\frac{1}{2}} \end{bmatrix}, \tag{2.15}$$

where $a_j = a(x_j),\ x \in (0,1),\ j = 1, ..., n$. Furthermore, it can be proven that:

$$\{A_n\}_n \sim_{\text{GLT}} a(x)g(\theta),$$

where $g(\theta) = 2 - 2\cos(\theta)$ was the symbol of (2.7) [5, p. 193].

Thus, the eigenvalues of a matrix A_{n^2} can be approximated as

$$\lambda_{\mathbf{j}}(A_{n^2}) \approx a(x_{i,n})g(\theta_{j,n}), \qquad \mathbf{j} = (i,j) = (1,1), \ldots, (n,n), \tag{2.16}$$

where $x_{i,n} = i/(n+1)$ and $\theta_{j,n} = j\pi/(n+1)$, see [10].

Now consider the following matrix,

$$G_n = D_n^{(i)}(a)\, T_n(g) = \begin{bmatrix} 2a_1 & -a_1 & & & \\ -a_2 & 2a_2 & -a_2 & & \\ & -a_3 & \ddots & \ddots & \\ & & \ddots & \ddots & -a_{n-1} \\ & & & -a_n & 2a_n \end{bmatrix}, \tag{2.17}$$

where $D_n^{(i)}(a)$ is the diagonal sampling matrix $\left(D_n^{(i)}(a)\right)_{j,j} = a\left(x_{j,n}^{(i)}\right), \quad \forall j = 1, \ldots, n$ with the sampling $x_{i,n}^{(i)}$ and $T_n(g)$ is the Laplacian matrix (2.7) [5, Thrm. 10.5 p. 193]. This matrix, (2.17), will also have the same symbol as (2.15) but different higher order symbols [5, p. 195].

A physical condition that assures wellposedness of (2.13) is $a(x) > 0$ and $a(x) \in C^1(x)$. It will also be numerically tested for $a(x) \in C^0$ or even discontinuous in a finite number of points.

In this report, we will also look whether two different samplings $x_{i,n}$ of the symbol $a(x)$, when generating $D_n(a)$ in (2.17), will result in the eigenvalues of G_n more accurately approximating the eigenvalues of (2.15).

The two different samplings we will consider when constructing $D_n(a)$ of (2.17) will be the standard grid as used in [5],

$$x_{i,n}^{(1)} = \frac{i}{n}, \quad \forall i = 1, \ldots, n \tag{2.18}$$

and a slightly shifted grid

$$x_{i,n}^{(2)} = \frac{i}{n+1}, \quad \forall i = 1, \ldots, n. \tag{2.19}$$

We now propose that we may use the same algorithm, Described in Section 2.2, for A_n defined in (2.15) (and also G_n in (2.17)), formulated in the following working hypothesis.

Working Hypothesis 1 *If* $\{A_n\}_n \sim_{\text{GLT}} a(x)g(\theta)$, *as defined in* (2.15) *and* (2.17), *then we assume that the eigenvalues* $\lambda_j(A_n)$ *behave as*

$$\lambda_j(A_n) \approx \sum_{k=0}^{\alpha} c_k(\xi_{j,n})h^k.$$

where $\xi_{j,n} = \frac{j\pi}{n+1}$ *is an equispaced grid.*

We will numerically test the working hypothesis for a wide range of functions $a(x)$ and for $g(\theta) = 2 - 2\cos(\theta)$.

Remark 1 *Note that in Working Hypothesis 1 we have a symbol of the form* $f(x, \theta) = a(x)g(\theta)$ *for the matrix sequence* $\{A_n\}_n$ *of interest and the expansion is a linear combination of univariate symbols* $c_0(\xi), c_1(\xi), c_2(\xi), \ldots$, *defined on* $\xi \in [0, \pi]$. *Some other domain, for example* $\xi \in [-\pi, \pi]$ *could have been chosen.*

Thus we have now outlined in detail the problem which we would like to investigate in this paper, and so we will now look to the numerical experiments which have been performed and their results.

3 Numerical Experiments

Now that we have introduced the problem, we can begin to numerically test a number of different functions $a(x)$ and investigate if the matrix-less method is able to compute the symbols (and higher order symbols) for a number of different functions $a(x)$ that determine the variable coefficient in the diffusion equation. We will first begin by looking at some simple smooth, monotone functions. Then, we will look at a non-monotone function before ending with a closer look at non-smooth functions. Note that in some experiments we compare the spectral results of using matrices (2.15) and (2.17) (in two variants), which have the same symbol. In Appendix B we show how (2.17) can be related to a symmetric matrix which has the same characteristic polynomial and thus allows us to use some optimised methods.

A brief summary of the numerical examples, with different function $a(x)$, are listed below:

Constant and Linear functions

- Example 1: $a(x) = 1$,
  The case of using a constant function, will result in matrix (2.7).
- Example 2: $a(x) = x$,
  A simple linear, monotone function to investigate the Working Hypothesis 1.
- Example 3: $a(x) = 1 - \varepsilon + \varepsilon x$,
  We investigate the transition from a purely constant case to a purely linear case.

Laplace-like

- Example 4: $a(x) = 2 - 2\cos(\pi x)$,
  In this example we study matrices giving spectral behaviour reminecent of the bi-Laplacian $T_n(f)$, $f(\theta) = (2 - 2\cos(\theta))^2$.

Non-monotonic

- Example 5: $a(x) = \sin(\pi x)^2$,
  We study the effect of two different grids when generating $G_n = D_n(a)(2 - 2\cos(\theta))$ (2.17) as compared with A_n (2.15).

Non-smooth

- Example 6: $a(x) = \begin{cases} 2^k (x-0.5)^{k+1} + 0.5, & x \leq 0.5 \\ 0.5 \exp\left(-\frac{1}{x-0.5} + 2\right) + 0.5, & x > 0.5 \end{cases}$,
  We study how the smoothness of $a(x)$ impacts the expansion functions c_k.

Discontinous

- Example 7: $a(x) = \begin{cases} 1-\varepsilon, & x \leq 0.5 \\ 1+\varepsilon, & x > 0.5 \end{cases}$,
  We study the different behaviour of c_k for different discontinuous $a(x)$.

From observation of the numerical examples, we found the following was typical behaviour for the approximations for the different symbols c_k (this approximation denoted as $\tilde{c}_k(\theta)$), which can be on part of or on the whole domain $[0, \pi]$ of the function $\tilde{c}_k(\theta)$:

(P1) Same curve for different α;

(P2) Different curve for different α, smooth behaviour;

(P3) Erratic behaviour for a finite number of samplings, behaviour changes for different α;

(P4) Fully chaotic behaviour in part of the domain.

3.1 Constant and Linear Examples

Example 1 *For the first example, we choose a rather trivial case, in that it leads to a standard Toeplitz matrix which the matrix-less method is already known to work well for. Choose,*

$$a(x) = 1, \tag{3.1}$$

which yield $c_0(\theta) = 2 - 2\cos(\theta)$ *(to machine precision), as shown in Figure 2.1, and all* $\tilde{c}_k, k > 0$ *are zero, since the eigenvalues of the matrix* $T_n(g)$*, where* $g(\theta) = 2 - 2\cos(\theta)$*, are given exactly by the grid* $\theta_{j,n} = j\pi/(n+1)$*, that is,* $\lambda_j(A_n) = g(\theta_{j,n})$*.*

Example 2 *Next, we have have a simple monotonically increasing function,*

$$a(x) = x, \tag{3.2}$$

which yields the symbol $f(x, \theta) = a(x)g(\theta) = x(2 - 2\cos(\theta))$ *for* $\{A_n\}_n$*.*

In Figure 3.1 we present the numerically computed $\tilde{c}_k(\theta_{j,n_0})$ *for* $n_0 = 1000$ *and* $\alpha = 3$*.*

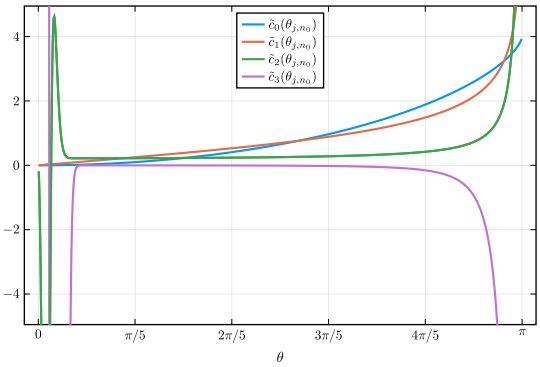

Figure 3.1: [Example 2: $a(x) = x$] Computed $\tilde{c}_k$ for $n_0 = 1000$ and $\alpha = 3$.

We can summarise some our observations of numerical experiments for this symbol in the following items

1. *The errors $|\lambda_j(A_n) - \tilde{\lambda}_{j,n}|$, for $n = 100000$, decrease as (n_0, α) increases, except close to $\theta = \{0, \pi\}$. See Figure 3.2.*
   - *Close to $\theta = 0$ we have erratic (but "smooth") behaviour.*
   - *Close to $\theta = \pi$ the error does not decrease.*

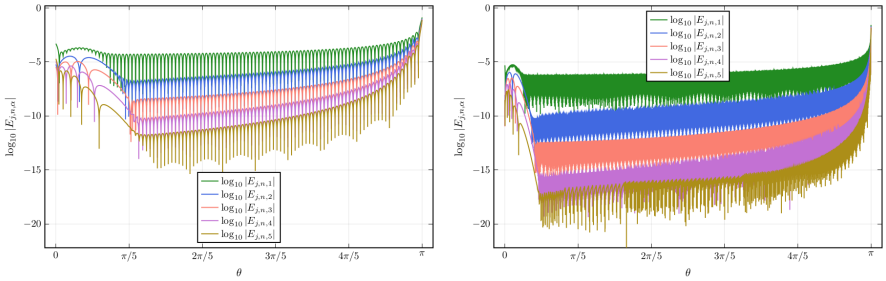

Figure 3.2: [Example 2: $a(x) = x$] Errors $\log_{10}|\lambda_j(A_n) - \tilde{\lambda}_{j,n}|$ for $\alpha = 1, \ldots, 5$ and $n = 100000$. Left: $n_0 = 100$. Right: $n_0 = 1000$.

2. *If we vary α in our computations $\tilde{c}_0$ and $\tilde{c}_1$ remain the same, as seen in Figure 3.3. This is clear (P1) behaviour.*

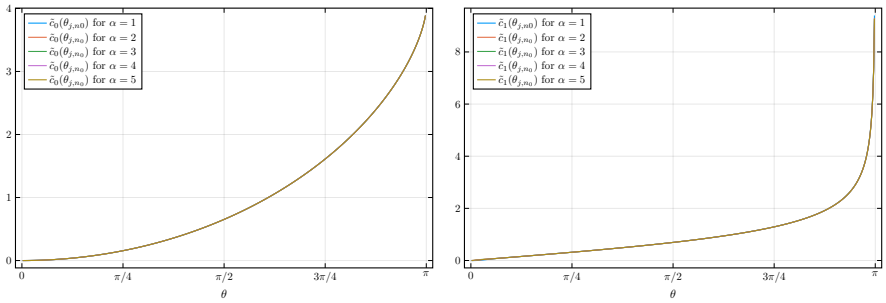

Figure 3.3: [Example 2: $a(x) = x$] Computed $\tilde{c}_k$ for different α and $n_0 = 500$. Left: $\tilde{c}_0$. Right: $\tilde{c}_1$.

3. *If we vary α in our computations $\tilde{c}_2$ has a different shape for different α for θ close to zero; see left panel of Figure 3.4. This is (P2) behaviour. For the rest of $\tilde{c}_2$ we have (P1) behaviour.*

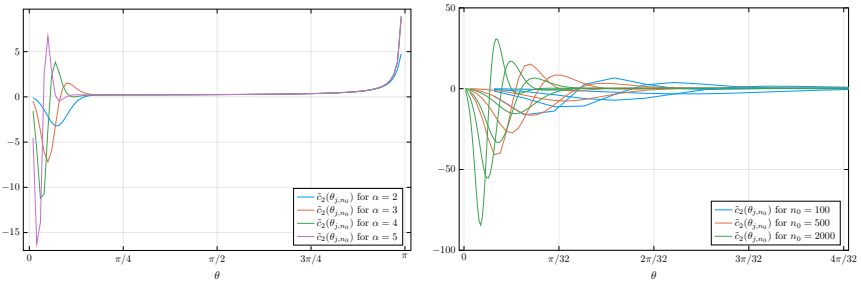

Figure 3.4: [Example 2: $a(x) = x$] Computed $\tilde{c}_2$ for different n_0 and α. Left: $n_0 = 100$ and $\alpha = 2, 3, 4, 5$. Right: Detail close to $\theta = 0$ $n_0 = 100, 500, 2000$ (with $\alpha = 2, 3, 4, 5$).

4. *If we increase n_0 in our computations the erratic region close to $\theta = 0$ shrinks; see right panel of Figure 3.4.*
5. *The magnitude of the maxima/minima close to $\theta = 0$ for the computed $\tilde{c}_2$ seems to increase with increasing α as in Figure 3.4*
6. *The computed $\tilde{c}_3$ has a similar behaviour as $\tilde{c}_2$ presented in Figure 3.4, that is (P1) behaviour is most of the domain and (P2) in a small part of it.*
7. *The computed symbols $\tilde{c}_2$ and $\tilde{c}_3$ seems to converge towards zero in the main part of the spectrum (away from $\theta = \{0, \pi\}$) as α is increased. As stated in previous items the size of this region, with (P1) behaviour, increases as n_0 and α increases.*

Remark 2 *We used* `Double64` *data type in Julia [3] for all computations to ensure that numerical errors should not affect our computations; see [14].*

We conclude that as n_0 and α increases the magnitude of the erratic region close to $\theta = 0$, where $\tilde{c}_2$ behaves differently for different α, region of (P2) behaviour, as in Figure 3.4, shrinks. Also, the "bad" region close to $\theta = \pi$ shrinks as n_0 and α are increased.

Example 3 *To analyse how the spectrum of* (2.15) *changes when we transition between the simple functions* (3.1) *in Example 1 and* (3.2) *in Example 2, we use a parameter ε to define*

$$a(x;\, \varepsilon) = (1 - \varepsilon) + \varepsilon x \tag{3.3}$$

such that $a(x;\, 0) = 1$ and $a(x;\, 1) = x$.

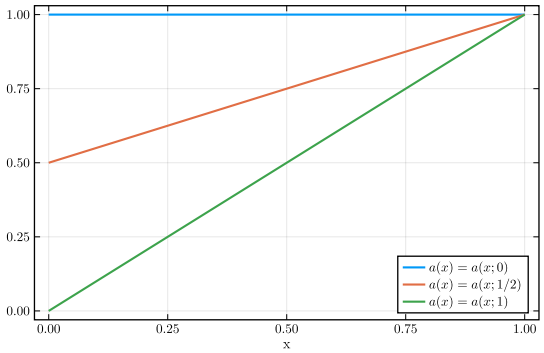

Figure 3.5: [Example 3: $a(x;\varepsilon) = (1 - \varepsilon) + \varepsilon x$] The function $a(x) = a(x;\varepsilon)$ for $\varepsilon = \{0, 1/2, 1\}$.

In Figure 3.6 is shown the computed $\tilde{c}_0$, $\tilde{c}_1$, and $\tilde{c}_2$ for various different $\varepsilon = \{0, 1/4, 1/2, 3/4, 1\}$.

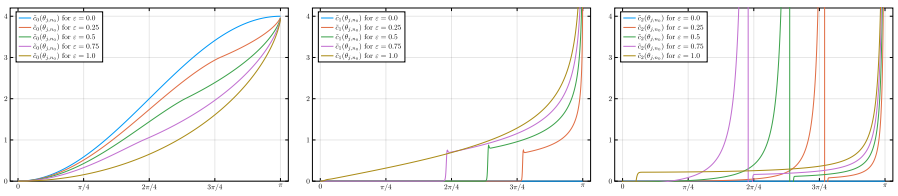

Figure 3.6: [Example 3: $a(x;\varepsilon) = (1 - \varepsilon) + \varepsilon x$] Computed $\tilde{c}_k$ for $n_0 = 1000$ and $\alpha = 2$ for different $\varepsilon = \{0, 1/4, 1/2, 3/4, 1\}$. Left: $\tilde{c}_0$. Middle: $\tilde{c}_1$. Right: $\tilde{c}_2$.

In Figure 3.7 we see the computed $\tilde{c}_k$ for $\varepsilon = 1/2$, for $n_0 = 1000$ and $\alpha = 3$.

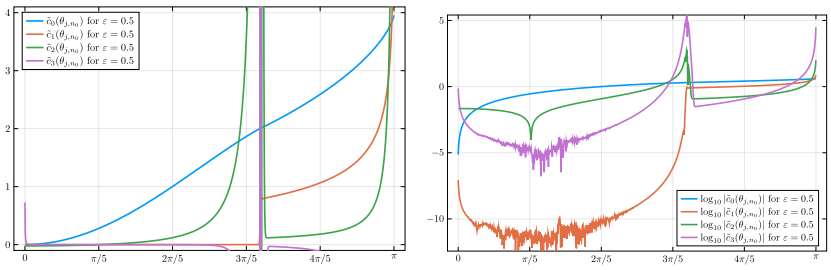

Figure 3.7: [Example 3: $a(x;\varepsilon) = (1-\varepsilon) + \varepsilon x$] The computed $\tilde{c}_k$ for $\varepsilon = 1/2$, $n_0 = 1000$, and $\alpha = 3$.

Numerical observations in this example are

1. *The erratic (P2) behaviour described in Example 2 for $\tilde{c}_2$ is present also in the example for varying ε but the erratic behaviour is no longer at around $\theta = 0$ but rather where the discontinuity in $\tilde{c}_1$ is, as seen in the middle panel of Figure 3.6. Computations for different α and $\varepsilon = 1/2$ is shown in Figure 3.8.*

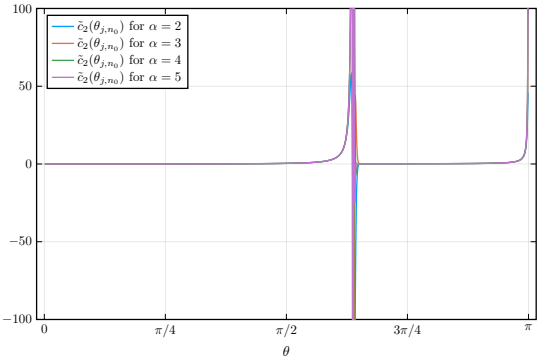

Figure 3.8: [Example 3: $a(x;\varepsilon) = (1-\varepsilon) + \varepsilon x$] $\tilde{c_2}$ for varying α with $\varepsilon = 1/2$ and $n_0 = 1000$

2. *$\tilde{c}_1$ appears to be zero until a discontinuity at a certain θ, depending on ε (as ε increases this discontinuity moves to the left). For $\varepsilon = 0$ it is at $\theta = \pi$ and for $\varepsilon = 1$ it is at $\theta = 0$. See middle panel Figure 3.6. If we consider the point $\tilde{c}_1(\theta_\varepsilon)$ at which the different $\tilde{c}_1(\theta)$ curves have a discontinuity, this appears to move like a smooth non-linear function of ε.*
3. *$\tilde{c}_2$, presented in the right panel of Figure 3.6 has a discontinuity in the same locations as $\tilde{c}_1$ in the middle panel.*
4. *In Figure 3.9 we present the computed errors $\log_{10}|\lambda_j(A_n) - \tilde{\lambda}_{j,n}|$ for $n = 100000$ with different n_0 and α. Clearly, there is a difficulty to compute the eigenvalue approximation accuratly close to the discontinuity discussed in previous items. However, we have nice (P1) behaviour in the rest of the domain except close to $\theta = \pi$.*

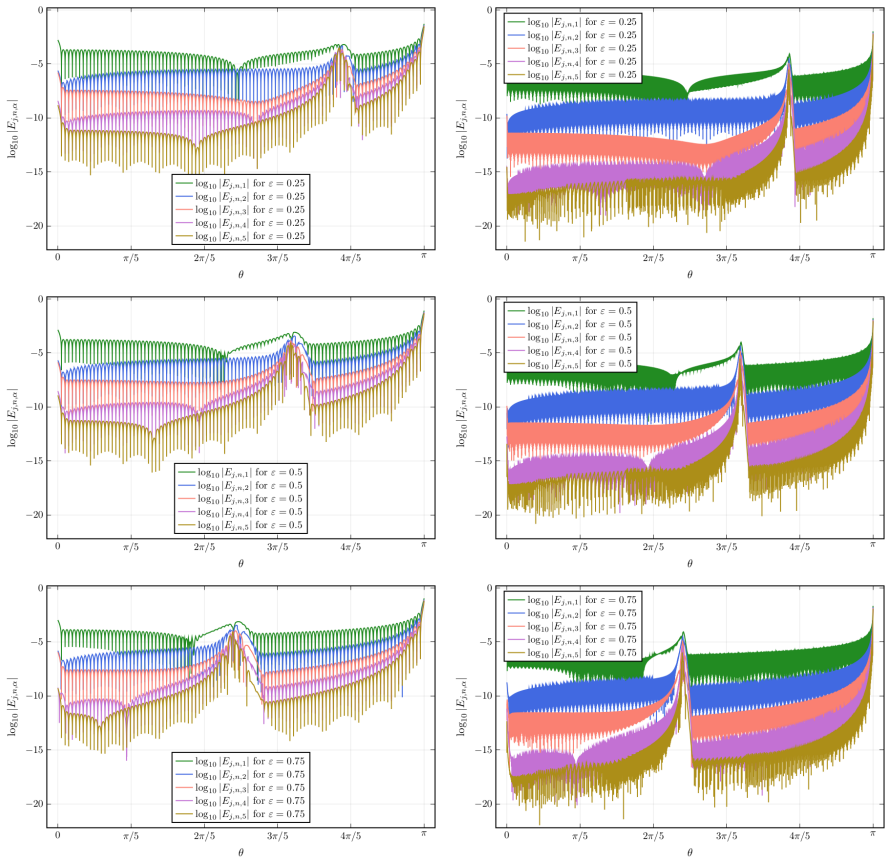

Figure 3.9: [Example different linear 2: $a(x) = x$] Errors $\log_{10}|\lambda_j(A_n) - \tilde{\lambda}_{j,n}|$ for $\alpha = 1, \ldots, 5$. and $n = 100000$ Top: $\varepsilon = 1/4$, Middle: $\varepsilon = 1/2$, Bottom: $\varepsilon = 3/4$. Left: $n_0 = 100$. Right: $n_0 = 1000$.

3.2 Laplace-like Example

Example 4 *A lot of interest has been given to the study of the symbol* $f(\theta) = (2 - 2\cos(\theta))^2 = 6 - 8\cos(\theta) + 2\cos(2\theta)$*; see for example [6, 12]. If we have* $g(\theta) = 2 - 2\cos(\theta)$*, then* $f(\theta) = g(\theta)^2$*. If we now define*

$$a(x) = 2 - 2\cos(\pi x), \tag{3.4}$$

with $x \in [0, 1]$ *we have* $f(x, \theta) = a(x)g(\theta)$ *which in some sense is related to the bivariate symbol* $f(\theta_1, \theta_2) = g(\theta_1)g(\theta_2)$ *(the generated matrix by this symbol is the matrix* $T_{\mathbf{n}}(f) = T_{n_1}(g) \otimes T_{n_2}(g)$*, where* n_1 *and* n_2 *are the number of discretization points in each dimension).*

As seen in Figure 3.10, the expansion can be computed. However, as seen close to $\theta = 0$ *there is erratic (P3) behaviour for* $\tilde{c}_2$ *and* $\tilde{c}_3$*. This is similar to the behaviour described in detail, for the related symbol* $f(\theta) = (2 - 2\cos(\theta))^2$ *in [6, 12].*

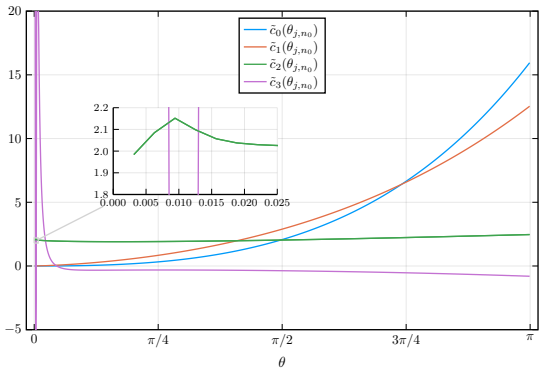

Figure 3.10: [Example 4: $a(x) = 2-2\cos(\pi x)$] The approximations $\tilde{c}_k$ computed for $n_0 = 1000$ and $\alpha = 3$.

In Figure 3.11 is seen the erratic (P3) behaviour close to $\theta = 0$. Using different α in the computations yield different solutions.

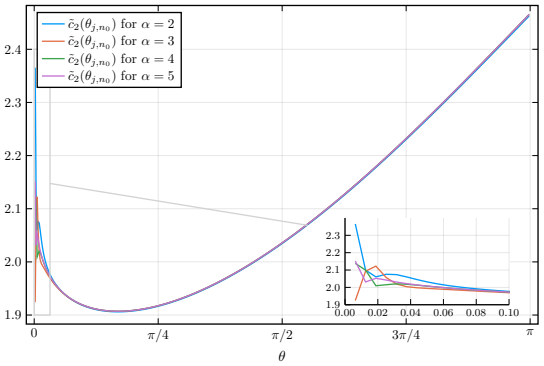

Figure 3.11: [Example 4: $a(x) = 2-2\cos(\pi x)$] Erratic behaviour of approximations of $\tilde{c}_2$ close to $\theta = 0$ when computing with different α.

Apart from these finite number of erratic value, with (P3) behaviour, in $\tilde{c}_2$ and $\tilde{c}_3$, the expansion works well and can be used to interpolate-extrapolate the $\tilde{c}_k$ for a large matrix; see Figure 3.12 for $n = 100000$.

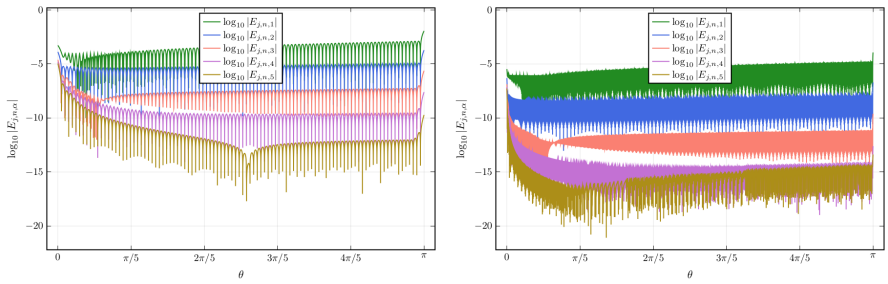

Figure 3.12: [Example 4: $a(x) = 2 - 2\cos(\pi x)$] Errors, for $n = 100000$, $\log_{10}|\lambda_j(A_n) - \tilde{\lambda}_{j,n}|$ for $\alpha = 1, \ldots, 5$. Left: $n_0 = 100$. Right: $n_0 = 1000$.

3.3 Non-Monotonic Example

So far, we have only considered functions which are monotone on the interval of interest. Next we will consider examples where $a(x)$ is not monotone over the interval.

Example 5 *In this example we consider the function,*

$$a(x) = \sin(\pi x)^2. \tag{3.5}$$

In Figure 3.13 the symbol $f(x,\theta) = a(x)(2 - 2\cos(\theta))$ is shown.

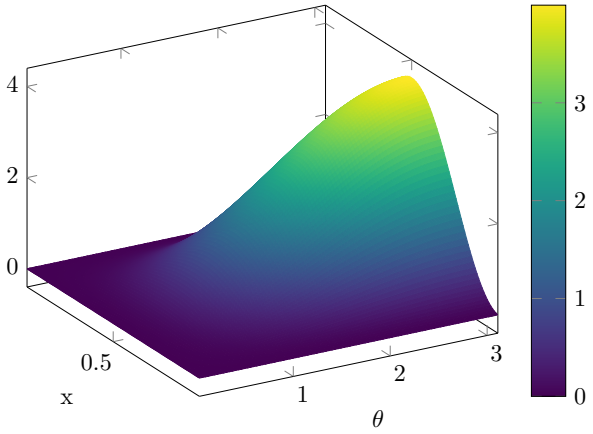

Figure 3.13: [Example 5 $a(x) = \sin(\pi x)^2$] Visualization of the symbol $f(x,\theta) = a(x)(2 - 2\cos(\theta))$.

However, the main point of this example is to study the different spectral behaviour of the expansion functions c_k for the three spectrally related matrices, (2.15) *and,* (2.17) *using the two grids $x_{i,n}^{(1)}$* (2.18) *and $x_{i,n}^{(2)}$* (2.19)*, that all share the same symbol $f(x,\theta) = a(x)(2 - 2\cos(\theta))$.*

We denote by $\tilde{c}_k^{(0)}$ the functions approximated for matrix (2.15)*, while $\tilde{c}_k^{(1)}$ and*

$\tilde{c}_k^{(2)}$ *denote the functions approximated for matrix* (2.17) *using the grids* (2.18) *and* (2.19).

In the left panels of Figure 3.14, $\tilde{c}_k$ *for the three matrix sequences defined above are shown, with* $k = 0, 1, 2$. *In the right panels the difference between* $\tilde{c}_k^{(0)}$ *and the two other discretisations are shown.*

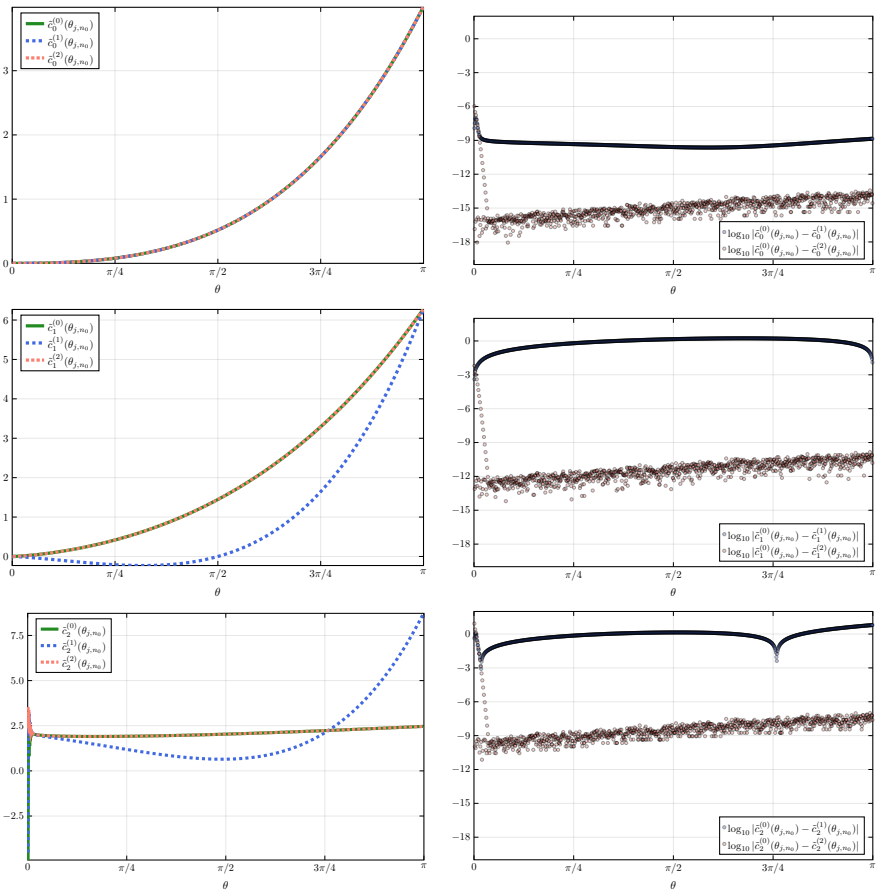

Figure 3.14: [of Example 5: $a(x) = \sin(\pi x)^2$] Visualisation and comparison of $\tilde{c}_k^{(0)}, \tilde{c}_k^{(1)}, \tilde{c}_k^{(2)}$.

We can summarise some our observations of numerical experiments for this symbol in the following items,

1. *Clearly seen in right panels of Figure 3.14 is that using* (2.19) *for generating the diagonal sampling matrix* $D_n(a)$ *in* (2.17) *yields a much more accurate approximation of the functions* $\tilde{c}_k$ *of* (2.15) *than using* (2.18).
2. *The functions* $\tilde{c}_k$ *have nice (P1) behaviour over the whole domain, except* $\tilde{c}_2$ *close to* $\theta = 0$, *as seen in Figure 3.15. This is similar to the behaviour present in Example 4.*

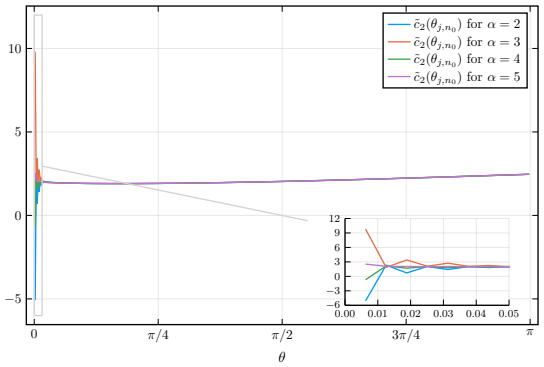

Figure 3.15: [Example 5: $a(x) = \sin(\pi x)^2$] The approximated $\tilde{c}_2$ for various different α. Note the erratic behaviour for a few approximations close to $\theta = 0$.

3.4 Non-Smooth Example

So far we have only considered functions which are infinitely many times differentiable on the domain of interest. However, we will now look at what affect differetiability, and indeed continuity, could possibly have on the higher order symbols. Since the final two examples will make use of piecewise defined functions, we will make the following remark:

Remark 3 *Assume we have a function:*

$$a(x) = \begin{cases} a_A(x), & x \leq 0.5, \\ a_B(x), & x > 0.5, \end{cases}$$

then the matrix A_n, defined in (2.15)*, is symmetric tridiagonal, and of the form*

$$A_n = \begin{bmatrix} A & R \\ R^T & B \end{bmatrix}, \tag{3.6}$$

Where R is a low rank matrix with only one non-zero element in the bottom left corner, and A and B are related to the two functions $a_A(x)$ and $a_B(x)$. Assuming n even, then two separate eigenvalue functions (one from the A ($f_A(x,\theta)$) part and one from B ($f_B(x,\theta)$) part will describe the spectrum). We would have then have that:

$$f_A(x,\theta) = a_A\left(\frac{x}{2}\right)(2 - 2\cos(\theta))$$
$$f_B(x,\theta) = a_B\left(\frac{x+1}{2}\right)(2 - 2\cos(\theta))$$

Note that if the discontinuity is not at $x = 0.5$ then R will be non-square and A and B differ in size. Also, if there are multiple intervals with functions defined on them in the piecewise function, then more blocks like R, A, B will be present.

We will utilise Remark 3 further in the examples that follow.

Example 6 *In this example we will consider the following* $\mathcal{C}^k([0,1])$ *function defined as:*

$$a(x) = \begin{cases} 2^k \left(x - 0.5\right)^{k+1} + 0.5, & x \leq 0.5 \\ 0.5 \exp\left(-\frac{1}{x-0.5} + 2\right) + 0.5, & x > 0.5 \end{cases}, \tag{3.7}$$

where $k \in [0, \infty)$ *indicates the number of times this function is differentiable on* $[0,1]$*. Figure 3.16 shows this function for* $k = 0$ *and* $k = 1$ *on the domain* $[0,1]$*.*

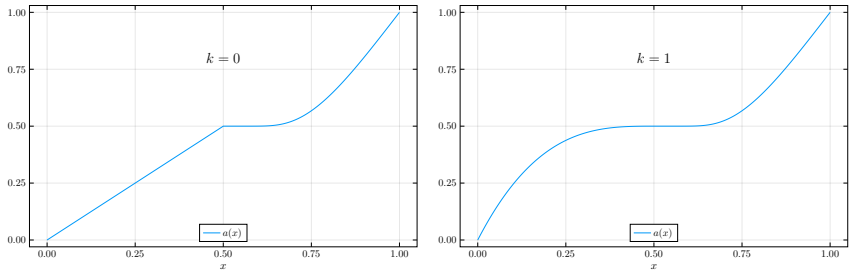

Figure 3.16: [Example 6: k-times differentiable function] Function $a(x)$ for different k. Left: $k = 0$. Right: $k = 1$.

For $n = 2048$ *we present in Figure 3.17 for* $k = 0$ *(left panel) and* $k = 1$ *(right panel) the following properties: The eigenvalues* $\lambda_j(A_n)$ *(blue line),* $\lambda_j(A)$ *(red line), and* $\lambda_j(B)$ *(green line). The sorted union of the eigenvalues of* A *and* B *(dashed black line) overlap well the eigenvalues of the matrix* A_n*. Also, the rearranged samplings of* $f_A(x,\theta) = (2^k \left(\frac{x}{2} - 0.5\right)^{k+1} + 0.5)(2 - 2\cos(\theta))$ *(dashed cyan line) and* $f_B(x,\theta) = (0.5 \exp\left(-\frac{1}{x/2} + 2\right) + 0.5)(2 - 2\cos(\theta))$ *(dashed orange lane) are shown, and they overlap the eigenvalues of* A *and* B *respectively.*

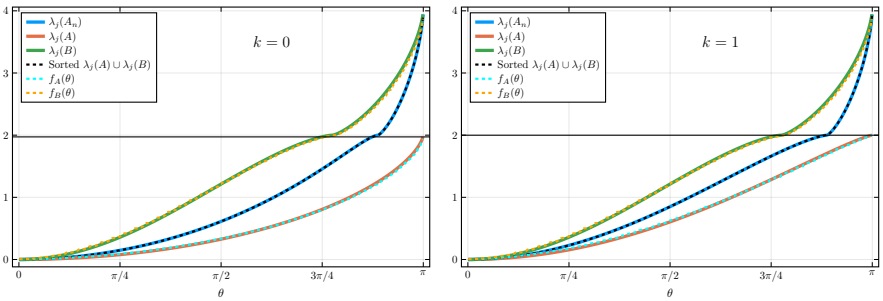

Figure 3.17: [Example 6: k-times differentiable function] Comparison of block matrix simplification of A_n and the respective symbols. Left: $k = 0$. Right: $k = 1$.

We have the following observations from the numerical experiments,

1. *The function* c_0 *is accurately approximated, with (P1) behaviour, both for* $k = 0$ *and* $k = 1$*. Visually looks like the presented eigenvalue curves* $\lambda_j(A_n)$ *in Figure 3.17.*

2. *Both for $k = 0$ and $k = 1$, most of the approximation $\tilde{c}_1$ has (P1) behaviour. However, close to the discontinuity, as seen in both panels of Figure 3.18, there is (P2) behaviour, that is as we change α in our computation we locally get different curves around the discontinuity.*

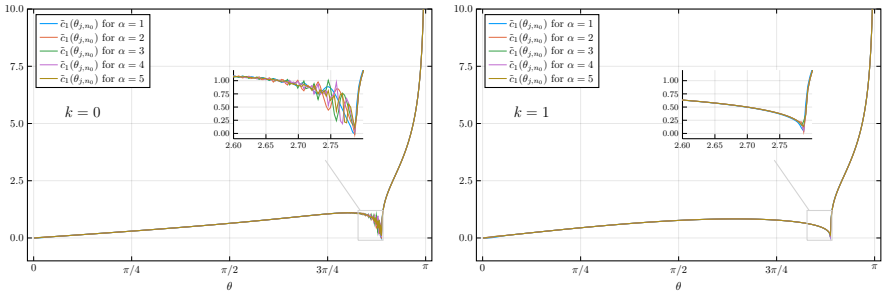

Figure 3.18: [Example 6: k-times differentiable function] Visualization of $\tilde{c}_1$. Left: $k = 0$. Right: $k = 1$.

3. *In the left panel of Figure 3.19 we see (P4) behaviour, that is chaotic behaviour, in most part of the domain. There is only a small region close to $\theta = \pi$ where it is smooth; this is the region where f_A does not overlap f_B.*
4. *In the right panel of Figure 3.19 we see, as opposed to the left panel, that most of the domain has nice (P1) behaviour. Only close to $\theta = 0$ we see erratic (P2) behaviour. For $\tilde{c}_3$ and $k = 1$ we found similar behaviour.*

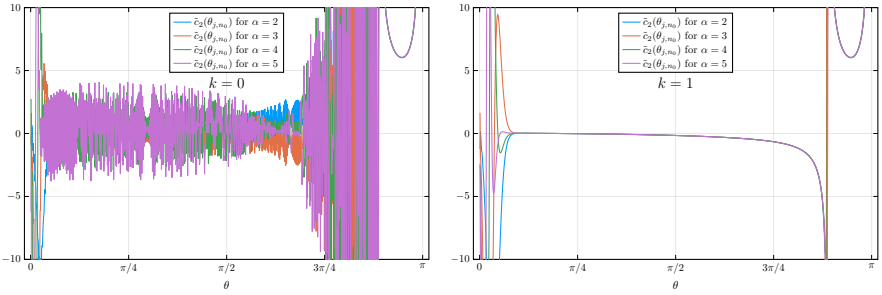

Figure 3.19: [Example 6: k-times differentiable function] Visualization of $\tilde{c}_2$. Left: $k = 0$. Right: $k = 1$.

5. *From this point we have observed that increasing k has tended to decrease the noise in $\tilde{c}_1$ and $\tilde{c}_2$ somewhat and also shifted it towards 0 and so we ask ourselves whether this trend would continue for larger values of k. Figure 3.20 shows the results for $k = 10$ (left panel) and $k = 100$ (right panel) where indeed we do see that this trend has continued and for the case $k = 100$, $\tilde{c}_1$ is completely 0 up until the notch and the noise in $\tilde{c}_2$ in now only concentrated around 0.*

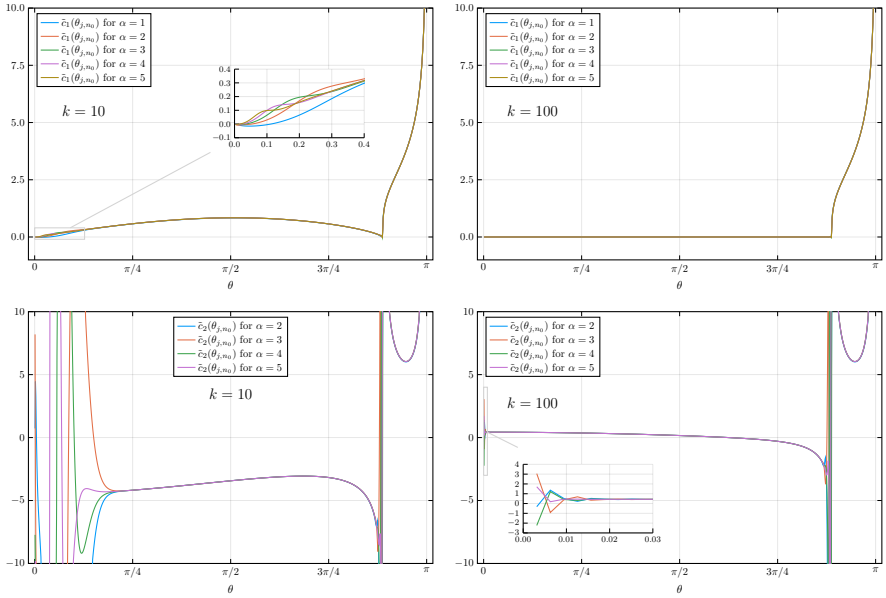

Figure 3.20: [Example 6: k-times differentiable function] Computed $\tilde{c}_1$ and $\tilde{c}_2$ for varying α. Left: $k = 10$. Right: $k = 100$.

3.5 Discontinuous Example

Example 7 *In this example we study a discontinuous step function $a(x)$, namely*

$$a(x) = \begin{cases} 1 - \varepsilon, & x \leq 0.5, \\ 1 + \varepsilon, & x > 0.5, \end{cases} \tag{3.8}$$

which could model a case of having an interface between two materials having different diffusion coefficients.

In the left panel of Figure 3.21 we show, for $\varepsilon = 0.1$ and $n = 2000$ the following properties: the eigenvalues $\lambda_j(A_n)$ (blue line), $\lambda_j(A)$ (red line), and $\lambda_j(B)$ (green line). Also shown are the sorted union of the eigenvalues of A and B, defined in Remark 3 (dashed black line). The symbols

$$f_A(x, \theta) = (1 - \varepsilon)(2 - 2\cos(\theta))$$
$$f_B(x, \theta) = (1 + \varepsilon)(2 - 2\cos(\theta))$$

are presented in rearranged form (sample on an equispaced grid over x and θ and sort samplings by size). As seen, the sorted union of the eigenvalues of A and B overlap the sorted eigenvalues of A_n.

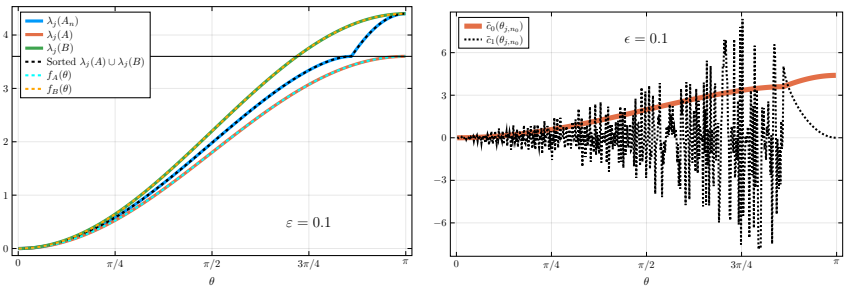

Figure 3.21: [Example 7: Discontinuous $a(x)$] For $\varepsilon = 0.1$ in (3.8). Left: Eigenvalues for $n = 2000$ of A_n (blue line), A (red line), B (green line), and sorted union of eigenvalues of A and B (black dashed line). Right: Computed $\tilde{c}_0$ and $\tilde{c}_1$ for $n_0 = 1000$ and $\alpha = 2$. Note that $\tilde{c}_0$ matches the sorted eigenvalues of A_n in the left panel.

We here list a few observations from the numerical experiments

1. *In the region where f_A and f_B overlap, the eigenvalues of A and B mix when sorting the union of the two, and then the matrix-less method will not be able to work; see the right panel of Figure 3.21 where $\tilde{c}_1$ is just noise left of the discontinuity in $\tilde{c}_0$. This corresponds to (P4) behaviour. Hence, as ε increases a larger portion of the spectrum can be reconstructed by the asymptotic expansion and the matrix-less method.*

2. *Of course $\varepsilon = 0$ is smooth, as it is the constant case and $\varepsilon = 1$ is a scaled up constant for $\theta > 1/2$ and zero otherwise but we are also interested in other values of ε. We can see in Figure 3.22 how the point with discontinuous derivative changes for different ε. This is until we reach $\varepsilon = 1$ and the second function that is continuous to the first derivative.*

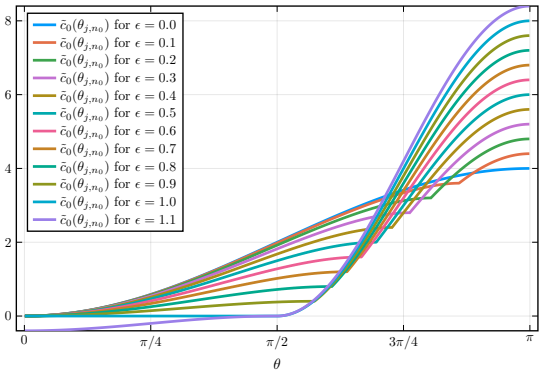

Figure 3.22: [Example 7: Discontinuous $a(x)$] Computed $\tilde{c}_0$ for various ϵ.

3. *In Figure 3.23 we show in the left panels $\tilde{c}_1$ and in the right panels $\tilde{c}_2$. In the top panels $\varepsilon = 0.1$, middle panels $\varepsilon = 0.9$, and bottom panels $\varepsilon = 1.1$. As we vary α in each panel we see (P2) behaviour in the smooth regions and (P4) in the regions when f_A and f_B overlap.*

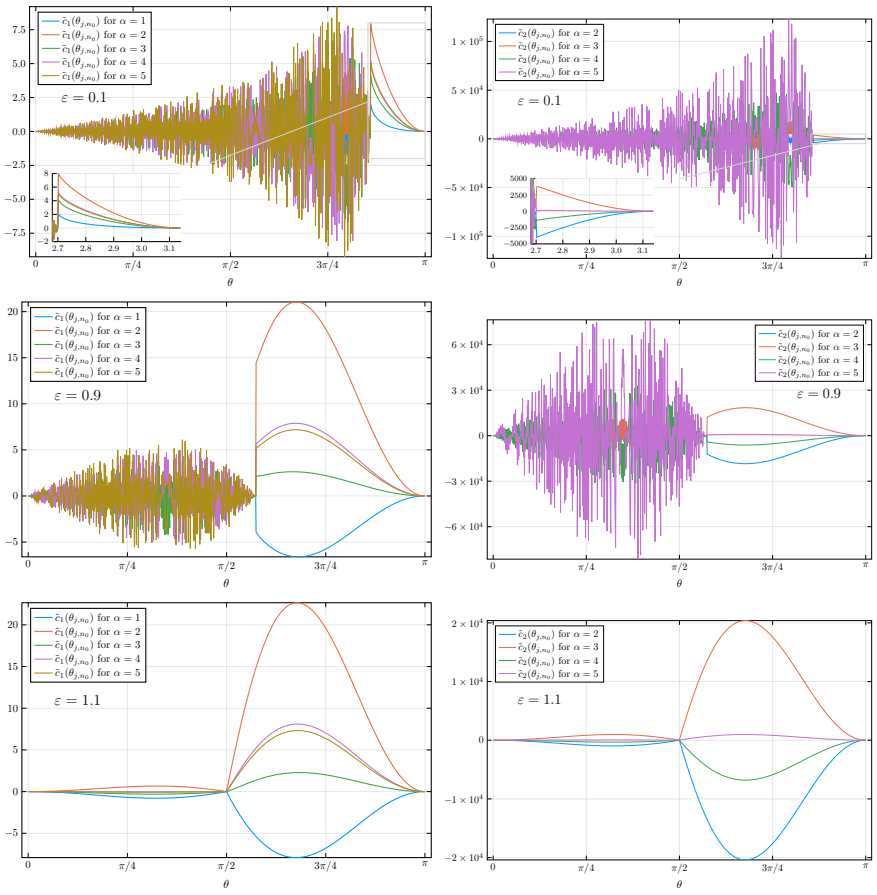

Figure 3.23: [Example 7: Discontinuous $a(x)$] The computed $\tilde{c}_1$ (left) and $\tilde{c}_2$ (right) for different α and for various different ε. Top: $\varepsilon = 0.1$. Middle: $\varepsilon = 0.9$. Bottom: $\varepsilon = 1.1$.

4. *In the bottom panel of Figure 3.23 , that is* $\varepsilon = 1.1$*, then* f_A *and* f_B *do not overlap, and we have (P2) behaviour over the whole domain. We also see two distinct regions* $[0, \pi/2]$ *and* $[\pi/2, \pi]$ *present in both* $\tilde{c}_1$ *and* $\tilde{c}_2$*.*

4 Conclusions

The goal of this paper was to numerically investigate whether the matrix-less method could successfully be used to approximate the eigenvalues of discretisations of the variable coefficient diffusion equation, when assuming Working Hypothesis 1.

The first non-trivial example was the simple linear case of $a(x) = x$ where we found that $\tilde{c}_0$, $\tilde{c}_1$ had (P1) behaviour (nice and smooth for different α). For $\tilde{c}_2$, and most likely the higher ordered symbols of larger degree, (P1) behaviour was observed in most of the domain, but also (P2) behaviour (varies with different α) and (P3) behaviour (a finite number of the values are erratic for every α). As we increase n_0 these bad regions decrease in size.

We then investigated the behaviour of the higher-order symbols when the function transitions from a constant case to a linear case. We found here that $\tilde{c}_1$, for $\varepsilon \in (0, 1)$, was zero up to some point when it would suddenly jump in value before moving quickly towards the discontinuity at π. This point of jumping in value moved towards 0 as ε increased, however not in a linear fashion and it should also be noted that the magnitude of this jump was also not linear with ε. The point where this jump occurred was also present in $\tilde{c}_2$ as an asymptote which had worse behaviour for larger values of α. We found though that the interpolation of eigenvalues worked well for most of the spectrum except for the point where this jump occurred.

We then tested a function which would result in a a symbol similar to that of a bi-Laplacian matrix. In this case, it was found that the expansions worked fairly well and it was possible to perform an interpolation-extrapolation successfully except for an erratic (P3) region near $\theta = 0$.

A non-monotone function was then tested to primarily compare the different grids, (2.18) and (2.19), used in (2.17) where it was generally found that the approximated higher order symbols $\tilde{c}_k$ of the matrix using the (2.19) grid performed much more similarly to the those of (2.15). It was also observed that there was some erratic (P2) behaviour in $\tilde{c}_2$ around $\theta = 0$ when changing the values of α.

Following the investigation into a non-monotone function, a $\mathcal{C}^k([0, 1])$ function was then tested, meaning a function with k continuous derivatives on the interval $[0, 1]$. It was found that for all k, the $\tilde{c}_1$ had a point at which it suddenly goes to 0 and then after hitting 0, very quickly increases. This region was erratic for $k = 0$ but fairly stable for larger k. Furthermore, $\tilde{c}_2$ was extremely erratic

across almost the entire domain of $k = 0$, which corresponds to (P4) behaviour, and is due to the overlap of f_A and f_B in this region. Surprisingly, this region is smooth for $k = 1$, but we notice a region of (P2) behaviour close to $\theta = 0$ (and also for $\tilde{c}_3$). If increasing continuity with $k = 100$ we see that $\tilde{c}_1$ is zero until the point where it quickly increases, and $\tilde{c}_2$ has a good (P1) behaviour in most of the domain.

A discontinuous function was then tested where the most striking feature in the discontinuous example is that when symbols f_A and f_B overlap we have (P4) behaviour, that is, chaotic. In the smooth regions, we have (P2) behaviour and we conjecture in item (E2) below as a possible explanation.

Finally we present the following list of possible explanations and conjectures on the reasons of behaviour (P1)–(P4) observed in the numerical results:

(E1) The expansion in Working Hypothesis 1 works and the c_k functions can be approximated.

(E2) We conjecture that the expansion in Working Hypothesis 1 should be modified to

$$\lambda_j(A_n) \approx \sum_{k=0}^{\alpha} c_k(\xi_{j,n}) h^{k\gamma}.$$

where $\gamma \in \mathbb{R}$ is some constant.

(E3) This behaviour can not be resolved since Working Hypothesis 1 is not correct for these values, just like described in [6]. The easiest solution is to discard these erratic samplings before doing interpolation-extrapolation of $\tilde{c}_k(\theta_{j,n_0})$ for large n.

(E4) In some cases this phenomenon is due to numerical noise, and can be remedied by either decreasing n_0 or increasing the precision of the computation. Also, due to non-monotone symbols or overlapping symbols (as seen in Example 7) it is not possible to do a correct ordering of eigenvalues of A_n. In most cases, no known solution to this problem is known.

For future research we suggest the following items:

1. Study the functions $a(x)$ where (P2) behaviour, that different c_k are computed for varying α, is present. Can a γ be found that makes the expansion mentioned above work, or can it be approximated?
2. Improved interpolation-extrapolations schemes for approximating $c_k(\theta_{j,n})$ for a large n should be investigated.
3. For piecewise $a(x)$, can expansions of the spectra of A and B in Remark 3 be used to efficiently approximate the spectrum of A_n, or does the matrix R influence the spectrum too much?
4. Can alternative grids be used in the expansion to achieve better results, for example, $\theta_{j,n} = (j-1)\pi/n$.
5. Complex-valued $a(x)$ should also be studied further; it was implemented during the project but omitted due to time constraints where the biggest issue found was dealing with how to correctly order the complex eigenvalues.

References

[1] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer New York, 2007. DOI: 10.1007/b98885. URL: https://doi.org/10.1007/b98885.

[2] Fayyaz Ahmad et al. "Are the eigenvalues of preconditioned banded symmetric Toeplitz matrices known in almost closed form?" In: *Numerical Algorithms* 78.3 (Aug. 2017), pp. 867–893. DOI: 10.1007/s11075-017-0404-z.

[3] Jeff Bezanson et al. "Julia: A fresh approach to numerical computing". In: *SIAM review* 59.1 (2017), pp. 65–98. URL: https://doi.org/10.1137/141000671.

[4] Sven-Erik Ekström, Carlo Garoni, and Stefano Serra-Capizzano. "Are the Eigenvalues of Banded Symmetric Toeplitz Matrices Known in Almost Closed Form?" In: *Experimental Mathematics* 27.4 (May 2017), pp. 478–487. DOI: 10.1080/10586458.2017.1320241. URL: https://doi.org/10.1080/10586458.2017.1320241.

[5] Carlo Garoni and Stefano Serra-Capizzano. *Generalized Locally Toeplitz Sequences: Theory and Applications*. Vol. 1. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-53679-8. URL: https://doi.org/10.1007/978-3-319-53679-8.

[6] Mauricio Barrera et al. "Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic". In: *The Diversity and Beauty of Applied Operator Theory*. Springer International Publishing, 2018, pp. 51–77. DOI: 10.1007/978-3-319-75996-8_2. URL: https://doi.org/10.1007/978-3-319-75996-8_2.

[7] Sven-Erik Ekström. "Matrix-Less Methods for Computing Eigenvalues of Large Structured Matrices". PhD thesis. Uppsala University, 2018. ISBN: 978-91-513-0288-1. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-346735.

[8] Sven-Erik Ekström, Isabella Furci, and Stefano Serra-Capizzano. "Exact formulae and matrix-less eigensolvers for block banded symmetric Toeplitz matrices". In: *BIT Numerical Mathematics* 58.4 (July 2018), pp. 937–968. DOI: 10.1007/s10543-018-0715-z. URL: https://doi.org/10.1007/s10543-018-0715-z.

[9] Sven-Erik Ekström et al. "Are the eigenvalues of the B-spline isogeometric analysis approximation of $-\Delta u = \lambda u$ known in almost closed form?" In: *Numerical Linear Algebra with Applications* 25.5 (July 2018), e2198. DOI: 10.1002/nla.2198. URL: https://doi.org/10.1002/nla.2198.

[10] Carlo Garoni and Stefano Serra-Capizzano. *Generalized Locally Toeplitz Sequences: Theory and Applications*. Vol. 2. Springer International Publishing, 2018. DOI: 10.1007/978-3-030-02233-4. URL: https://doi.org/10.1007/978-3-030-02233-4.

[11] Sven-Erik Ekström. "A matrix-less method to approximate the spectrum and the spectral function of Toeplitz matrices with real eigenvalues". In: *arXiv preprint arXiv:1902.08488* (2019).

[12] Sven-Erik Ekström and Carlo Garoni. "A matrix-less and parallel interpolation–extrapolation algorithm for computing the eigenvalues of preconditioned banded symmetric Toeplitz matrices". In: *Numerical Algorithms* 80.3 (Mar. 1, 2019), pp. 819–848. ISSN: 1572-9265. DOI: 10.1007/s11075-018-0508-0. URL: https://doi.org/10.1007/s11075-018-0508-0.

[13] Sven-Erik Ekström and Paris Vassalos. "A matrix-less method to approximate the spectrum and the spectral function of Toeplitz matrices with complex eigenvalues". In: *arXiv preprint arXiv:1910.13810* (2019).

[14] Julia Math. *Julia Math - DoubleFloats*. URL: https://github.com/JuliaMath/DoubleFloats.jl. (accessed: 10.01.2021).

[15] Julia Programming Language. *Julia Documentation - Standard Library - Linear Algebra*. URL: https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/. (accessed: 19.11.2020).

A Code

The code to generate as well as to interpolate and calculate the eigenvalues from the interpolated values is given below.

```
function compute_c(
    n :: Integer,
    α :: Integer,
    eigfun :: Function,
    a :: Function,
    T :: DataType
    )

    j0 = 1:n
    E  = zeros(T, α + 1, n)
    hs = zeros(real(T), α + 1)
    for kk = 0:α
        nk        = (2∧kk) * (n+1) - 1
        jk        = (2∧kk) * j0
        hs[kk+1]  = convert(T, 1) / (nk+1)
        eTnk      = eigfun(a, nk, T)
        E[kk+1,:] = eTnk[jk]
    end
    V = zeros(T, α + 1, α + 1)
    for ii = 1:α + 1, jj = 1:α + 1
        V[ii, jj] = hs[ii]∧(jj - 1)
    end
    return C=V\E
end
```

```
function localization(x, m)
    b = mod(m, 2)
    v = div(m+b, 2)
    fx = floor(Int64, x);
    cx = ceil(Int64, x);
    if x - fx <= cx - x
        u = (fx - v + 1):(fx + v - b);
    else
```

```
        u = (cx - v + b):(cx + v - 1);
    end
    return u
end
```

```
function interpolate_c(nf :: Integer, C; applicable_ck = 0)
    T = eltype(C)
    α, n0 = size(C)
    if applicable_ck == 0 # if applicable_ck not defined, then, use all data
        applicable_ck=repeat([1 n0],α+1)
    end
    h0 = 1 / (convert(T,n0)+1)
    hf = 1 / (convert(T,nf)+1)
    tf = LinRange(convert(T, pi) / (nf+1), nf * convert(T,pi) / (nf+1), nf)
    CC = zeros(T, α, nf)

    for jj in 1:nf
        for kk in 1:α
            indices = localization(tf[jj] * (n0+1) / pi, α - kk + 1)
            if indices[1] < applicable_ck[kk,1]
                indices = indices .- indices[1] .+ applicable_ck[kk, 1]
            end
            if indices[end] > applicable_ck[kk, 2]
                indices = indices .- indices[end] .+ applicable_ck[kk, 2]
            end
            tt = indices * pi * h0
            ccfit = fit(collect(tt), C[kk, indices], α - kk)
            CC[kk, jj] = ccfit(tf[jj])
        end
    end
    return CC
end
```

```
function get_eigs(CC, nf :: Integer)
    alpha = size(CC, 1) - 1
    hf = 1 / (nf + 1)
    H = ones(1,1) .* (hf .^ (0:α))'
    return (H * CC)[:]
end
```

B Matrix Symmetrisation

For the simplifications of (2.15) we get non-symmetric tridiagonal matrices, however from the Julia manual we see that non-symmetric matrices can not use any of the optimised methods for the *eigvals* function but if we could somehow symmetrise the matrix, we could use the optimised methods available for *eigvals* [15, Special matrices].

Consider the following matrices $\mathcal{S}$ and $\mathcal{S}^{\text{sym}}$:

$$\mathcal{S} = \begin{bmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & c_2 & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ & & & c_{n-1} & a_n \end{bmatrix}, \qquad \mathcal{S}^{\text{sym}} = \begin{bmatrix} a_1 & \sqrt{b_1 c_1} & & & \\ \sqrt{b_1 c_1} & a_2 & \sqrt{b_2 c_2} & & \\ & \sqrt{b_2 c_2} & \ddots & \ddots & \\ & & \ddots & \ddots & \sqrt{b_{n-1} c_{n-1}} \\ & & & \sqrt{b_{n-1} c_{n-1}} & a_n \end{bmatrix},$$

where we will prove that $\mathcal{S}$ and $\mathcal{S}^{\text{sym}}$ have the same characteristic polynomial and thus the same spectrum. First consider the sequence of submatrices of $\mathcal{S}$, which we will denote $\mathcal{K}_j$, that begin in the bottom right corner of $\mathcal{S}$. That is that we have:

$$\mathcal{K}_0 = a_n, \qquad \mathcal{K}_1 = \begin{bmatrix} a_{n-1} & b_{n-1} \\ c_{n-1} & a_n \end{bmatrix}, \qquad \mathcal{K}_2 = \begin{bmatrix} a_{n-2} & b_{n-2} & \\ c_{n-2} & a_{n-1} & b_{n-1} \\ & c_{n-1} & a_n \end{bmatrix}, \qquad \text{etc.},$$

until we finally have that $\mathcal{K}_{n-1} = \mathcal{S}$.
Let us now denote P_j to be the characteristic polynomial of $\mathcal{K}_j$. We can then immediately note that $P_0 = \lambda - a_n$ and $P_1 = (\lambda - a_{n-1})(\lambda - a_n) - b_{n-1}c_{n-1}$, and as for P_2, we can expand along the top row and obtain that $P_2 = (\lambda - a_{n-1})P_1 - b_{n-2}c_{n-2}P_0$. Here, we now make the following claim:

Claim: The characteristic polynomial, P_j, of the matrix $\mathcal{K}_j$, as defined above, can be expressed recursively as:

$$P_j = (\lambda - a_{n-j})P_{j-1} - b_{n-j}c_{n-j}P_{j-2}$$

for $j = 2, 3, ..., n-1$

Proof: We will prove this claim using mathematical induction. It has already been show to hold for $j = 2$, thus we just need to show that the induction step is true. Assume now that the claim holds for some $j = t$, we will now show that

it holds for $j = t+1$. Consider the matrix $\mathcal{K}_{t+1}$:

$$\mathcal{K}_{t+1} = \begin{bmatrix} a_{n-(t+1)} & b_{n-(t+1)} & & & \\ c_{n-(t+1)} & a_{n-t} & b_{n-t} & & \\ & c_{n-t} & \ddots & \ddots & \\ & & \ddots & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{bmatrix},$$

If we now calculate the determinant of $\lambda \mathbf{I}_{t+1} - \mathcal{K}_{t+1}$, by expand along the top row, we get the following where $|\cdot|$ denotes the determinant:

$$P_{t+1} = \det(\lambda \mathbf{I}_{t+1} - \mathcal{K}_{t+1}) = (\lambda - a_{n-(t+1)}) \begin{vmatrix} \lambda - a_{n-t} & -b_{n-t} & & & \\ -c_{n-t} & \ddots & \ddots & & \\ & \ddots & \lambda - a_{n-1} & -b_{n-1} & \\ & & -c_{n-1} & \lambda - a_n & \end{vmatrix}$$
$$+ b_{n-(t+1)} \begin{vmatrix} -c_{n-(t+1)} & -b_{n-t} & & & \\ & \lambda - a_{n-(t-1)} & \ddots & & \\ & & \ddots & \ddots & -b_{n-1} \\ & & & -c_{n-1} & \lambda - a_n \end{vmatrix}$$

For the second matrix we can expand along the first column and thus get:

$$= (\lambda - a_{n-(t+1)}) P_t - b_{n-(t+1)} c_{n-(t+1)} P_{t-1}$$

Thus, we have show that the claim holds for the initial step, $j = 2$ and also for the arbitrary step $j = t+1$, and so by the principals of mathematical induction, the claim is proven. ■

We now have a recursive formulation for the characteristic polynomials for each $\mathcal{K}_j$. We can similarly define $\mathcal{K}_j^{\text{sym}}$ to be the sequence of submatrices of $\mathcal{S}^{\text{sym}}$, that begin in the bottom right corner. For each $\mathcal{K}_j^{\text{sym}}$, we can associate P_j^{sym} to be the characteristic polynomial for that submatrix. By inspection, we can immediately note that $P_0^{\text{sym}} = \lambda - a_n$ and $P_1^{\text{sym}} = (\lambda - a_{n-1})(\lambda - a_n) - b_{n-1}c_{n-1}$ and generally we will have a similar claim as before that $P_j^{\text{sym}} = (\lambda - a_{n-j})P_{j-1} - b_{n-j}c_{n-j}P_{j-2}$ which can be proven identically as the previous proof using mathematical induction. We thus have that $P_j = P_j^{\text{sym}}$ for all $j = 0, 1, ..., n-1$ but P_{n-1} is the characteristic polynomial for $\mathcal{K}_{n-1} = \mathcal{S}$ and P_{n-1}^{sym} is the characteristic polynomial for $\mathcal{K}_{n-1}^{\text{sym}} = \mathcal{S}^{\text{sym}}$, thus we have that $\mathcal{S}$ and $\mathcal{S}^{\text{sym}}$ have the same characteristic polynomial and thus the same spectrum.

Our method for calculating the eigenvalues of the simplifications is therefore to

construct the $\mathcal{S}^{\text{sym}}$ matrix for (2.17). We thus get:

$$G'_n = \begin{bmatrix} 2a_1 & \sqrt{a_1 a_2} & & & \\ \sqrt{a_1 a_2} & 2a_2 & \sqrt{a_2 a_3} & & \\ & \sqrt{a_2 a_3} & \ddots & \ddots & \\ & & \ddots & \ddots & \sqrt{a_{n-1} a_n} \\ & & & \sqrt{a_{n-1} a_n} & 2a_n \end{bmatrix}, \tag{B.1}$$

Computationally this adds $\mathcal{O}(n)$ operations for the off diagonal multiplications but the optimisations for *eigvals*.