UPPSALA
UNIVERSITET

IT 21 082
Examensarbete 30 hp
September 2021

A Matrix-less Method for Approximating the Eigenvectors of Toeplitz-like Matrices

David Meadon

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

A Matrix-less Method for Approximating the Eigenvectors of Toeplitz-like Matrices

David Meadon

□Matrix-less methods (MLM) have successfully been used to efficiently approximate the eigenvalues of certain classes of structured matrices. Specifically, the method has been used to approximate the eigenvalues of Toeplitz and Toeplitz-like matrices. The method exploits the inherent structure of the eigenvalues, which is maintained when the matrix size changes, and thus can use the eigenvalues of a set of smaller matrices to approximate the eigenvalues for much larger matrices.
□□
□In this thesis, we investigate whether there exists a similar structure for the eigenvectors for some of these matrices and if we can apply an MLM such that we can efficiently approximate the eigenvectors of Toeplitz(-like) matrices. We here study symmetric Toeplitz(-like) matrices generated by monotone symbols. Specifically, we investigate the use of this method on four different types matrix sequences: the (1) Laplacian matrix in one dimension, the closely related (2) bi-Laplacian matrix, (3) Isogeometric analysis discretisation matrices and also (4) a `full' matrix related to the discretisation of fractional diffusion.
□
□For the first three types of investigated matrix sequences, it is found that the matrix-less method is able to well-approximate the eigenvectors, where as for the fourth case (fractional derivative related matrix sequence) where it does not work, we discuss some potential adjustments that may allow for MLM to work in that case as well. We also discuss why the method cannot be immediately used for Toeplitz matrices with non-monotone symbols. We conclude the thesis with suggestions for future avenues of research including how to possibly deal with matrix sequences generated by non-monotone symbols.

Handledare: Sven-Erik Ekström
Ämnesgranskare: Maya Neytcheva
Examinator: Salman Toor
IT 21 082
Tryckt av: Reprocentralen ITC

Contents

I

Introduction

Consider the linear system

$$Av = \lambda v \qquad\qquad \lambda \in \mathbb{C},\ v \in \mathbb{C}^n,\ A \in \mathbb{C}^{n\times n}$$

which is generally known as the eigenvalue equation, where λ is an eigenvalue of the matrix A and v is the associated eigenvector. The calculation of eigenvalues and their corresponding eigenvectors is an important area in many different fields. For example they may be the solution to a physical problem (e.g. see [5], [11], [34]) or used to evaluate the stability and convergence of different numerical methods [27] or to construct preconditioners for solving linear systems using deflation methods [10], [28]. Most uniform discretisations of partial differential equations give rise to matrices of a specific form, namely Toeplitz and Toeplitz-like matrices.

Substantial research has been done on the approximation of the eigenvalues of Toeplitz(-like) matrices [21]–[23], [30], [33], where we will highlight the theory of Generalized Locally Toeplitz (GLT) sequences. More specifically, the matrix-less method (MLM) which is derived from the theory of GLT sequences and has been shown to be an efficient method for the approximation of the eigenvalues for different classes of Toeplitz-like matrices, e.g. [14], [16], [17], [19], [20]. Specifically, MLM is able to use the eigenvalues for a few smaller matrices (e.g. 3 matrices of size $\mathcal{O}(100)$) to approximate the eigenvalues for a larger matrix in the same matrix sequence.

Thus far, MLM has only been used for the approximation of eigenvalues, however in this thesis, we will consider whether we can develop an MLM to approximate the eigenvectors of real symmetric Toeplitz matrices, which will thus be real and also have real associated eigenvalues $\lambda_i \in \mathbb{R}$. That is, can we use the structure of the eigenvectors that is maintained in matrices in the same matrix sequence, to use the eigenvectors of a few smaller matrices to approximate those of a larger matrix, in the same matrix sequence.

In this first section we provide an overview of the before mentioned theory on GLT sequences and MLM. Then we conclude the section by detailing why this method which has so far been used to approximate eigenvalues, can also be used to approximate eigenvectors and fully describing the problem we will investigate in this thesis. In the second section we outline the results from the numerical experiments on using the derived MLM for four different matrix sequences. Then, we discuss considerations about the results, before finally concluding the report with suggestions for future research.

1.1 Generalized Locally Toeplitz (GLT) Sequences

The theory of GLT sequences is a well researched topic, e.g. [2], [3], [21]–[23]. We begin by stating the definition of a Toeplitz matrix.

Definition 1.1. A matrix $A_n \in \mathbb{C}^{n\times n}$ is called ***Toeplitz*** if it has constant diagonals, that is,

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

Furthermore, for each Toeplitz matrix A_n we can associate a function $f(\theta)$, called a symbol.

Definition 1.2. Consider a function $f \in L^1(-\pi, \pi)$, where its Fourier coefficients are given by:

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

Then the Toeplitz matrix of size n associated with f is given by,

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

We may then consider $\{T_n(f)\}_n$ which is the sequence of Toeplitz matrices of difference sizes n which are all associated with the function f. We then call f the ***symbol*** of $T_n(f)$ which generates the Toeplitz matrix sequence $\{T_n(f)\}_n$.

Note. If the Toeplitz matrix is banded, deriving the symbol is straight forward. However, if the Toeplitz matrix is not banded, the symbol may often be derived by the study of the Fourier coefficients, given by the generated matrices in the sequence; e.g., see numerical example 4. If the symbol is known though, then the matrix of any size in the matrix sequence associated with the symbol can be generated.

We now provide two examples of banded Toeplitz matrices as well as their associated symbols.

Example 1.1.1. Consider the discrete ***Laplacian*** matrix in one dimension of size n:

$$T_n(f) = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix} = \begin{bmatrix} \hat{f}_0 & \hat{f}_{-1} & & & \\ \hat{f}_1 & \hat{f}_0 & \hat{f}_{-1} & & \\ & \hat{f}_1 & \ddots & \ddots & \\ & & \ddots & \ddots & \hat{f}_{-1} \\ & & & \hat{f}_1 & \hat{f}_0 \end{bmatrix},$$

where we have that $\hat{f}_1 = -1, \hat{f}_0 = 2, \hat{f}_{-1} = -1$ and $\hat{f}_k = 0$ for all other k, thus this matrix will have associated symbol:

$$\begin{aligned} f(\theta) &= \sum_{k=-\infty}^{\infty} \hat{f}_k \mathrm{e}^{k\mathbf{i}\theta} \\ &= -\mathrm{e}^{-\mathbf{i}\theta} + 2 - \mathrm{e}^{\mathbf{i}\theta} \\ &= 2 - 2\cos(\theta). \end{aligned}$$

Example 1.1.2. The ***bi-Laplacian*** matrix of size n:

$$T_n(f) = \begin{bmatrix} 6 & -4 & 1 & & \\ -4 & 6 & -4 & \ddots & \\ 1 & -4 & \ddots & \ddots & 1 \\ & \ddots & \ddots & \ddots & -4 \\ & & 1 & -4 & 6 \end{bmatrix} = \begin{bmatrix} \hat{f}_0 & \hat{f}_{-1} & \hat{f}_{-2} & & \\ \hat{f}_1 & \hat{f}_0 & \hat{f}_{-1} & \ddots & \\ \hat{f}_2 & \hat{f}_1 & \ddots & \ddots & \hat{f}_{-2} \\ & \ddots & \ddots & \ddots & \hat{f}_{-1} \\ & & \hat{f}_2 & \hat{f}_1 & \hat{f}_0 \end{bmatrix},$$

is a discretisation of the fourth derivative and in this case we have that $\hat{f}_2 = 1, \hat{f}_1 = -4, \hat{f}_0 = 6, \hat{f}_{-1} = -4, \hat{f}_{-2} = 1$ and $\hat{f}_k = 0$ for all other k, meaning the associated symbol is:

$$\begin{aligned} f(\theta) &= \sum_{k=-\infty}^{\infty} \hat{f}_k \mathrm{e}^{k\mathbf{i}\theta} \\ &= \mathrm{e}^{-2\mathbf{i}\theta} - 4\mathrm{e}^{-\mathbf{i}\theta} + 6 - 4\mathrm{e}^{\mathbf{i}\theta} + \mathrm{e}^{2\mathbf{i}\theta} \\ &= 6 - 8\cos(\theta) + 2\cos(2\theta) \\ &= (2 - 2\cos(\theta))^2. \end{aligned}$$

Note. The symbols of these matrices are related to the asymptotic behaviour of the matrix sequence and as such they will also be the symbols for matrices which have been perturbed in the corners, for example by boundary conditions. These matrices are called Toeplitz-like and are discussed further at the end of the section. It is also important to note that the relation between symbols and matrices is two sided, such that you may begin with a symbol and then generate a matrix from that symbol or we may (as in Example 1.1.1 and Example 1.1.2) describe a symbol from the matrix.

The spectral behaviour of these matrix sequences can be described by the theory of GLT sequences; see [21]–[23]. If $\{A_n\}_n$ is a GLT sequence with symbol

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

That is,

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

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

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

$$\lambda_j(A_n) = f(\theta_{j,n}) + E_{j,n,0}, \tag{1.1}$$

where again typically the error term is $E_{j,n,0} = \mathcal{O}(h)$. We here provide some further properties of these relations as described in [22]:

GLT 1. If $\{A_n\}_n \sim_{\text{GLT}} f$ then $\{A_n\}_n \sim_\sigma f$. If $\{A_n\}_n \sim_{\text{GLT}} f$ and the matrices A_n are Hermitian then $\{A_n\}_n \sim_\lambda f$.

GLT 2. If $\{A_n\}_n \sim_{\text{GLT}} f$ and $A_n = X_n + Y_n$, where

- every X_n is Hermitian,
- $\|X_n\|, \|Y_n\| \leq C$ for some constant C independent of n,
- $n^{-1}\|Y_n\|_1 \to 0$,

then $\{A_n\}_n \sim_\lambda f$.

GLT 3. We have

- $\{T_n(h)\}_n \sim_{\text{GLT}} f(x, \theta) = h(\theta)$ if $h \in L^1([-\pi, \pi])$,
- $\{D_n(a)\}_n \sim_{\text{GLT}} f(x, \theta) = a(x)$ if $a \ : \ [0, 1] \to \mathbb{C}$ is continuous a.e.,
- $\{Z_n\}_n \sim_{\text{GLT}} f(x, \theta) = 0$ if and only if $\{Z_n\}_n \sim_\sigma 0$.

GLT 4. If $\{A_n\}_n \sim_{\text{GLT}} f$ and $\{B_n\}_n \sim_{\text{GLT}} g$ then

- $\{A_n^*\}_n \sim_{\text{GLT}} \bar{f}$
- $\{\alpha A_n + \beta B_n\}_n \sim_{\text{GLT}} \alpha f + \beta g$ for all $\alpha, \beta \in \mathbb{C}$
- $\{A_n B_n\}_n \sim_{\text{GLT}} fg$

GLT 5. If $\{A_n\}_n \sim_{\text{GLT}} f$ and $f \neq 0$ a.e. then $\{A_n^\dagger\}_n \sim_{\text{GLT}} f^{-1}$.

GLT 6. If $\{A_n\}_n \sim_{\text{GLT}} f$ and each A_n is Hermitian, then $\{h(A_n)\}_n \sim_{\text{GLT}} h(f)$ for every continuous function $h \ : \ \mathbb{C} \to \mathbb{C}$.

Note. We may also consider larger classes of matrix sequences by considering zero distributed sequences. That is consider the Toeplitz-like matrix $A_n = T_n(f) + R_n + N_n$ where $T_n(f)$ is the pure Toeplitz matrix generated from the symbol f, while R_n and N_n are examples of Z_n in **GLT 3**. Specifically R_n may be a low-rank matrix which may be due to boundary conditions and N_n is a small-norm matrix, for example of the form $hT_n(g)$, which goes to the zero matrix as n goes to infinity. Then by the linearity in **GLT 4**, we have that $\{A_n\}_n \sim_{\text{GLT}} f$.

1.2 The Matrix-less Method (MLM)

A class of methods, here denoted *matrix-less*, first introduced in [8, pp. 1329] and then [18], have been extended to a larger class of Hermitian matrices, see e.g. [14] and lately also to non-Hermitian matrices [19], [20]. The GLT eigenvalue approximation

$$\lambda_j(A_n) = f(\theta_{j,n}) + E_{j,n,0} \approx f(\theta_{j,n}) \tag{1.2}$$

works when we can find or define the symbol $f(\theta)$ for the sequence $\{A_n\}_n$, but the error, $E_{j,n,0}$, which depends on the regularity of f and is usually of size $\mathcal{O}(h)$ might be prohibitively large for applications.

In MLM we assume and exploit an asymptotic expansion of the form

$$\begin{aligned}\lambda_j(A_n) &= f(\theta_{j,n}) + E_{j,n,0}\\ &= f(\theta_{j,n}) + \sum_{k=1}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha}\\ &= \sum_{k=0}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha},\end{aligned} \tag{1.3}$$

where by definition $f(\theta) = c_0(\theta)$. The $c_k(\theta)$'s are typically unknown functions which MLM can approximate by samplings $\tilde{c}_k(\theta_{j,n_0})$. The parameters $\alpha \in \mathbb{Z}_+$ and $n_0 \in \mathbb{Z}_+$ are chosen by the user, where typical values are around $\alpha = 3$ and $n_0 = \mathcal{O}(10^2 \sim 10^3)$. The resulting error is $E_{j,n,\alpha} = \mathcal{O}(h^{\alpha+1})$.

The basic idea of MLM is to numerically calculate the eigenvalues for $\alpha + 1$ small-sized matrices using standard numerical eigen-solvers, approximate the functions c_k for $k = 0, \dots, \alpha$ on the grid θ_{j,n_0}, and then use these to approximate the spectrum for a matrix A_n where $n \gg n_0$, by interpolation-extrapolation on a grid $\theta_{j,n}$.

First the matrices A_{n_k} of sizes $n_k = 2^k(n_0 + 1) - 1$ for $k = 0, \dots, \alpha$ are constructed and their eigenvalues are computed using a standard numerical eigensolver; we use Julia's [6] `eigvals` supplemented with `GenericSchur.jl` [31] or `GenericLinearAlgebra.jl` [25] mainly due to the support of high precision datatypes. After sorting the eigenvalues in non-decreasing order for each level k, where we have assumed the matrix has a monotone symbol, and then choose every 2^k-th eigenvalue to construct a matrix $E \in \mathbb{C}^{\alpha+1\times n_0}$

$$E = \begin{bmatrix} \lambda_1(A_{n_0}) & \lambda_2(A_{n_0}) & \lambda_3(A_{n_0}) & \dots & \lambda_{n_0}(A_{n_0}) \\ \lambda_2(A_{n_1}) & \lambda_4(A_{n_1}) & \lambda_6(A_{n_1}) & \dots & \lambda_{2n_0}(A_{n_1}) \\ \vdots & \vdots & \vdots & \ddots & \\ \lambda_{2^\alpha}(A_{n_\alpha}) & \lambda_{2\cdot 2^\alpha}(A_{n_\alpha}) & \lambda_{3\cdot 2^\alpha}(A_{n_\alpha}) & \dots & \lambda_{n_0\cdot 2^\alpha}(A_{n_\alpha}) \end{bmatrix}. \tag{1.4}$$

This choice of grids and the corresponding subsets of eigenvalues for each level is presented in Figure 1.1.

Once we have calculated the eigenvalues that the estimation is based on, we

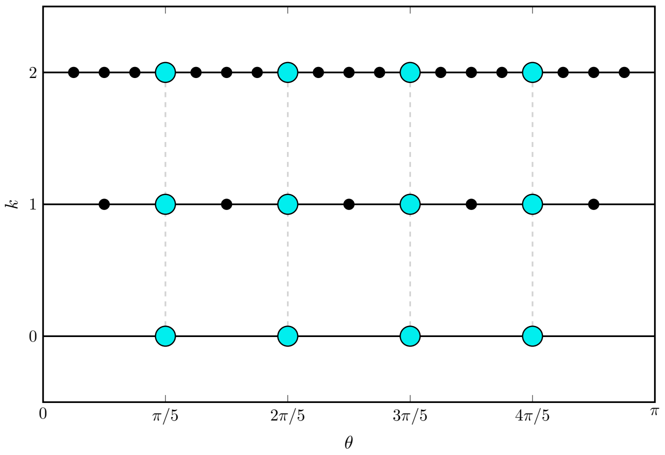


Figure 1.1: Grids for $\alpha = 2$ and $n_0 = 4$.

then use a specific Vandermonde matrix:

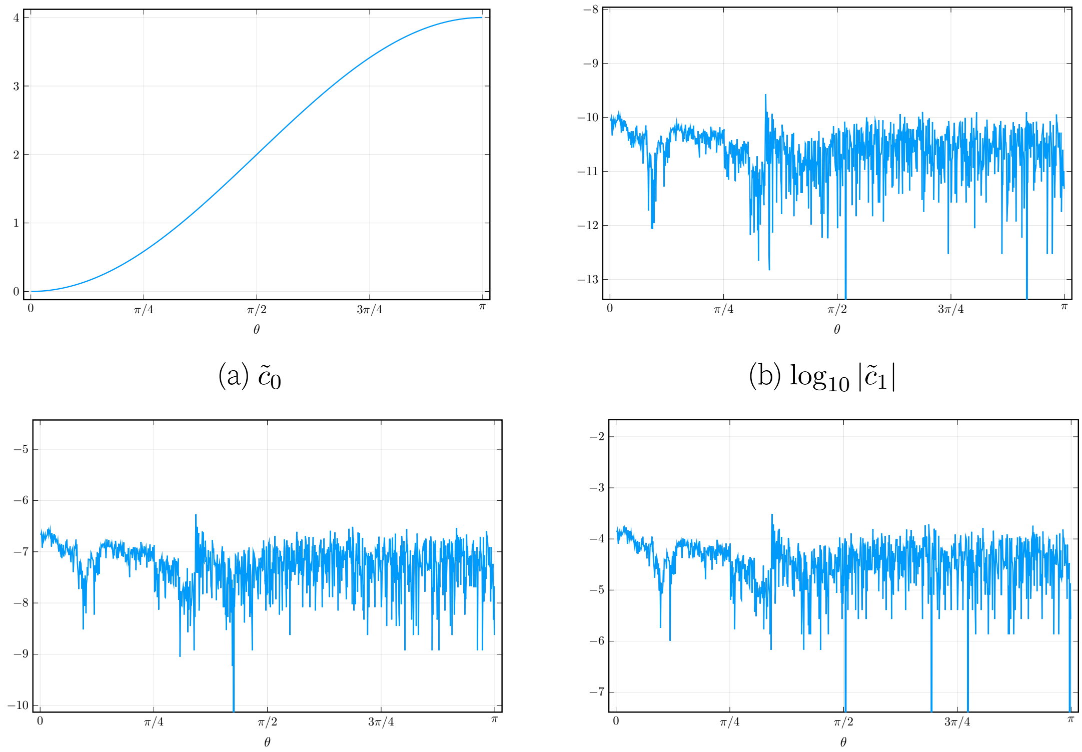
$$V = \begin{bmatrix} 1 & h_0 & h_0^2 & h_0^3 & \dots & h_0^\alpha \\ 1 & h_1 & h_1^2 & h_1^3 & \dots & h_1^\alpha \\ 1 & h_2 & h_2^2 & h_2^3 & \dots & h_2^\alpha \\ 1 & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & h_\alpha & h_\alpha^2 & h_\alpha^3 & \dots & h_\alpha^\alpha \end{bmatrix}, \quad h_k = \frac{1}{1+n_k}, \tag{1.5}$$

and solve the linear system $E = VC$ to find our approximation $\widetilde{C}$:

$$\widetilde{C} = \begin{bmatrix} \tilde{c}_0\left(\theta_{1,n_0}\right) & \tilde{c}_0\left(\theta_{2,n_0}\right) & \cdots & \tilde{c}_0\left(\theta_{n_0,n_0}\right) \\ \tilde{c}_1\left(\theta_{1,n_0}\right) & \tilde{c}_1\left(\theta_{2,n_0}\right) & \cdots & \tilde{c}_1\left(\theta_{n_0,n_0}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_\alpha\left(\theta_{1,n_0}\right) & \tilde{c}_\alpha\left(\theta_{2,n_0}\right) & \cdots & \tilde{c}_\alpha\left(\theta_{n_0,n_0}\right) \end{bmatrix}, \tag{1.6}$$

and then can use this approximation in an interpolation–extrapolation scheme [17] to compute $\lambda_j(A_{n_f}) \approx \tilde{\lambda}_{j,n_f} = \sum_{k=0}^{n} h^k \tilde{c}_k(\theta_{j,n_f})$. Extending Examples 1.1.1 and 1.1.2, we here perform MLM on the matrices in those examples.

Example 1.2.1. Recall the discrete Laplacian matrix in Example 1.1.1 where we would now like to approximate the eigenvalues for this matrix of size $n_f = 100000$ using a number of smaller matrices, with the smallest being of size $n_0 = 100$. Note that for the Laplacian matrix we have that the eigenvalues are exactly given by sampling the symbol, $f(\theta) = 2 - 2\cos(\theta)$, with the standard grid $\theta_{j,n} = \frac{j\pi}{n+1}$, that is $\lambda_j(T_n(f)) = f(\theta_{j,n})$, and so we would expect that $c_k = 0$ for $k > 0$ [14]. Figure 1.2 shows the approximated expansion functions obtained by following the steps described above, using $\alpha = 3$. Figure 1.3 shows the error of the extrapolated eigenvalues.



(a) $\tilde{c}_0$

(b) $\log_{10}|\tilde{c}_1|$

(c) $\log_{10}|\tilde{c}_2|$

(d) $\log_{10}|\tilde{c}_3|$

Figure 1.2: Approximated $\tilde{c}_k$ of the Laplacian matrix for $k = 0 \ldots 3$ (n_0, α, Data type) = (1000, 3, `Double64`).

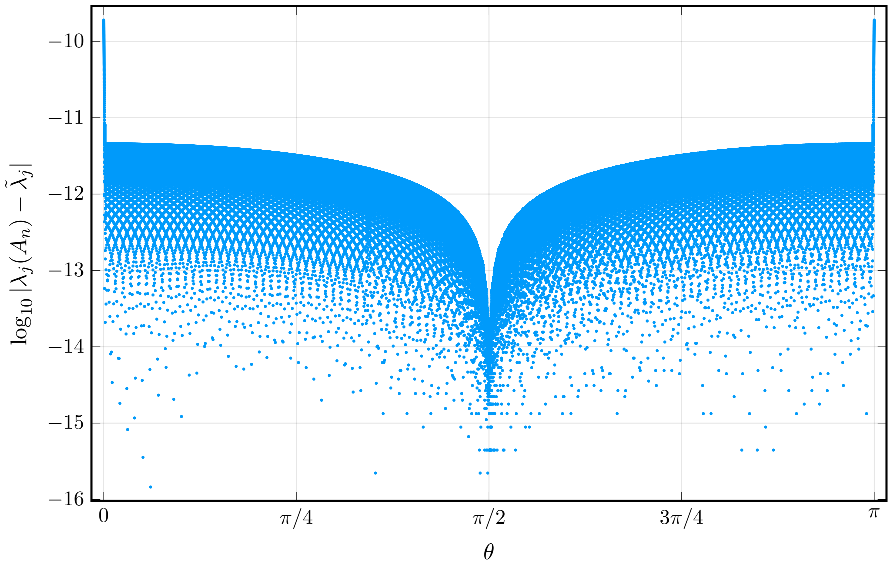
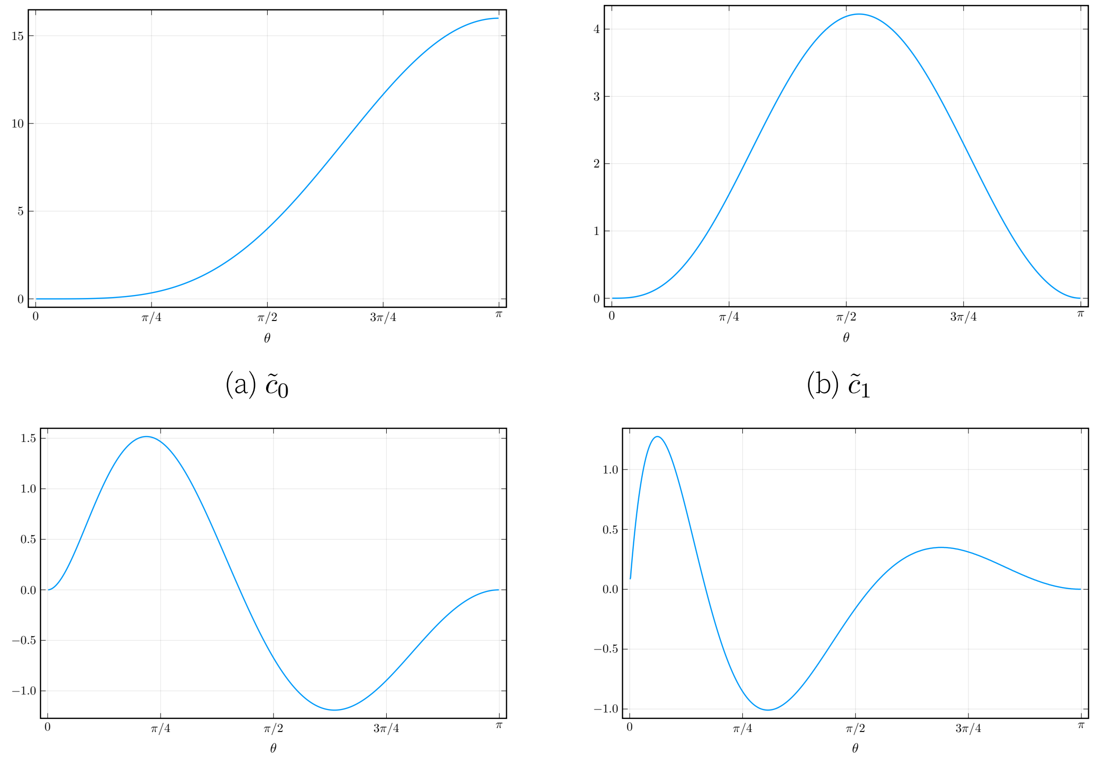


Figure 1.3: Errors $\log_{10}|\lambda_j(A_{n_f}) - \tilde{\lambda}_{j,n_f}|$ of the Laplacian matrix for $n_f = 100000$ and (n_0, α, Data type) = (1000, 3, `Double64`).

Note. Figure 1.2 shows that the expansion functions for the eigenvalues of the Laplacian are near zero, this is why the $\tilde{c}_k$, for $k > 0$, are plotted logarithmically and so we do indeed observe the expected result.

Example 1.2.2. Similar to the previous example, we here will consider the bi-Laplacian matrix encountered in Example 1.1.2 where we would also like to approximate the eigenvalues for this matrix of size $n_f = 100000$ using a number of smaller matrices, with the smallest being of size $n_0 = 100$. Figure 1.4 shows the approximated expansion functions, again using $\alpha = 3$. Figure 1.5 shows the error of the extrapolated eigenvalues.



(a) $\tilde{c}_0$

(b) $\tilde{c}_1$

(c) $\tilde{c}_2$

(d) $\tilde{c}_3$

Figure 1.4: Approximated $\tilde{c}_k$ of the bi-Laplacian matrix for $k = 0 \ldots 3$ (n_0, α, Data type) = (1000, 3, `Double64`).

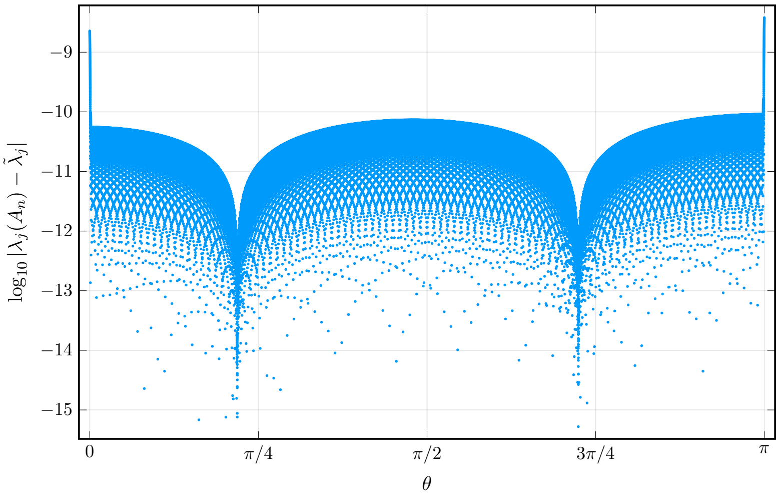


Figure 1.5: Errors $\log_{10} |\lambda_j(A_{n_f}) - \tilde{\lambda}_{j,n_f}|$ of the bi-Laplacian matrix for $n_f = 100000$ and (n_0, α, Data type) = (1000, 3, `Double64`).

Note. In contrast to the previous example, we do not have that the eigenvalues of the bi-Laplacian are given by the sampling of the symbol using the used grid and thus we do see that the approximated expansion functions, $\tilde{c}_k$, are non-zero in this case as shown in Figure 1.4.

So far, MLM have successfully been employed for a wide class of matrices A_n, most importantly,

- $A_n = T_n(f)$ and $\{A_n\}_n \sim_\lambda f$, f Hermitian [18],
- $A_n = T_n(g)^{-1}T_n(f)$ and $\{A_n\}_n \sim_\lambda f/g$ [1],
- $A_n = T_n(f)$ and $\{A_n\}_n \not\sim_\lambda f$, $\{A_n\}_n \sim_\lambda c_0$, f non-Hermitian [19], [20],
- $A_n = T_n(f) + R_n$ and $\{A_n\}_n \sim_\lambda f$ where R_n is a low-rank matrix [15],
- $A_{\mathbf{n}} = T_{\mathbf{n}}(\mathbf{f})$ and $\{A_n\}_n \sim_\lambda \mathbf{f}$ where $\mathbf{f}$ is matrix-valued [16].

Thus, MLM is a very powerful method for approximating eigenvalues. We will now show that it can also be used for the approximation of eigenvectors.

1.3 Problem Description

So far, all the theory we have considered concerns approximating the *eigenvalues* of Toeplitz(-like) matrices. In a general sense, MLM uses the fact that Toeplitz matrices in a sequence of matrices share the same symbol that describes the eigenvalues. This is shown in Figure 1.6 where we have plotted the eigenvalues of the Laplacian matrix for a number of different sizes.

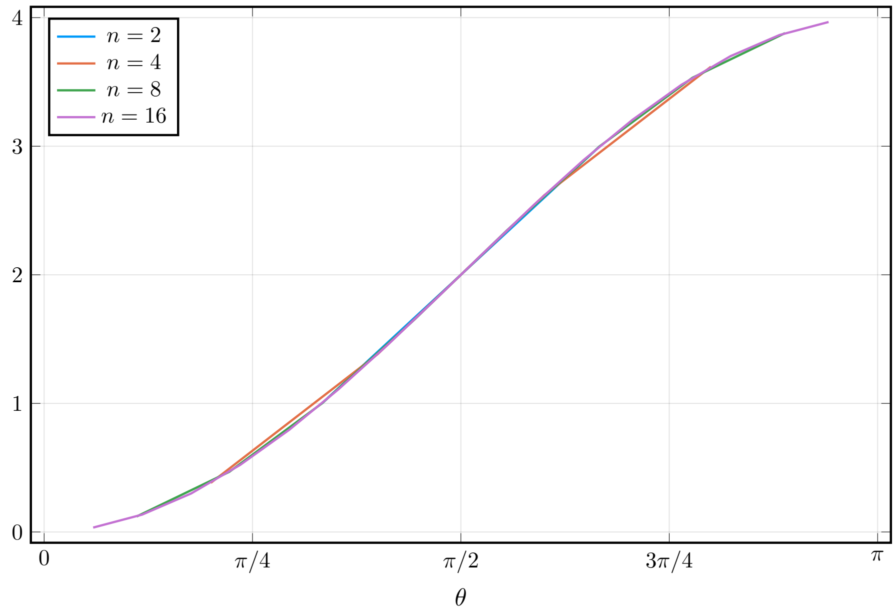
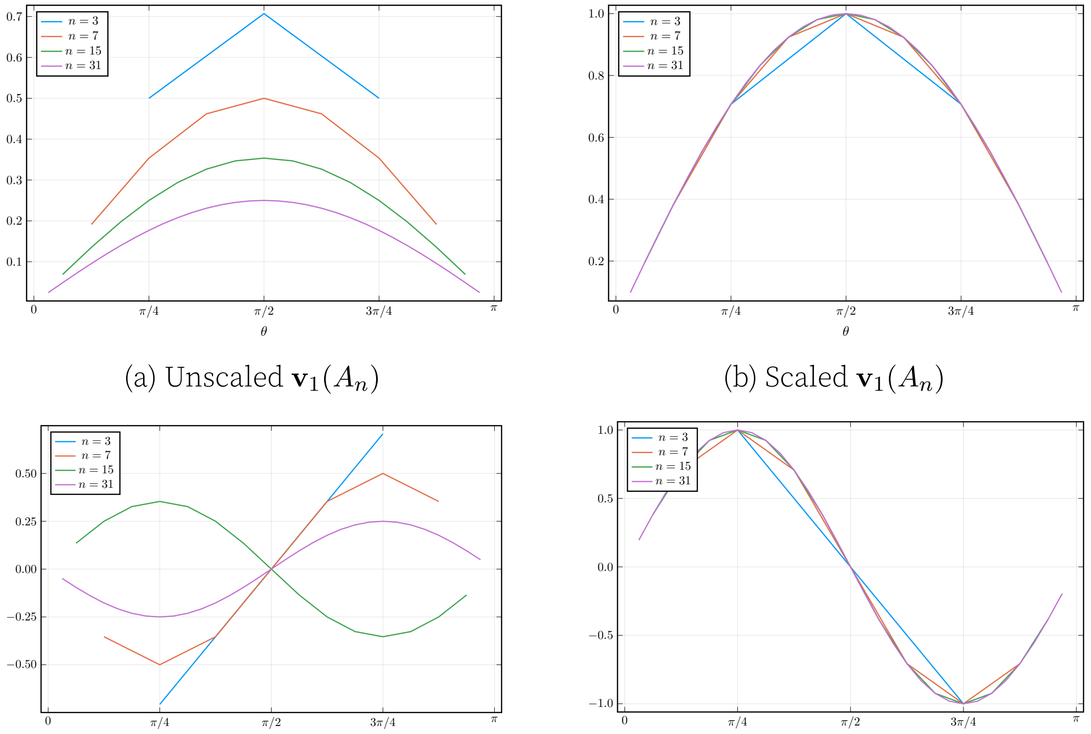


Figure 1.6: Eigenvalues of Laplacian matrix for n = 2, 4, 8 and 16.

Similarly, we would assume there to be some '*eigenvector* symbol' that is maintained for eigenvectors of matrices in the same matrix sequence. Since *eigenvectors* are not unique, and the current standard is for eigen-solvers to return eigenvectors with unit 2-norm, we would need to apply a suitable scaling to the eigenvectors when changing sizes. In Figure 1.7, we see that changing the size of the matrices retains a certain structure of the eigenvectors similar to that of the eigenvalues, after the eigenvectors have been suitably scaled, and it is this structure which MLM can exploit and allow us to approximate the eigenvectors. A 'correct' scaling of the eigenvectors, which is needed since it is numerically very sensitive for MLM to work, is detailed later in the thesis, but we should highlight that we have chosen our scaling such that the eigenvectors may be compared to a sampling of the sine function using an equispaced grid ($\sin(j\theta_{i,n})$), that is we may compare it to the non-normalised Discrete Sine Transform (DST) [9].



(a) Unscaled $\mathbf{v}_1(A_n)$

(b) Scaled $\mathbf{v}_1(A_n)$

(c) Unscaled $\mathbf{v}_2(A_n)$

(d) Scaled $\mathbf{v}_2(A_n)$

Figure 1.7: Comparison of structure for scaled and unscaled eigenvectors of the Laplacian matrix of different sizes.

Thus, in this report we investigate the following hypothesis.

Working Hypothesis. *For eigenvectors of symmetric Toeplitz and Toeplitz-like matrices, there exists an expansion of the form:*

$$\mathbf{v}_{i,j}(A_n) = c_0^{[j]}(\theta_{i,n}) + \sum_{k=1}^{\alpha} h^k c_k^{[j]}(\theta_{i,n}) + E_{i,n,\alpha}^{[j]}$$

where $\mathbf{v}_{i,j}(A_n)$ is the ith component of the jth eigenvector of a symmetric Toeplitz or Toeplitz-like matrix A_n of size n, $\theta_{i,n} = \frac{i\pi}{n+1}$ is an equispaced grid, the $c_k^{[j]}$'s are the ex-

pansion functions for the jth eigenvector and $E^{[j]}_{i,n,\alpha}$ is some error term which goes to zero as α and n goes to infinity.

Note. Some notes about the above hypothesis:

- The $c^{[j]}_k$ are typically unknown, and must be numerically approximated. In Section 2.1 we propose an algorithm to compute the approximations of the expansion functions $\tilde{c}^{[j]}_k$.
- It is known that the above expansion will not be completely true in general (e.g. see [4] in the eigenvalue version) but we assume that numerically it does work, except for possibly a few outlier elements.
- The method as seen in Example 1.2.1 was able to approximate all the *eigenvalues* of the matrix at once, however as can be seen in the above expansion, the method will only be able to approximate a single eigenvector as a time. Moreover since we begin with a matrix of size n_0, we would be unable to approximate more than the first n_0 eigenvectors of the larger matrix as we will not have the required information for the eigenvectors of $j > n_0$.

II

Main Results

We now investigate whether the working hypothesis holds for a number of different matrix sequences, that is if we are able to use MLM to approximate the expansion functions, which from here out the approximations will be denoted $c_k^{[j]}$. We primarily consider the following four different cases:

1. The Laplacian matrix which was defined in Example 1.1.1 and then it was further shown in Example 1.2.1 that the eigenvalues of this matrix were able to be well approximated using MLM. This matrix has known eigenvectors, being exactly given by DST [9] sampled using the standard grid, and so we would expect it to behave similarly as in the eigenvalue case and have $c_k = 0$ for $k > 0$.

2. The bi-Laplacian matrix already discussed in Examples 1.1.2 and 1.2.2 where we had that $c_k \neq 0$ for $k > 0$ for the eigenvalues.

3. A not purely Toeplitz, but Toeplitz-like matrix sequence which arise from using Isogeometric analysis (IGA).

4. A not banded but full Toeplitz matrix sequence which is related to fractional derivatives.

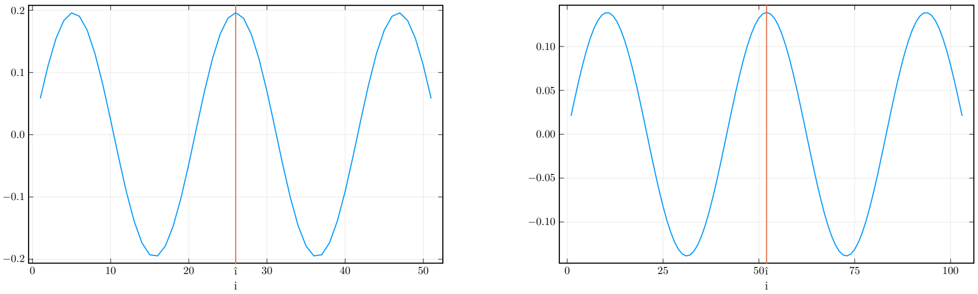
We first outline the algorithm that will be use to approximate the expansion functions for the eigenvectors.

2.1 Algorithm

In order to approximate the jth eigenvector of a symmetric Toeplitz(-like) matrix A_{n_f}, the following main steps must be performed:

1. **Generate Eigenvectors**: Using an odd n_0, the different sized matrices A_{n_k} of sizes $n_k = 2^k(n_0 + 1) - 1$ for $k = 0, \ldots, \alpha$ are first constructed. For each of these matrices, the jth eigenvector is computed using a standard numerical eigenvector solver (Here the jth eigenvector is the eigenvector associated with the jth eigenvalue of the sorted eigenvalues in non-decreasing order) and so we denote these vectors as $\mathbf{v}_j(A_{n_k})$.

2. **Scale Eigenvectors**:For the jth eigenvector of the smallest matrix, of size n_0, we find the index of its first extremum after or at the middle element, and denote this index as î. For each k, we then divide eigenvector $\mathbf{v}_j(A_{n_k})$ by its value at index 2^kî. There is a small difference depending whether we are approximating an even indexed or odd indexed eigenvector:

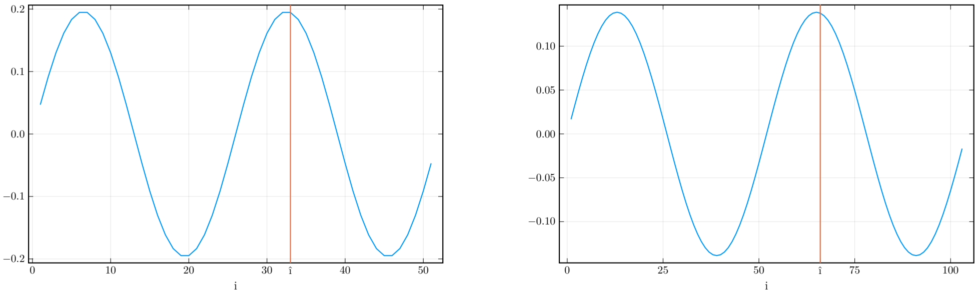


(a) $n = 51$

(b) $n = 103$

Figure 2.1: Eigenvector 5 of the Laplacian matrix showing the point î.

- The odd index jth eigenvectors, which are symmetric, always attain an extremum at their central element, thus they are scaled by dividing by their central element, that is by element $\mathbf{v}_{(n_k+1)/2,j}(A_{n_k})$ (where n_k is always odd for an odd n_0). See Figure 2.1.
- Scaling by the middle element is not possible for the even index jth eigenvectors, which are skew-symmetric, as they are (approximately) zero at their central element (odd n_0) thus the even eigenvectors are scaled by using the above detailed method, where we use the index, î, of the extremum after the central element. See Figure 2.2.



(a) $n = 51$

(b) $n = 103$

Figure 2.2: Eigenvector 4 of the Laplacian matrix showing the point î.

- The eigenvectors are further scaled by a ±1 so as to have the same form as the DST [9], $\sin(j\theta_{j,n})$. See Figure 2.3.

The scaling of the eigenvectors, shown in Figure 1.7, allows us to use the structure of the Topelitz(-like) matrices of different sizes, which we cannot do with the 'plain' eigenvectors from the standard numerical solvers as they return the eigenvectors with unit 2-norm.

3. **Generate $E^{[j]}$ matrix**: We are then able to analogously construct the matrix $E^{[j]} \in \mathbb{C}^{\alpha+1\times n_0}$ but now we have:

$$E^{[j]} = \begin{bmatrix} \mathbf{v}_{1,j}(A_{n_0}) & \mathbf{v}_{2,j}(A_{n_0}) & \mathbf{v}_{3,j}(A_{n_0}) & \cdots & \mathbf{v}_{n_0,j}(A_{n_0}) \\ \mathbf{v}_{2,j}(A_{n_1}) & \mathbf{v}_{4,j}(A_{n_1}) & \mathbf{v}_{6,j}(A_{n_1}) & \cdots & \mathbf{v}_{2n_0,j}(A_{n_1}) \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{v}_{2^\alpha,j}(A_{n_\alpha}) & \mathbf{v}_{2\cdot 2^\alpha,j}(A_{n_\alpha}) & \mathbf{v}_{3\cdot 2^\alpha,j}(A_{n_\alpha}) & \cdots & \mathbf{v}_{n_0\cdot 2^\alpha,j}(A_{n_\alpha}) \end{bmatrix}. \quad (2.1)$$

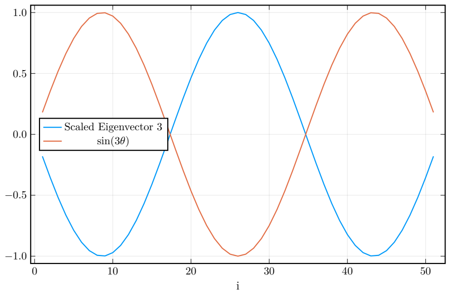


Figure 2.3: Eigenvector 3 of the Laplacian matrix after scaling by the central element and compared with $\sin(3\theta)$

4. **Solve Linear System**: Using the same Vandermonde matrix as in 1.5, we solve the linear system $E^{[j]} = VC^{[j]}$ to find our approximation $\widetilde{C}^{[j]}$ in (1.6) but now it will be for eigenvector $\mathbf{v}_j(A_{n_f})$.

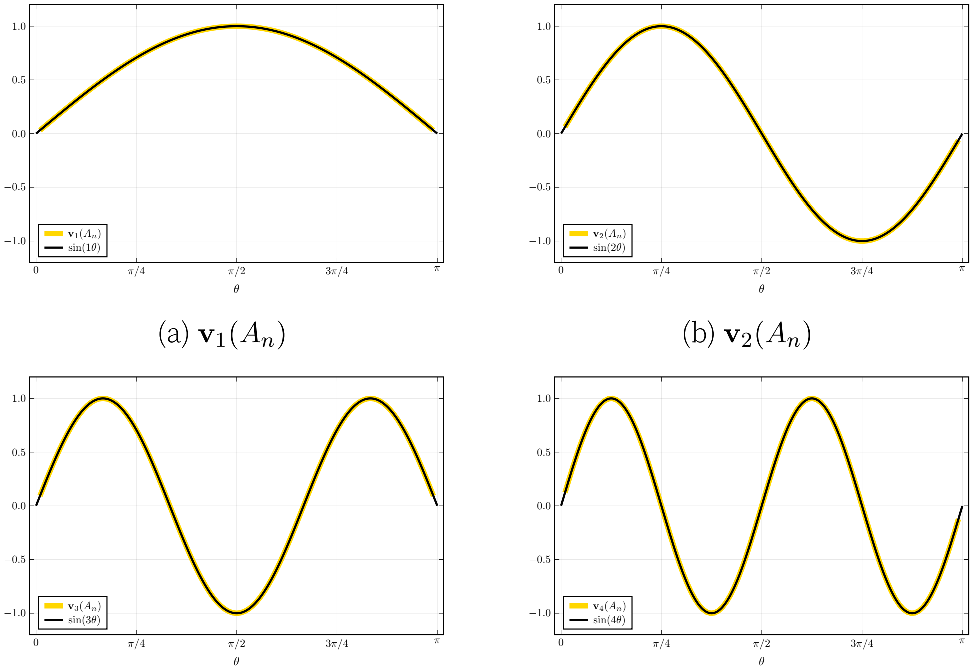
The algorithm up to this point is summarised in the following JULIA function:

```julia
function compute_c_vecs(n0 :: Integer, eigVecFunc, alpha :: Integer,
                        p :: Integer, name :: String, T :: DataType)
    j0 = 1:n0
    E  = zeros(T,alpha+1,n0)
    hs = zeros(T,alpha+1)
    maxIDX = 0
    for kk = 0:alpha
        nk        = (2^kk)*(n0+1)-1
        jk        = (2^kk)*j0
        hs[kk+1]  = convert(T,1)/(nk+1)
        # 1. Generate Eigenvectors
        EV = eigVecFunc(nk, p, T) #Returns the pth eigenvector of size nk
        # 2. Scale Eigenvectors
        if mod(p,2) == 1 #Odd index eigenvector
            EV = EV ./ EV[ceil(Int64, end/2)]
            EV = iseven((p - 1) / 2) ? EV : -1*EV #To get same form as DST
        elseif mod(p,2) == 0 && p <= n0 #Even index eigenvector
            i = ceil(Int64, nk / 2)
            EV = EV[i + 1] > 0 ? EV : -1*EV
            if kk == 0
                while(EV[i] < EV[i+1]) #Loop to find index of first local max
                    i += 1
                end
                maxIDX = i
            end
            EV = EV ./ EV[convert(Int64, 2^(kk)*maxIDX)]
            EV = iseven(p/2) ? EV : -1*EV #To get same form as DST
        end
        # 3. Generate E^j matrix
        eTnk       = EV
        E[kk+1,:] = eTnk[jk] #Constructing E matrix
    end
    # 4. Solve Linear System
    V = zeros(T,alpha+1,alpha+1)

    for ii = 1:alpha+1, jj = 1:alpha+1
        V[ii,jj] = hs[ii]^(jj-1)
    end

    return V\E
end
```

5. **Interpolation-Extrapolation**: Perform an interpolation-extrapolation procedure to compute the eigenvector of size n_f.

2.2 Laplacian Matrix

The first case we consider is for a matrix that we have already described in Example 1.2.1, where it was shown that MLM is able to approximate well the *eigenvalues* for these matrices. We investigate here how well we can approximate the *eigenvectors* which should be exactly given by DST, $\sin(j\theta_{i,n})$ where j is the eigenvector number and hence $c_k^{[j]} = 0$ for $k > 0$. This is because these trigonometric functions are eigenfunctions for the associated differential equation of the Laplacian matrix. In Figure 2.4 we plot the first four scaled eigenvectors of the Laplacian matrix of size $n = 101$.



(a) $\mathbf{v}_1(A_n)$

(b) $\mathbf{v}_2(A_n)$

(c) $\mathbf{v}_3(A_n)$

(d) $\mathbf{v}_4(A_n)$

Figure 2.4: First four scaled eigenvectors of the Laplacian matrix computed using (n_0, Data type) = (101, `Double64`) and compared with $\sin(j\theta)$.

The ordering of the eigenvectors is derived from the ordering of their associated eigenvalues (which are themselves ordered according to the associated symbol of the matrix, which is monotone, and thus in this case are ordered from smallest to largest), thus the first eigenvector is the one associated to the smallest eigenvalue.

Observation 2.2.1. As expected, we do see in Figure 2.4 that the eigenvectors are exactly given by DST. We further explore how different the other tested matrices are from $\sin(j\theta_{i,n})$ in their respective sections. It has been shown in [7] that for certain classes of Toeplitz matrices, specifically those with a symbol satisfying the ‘smooth simple-loop’ condition, the eigenvectors can be approximated by a specific DST with some corrective vectors towards the end points of the DST.

Using the Laplacian eigenvectors, we may now use the algorithm as described in Section 2.1 to compute the expansion functions for the different eigenvectors. In Figure 2.5, we show the computed expansion functions $c_k^{[j]},\ j = 1, \ldots, 4,\ k = 0, \ldots, 3$ for a number of different values of α.

We observe that, similarly to when MLM was used to approximate the eigenvalues of the Laplacian matrix in Example 1.2.1, all the expansion functions are 0 except $c_0^{[j]}$. We show in Figure 2.5 only the first four eigenvectors but indeed all the eigenvectors, up till $j = 101$ in this case, have the same behaviour where the only non-zero expansion function is that of $c_0^{[j]}$.

2.3 Bi-Laplacian Matrix

Consider the bi-Laplacian matrices described in Example 1.1.2 which are discretisation matrices using second order central finite difference approximations of size n. From Example 1.2.2 we observe that the eigenvalues of the bi-Laplacian matrix can be approximated by MLM and in that case it is not as trivial as with the Laplacian matrix, since we do not have an exact grid to sample the symbol.

Note. The square of the Laplacian matrix $L_n = T_n(2 - 2\cos(\theta))$, that is $(L_n)^2$, shares the same symbol as the bi-Laplacian matrix $B_n = T_n((2-2\cos(\theta))^2)$. However, $(L_n)^2 = B_n + R_n$ where R_n is a low-rank matrix with two non-zero elements, -1, in the top left and bottom right of the matrix. For the matrix $(L_n)^2$ we have the full eigendecomposition, where the eigenvalues are given by the sampling of the symbol $f(\theta) = (2-2\cos(\theta))^2$ with $\theta_{j,n} = \frac{j\pi}{n+1}$ and the eigenvector matrix is the DST [9].

In Figure 2.6, we show the first four eigenvectors of the bi-Laplacian as well as $\sin(j\theta)$ as was done for the Laplacian matrix. Whereas before, the jth eigenvector of the Laplacian matrix was an exact sampling of $\sin(j\theta)$, we see here there is some similarity but the curves no longer are exactly the same, and there is now quite a visible difference.

Observation 2.3.1. Note specifically how the final extrema (those extrema that are closest to the end-points) are always exceeding ± 1, with the employed scaling, but all other local extrema are ± 1 (as $\sin(j\theta)$ is). Also note the 'compression' of the vectors, which condense towards the centre. This is further highlighted in Figure 2.7 which shows the comparison for eigenvector $j = 11$.

In addition, we should note that the extrema in the first and second eigenvector will always be ± 1 using the scaling method we have described. If we follow though the same pattern as observed for the other eigenvectors, then the extrema would not 'truely' be ± 1 but exceed this, thus these eigenvectors are not perfectly scaled to some theoretical 'eigenvector symbol'.

In Figure 2.8 we see the computed $c_k^{[j]}$ for the first four eigenvectors, with a matrix size of $n_0 = 201$ and for a number of different values of α, which is explained in Observation 2.3.2. We see that the expansion functions are able to be computed

Figure 2.5: Computed expansions for the first four eigenvectors of the Laplacian matrix for (n_0, α, Data type) = (101, 3, `Double64`).

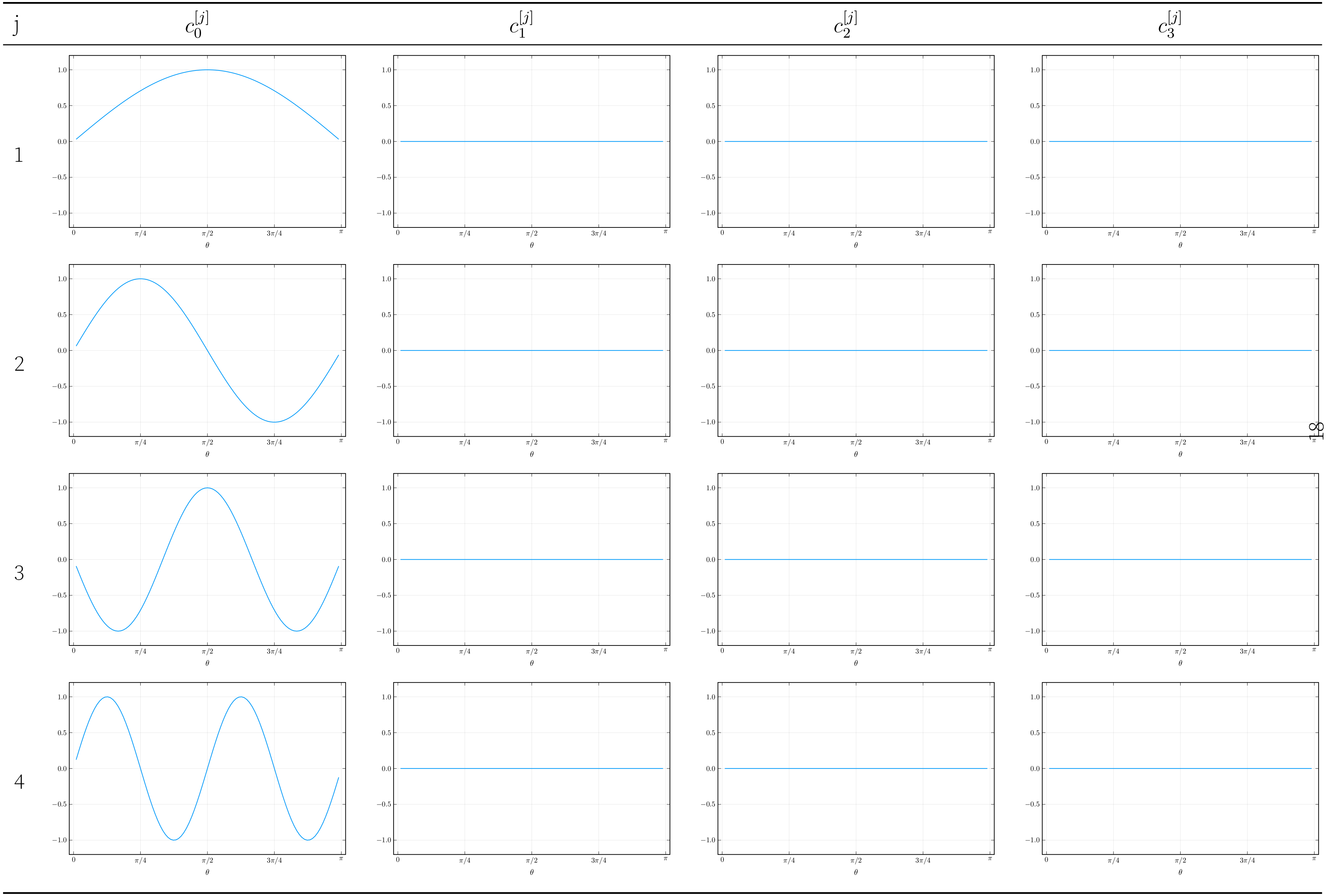
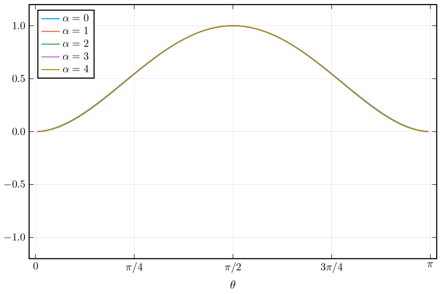
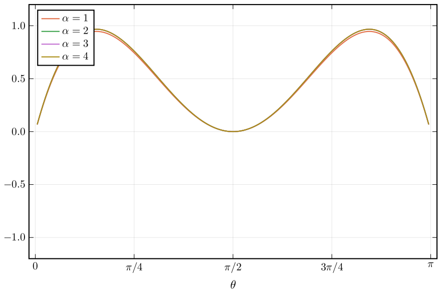
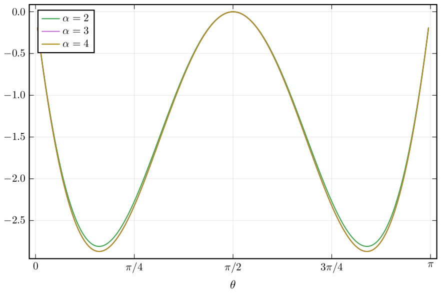
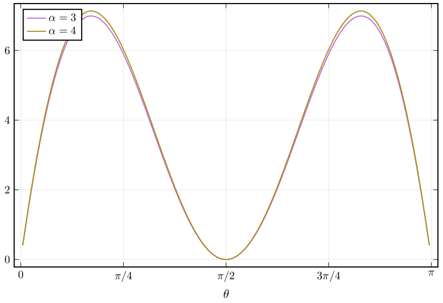
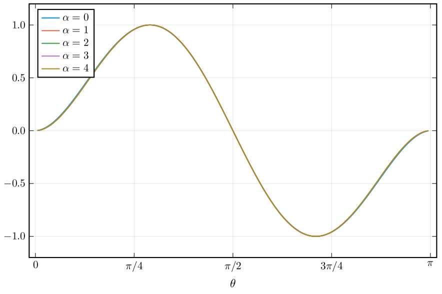
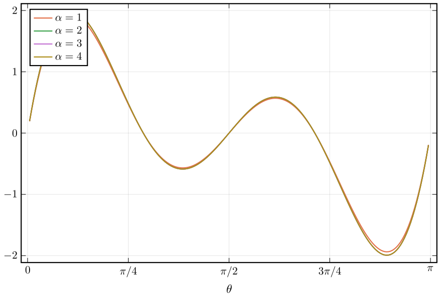
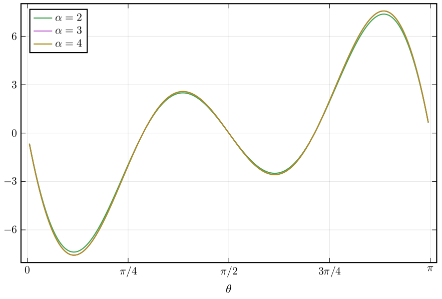
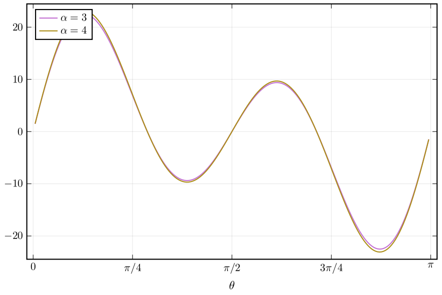
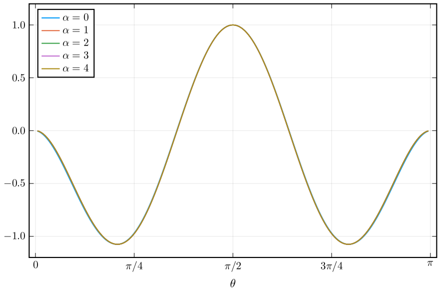
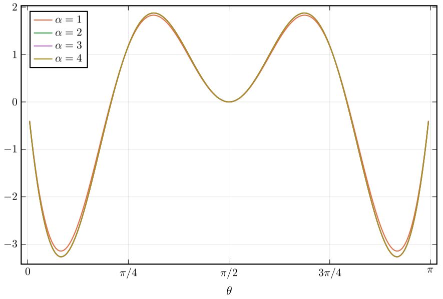
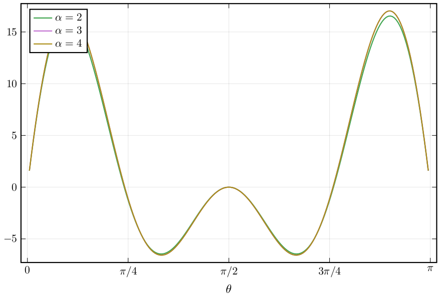
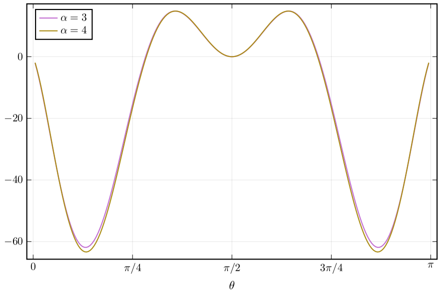
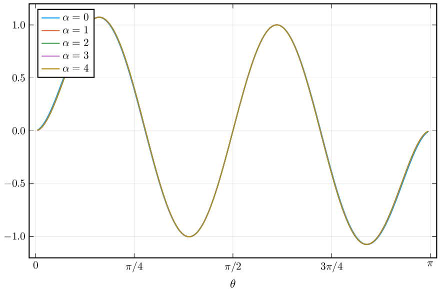
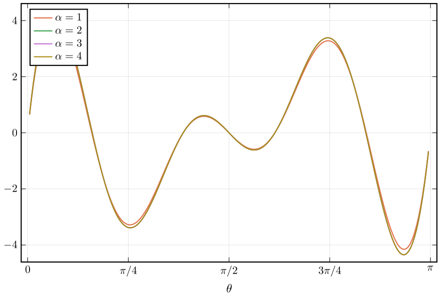
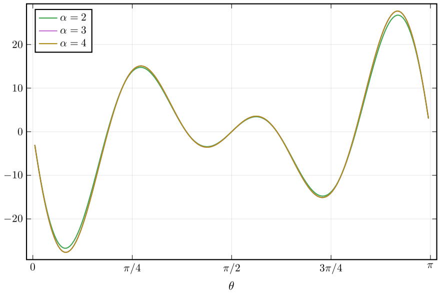
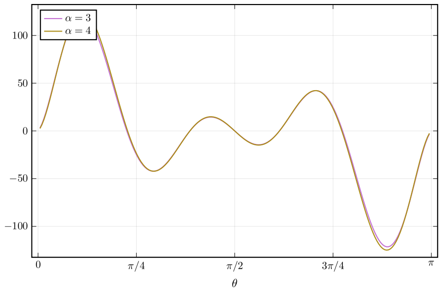
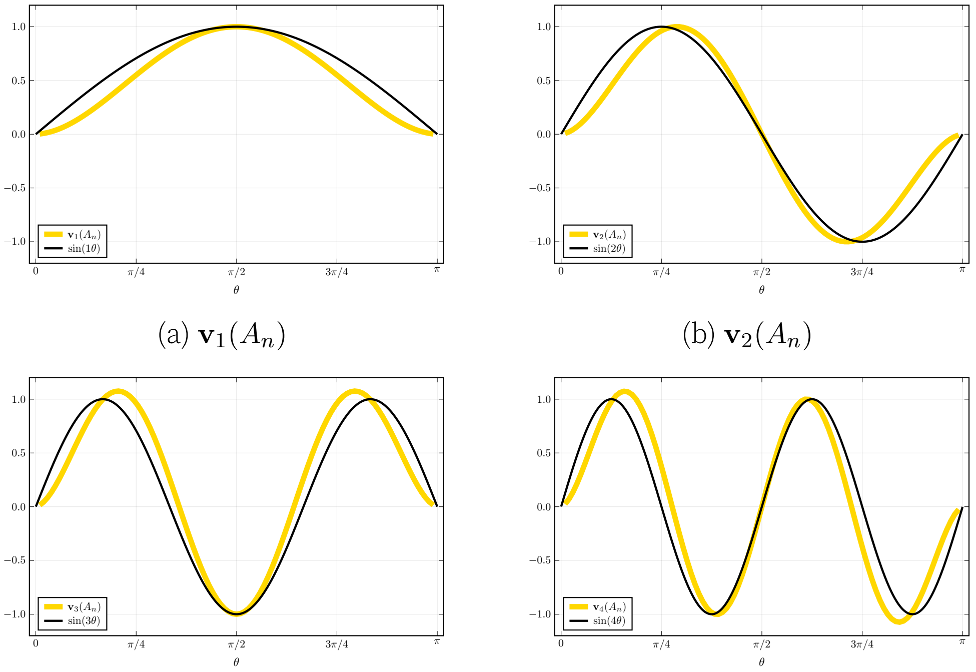


Figure 2.8: Computed expansions for the first 4 eigenvectors of the bi-Laplacian matrix for (n_0, Data type) = (201, `Double64`) with different α.

j	$c_0^{[j]}$	$c_1^{[j]}$	$c_2^{[j]}$	$c_3^{[j]}$
1				
2				
3				
4				



(a) $\mathbf{v}_1(A_n)$

(b) $\mathbf{v}_2(A_n)$

(c) $\mathbf{v}_3(A_n)$

(d) $\mathbf{v}_4(A_n)$

Figure 2.6: First four scaled eigenvectors of the bi-Laplacian matrix computed using (n_0, Data type) = (101, `Double64`) and compared with $\sin(j\theta)$.

by the method, and for the bi-Laplacian matrix we note that none of the expansion functions are zero as we observed for the special case with the Laplacian matrix.

There are two important aspects to note about the above result:

- The computation of the $c_k^{[j]}$'s are numerically sensitive. It was observed that the standard `Float64` data type (Double precision with machine epsilon, $\epsilon = \mathcal{O}(10^{-16})$), as used by default in many programming languages such as MATLAB, did not yield the same curves for different alpha with $c_1^{[j]}$ and $c_2^{[j]}$ for the tested eigenvectors. Using a high precision data type (`Double64` with machine epsilon, $\epsilon = \mathcal{O}(10^{-32})$ from Julia [26] was used in this case) ensured the expansion functions were then all correctly approximated. This sensitivity is highlighted in Figure 2.9.

- The expansion functions, for $k > 0$, increase in magnitude with the eigenvalue number. That is, consider $c_3^{[1]}$ and $c_3^{[4]}$ and note how much larger in magnitude $c_3^{[4]}$ is over $c_3^{[1]}$. This indicates that there will be issues when using the computed expansion functions to approximate more than just the first few eigenvectors. This point will be readdressed in Section 3.1.

- Consider Figure 2.8, and note that for each figure the lowest α value has a slightly different shape to the larger alpha values, that is, that the $c_k^{[j]}$ for which $\alpha > k$ appear to perform better. So if we would like to calculate up to a specific k_{max}, k_{max} is usually taken to be 3 in this paper, then we should use an α which is larger than k_{max}, we here will typically use $\alpha = 4$.

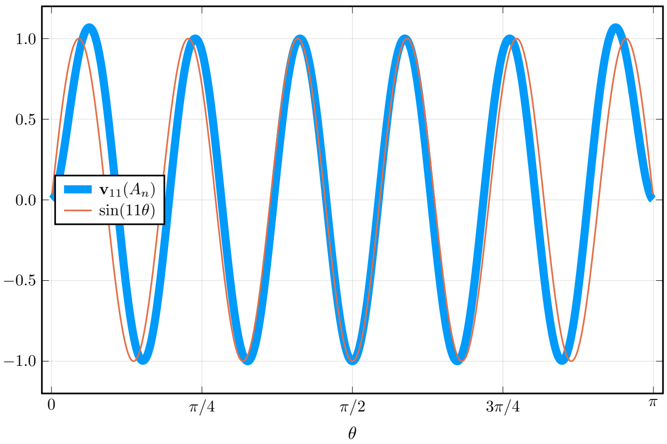
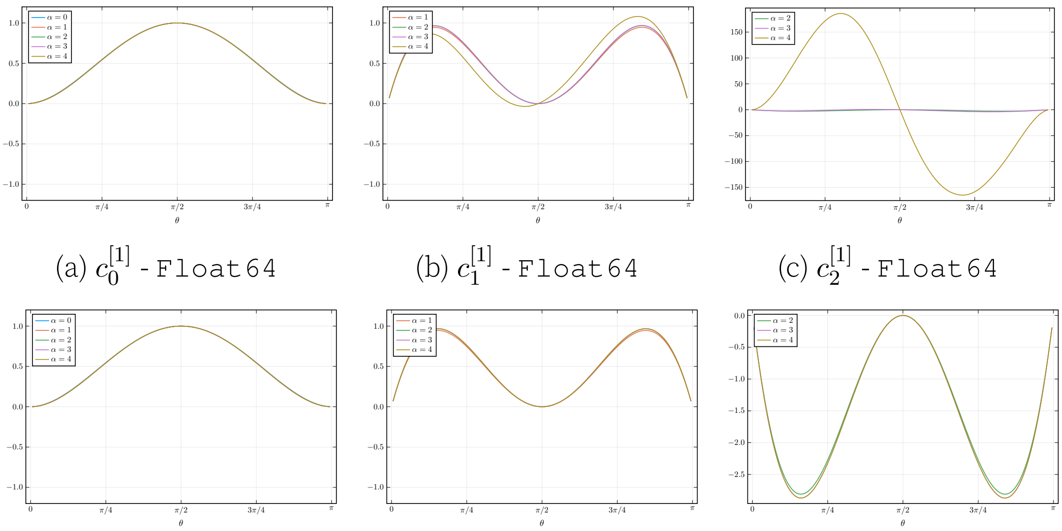


Figure 2.7: Comparison of bi-Laplacian eigenvector $j = 11$ with sine highlighting the differences (n_0, Data type) = (501, `Double64`).



(a) $c_0^{[1]}$ - `Float64`

(b) $c_1^{[1]}$ - `Float64`

(c) $c_2^{[1]}$ - `Float64`

(d) $c_0^{[1]}$ - `Double64`

(e) $c_1^{[1]}$ - `Double64`

(f) $c_2^{[1]}$ - `Double64`

Figure 2.9: Comparison of approximated expansion functions with different data types for the first eigenvector of the bi-Laplacian matrix (n_0) = (201).

Observation 2.3.2. The reason that Figure 2.8 has curves for different values of α is that if the method is successfully working, then we would observe approximately the same curves for different values of α and for different starting sizes n_0. This gives us a condition to check whether the method is able to correctly compute the expansion functions.

We here also look at using the full method to extrapolate the computed expansion functions to approximate an eigenvector of larger size. Figure 2.10 shows the $\log_{10}$ of the error when extrapolating from a size of $n_0 = 201$ to a size $n_f = 10000$ for $\alpha = 4$.

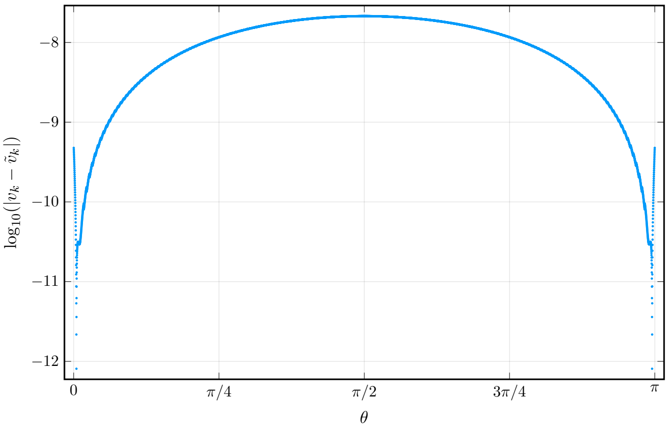


Figure 2.10: $\log_{10}$ error $\log_{10}\left|\mathbf{v}_j(A_{n_f}) - \tilde{\mathbf{v}}_j(A_{n_f})\right|$ of the interpolated-extrapolated first eigenvector for $n_f = 10000$ and (n_0, α, Data type) = (201, 4, `Double64`).

2.4 Isogeometric Analysis (IGA) Matrix

In this section we investigate whether we can apply MLM to the IGA matrices, specifically the $\mathbf{L_n}$ matrix defined in [15, p. 1652]. This matrix is not purely Toeplitz as we have so far dealt with, but is Toeplitz-like as the enforced boundary conditions in the differential equation cause the 'corners' of the matrix to be perturbed. The IGA $\mathbf{L_n}$ matrix has symbol, describing the eigenvalues,

$$\begin{aligned} \left\{\frac{1}{n^2}\mathbf{L}_n\right\} &\sim_{\text{GLT}} f \\ f &= \frac{42(40 - 15\cos(\theta) - 24\cos(2\theta) - \cos(3\theta))}{1208 + 1191\cos(\theta) + 120\cos(2\theta) + \cos(3\theta)} \end{aligned}$$

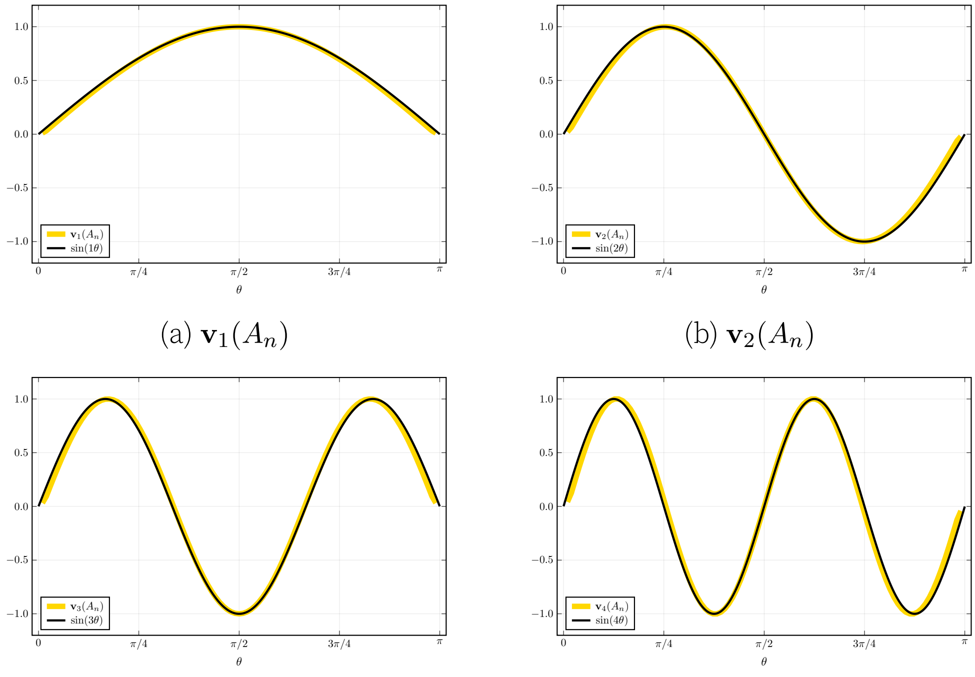
where f is a monotonically increasing function. $\mathbf{L_n}$ is here created by using the mass $\mathbf{M_n}$ and stiffness $\mathbf{K_n}$ matrices, such that $\mathbf{L_n} = (\mathbf{M_n})^{-1}\mathbf{K_n}$, where:

$$n\mathbf{M_n} = \frac{1}{10080}\begin{bmatrix} 2232 & 1575 & 348 & 3 & & & & & & & \\ 1575 & 3294 & 2264 & 239 & 2 & & & & & & \\ 348 & 2264 & 4832 & 2382 & 240 & 2 & & & & & \\ 3 & 239 & 2382 & 4832 & 2382 & 240 & 2 & & & & \\ & 2 & 240 & 2382 & 4832 & 2382 & 240 & 2 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & 2 & 240 & 2382 & 4832 & 2382 & 240 & 2 & \\ & & & & 2 & 240 & 2382 & 4832 & 2382 & 239 & 3 \\ & & & & & 2 & 240 & 2382 & 4832 & 2264 & 348 \\ & & & & & & 2 & 239 & 2264 & 3294 & 1575 \\ & & & & & & & 3 & 348 & 1575 & 2232 \end{bmatrix}$$

and

$$\frac{1}{n}\mathbf{K_n} = \frac{1}{240}\begin{bmatrix} 360 & 9 & -60 & -3 & & & & & & & \\ 9 & 162 & -8 & -47 & -2 & & & & & & \\ -60 & -8 & 160 & -30 & -48 & -2 & & & & & \\ -3 & -47 & -30 & 160 & -30 & -48 & -2 & & & & \\ & -2 & -48 & -30 & 160 & -30 & -48 & -2 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & -2 & -48 & -30 & 160 & -30 & -48 & -2 & 0 \\ & & & & -2 & -48 & -30 & 160 & -30 & -47 & -3 \\ & & & & & -2 & -48 & -30 & 160 & -8 & -60 \\ & & & & & & -2 & -47 & -8 & 162 & 9 \\ & & & & & & & -3 & -60 & 9 & 360 \end{bmatrix}$$

Figure 2.11 shows the first four scaled *eigenvectors* of the IGA $\mathbf{L_n}$ matrix.



(a) $\mathbf{v}_1(A_n)$

(b) $\mathbf{v}_2(A_n)$

(c) $\mathbf{v}_3(A_n)$

(d) $\mathbf{v}_4(A_n)$

Figure 2.11: First four eigenvectors of IGA $\mathbf{L_n}$ matrix computed using (n_0, Data type) = (101, `Double64`) and compared with $\sin(j\theta)$.

We observe that the eigenvectors are extremely close to the DST with a very small amount of 'drift' outwards. We do not observe the behaviour of the final extrema as we did with the bi-Laplacian in Figure 2.6, with all the extrema in this case being the same magnitude. However, we do note that there is a small 'cusp' near both ends for all eigenvectors, see Figure 2.12, which we anticipate may pose a problem in the approximated expansion functions.

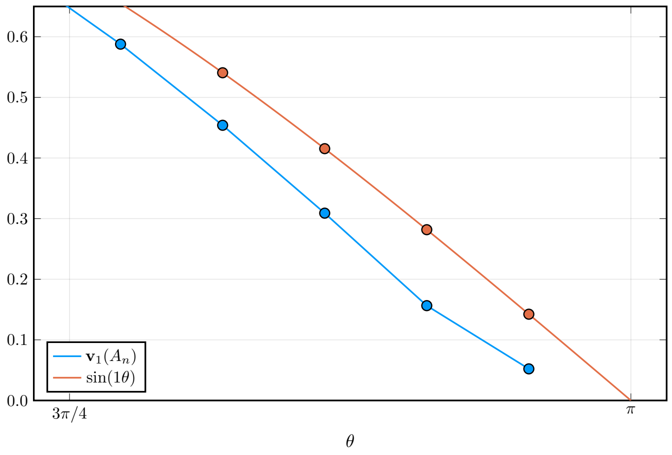


Figure 2.12: Zoomed plot of $\mathbf{v}_1(A_n)$ using (n_0, Data type) = (21, `Double64`) and compared with $\sin(\theta)$.

We observe from Figure 2.13 that we do get consistent $c_k^{[j]}$ in the interior of the eigenvectors (the same curves for different α) however there are large asymptotes near the end points for $c_k^{[j]},\ k > 0$. This behaviour is most likely due to the presence of 'cusps' near the ends of the eigenvectors as previously described, and shown in Figure 2.12. This problem towards the end points is consistent with the finding in [24] where it was found that using MLM for the eigenvalues also had difficulty near the end points.

2.5 Fractional Derivative Related Matrix

For the final test case, we consider a 'full' symmetric Toeplitz matrix, that is, a symmetric Toeplitz matrix with only non-zero coefficients. This means that for any finite n the full set of Fourier coefficients is not known. To that end, consider a first order discretisation of the fractional derivative [12]. Specifically for the 1.5 derivative, the symbol of the matrix will then be:

$$f(\theta) = e^{-\mathrm{i}\theta}\left(1 - e^{\mathrm{i}\theta}\right)^{1.5}.$$

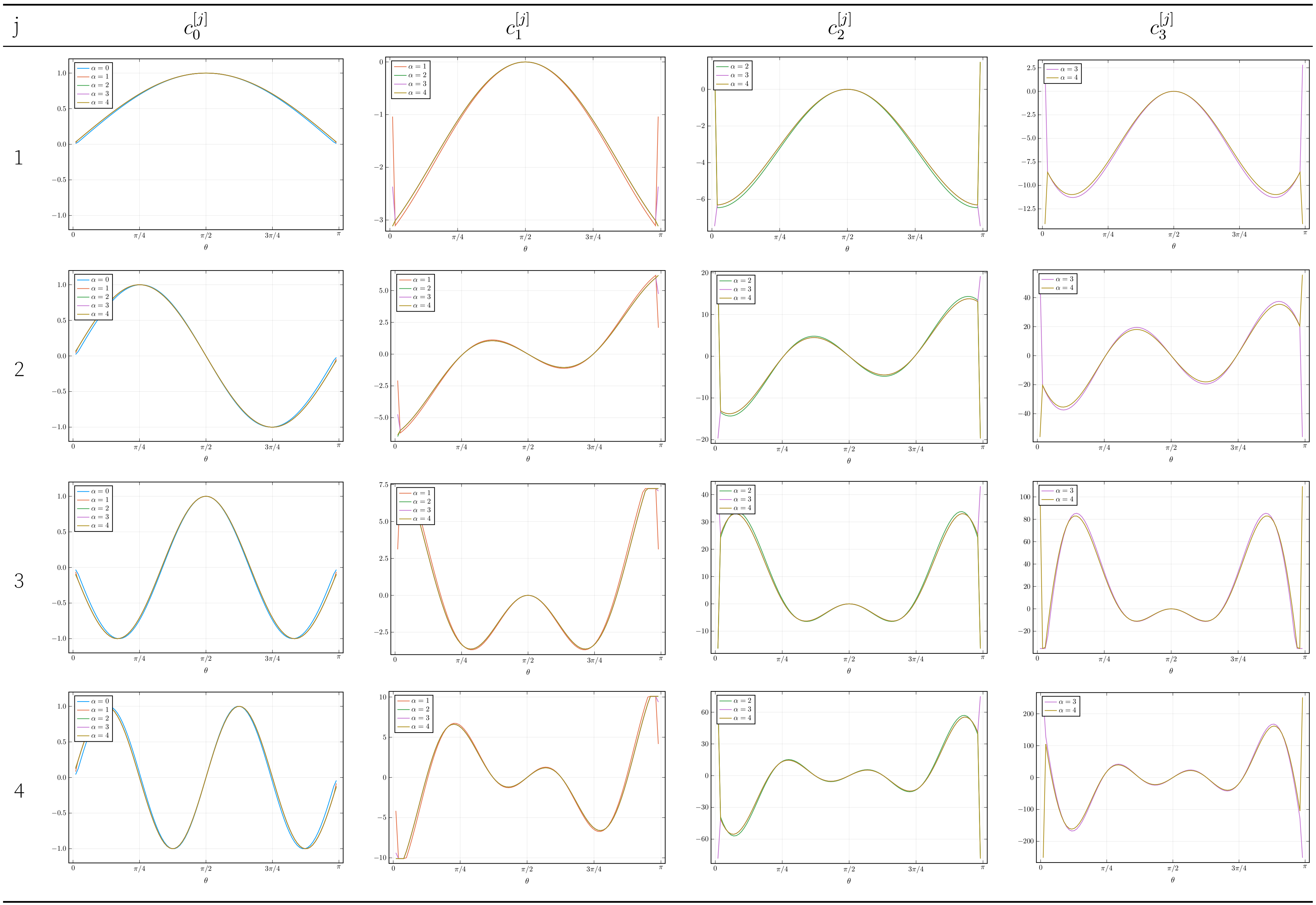
Using the Grünwald formula [29], we obtain, e.g.,

$$T_5(f) = \begin{bmatrix} \frac{1}{2} & -1 & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{2} & -1 & 0 & 0 \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{2} & -1 & 0 \\ \frac{5}{128} & \frac{1}{16} & \frac{1}{8} & \frac{1}{2} & -1 \\ \frac{7}{256} & \frac{5}{128} & \frac{1}{16} & \frac{1}{8} & \frac{1}{2} \end{bmatrix}, \qquad T_6(f) = \begin{bmatrix} \frac{1}{2} & -1 & 0 & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{2} & -1 & 0 & 0 & 0 \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{2} & -1 & 0 & 0 \\ \frac{5}{128} & \frac{1}{16} & \frac{1}{8} & \frac{1}{2} & -1 & 0 \\ \frac{7}{256} & \frac{5}{128} & \frac{1}{16} & \frac{1}{8} & \frac{1}{2} & -1 \\ \frac{21}{1024} & \frac{7}{256} & \frac{5}{128} & \frac{1}{16} & \frac{1}{8} & \frac{1}{2} \end{bmatrix}.$$

As can be seen above, the corresponding matrix sequences $T_n(f)$ are not symmetric (Hermitian) which we require in this report, thus here we consider the symmetric part of these matrices, defined as:

$$A_n = \frac{T_n(f) + T_n(f)^{\mathbf{T}}}{2}.$$

Figure 2.13: Computed expansions for the first four eigenvectors of the IGA $\mathbf{L_n}$ matrix for $(n_0$, Data type$)$ = (101, `Double64`) with different α.

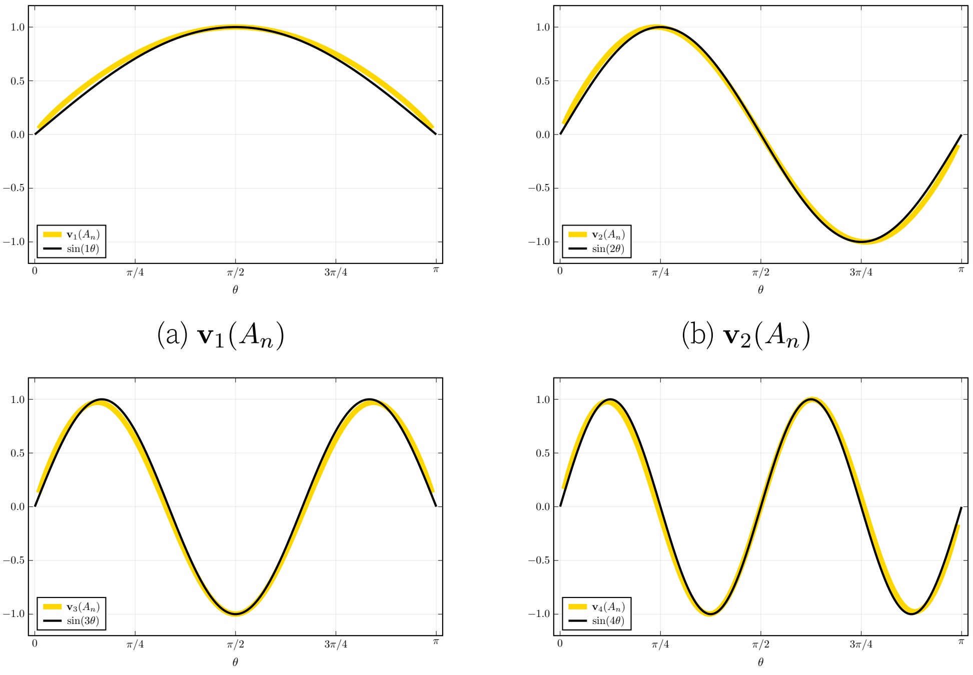


25

Then we obtain:

$$A_5 = \begin{bmatrix} \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} & \frac{5}{256} & \frac{7}{512} \\ \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} & \frac{5}{256} \\ \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} \\ \frac{5}{256} & \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} \\ \frac{7}{512} & \frac{5}{256} & \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} \end{bmatrix}, \qquad A_6 = \begin{bmatrix} \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} & \frac{5}{256} & \frac{7}{512} & \frac{21}{2048} \\ \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} & \frac{5}{256} & \frac{7}{512} \\ \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} & \frac{5}{256} \\ \frac{5}{256} & \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} & \frac{1}{32} \\ \frac{7}{512} & \frac{5}{256} & \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} & \frac{-7}{16} \\ \frac{21}{2048} & \frac{7}{512} & \frac{5}{256} & \frac{1}{32} & \frac{-7}{16} & \frac{1}{2} \end{bmatrix}.$$

Figure 2.14 shows the first four scaled eigenvectors of the matrix A_n as described above.



(a) $\mathbf{v}_1(A_n)$

(b) $\mathbf{v}_2(A_n)$

(c) $\mathbf{v}_3(A_n)$

(d) $\mathbf{v}_4(A_n)$

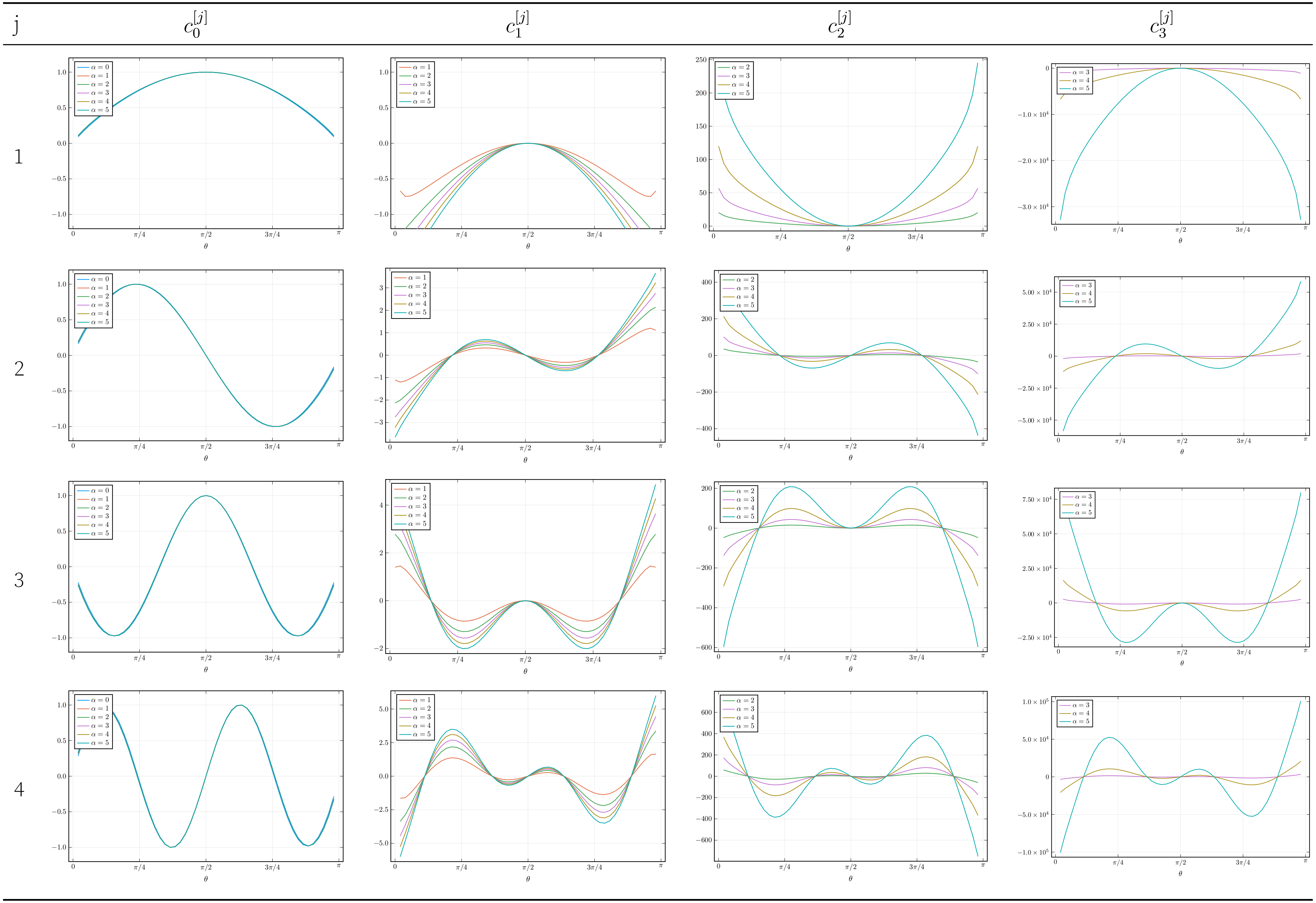
Figure 2.14: First four scaled eigenvectors of the symmetrised Fractional Derivative matrix computed using (n_0, Data type) = (101, `Double64`) and compared with $\sin(j\theta)$.

Observation 2.5.1. We can observe that the behaviour of the eigenvectors is almost the opposite of what we observe for the bi-Laplacian. Here we have that the final extrema are less that ± 1 (rather than larger) and there is a general outward expansion (rather than inward compression).

In Figure 2.15, we plot the computed expansion functions for the first four eigenvectors.

Observation 2.5.2. We observe from Figure 2.15, that in this case we do not obtain consistent curves for different α, which means we cannot in this case use our derived MLM to approximate the eigenvectors. We can note however that for the different α curves, they do have the same shape and rather there appears to be a scaling issue which indicates that it might be possible to create another MLM which allows for different scaling of the expansion functions depending on α, or a possible modification of the working hypothesis may be required.

Figure 2.15: Computed expansions for the first 4 eigenvectors of the symmetrised Fractional Derivative matrix for (n_0, Data type) = (51, `Double64`) with different α.



We have now tested our MLM for eigenvectors on four different test cases. In the next section we look into some considerations one must make when using this method.

III

Considerations and Extensions

We observed in the results that the method as described in Section 1.3 does work for a number of different matrices, however also observed an example where it was unable to accurately approximate the expansion functions, however it may be possible that the expansion could be altered to include scaling terms which would resolve the issue, or modify the working hypothesis.

In this section we detail some considerations one must take into account when using this eigenvector MLM method as well as some possible extensions of the method.

3.1 Number of Eigenvectors to be Approximated

In all the results in the previous section, we have only considered the first few eigenvectors, but generally how does the method perform beyond that? Firstly, there is the practical limitation that we can only perform the expansion on the first n_0 eigenvectors, as we will not be able to generate expansion functions for the remaining eigenvectors eigenvectors. Secondly, we already made the note in the bi-Laplacian results that the expansion functions increase in magnitude with eigenvector number which already indicates that there could be a degredation in accuracy as the eigenvector number increases. Indeed, consider Figure 3.1 which shows the maximum-norm of the error of the approximated eigenvectors and we observe that the error does generally grow with the eigenvector number. The chosen error estimate used is $\left\|A_{n_f}\tilde{\mathbf{v}}_j(A_{n_f}) - \lambda_j\tilde{\mathbf{v}}_j(A_{n_f})\right\|_\infty$ where A_{n_f} is the bi-Laplacian matrix of size n_f, λ_j is the jth eigenvalue and $\tilde{\mathbf{v}}_j(A_{n_f})$ is the approximation of the jth eigenvector.

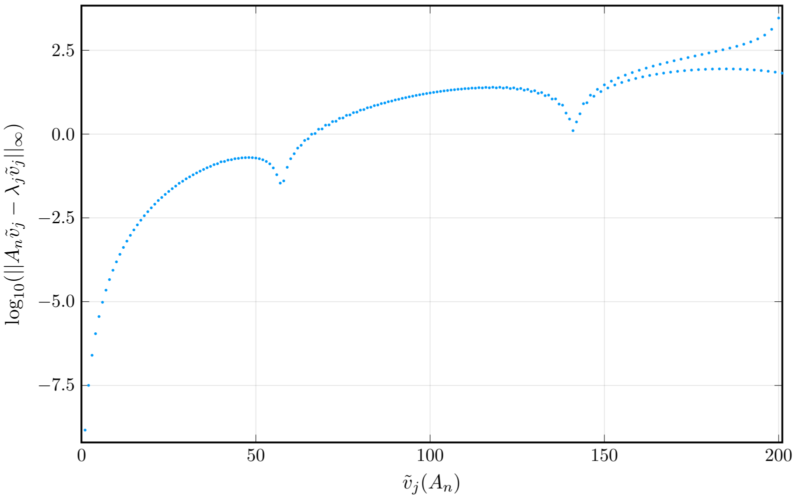


Figure 3.1: Errors $\log_{10}\left\|A_{n_f}\tilde{\mathbf{v}}_j(A_{n_f}) - \lambda_{j,n_f}\tilde{\mathbf{v}}_j(A_{n_f})\right\|_\infty$ of the bi-Laplacian matrix for $n_f = 10000$ and (n_0, α, Data type) = (201, 4, `Double64`).

Choosing a larger n_0 will of course decrease the error as can be seen in Figure 3.2 where we show the error in the approximation for a number of different starting n_0.

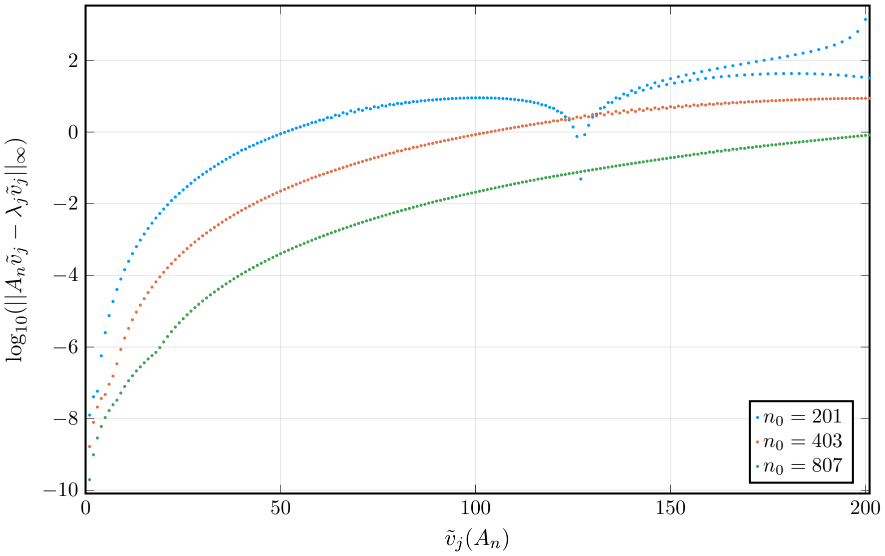


Figure 3.2: Errors $\log_{10}\left\|A_{n_f}\tilde{\mathbf{v}}_j(A_{n_f}) - \lambda_{j,n_f}\tilde{\mathbf{v}}_j(A_{n_f})\right\|_\infty$ of the bi-Laplacian matrix for $n_f = 10000$, $n_0 = \{201, 403, 807\}$ and (α, Data type) = (3, `Double64`).

3.2 Non-monotone Symbols

So far, in this report, we have considered symmetric Toeplitz matrices with monotone symbols, and thus we may ask ourselves how the method performs generally on symmetric Toeplitz matrices with non-monotone symbols. In this case, consider the following symbol which is related to the bi-Laplacian:

$$f(\theta) = \gamma e^{-2\mathbf{i}\theta} - 4e^{-\mathbf{i}\theta} + 6 - 4e^{\mathbf{i}\theta} + \gamma e^{2\mathbf{i}\theta}.$$

This symbol is monotone for $\gamma \in [-1, 1]$ and non-monotone outside this interval. A value of $\gamma = 1$ would exactly give us the symbol of the bi-Laplacian matrices. The standard functions which numerically generate the eigenvalues (and vectors) will sort them into non-decreasing order of the eigenvalues, meaning that for a matrix corresponding to a non-monotone symbol, the eigenvector corresponding to the first eigenvalue in the symbol will not be the first one in the matrix of eigenvectors.

Consider the case for $\gamma = 2$, where Figure 3.3 plots the symbol as well as the sorted eigenvalues which are computed by a standard eigenvalue solver. We observe that below the green line, the symbol is non-monotone, and that the eigenvalues do not follow the same shape as the symbol in this part since they are sorted.

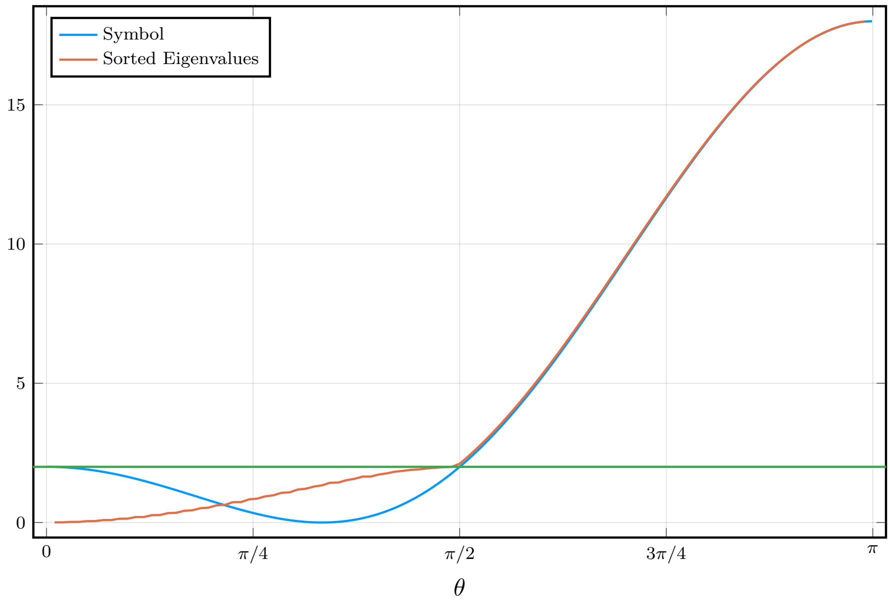
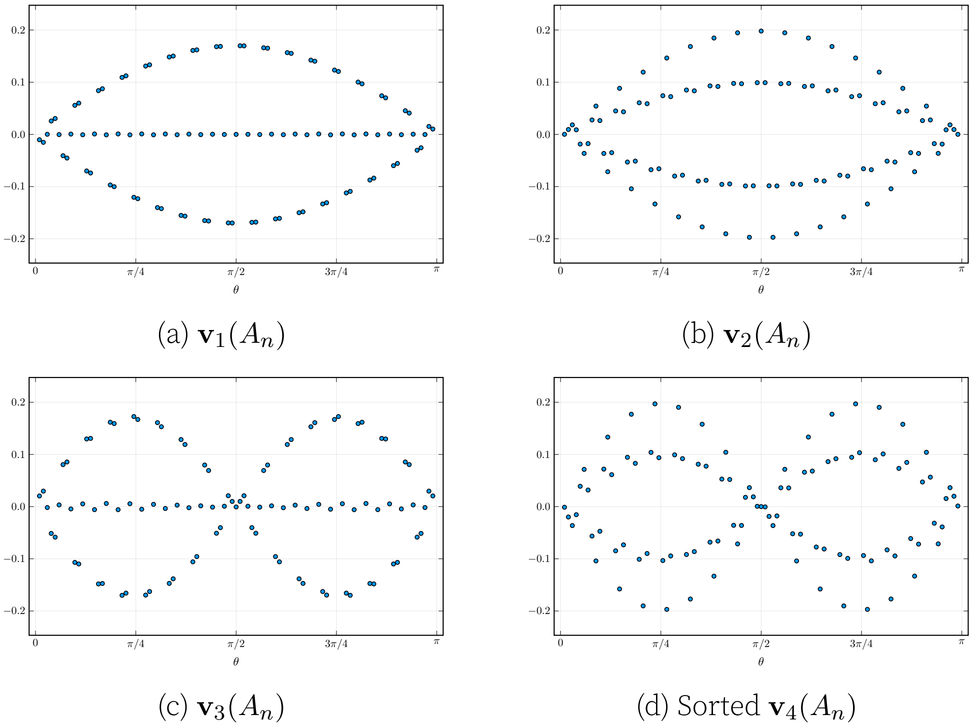


Figure 3.3: Symbol and sorted eigenvalues (n_0, Data type) = (101, `Double64`).

Thus, we would then expect that the first column of the matrix containing the eigenvectors returned by standard solvers, may not be the true 'first' eigenvector according to the symbol. In Figure 3.4 we have plotted the first four eigenvectors which are computed by a standard solver (`eigvecs` in Julia).



(a) $\mathbf{v}_1(A_n)$

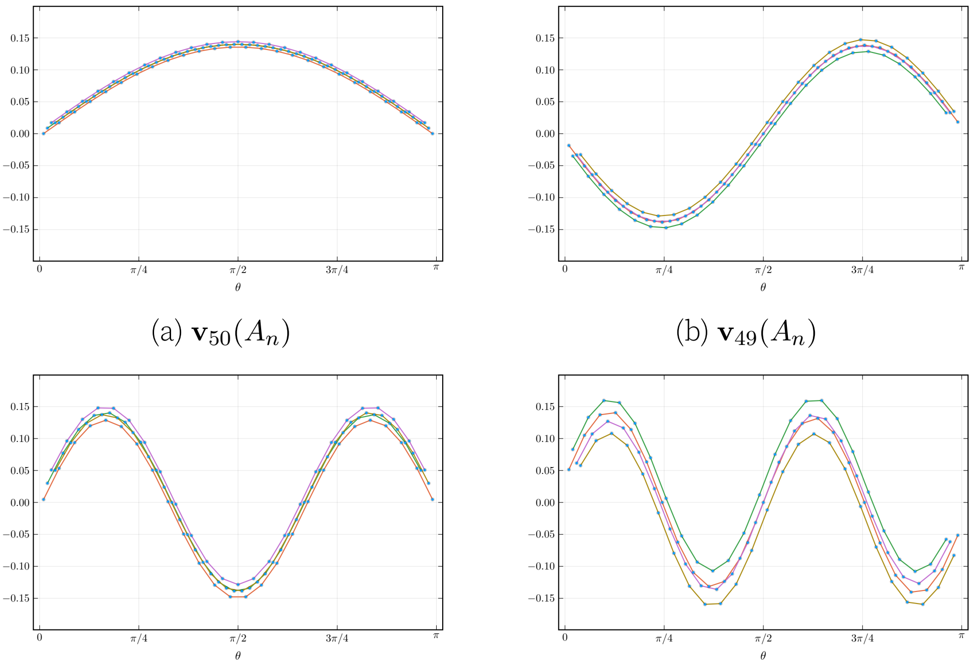
(b) $\mathbf{v}_2(A_n)$

(c) $\mathbf{v}_3(A_n)$

(d) Sorted $\mathbf{v}_4(A_n)$

Figure 3.4: First four eigenvectors of the non-monotone bi-Laplacian matrix as computed by a standard solver (n_0, Data type) = (101, `Double64`).

Indeed we observe in Figure 3.4 that the obtained eigenvectors from the standard solver are not of the form we would generally expect, and have so far seen in the report. By applying a 'reordering' of the eigenvectors, where we now sort them according to the number of times they change sign, we then obtain Figure 3.5, that shows the first four resorted eigenvectors which correspond to eigenvectors 50, 49, 48 and 47 respectively.



(a) $\mathbf{v}_{50}(A_n)$

(b) $\mathbf{v}_{49}(A_n)$

(c) $\mathbf{v}_{48}(A_n)$

(d) $\mathbf{v}_{47}(A_n)$

Figure 3.5: First four eigenvectors of the non-monotone bi-Laplacian matrix after reordering (n_0, Data type) = (101, `Double64`).

The eigenvectors in Figure 3.5 are now more of the shape we would usually expect, being similar to DST [9]. If the points were to be plotted using a single line, we would observe an oscillating line, but using points we can observe that there appears to be three or four (or more) distinct curves in all the eigenvectors (which have been plotted).

Thus, we have not only the issue of the ordering of the eigenvectors when dealing with matrices resulting from non-monotone symbols, we also appear to have interlaced samplings which means that the current eigenvector MLM we have proposed in Section 1.3 would be unable to approximate the expansion functions.

IV

Conclusions

The goal of this report is to investigate whether it is possible to use the matrix-less method defined in Section 1.3 to approximate the eigenvectors of symmetric Toeplitz and Toeplitz-like matrices with monotone symbols. It is specifically highlighted that care must be taken to ensure that the eigenvectors are correctly scaled so that the method can use the structure that is maintained in the eigenvectors when the matrix changes size.

The first case that is treated in Section 2.2 is that of the Laplacian matrix where we have the exact eigendecomposition, and this is used to check that MLM behaves as expected. We have in this case that the eigenvectors of the Laplacian matrix are given exactly by the DST of the same size, that is that the j-th eigenvector is an equispaced sampling of $\sin(j\theta)$ where $\theta_{j,n}$ is an equispaced grid. Applying MLM to the Laplacian matrix, it was found that the expansion functions $\tilde{c}_{k;j},\ k > 0$ are zero, similar to the behaviour observed when MLM is used on the eigenvalues.

The second case in Section 2.3 involved investigating the performance on the bi-Laplacian matrix, which is related to the Laplacian matrix since the square of the Laplacian is the bi-Laplacian with a small rank perturbation. The eigenvectors in this case were not given exactly by the DST but were still extremely similar in shape, with the differences being a contraction of the points towards the centre and that the final extrema were larger than ± 1.

The expansion functions for the eigenvalues of the bi-Laplacian were not trivial as they were for the Laplacian matrix, and this same trend continued for the expansion functions of the eigenvectors. Consistent curves were observed in this case, meaning that approximately the same expansion function was computed for different values of α, indicating that the method does work in this case. It was noted that the expansion functions are numerically sensitive and a high accuracy data-type needs to be used in order to correctly approximate the expansion functions.

The third case in Section 2.4 uses a Toeplitz-like matrix rather than a purely Toeplitz matrix as in the previous two cases. The matrix in question is derived using IGA and it was observed that the eigenvectors in this case were again very similar to the DST, even more than the bi-Laplacian with only slight differences. Notably though, there was a small cusp near both ends of the eigenvectors, and it was this cusp which caused some small issues when computing the expansion functions. The expansion functions are found to have consistent curves for different alpha with the only issue of an asymptote in $c_k^{[j]},\ k > 0$ at the point of the cusp. This indicates that it may be necessary to only use the parts of the higher expansion functions which are unaffected by the asymptotes, and then only use a lower order approximation at the cusps.

The final case in Section 2.5 deals with a matrix related to the fractional derivative, in that it is the symmetric part of a discretisation of the fractional derivative and it is a 'full' symmetric Toeplitz matrix. The eigenvectors in this case are again similar to the DST but have the opposite behaviour to the bi-Laplacian where here there is an expansion away from the centre and the final extrema are smaller than ± 1.

The expansion functions in this case did not get consistent curves for different α meaning that the method was not able to correctly approximate the expansion functions. The shape of the expansion functions was consistent though, however there appeared to be a different scaling between them. A modification to the MLM to allow for a scaling parameter that depends on α may resolve the issue observed here, or the working hypothesis may need to be modified.

Some general considerations with the method were then discussed, where it was highlighted that the error in the method increases with the eigenvector number. The issues arising when using symmetric Toeplitz matrices with non-monotone symbols was also briefly discussed whether the main issues were that the eigenvectors were no longer correctly ordered according to the symbol, which is the same issue when performing MLM for the eigenvalues of these matrices, and also that the eigenvectors appeared to have three different interlaced samplings.

For future research, the following items are suggested:

1. Can the method be applied as is to deflation methods to approximate the necessary eigenvectors?

2. Can a method be created where the eigenvalues and eigenvectors are simultaneously approximated and improved, using for example [13] which is implemented in `IterativeRefinement.jl` [32], where the eigenvectors which are not in the first n_0, are approximated using the DST?

3. Can the method be modified to allow a scaling parameter that depends on α and scales the expansion functions as needs for certain matrices?

4. How can the eigenvectors for symmetric Toeplitz matrices with non-monotone symbols be correctly approximated? Can the different interlaced sampling be individually expanded or do they have some dependence on each other?

5. Since there is also an issue when approximating the eigenvalues of Toeplitz and Toeplitz-like matrices with non-monotone symbols due to the incorrect ordering, can the eigenvectors be used to the assign the correct ordering to the eigenvalues and then can MLM be used for non-monotone symbols?

6. In this thesis we have created expansions for the columns of the matrix of eigenvectors for a given matrix sequence. Would it however be possible to create expansions for the rows as an alternative or complement?

V

Acknowledgements

Firstly, I would like to give my heartfelt gratitude to my thesis supervisor Sven-Erik Ekström for not only his support and guidance during the thesis but also for introducing me to this very interesting topic. He really allowed me to explore the topic and expand in many different directions. He was always available to answer questions, no matter how big or small.

Secondly, I would like to give thanks to my thesis reviewer Maya Neytcheva for taking the time to share her vast knowledge on the topic but also her great comments and feedback.

Finally, I would also like to thank my family and friends for their love and encouragement during the whole process. Without their support I would not be where I am today and they really have made this possible.

References

[1] F. Ahmad, E. S. Al-Aidarous, D. A. Alrehaili, S.-E. Ekström, I. Furci, and S. Serra-Capizzano, "Are the eigenvalues of preconditioned banded symmetric Toeplitz matrices known in almost closed form?" *Numerical Algorithms*, vol. 78, no. 3, pp. 867–893, Aug. 2017. DOI: 10.1007/s11075-017-0404-z.

[2] G. Barbarino, "A systematic approach to reduced GLT," *BIT Numerical Mathematics*, Sep. 14, 2021, ISSN: 1572-9125. DOI: 10.1007/s10543-021-00896-7.

[3] G. Barbarino, M. Claesson, S.-E. Ekström, C. Garoni, and D. Meadon, "Matrix-less eigensolver for large structured matrices," Department of Information Technology, Uppsala University, 2021-005, Aug. 2021.

[4] M. Barrera, A. Böttcher, S. M. Grudsky, and E. A. Maximenko, "Eigenvalues of even very nice Toeplitz matrices can be unexpectedly erratic," in *The Diversity and Beauty of Applied Operator Theory*, A. Böttcher, D. Potts, P. Stollmann, and D. Wenzel, Eds., Cham: Springer International Publishing, 2018, pp. 51–77, ISBN: 978-3-319-75996-8.

[5] N. Bender, A. Yamilov, H. Yılmaz, and H. Cao, "Fluctuations and Correlations of Transmission Eigenchannels in Diffusive Media," *Physical Review Letters*, vol. 125, no. 16, p. 165 901, Oct. 14, 2020. DOI: 10.1103/PhysRevLett.125.165901.

[6] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017. DOI: 10.1137/141000671.

[7] J. M. Bogoya, A. Böttcher, S. M. Grudsky, and E. A. Maximenko, "Eigenvectors of Hermitian Toeplitz matrices with smooth simple-loop symbols," *Linear Algebra and its Applications*, vol. 493, pp. 606–637, 2016, ISSN: 0024-3795. DOI: 10.1016/j.laa.2015.12.017.

[8] J. Bogoya, A. Böttcher, S. Grudsky, and E. Maximenko, "Eigenvalues of Hermitian Toeplitz matrices with smooth simple-loop symbols," *Journal of Mathematical Analysis and Applications*, vol. 422, no. 2, pp. 1308–1334, Feb. 15, 2015, ISSN: 0022-247X. DOI: 10.1016/j.jmaa.2014.09.057.

[9] M. Bolten, S.-E. Ekström, and I. Furci, "Momentary Symbol: Spectral Analysis of Structured Matrices," *arXiv e-prints*, arXiv:2010.06199, Oct. 1, 2020.

[10] E. Carson, N. Knight, and J. Demmel, "An efficient deflation technique for the communication-avoiding conjugate gradient method," *Electronic transactions on numerical analysis ETNA*, vol. 43, pp. 125–141, Jan. 1, 2014.

[11] O. Diekmann, J. A. P. Heesterbeek, and J. A. J. Metz, "On the definition and the computation of the basic reproduction ratio R0 in models for infectious diseases in heterogeneous populations," *Journal of Mathematical Biology*, vol. 28, no. 4, pp. 365–382, Jun. 1, 1990, ISSN: 1432-1416. DOI: 10.1007/BF00178324.

[12] M. Donatelli, M. Mazza, and S. Serra-Capizzano, "Spectral analysis and structure preserving preconditioners for fractional diffusion equations," *Journal of Computational Physics*, vol. 307, pp. 262–279, Feb. 15, 2016, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.11.061.

[13] J. J. Dongarra, C. B. Moler, and J. H. Wilkinson, "Improving the Accuracy of Computed Eigenvalues and Eigenvectors," *SIAM Journal on Numerical Analysis*, vol. 20, no. 1, pp. 23–45, Feb. 1, 1983, ISSN: 0036-1429. DOI: 10.1137/0720002.

[14] S.-E. Ekström, "Matrix-Less Methods for Computing Eigenvalues of Large Structured Matrices," PhD Thesis, Uppsala University, 2018, ISBN: 978-91-513-0288-1.

[15] S.-E. Ekström, I. Furci, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers, "Are the eigenvalues of the B-spline isogeometric analysis approximation of $-\Delta u=\lambda u$ known in almost closed form?" *Numerical Linear Algebra with Applications*, vol. 25, no. 5, e2198, Jul. 2018. DOI: 10.1002/nla.2198.

[16] S.-E. Ekström, I. Furci, and S. Serra-Capizzano, "Exact formulae and matrix-less eigensolvers for block banded symmetric Toeplitz matrices," *BIT Numerical Mathematics*, vol. 58, no. 4, pp. 937–968, Jul. 2018. DOI: 10.1007/s10543-018-0715-z.

[17] S.-E. Ekström and C. Garoni, "A matrix-less and parallel interpolation–extrapolation algorithm for computing the eigenvalues of preconditioned banded symmetric Toeplitz matrices," *Numerical Algorithms*, vol. 80, no. 3, pp. 819–848, Mar. 1, 2019, ISSN: 1572-9265. DOI: 10.1007/s11075-018-0508-0.

[18] S.-E. Ekström, C. Garoni, and S. Serra-Capizzano, "Are the Eigenvalues of Banded Symmetric Toeplitz Matrices Known in Almost Closed Form?" *Experimental Mathematics*, vol. 27, no. 4, pp. 478–487, May 2017. DOI: 10.1080/10586458.2017.1320241.

[19] S.-E. Ekström and P. Vassalos. "A matrix-less method to approximate the spectrum and the spectral function of Toeplitz matrices with complex eigenvalues." arXiv: 1910.13810. (2019).

[20] S.-E. Ekström and P. Vassalos, "A matrix-less method to approximate the spectrum and the spectral function of Toeplitz matrices with real eigenvalues," *Numerical Algorithms*, Jun. 24, 2021, ISSN: 1572-9265. DOI: 10.1007/s11075-021-01130-9.

[21] C. Garoni and S. Serra-Capizzano, "Generalized Locally Toeplitz Sequences: Theory and Applications," Department of Information Technology, Uppsala University, 2017-002, Feb. 2017.

[22] C. Garoni and S. Serra-Capizzano, *Generalized Locally Toeplitz Sequences: Theory and Applications*. Springer International Publishing, 2017, vol. 1. DOI: 10.1007/978-3-319-53679-8.

[23] C. Garoni and S. Serra-Capizzano, *Generalized Locally Toeplitz Sequences: Theory and Applications*. Springer International Publishing, 2018, vol. 2. DOI: 10.1007/978-3-030-02233-4.

[24] C. Garoni, H. Speleers, S.-E. Ekström, A. Reali, S. Serra-Capizzano, and T. J. R. Hughes, "Symbol-Based Analysis of Finite Element and Isogeometric B-Spline Discretizations of Eigenvalue Problems: Exposition and Review," *Archives of Computational Methods in Engineering*, vol. 26, no. 5, pp. 1639–1690, Nov. 1, 2019, ISSN: 1886-1784. DOI: 10.1007/s11831-018-9295-y.

[25] Julia Linear Algebra. "Julia Linear Algebra - Generic Linear Algebra." (Jun. 2, 2021), [Online]. Available: https://github.com/JuliaLinearAlgebra/GenericLinearAlgebra.jl/ (visited on 06/02/2021).

[26] Julia Math. "Julia Math - DoubleFloats." (Jun. 2, 2021), [Online]. Available: https://github.com/JuliaMath/DoubleFloats.jl (visited on 06/02/2021).

[27] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. Springer New York, 2007. DOI: 10.1007/b98885.

[28] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, "A Deflated Version of the Conjugate Gradient Algorithm," *SIAM Journal on Scientific Computing*, vol. 21, no. 5, pp. 1909–1926, Jan. 1, 2000, ISSN: 1064-8275. DOI: 10/dcxjkb.

[29] R. Scherer, S. L. Kalla, Y. Tang, and J. Huang, "The Grünwald–Letnikov method for fractional differential equations," *Special Issue on Advances in Fractional Differential Equations II*, vol. 62, no. 3, pp. 902–917, Aug. 1, 2011, ISSN: 0898-1221. DOI: 10.1016/j.camwa.2011.03.054.

[30] S. Serra-Capizzano, "Generalized locally Toeplitz sequences: Spectral analysis and applications to discretized partial differential equations," *Linear Algebra and its Applications*, vol. 366, pp. 371–402, 2003, ISSN: 0024-3795. DOI: 10.1016/S0024-3795(02)00504-9.

[31] R. Smith. "GenericSchur.jl." (Jun. 2, 2021), [Online]. Available: https://github.com/RalphAS/GenericSchur.jl.

[32] R. Smith. "IterativeRefinement.jl." (Jun. 2, 2021), [Online]. Available: https://github.com/RalphAS/IterativeRefinement.jl.

[33] P. Tilli, "Locally Toeplitz sequences: Spectral properties and applications," *Linear Algebra and its Applications*, vol. 278, no. 1, pp. 91–120, 1998, ISSN: 0024-3795. DOI: 10.1016/S0024-3795(97)10079-9.

[34] Y. Xirouhakis, G. Votsis, and A. Delopoulos, "Estimation of 3D Motion and Structure of Human Faces," in *Advances in Intelligent Systems: Concepts, Tools and Applications*, S. G. Tzafestas, Ed., Dordrecht: Springer Netherlands, 1999, pp. 333–344, ISBN: 978-94-011-4840-5. DOI: 10.1007/978-94-011-4840-5_30.

A

Code

- Construction of Toeplitz matrices

```
function toeplitz(n :: Integer, vc :: Array{T,1}, vr :: Array{T,1}) where T
    s  = size(vc[1],2)
    Tn = zeros(eltype(T),s*n,s*n)
    for ii = 1:length(vc)
        Tn = Tn + kron(diagm(-ii+1=>ones(eltype(T),n-ii+1)),vc[ii])
    end
    for jj = 2:length(vr)
        Tn = Tn + kron(diagm( jj-1=>ones(eltype(T),n-jj+1)),vr[jj])
    end
    return Tn
end
```

- Eigenvectors of Laplacian matrix

```
function LapEV(n :: Integer, p:: Integer, DT :: DataType)
    v = [2, -1]
    mat = toeplitz(n, convert.(DT, v), convert.(DT, v))
    EV = real(eigvecs(mat))
    return EV[:,p]
end
```

- Eigenvectors of bi-Laplacian matrix

```
function biLapEV(n :: Integer, p:: Integer, DT :: DataType)
    v = [6, -4, 1]
    mat = toeplitz(n, convert.(DT, v), convert.(DT, v))
    EV = real(eigvecs(mat))
    return EV[:,p]
end
```

- Eigenvectors of IGA matrix

```
function IGA_Mass_mat(n ::Integer, DT :: DataType)
    v = [4832, 2382, 240, 2]
    M = toeplitz(n,convert.(DT,v), convert.(DT,v))

    B_term = [2232 1575 348 3;
    1575 3294 2264 239;
    348 2264 4832 2382;
    3 239 2382 4832]
    M[1:4, 1:4] = convert.(DT, B_term)
    M[end-3:end, end-3:end] = rotl90(convert.(DT, B_term), 2)

    return convert(DT, (1/10080)) .* M
end
```

```
function IGA_Stiff_mat(n ::Integer, DT :: DataType)
    v = [160, -30, -48, -2]
    M = toeplitz(n,convert.(DT,v), convert.(DT,v))

    B_term = [360 9 -60 -3;
    9 162 -8 -47;
    -60 -8 160 -30;
    -3 -47 -30 160]
    M[1:4, 1:4] = convert.(DT, B_term)
    M[end-3:end, end-3:end] = rotl90(convert.(DT, B_term), 2)

    return convert(DT, 1/240) .* M
end

function IGAEV(n :: Integer, p:: Integer, DT :: DataType)
    M = IGA_Mass_mat(n, DT)
    K = IGA_Stiff_mat(n, DT)
    L = M \ K
    EV = real(eigvecs(L))
    return EV[:,p]
end
```

- Eigenvectors of symmetrised fractional derivative matrix

```
function generateFracDerivTn(n :: Integer, alpha :: AbstractFloat,
                             DT :: DataType)
    g=zeros(n+1)
    g[1]=1
    for ii=2:n+1
        g[ii]=g[ii-1]*(-(alpha-(ii-2))/(ii-1))
    end

    Tn=-toeplitz(n,convert.(DT, g[2:n+1]),convert.(DT, [g[2],g[1]]))
    return Tn
end

function FracDerivEV(n :: Integer, p:: Integer, DT :: DataType)
    Tn = generateFracDerivTn(n, 1.5, DT)
    mat = (Tn + Tn') ./ 2
    EV = real(eigvecs(mat))
    return EV[:,p]
end
```