UPPSALA
UNIVERSITET

Ordering the Eigenvalues of Toeplitz Matrices

Investigating eigenvalue orderings of matrices generated by non-monotone symbols

Johan Bürger, Jonathan Carlgren,
Per William Oskarsson, Simon Persson

Project in Computational Science: Report

January 2023

PROJECT REPORT

Institutionen för informationsteknologi

Abstract

Toeplitz matrices arise in many settings in scientific computing, and the spectrum (eigenvalues) of these matrices is often of interest. These matrices are generated by a function called the symbol, and if they are symmetric (Hermitian) then the spectrum behaves asymptotically as this function. Matrix-less methods (MLM) are one of the numerical tools that efficiently approximates the spectrum of these matrices with high accuracy. However, for matrices generated by non-monotone symbols these methods do not work for the full spectrum; only the spectrum in the monotone regions can be accurately approximated. One hypothesis that has arisen as a result is that there exists some intrinsic ordering of the eigenvalues which hopefully would make MLM work for the full spectrum even for non-monotone symbols.

This project aims to sort these eigenvalues by introducing four different sorting methods: symbol sort, DST sort, imaginary sort and similar sort. These sorting methods are used on mainly two different non-monotone generating symbols, and tested by applying these tests; the MLM, symmetry test and Hankel test.

The results show that the methods have different strengths and weaknesses, and no method is truly able to sort all eigenvalues of a Toeplitz matrix with non-monotone symbol consistently. Symbol sort does not work due to the remaining error of the eigenvalue approximation and DST sort has issues when there are different eigenvectors with maximum amplitude at the same frequency. Imaginary sort gives a correct ordering, but for a matrix with with slightly perturbed eigenvalues compared to the matrix of interest. Similar sort can break down due to errors in the perturbed eigenvalues being greater than the difference between the two closest ones. In future projects one could analyze the frequency spectrum of the eigenvectors further to improve DST sort. To improve imaginary sort a more in depth analysis of how to minimize the perturbation of the eigenvalues is necessary.

Contents

1 Introduction

In the last couple of years new types of eigenvalue solvers, matrix-less methods (MLM), have been developed; e.g., see [6, 15]. They exploit the fact that eigenvalues of certain matrix sequences, so called generalized locally Toeplitz (GLT) sequences, can be approximated using an asymptotic expansion. The current version of MLM works exclusively for the full spectrum for matrices with monotone generating symbols since the ordering of the eigenvalues for those matrices is trivial. MLM also works for the monotone part of a non-monotone symbol if such part exists. For matrices with fully non-monotone generating symbols MLM breaks down. The objective of this project is to find methods to correctly order the eigenvalues, with the hope that the full spectrum of matrices with non-monotone generating symbols can be accurately approximated using MLM.

1.1 Toeplitz matrices and their symbols

This project focuses on real symmetric Toeplitz matrices. These matrices appear frequently when discretizing PDE:s into linear systems when using numerical methods; such as for example finite elements, finite volumes, and finite differences. They are named after the German mathematician Otto Toeplitz and are defined by their constant diagonals. A square Toeplitz matrix of size $n \times n$ has the form

$$T_n(f) = \begin{bmatrix} \hat{f}_0 & \hat{f}_{-1} & \cdots & \cdots & \hat{f}_{1-n} \\ \hat{f}_1 & \hat{f}_0 & \hat{f}_{-1} & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \hat{f}_1 & \hat{f}_0 & \hat{f}_{-1} \\ \hat{f}_{n-1} & \ddots & \ddots & \hat{f}_1 & \hat{f}_0 \end{bmatrix}. \tag{1}$$

Here f is the so called generating symbol and $\hat{f}_k$, where k goes from $1-n$ to $n-1$ are the Fourier coefficients of $f(\theta)$, that is,

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

The Fourier sum of the symbol f is

$$f(\theta) = \sum_{k=-\infty}^{\infty} \hat{f}_k e^{k\hat{\imath}\theta}. \tag{2}$$

Example 1.1.1. *One example of a Toeplitz matrix that is studied throughout this project is the second order finite difference discretisation of the bi-Laplacian (∇^4). It has the diagonal entries: $\hat{f}_0 = 6$, $\hat{f}_{-1} = \hat{f}_1 = -4$, $\hat{f}_{-2} = \hat{f}_2 = 1$ and $\hat{f}_k = 0$ for all other k,*

$$T_6(f) = \begin{bmatrix} 6 & -4 & 1 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 1 & -4 & 6 & -4 \\ 0 & 0 & 0 & 1 & -4 & 6 \end{bmatrix}. \tag{3}$$

The generating symbol for this matrix is

$$\begin{aligned} f(\theta) &= 1e^{-2\hat{\imath}\theta} - 4e^{-\hat{\imath}\theta} + 6 - 4e^{\hat{\imath}\theta} + 1e^{2\hat{\imath}\theta} \\ &= 6 - 8\cos(\theta) + 2\cos(2\theta). \end{aligned} \tag{4}$$

1.2 Monotone function

A function is monotone if it is either non-increasing or non-decreasing, see Example 1.2.1. Since the order of the eigenvalues is related to the symbol, whether a symbol is monotone or not is an important property when it comes to sorting the eigenvalues. If we are interested in the whole spectrum the current version of the MLM only works for matrices with monotone symbols, such as the bi-Laplacian in (3). For matrices generated by non-monotone symbols it only works for the monotone regions, see the following example.

Example 1.2.1. *Monotone bi-Laplacian and non-monotone modified bi-Laplacian*

bi-Laplacian

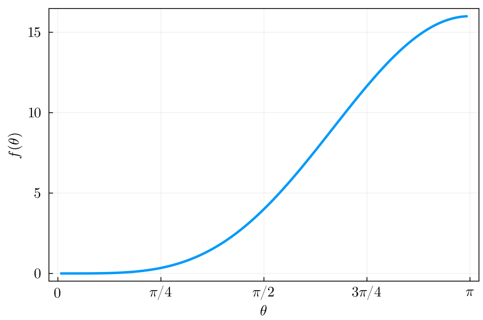
$$\begin{bmatrix} 6 & -4 & \mathbf{1} & 0 & 0 & 0 \\ -4 & 6 & -4 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & -4 & 6 & -4 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & -4 & 6 & -4 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & -4 & 6 & -4 \\ 0 & 0 & 0 & \mathbf{1} & -4 & 6 \end{bmatrix}$$

$$f(\theta) = 6 - 8\cos(\theta) + \mathbf{2}\cos(2\theta)$$

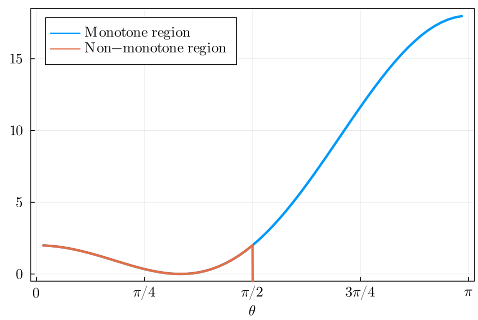
Modified bi-Laplacian

$$\begin{bmatrix} 6 & -4 & \mathbf{2} & 0 & 0 & 0 \\ -4 & 6 & -4 & \mathbf{2} & 0 & 0 \\ \mathbf{2} & -4 & 6 & -4 & \mathbf{2} & 0 \\ 0 & \mathbf{2} & -4 & 6 & -4 & \mathbf{2} \\ 0 & 0 & \mathbf{2} & -4 & 6 & -4 \\ 0 & 0 & 0 & \mathbf{2} & -4 & 6 \end{bmatrix}$$

$$f(\theta) = 6 - 8\cos(\theta) + \mathbf{4}\cos(2\theta)$$



Monotone symbol



Non-monotone symbol

(5)

1.3 GLT approximation

From the theory of GLT sequences ([14]) we know that the eigenvalues of a matrix generated by even trigonometric polynomials, $f(\theta) = \sum_{k=-\infty}^{\infty} \hat{f}_k \cos(k\theta)$, can be approximated by sampling the generating symbol on an equispaced grid $\theta_{j,n} \in [0, \pi]$. This GLT eigenvalue approximation is described as follows,

$$\lambda_j(A_n) = f(\theta_{j,n}) + E_{j,n,0} \tag{6}$$

where $f(\theta_{j,n})$ is the sampled generating symbol and $E_{j,n,0}$ is the error which tends to 0 when the matrix size n approaches infinity [6]. In the report the matrix $A_n = T_n(f) + R_n$ is a Toeplitz-like matrix where R_n is a low rank perturbation matrix; GLT matrices are a wider class of matrices where the presented theory would apply, see [14].

1.4 Matrix-less method

Matrix-less methods (MLM) are based on the assumption of the existence of the asymptotic expansion

$$\begin{aligned} \lambda_j(A_n) &= f(\theta_{j,n}) + E_{j,n,0} \\ &= f(\theta_{j,n}) + \sum_{k=1}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha} \\ &= \sum_{k=0}^{\alpha} h^k c_k(\theta_{j,n}) + E_{j,n,\alpha}. \end{aligned} \tag{7}$$

.

For a symmetric Toeplitz-like matrix A_n, with $\{A_n\}_n \sim_\lambda f$, the GLT eigenvalue approximation is given by $f(\theta_{j,n}) = c_0(\theta_{j,n})$[14]. The higher order symbols $c_1, ..., c_\alpha$ reduces the error of the approximation of the eigenvalues.

The MLM is used to approximate the eigenvalues of a large Toeplitz matrix A_{n_f} of size $n_f \times n_f$. This is done by calculating the the eigenvalues of a sequence of smaller matrices A_{n_k} with sizes $n_k \times n_k$, where $n_k = 2^k(n_0+1) - 1$ and $k = 0, ..., \alpha$. This is done by some standard numerical eigenvalue solver such as `eigvals` in Julia[3] or `eig` in Matlab. With the calculated sequence of eigenvalues E the approximation of the higher order symbols $\tilde{C}$ for the grid θ_{j,n_0} are calculated by solving the system $E = V\tilde{C}$, where

$$E = \begin{bmatrix} \lambda_1(A_{n_0}) & \lambda_2(A_{n_0}) & \lambda_3(A_{n_0}) & \dots & \lambda_{n_0}(A_{n_0}) \\ \lambda_2(A_{n_1}) & \lambda_4(A_{n_1}) & \lambda_6(A_{n_1}) & \dots & \lambda_{2n_1}(A_{n_0}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{2\alpha}(A_{n_\alpha}) & \lambda_{4\alpha}(A_{n_\alpha}) & \lambda_{6\alpha}(A_{n_\alpha}) & \dots & \lambda_{2\alpha n_0}(A_{n_\alpha}) \end{bmatrix}, \tag{8}$$

$$\tilde{C} = \begin{bmatrix} \tilde{c}_0(\theta_{1,n_0}) & \tilde{c}_0(\theta_{2,n_0}) & \dots & \tilde{c}_0(\theta_{n_0,n_0}) \\ \tilde{c}_1(\theta_{1,n_0}) & \tilde{c}_1(\theta_{2,n_0}) & \dots & \tilde{c}_1(\theta_{n_0,n_0}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_\alpha(\theta_{1,n_0}) & \tilde{c}_\alpha(\theta_{2,n_0}) & \dots & \tilde{c}_\alpha(\theta_{n_0,n_0}) \end{bmatrix}, \tag{9}$$

$$V = \begin{bmatrix} 1 & h_0 & h_0^2 & h_0^3 & \dots & h_0^\alpha \\ 1 & h_1 & h_1^2 & h_1^3 & \dots & h_1^\alpha \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & h_\alpha & h_\alpha^2 & h_\alpha^3 & \dots & h_\alpha^\alpha \end{bmatrix}, \quad h_k = \frac{1}{1+n_k}. \tag{10}$$

The higher order symbols of the θ_{j,n_0} grid is then interpolated and extrapolated to obtain the higher order symbol samplings for the θ_{j,n_f} grid. At this point (7) is used to finally approximate the eigenvalues of the large $n_f \times n_f$ matrix, [1, 7, 8, 9, 10, 11, 12].
In this project MLM is used as a test to see if the the eigenvalues are correctly ordered. By assuming the correct ordering leads to continuous higher order symbols we look at $c_1, c_2, \dots$ to verify the results. If the higher order symbols are erratic the ordering of eigenvalues is classified as incorrect.

Working Hypothesis: *Given a Toeplitz(-like) matrix with non-monotone generating symbol, there exists a correct ordering of the eigenvalues such that MLM works for these matrices.*

This report is organized as follows. In the following sections four different approaches to finding the correct order of the eigenvalues are explored. Section 2 includes descriptions of the implemented sorting methods and tests used to verify the correctness of the orderings. In Section 3 the sorting methods are applied to matrices generated by non-monotone symbols and the results are discussed. The findings are then summarized in Section 4 together with ideas of future works.

2 Sorting Strategies and Ordering Tests

2.1 Sort methods

Here follows explanations of the four heuristic sorting sorting methods used in this project: symbol sort, DST sort, imaginary sort, and similar sort.

2.1.1 Symbol sort

This sorting method is based on the GLT eigenvalue approximation. The generating symbol of the matrix is sampled on an equispaced grid which produces an approximation of the eigenvalues with a certain ordering. The eigenvalues of the matrix are then calculated with some built in function, like `eigvals` in Julia, and then sorted with the ordering from the GLT approximation.

2.1.2 DST sort

An important tool in signal processing is the discrete Fourier transform (DFT), since it allows us to analyze signals in the frequency domain by calculating the frequency spectrum. Given a finite set of discrete samplings of a function, the DFT transforms it into samplings of its frequency domain function. It is calculated as

$$F_k = \sum_{j=0}^{n-1} \hat{f}_k \left(\cos\left(\frac{2\pi kj}{n}\right) - \hat{\imath} \sin\left(\frac{2\pi kj}{n}\right)\right), \tag{11}$$

where $\hat{f}_k$ are the Fourier coefficients. When the number of samples increase, the DFT quickly becomes expensive to calculate. Instead, one can use the fast Fourier transform (FFT) algorithm to efficiently compute the DFT. In JULIA, the FFT is implemented in the package FFTW.JL [13].

The discrete sine transform (DST) is a family of transforms similar to the DFT and discrete cosine transforms (DCT). As for DFTs, the discrete sine transform calculates the frequency spectrum of a set of finite, sampled data points. Instead of describing the frequency spectrum as a combination of sines and cosines as in DFTs, the DST only uses sines. In the discrete setting, there exists multiple different sine transforms due to different choices when specifying boundary conditions [5], for instance

$$Y_k = 2\sum_{j=0}^{n-1} X_j \sin\left(\frac{\pi(j+1)(k+1)}{n+1}\right) \tag{12}$$

is an example of transform DST-I.

Similarly, DCT exists as another option but consists of a sum of cosines and has 8 different transforms. Both the DST and DCT exists in the FFTW.JL package.

Another way to express the DST-I is through the DST-I matrix. It is an $n \times n$ matrix with normalized entries calculated as $[S]_{j,k=1}^{n} = \sqrt{\frac{2}{n+1}} \sin\left(\frac{\pi jk}{n+1}\right)$. If you would take the DST-I of a vector x, the same result would be obtained by multiplying it with the DST-I matrix S_n. What is interesting is that for the matrix generated by the non-monotone bi-Laplacian symbol $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ but with the number 4 instead of 6 in both corners, the DST-I matrix gives the exact eigenvector matrix and diagonalizes the generated matrix. When having the number 6 in the corners, the DST-I matrix is close to diagonalizing the generated matrix since the two cases are similar.

It is observed that the eigenvectors follows a sinusoidal behaviour with increasing frequencies. Naturally, the Fourier transform comes to mind and our first proposal is to sort the eigenvectors by their frequencies. We start by investigating the behaviour of the eigenvectors in the frequency domain using a variety of transforms from the FFTW.JL library in JULIA. Using the observations, we attempt to create a sorting algorithm using the frequency behaviour of the eigenvectors. The algorithm begins with the transformations using the `FFTW.rfft` and `FFTW.r2r` functions, these include the standard FFT, the DST (DST-I, -II, -III, -IV) and the DCT (DCT-I, -II, -III, -IV). It then orders the eigenvectors by the frequencies of maximum amplitude and rearranges the result form JULIA's standard eigenvalue solver in this new ordering.

2.1.3 Imaginary sort

A third approach to sorting the eigenvalues involves creating a complex Toeplitz matrix by adding an imaginary matrix constructed from a monotone symbol $g(\theta)$ to the real non-monotone matrix constructed by $f(\theta)$. Together, the overall symbol becomes $h(\theta, \delta) = f(\theta) + \hat{\imath}\delta g(\theta)$, and the resulting complex Toeplitz matrix reads

$$T_n(h(\theta, \delta)) = T_n(f) + \hat{\imath}\delta T_n(g). \tag{13}$$

Now, the matrix $T_n(h)$ can be accurately sorted in the monotone order of the imaginary part. However, there is a trade-off. By adding an imaginary part, the nature of the real eigenvalue components is slightly altered compared to those from the corresponding real matrix $T_n(f)$. When the scaling factor δ is decreased, $T_n(h)$ approaches $T_n(f)$.

An alternative method is to first sort the non-monotone eigenvalues of the real matrix and the complex matrix eigenvalues in increasing order according to its real part (automatically done with the function `eigvals` in JULIA). Secondly sort the complex matrix eigenvalues according to the increasing imaginary part and save how eigenvalue positions change. We then change the order of the real eigenvalues using the so-obtained ordering. This would give us the exact eigenvalues, but the ordering might be faulty.

2.1.4 Similar sort

The similar sort method constructs a new generating symbol $h(\theta)$ by adding an imaginary generating symbol $\hat{\imath}g(\theta)$ to $f(\theta)$ which is the generating symbol of the original matrix with the eigenvalues we want to sort. The similar matrix $T_n(g(\theta))$ has the same eigenvalues as $T_n(f(\theta))$ but in the reverse order [4, 16].

$$T_n(h(\theta)) = T_n(f(\theta)) + \hat{\imath}T_n(g(\theta)) \tag{14}$$

For the method to be able to sort all the eigenvalues of the whole spectrum $f(\theta)$ has to be monotone and uniquely valued for at least one half of the spectrum. Uniquely valued means the values in that part of the spectrum do not appear anywhere else. The next step of the method is to calculate and sort the eigenvalues of $T_n(h(\theta))$. If the monotone part of $f(\theta)$ is non-decreasing the real part of the eigenvalues should be non-decreasingly sorted and if the monotone part of $f(\theta)$ is non-increasing the real part of the eigenvalues should be non-increasingly sorted. Then the real part of the eigenvalues will be correctly sorted in the monotone part of the spectrum. Because of this, by assumption, the imaginary part of the eigenvalues correspond to the eigenvalues in the non-monotone part of the spectrum in reverse order. Thus, $\lambda_n(T_n(h))$ corresponds to $\lambda_1(T_n(f))$, $\lambda_{n-1}(T_n(h))$ corresponds to $\lambda_2(T_n(f))$ and so on.

2.2 Ordering Tests

The following three tests are used to check the correctness of the orderings produced by the sorting methods; Hankel, symmetry, and MLM.

2.2.1 Hankel test

Here the Hankel matrix $H_n(f)$ is defined as the anti-diagonal version of the Toeplitz matrix $T_n(f)$. The eigenvalues of the Hankel matrix has the same magnitude as the eigenvalues of the Toeplitz matrix but with opposite sign for every other eigenvalue ([2]). We have,

$$H_n(f) = Y_nT_n(f), \qquad Y_n = \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & \iddots & & \\ & 1 & & & \\ 1 & & & & \end{bmatrix}, \tag{15}$$

and

$$\lambda_j(H_n(f)) = (-1)^{(i+1)}\lambda_j(T_n(f)), \quad j = 1, \ldots, n. \tag{16}$$

To verify that a sorting algorithm works correctly the algorithm is applied to a Toeplitz matrix and its Hankel version separately. If the resulting eigenvalues do not satisfy the condition (16) they are assumed to be ordered incorrectly.

2.2.2 Symmetry test

Another verification method comes by checking the symmetry of the elements of the sorted eigenvectors. For odd-numbered eigenvectors, the elements of the sorted eigenvector should also be odd and vice versa for even-numbered eigenvectors. This is checked by comparing the boundary points. If the eigenvector is odd, it is symmetric meaning the sum of the boundary points should be approximately 0. Even eigenvectors should be skew-symmetric, with the difference between boundary points approximately 0. For the even eigenvectors the value at the middle point $\theta = \pi/2$ should also be close to 0. An example is given in Figure 1.

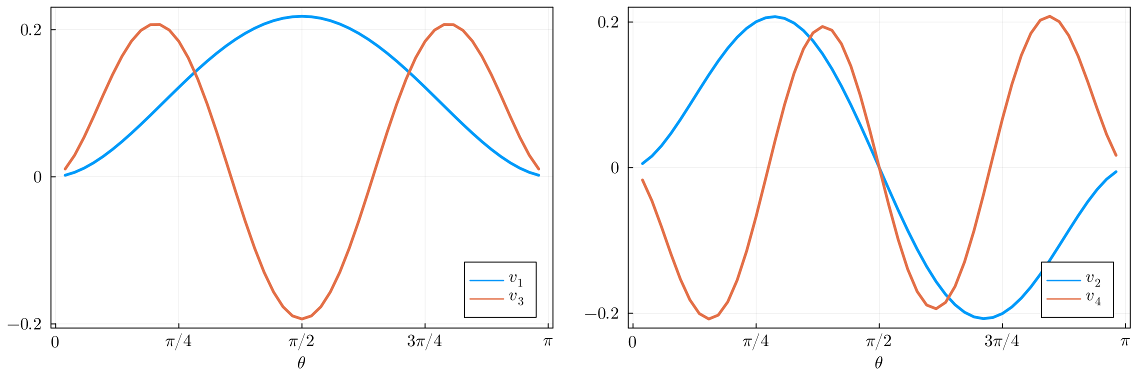


Figure 1: Eigenvectors for the monotone bi-Laplacian symbol $f(\theta) = 6 - 8\cos(\theta) + 2\cos(2\theta)$ sampled on a equidistant grid with n=127 points. **Left:** Odd numbered eigenvectors. **Right:** Even numbered eigenvectors

2.2.3 MLM test

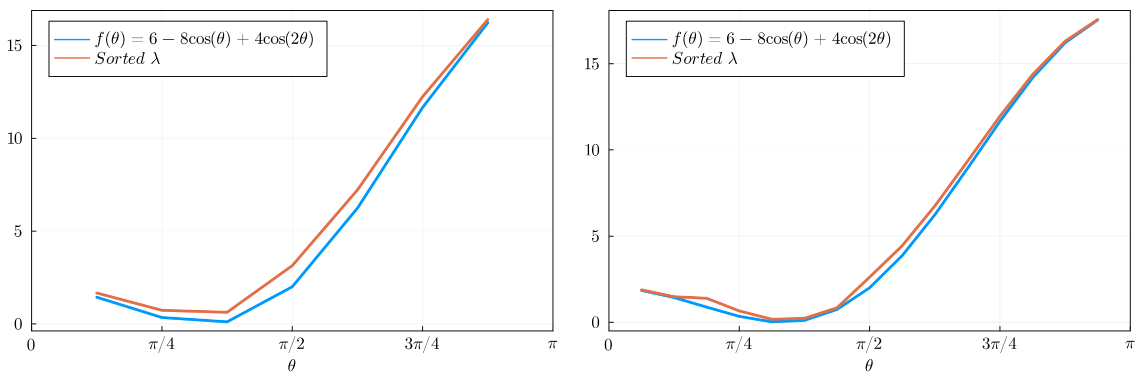
The MLM test analyses the behaviour of the higher order symbols. It is assumed that if the higher order symbols that are not smooth means the ordering of eigenvalues is faulty. The amplitude and pattern of non-smooth higher order symbols may indicate how close each eigenvalue is to its accurate position. An example of smooth curves are the plots in Figure 14. Examples of non-smooth higher order symbols are c_1 and c_2 in Figure 10 and more amplitude-bound non-smooth higher order symbols can be seen in Figure 22.

3 Results and Discussions

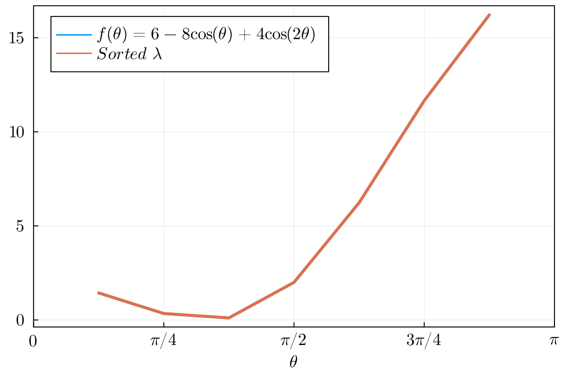
In this section the sorting methods are tested for some Toeplitz matrices of varying size. The main focus is sorting the spectrum generated from the non-monotone modified bi-Laplacian and using the MLM test for evaluating the results.

3.1 Symbol sort

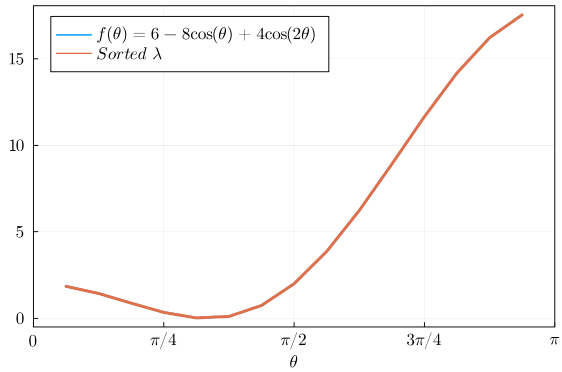
Looking at Figure 2, we see that the symbol sort method successfully manages to sort the eigenvalues for $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ in the special case when the generated matrix has 4 in the corners. Looking at Figure 2, we examine how the sorted eigenvalues for $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ compares to its symbol. In the special case when the generated matrix has 4 in the corners, the eigenvalues follows the symbol. This is because in this case, the eigenvalues are given exactly by $f(\theta_{j,n})$. In the regular case where the corner elements of the matrix are equal to 6, the sorted eigenvalues follows a different shape compared to the symbol. This is because the standard $\theta_{j,n} = \frac{j\pi}{n+1}$ grid is used to sample these eigenvalues which is not always the correct grid to use. For the second symbol $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$ in Figure 4 this is also the case.



(a) n=7 **(b)** n=15

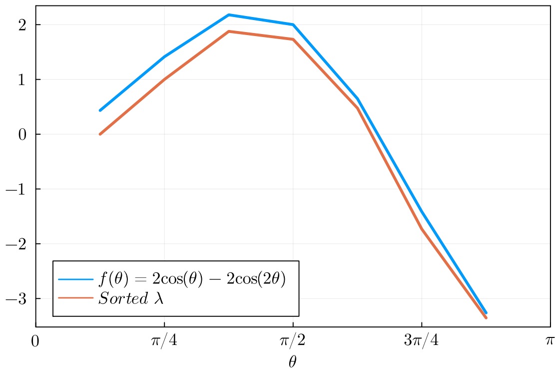


(c) n=7 with $(T_n(f))_{1,1} = (T_n(f))_{n,n} = 4$

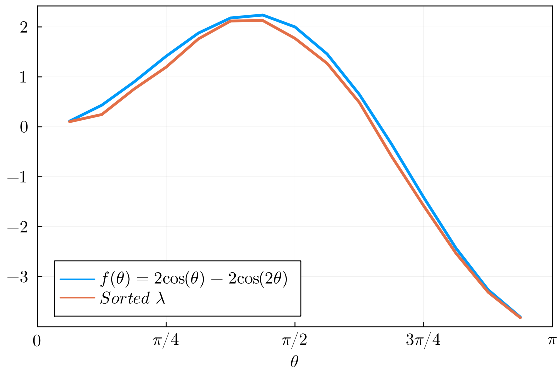


(d) n=15, with $(T_n(f))_{1,1} = (T_n(f))_{n,n} = 4$

Figure 2: Symbol sort method performed using samplings of the symbol $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$



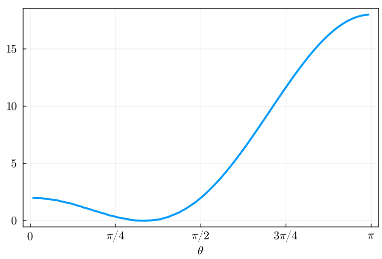
(a) n=7



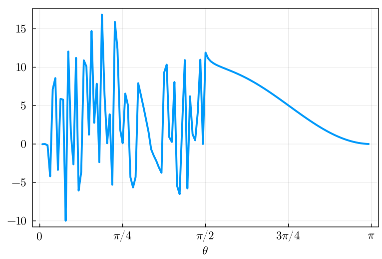
(b) n=15

Figure 3: Symbol sort method performed using samplings of the symbol $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$

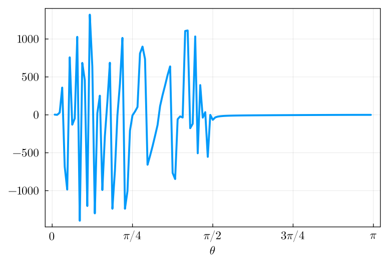
In Figures 4 and 5, no sensible result for any of the symbols are obtained. The MLM has failed, likely due to the ordering of eigenvalues given by the symbol sort method being incorrect. In the regions where the two symbols are monotone, the higher order symbols look nice, but for the other non-monotone regions, it is completely erratic. While sampling the symbol can give an approximation of the eigenvalues for the generated Toeplitz matrix, the error of the approximation is an indicator for an incorrect ordering of this method.



(a) c_0

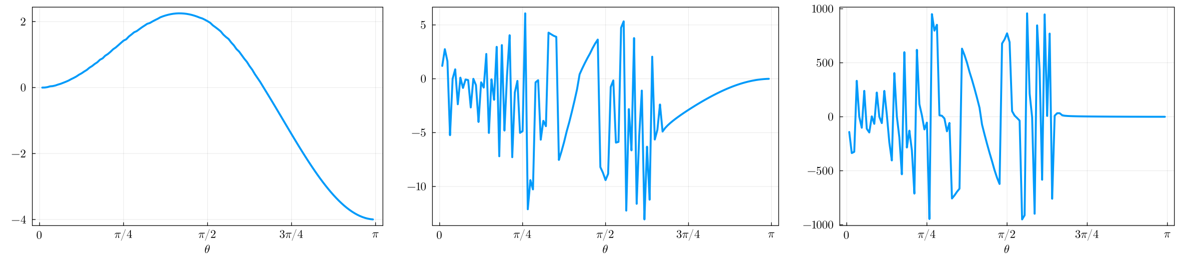


(b) c_1



(c) c_2

Figure 4: Higher order symbols computed for $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ for $n_0 = 127$ using symbol sort

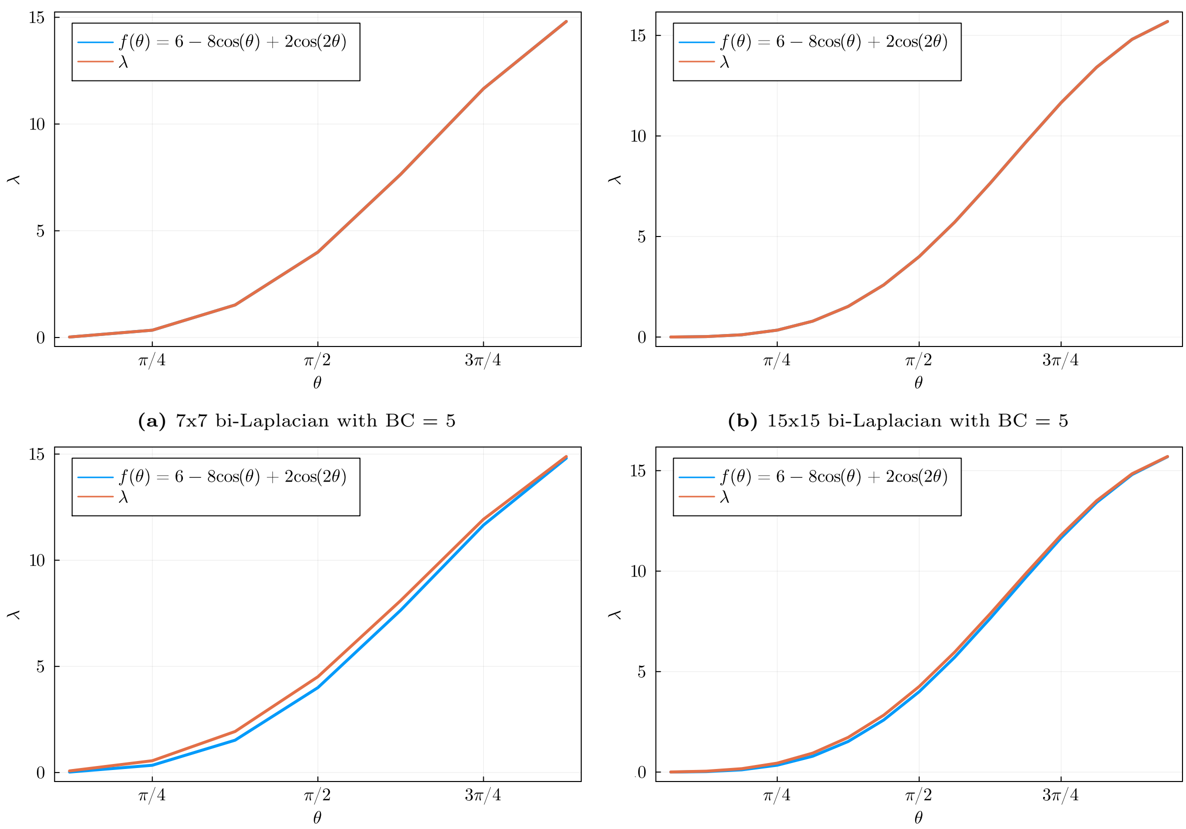


(a) c_0 **(b)** c_1 **(c)** c_2

Figure 5: Higher order symbols computed for $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$ with $n_0 = 127$ using symbol sort

3.2 DST sort

Figure 6 displays the resulting eigenvalue order for the 7x7 and 15x15 bi-Lapacian and modified bi-Lapacian after sorting by maximum amplitude of the DST-I coefficients. The algorithm works for the bi-Laplacian symbols but seems to fail for 15x15 and other larger modified bi-Laplacian. This is in line with the results from the MLM in Figure 7, where the higher order symbols are smooth in the bi-Lapacian case, but grow very large for the modified bi-Laplacian case.

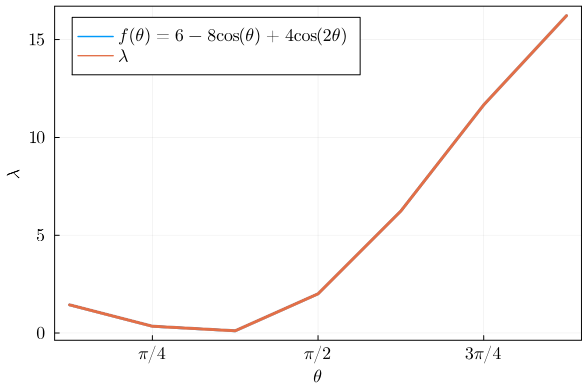


(a) 7x7 bi-Laplacian with BC = 5

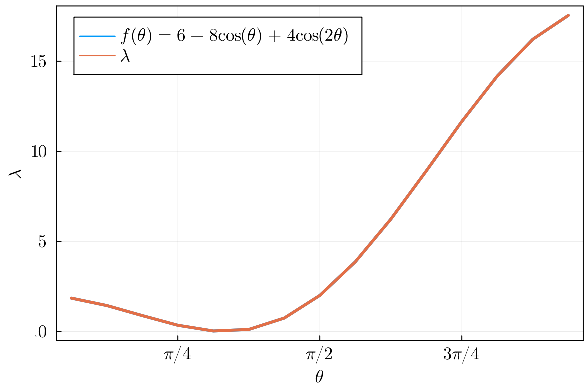
(b) 15x15 bi-Laplacian with BC = 5

(c) 7x7 bi-Laplacian

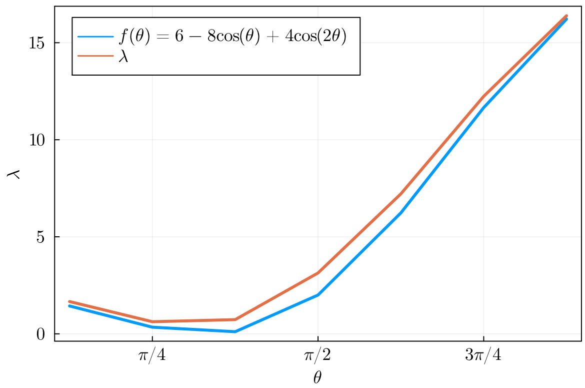
(d) 15x15 bi-Laplacian



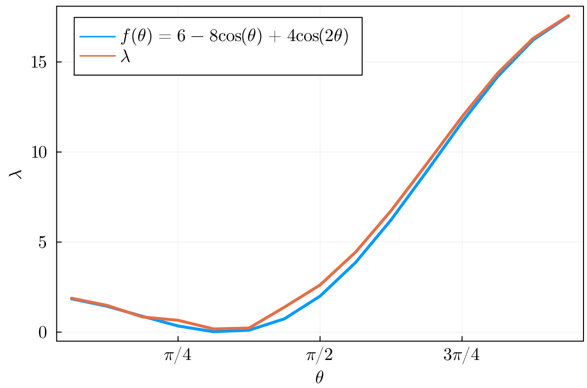
(e) 7x7 modified bi-Laplacian with BC = 4



(f) 15x15 modified bi-Laplacian with BC = 4

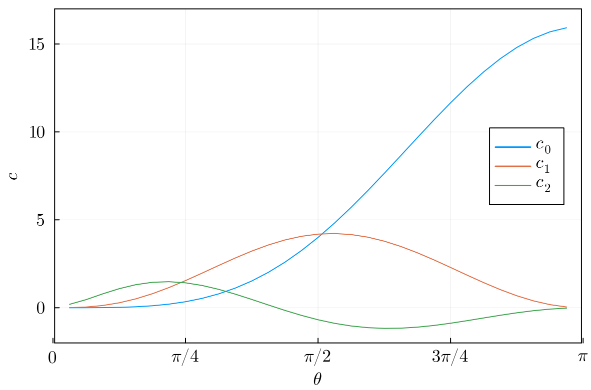


(g) 7x7 modified bi-Laplacian

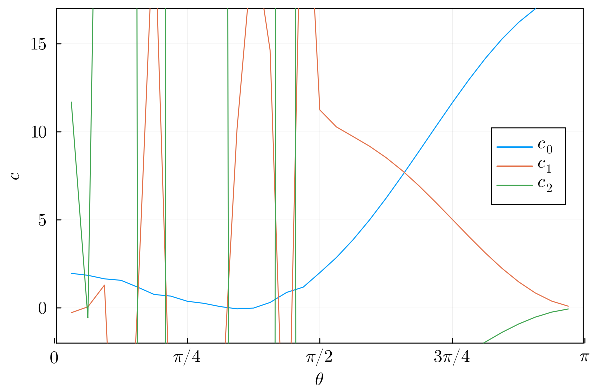


(h) 15x15 modified bi-Laplacian

Figure 6: Resulting eigenvalue order from DST sort together with its symbol



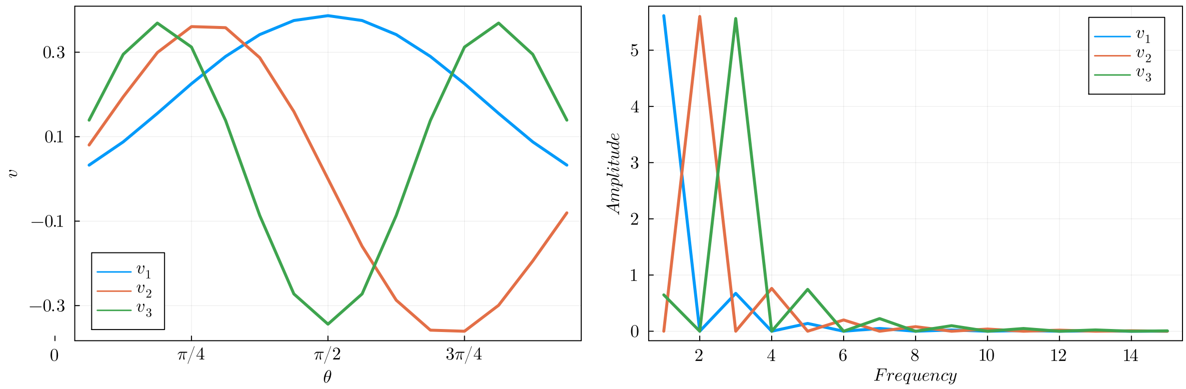
(a) bi-Laplacian



(b) Modified bi-Laplacian

Figure 7: Higher order symbols using MLM with DST sort and $n_0 = 31$

The resulting first three eigenvectors of the bi-Laplacian illustrates the idea behind the method. In Figure 8 the increasing frequencies of the eigenvectors are distinguishable in the frequency domain and the order is found accordingly.

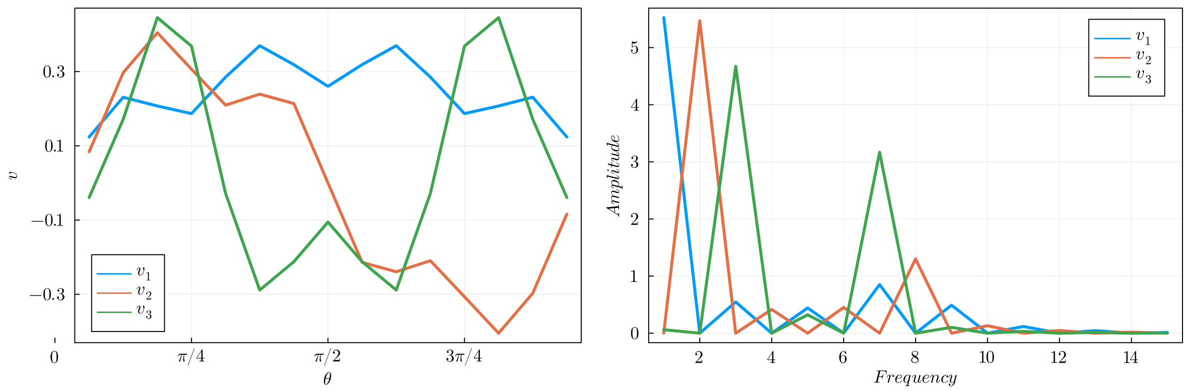


(a) Real domain

(b) Frequency domain

Figure 8: First three eigenvectors after sorting the 15x15 bi-Laplacian with DST sort

For the non-monotone modified bi-Laplacian, in Figure 9, we observe additional high frequency components. For the first three eigenvectors the low frequencies dominates the signals and allows the method to work, but we notice the increasing amplitude of the high frequency components.

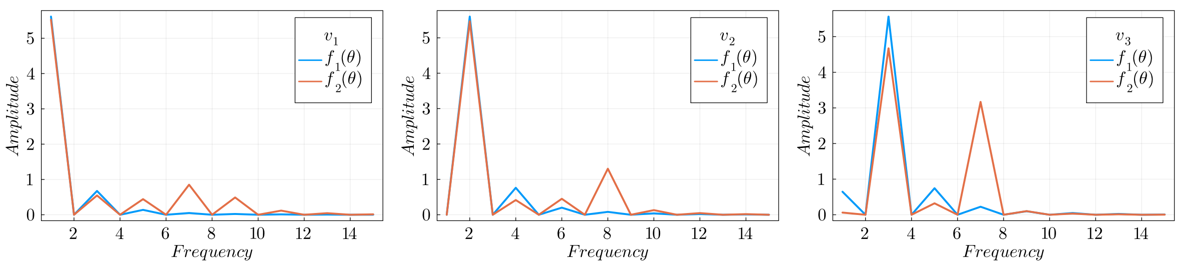


(a) Real domain

(b) Frequency domain

Figure 9: First three eigenvectors after sorting the 15x15 modified bi-Laplacian with DST sort

DST-I for all the eigenvectors are presented in Figure 10. For the bi-Laplacian, solitary amplitude peaks can be seen for all eigenvectors. The modified bi-Laplacian instead shows multiple amplitude peaks in its non-monotone region ($v_1 - v_7$). This proves challenging for the algorithm because e.g. v_4 and v_6 both have major amplitude peaks at the same frequencies 4 and 6. The main peaks ascends by one frequency unit per eigenvector for both matrices, while the significant secondary peaks in the non-monotone region (see v_2 - v_7) descends by one unit per eigenvector with the exception of v_5 where the main and secondary peaks lines up.



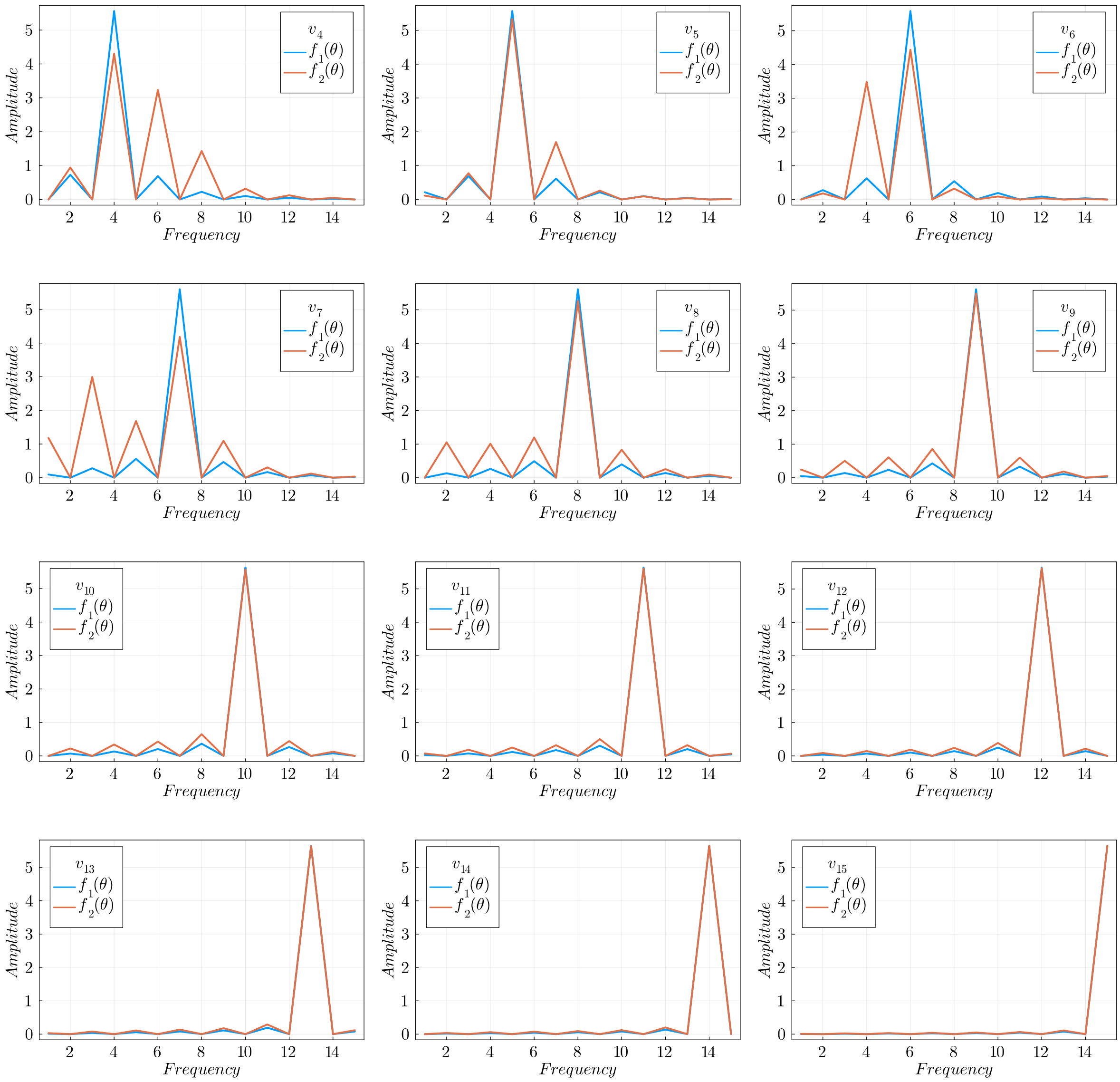
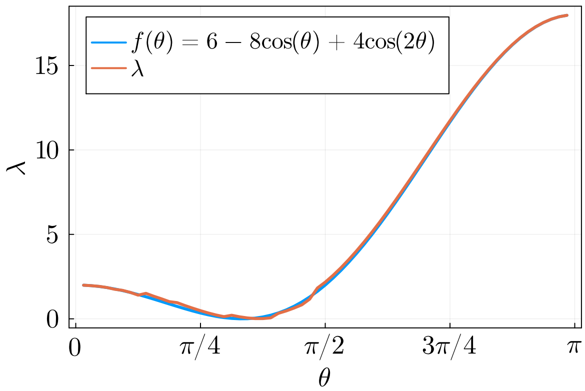
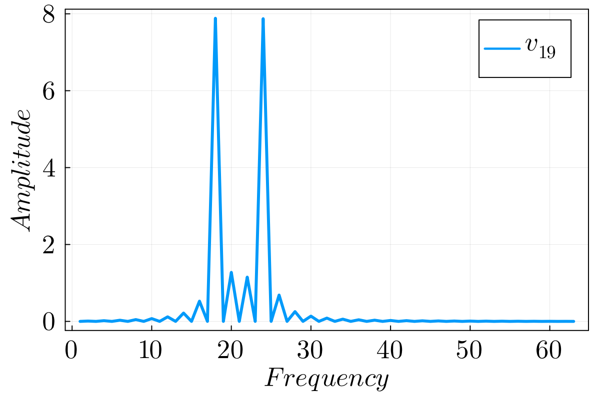


Figure 10: DST-I of eigenvectors 1-15. $\boldsymbol{f_1(\theta)}$**:** The 15x15 bi-Laplacian with symbol $f_1(\theta) = 6 - 8\cos(\theta) + 2\cos(2\theta)$. $\boldsymbol{f_2(\theta)}$**:** The 15x15 modified bi-Laplacian with symbol $f_2(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$

The problem with multiple peaks becomes even more pronounced for larger matrices. See for example the 63x63 modified bi-Laplacian in Figure 11, the amplitudes of the main and secondary peaks in v_{19} are just about the same.



(a) Resulting eigenvalue order together with its symbol



(b) DST-I of eigenvector 19

Figure 11: DST sort of the 63x63 modified bi-Laplacian

3.3 Imaginary sort

Imaginary sorting is done by combining the matrices

$$T_n(h) = \begin{bmatrix} 6 & -4 & 2 & 0 & 0 \\ -4 & 6 & -4 & 2 & 0 \\ 2 & -4 & 6 & -4 & 2 \\ 0 & 2 & -4 & 6 & -4 \\ 0 & 0 & 2 & -4 & 6 \end{bmatrix} + \delta\hat{\imath} \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}, \tag{17}$$

where the complex symbol $h(\theta, \delta)$ is a combination of the real symbol $f(\theta)$ and the imaginary symbol $g(\theta, \delta)\hat{\imath}$, namely,

$$h(\theta, \delta) = f(\theta) + \hat{\imath}g(\theta, \delta) = 6 - 8\cos(\theta) + 4\cos(2\theta) - 2\delta\cos(\theta)\hat{\imath}. \tag{18}$$

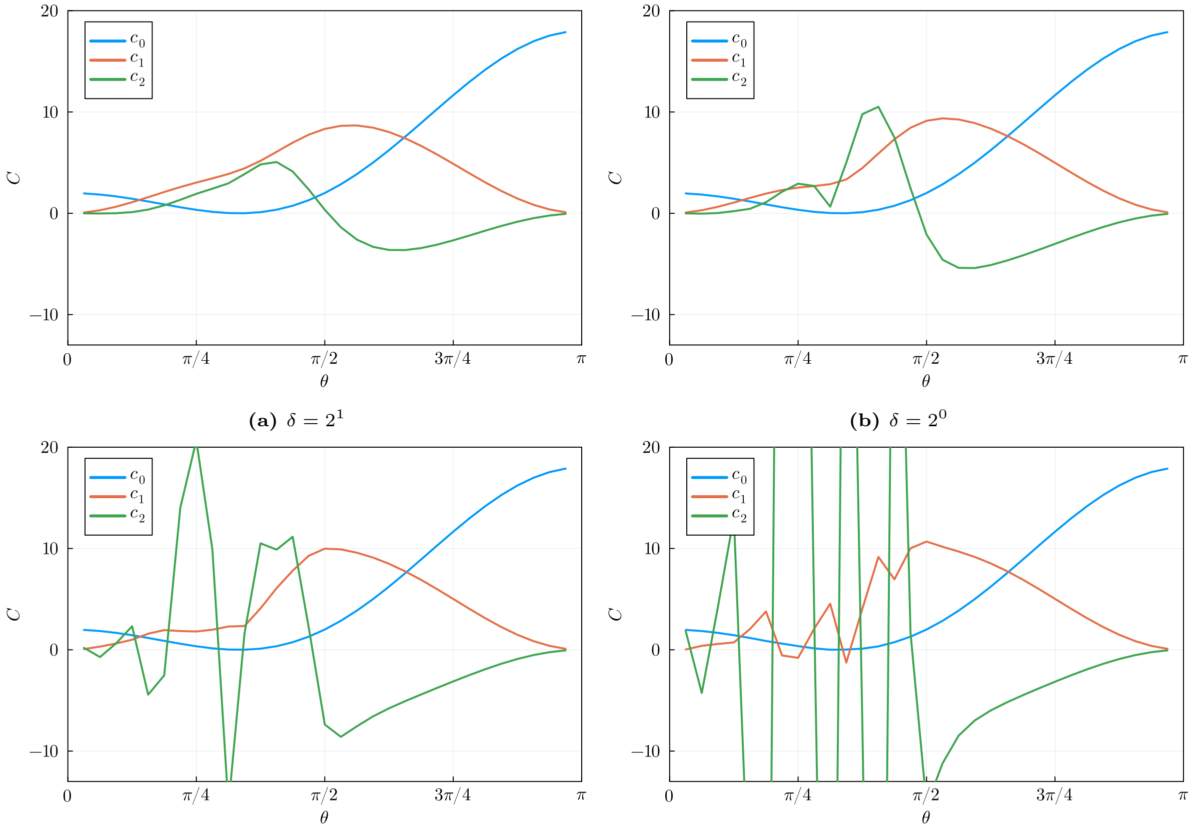
$$f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta). \tag{19}$$

$$g(\theta, \delta)\hat{\imath} = -2\delta\cos(\theta)\hat{\imath}. \tag{20}$$

The combination yields the complex matrix

$$\begin{bmatrix} 6 & -4-\delta\hat{\imath} & 2 & 0 & 0 \\ -4-\delta\hat{\imath} & 6 & -4-\delta\hat{\imath} & 2 & 0 \\ 2 & -4-\delta\hat{\imath} & 6 & -4-\delta\hat{\imath} & 2 \\ 0 & 2 & -4-\delta\hat{\imath} & 6 & -4-\delta\hat{\imath} \\ 0 & 0 & 2 & -4-\delta\hat{\imath} & 6 \end{bmatrix}. \tag{21}$$

Here as a 5×5 matrix but can be scaled diagonally to any size. This matrix with the corresponding generating symbol is used for all following calculations if nothing else is stated. In Figure 12 are four plots showing how the higher order symbols behave while sorting with the imaginary method depending on the size of δ. We want to decrease δ to make the real part of the eigenvalues correspond as closely as possible to the eigenvalues from the real matrix in equation (17).



(a) $\delta = 2^1$

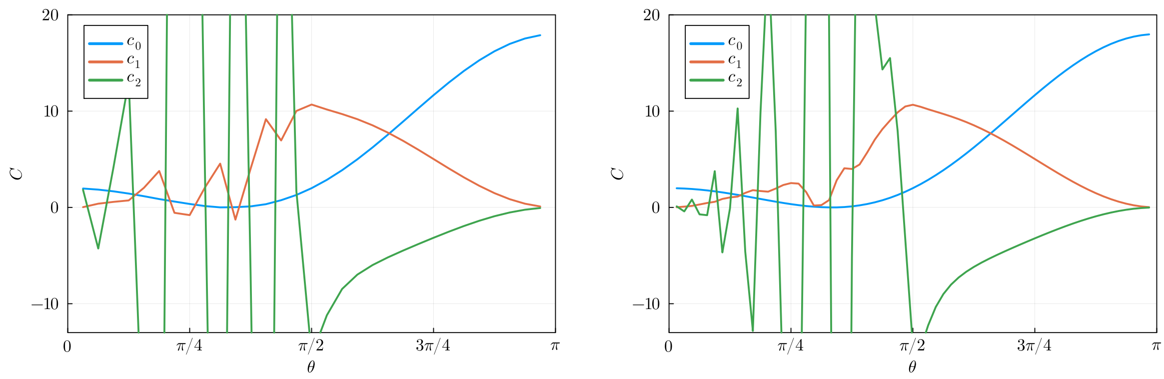
(b) $\delta = 2^0$

(c) $\delta = 2^{-1}$

(d) $\delta = 2^{-2}$

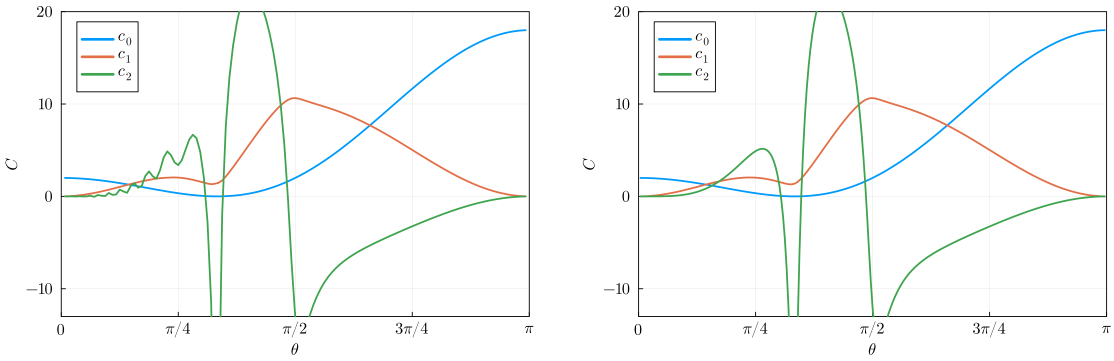
Figure 12: Plots showing how the sorting breaks down when decreasing δ. A good sorting results in smooth C-curves. The matrix size is 31×31

For the two plots Figure 12 (a) and (b) all three higher order symbols are smooth for most of their values, but when δ approaches smaller values two of the higher order symbols c_1 and c_2 start to break down getting sporadic. This indicates that the order of the real part of the eigenvalues may not be correct although more testing is needed to know exactly where this happens. To stabilize the higher order symbols an idea is to increase the matrix size n_0. In Figure 13 this is visualized with δ as 2^{-1}.



(a) $n_0 = 31$

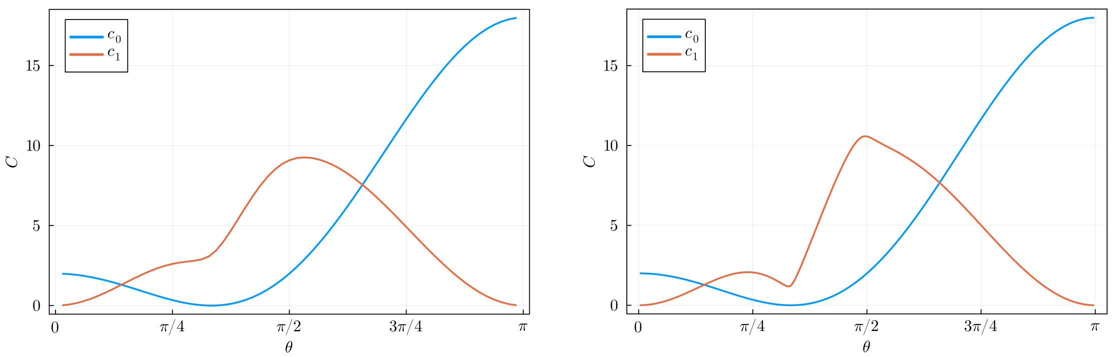
(b) $n_0 = 63$



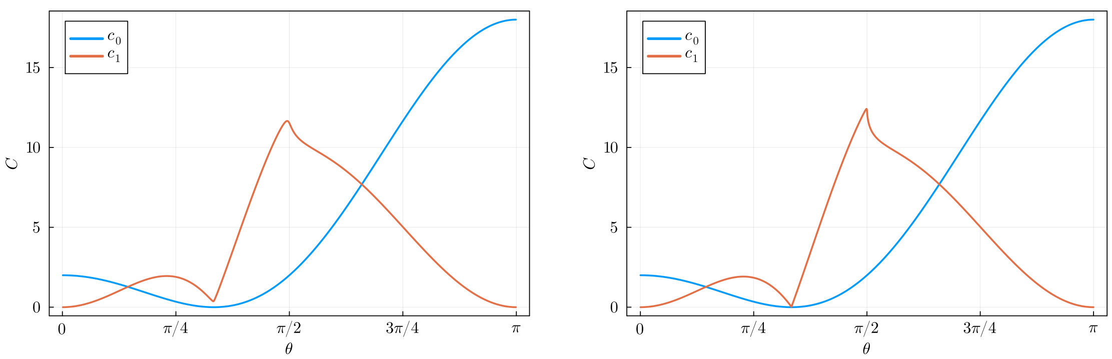
(c) $n_0 = 127$ **(d)** $n_0 = 255$

Figure 13: Plots showing how by increasing the matrix size, it is possible to use smaller $\delta(= 2^{-2})$ and still have higher order symbols that are not breaking

As seen in Figure 13, the results show that increasing the matrix size will allow for smaller δ values. Why this works is currently unclear to us but this possibly means we can increase the accuracy when using this sorting method for the MLM. First we try to decrease δ while increasing the matrix size such that the matrix is not larger than it has to be for each value of δ. In Figure 14 we track how the higher order symbols behave when $\delta \in [2^0, 2^{-2}, 2^{-4}, 2^{-8}]$ and the corresponding matrix sizes are $n_0 \in [2^6 - 1, 2^8 - 1, 2^{10} - 1, 2^{14} - 1] = [63, 255, 1023, 16383]$. There seems to be a stability condition that the matrix size must be doubled if δ is to be halved. When looking at how the c_1 curve changes between the observed four plots an important observed phenomena is how c_1 seems to converge towards three smooth curve segments as δ approaches zero, possibly pointing towards there existing an analytical solution for the perfect higher order symbol.



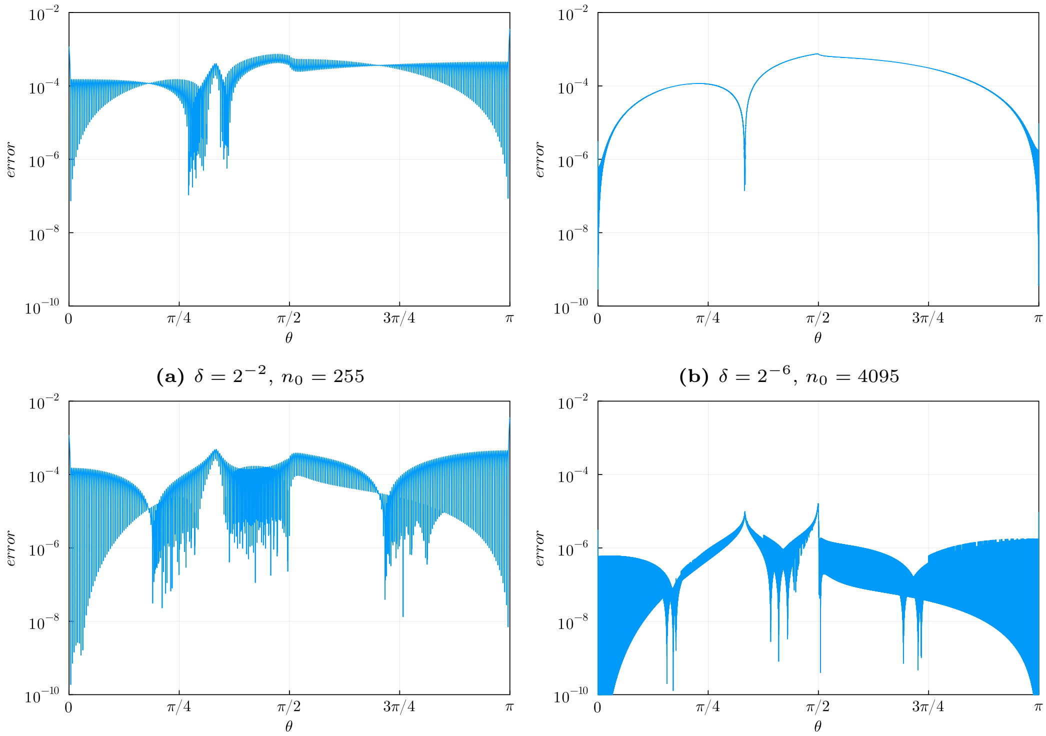
(a) $\delta = 2^0$, $n_0 = 63$ **(b)** $\delta = 2^{-2}$, $n_0 = 255$



(c) $\delta = 2^{-4}$, $n_0 = 1023$ **(d)** $\delta = 2^{-8}$, $n_0 = 16383$

Figure 14: Plots showing how the higher order symbols changes as δ decrease with matching matrix size

To investigate if the imaginary method gives a relevant result we monitor how the MLM error changes with changing δ. MLM generates eigenvalues for a matrix of size $10,000$ and $100,000$ and then is compared to JULIA's eigenvalue solver (`eigvals`). But how can the error be calculated for the non-monotone part of the curve? In Section 3.4 the eigenvalues from the real matrix are sorted in the same order as the imaginary sorting method. The result shows that the higher order symbol c_1 of the real matrix is not smooth, indicating that it likely is not the correct order. Three different absolute error measurements are shown in Figures 15, 16. In Figure 15 MLM is compared to the imaginary sorted eigenvalues of the real matrix. In Figure 15 **(a)** and **(b)** only uses one higher order symbol c_0 for the MLM while **(c)** and **(d)** uses two higher order symbols (c_0, c_1). Figure 16 displays the difference between the MLM generated by c_0 and c_1 and the actually eigenvalues with both ordered by rising values.



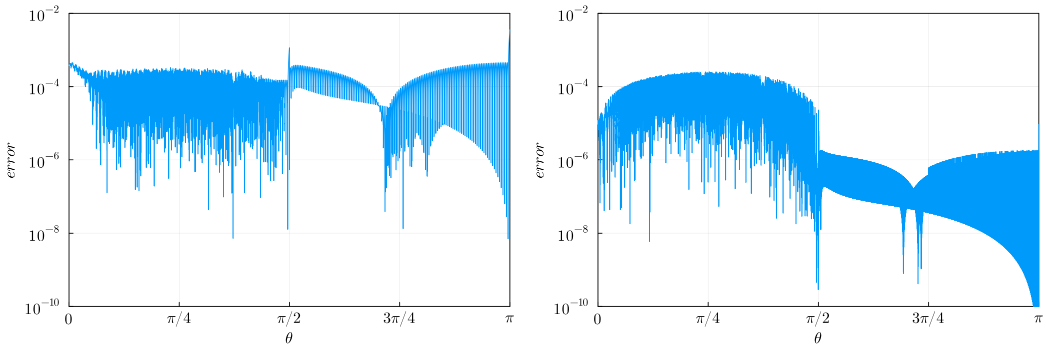
(a) $\delta = 2^{-2}$, $n_0 = 255$

(b) $\delta = 2^{-6}$, $n_0 = 4095$

(c) $\delta = 2^{-2}$, $n_0 = 255$

(d) $\delta = 2^{-6}$, $n_0 = 4095$

Figure 15: Error structure for MLM with α as 0 in **(a)** and **(b)** and α as 1 in **(c)** and **(d)**. This means using one higher order symbol c_0 for $\alpha = 0$ and two higher order symbols (c_0, c_1) for $\alpha = 1$. MLM is compared to the sorted eigenvalues from the real matrix according to the imaginary sorting method. Size of comparison is $n_f = 16383$



(a) $\delta = 2^{-2}$, $n_0 = 255$

(b) $\delta = 2^{-6}$, $n_0 = 4095$

Figure 16: Error structure for MLM with α as 1, which means using two higher order symbols c_0 and c_1. The MLM is sorted in rising order compared to the eigenvalues from the real matrix in rising order. Size of comparison is $n_f = 16383$

There are a couple of important and insightful behaviours displayed in Figures 15 and 16. One is that using two higher order symbols (c_0, c_1) instead of just c_0 gives better approximations. Not only in the monotone part which is shown in previous papers [9] and [1], but also likely for the non monotone part as seen by comparing Figure 15 **(b)** and **(d)**. Even if the sorting of the eigenvalues of the real matrix are not exactly correct, the curves clearly points towards two higher order symbols giving higher accuracy. Adding more levels of higher order symbols has been done in the previous papers but needs to be tested more for the whole curve.

The other significant result is that decreasing δ clearly results in decreased error for the whole curve if we assume the imaginary sorted comparison in Figure **??** is a valid comparison. In Figure 16 only the monotone part decreases showing the importance of how the error is measured.

3.3.1 DST sort of complex matrix

As already shown in Section 3.2 there are difficulties in sorting the eigenvalues through DST-sorting. In the non-monotone part more than one peak will occur for each eigenvector making it difficult for any observer to know where in the order the eigenvector belongs. This is with the assumption that the position of high amplitude peaks tells us something about the ordering position of the eigenvector and it's corresponding eigenvalue.

In Figure 17 the DST of an eigenvector taken from four versions of the complex matrix in equation 21 are shown. The variable differentiating the four matrices is δ and each eigenvector is from the non-monotone part. Changing δ is equivalent to changing the amplitude of the imaginary part of the complex matrix. What can be witnessed is how there is one pronounced peak for $\delta = 2^0$ compared to for example $\delta = 2^{-3}$ where there are multiple peaks with higher amplitude. Note that when using the DST only the real part of the eigenvectors are used.

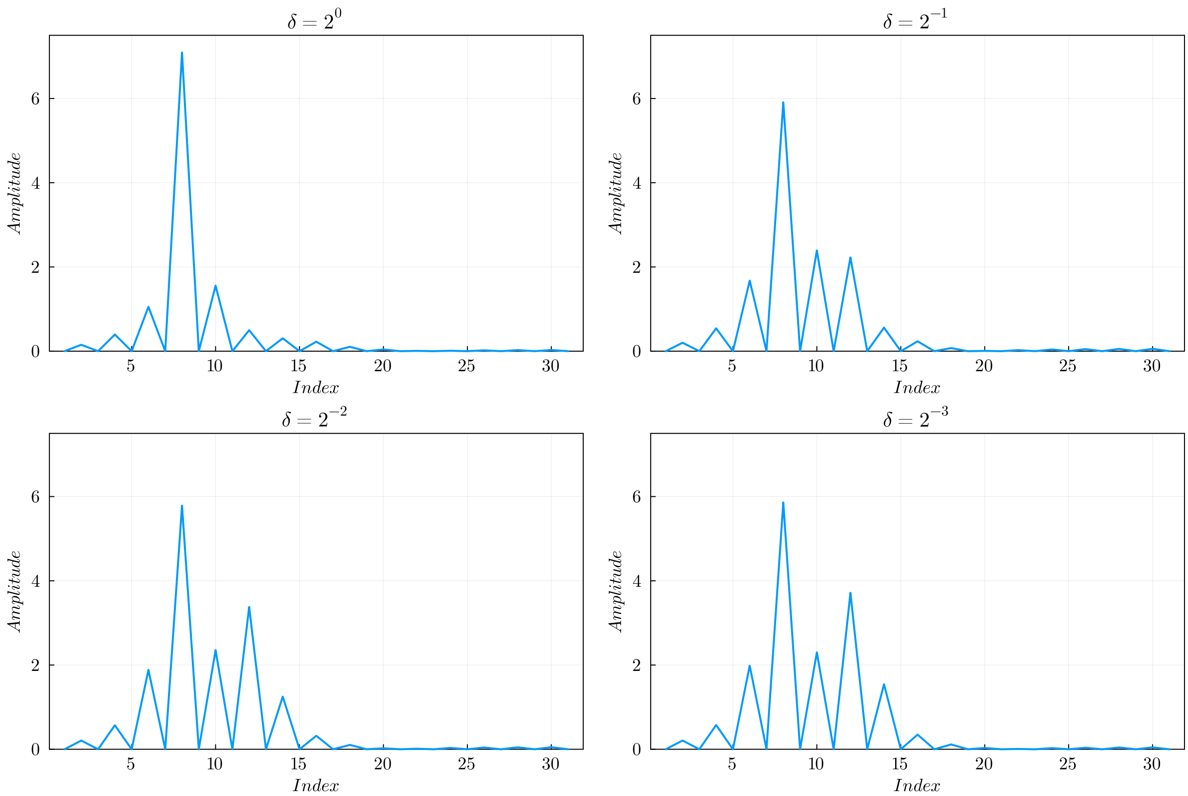


Figure 17: The DST of four different eigenvectors with different values on δ. Larger δs corresponding to a larger eigenvalue part for the complex matrix. The matrix size here is 32×32 and the eigenvector is chosen as the one corresponding with the eighth positioned eigenvalue sorted with the imaginary method

The DSTs in Figure 17 and the DST sorting method in Section 2.1.2 raises a question. If we have a complex matrix with a large enough imaginary part and thus pronounced peaks. Is it possible to sort the eigenvalues and eigenvectors by finding the index position of the largest peak for each corresponding DST? The answer appears to be yes. If the eigenvalues are sorted according to both the DST method and the imaginary method separately they end up in the exact same order. Figure 18 shows the ordered DSTs of the eigenvectors. Here early positions are represented by orange peaks to the left and later positions are represented by the blue peaks to the right. The ordering can be checked by comparing the indexing of the eigenvalues for both methods.

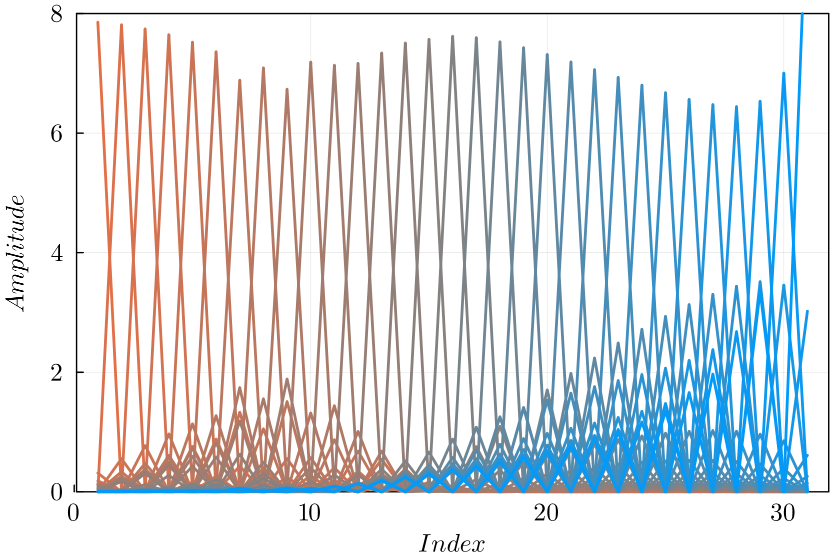
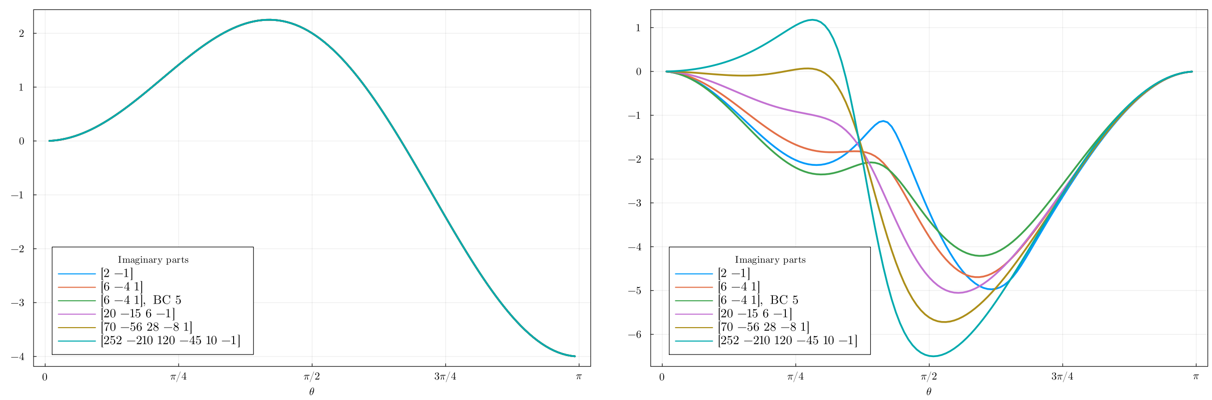


Figure 18: The sorted eigenvectors where orange/red DST-peaks are early positions and blue are later positions. Here δ is 2^0 and n_0 is 31. The sorting method used here is imaginary sort, but DST sort would generate the exact same result

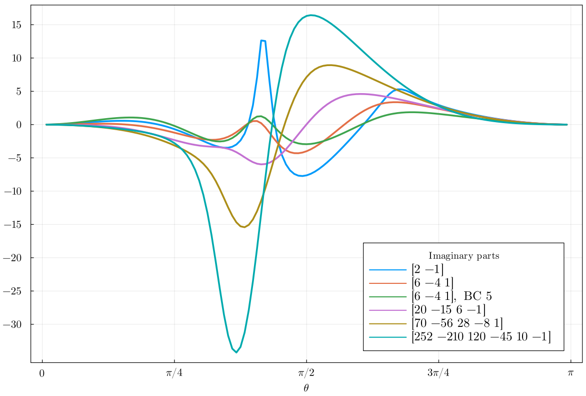
3.3.2 Structure of the imaginary part

Figures 19 and 21 show the resulting higher order symbols up to c_2 for the symbol $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$ and $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$, respectively. They are computed for different added imaginary symbols $g(\theta)$, with one of the cases including a boundary condition (different values in the corners of the matrix). The higher order symbols c_1 and c_2 for $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$ seems reasonable, including the case with a BC. In the case of the modified bi-Laplacian $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ it can be observed from c_0, c_1 and c_2 that there is some instability in the MLM when using a BC in the generated imaginary matrix, as their corresponding higher order symbols have grown very large. For the added imaginary symbol $g(\theta) = 2 - 2\cos(\theta)$, the computed c_2 seems oscillatory, which could suggest that a matrix size larger than $n_0 = 127$ might be needed.



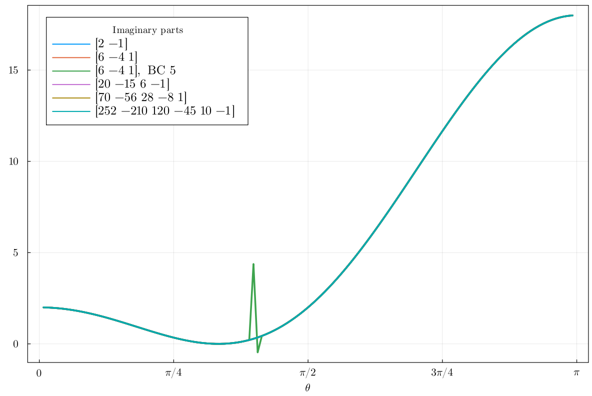
(a) c_0

(b) c_1

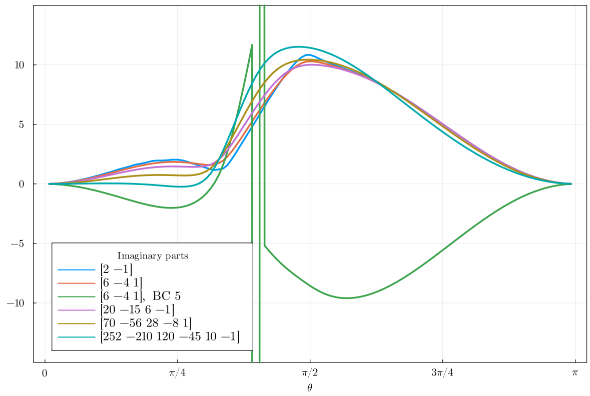


(c) c_2

Figure 19: Higher order symbols c_0, c_1 and c_2 calculated for the symbol $f(\theta) = 2\cos(\theta) - 2\cos(2\theta)$ for different imaginary symbols $g(\theta)$ with $\delta = 0.2$, $n_0 = 127$. Here "Imaginary parts" specifies the diagonals of the added imaginary matrix



(a) c_0



(b) c_1

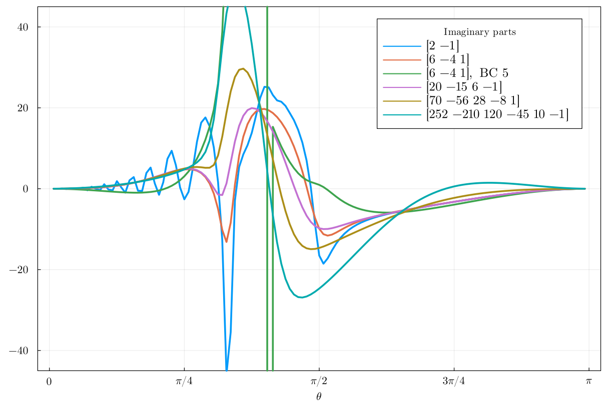
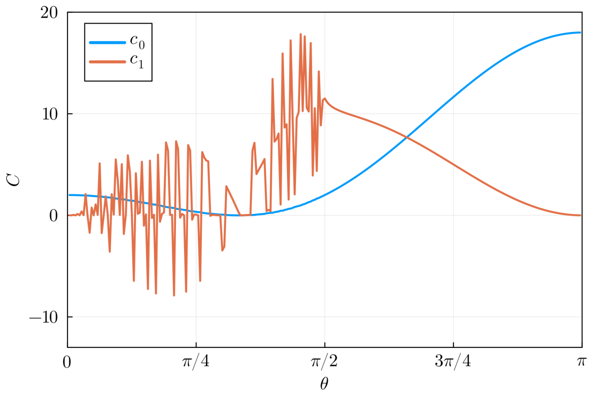


Figure 20: c_2

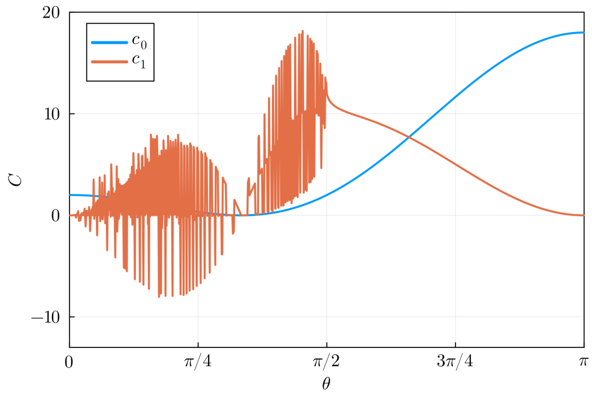
Figure 21: Higher order symbols c_0, c_1 and c_2 calculated for the symbol $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ for different imaginary symbols $g(\theta)$ with $\delta = 0.2$, $n_0 = 127$. Here "Imaginary parts" specifies the diagonals of the added imaginary matrix

3.4 Alternative imaginary sort

When pondering upon how to get rid of the errors from the imaginary sorting method one might propose to take the eigenvalues from the real matrix made of the modified bi-Laplacian generating symbol and sort it with the imaginary eigenvalue part from the complex matrix in (21).



(a) $\delta = 2^{-2}$, $n_0 = 255$



(b) $\delta = 2^{-4}$, $n_0 = 1023$

Figure 22: Eigenvalues of real matrix sorted in order of rising imaginary eigenvalues from complex matrix. Here with the left plot using $\delta = 2^{-2}$ and $n_0 = 255$ and the right plot using $\delta = 2^{-4}$ and $n_0 = 1023$

The higher order symbol c_1 is not smooth but it seems like there is some kind of pattern to the jumping. This may indicate that the ordering is wrong but that it still may not be too far out from the real index positions. Combining this method with some kind of DST analysis could bring a better understanding of the problem. Decreasing δ does not seem to have any positive impact on the sorting.

3.5 Similar sort

Figures 23, 24 and 25 illustrates how the similar sort method is applied to the modified bi-Laplacian matrix with generating symbol $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$. First the generating symbol $g(\theta) = 6 + 8\cos(\theta) + 4\cos(2\theta)$ of the similar matrix is multiplied with the imaginary unit and added to $f(\theta)$. Then the eigenvalues of the matrix generated by the resulting symbol $h(\theta) = f(\theta) + \hat{\imath}g(\theta)$ are calculated and sorted with a non-decreasing real part. Finally the imaginary eigenvalues from the correctly ordered part of the spectrum $[\pi/2, \pi]$ are reversed into the other half of the spectrum $[0, \pi/2]$.

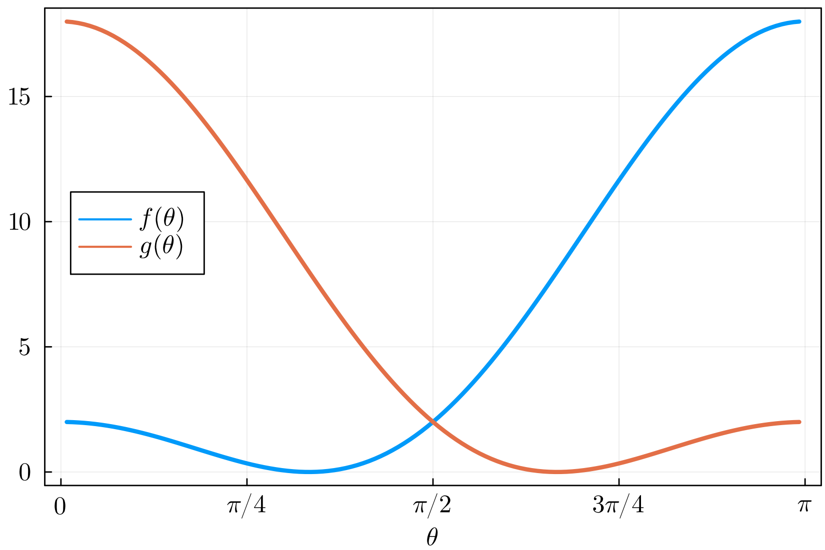


Figure 23: The generating symbol of the modified bi-Laplacian matrix $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$ and the generating symbol of the similar matrix $g(\theta) = 6 + 8\cos(\theta) + 4\cos(2\theta)$

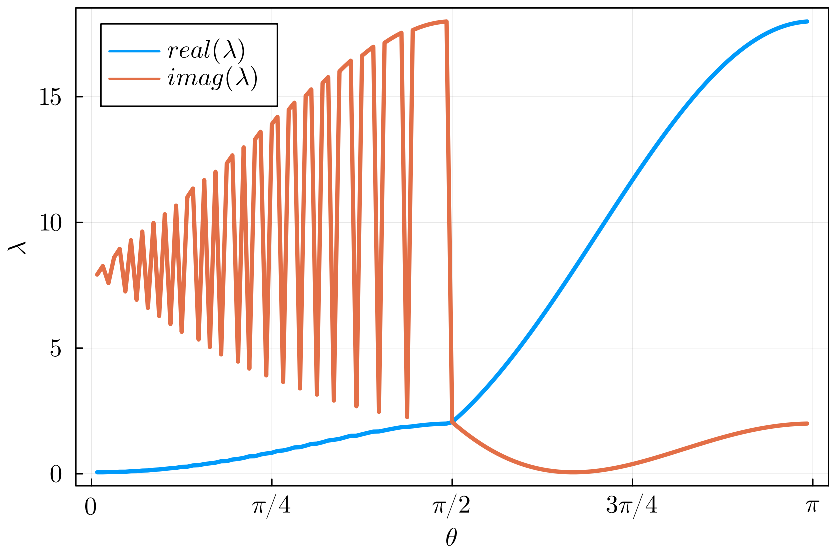


Figure 24: The real and imaginary parts of the eigenvalues of the matrix generated by symbol $h(\theta) = f(\theta) + \hat{\imath}g(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta) + \hat{\imath}(6 + 8\cos(\theta) + 4\cos(2\theta))$ sorted with non-decreasing real part. n=127

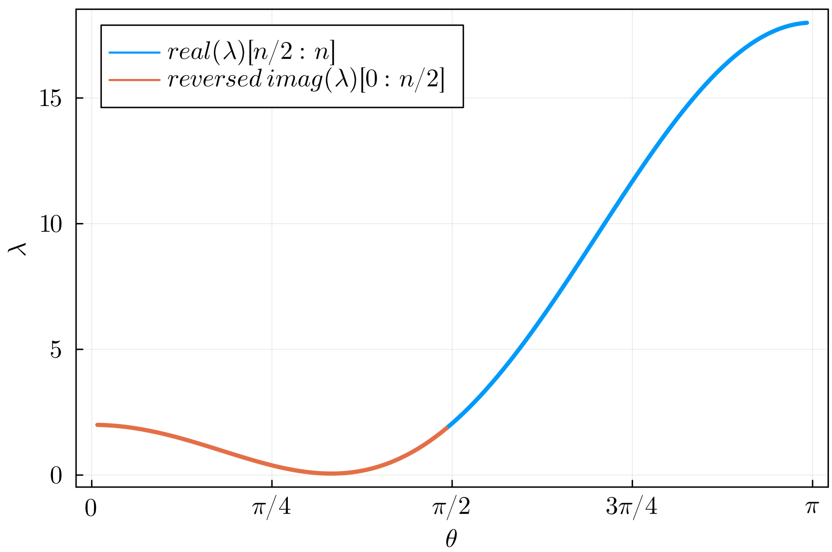
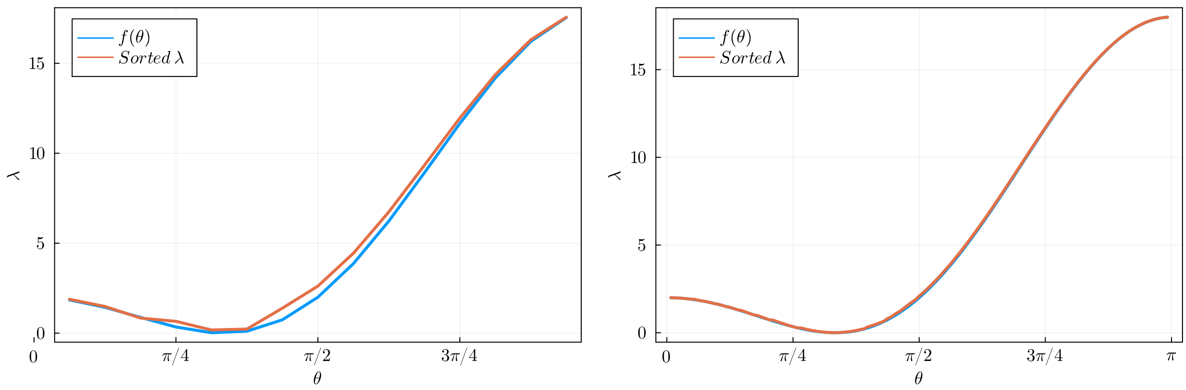


Figure 25: The real and imaginary parts of the eigenvalues of the matrix generated by the symbol $h(\theta) = f(\theta) + \hat{\imath}g(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta) + \hat{\imath}(6 + 8\cos(\theta) + 4\cos(2\theta))$. n=127. The reversed imaginary parts are placed on the first half of the spectrum $[0, \pi/2]$, while the real parts are kept on the second half $[\pi/2, \pi]$

The ordering obtained from the previous steps is then applied to the eigenvalues of the modified bi-Laplacian matrix with generating symbol $f(\theta)$ to finally receive the desired eigenvalues ordered by the similar sort method. In Figure 26 the resulting ordered eigenvalues are plotted together with the generating symbol.

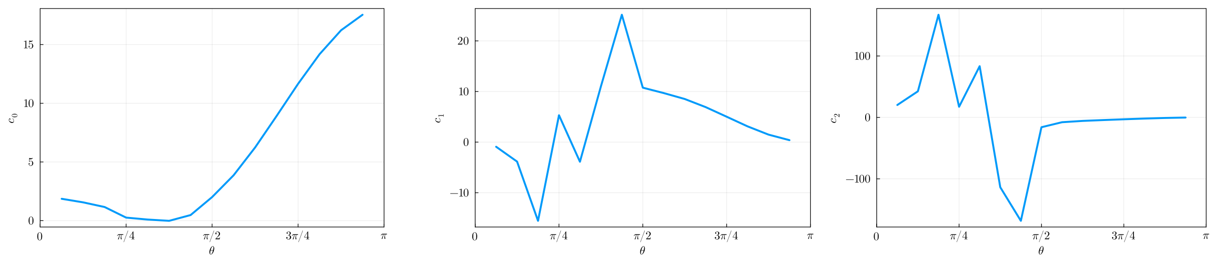


(a) n=15

(b) n=127

Figure 26: The sorted eigenvalues of the modified bi-Laplacian matrix generated by the symbol $f(\theta) = 6 - 8\cos(\theta) + 4\cos(2\theta)$

To further investigate the correctness of similar sort we construct the higher order symbols used in the MLM. In Figures 27 and 28 we can see that the higher order symbols c_1 and c_2 become erratic in the first half of the spectrum $[0, \pi/2]$ for both n=15 and n=127. This means the ordering produced by similar sort is incorrect. In an attempt to figure out why the method does not give the correct ordering for the eigenvalues of the modified bi-Laplacian we take a look at the errors introduced by the addition of the imaginary similar matrix. In Figure 29 the eigenvalue errors are compared to the minimum difference between the two size-wise closest eigenvalues of the modified bi-Laplacian. Since the median error is larger than this difference it is very likely that the ordering will be altered by the addition of the imaginary matrix. In other words, the correct ordering for the eigenvalues of modified bi-Laplacian differs from the correct ordering of the eigenvalues of the modified bi-Laplacian with the added imaginary similar matrix.

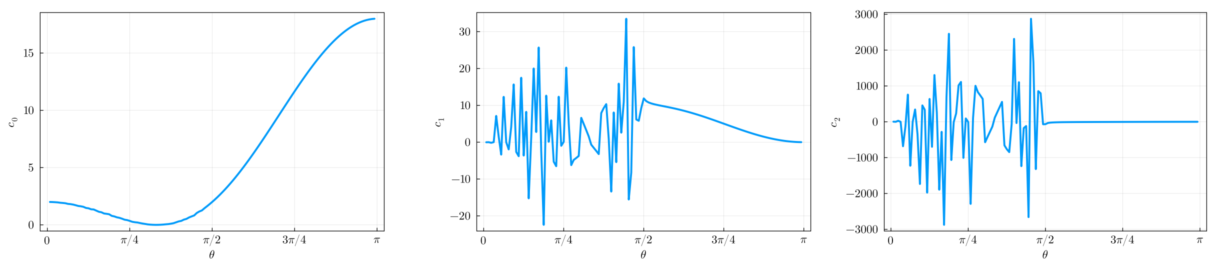


(a) c_0

(b) c_1

(c) c_2

Figure 27: Higher order symbols for the modified bi-Laplacian matrix, $n = 15$



(a) c_0

(b) c_1

(c) c_2

Figure 28: Higher order symbols for the modified bi-Laplacian matrix, $n = 127$

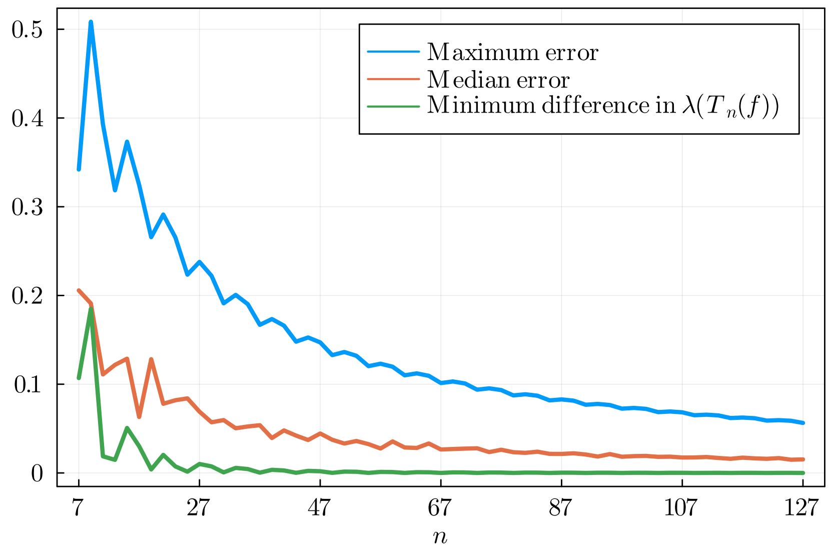


Figure 29: The blue and red lines shows the differences between the eigenvalues of the modified bi-Laplacian $[6, -4, 2]$ and the real part of the eigenvalues of the same matrix with the added imaginary similar matrix $[6, -4, 2] + \hat{\imath}[6, 4, 2]$. The green line shows the difference between the two size-wise closest eigenvalues of the modified bi-Laplacian $[6, -4, 2]$. The matrix sizes n are all odd numbers between 7 and 127

3.6 Hankel and symmetry tests

In this section we apply the Hankel and symmetry tests to the resulting orderings of eigenvalues and eigenvectors for symbol sort, DST sort, imaginary sort and similar sort. The parameter δ is fixed to 0.2. Then we compare the orderings for a specific matrix size to see where the orderings differ.

Table 1: Hankel and symmetry tests for the eigenvalues and eigenvectors of the modified bi-Laplacian sorted by all methods. The matrices are of sizes between 15x15 and 77x77 and $\delta = 0.2$ for imaginary sort. A passed test is marked as ✓and a failed test is marked as ✗.

	Symbol		DST		Imag		Similar	
n	Hankel	Sym	Hankel	Sym	Hankel	Sym	Hankel	Sym
15	✓	✓	✓	✓	✓	✓	✓	✓
17	✗	✗	✓	✓	✗	✗	✓	✓
19	✗	✗	✓	✓	✗	✗	✓	✓
21	✗	✗	✓	✓	✗	✗	✗	✗
23	✗	✗	✗	✗	✗	✗	✗	✗
25	✗	✗	✓	✓	✗	✗	✗	✗
27	✗	✗	✗	✗	✗	✗	✓	✓
29	✗	✗	✓	✓	✗	✗	✓	✓
31	✗	✗	✓	✓	✗	✗	✗	✗
51	✗	✗	✗	✗	✗	✗	✓	✓
53	✗	✗	✓	✓	✗	✗	✓	✓
63	✗	✗	✗	✗	✗	✗	✗	✗
77	✗	✗	✗	✗	✗	✗	✓	✓

The first observation from Table 1 is that the Hankel and symmetry tests essentially verifiy the same thing. To pass the symmetry test every other eigenvector has to be even and every other has to be odd. To pass the Hankel every other eigenvalue has to have the opposite sign compared to the eigenvalues of the Hankel version of the matrix. Therefore these two tests yield the same result when they are applied to the same ordering of eigenvalues and eigenvectors.
Similar sort passes the tests for n=15 but when looking at the higher order symbols presented in Figure 27 we can see that the ordering is not correct. This is because the Hankel and symmetry tests do not

completely verify the correctness of the ordering. A correct ordering will pass the tests but some incorrect orderings will also pass the test.

In Table 1 the imaginary sort fails the test for all matrix sizes larger than 15 but this result is a bit misleading. The ordering produced by imaginary sort aims to correctly order the eigenvalues which has been altered by the addition of the imaginary part, while the ordering produced by the other three sorting methods aims to sort the original unaltered eigenvalues. In the tests the orderings are directly applied to the bi-Laplacian which leads to imaginary sort failing. So it is essentially the alternative imaginary sort method which is tested here. It should also be noted that a fixed $\delta = 0.2$ is used for all imaginary sort tests.

Tables 2 and 3 show the orderings produced by the four methods for the modified bi-Laplacian matrix of size 15x15 and 21x21. The numbers are the size indices of the eigenvalues and their position in the row is where they should be placed. E.g., for the matrix of size 15x15 symbols sort would order the 7th largest eigenvalue first followed by the 6th largest followed by the 5th largest and so on. Since the modified bi-Laplacian is monotone and uniquely valued in the second half of the spectrum the ordering of the eigenvalues are there trivial. Hence, all sorting methods give the same ordering for the last eight eigenvalues for the 15x15 matrix and last 11 for the 21x21 matrix. Interestingly DST sort, imaginary sort and similar sort produces the same ordering for the 15x15 matrix. This ordering passes the Hankel and symmetry tests but is previously proven wrong by the higher order symbols of the MLM. For the 21x21 matrix every sorting method produces a unique ordering but only DST passes the Hankel and symmetry tests.

Table 2: Eigenvalue ordering of the matrix generated by $f(\theta) = 6 - 8\cos\theta + 4\cos 2\theta$, size 15x15.

Symbol	7, 6, 5, 3, 1, 2, 4, 8, 9, 10, 11, 12, 13, 14, 15
DST	7, 6, 4, 3, 1, 2, 5, 8, 9, 10, 11, 12, 13, 14, 15
Imaginary	7, 6, 4, 3, 1, 2, 5, 8, 9, 10, 11, 12, 13, 14, 15
Similar	7, 6, 4, 3, 1, 2, 5, 8, 9, 10, 11, 12, 13, 14, 15

Table 3: Eigenvalue ordering of the matrix generated by $f(\theta) = 6 - 8\cos\theta + 4\cos 2\theta$, size 21x21.

Symbol	10, 9, 8, 6, 5, 3, 1, 2, 4, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
DST	10, 9, 7, 6, 4, 3, 1, 2, 5, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
Imaginary	10, 7, 9, 6, 4, 3, 1, 5, 2, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21
Similar	10, 9, 8, 6, 4, 3, 1, 2, 5, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

4 Conclusions and Future Works

The objective of the project is to find methods to correctly order the eigenvalues of matrices with non-monotone generating symbols. Four different sorting methods are implemented and although none of the methods correctly sorts the eigenvalues for the whole spectrum without introducing perturbations the results could be useful in future projects. Even the current results from the imaginary sorting method may be useful for some applications if the order is not crucial, since the errors of the MLM approximations are very small. We summarize here the performance of each method together with ideas for future work.

Symbol sort, which is based on the GLT approximation, only works if the sampled generating symbol gives the exact eigenvalues. Therefore the method works for the matrix with generating symbol $f(\theta) = 6 - 8\cos\theta + 4\cos 2\theta$ but with fours in the top left and bottom right corners of the matrix. For other matrices where the sampled generating symbol does not give the exact eigenvalues the method produces incorrect orderings due to the errors in the GLT approximation.

DST sort applies the discrete sine transform to the eigenvectors corresponding to each eigenvalue and then orders them by the frequencies of maximum amplitude. For matrices with non-monotone generating symbols the DST shows multiple peaks of significant magnitude. The current version of the method considers only the largest peak when ordering the eigenvectors which leads to incorrect orderings. In future works one could analyse the frequency spectrum in other ways to possibly improve the DST sorting method.

Imaginary sort adds an imaginary matrix with a monotone generating symbol to the matrix whose eigenvalues are to be sorted. The eigenvalues of the combined matrix are then sorted in the monotone order of their imaginary part. This produces the correct ordering but with a perturbation on the eigenvalues because of the added imaginary matrix. To minimize this perturbation the imaginary matrix is multiplied with a scaling factor δ. Smaller δ leads to smaller perturbations of the eigenvalues but too small δ leads to incorrect orderings. From the results it is observed that bigger matrices allows smaller a smaller scaling factor. In future research one could try to find the exact mathematical relationship between δ and the matrix size n. With that relationship δ could always be chosen as small as possible without ruining the eigenvalue ordering. Since larger n allows smaller δ it may also be possible to decrease the scaling factor when refining the grid within the MLM. One may also look at how good an indicator unstable higher order symbols are in showing if the eigenvalues are incorrectly sorted or not. A third idea would be to look closer on the DST of the eigenvectors after using the alternative imaginary sorter in Section 3.4, and see if one can use them to slightly alter the order to a correct one. For theoretical mathematicians the most interesting point of interest is to further investigate the nature of higher order symbols with regards to δ approaching zero in the imaginary sorting method.

Similar sort exploits that the order of eigenvalues corresponding to the monotone part of the generating symbol is known. By adding an imaginary similar matrix with the same eigenvalues but in reverse order the eigenvalues corresponding to the non-monotone part of the original matrix can also be sorted. To be able to sort all the eigenvalues of a matrix its generating symbol must be monotone on at least half of the spectrum. If it is monotone on a smaller part of the spectrum the method can still be used to sort some eigenvalues. For example if the symbol is only monotone on the last fourth of the spectrum the method can be used to sort the eigenvalues on the first fourth of the spectrum. From the results it is seen that the method fails to obtain the correct ordering for the eigenvalues. This is likely due to the error introduced by adding the imaginary part. The eigenvalues of the summed matrix are skewed and may have another order by size compared to the eigenvalues of the original matrix.

5 Acknowledgements

We thank our supervisors Sven-Erik Ekström and David Meadon for both introducing us to the subject and guiding us through the entire project. Their feedback has been very important to us in all stages of the research process.

The computations were enabled by resources at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX).

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*, 78(3):867–893, aug 2017.

[2] G. Barbarino, S.-E. Ekström, S. Serra-Capizzano, and P. Vassalos. *Theoretical results for eigenvalues, singular values, and eigenvectors of (flipped) Toeplitz matrices and related computational proposals*, 2022.

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, jan 2017.

[4] A. Böttcher, J. Gasca, S. M. Grudsky, and A. V. Kozak. Eigenvalue clusters of large tetradiagonal toeplitz matrices. *Integral Equations and Operator Theory*, 93(8), feb 2021.

[5] S. Dhamija and P. Jain. Comparative analysis for discrete sine transform as a suitable method for noise estimation. *International Journal of Computer Science Issues*, 8, 09 2011.

[6] S.-E. Ekström. *Matrix-Less Methods for Computing Eigenvalues of Large Structured Matrices*, 2018.

[7] S.-E. Ekström, I. Furci, and S. Serra-Capizzano. Exact formulae and matrix-less eigensolvers for block banded symmetric toeplitz matrices. *BIT Numerical Mathematics*, 58:937–968, jul 2018.

[8] S.-E. Ekström, I. Furci, S. Serra-Capizzano, H. Speleers, C. Garoni, and C. Manni. Are the eigenvalues of the b-spline isogeometric analysis approximation of $\delta u = \lambda u$ known in almost closed form? *Numerical Linear Algebra*, 25, jul 2018.

[9] 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*, 80(3):819–848, mar 2018.

[10] S.-E Ekström, C. Garoni, and S. Serra-Capizzano. Are the eigenvalues of banded symmetric toeplitz matrices known in almost closed form? *Experimental Mathematics*, 27:478–487, 2018.

[11] S.-E. Ekström, S. Serra-Capizzano, and M. Bogoya. Fast toeplitz eigenvalue computations, joining interpolation-extrapolation matrix-less algorithms and simple-loop theory. *Numerical Algorithms*, 91:1653–1676, may 2022.

[12] 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*, 89:701–720, feb 2022.

[13] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[14] C. Garoni and S. Serra-Capizzano. *Generalized Locally Toeplitz Sequences: Theory and Applications*, 2017.

[15] D. Meadon. *A Matrix-less Method for Approximating the Eigenvectors of Toeplitz-like Matrices*, 2021.

[16] P. Schmidt and F. Spitzer. *The Toeplitz Matrices of an Arbitrary Laurent Polynomial*, 1960.