Relating the Arnoldi approximation to the best Chebyshev approximation of the characteristic polynomial

A study using a complex valued variant of the Remez algorithm

Thomas Renström

Bachelor's thesis
2022:K16

Lund University

Faculty of Science
Centre for Mathematical Sciences
Numerical Analysis

Centrum Scientiarum Mathematicarum

Abstract

The Remez algorithm is a beautiful algorithm that finds the best approximation to a function by finding points satisfying the alternation condition. In 1987 Ping Tang published his doctoral thesis detailing a version of the Remez algorithm modified for functions of complex values with complex coefficients [1]. He later followed the thesis with an article presenting an algorithm for finding the coefficients of a complex valued function [2]. The Arnoldi approximation is a well know iterative method for approximating the m eigenvalues of a matrix by a smaller set of n eigenvalues. The accuracy of the Arnoldi approximation is however dependent on an initial choice of a vector. The best choice of this initial vector leads to the so called ideal Arnoldi approximation. In 1994 Anne Greenbum and Lloyd Trefethen published an article [3] discussing the possibility of finding the ideal Arnoldi approximation using Chebyshev approximation. The purpose of this thesis is to experiment with Ping Tangs algorithm to approximate the characteristic equation of a matrix on a circle set and compare the results to that of the ideal Arnoldi approximation to see if the ideal Arnoldi is a best approximation in a Chebyshev sense.

Populärvetenskaplig sammanfattning

Beräkningar med stora matriser är svåra och långsamma att utföra, även för datorer. Med hjälp av en matrisens egenvärden kan man istället arbeta med dess karaktäristiska polynom, där egenvärdena är nollställen. För mycket stora matriser får detta polynom en hög grad och vi skulle vilja approximera det med ett av lägre grad för att förenkla beräkningarna ytterligare. En populär metod för att göra detta är den så kallade Arnoldimetoden som ger oss ett set av färre egenvärden som approximerar de ursprungliga. Den metoden kan dock ge olika resultat beroende på dess startparametrar. I detta arbete kommer vi jämföra det bästa möjliga resultatet ifrån Arnoldimetoden, med optimala startparametrar, med en annan metod presenterad av Ping Tang som approximerar komplexvärda funktioner med polynom. Vi kommer att applicera Tangs metod på den ursprungliga matrisens karaktäristiska polynom och se hur de olika metodernas resultat skiljer sig åt.

Acknowledgements

Foremost I would like to thank Claus Führer for his advice on my thesis as well as on other matters, and for the beeswax. I also want to thank my wife Sofie for her constant support during my studies.

Contents

Chapter 1

Introduction

1.1 Theme

The goal of this thesis is to anchor aspects of approximation theory to the realities of linear algebra. The idea originally stems from the bachelor thesis of Kateryna Ufymtseva [4] where she examined the Arnoldi iteration, an algorithm for finding the eigenvalues of a matrix. At the first step of the Arnoldi iteration an arbitrary vector is used. However, the accuracy of the end result dependens on that arbitrary vector. There is a best way to choose this vector and that result is known as the ideal Arnoldi approximation. Finding a vector that produces a ideal Arnoldi approximation would normally involve experimenting with different vectors in order to find the one yielding the best result, which can be very costly.

Since only two elements go into the Arnoldi approximation, the matrix A and the arbitrary vector b, the behavior of the algorithm is solely dependant on these two factors. In their article from 1994 Greenbaum and Trefethen hypothesize that even though the special properties of the arbitrary vector b sometimes impact the outcome, it is more common that the relevant features do not differ between different choices of initial vector. Rather it is the properties of the matrix A that decides the convergence of the iteration, therefor preconditioning of the matrix A should be more beneficial than preconditioning of b [3, p. 362]. They further mention the algorithm presented by Tang in [2] as one possibility to approximate the ideal Arnoldi approximation for normal matrices and say that for non-normal A, however, they know of no simple algorithm that is guaranteed to compute its ideal Arnoldi approximation [3, p. 365].

We would like to experiment with Tangs algorithm and see how close we actually get to the ideal Arnoldi approximation when approximating non-normal matrices on a naïve candidate for the domain.

Our plan is to generate a $m \times m$ matrix A and from this generate a suitable domain. On this domain we will then approximate the characteristic polynomial of A with a polynomial of degree $n \leq m$ using Tangs version of the Remez algorithm. We will then compare the roots of the polynomial to the eigenvalues of the ideal n-degree Arnoldi approximation to A.

This thesis has three main parts. The first part deals with the aspects of Arnoldi iteration in Chapter 2 and will help us familiarize ourselves with the

problem at hand.

The second part deals with the relevant aspects of approximation theory, more specifically the idea of best approximation in Chapter 3, that of Haar spaces in Chapter 4, the Remez algorithm in Chapter 5 and the algorithm proposed by Tang in Chapter 6.

As a third part we present the main idea of the thesis in Chapter 7 followed by our experiment and its results in discussion in Chapters 8 and 9 respectively.

1.2 Preliminaries

In this paper we will make use of two different spaces of polynomial functions. One of ordinary polynomials and one of monic polynomials. They are defined as follows:

Definition 1.2.1. *For a non-negative integer n, the space of polynomial functions $\mathcal{P}_n$ over the domain Ω is defined as*

$$\mathcal{P}_n(\Omega) := \left\{ p(\omega) = \sum_{k=0}^{n} c_k \omega^k, \omega \in \Omega, c_k \in \mathbb{R} \right\}.$$

Note that $\dim(\mathcal{P}_n) = n + 1$.

Definition 1.2.2. *For a non-negative integer n, the space of monic polynomial functions of degree n, $\mathcal{P}^n(\Omega)$, over the domain Ω is defined as*

$$\mathcal{P}^n(\Omega) := \left\{ p(\omega) = \sum_{k=0}^{n-1} c_k \omega^k + \omega^n, \omega \in \Omega, c_k \in \mathbb{R} \right\}.$$

Note that $\dim(\mathcal{P}^n) = n + 1$.

Chapter 2

The Arnoldi Method

Arnoldi approximation is an iterative method for finding the eigenvalues of a matrix $A \in \mathbb{R}^{m\times m}$. Arnoldi approximates the eigenvalues by way of finding the eigenvalues of the matrix $H \in \mathbb{R}^{n\times n}$, with $n < m$.

The way that Arnoldi iterations works is that for the $m \times m$ matrix A, we define the upper Hessenberg matrix H_n as

$$H_n = Q_n^{\mathrm{T}} A Q_n,$$

where Q_n is a matrix of n orthonormal vectors of length m as columns. Then we solve

$$AQ_n = Q_{n+1}\tilde{H}_n,$$

where $\tilde{H}_n$ is a $n+1 \times n$ with the first n rows making up H_n.

By letting $Q_0 = \begin{bmatrix} q_0 \end{bmatrix}$, an arbitrary $m \times 1$ matrix, we define subsequent iterations $Q_k = \begin{bmatrix} q_0 & Aq_0 & \cdots & A^k q_0 \end{bmatrix}$, we say that the columns of Q_n are the basis of a Krylov subspace.

From [3] we have that the Arnoldi approximation problem is finding the minimal polynomial $p_b^* \in \mathcal{P}^n$, where $\mathcal{P}^n$ is the space of monic polynomials of degree n, such that

$$p_b^* = \arg\min_{p\in\mathcal{P}^*} \|p(A)b\|_2 ,$$

and the ideal Arnoldi approximation, independent of the vector b is defined as

$$q = \arg\min_{p\in\mathcal{P}^*} \|p(A)\|_2 .$$

By the definition of the 2-norm we have that $\frac{\|Ax\|_2}{\|x\|_b} \leq \|A\|_2$ which gives us the inequality

$$\frac{\|q_b(A)b\|_2}{\|b\|_2} \leq \|q(A)\|_2 \leq \|q_b(A)\|_2$$

In the specific case where the matrix A is normal we have by the diagonalizability of A that for any polynomial p that $p(A) = p(Q^{-1}\Lambda(A)Q) = Q^{-1}p(\Lambda(A))Q$ for some orthogonal matrix Q. Since $\|Q\|_2 = 1$ we then have that

$$\|p(A)\|_2 = \|p(\Lambda(A))\|_2 = \max_{\lambda\in\Lambda(A)} |p(\lambda)|$$

Therefore, for a normal matrix A we have that

$$q = \arg\min_{p \in \mathcal{P}^*} \max_{\lambda \in \Lambda(A)} |p(\lambda)| ,$$

which could also be written as

$$q = \arg\min_{p \in \mathcal{P}^*} \|p\|_{\infty, \Lambda(A)} .$$

In order to create a new set to experiment with for non-normal matrices we need a different set. As stated in [3] this set is not known, but we will use the smallest circle centered at 0 that encompasses all eigenvalues.

By the maximum principle we know that out of the extreme points of $|f(z)|, z \in \Omega$ at least one occurs at the border of Ω, thus we choose the circle with radius $r = \max_{\lambda \in \Lambda(A)} |\lambda|$ as our naïve domain for our approximations.

Chapter 3

Best approximation

Basically, to approximate the function $f \in \mathcal{F}$ on a closed interval $[a, b]$ we take a set of simpler, more tangible functions, $\mathcal{S}$. In most cases this tangible set would be the set of polynomials of degree n, $\mathcal{S} = \mathcal{P}_n$. The function s in the set $\mathcal{S}$ that is closest to f over the interval is called the best. Generally, the best approximation is denoted as s^*.

A central part of approximation theory is the notion of best approximation. To determine which of the approximations in a certain set is the most accurate we make use of norms, and for any given norm we define the best approximation.

Definition 3.0.1. *[5, p. 61] Let $\mathcal{F}$ be a linear space with the norm $\|\cdot\|$ and let $\mathcal{S} \subset \mathcal{F}$ be a non-empty subset of $\mathcal{F}$. For $f \in \mathcal{F}$, an element $s^* \in \mathcal{S}$ is said to be a best approximation to f in $\mathcal{S}$ if*

$$\|s^* - f\| = \inf_{s \in \mathcal{S}} \|s - f\| .$$

Moreover,

$$\eta \equiv \eta(f, \mathcal{S}) = \inf_{s \in \mathcal{S}} \|s - f\|$$

is called the minimal distance between f and $\mathcal{S}$.

In this paper we are interested in finding monic polynomial approximations. A monic polynomial is a polynomial that has leading coefficient 1. We note:

Theorem 3.0.2. *If $q^* \in \mathcal{P}^n$ is the best monic approximation to $f \in \mathcal{F}$ on Ω then $p^*(\omega) = q^*(\omega) - \omega^n \in \mathcal{P}_{n-1}$ is the best approximation to $g(\omega) = f(\omega) - \omega^n$.*

Proof. Let $q^* \in \mathcal{P}^n(\Omega)$ be the best nth degree monic polynomial approximation to f. We have then that

$$\|q^* - f\| = \inf_{q \in \mathcal{P}^n(\Omega)} \|q - f\| .$$

A monic polynomial $q(\omega)$ in $\mathcal{P}^n(\Omega)$ can be written as $q(\omega) = \omega^n + p(\omega)$, where

$p(\omega) \in \mathcal{P}_{n-1}(\Omega)$ and thus

$$\begin{aligned} \|q^* - f\| &= \inf_{p \in \mathcal{P}_{n-1}(\Omega)} \|\omega^n + p - f\| \\ &= \inf_{p \in \mathcal{P}_{n-1}(\Omega)} \|p - (f - \omega^n)\| \\ &= \|p^* - (f - \omega^n)\| \end{aligned}$$

□

Thus the problem of finding a nth degree monic approximation can be solved by finding a $n-1$ non-monic approximation to a slightly modified function.

Throughout this paper we will sometimes make use of the general norm $\|\cdot\|$ but mostly we will be using the so called Chebyshev norm, also referred to as the max- or infinity norm, notated as $\|\cdot\|_\infty$.

Definition 3.0.3. *[5, p. 139] Let Ω be a compact domain and $\mathcal{C}(\Omega)$ a linear space of all continuous functions on Ω then the maximum norm is defined as*

$$\|u\|_{\infty,\Omega} := \max_{\omega \in \Omega} |u(\omega)| \quad \textit{for all } u \in \mathcal{C}(\Omega).$$

When the set Ω is explicitly stated, we will simply use the notation $\|\cdot\|_\infty$ instead.

Note that in some publications, [5] for example, the difference between an approximation and the function it approximates, $(s - f)(\omega)$, is referred to as "the error function". As it is not strictly speaking an error, we will refrain from using this term an instead strive to refer to it as "the difference function" where applicable.

Throughout the following sections we will be referring to the set of extremal points several times. We define this set here:

Definition 3.0.4. *Let Ω be a compact domain with $\|\cdot\|_\infty$ and $u \in \mathcal{C}(\Omega)$ a continuous functions on Ω then the set of extremal points is defined as*

$$E_u := \{\omega \in \Omega : |u(\omega)| = \|u\|_\infty\}.$$

Chapter 4

Haar spaces

A simple way to tell if a space contains a best approximation is through the definition of Haar spaces. In this chapter we will present some definitions and theorems regarding Haar spaces.

A Haar space is defined as follows.

Definition 4.0.1. *[5, p. 158] A linear space $\mathcal{S} \subset \mathcal{C}(\Omega)$ with $\dim(\mathcal{S}) = n < \infty$ is called a Haar space of dimension $n \in \mathbb{N}$ on Ω, if any $s \in \mathcal{S}\backslash\{0\}$ has at most $n-1$ zeros on Ω.*

Haar spaces have the following properties.

Theorem 4.0.2. *[5, p. 159] Let $\mathcal{S} \subset \mathcal{C}(\Omega)$ be a linear space of dimension $n \in \mathbb{N}$ and $X = \{x_1, \ldots, x_n\} \subset \Omega$ a set of n pairwise distinct points. Then the following statements are equivalent.*

1. *Any $s \in \mathcal{S}\backslash\{0\}$ has at most $n-1$ zeros on X.*
2. *For $s \in \mathcal{S}$, the implication*

$$s_X = 0 \Longrightarrow s \equiv 0 \text{ on } \Omega$$

   holds.

3. *For any $f_X \in \mathbb{R}^n$, there is a unique $s \in \mathcal{S}$ satisfying $s_X = f_X$.*
4. *For any basis $\mathcal{H} = (s_1, \ldots, s_n) \in \mathcal{S}^n$ of $\mathcal{S}$, the Vandermonde matrix*

$$V_{\mathcal{H},X} = \begin{bmatrix} s_1(x_1) & \cdots & s_1(x_n) \\ \vdots & \ddots & \vdots \\ s_n(x_1) & \cdots & s_n(x_n) \end{bmatrix} \in \mathbb{R}^{n \times n}$$

   is regular.

We also have the alternation theorem.

Theorem 4.0.3. *[5, p.165] Let $\mathcal{S} \subset \mathcal{C}(\Omega)$ be a n dimensional Haar space of $n \in \mathbb{N}$ on an interval $\Omega \subset \mathbb{R}$ and let $I \subset \Omega$ be a compact subset with at least $n+1$ elements. Then, for any $f \in \mathcal{C}(I)$ there exists a best approximation $s^* \in \mathcal{S}$ to f with respect to $\|\cdot\|_{\infty,I}$. This best approximation s^* is characterized by the existence of an alternation set of $n+1$ elements $X \in E_{s^*-f} \subset I$.*

We now define the space of complex trigonometric polynomials with complex coefficients.

Definition 4.0.4. *[5, p. 47] For any non-negative integer n we define the space of all complex trigonometric polynomials of degree n or less by*

$$\mathcal{T}_n^{\mathbb{C}} = \{T | T(x) = \sum_{k=0}^{n} c_k \, \mathrm{e}^{\mathrm{i}kx}, \quad c_k \in \mathbb{C}\}.$$

From Iske we have that the space of complex trigonometric polynomials is a Haar space.

Theorem 4.0.5. *[5, p. 162] For any non-negative integer n, $\mathcal{T}_n^{\mathbb{C}}$ is a Haar space of dimension $n+1$ over $\mathbb{C}$.*

Chapter 5

The Remez algorithm

In its classic form the Remez algorithm is an algorithm for finding the best approximation $s^* \in \mathcal{S}$ to f from a Haar space $\mathcal{S}$ on the interval $[a, b]$ such that

$$s^* = \arg\min_{s \in \mathcal{S}} \|f - s\|_{\infty,[a,b]}$$

In any of its iterations, the Remez algorithm computes for a ordered set $X = (x_1, \ldots, x_{n+1}) \in [a, b]^{n+1}$ the corresponding best approximation s^*_X to f with respect to $\|s - f\|_{\infty,X}$ so that

$$s^*_X = \arg\min_{s \in \mathcal{S}} \|f - s\|_{\infty,X}$$

We fix an ordered basis $\mathcal{H} = (s_1, \ldots, s_n)$ of the Haar space $\mathcal{S}$ so that s^*_X can be represented as

$$s^*_X = \sum_{k=1}^{n} \alpha^*_k s_k \in \mathcal{S}$$

with $\alpha^* = (\alpha^*_1, \ldots, \alpha^*_n)^{\mathrm{T}} \in \mathbb{R}^n$. According to Theorem 4.0.3 s^*_X satisfies

$$(s^*_X - f)(x_k) = (-1)^{k-1}\sigma \|s^*_X - f\|_{\infty,X} \quad \text{for } k = 1, \ldots, n+1$$

for some $\sigma \in \pm 1$. Therefore, η_X and α^* are the solution to the system

$$\begin{bmatrix} -1 & s_1(x_1) & \cdots & s_n(x_1) \\ 1 & s_1(x_2) & \cdots & s_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^{n+1} & s_1(x_{n+1}) & \cdots & s_n(x_{n+1}) \end{bmatrix} \begin{bmatrix} \eta_X \\ \alpha^*_1 \\ \vdots \\ \alpha^*_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n+1}) \end{bmatrix}$$

Since we have that

$$\|f - s^*\|_{\infty,\Omega} \geq \|f - s^*\|_{\infty,X}$$

for all sets $X \in [a,b]^{n+1}$, we know that if for some $X^* \in [a,b]^{n+1}$ we have equality, $s^*_X \equiv s^*$. If on the other hand we do not have equality that means that there exist an element $\hat{x} \in [a,b]$, $\hat{x} \notin X$ such that $|(f - s^*_X)(\hat{x})| \geq \|f - s^*\|_{\infty,X}$. In each iteration of the Remez algorithm, if such $\hat{x}$ exist, we replace a suitable element in X. Thus for each step $X \to X^*$ which implies that $s^*_X \to s^*$.

We present here the code structure for $\mathcal{S} = \mathcal{P}_{n-1}$.

Algorithm 1 The Remez algorithm

procedure REMEZALG($n, f(x), [a,b]$)
  $X \leftarrow \{x_0, \ldots, x_n\}$, $x_k \in [a,b]$
  repeat
    $Y \leftarrow \{f(x_0), \ldots, f(x_n)\}$
    $V \leftarrow \begin{bmatrix} x_0^{n-1} & \cdots & x_0^0 & (-1)^0 \\ \vdots & & \vdots & \vdots \\ x_n^{n-1} & \cdots & x_n^0 & (-1)^n \end{bmatrix}$
    $\{c_{n-1}, \ldots, c_0, \eta\} \leftarrow V^{-1}Y$
    $p(x) \leftarrow c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$
    $x_{new} \leftarrow \arg\max_x |f(x) - p(x)|$ for $x \in [a,b]$
    $X \leftarrow$ PLACENEWPOINT($X, Y, p(x), x_{new}$)
  until η optimal
  return $p(x)$

procedure PLACENEWPOINT($X, Y, p(x), x_{new}$)
  if $x_{new} \leq x_0$ **then**
    if $(f(x_0) - p(x_0)) \cdot (f(x_{new}) - p(x_{new})) \geq 0$ **then**
      $x_0 \leftarrow x_{new}$
    else
      $x_{n+1} \leftarrow x_{new}$
  else if $x_{new} \geq x_{n+1}$ **then**
    if $(f(x_{n+1}) - p(x_{n+1})) \cdot (f(x_{new}) - p(x_{new})) \geq 0$ **then**
      $x_{n+1} \leftarrow x_{new}$
    else
      $x_0 \leftarrow x_{new}$
  else
    for k in $[1,n]$ **do**
      if $x_k \geq x_{new}$ **then**
        if $(f(x_k) - p(x_k)) \cdot (f(x_{new}) - p(x_{new})) \geq 0$ **then**
          $x_k \leftarrow x_{new}$
        else
          $x_{k-1} \leftarrow x_{new}$
        break
  sort X
  return X

Chapter 6

Tangs method

In 1988 Ping Tang defined an algorithm based on the Remez algorithm but for the purpose of approximating complex valued functions with complex coefficients. In the article he assumes a periodic function on the domain $[0, 1]$. Since the algorithm works also for non-polynomial basis functions, he formulates the problem for approximations on a general basis [2, p. 722]. He defines the problem as finding the n real parameters $\lambda_1^*, \ldots, \lambda_n^*$ such that

$$\max_{t\in[0,1]} \left| f(t) - \sum_{k=1}^{n} \lambda_k^* \varphi_k(t) \right| \leq \max_{t\in[0,1]} \left| f(t) - \sum_{k=1}^{n} \lambda_k \varphi_k(t) \right| \tag{6.1}$$

for all $\lambda = [\lambda_1, \ldots, \lambda_n]^{\mathrm{T}} \in \mathbb{R}^n$.

We define the function $p(t) = \sum_{k=1}^{n} \lambda_k \varphi_k(t)$ and rewrite the problem as finding $p^*(t) \in \mathcal{P}_n$ such that

$$\max_{t\in[0,1]} |f(t) - p^*(t)| \leq \max_{t\in[0,1]} |f(t) - p(t)| \tag{6.2}$$

for all $p \in \mathcal{P}_n$.

Tang notes that while the basis used is linearly independent on $[0, 1]$ is only assumed to satisfy uniqueness, since it does not satisfy the Haar condition. With this reformulated problem we take a look at the Kolmogorov theorem as presented by Tang.

Theorem 6.0.1. *[1, p. 6] An approximant $p^* \in \mathcal{P}_n$ is a best approximation of f from $\mathcal{P}_n$ if and only if*

$$\max_{t_k\in E_{f-p^*}} \mathrm{Re}(\overline{(f-p^*)(t_k)}p(t_k)) \geq 0 \quad \textit{for all } p \in \mathcal{P}_n. \tag{6.3}$$

Proof. We assume that there exists a positive $\epsilon \in \mathbb{R}$ and $p \in \mathcal{P}_n$ such that

$$\max_{t_k\in E_{f-p^*}} \mathrm{Re}(\overline{(f-p^*)(t_k)}p(t_k)) = -\epsilon.$$

We shall see that there then exists a $\beta > 0$ that makes $q^* = p^* + \beta p$ that approximates f better than p^*. We separate $[0, 1]$ into two subintervals I_1 and

I_2 where

$$I_1 := \left\{ t \in [0,1] \,;\, \mathrm{Re}(\overline{(f-p^*)(t)}p(t)) < -\frac{\epsilon}{2} \right\}$$

and

$$I_2 := [0,1] \setminus I_1.$$

We have then for any $t \in I_1$ that

$$\begin{aligned} |(f-q^*)(t)|^2 &= |(f-p^*)(t) + \beta p(t)|^2 \\ &= |(f-p^*)(t)|^2 + \beta^2 |p(t)|^2 + 2\beta\,\mathrm{Re}(\overline{(f-p^*)(t)}p(t)) \\ &\leq |(f-p^*)(t)|^2 + \beta^2 |p(t)|^2 + -2\beta\frac{\epsilon}{2} \\ &\leq \|f-p^*\|_\infty^2 + \beta\left(\beta|p(t)|^2 + -\epsilon\right) \end{aligned}$$

Thus $|(f-q^*)(t)| < \|f-p^*\|_\infty$ for all $t \in I_1$ when $0 < \beta < \frac{\epsilon}{\|p\|_\infty^2} \leq \frac{\epsilon}{|p(t)|^2}$.

Since I_1 contains all the extremal points of $|(f-p^*)(t)|$ we know that $\max_{t\in I_2} |(f-p^*)(t)| < \|f-p^*\|_\infty$. We define

$$\delta := \|f-p^*\|_\infty - \max_{t\in I_2} |(f-p^*)(t)|$$

then for $t \in I_2$

$$\begin{aligned} |(f-q^*)(t)| &= |(f-p^*)(t) + \beta p(t)| \\ &\leq |(f-p^*)(t)| + \beta\,|p(t)| \\ &\leq \|f-p^*\| - \delta + \beta\,|p(t)| \end{aligned}$$

Thus $|(f-q^*)(t)| < \|f-p^*\|_\infty$ for all $t \in I_2$ when $\beta < \frac{\delta}{\|p\|_\infty} \leq \frac{\delta}{|p(t)|}$.

From this we have that $|(f-q^*)(t)| < \|f-p^*\|_\infty$ for all $t \in [0,1]$ for $0 < \beta < \min\left\{ \frac{\epsilon}{\|p\|_\infty^2}, \frac{\delta}{\|p\|_\infty} \right\}$ and thus q^* is a better approximant to f. □

Following are three remarks that are implications of Theorem 6.0.1. They are not explicitly mentioned by Tang in [2] but in the interest of clarity we shall state them here.

Remark 6.0.2. *Inequality 6.3 in Theorem 6.0.1 can be rewritten as*

$$\max_{t_k \in E_{f-p^*}} \mathrm{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k}(f-p)(t_k)) \geq \|f-p^*\|_\infty \quad \textit{for all } p \in \mathcal{P}_n. \tag{6.4}$$

for $\alpha_k = \mathrm{Arg}((f-p^*)(t_k))$.

Proof. We have from Theorem 6.0.1 that

$$\max_{t_k \in E_{f-p^*}} \mathrm{Re}(\overline{(f-p^*)(t_k)}p(t_k)) \geq 0 \quad \text{for all } p \in \mathcal{P}_n.$$

Since $\mathcal{P}_n$ is a linear space we have that all elements therein can be written as a combination of p^* and another element in $\mathcal{P}_n$. And thus

$$\max_{t_k \in E_{f-p^*}} \mathrm{Re}(\overline{(f-p^*)(t_k)}(p^*-p)(t_k)) \geq 0 \quad \text{for all } p \in \mathcal{P}_n.$$

We further rewrite the left hand side

$$\begin{aligned}
\operatorname{Re}(\overline{(f-p^*)(t_k)}(p^*-p)(t_k)) &= \operatorname{Re}(\overline{(f-p^*)(t_k)}(p^*-f+f-p)(t_k)) \\
&= \operatorname{Re}(\overline{(f-p^*)(t_k)}(f-p)(t_k)) + \operatorname{Re}(\overline{(f-p^*)(t_k)}(p^*-f)(t_k)) \\
&= \operatorname{Re}(\overline{(f-p^*)(t_k)}(f-p)(t_k)) - \operatorname{Re}(\overline{(f-p^*)(t_k)}(f-p^*)(t_k)) \\
&= |(f-p^*)(t_k)| \operatorname{Re}(\mathrm{e}^{\mathrm{i}\operatorname{Arg}(\overline{(f-p^*)(t_k)})}(f-p)(t_k)) - |(f-p^*)(t_k)|^2 \\
&= |(f-p^*)(t_k)| \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}(f-p)(t_k)) - |(f-p^*)(t_k)|^2
\end{aligned}$$

Getting thus that

$$\max_{t_k \in E_{f-p^*}} |(f-p^*)(t_k)| \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}(f-p)(t_k)) \geq |(f-p^*)(t_k)|^2 \quad \text{for all } p \in \mathcal{P}_n,$$

or equivalently

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}(f-p)(t_k)) \geq \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n.$$

□

Remark 6.0.3. *Inequality 6.4 in Remark 6.0.2 can be further rewritten as*

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{\mathrm{i}\alpha_k} f(t_k)) \geq \operatorname{Re}(\mathrm{e}^{\mathrm{i}\alpha_k} p(t_k)) + \|f-p^*\|_\infty \quad \textit{for all } p \in \mathcal{P}_n \tag{6.5}$$

for $\alpha_k = \operatorname{Arg}((f-p^*)(t_k))$.

Proof. We have from Remark 6.0.2 that

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k}(f-p)(t_k)) \geq \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n.$$

We rewrite the left hand side

$$\operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k}(f-p)(t_k)) = \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} f(t_k)) - \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} p(t_k)).$$

Thus we get

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} f(t_k)) - \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} p(t_k)) \geq \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n.$$

or equivalently

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} f(t_k)) \geq \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\alpha_k} p(t_k)) + \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n.$$

□

Remark 6.0.4. *Inequality 6.4 in Remark 6.0.2 can also be rewritten as*

$$\max_{t_k \in E_{f-p^*}} |(f-p)(t_k)| \cdot \cos(\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))) \geq \|f-p^*\|_\infty \quad \textit{for all } p \in \mathcal{P}_n. \tag{6.6}$$

Furthermore, we have that

$$\cos(\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))) \geq 0,$$

which implies

$$|\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))| \geq \frac{\pi}{2}.$$

Proof. We have from Remark 6.0.2 that

$$\max_{t_k \in E_{f-p^*}} \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}(f-p)(t_k)) \geq \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n.$$

We rewrite the left hand side

$$\begin{aligned}
\operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}(f-p)(t_k)) &= |(f-p)(t_k)| \operatorname{Re}(\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p^*)(t_k))}\,\mathrm{e}^{-\mathrm{i}\operatorname{Arg}((f-p)(t_k))}) \\
&= |(f-p)(t_k)| \operatorname{Re}(\mathrm{e}^{\mathrm{i}(\operatorname{Arg}((f-p)(t_k))-\operatorname{Arg}((f-p^*)(t_k)))}) \\
&= |(f-p)(t_k)| \cos(\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))).
\end{aligned}$$

Thus we arrive at

$$\max_{t_k \in E_{f-p^*}} |(f-p)(t_k)| \cos(\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))) \geq \|f-p^*\|_\infty \quad \text{for all } p \in \mathcal{P}_n. \tag{6.7}$$

Furthermore, as $\|f-p^*\|_\infty \geq 0$ we have that for t_k fulfilling Equation 6.7 that

$$\cos(\operatorname{Arg}((f-p)(t_k)) - \operatorname{Arg}((f-p^*)(t_k))) \geq 0.$$

□

This tells us that in Tangs version of Remez algorithm we have that the argument of two neighboring points cannot be within $\frac{\pi}{2}$ of each other. This implies the characterization theorem of the original Remez algorithm D.0.5, where the error alternates between points on the positive and negative part of the real axis and thus the argument of the error has a difference of π.

Like in Chapter 5 our problem is to find in each iteration, for a ordered set $X = \{t_0, \ldots, t_n\} \in [0, 2\pi]^{n+1}$ the corresponding best approximation p_X^* to f with respect to $\|p-f\|_{\infty,X}$ so that

$$p_X^* = \arg\min_{p \in \mathcal{P}_n}$$

with the characterization clarified in Remark 6.0.3

$$\begin{bmatrix} 1 & \operatorname{Re}(\varphi_0(t_0)\cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \operatorname{Re}(\varphi_{n-1}(t_0)\cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) \\ 1 & \operatorname{Re}(\varphi_0(t_1)\cdot \mathrm{e}^{-\mathrm{i}\alpha_1}) & \cdots & \operatorname{Re}(\varphi_{n-1}(t_1)\cdot \mathrm{e}^{-\mathrm{i}\alpha_1}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \operatorname{Re}(\varphi_0(t_n)\cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) & \cdots & \operatorname{Re}(\varphi_{n-1}(t_n)\cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \end{bmatrix} \begin{bmatrix} \|f-p\|_\infty \\ \lambda_0^* \\ \vdots \\ \lambda_{2n+1}^* \end{bmatrix} \leq \begin{bmatrix} \operatorname{Re}(f(t_0)\cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) \\ \operatorname{Re}(f(t_1)\cdot \mathrm{e}^{-\mathrm{i}\alpha_1}) \\ \vdots \\ \operatorname{Re}(f(t_n)\cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \end{bmatrix}.$$

We call this relation $Ax \leq c$. We further define b as the unit vector of the same size as x,

$$b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and our problem becomes to maximize $\|f-p\|_\infty = b^{\mathrm{T}}x$ subject to

$$Ax \leq c.$$

In contrast to the classic Remez algorithm, Tangs method instead involves solving the problems dual which is to minimize $h = c^{\mathrm{T}}y$ subject to

$$A^{\mathrm{T}}y = b$$

with the constraint that $y \geq 0$.

We present Tangs algorithm with the following pseudocode:

Algorithm 2 Tangs implementation of the Remez algorithm

1: **procedure** RemezAlg($n, f(t), \{\varphi_0(t), \ldots, \varphi_{n-1}(t)\}, [a, b]$)
2: $p(t), T, A, h \leftarrow$ FirstStep($n, f(t), \{\varphi_0(t), \ldots, \varphi_{n-1}(t)\}, [a, b]$)
3: **while** h not optimal **do**
4: $p(t), T, A, h \leftarrow$ RemezStep($p(t), f(t), \{\varphi_0(t), \ldots, \varphi_{n-1}(t)\}, T, A, [a, b]$)
5: **return** $p(t)$

Algorithm 3 Tangs implementation of the Remez algorithm, continued

procedure FIRSTSTEP($n, f(t), \{\varphi_0(t), \ldots, \varphi_{n-1}(t)\}, [a, b]$)
  repeat
    $T \leftarrow \{t_0, \ldots, t_n\}$, $t_k \in [a, b]$
    $A \leftarrow \{\alpha_0, \ldots, \alpha_n\}$, $\alpha_k \in [0, 2\pi]$
    $$V \leftarrow \begin{bmatrix} 1 & \cdots & 1 \\ \mathrm{Re}(\varphi_0(t_0) \cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_0(t_n) \cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \\ \vdots & & \vdots \\ \mathrm{Re}(\varphi_{n-1}(t_0) \cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_{n-1}(t_n) \cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \end{bmatrix}$$
    $Y \leftarrow \{\mathrm{Re}(f(t_0) * \mathrm{e}^{-\mathrm{i}\alpha_0}), \ldots, \mathrm{Re}(f(t_n) * \mathrm{e}^{-\mathrm{i}\alpha_n})\}$
  until V is non-singular **and** LINPROG(V, Y) solvable
  $h \leftarrow$ LINPROG(V,Y)
  $\{\eta, \lambda_0, \ldots, \lambda_{n-1}\} \leftarrow (V^{\mathrm{T}})^{-1} Y$
  $p(t) \leftarrow \lambda_0 \varphi_0(t) + \ldots + \lambda_{n-1} \varphi_{n-1}(t)$
  return $p(t), T, A, h$

procedure LINPROG(V, Y)
  $h = \min_r Y^{\mathrm{T}} r$ **subject to** $Vr = [1, 0, \ldots, 0]$ **and** $0 \leq r_k \leq 1$
  return h

procedure REMEZSTEP($p(t), f(t), \{\varphi_0(t), \ldots, \varphi_{n-1}(t)\}, T, A, [a, b]$)
  $t_{new} \leftarrow \arg\max_t |(f - p)(t)|$ for $t \in [a, b]$
  $\alpha_{new} \leftarrow \mathrm{Arg}((f - p)(t_{new}))$
  for k in $[0, n]$ **do**
    $t_{temp}, \alpha_{temp} \leftarrow t_k, \alpha_k$
    $t_k, \alpha_k \leftarrow t_{new}, \alpha_{new}$
    $$V \leftarrow \begin{bmatrix} 1 & \cdots & 1 \\ \mathrm{Re}(\varphi_0(t_0) \cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_0(t_n) \cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \\ \vdots & & \vdots \\ \mathrm{Re}(\varphi_{n-1}(t_0) \cdot \mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_{n-1}(t_n) \cdot \mathrm{e}^{-\mathrm{i}\alpha_n}) \end{bmatrix}$$
    $Y \leftarrow \{\mathrm{Re}(f(t_0) * \mathrm{e}^{-\mathrm{i}\alpha_0}), \ldots, \mathrm{Re}(f(t_n) * \mathrm{e}^{-\mathrm{i}\alpha_n})\}$
    if V non-singular **and** LINPROG(V, Y) solvable **then**
      $h \leftarrow$ LINPROG(V,Y)
      $\{\eta, \lambda_0, \ldots, \lambda_{n-1}\} \leftarrow (V^{\mathrm{T}})^{-1} Y$
      $p(t) \leftarrow \lambda_0 \varphi_0(t) + \ldots + \lambda_{n-1} \varphi_{n-1}(t)$
      return $p(t), T, A, h$
    else
      $t_k, \alpha_k \leftarrow t_{temp}, \alpha_{temp}$

Chapter 7

The problem

In order to create a new set to experiment with for non-normal matrices we need a different set. As stated in [3] this set is not known, but we will use the smallest circle centered at 0 that encompasses all eigenvalues.

We assume that the eigenvalues are known and that

$$p^* = \arg\min_{p \in \mathcal{P}_n} \|\chi_A - p\|_{\infty,\Lambda(A)}$$

holds for non-normal matrices.

We define $r := \min_{\lambda \in \Lambda(A)} |\lambda|$ to be the radius of the disc with the largest eigenvalue on the border. We have then that for the circle

$$\Omega = \{z \in \mathbb{C}; |z| \leq r\}$$

that

$$\|\chi_A - p^*\|_{\infty,\Omega} \geq \|\chi_A - p^*\|_{\infty,\Lambda(A)}.$$

We have by the maximum principle that the modulus of any analytic function $|f(z)|$ obtains its maximum on the boundary.

Theorem 7.0.1. *[6, p. 92] Let f be an analytic function on the connected domain $\Omega \subset \mathbb{R}$ and $\omega_0 \in \Omega$ a point in Ω. If $|f(\omega_0)| > |f(\omega)|$ for all $\omega \in \Omega$, then ω_0 is on the boundary of Ω.*

Therefore with the border of Ω denoted as $\delta\Omega$ we have that

$$\|\chi_A - p^*\|_{\infty,\delta\Omega} = \|\chi_A - p^*\|_{\infty,\Omega} \geq \|\chi_A - p^*\|_{\infty,\Lambda(A)}.$$

A polynomial of degree n defined on $\delta\Omega$ can be written as

$$P(z) = \sum_{k=0}^{n} c_k z^k \quad z \in \delta\Omega, c_k \in \mathbb{C}.$$

We rewrite this in polar form as

$$p(t) = \sum_{k=0}^{n} c_k r\, \mathrm{e}^{\mathrm{i}2\pi kt} \quad t \in [0,1], c_k \in \mathbb{C}.$$

We further define $a_k + \mathrm{i}b_k := c_k r$ and get

$$p(t) = \sum_{k=0}^{n} (a_k + \mathrm{i}b_k)\, \mathrm{e}^{\mathrm{i}2\pi kt} \quad t \in [0,1]\,, a_k, b_k \in \mathbb{R}. \tag{7.1}$$

Now we define

$$\varphi_k(t) := \begin{cases} \mathrm{e}^{\mathrm{i}2\pi kt} & \text{for } 0 \le k \le n \\ \mathrm{i}\,\mathrm{e}^{\mathrm{i}2\pi kt} & \text{for } n+1 \le k \le 2n+1 \end{cases}$$

and

$$\lambda_k := \begin{cases} a_k & \text{for } 0 \le k \le n \\ b_k & \text{for } n+1 \le k \le 2n+1 \end{cases}$$

which gives us

$$\begin{aligned} p(t) &= \sum_{k=0}^{n} \lambda_k \varphi_k(t) + \sum_{k=n+1}^{2n+1} \lambda_k \varphi_k(t) \\ &= \sum_{k=0}^{2n+1} \lambda_k \varphi_k(t) \end{aligned}$$

for $t \in [0,1]$, $\lambda_k \in \mathbb{R}$.

This we recognize as being of the form used in the problem as presented by Tang in Equation 6.1 and thus his algorithm will be applicable.

Remark 7.0.2. *It should be noted that despite Tang not fully making use of Haar spaces, the polynomial in Equation 7.1 evaluated on a circle is a complex trigonometric polynomial, which is in fact a Haar space and thus has a best approximation.*

7.1 Approximating the characteristic polynomial with Tangs algorithm.

In order to approximate a function χ of the $m \times m$ matrix A with a polynomial $p^* \in \mathcal{P}_n$ using Tangs algorithm we follow the three steps as outlined in [2, p. 725].

Step 0 Initialize the system by generating $2n+3$ points $(t_k, \alpha_k) \in [0,1] \times [0, 2\pi]$ with $k = 0, \ldots 2n+2$ and set up the Vandermonde matrix A such that

$$A^{\mathrm{T}} = A(t,\alpha)^{\mathrm{T}} = \begin{bmatrix} 1 & \cdots & 1 \\ \mathrm{Re}(\varphi_0(t_0)\,\mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_0(t_{2n+2})\,\mathrm{e}^{-\mathrm{i}\alpha_{2n+2}}) \\ \vdots & & \vdots \\ \mathrm{Re}(\varphi_{2n+1}(t_0)\,\mathrm{e}^{-\mathrm{i}\alpha_0}) & \cdots & \mathrm{Re}(\varphi_{2n+1}(t_{2n+2})\,\mathrm{e}^{-\mathrm{i}\alpha_{2n+2}}) \end{bmatrix},$$

and the vector

$$c = c(t,\alpha) = \begin{bmatrix} \mathrm{Re}(\chi_A(t_0)\,\mathrm{e}^{-\mathrm{i}\alpha_0}) \\ \vdots \\ \mathrm{Re}(\chi_A(t_{2n+2})\,\mathrm{e}^{-\mathrm{i}\alpha_{2n+2}}) \end{bmatrix}$$

We then minimize $h = c^{\mathrm{T}} r$ subject to the constraints

$$r := A^{-1} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \geq 0,$$

i.e. $r_k \in [0, 1]$ for $k = 0, \ldots, 2n + 2$.

Step 1 We find a updating element $(\hat{t}, \hat{\alpha}) \in [0, 1] \times [0, 2\pi]$ such that

$$\left(\chi_A(\hat{t}) - \sum_{k=0}^{2n+1} \lambda_k \varphi_k(\hat{t}) \right) \mathrm{e}^{-i\hat{\alpha}} = \left\| \chi_A - \sum_{k=0}^{2n+1} \lambda_k \varphi_k \right\|_\infty .$$

If $\left\| \chi_A - \sum_{k=0}^{2n+1} \lambda_k \varphi_k \right\|_\infty - h \leq h\epsilon$ for some tolerance ϵ we terminate the algorithm.

Step 2 Swap an appropriate element (t_k, α_k) with the new $(\hat{t}, \hat{\alpha})$ and then go back to step 1.

Chapter 8

Experiment

The matrices to be approximated were constructed as companion matrices to characteristic equations of the desired eigenvalues. Although, in the case of purely real eigenvalues, a diagonal matrix of the eigenvalues was instead used.

In order to find the new candidate in Step 2 of Tangs algorithm we generated a set of 512 possible candidates T, selecting $\hat{t} = \max_{t \in T} |(\chi_A - p)(t)|$. The corresponding $\hat{\alpha}$ was taken as $\hat{\alpha} = \text{Arg}((\chi_A - p)(\hat{t}))$.

The Tang algorithm was repeated until the change in $(\|\chi_A - p\|_\infty - h)/h$ between iterations was below a tolerance of 10^{-8}. The algorithm was also aborted if the optimal point to swap in was already used, if no feasible swap was possible or if the number of iterations performed passed $100 \times \deg(p)$.

After the algorithm finished the roots were calculated as the eigenvalues of the companion matrix or as the zeros of the polynomial using the roots function found in the python package numpy.

The optimal Arnoldi q was constructed by using a minimizing function found in the python package scipy in order to find the vector b maximizing $\|p(A)b\|_2$.

Then the results from Tangs algorithm and the eigenvalues of the optimal Arnoldi matrix were plotted together with the true eigenvalues of the original matrix.

The experiment was repeated with both monic and non-monic approximations.

All the code used in the experiment can be requested from the department of numerical analysis at Lund University.

Chapter 9

Results and Discussion

We note that the approximations of the same degree as the characteristic polynomials all converge to the correct eigenvalues, thereby we can confidently state that the algorithm works.

When we look at how the methods perform compared to each other it is not always clear from the images which is best. In majority of the cases with eigenvalues on the unit circle however the Arnoldi method produce a closer approximation than Tangs method.

The first set of plots, found in Appendix A, were generated for six different 10×10 matrices. The different matrices had the real part of the eigenvalues either evenly spaced or in clusters. Further they were either all real, had a complex part chosen as to put them on a circle or had their complex part evenly spaced between 0 and ± 1, per cluster if applicable. All different types of matrices were approximated with a monic and a non-monic approximand and the ideal Arnoldi approximation done with scipy.fsolve. Both the Tang algorithm and fsolve had 10^{-8} as a tolerance.

The second set of plots, found in Appendix B were generated in the same way but using 11×11 matrices.

Some interesting things to note here is that for the non-monic approximations, the Tang algorithms performance was very bad for most odd degree approximations of the 10×10 matrix and the even approximations of the 11×11 matrix. Indicating an issue dealing with odd approximands to even functions and vice versa. For both 10×10 and 11×11 the non-monic performed best for the matrices with clustered eigenvalues situated on a circle.

The monic approximations performed much more evenly, however not always better.

For the matrices with purely real eigenvalues the Arnoldi approximation works very well, which the Tang algorithm does not. It is important to note here that for these normal matrices we did not use the $\Lambda(A)$ set detailed in Chapter 2 but rather the same naïve set as for the non-normal matrices. The reason for this seems to be that the Arnoldi method yields eigenvalues without a imaginary part, while the Tang method still produces complex roots.

In Appendix C we see the result of an experiment where the eigenvalues

of a matrix were placed in clusters with the complex part of th eigenvalues chosen in such a way as to place them on a circle. The experiment was repeated with circles of different radii. Here we can note something interesting. Our approximation has five eigenvalues clustered around the true ones that are the same from radius 2 and onward, only scaled differently. The same is true for the three centered eigenvalues of Arnoldi. But if we look at the average of modulus the eigenvalues of our approximation, we can see that between radius 13 and 14 in increases by about 10, from 14 to 15 by about 11, then 12, 12, 13. No huge steps. While for the Arnoldi approximation we see that for the same steps we have increments of 90, -206, 2588, -1311, 1672. Both larger increments, and also alternating between increasing and decreasing.

On the machine used for these approximation the Arnoldi approximations of this degree all finished in around one second, while Tangs method finished in between about 8 and 12 seconds.

While at lower degrees both methods finish after a similar time, at higher degrees the time it takes for Tangs algorithm to run increases significantly. In particular, it is finding appropriate initial points that takes time. On the computer the simulations were done, finding initial points for a degree 15 approximation of a 20×20 matrix with eigenvalues with real part on $[-1, 1]$ takes over 10 minutes, comparatively the approximations of a 10×10 matrix with eigenvalues with real part on $[-1, 1]$ takes less than a minute. Experiments point to the range in which the α values of Tangs algorithm are generated to be a deciding factor. Higher degree approximations seem to be faster for a narrower range, while lower degrees fare better with a wide range. This could be a subject for further study, one interesting experiment would be to set up a range of Chebyshev points, picking the the first range between the two outermost points, and for every increase of degree use a range closer to the middle of the Chebyshev points.

As it works now, the implementation randomizes one alpha on $[0, 2\pi]$ and then the subsequent ones are generated in such a way that the algorithm is sure that they have the properties we desire. In the version presented in [2], the suggested method is to generate a set of α in $[0, 2\pi]$ and then change the individual points from α_k to $\alpha'_k \in \{\alpha_k, \alpha_k + \pi\}$ until such a time as that the Vandermonde matrix is not singular, but the time complexity using this method was way too high to comfortably use.

Since we are working with a larger than optimal set of points for the Tang algorithm we will always be overshooting, getting a larger value for $\|f - s^*\|_{\infty,\Omega}$ then we would for a better set. In the few cases where the Tang method outperformed the Arnoldi method, this can probably be explained by the property of Arnoldi to instead of always minimizing the difference to the function to be approximated, instead minimizes the difference to the next step of the iterative algorithm.

Chapter 10

Future work

At the moment the initiation of the algorithm is the slowest part, in particular choosing the initial *alpha*-values. Some experiments indicate that a higher degrees of approximation are faster with a narrower interval in which the alphas are generated from. However, the speed of lower degrees of approximation seem to be negatively effected by this. Experiments with a varying range for the alphas would be interesting, perhaps a interval of twice as many Chebyshev points as alphas centered around a point opposite the previous alpha. For the first the degree of approximation the two points furthest away from the center would span the interval and for each degree higher, a step inwards would reduce the size of the interval.

Tangs algorithm does not force the coefficients to be real. The solution just converges to the solution that has real coefficients. If we instead of using linear programming to find the solution were to use constrained least squares we could perhaps minimize the number of iterations necessary to arrive at a correct solution.

Bibliography

[1] P. Tang, *Chebyshev approximation on the complex plane.* University of California, Berkeley, 1987.

[2] ——, "A fast algorithm for linear complex Chebyshev approximations," *Mathematics of Computation*, vol. 51, no. 184, pp. 721–739, 1988.

[3] A. Greenbaum and L. Trefethen, "GMRES/CR and Arnoldi/Lanczos as matrix approximation problems," *SIAM Journal on Scientific Computing*, vol. 15, no. 2, pp. 359–368, 1994.

[4] K. Ufymtseva, "Arnoldi iteration and chebyshev polynomials," Master's thesis, Lund University, 2021.

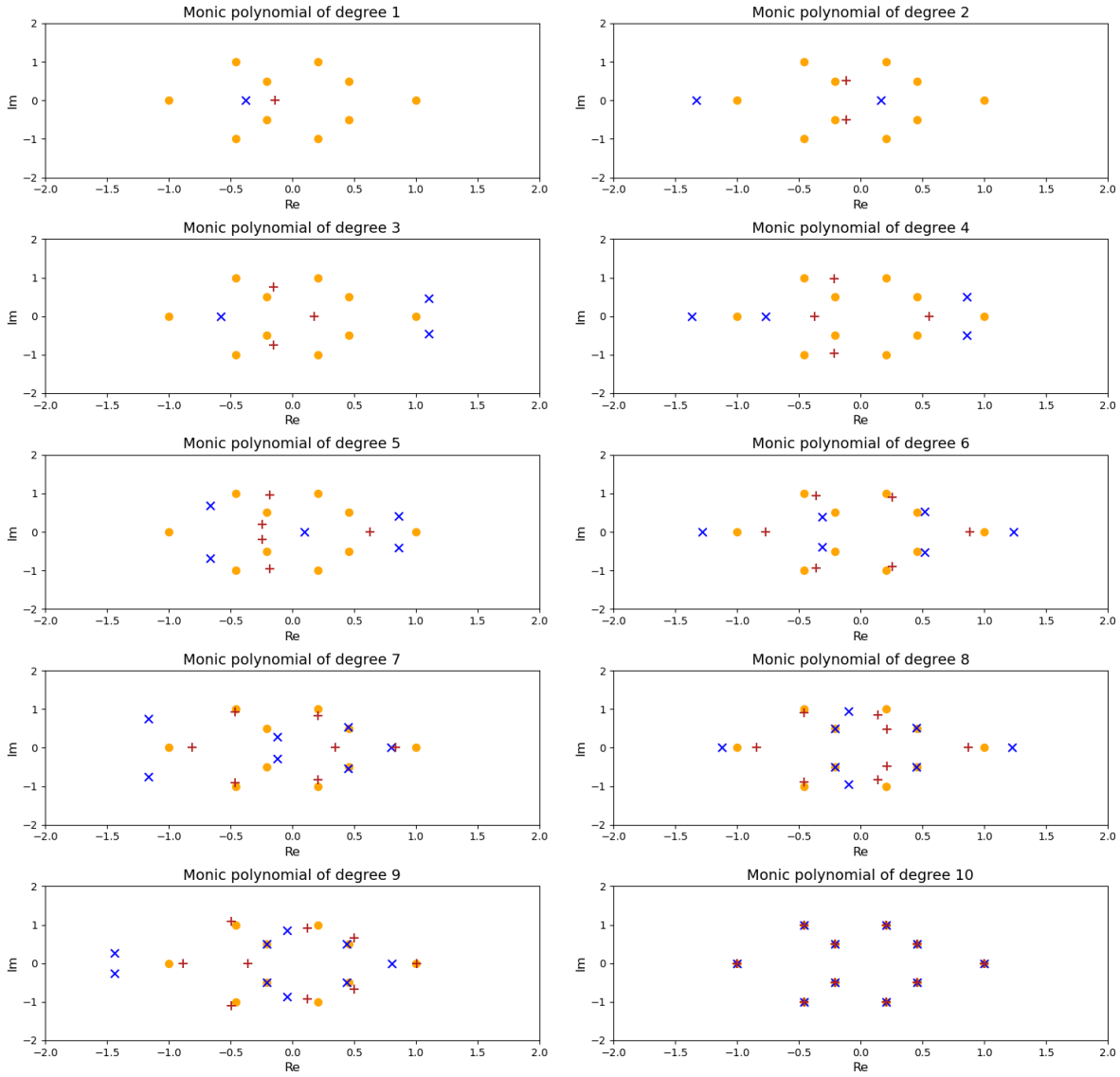
[5] A. Iske, *Approximation Theory and Algorithms for Data Analysis.* Cham Springer, 2018.

[6] S. Lang, *Complex Analysis.* Springer New York, NY, 1999.

Appendix A

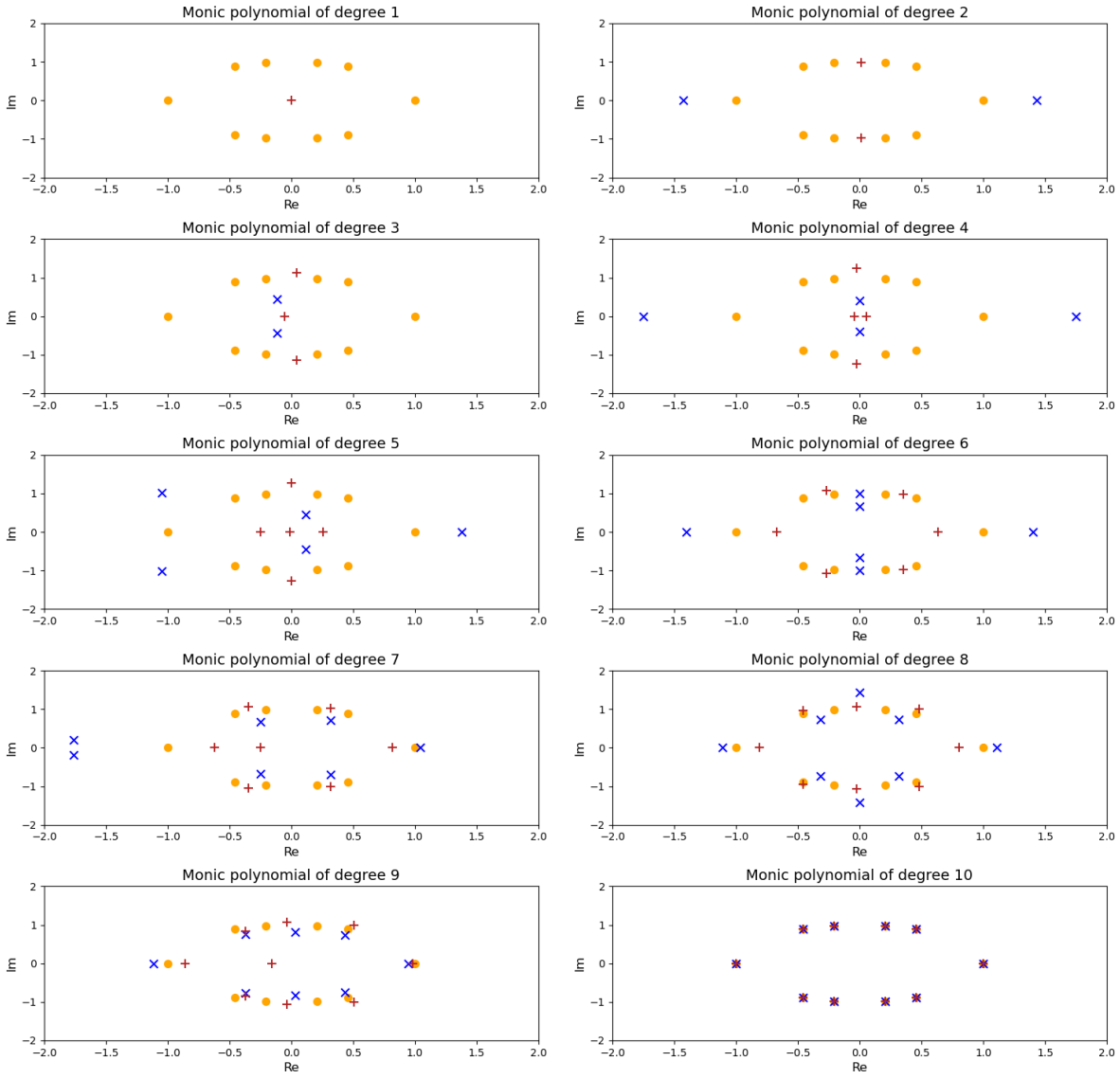
Approximations of 10×10 matrices

This appendix contains the figures of eigenvalues for different 10×10 matrices and approximations thereof, coupled with a table of relevant values. The true eigenvalues of the original matrix are represented by orange circles, the eigenvalues of the Arnoldi approximation as red crosses and our approximations using Tangs method are shown as blue plusses. In the case of monic polynomials, they are the eigenvalues of the companion matrix to the polynomial. For the non-monic polynomials they are values returned by numpy.roots.



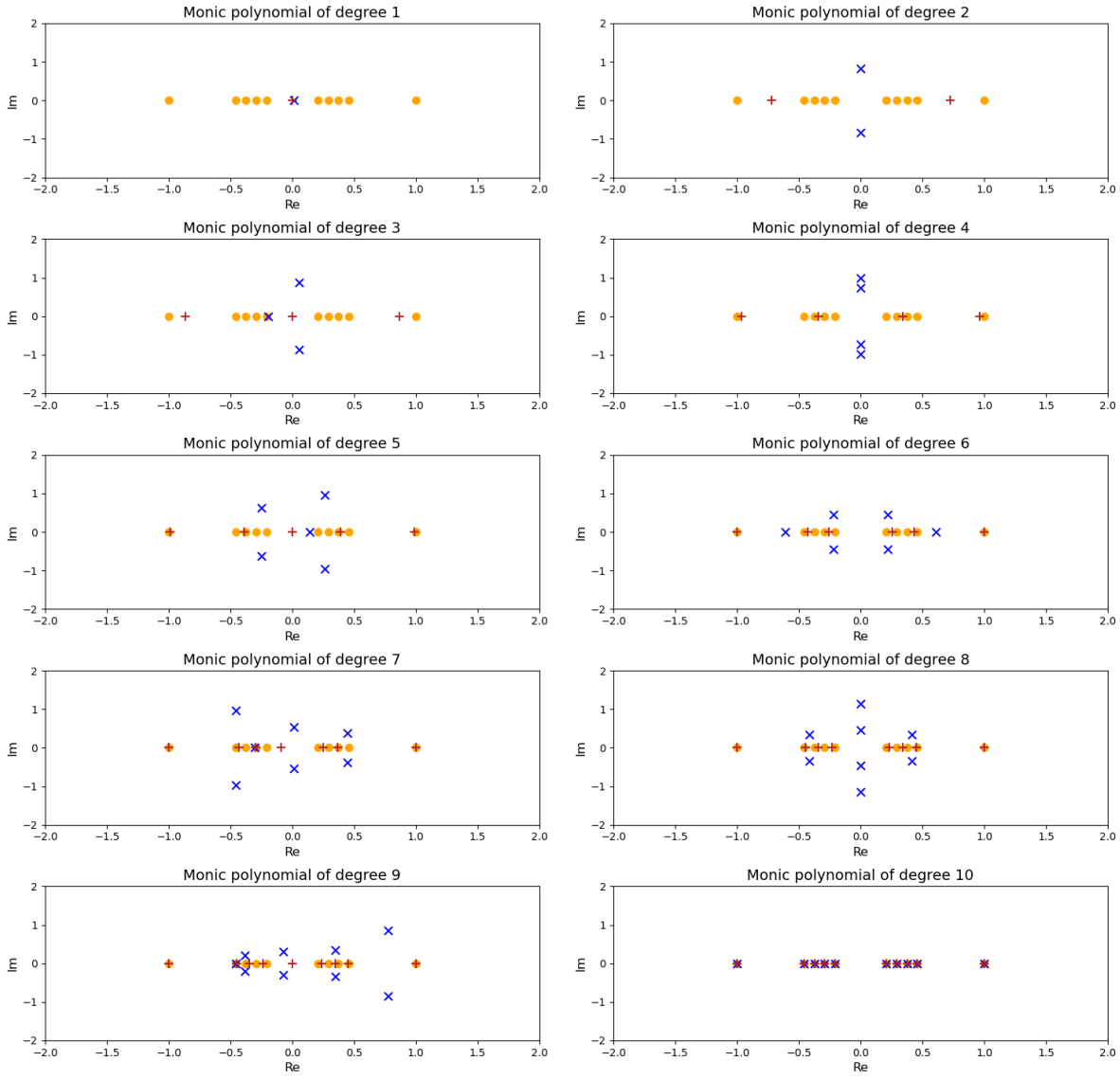
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.375	0.375	0.375	0.143	0.143	0.143
2	0.163	1.33	0.746	0.522	0.522	0.522
3	0.581	1.195	0.99	0.179	0.774	0.575
4	0.772	1.367	1.034	0.371	0.991	0.726
5	0.097	0.953	0.78	0.309	0.978	0.641
6	0.503	1.283	0.836	0.77	1.014	0.924
7	0.309	1.384	0.798	0.345	1.031	0.825
8	0.542	1.225	0.837	0.525	1.008	0.81
9	0.542	1.461	0.872	0.362	1.197	0.908
10	0.542	1.1	0.868	0.542	1.1	0.868

Figure A.1: Plot and table for different degrees of monic approximations of a 10×10 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts equidistantly placed on $[-1, 1]$ per cluster.



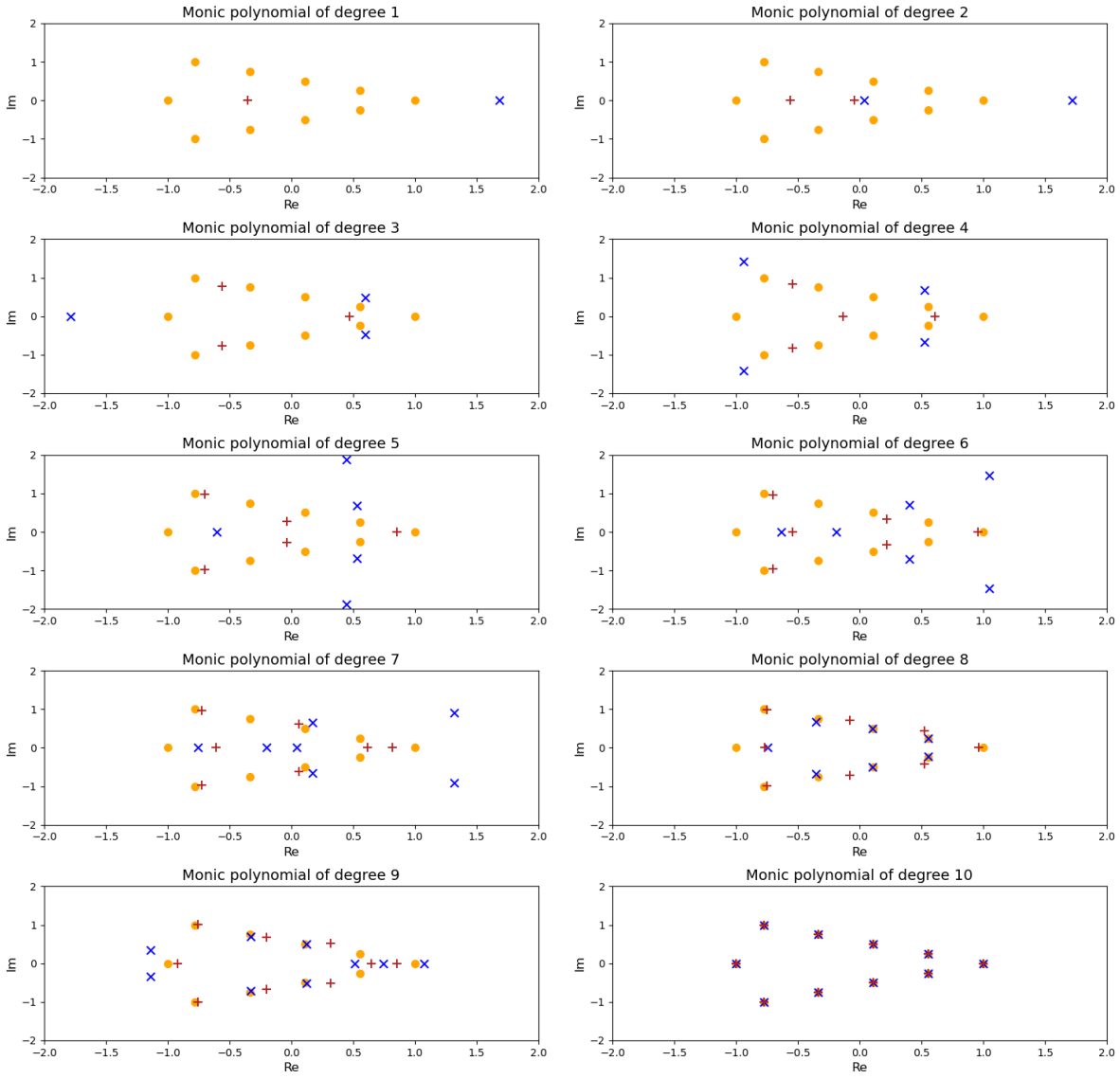
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	2.468	2.468	2.468	0.001	0.001	0.001
2	1.43	1.43	1.43	0.977	0.977	0.977
3	0.451	2.883	1.261	0.055	1.137	0.776
4	0.4	1.748	1.074	0.045	1.246	0.648
5	0.474	1.463	1.05	0.01	1.273	0.613
6	0.674	1.402	1.023	0.635	1.101	0.929
7	0.724	1.774	1.084	0.252	1.11	0.863
8	0.8	1.429	1.035	0.803	1.118	1.013
9	0.822	2.617	1.083	0.155	1.122	0.914
10	1.0	1.0	1.0	1.0	1.0	1.0

Figure A.2: Plot and table for different degrees of monic approximations of a 10×10 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts chosen as to place them on a circle.



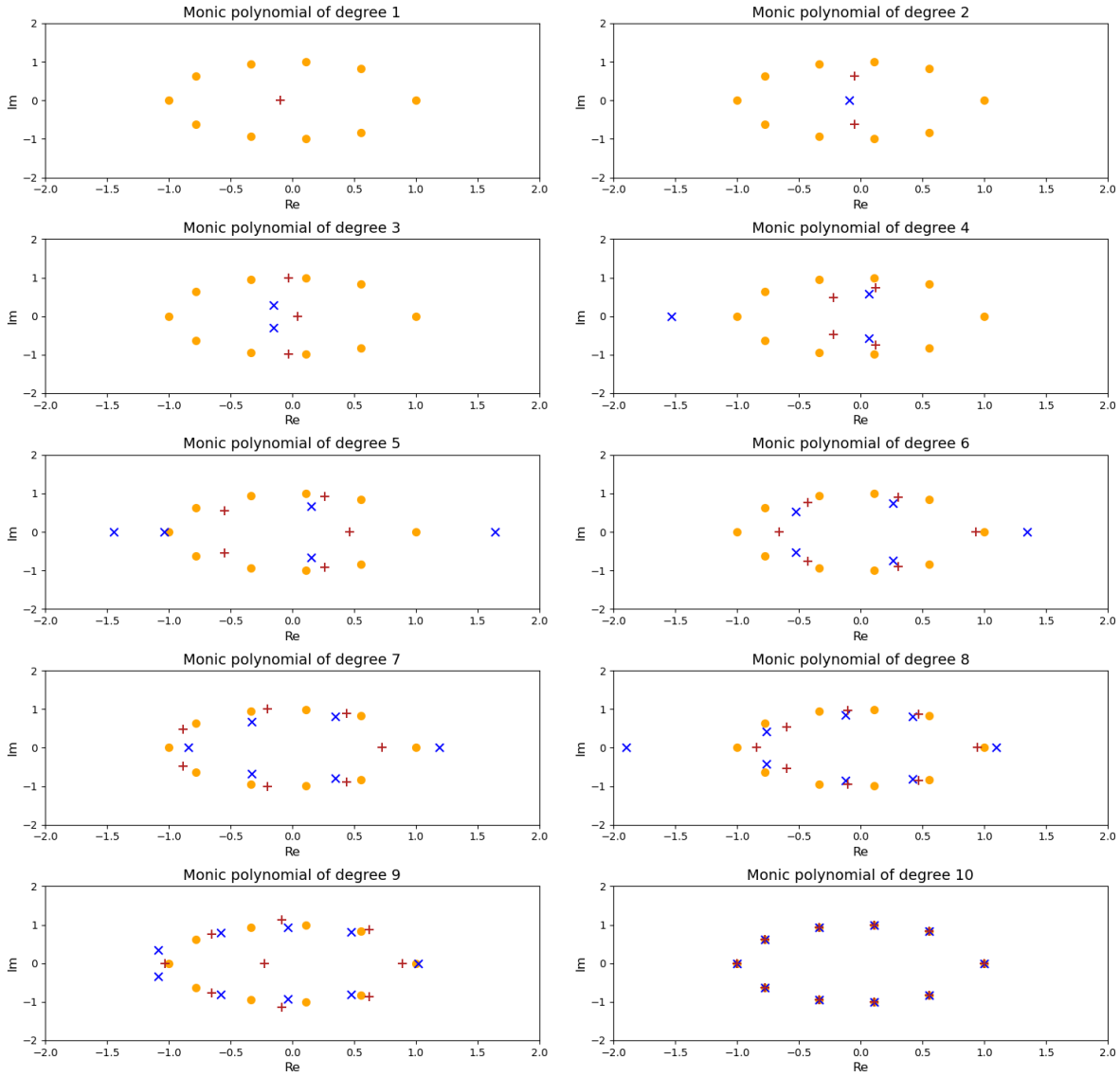
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.013	0.013	0.013	0.001	0.001	0.001
2	0.831	0.831	0.831	0.722	0.722	0.722
3	0.197	0.867	0.643	0.0	0.867	0.578
4	0.728	0.982	0.855	0.344	0.962	0.653
5	0.139	1.002	0.701	0.0	0.988	0.55
6	0.501	0.61	0.537	0.254	0.998	0.561
7	0.301	1.067	0.67	0.091	1.0	0.491
8	0.463	1.142	0.671	0.23	1.0	0.505
9	0.314	1.15	0.581	0.0	1.0	0.453
10	0.208	1.0	0.467	0.208	1.0	0.467

Figure A.3: Plot and table for different degrees of monic approximations of a 10×10 matrix with purely real eigenvalues in clusters.



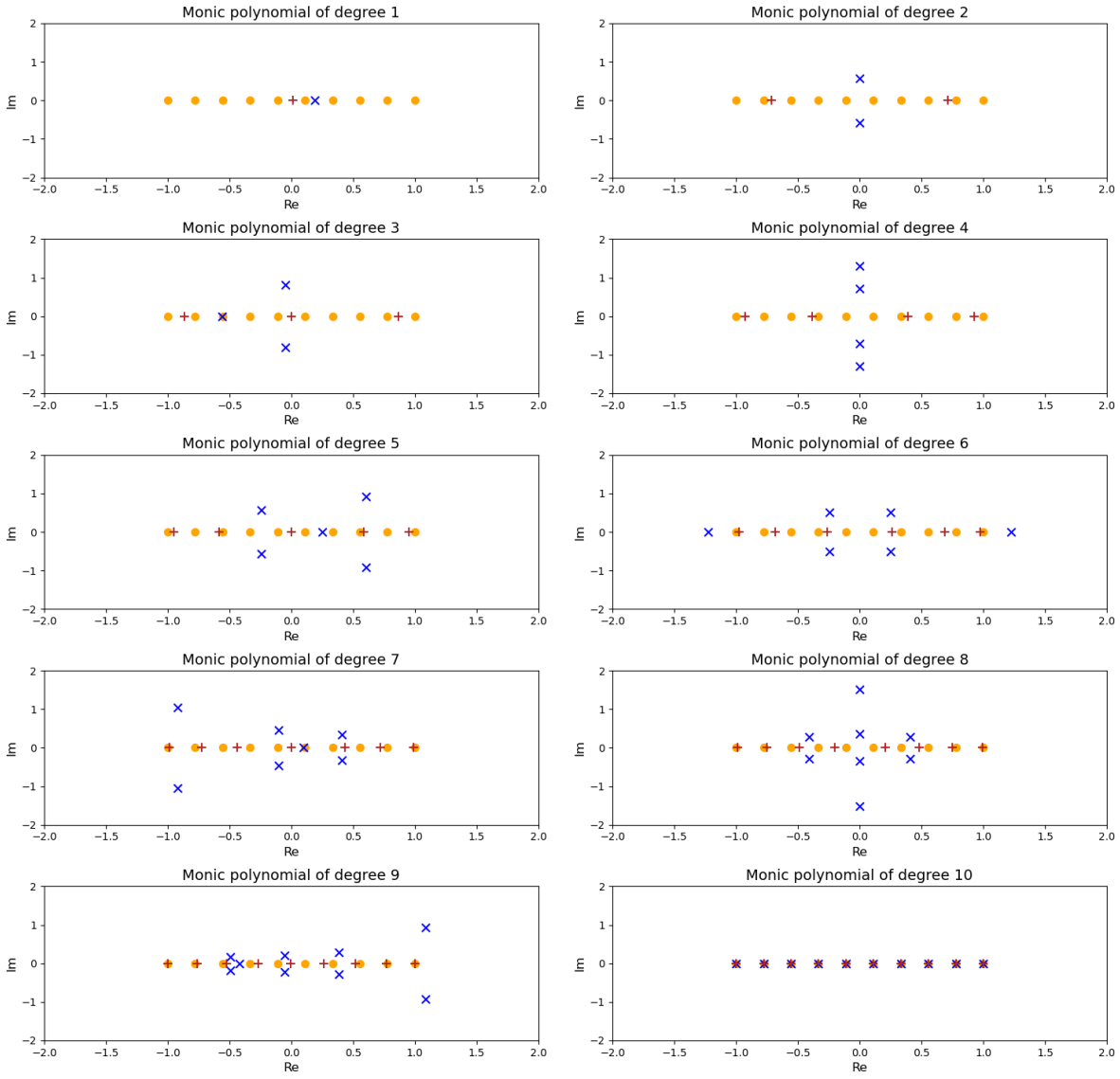
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	1.682	1.682	1.682	0.351	0.351	0.351
2	0.036	1.72	0.878	0.044	0.562	0.303
3	0.762	1.789	1.104	0.47	0.954	0.792
4	0.856	1.703	1.28	0.133	0.991	0.681
5	0.601	1.935	1.243	0.268	1.2	0.758
6	0.192	1.811	1.012	0.406	1.19	0.782
7	0.042	1.6	0.791	0.606	1.209	0.812
8	0.511	2.386	0.859	0.676	1.243	0.875
9	0.515	1.187	0.811	0.613	1.262	0.843
10	0.512	1.267	0.842	0.512	1.267	0.842

Figure A.4: Plot and table for different degrees of monic approximations of a 10×10 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts equidistantly placed on $[-1, 1]$.



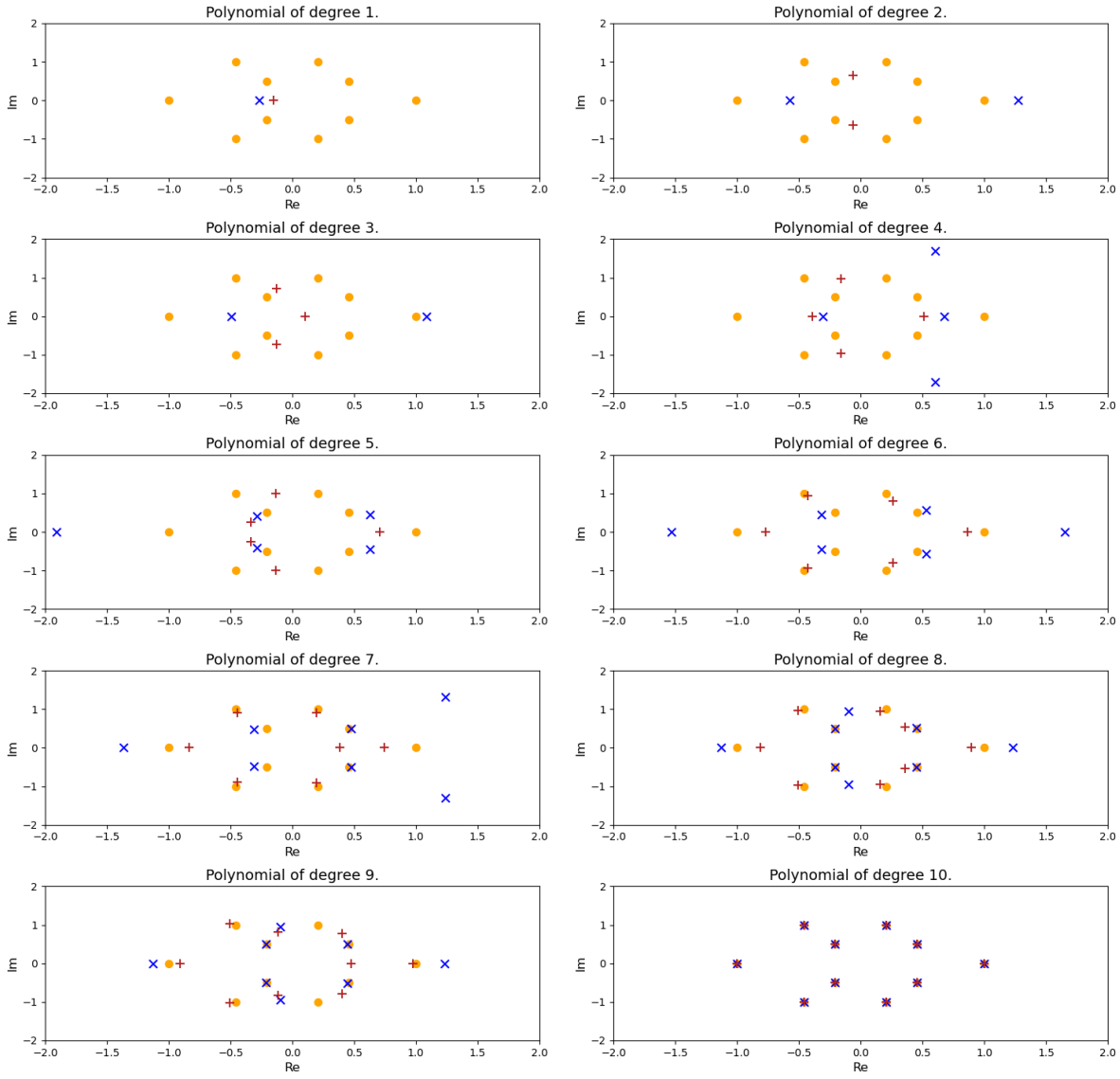
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	3.024	3.024	3.024	0.094	0.094	0.094
2	0.094	3.516	1.805	0.628	0.628	0.628
3	0.329	3.598	1.419	0.044	0.984	0.671
4	0.584	2.276	1.243	0.523	0.754	0.639
5	0.684	1.641	1.098	0.464	0.96	0.784
6	0.743	2.296	1.118	0.661	0.942	0.87
7	0.754	2.197	1.069	0.728	1.021	0.967
8	0.86	1.897	1.037	0.807	0.978	0.91
9	0.927	1.14	1.003	0.227	1.137	0.952
10	1.0	1.0	1.0	1.0	1.0	1.0

Figure A.5: Plot and table for different degrees of monic approximations of a 10×10 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts chosen as to place them on a circle.



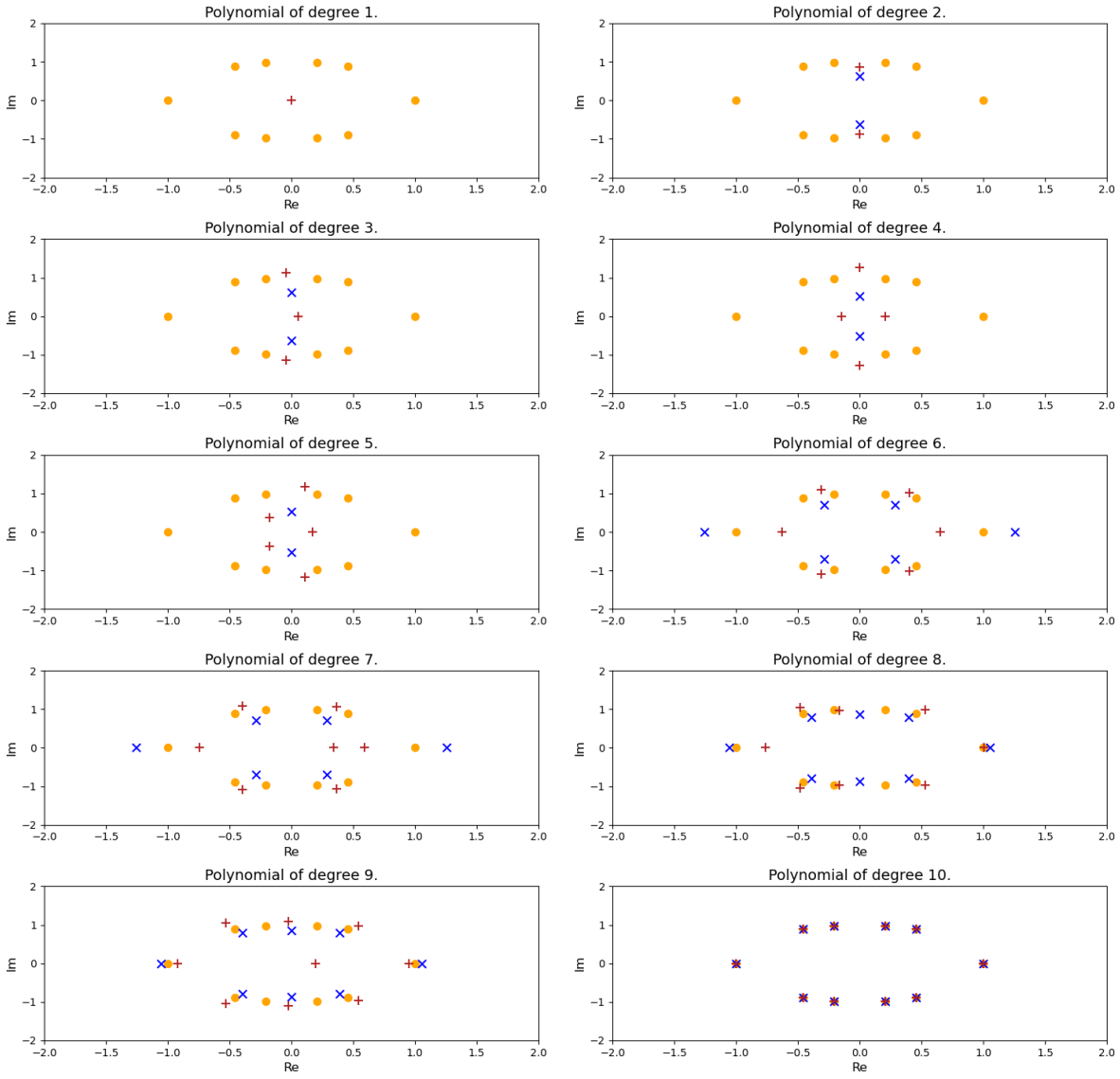
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.187	0.187	0.187	0.012	0.012	0.012
2	0.581	0.581	0.581	0.711	0.711	0.711
3	0.563	0.813	0.73	0.0	0.868	0.579
4	0.72	1.303	1.011	0.387	0.929	0.658
5	0.249	1.1	0.734	0.001	0.952	0.615
6	0.558	1.225	0.78	0.265	0.975	0.642
7	0.098	1.394	0.696	0.001	0.987	0.613
8	0.352	1.511	0.715	0.204	0.993	0.608
9	0.218	1.433	0.635	0.004	1.0	0.568
10	0.111	1.0	0.556	0.111	1.0	0.556

Figure A.6: Plot and table for different degrees of monic approximations of a 10×10 matrix with purely real eigenvalues equidistantly distributed on $[-1, 1]$.



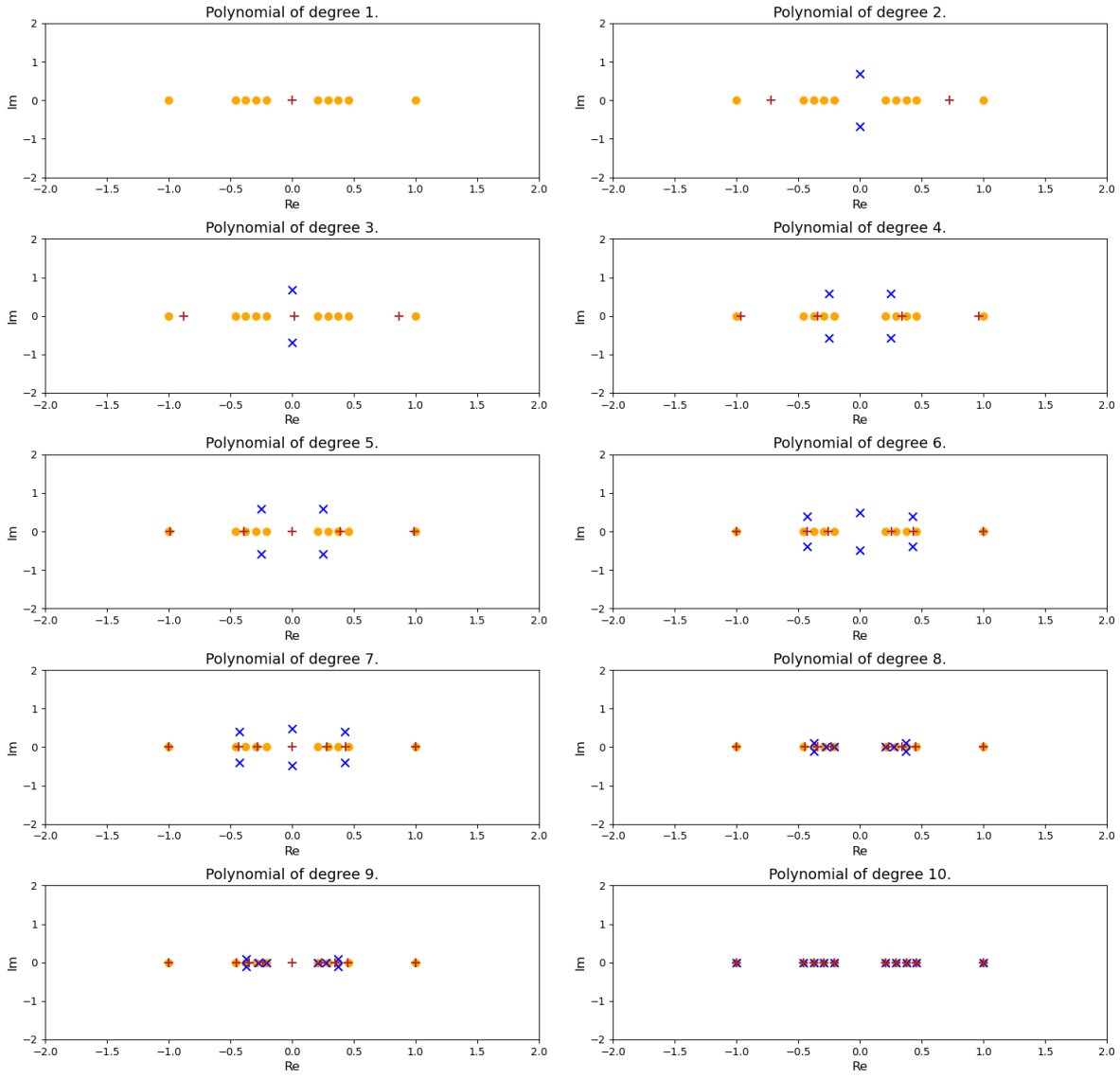
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.265	0.265	0.265	0.152	0.152	0.152
2	0.576	1.274	0.925	0.652	0.652	0.652
3	0.496	2.29	1.29	0.105	0.733	0.523
4	0.305	1.805	1.148	0.39	0.979	0.715
5	0.499	1.912	0.893	0.416	1.008	0.711
6	0.546	1.652	0.969	0.77	1.023	0.897
7	0.568	1.802	1.068	0.383	1.004	0.834
8	0.542	1.231	0.839	0.648	1.099	0.89
9	0.542	1547860568.436	171984508.35	0.477	1.141	0.894
10	0.542	1.1	0.868	0.542	1.1	0.868

Figure A.7: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts equidistantly placed on $[-1, 1]$ per cluster.



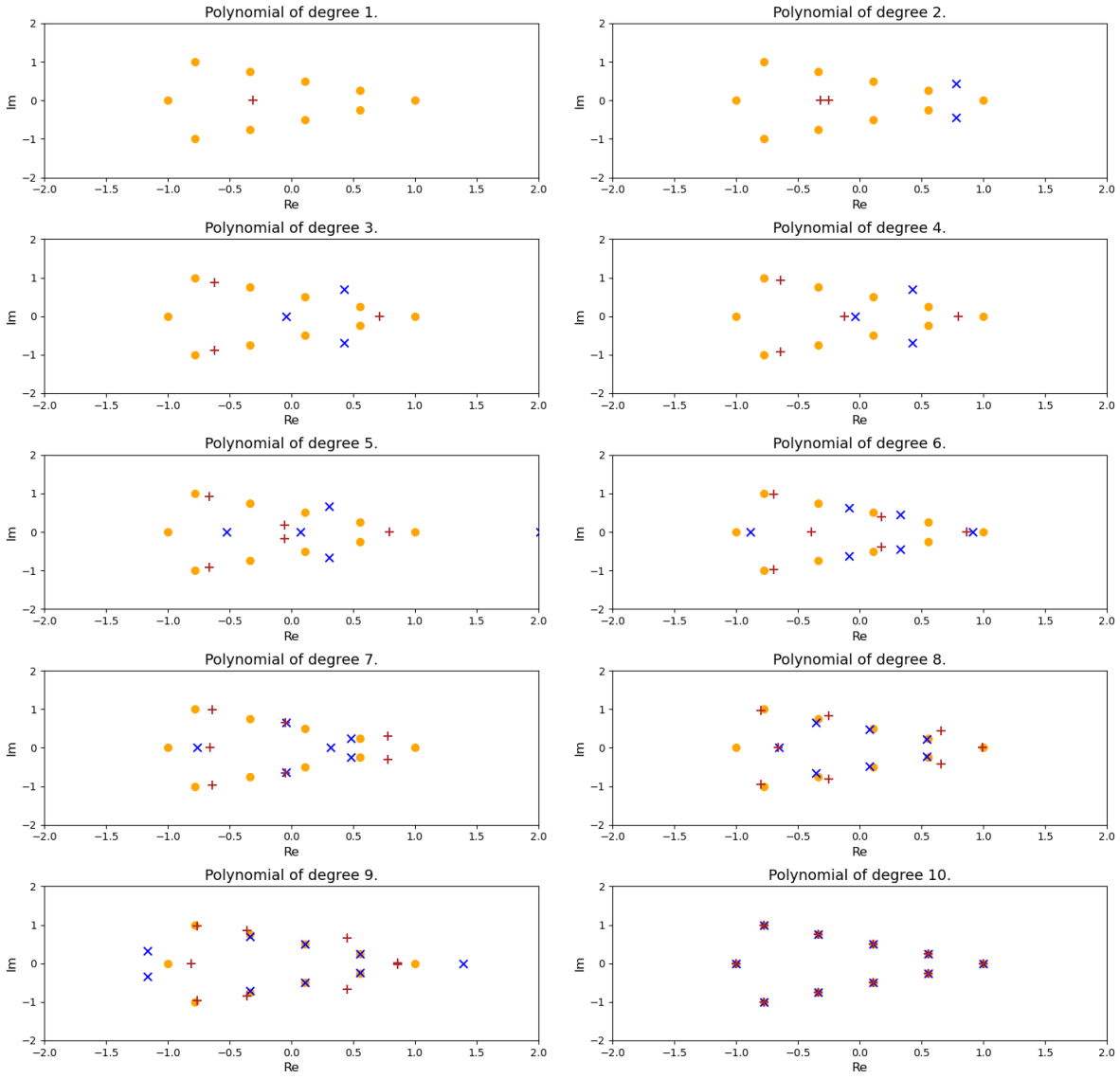
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	236549767.175	236549767.175	236549767.175	0.0	0.0	0.0
2	0.627	0.627	0.627	0.873	0.873	0.873
3	0.627	2531432556.841	843810852.698	0.055	1.137	0.776
4	0.523	3.069	1.796	0.147	1.269	0.722
5	0.523	34129268.512	6825855.139	0.173	1.176	0.668
6	0.762	1.256	0.927	0.628	1.134	0.953
7	0.762	5665689970.352	809384282.273	0.34	1.153	0.89
8	0.865	1.055	0.921	0.765	1.153	1.033
9	0.865	6819969392.713	757774377.787	0.193	1.169	0.981
10	1.0	1.0	1.0	1.0	1.0	1.0

Figure A.8: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts chosen as to place them on a circle.



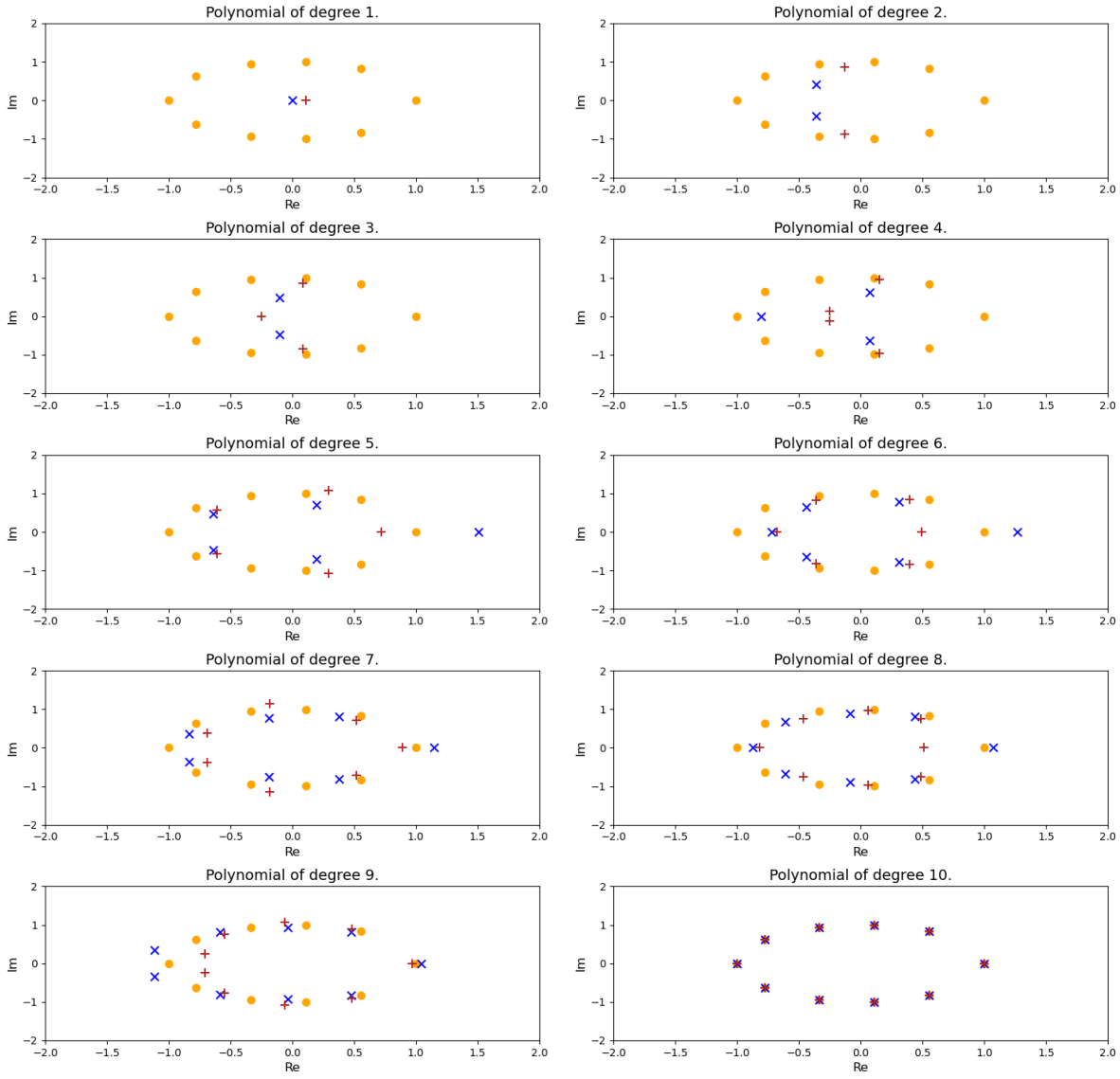
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	229967918.604	229967918.604	229967918.604	0.0	0.0	0.0
2	0.684	0.684	0.684	0.722	0.722	0.722
3	0.684	181576739.542	60525580.304	0.016	0.877	0.587
4	0.634	0.634	0.634	0.344	0.962	0.653
5	0.634	504397864.285	100879573.364	0.0	0.988	0.55
6	0.484	0.582	0.549	0.254	0.998	0.561
7	0.484	668563223.641	95509032.42	0.0	0.999	0.489
8	0.209	0.387	0.314	0.23	1.0	0.505
9	0.209	1970511555.87	218945728.709	0.001	1.0	0.453
10	0.208	1.0	0.467	0.208	1.0	0.467

Figure A.9: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with purely real eigenvalues in clusters.



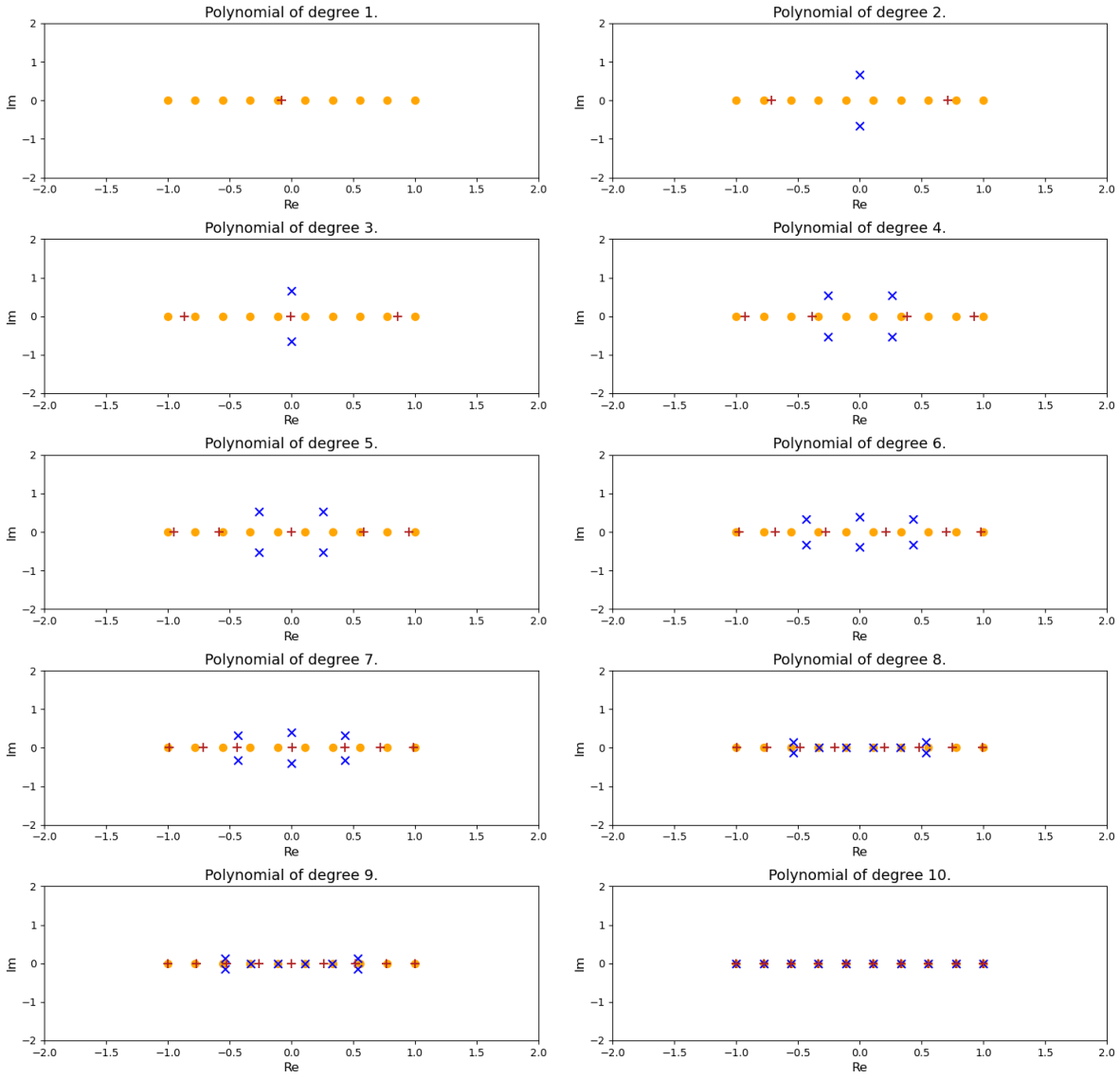
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	3.283	3.283	3.283	0.311	0.311	0.311
2	0.897	0.897	0.897	0.25	0.319	0.285
3	0.041	0.817	0.558	0.715	1.076	0.956
4	0.035	605.405	151.769	0.125	1.134	0.797
5	0.075	2.011	0.817	0.18	1.14	0.686
6	0.554	0.917	0.695	0.392	1.192	0.753
7	0.319	5.538	1.285	0.653	1.167	0.854
8	0.483	2.375	0.832	0.657	1.247	0.93
9	0.512	1.393	0.844	0.803	1.227	0.938
10	0.512	1.267	0.842	0.512	1.267	0.842

Figure A.10: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts equidistantly placed on $[-1, 1]$.



	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.0	0.0	0.0	0.113	0.113	0.113
2	0.547	0.547	0.547	0.876	0.876	0.876
3	0.497	5.853	2.282	0.25	0.86	0.657
4	0.63	2.061	1.031	0.278	0.97	0.624
5	0.737	1.507	0.914	0.717	1.108	0.918
6	0.718	1.266	0.869	0.492	0.924	0.8
7	0.786	1.148	0.904	0.783	1.155	0.933
8	0.871	1.074	0.925	0.509	0.969	0.854
9	0.934	1.167	1.017	0.748	1.076	0.949
10	1.0	1.0	1.0	1.0	1.0	1.0

Figure A.11: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts chosen as to place them on a circle.



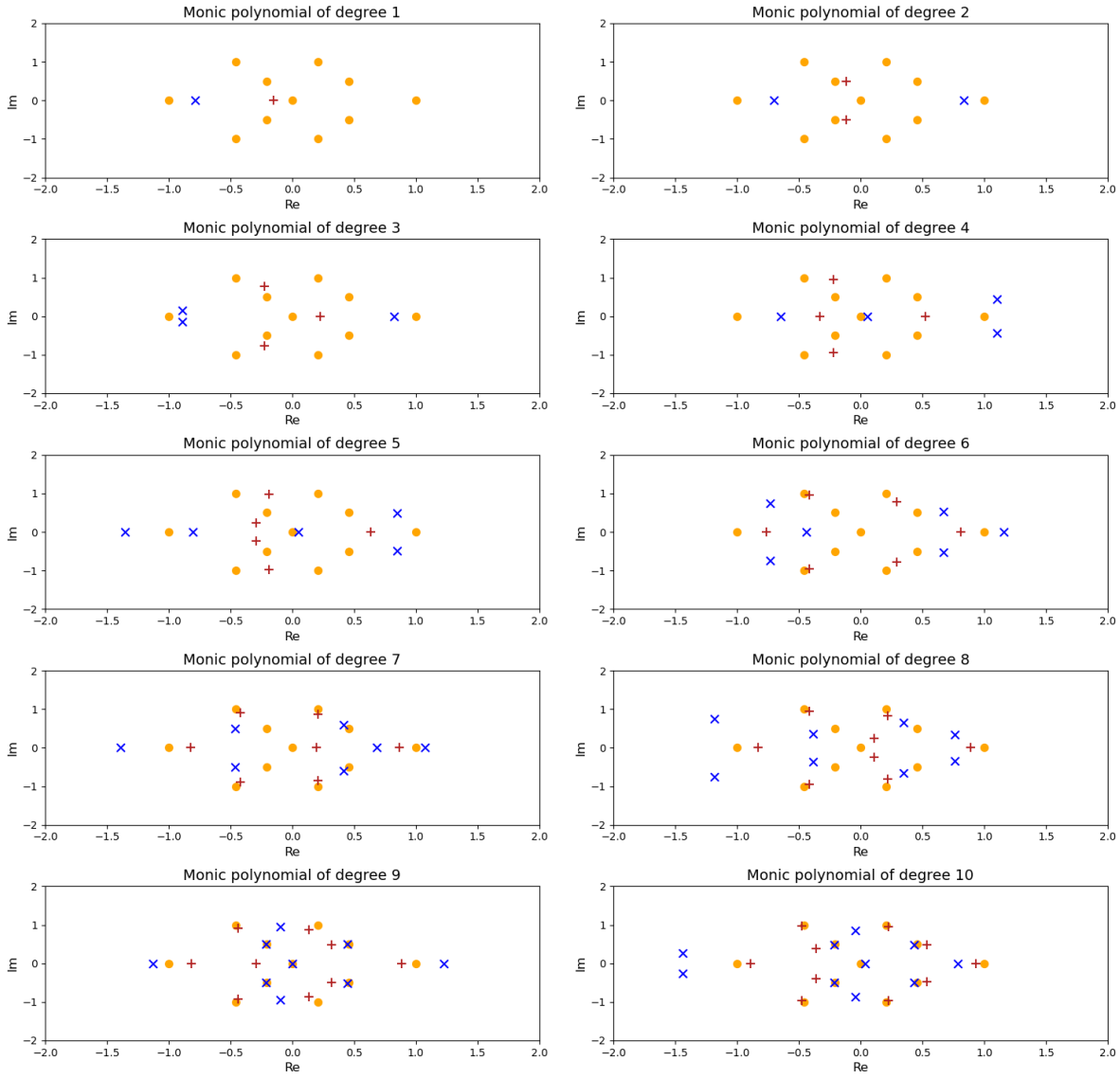
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	15890155.095	15890155.095	15890155.095	0.08	0.08	0.08
2	0.662	0.662	0.662	0.711	0.711	0.711
3	0.661	65452925.424	21817642.249	0.006	0.864	0.577
4	0.592	0.592	0.592	0.385	0.928	0.657
5	0.592	627031074.959	125406215.465	0.0	0.952	0.615
6	0.396	0.546	0.496	0.211	0.978	0.638
7	0.396	28391814393.806	4055973485.255	0.006	0.988	0.613
8	0.111	0.553	0.387	0.203	0.995	0.608
9	0.111	3008226366.875	334247374.441	0.0	0.998	0.567
10	0.111	1.0	0.556	0.111	1.0	0.556

Figure A.12: Plot and table for different degrees of non-monic approximations of a 10×10 matrix with purely real eigenvalues equidistantly distributed on $[-1, 1]$.

Appendix B

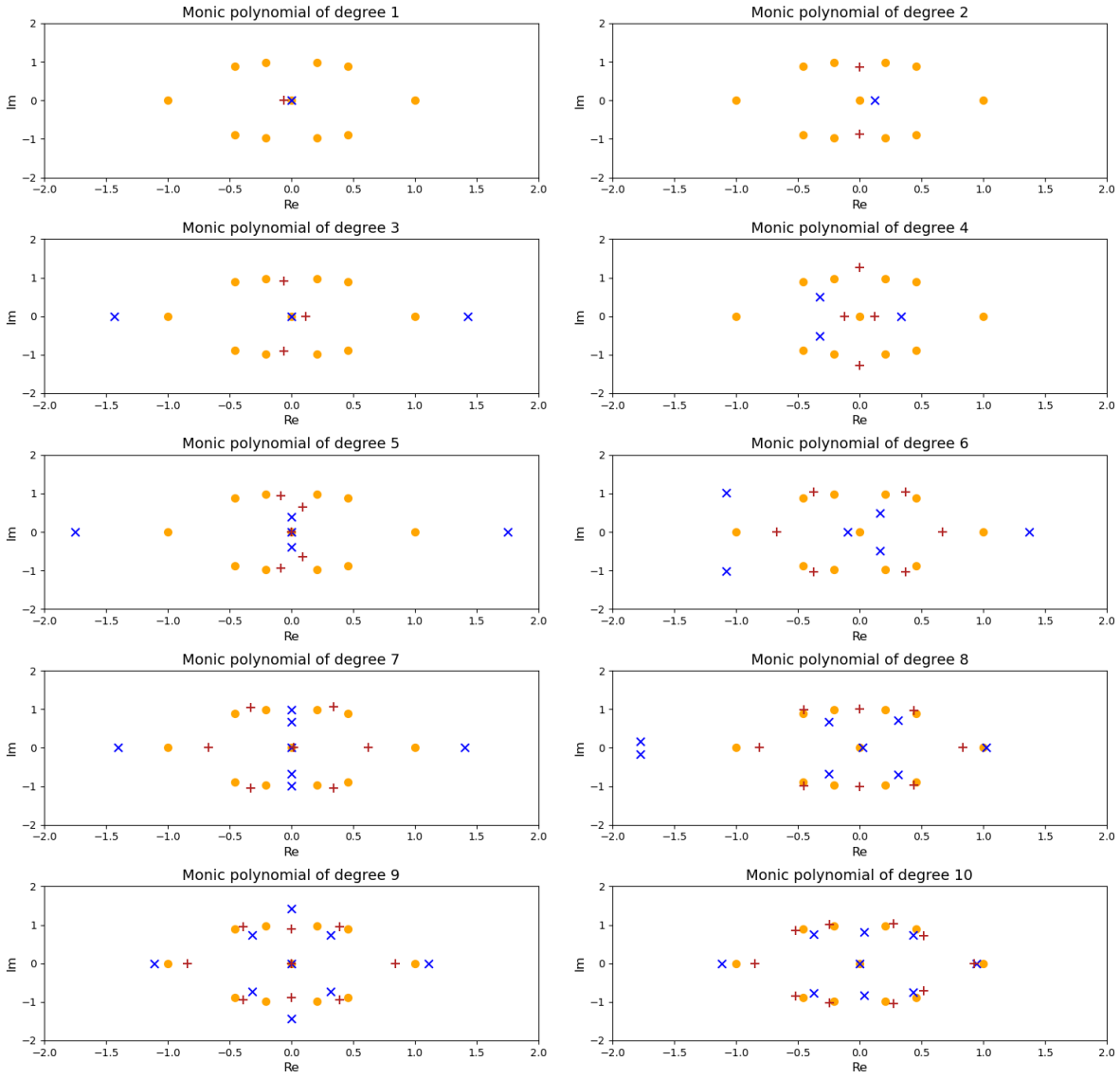
Approximations of 11×11 matrices

This appendix contains the figures of eigenvalues for different 11×11 matrices and approximations thereof, coupled with a table of relevant values. The true eigenvalues of the original matrix are represented by orange circles, the eigenvalues of the Arnoldi approximation as red crosses and our approximations using Tangs method are shown as blue plusses. In the case of monic polynomials, they are the eigenvalues of the companion matrix to the polynomial. For the non-monic polynomials they are values returned by numpy.roots.



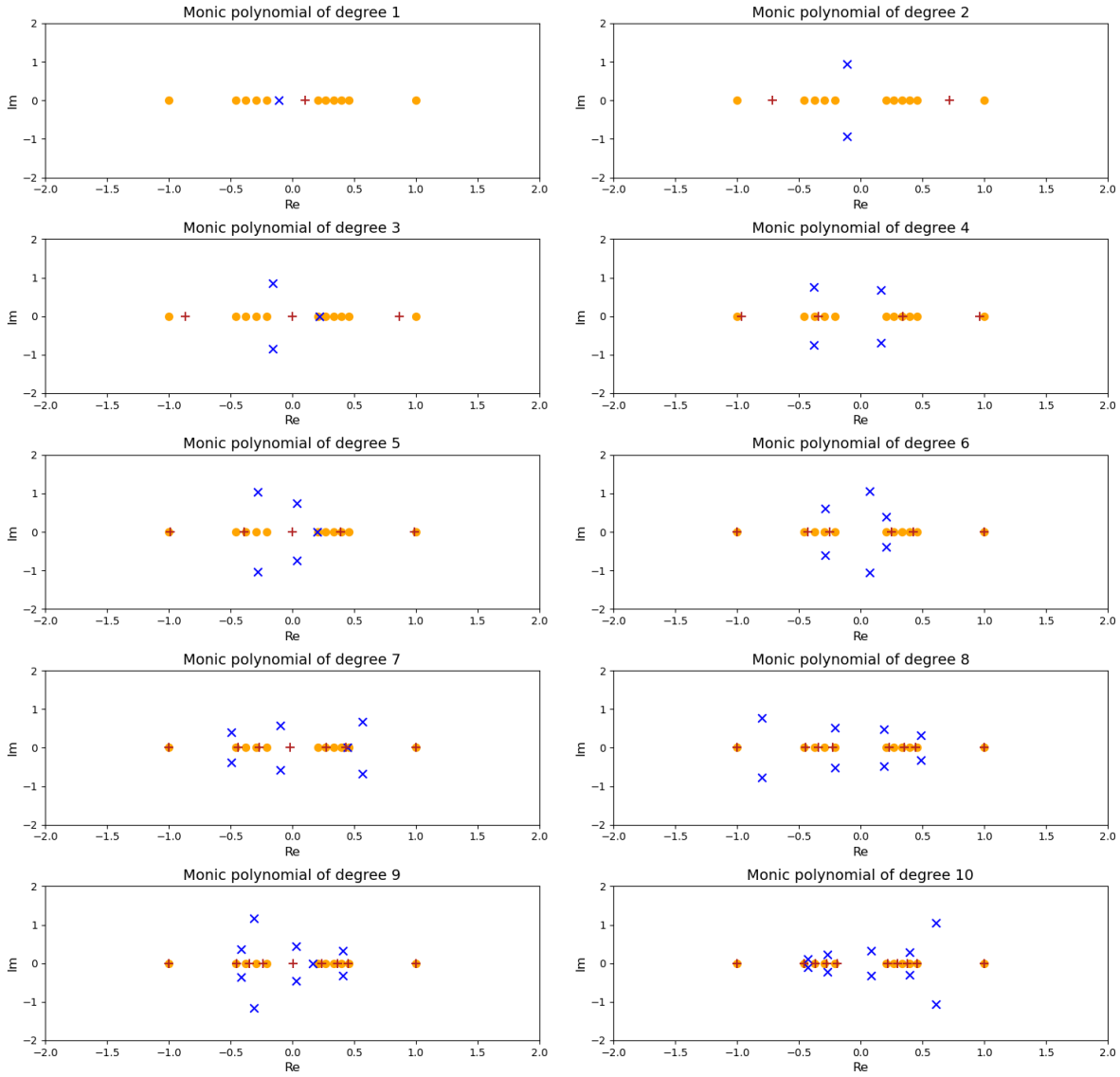
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.787	0.787	0.787	0.151	0.151	0.151
2	0.7	0.833	0.767	0.517	0.517	0.517
3	0.822	0.903	0.876	0.227	0.809	0.615
4	0.057	1.189	0.77	0.329	0.978	0.703
5	0.048	1.356	0.834	0.381	0.999	0.679
6	0.442	1.155	0.897	0.762	1.048	0.891
7	0.683	1.392	0.852	0.199	0.993	0.806
8	0.523	1.401	0.876	0.266	1.036	0.753
9	0.0	1.225	0.744	0.295	1.015	0.771
10	0.038	1.462	0.784	0.534	1.077	0.846
11	0.0	1.1	0.789	0.0	1.1	0.789

Figure B.1: Plot and table for different degrees of monic approximations of an 11×11 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts equidistantly placed on $[-1, 1]$ per cluster.



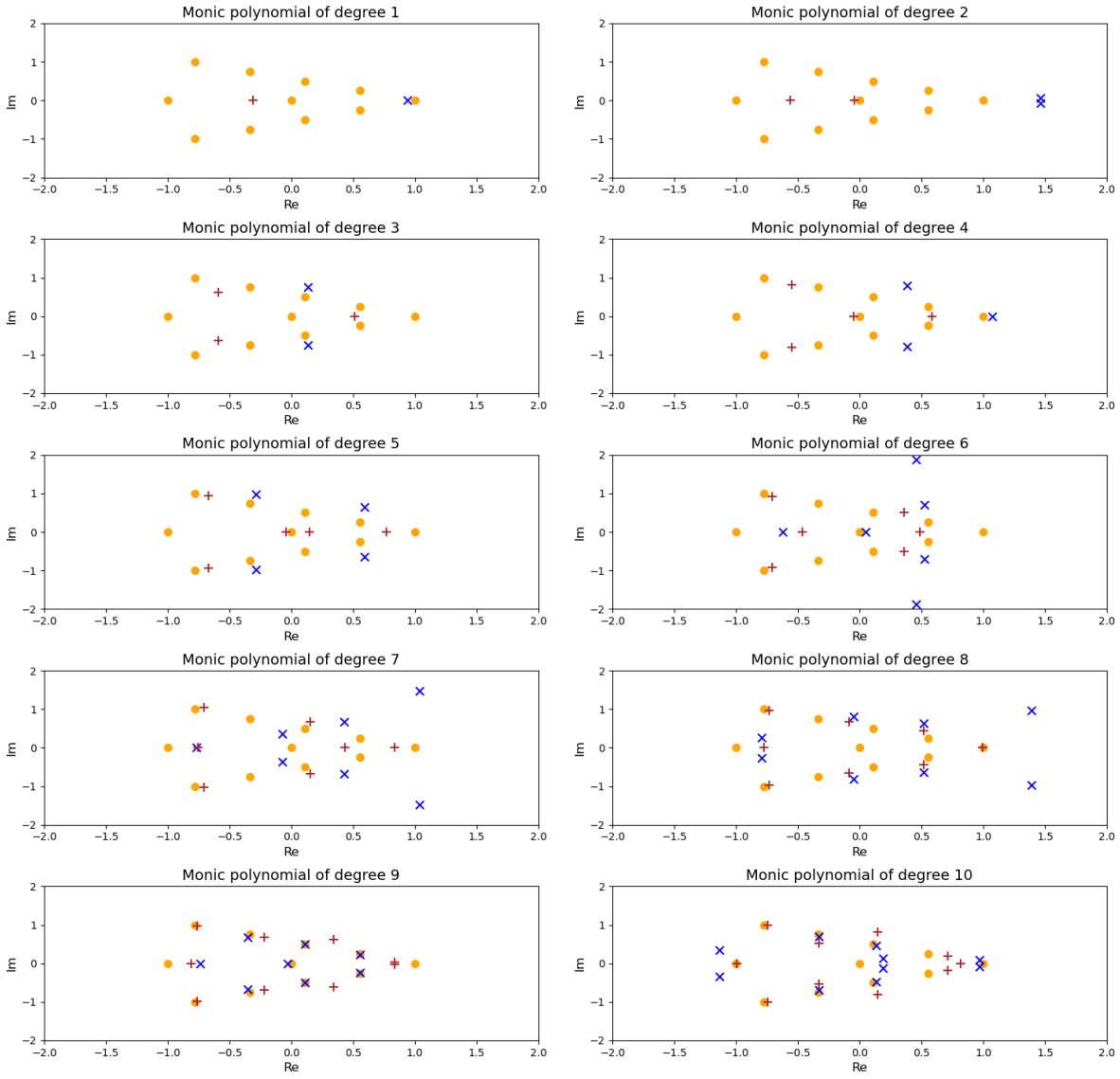
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.0	0.0	0.0	0.059	0.059	0.059
2	0.119	2.258	1.189	0.87	0.87	0.87
3	0.0	1.43	0.953	0.118	0.918	0.651
4	0.332	2.873	1.103	0.121	1.269	0.695
5	0.0	1.748	0.86	0.002	0.936	0.639
6	0.095	1.485	0.912	0.671	1.098	0.956
7	0.0	1.402	0.877	0.017	1.111	0.818
8	0.026	1.78	0.948	0.809	1.079	0.994
9	0.0	1.429	0.92	0.0	1.03	0.843
10	0.002	2.618	0.974	0.845	1.072	0.976
11	0.0	1.0	0.909	0.0	1.0	0.909

Figure B.2: Plot and table for different degrees of monic approximations of an 11×11 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts chosen as to place them on a circle.



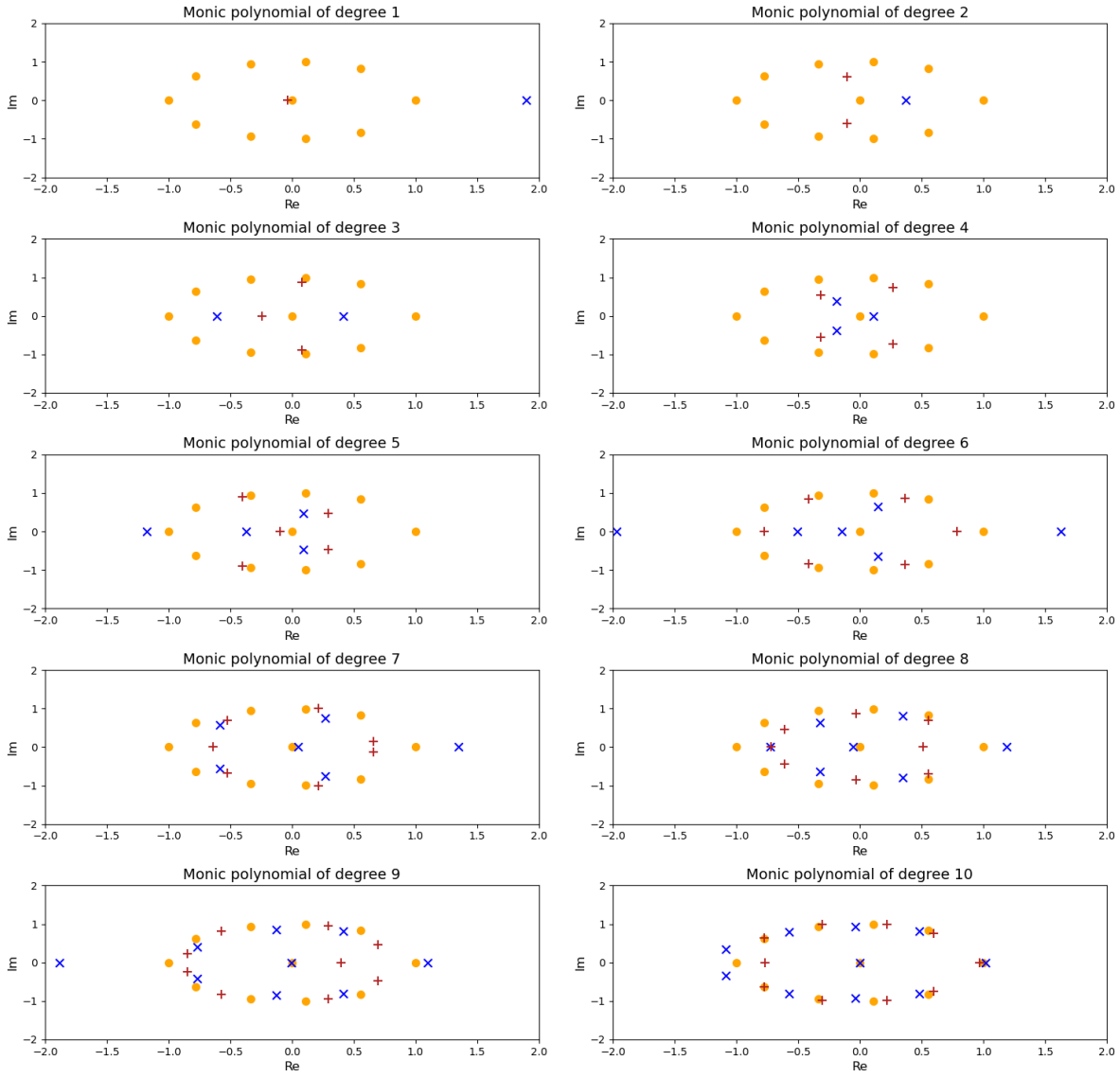
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.109	0.109	0.109	0.105	0.105	0.105
2	0.944	0.944	0.944	0.713	0.72	0.716
3	0.223	0.864	0.65	0.0	0.867	0.577
4	0.704	0.842	0.773	0.344	0.962	0.653
5	0.202	1.067	0.762	0.0	0.988	0.55
6	0.435	1.062	0.725	0.25	1.0	0.56
7	0.444	0.877	0.659	0.016	1.0	0.491
8	0.52	1.116	0.697	0.225	1.0	0.505
9	0.167	1.202	0.623	0.005	1.0	0.455
10	0.33	1.22	0.566	0.192	1.0	0.464
11	0.208	1.0	0.455	0.208	1.0	0.455

Figure B.3: Plot and table for different degrees of monic approximations of an 11×11 matrix with purely real eigenvalues in clusters.



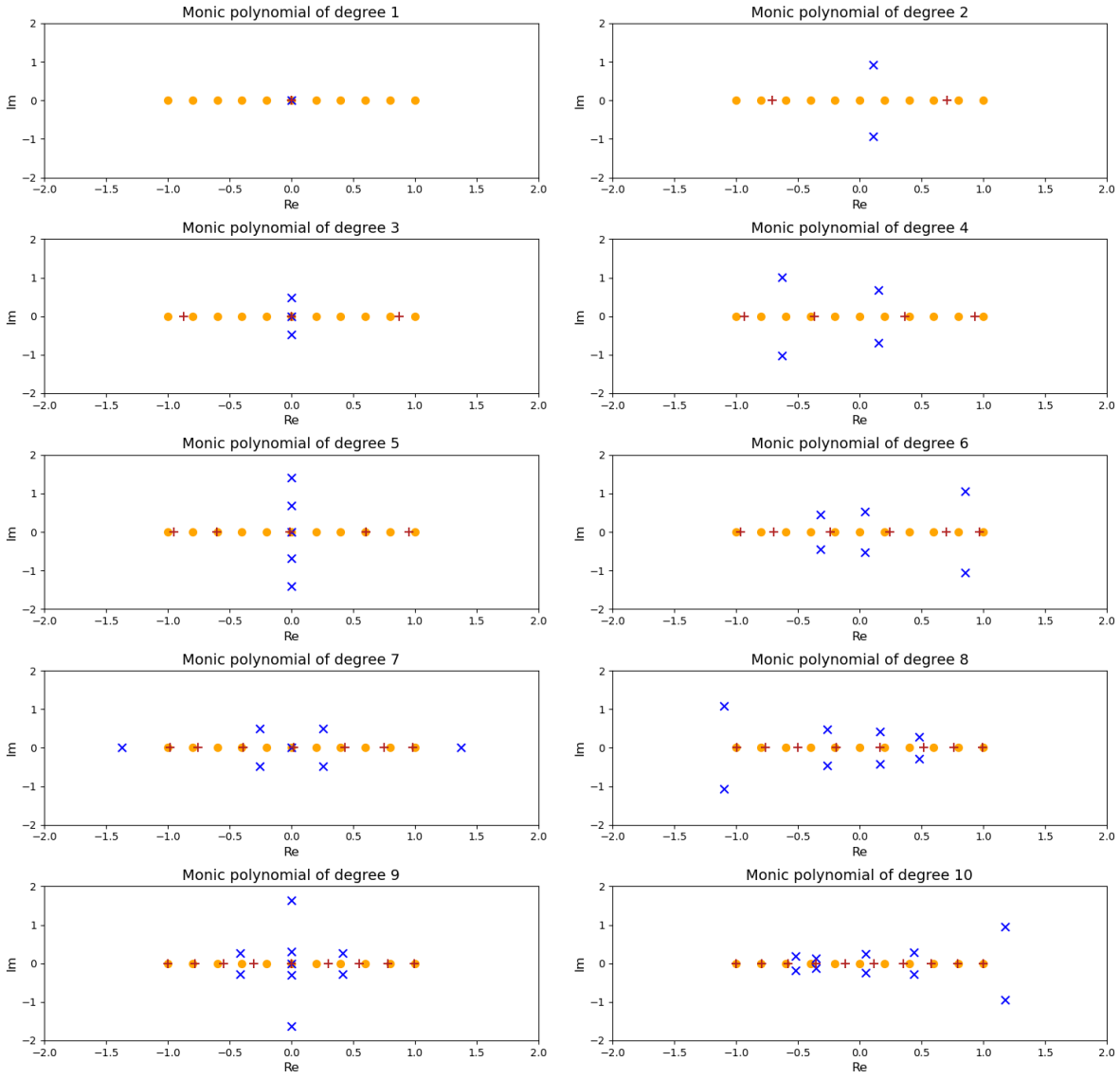
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.939	0.939	0.939	0.311	0.311	0.311
2	1.466	1.466	1.466	0.043	0.564	0.303
3	0.767	2.333	1.289	0.516	0.862	0.747
4	0.881	3.494	1.582	0.052	0.976	0.647
5	0.875	2.032	1.166	0.041	1.145	0.65
6	0.048	1.94	1.05	0.465	1.161	0.751
7	0.372	1.808	0.962	0.432	1.255	0.845
8	0.81	1.688	1.038	0.671	1.216	0.863
9	0.031	2.386	0.766	0.708	1.238	0.87
10	0.226	1.187	0.73	0.621	1.246	0.867
11	0.0	1.267	0.765	0.0	1.267	0.765

Figure B.4: Plot and table for different degrees of monic approximations of an 11×11 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts equidistantly placed on $[-1, 1]$.



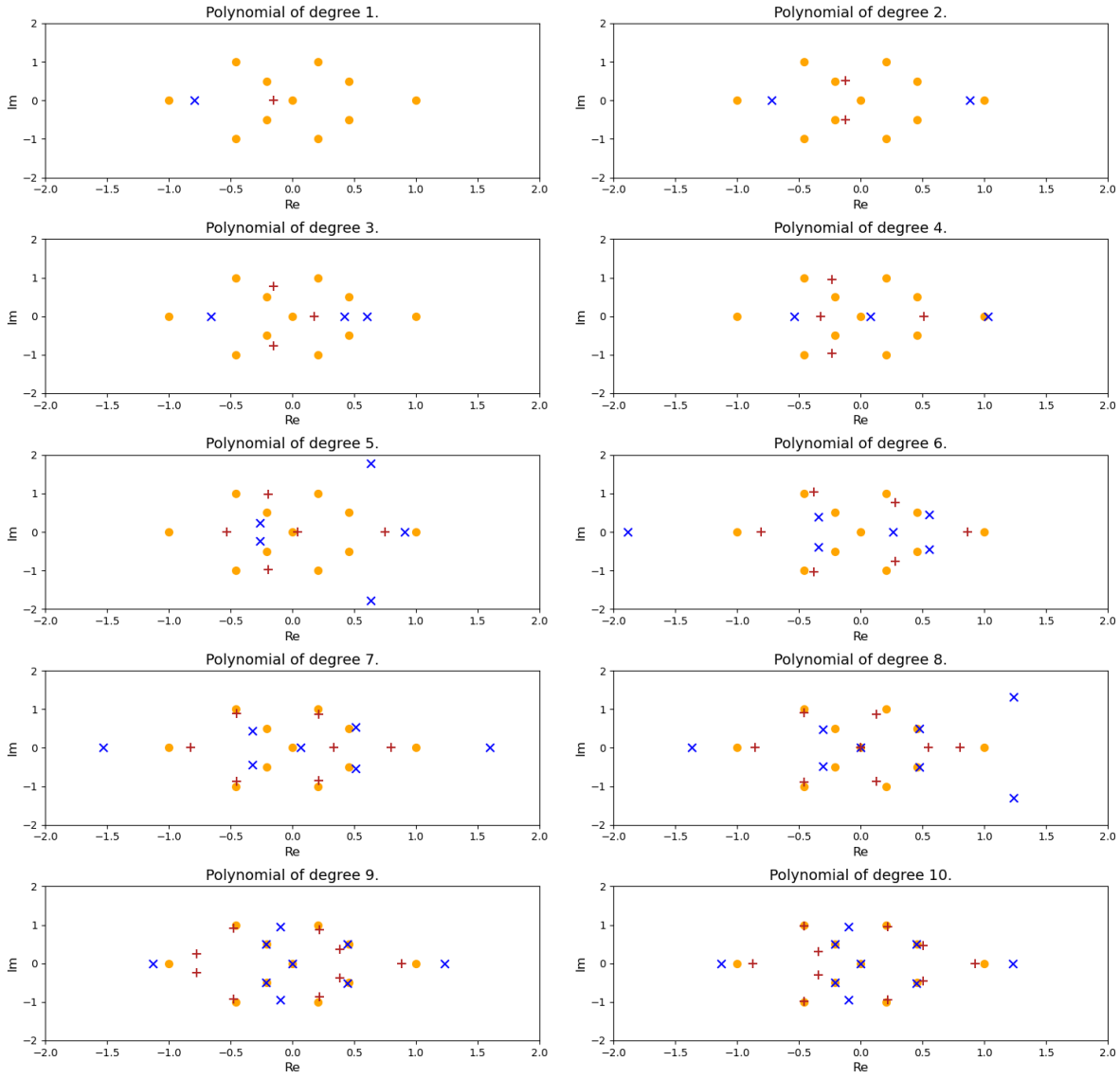
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	1.896	1.896	1.896	0.039	0.039	0.039
2	0.369	3.705	2.037	0.612	0.612	0.612
3	0.415	3.664	1.563	0.243	0.885	0.671
4	0.111	3.67	1.157	0.633	0.777	0.705
5	0.369	2.263	0.952	0.1	0.975	0.632
6	0.148	1.973	0.93	0.773	0.931	0.88
7	0.051	2.234	0.979	0.639	1.035	0.825
8	0.056	2.311	0.933	0.51	0.887	0.781
9	0.004	1.886	0.92	0.398	1.005	0.871
10	0.001	1.14	0.903	0.767	1.029	0.974
11	0.0	1.0	0.909	0.0	1.0	0.909

Figure B.5: Plot and table for different degrees of monic approximations of an 11×11 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts chosen as to place them on a circle.



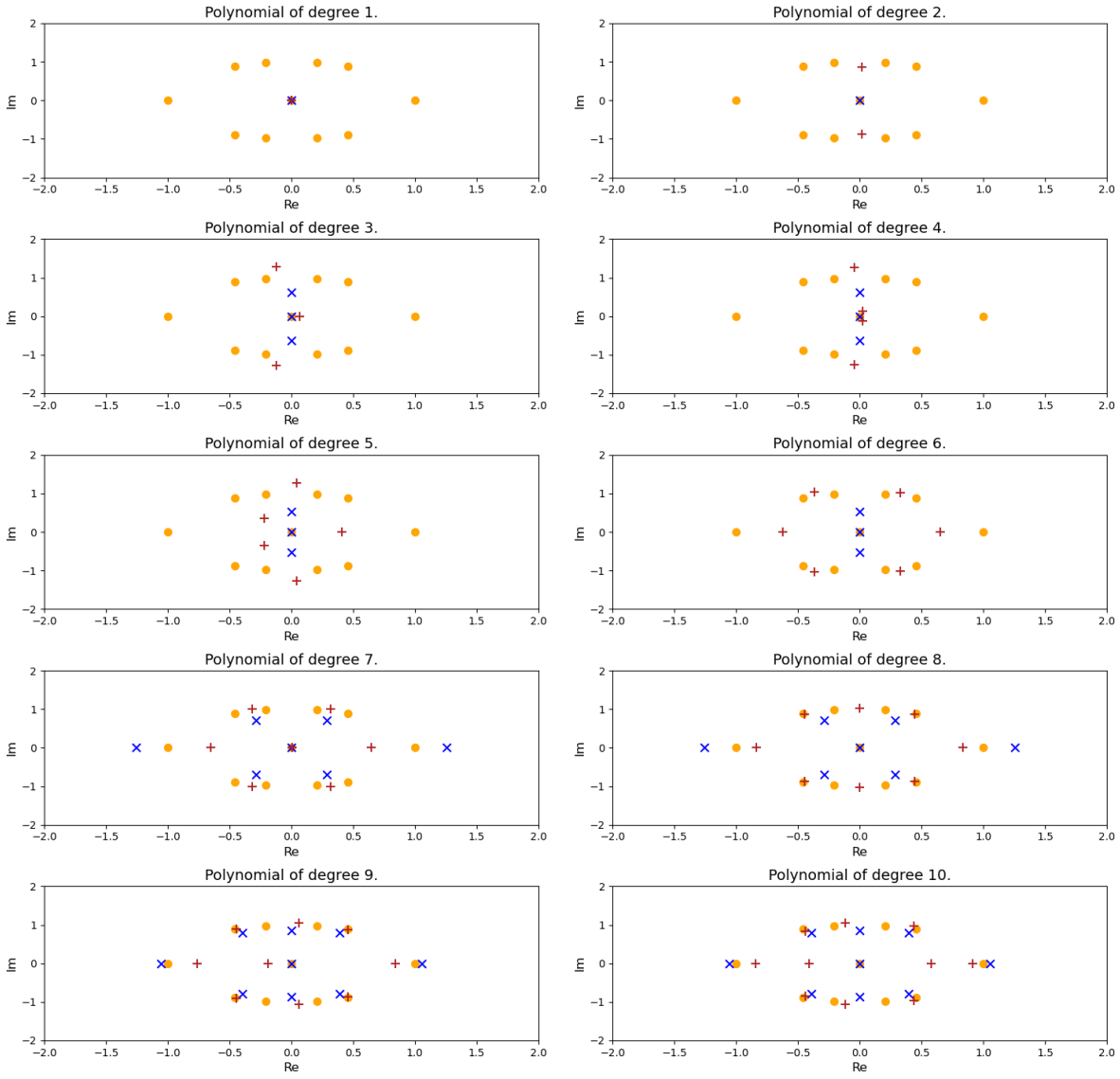
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.0	0.0	0.0	0.003	0.003	0.003
2	0.939	0.939	0.939	0.707	0.707	0.707
3	0.0	0.479	0.32	0.0	0.872	0.581
4	0.702	1.196	0.949	0.364	0.931	0.648
5	0.0	1.412	0.841	0.009	0.953	0.624
6	0.533	1.357	0.811	0.237	0.971	0.636
7	0.0	1.374	0.709	0.021	0.982	0.616
8	0.448	1.536	0.77	0.164	0.993	0.61
9	0.0	1.629	0.651	0.0	0.997	0.585
10	0.252	1.512	0.642	0.117	0.999	0.568
11	0.0	1.0	0.545	0.0	1.0	0.545

Figure B.6: Plot and table for different degrees of monic approximations of an 11×11 matrix with purely real eigenvalues equidistantly distributed on $[-1, 1]$.



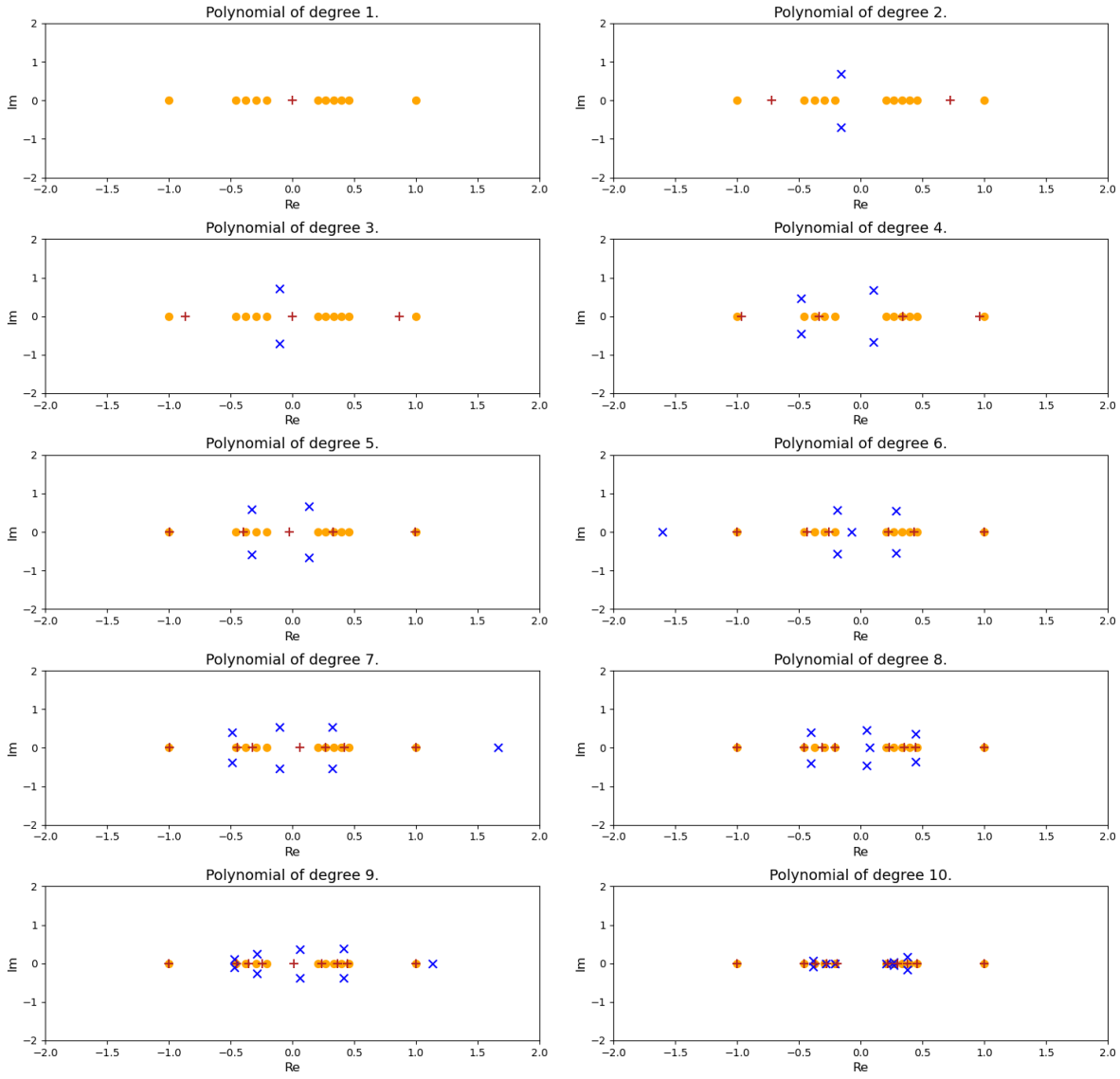
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.794	0.794	0.794	0.151	0.151	0.151
2	0.722	0.882	0.802	0.526	0.526	0.526
3	0.419	0.66	0.561	0.177	0.78	0.579
4	0.082	2.296	0.986	0.325	0.984	0.701
5	0.347	1.887	1.075	0.041	0.991	0.661
6	0.26	1.886	0.768	0.805	1.104	0.919
7	0.066	1.596	0.825	0.335	0.989	0.815
8	0.0	1.802	0.935	0.001	1.009	0.748
9	0.0	1.231	0.746	0.532	1.038	0.826
10	0.0	1472960635.155	147296064.187	0.454	1.081	0.818
11	0.0	1.1	0.789	0.0	1.1	0.789

Figure B.7: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts equidistantly placed on $[-1, 1]$ per cluster.



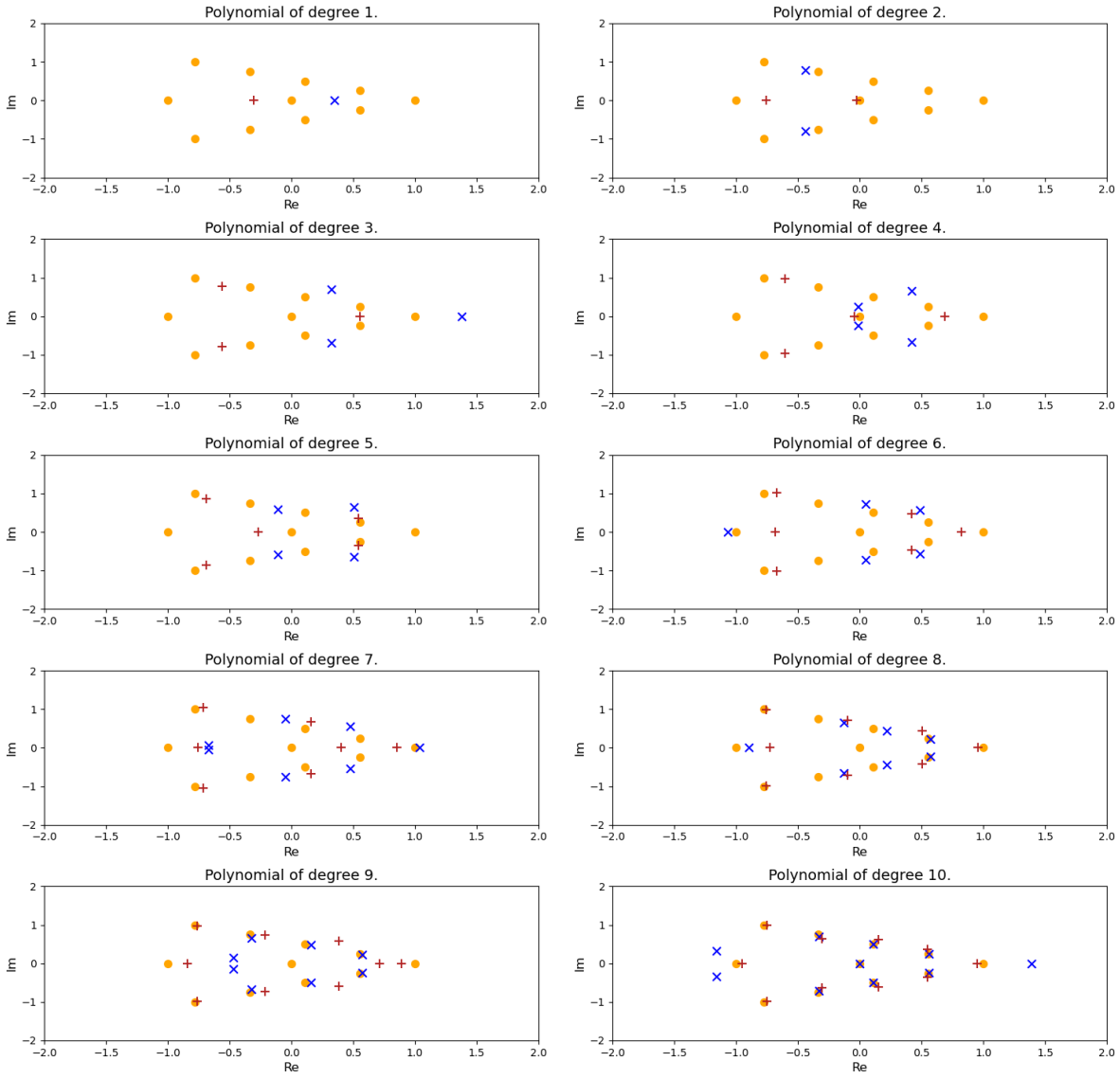
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	11073242.352	5536621.176	0.872	0.872	0.872
3	0.0	0.627	0.418	0.07	1.28	0.877
4	0.0	13132.05	3283.326	0.132	1.256	0.694
5	0.0	3.069	1.437	0.41	1.28	0.759
6	0.0	3494312.085	582386.545	0.621	1.097	0.932
7	0.0	1.256	0.794	0.009	1.053	0.789
8	0.0	60601.567	7575.891	0.836	1.028	0.952
9	0.0	1.055	0.819	0.191	1.054	0.875
10	0.0	3425675726.624	342567573.399	0.409	1.062	0.89
11	0.0	1.0	0.909	0.0	1.0	0.909

Figure B.8: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with eigenvalues having their real parts situated in clusters and their imaginary parts chosen as to place them on a circle.



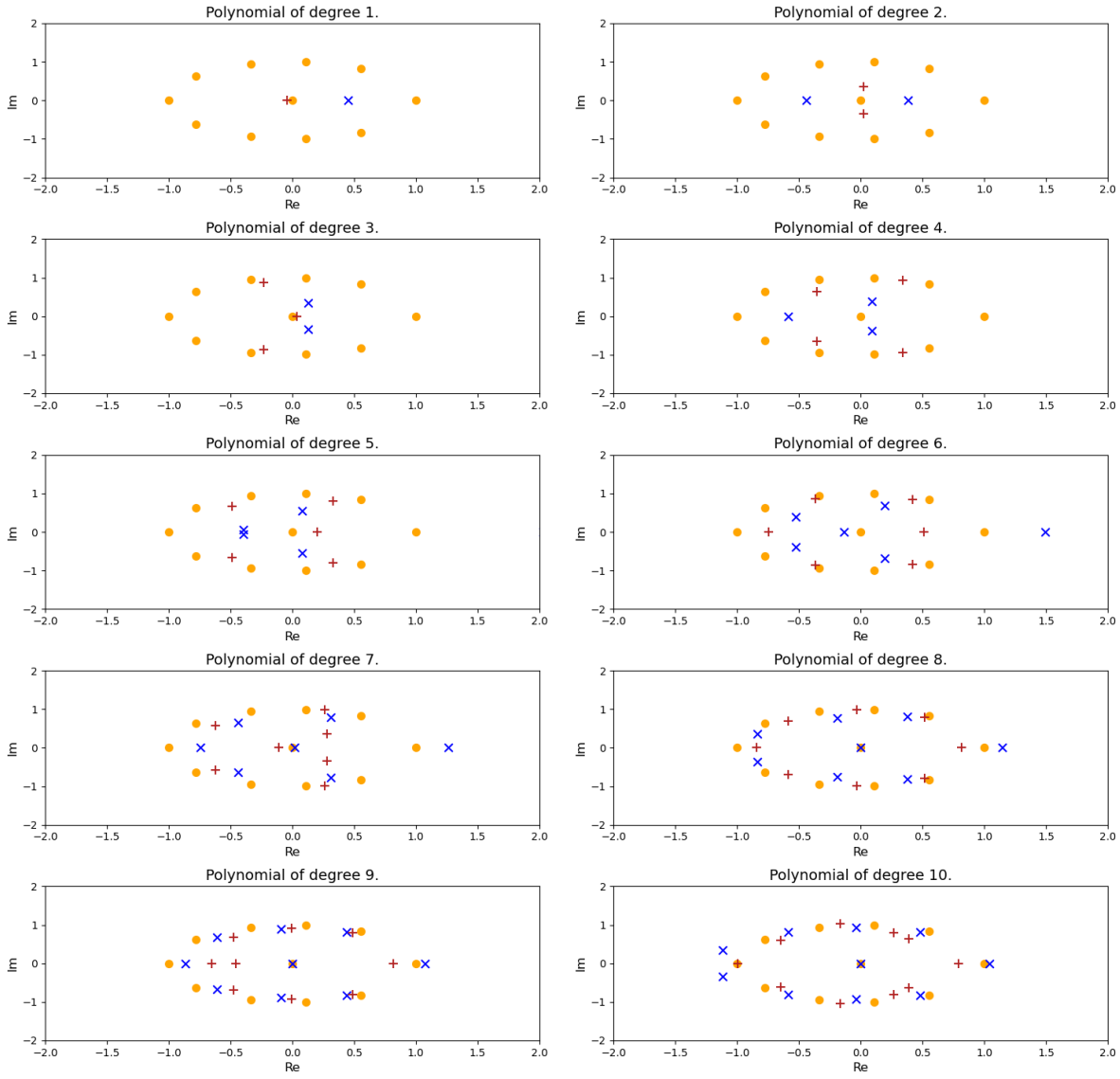
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	4.057	4.057	4.057	0.0	0.0	0.0
2	0.713	0.713	0.713	0.722	0.722	0.722
3	0.721	2.997	1.479	0.0	0.867	0.578
4	0.671	0.679	0.675	0.333	0.963	0.65
5	0.669	2.281	0.992	0.023	0.993	0.546
6	0.074	1.603	0.683	0.228	0.998	0.557
7	0.547	1.667	0.751	0.064	1.0	0.502
8	0.07	2.685	0.749	0.206	1.0	0.501
9	0.375	1.135	0.527	0.015	1.0	0.457
10	0.207	4.678	0.752	0.192	1.0	0.464
11	0.208	1.0	0.455	0.208	1.0	0.455

Figure B.9: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with purely real eigenvalues in clusters.



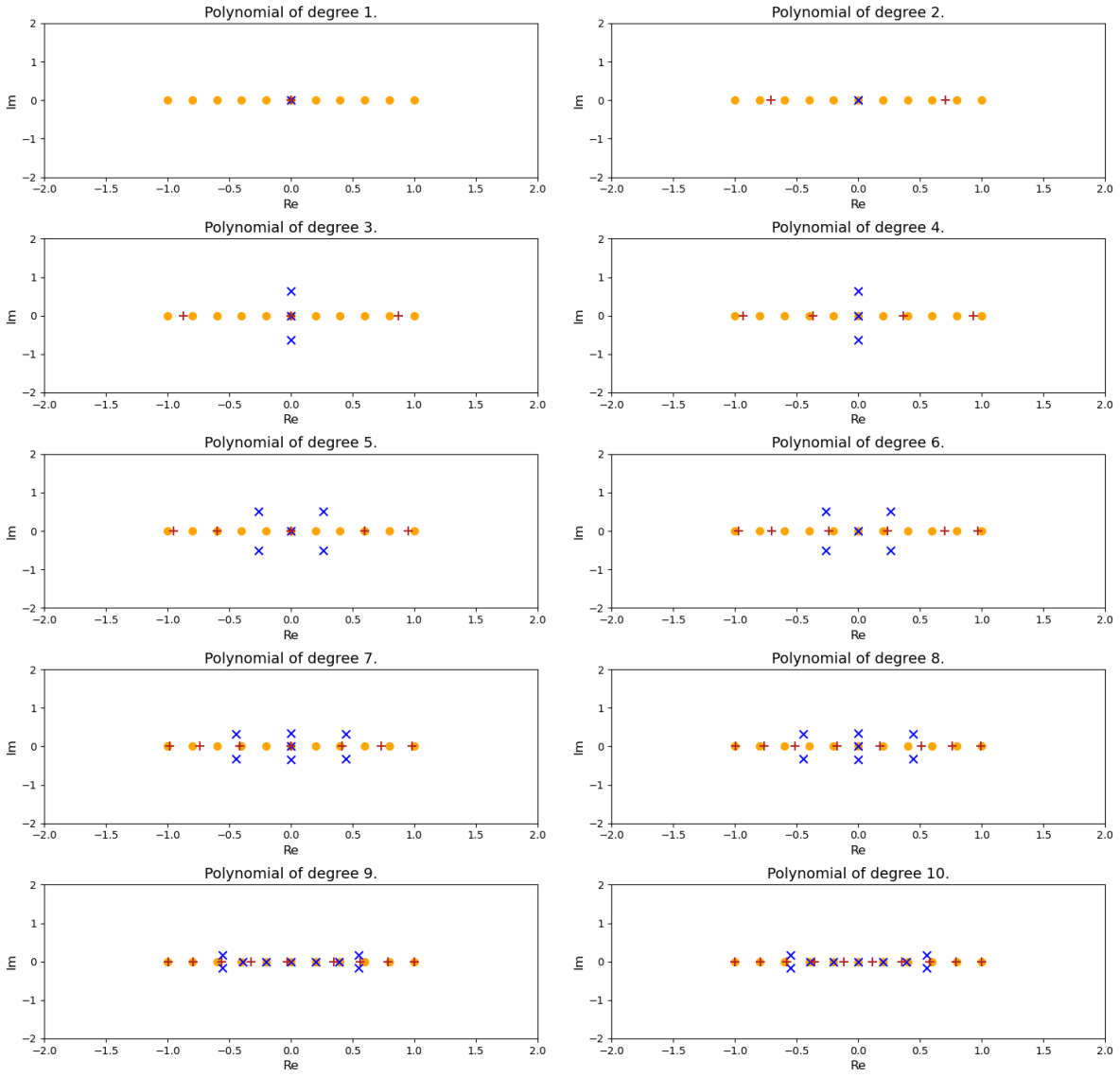
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.35	0.35	0.35	0.306	0.306	0.306
2	0.904	0.904	0.904	0.023	0.759	0.391
3	0.769	1.378	0.972	0.553	0.965	0.827
4	0.241	0.788	0.515	0.045	1.141	0.755
5	0.59	16.331	3.829	0.268	1.098	0.75
6	0.728	2.157	1.03	0.63	1.211	0.865
7	0.671	1.036	0.761	0.405	1.261	0.847
8	0.492	5.608	1.256	0.664	1.247	0.869
9	0.493	2.369	0.789	0.705	1.242	0.873
10	0.0	1.393	0.76	0.633	1.243	0.839
11	0.0	1.267	0.765	0.0	1.267	0.765

Figure B.10: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts equidistantly placed on $[-1, 1]$.



	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.451	0.451	0.451	0.045	0.045	0.045
2	0.383	0.442	0.413	0.357	0.357	0.357
3	0.361	2.052	0.925	0.036	0.895	0.609
4	0.387	5.414	1.693	0.737	0.996	0.867
5	0.4	2.031	0.786	0.201	0.858	0.714
6	0.137	1.493	0.73	0.51	0.941	0.833
7	0.017	1.266	0.753	0.112	1.024	0.681
8	0.002	1.148	0.792	0.816	0.984	0.918
9	0.002	1.074	0.822	0.459	0.934	0.813
10	0.0	1.167	0.915	0.747	1.048	0.884
11	0.0	1.0	0.909	0.0	1.0	0.909

Figure B.11: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with eigenvalues having their real parts equidistantly distributed on $[-1, 1]$ and their imaginary parts chosen as to place them on a circle.



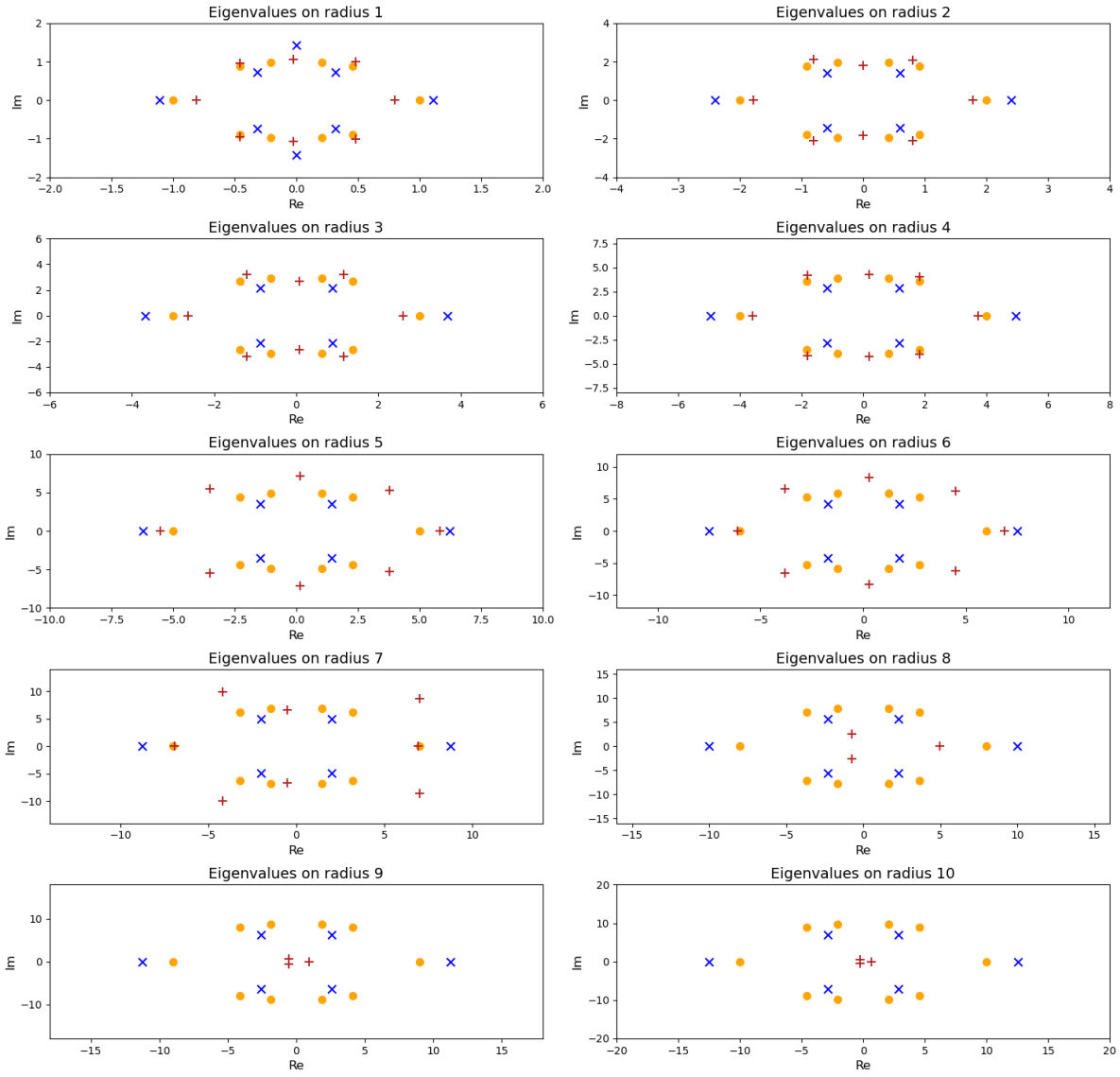
	Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Degree	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	16993916.707	8496958.354	0.707	0.707	0.707
3	0.0	0.641	0.427	0.0	0.872	0.581
4	0.001	1931.318	483.15	0.364	0.931	0.648
5	0.0	0.575	0.46	0.0	0.952	0.621
6	0.0	21723250.102	3620542.067	0.239	0.972	0.638
7	0.0	0.546	0.411	0.0	0.984	0.61
8	0.0	1009561.622	126195.563	0.174	0.992	0.61
9	0.0	0.577	0.387	0.033	1.0	0.6
10	0.0	8189895218.472	818989522.196	0.118	0.999	0.568
11	0.0	1.0	0.545	0.0	1.0	0.545

Figure B.12: Plot and table for different degrees of non-monic approximations of an 11×11 matrix with purely real eigenvalues equidistantly distributed on $[-1, 1]$.

Appendix C

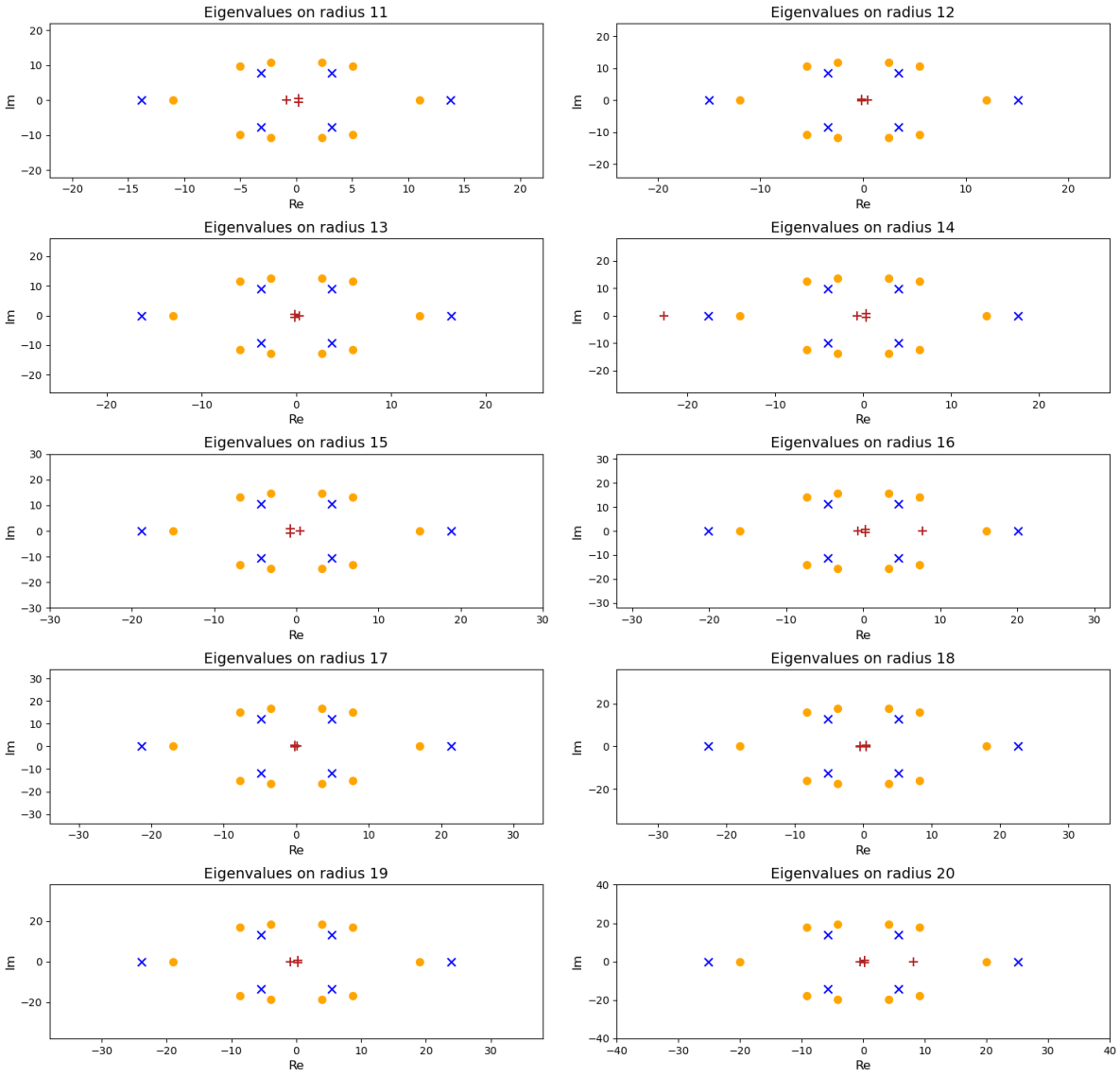
Comparison with different radii

This appendix contains the figures of eigenvalues of monic approximations made of matrices with eigenvalues with real parts clustered on $[-k, k]$ with $k = 1, \ldots, 20$, the complex parts were chosen as to put them on a circle. The true eigenvalues of the original matrix are represented by orange circles, the eigenvalues of the Arnoldi approximation as red crosses and our approximations using Tangs method are shown as blue plusses.



		Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Radius	Iterations	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
1	92	0.8	1.429	1.035	0.803	1.118	1.013
2	127	1.54	5.716	2.799	1.782	2.253	2.028
3	101	2.297	12.699	5.244	2.602	3.424	3.022
4	120	3.057	22.442	8.378	3.588	4.558	4.221
5	115	3.818	34.959	12.205	5.506	7.139	6.446
6	125	4.579	50.254	16.725	6.119	8.333	7.522
7	107	5.341	68.327	21.94	6.675	11.115	8.872
8	113	6.102	89.18	27.849	2.649	29.825	17.341
9	116	6.864	112.813	34.453	0.857	76.939	44.747
10	110	7.626	139.226	41.752	0.562	356.097	177.693

Figure C.1: Monic approximations of degree 8, approximating 10×10 matrices with clustered eigenvalues with complex part chosen as to situate them on a circles with different radii.



		Eigenvalue sizes, Tangs Method			Eigenvalue sizes, The Arnoldi Method		
Radius	Iterations	Least eig	Greatest eig	Average eig	Least eig	Greatest eig	Average eig
11	114	8.388	168.42	49.746	0.623	167.644	97.325
12	105	9.151	200.393	58.435	0.258	464.326	205.724
13	141	9.913	235.146	67.819	0.347	997.56	402.421
14	129	10.675	272.68	77.897	0.703	901.48	443.722
15	100	11.437	312.993	88.671	0.453	293.976	147.031
16	140	12.199	356.087	100.14	0.64	7848.051	3924.562
17	110	12.962	401.962	112.304	0.139	3625.865	1828.428
18	101	13.724	450.616	125.163	0.484	14050.797	5265.764
19	153	14.486	502.051	138.717	0.583	8126.497	3401.653
20	99	15.248	556.266	152.966	0.485	26020.661	9792.868

Figure C.2: Monic approximations of degree 8, approximating 10×10 matrices with clustered eigenvalues with complex part chosen as to situate them on a circles with different radii.

Appendix D

Original characterization theorem

In order to make this paper self contained we include the characterization theorem for approximations of real valued functions. Since the algorithm is well established, the theorems and proofs can be found in many different publications. In this paper we will refer to [5].

In order to prove the characterization theorem we will need the so called Kolmogorov criterion which is a corollary to the following theorem. It will show that if we have a minimum for a functional then the Gâteaux derivative, see Definition D.0.2, from that point in any direction will be positive.

Theorem D.0.1. *[5, p. 90] Let $\varphi : \mathcal{F} \to \mathbb{R}$ be a convex functional. Let the subset $\mathcal{K} \subset \mathcal{F}$ be convex and $u_0 \in \mathcal{K}$. Then the following statements are equivalent.*

1. $\varphi(u_0) = \inf_{u \in \mathcal{K}} \varphi(u)$.
2. $\varphi'_+(u_0, u - u_0) \geq 0$ *for all* $u \in \mathcal{K}$

φ'_+ is the Gâteaux derivative.

Definition D.0.2. *[5, p. 87] For a functional $\varphi : \mathcal{F} \to \mathbb{R}$,*

$$\varphi'_+(u, v) := \lim_{h \to 0^+} \frac{1}{h} \left(\varphi(u + hv) - \varphi(u)\right) \quad \text{for } u, v \in \mathcal{F}$$

is said to be the Gâteaux derivative of φ at u in direction v, provided that the limit on the right hand side in the equation exists.

The Kolmogorov criterion show us that for the Gâteaux derivative, see Definition D.0.2, of a norm is always positive when going from the difference function $s^* - f$ in the direction of another element $s - s^* \in \mathcal{S}$.

Corollary D.0.3 (Kolmogorov criterion)**.** *[5, p. 92] For $f \in \mathcal{F}$ and $s^* \in \mathcal{S}$, $\mathcal{S} \subset \mathcal{F}$ convex, the following statements are equivalent.*

1. *s^* is a best approximation to f.*

2. $\|\cdot\|'_+(s^* - f, s - s^*) \geq 0$ *for all* $s \in \mathcal{S}$

Looking now specifically at the Chebyshev norm 3.0.3, we will in the following theorem show that the Gâteaux derivative, see Definition D.0.2, from u to v will attain the greatest value of $v(x)$ with the sign of $u(x)$ at the extreme points of $|u(x)|$.

Theorem D.0.4. *[5, p. 95] Let* $\Omega \subset \mathbb{R}^d$ *be compact. Then, for the Gâteaux derivative of the maximum norm* $\|\cdot\| = \|\cdot\|_\infty$ *on* $\mathcal{C}(\Omega)$*, we have*

$$\|\cdot\|'_+(u, v) = \max_{\substack{x \in \Omega \\ |u(x)| = \|u\|_\infty}} v(x)\operatorname{sgn}(u(x))$$

for any $u, v \in \mathcal{C}(\Omega), u \not\equiv 0$

Using the previous theorem in conjecture with the Kolmogorov criterion we now arrive at the characterization theorem.

Theorem D.0.5. *[5, p. 140] Let* $\mathcal{S} \subset \mathcal{C}(\Omega)$ *be a linear subspace of* $\mathcal{C}(\Omega)$ *and suppose* $f \in \mathcal{C}(\Omega)$*. Then* $s^* \in \mathcal{S}$ *is a best approximation to* f *with respect to* $\|\cdot\|_\infty$*, if and only if*

$$\max_{\substack{x \in \Omega \\ |(s^* - f)(x)| = \|s^* - f\|_\infty}} (s - s^*)(x) \cdot \operatorname{sgn}((s^* - f)(x)) \geq 0, \quad \forall s \in \mathcal{S}.$$

As a consequence of Theorem D.0.5 we get the following Corollary. Note that at this point we take the step from the general best approximation $s^* \in \mathcal{S}$ to the best polynomial polynomial approximation $p^* \in \mathcal{P}_{n-1}$. The reason for this is that we are using the knot polynomial $\omega_{X^*}(x)$ in the proof.

Corollary D.0.6 (Alternation condition)**.** *[5, p. 142] Let* $[a, b]$ *be a compact subset to* $\mathbb{R}$ *and* $f \in \mathcal{C}$*. Then with* $n \in \mathbb{N}$ *there exists a unique best approximation,* $p^* \in \mathcal{P}_{n-1}$*, to* f *with respect to the norm* $\|\cdot\|_\infty$*. Also, there are at least* $n+1$ *ordered extremal points to* $(p^* - f)(x)$ *denoted* $\{x_0, \ldots, x_n\} \subset E_{p^* - f}$*, with the relation* $a \leq x_0 < \cdots < x_n \leq b$*, that are satisfying the alternation condition*

$$(p^* - f)(x_k) = (-1)^k \sigma \|p^* - f\|_\infty \qquad \text{for } k = 0, \ldots, n$$

for $\sigma \in \{-1, 1\}$.

Bachelor's Theses in Mathematical Sciences 2022:K16
ISSN 1654-6229

LUNFNA-4042-2022

Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lu.se/