FINDING ZEROS OF EXPONENTIAL SUMS

WANG YANHUA

Capstone Final Report for BSc (Honours) in

Mathematical, Computational and Statistical Sciences

Supervised by: David Smith

AY 2021/2022

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.

2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property (Yale-NUS HR 039).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

   ✓ Unrestricted access
   Make the Thesis immediately available for worldwide access.

   o Access restricted to Yale-NUS College for a limited period
   Make the Thesis immediately available for Yale-NUS College access only from ____________ (mm/yyyy) to ______________ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):
   ____.

   After this period, the Thesis will be made available for worldwide access.

   o Other restrictions: (please specify if any part of your thesis should be restricted)
   ____
   ____

Wang Yanhua Elm
Name & Residential College of Student

[signature]
Signature of Student

03 Apr 2022
Date

DAVID A SMITH
Name & Signature of Supervisor

01 APR 2022
Date

Acknowledgements

My heartfelt thanks goes to Prof. Dave Smith, for being an amazing supervisor who is always full of encouragement, introducing me to the joys of writing on a chalkboard, and being a source of inspiration.

Thank you to fellow Unified Transform Lab members Zhang Liu and Lee Juwon, whose previous work I built on;

Dr. Francesca Spagnuolo, who inspired me to do mathematics;

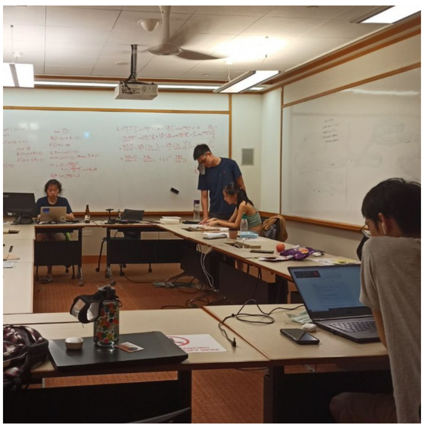
MCS juniors Naman Kedia, Stefan Roata, Koa Zhao Yuan and alumnus (!!) Luo Xinyu for taking time out of their weekends to be superb, eagle-eyed reviewers, and for being so forthcoming with their thoughtful insights;

Zhao Yuan again for giving me valuable advice on making open source software;

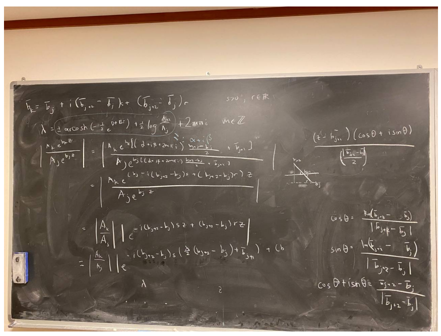
The capstone support group that struggled together: Linda, Mark, Ziting, Dani, and Zhang Liu, for filling the many capstone hours with joy and laughter;

My dearest friends, for your love, generosity, and care;

My family, who has been there for me since day one.



despair and chaos
(photo by linda)



what my photo album looked
like for the past semester

YALE-NUS COLLEGE

Abstract

B.Sc (Hons)

Finding Zeros of Exponential Sums

by Wang Yanhua

In this paper, I present analytic formulae for the zeros of some cases of exponential sums, and a complex root finder algorithm implemented in a Julia package.

In general, there is no analytic formula for the zeros of exponential sums. However, their zeros correspond to the eigenvalues of differential operators, so finding the zeros with good accuracy is crucial to solving differential equations.

In the analytical part, I derived formulae for asymptotic loci of zeros based on the theory of Langer. I showed that if an exponential sum satisfies certain properties, then we can obtain asymptotic formulae for *all* zeros with the modulus being sufficiently large.

In the numerical part, I designed a tail-recursive subdivision algorithm that finds approximations of zeros of an analytic function within a given rectangular domain on the complex plane. I implemented the algorithm in an open-source package in Julia, and designed unit tests and randomised testing to evaluate its effectiveness.

Keywords: exponential sum, Langer, subdivision algorithm

Contents

Chapter 1

Introduction

The goal of this paper is simple. Given an exponential sum of the form

$$f(z) = \sum_{j=1}^{n} A_j e^{b_j z}, \tag{1.1}$$

where $A_j, b_j \in \mathbb{C}$, we want to know where its zeros are.

Exponential sums arise in the study of monomial symbol differential operators. Their zeros correspond to the eigenvalues of the differential operator, so finding the zeros with good accuracy is crucial to solving differential equations. Solving differential equations is important as they model many processes and systems, such as heat transfer and water waves.

For the simplest examples of exponential sums, such as sine and cosine, it is easy to find the zeros exactly, but this is not possible in general.

There are two parts to my project. In the first part, I derive an analytical method, based on the theory of Langer, to find the asymptotic locus of zeros.

The second part is numerical. I will be using the argument principle, a theorem in complex analysis. The argument principle can be used to find the number of zeros of a given exponential sum that are contained within a given finite region. By successively applying this theorem to smaller regions, I numerically locate the zeros of exponential sums within a given region. The algorithm is available to use as part of an open-source Julia package, FindComplexZeros [4].

Chapter 2

Preliminaries

Exponential sums arise in monomial symbol differential problems, such as Sturm-Liouville problems.

Birkhoff first developed the asymptotic character of the solutions of eigenvalue problems for high order linear differential operators [1]. If the operator has monomial symbol, this characterisation reduces to studying the locus of zeros of exponential sums.

In general, there is no analytic formula for the zeros of an exponential sum. The fact that we have it for sine and cosine is very special.

Many have calculated the zeros for specific exponential sums as part of their research. For example, Pelloni derived the locus of zeroes of $F_0(z) = e^z + e^{\alpha z} + e^{\alpha^2 z}$, where $\alpha = e^{\frac{2\pi i}{3}}$, and also the zeros of its derivatives $F_1(z), F_2(z)$ [8][Proposition A.1].

However, even the apparently simple $F_0(z)$ studied by Pelloni has no analytic formula. Pelloni only proved that they lie exactly on certain rays and found an asymptotic formula for their distribution along these rays. So, in most cases, we can only provide asymptotic formulae (large $|z|$) or numerical approximations for the locus of zeros.

Nevertheless, in Chapter 3, we provide an analytic formula that covers some cases of (1.1).

2.1 Summary and discussion of Langer's method

2.1.1 Real commensurable exponents

If the exponents b_j of (1.1) are real and commensurable, Langer found an explicit formula for the distribition of zeros [7][Theorem 1].

In this case, the exponential sum is of the form

$$\phi(z) = \sum_{j=0}^{n} A_j (e^{az})^{p_j}, p_0 = 0, \tag{2.1}$$

which is a polynomial of degree p_n in the quantity e^{az}. If this polynomial admits as zeros the values $\zeta_1, \zeta_2, \cdots, \zeta_{p_n}$, the function (2.1) vanishes if and only if $e^{az} = \zeta_j$.

The zeros of (2.1) are therefore given by the formula

$$z = \frac{1}{a}\{2m\pi i + \log \zeta_j\},$$

where $j = 1, 2, \cdots, p_n$, and $m = 0, \pm 1, \pm 2, \cdots$.

They are infinite and distributed in the complex plane at regular intervals of length $\frac{2\pi}{a}$ along p_n lines which are vertical, or normal to the axis of reals.

Example 2.1.1. Consider the exponential sum with real commensurable exponents

$$f(z) = (1+i)e^{z} + 2ie^{2z} + (2+3i)e^{-z}. \tag{2.2}$$

By looking at the complex phase portrait of (2.2), we can see that the zeros indeed lie on vertical lines.

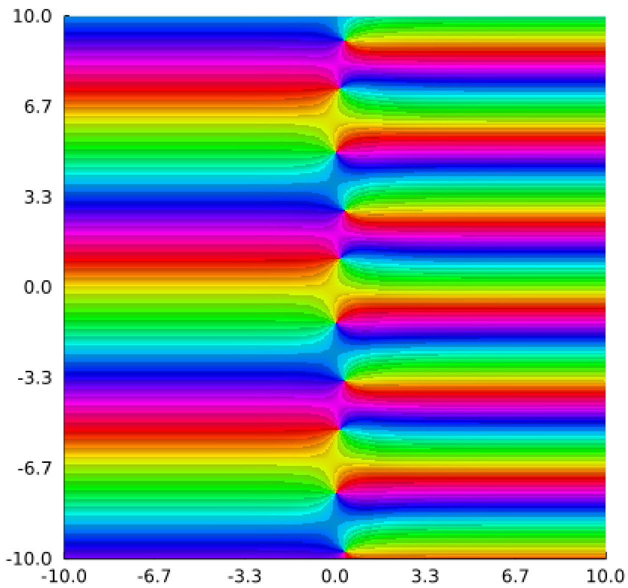


FIGURE 2.1: Complex phase portrait of (2.2)

Note on complex phase portraits: In a complex phase portrait, the phase of $f(z)$ is colour-coded, and the zeros are the points of discontinuity. Please refer to Appendix A to find out more.

In this paper, the portraits with axes labels are generated using my fork of the Julia package ComplexPortraits [3]. This is because the original axes labels are inverted, and at the time of writing, my pull request to fix it has not been merged.

The portraits without axes labels are generated using the Julia package ComplexPhasePortrait [2].

2.1.2 General real exponents

If the exponents are general real constants, Langer found that the zeros are confined to a vertical strip in the complex plane, and gave a limit to the number of zeros in any portion of the strip [7][Theorem 3]. However, he did not provide a formula for the zeros.

Example 2.1.2. Consider the exponential sum

$$f(z) = (1+i)e^{2\pi z} + 2ie^{2z} + (2+3i)e^{-z}. \tag{2.3}$$

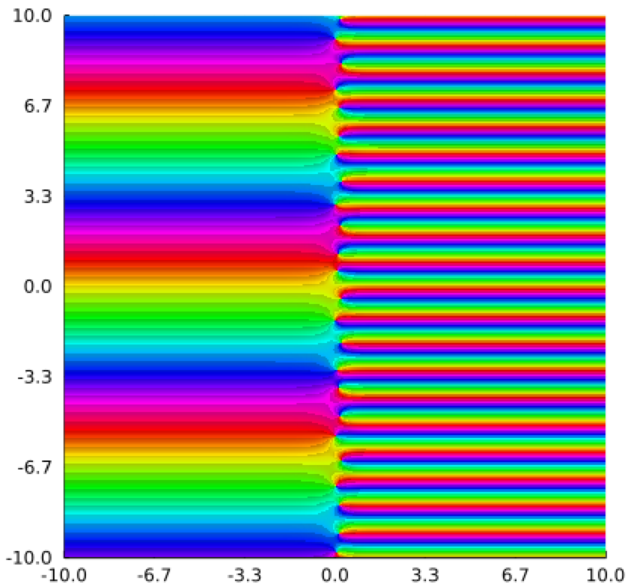


FIGURE 2.2: Complex phase portrait of (2.3)

2.1.3 General complex exponents

Allowing b_j to be complex, Langer located the zeros of large $|z|$ in strips on the complex plane [7][Theorem 8], by using a geometric method. Following Langer's approach, given an exponential sum (1.1), we plot the set $\overline{b_j}$

(complex conjugates of the exponents) in the complex plane.

Construct the polygon P which (i), is convex, (ii), has vertices only at points of the set, and (iii), includes all points of the set either in its interior or on its perimeter. We call P the *convex polygon hull* of f.

Langer proves the following theorem:

Theorem 2.1.3. The zeros of (1.1) are confined for $|z| > M$ to a finite number of strips each of asymptotically constant width. These strips are associated in groups with the exterior normals to the sides of P, and approach parallelism with the respective normals [7].

Example 2.1.4. Consider the exponential sum

$$f(z) = (1+i)e^{iz} + (3.5+2i)e^{z} + (3-i) + (2+3i)e^{-z}. \tag{2.4}$$

We notice that we get a triangle with vertices at $1, -1, -i$ when we plot the the convex polygon hull using the conjugates of the exponents. By looking at the complex phase portrait of (2.4), we can visually confirm Theorem 2.1.3.

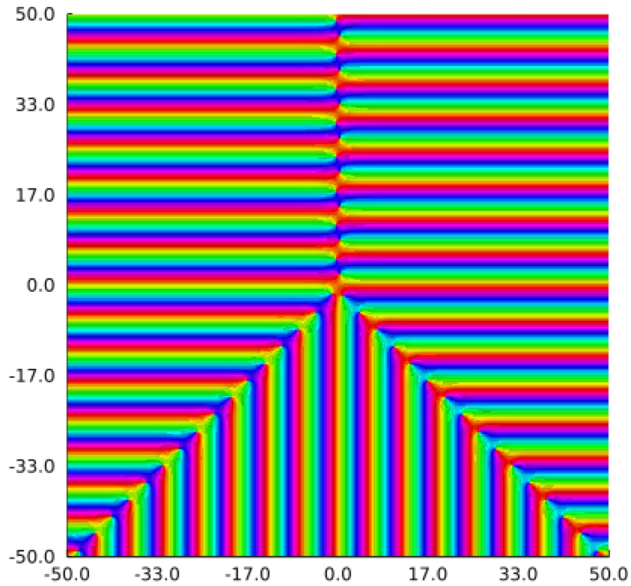


FIGURE 2.3: Complex phase portrait of (2.4)

Furthermore, the zeros associated asymptotically with each side of P come from the exponential sum arising from each side.

Hence, the asymptotic locus of zeros of (1.1) is the union of zeros from the exponential sum arising from each side of the polygon.

2.1.4 Discussion

In conclusion, Langer does not provide a specific analytic solution for (1.1). Moreover, because Langer studies functions more general than (1.1), his asymptotic information is less precise than what we shall derive for exponential sum (1.1). As we shall argue, the locus of zeros (asymptotically) on rays, like in Figure 2.3, may often be extablished for exponential sums, improving on Langer's "strips" characterisation.

Chapter 3

Analytical Method

In this chapter, we will take a look at how to obtain the zeros analytically. I will be deriving the asymptotic locus of zeros for different cases of exponential sums, based on Langer's method.

First, we prove a lemma about convex polygons that will be useful later on.

Lemma 3.0.1. Suppose $c_1, c_2, \cdots, c_n \in \mathbb{C}$ form a convex polygon on the complex plane, anti-clockwise, and let $c_{n+1} = c_1$. For $k, j \in \{1, 2, \cdots, n\}$, where $k \notin \{j, j+1\}$,

$$c_k = c_j + (c_{j+1} - c_j)(r + is)$$

for $r, s \in \mathbb{R}, s \geq 0$. Moreover, if c_k, c_j, c_{j+1} are not collinear, then s > 0.

Proof. Let α be a vector in the direction from c_j to c_{j+1}, and β is α rotated 90 degrees anti-clockwise. Then $\alpha = r\left(c_{j+1} - c_j\right)$, $\beta = se^{i\frac{\pi}{2}}\left(c_{j+1} - c_j\right)$, $\exists r, s \in \mathbb{R}$. Any point in the plane may be written as

$$c_j + \alpha + \beta \tag{3.1}$$

by the orthogonality of α, β. But, if the polygon is convex, then c_k must lie on or to the left of the line that passes through c_j in the direction of α.

Therefore, $s \geq 0$. If c_k cannot lie on that line, then $s > 0$. Equation (3.1) may be re-expressed as

$$\begin{aligned} c_k &= c_j + \alpha + \beta \\ &= c_j + r\left(c_{j+1} - c_j\right) + s e^{i\frac{\pi}{2}}\left(c_{j+1} - c_j\right) \\ &= c_j + \left(c_{j+1} - c_j\right)(r + is). \end{aligned}$$

□

3.1 Side of polygon with 2 points as corners

The special case of noncollinear c_k, c_j, c_{j+1} suggests that a side of the convex polygon hull of (1.1) in which there is no third exponent $\overline{b_k}$ is worthy of special attention. In this section, we study that case.

Firstly, we obtain the zeros of an exponential sum with 2 terms, which is fairly straightforward.

Proposition 3.1.1. For $A, a, B, b \in \mathbb{C}$, $A, B \neq 0$, the zeros of $Ae^{az} + Be^{bz}$ lie on the line $z = \frac{i\pi(2k+1)+\log\frac{B}{A}}{a-b}$, $k \in \mathbb{Z}$.

Proof. The zeros of cosh are at odd integer multiples of $\frac{i\pi}{2}$. We calculate

$$\begin{aligned} 0 = Ae^{az} + Be^{bz} &= A\left(e^{az} + \frac{B}{A}e^{bz}\right) \\ &= A\left(e^{az} + e^{bz+\log\frac{B}{A}}\right) \\ &= Ae^{\frac{a+b}{2}z+\frac{1}{2}\log\frac{B}{A}}\left(e^{\frac{a-b}{2}z-\frac{1}{2}\log\frac{B}{A}} + e^{-\left(\frac{a-b}{2}z-\frac{1}{2}\log\frac{B}{A}\right)}\right) \\ &= Ae^{\frac{a+b}{2}z+\frac{1}{2}\log\frac{B}{A}}2\cosh\left(\frac{a-b}{2}z - \frac{1}{2}\log\frac{B}{A}\right) \\ \iff \quad \frac{a-b}{2}z - \frac{1}{2}\log\frac{B}{A} &= \frac{i\pi(2k+1)}{2}, k \in \mathbb{Z} \\ \iff \quad z &= \frac{i\pi(2k+1) + \log\frac{B}{A}}{a-b}. \end{aligned}$$

□

As the proof shows, finding the zeros of an exponential sum with 2 terms can be reduced to finding the zeros of cosh, which is a much simpler problem. We will use Proposition 3.1.1 as a lemma to investigate more complicated exponential sums.

Theorem 3.1.2. Suppose $\overline{b_1}, \overline{b_2}, \cdots, \overline{b_n} \in \mathbb{C}$ is the convex polygon hull of

$$f(z) = \sum_{k=1}^{N} A_k e^{b_k z},$$

anti-clockwise, $n \leq N$, and $\overline{b_j}, \overline{b_{j+1}}$ are the corners of one side.

Then, the zeros of $f(z)$ arising from that particular side lie asymptotically on the ray

$$z = \frac{i\pi(2m+1) + \log \frac{B}{A}}{a-b},$$

$m \in \mathbb{N}^0$.

Proof. For the side formed by $\overline{b_j}, \overline{b_{j+1}}$, the zeros of $A_j e^{b_j z} + A_{j+1} e^{b_{j+1} z}$ are given by $z_m = \frac{i\pi(2m+1) + \log \frac{A_{j+1}}{A_j}}{b_j - b_{j+1}}$, $m \in \mathbb{Z}$, from Proposition 3.1.1.

We claim that for large $|m|$, the $m > 0$ corresponds to zeros of f. The $m << 0$ zeros of $A_j e^{b_j z} + A_{j+1} e^{b_{j+1} z}$ are irrelevant because, for $m << 0$, other terms $A_k e^{b_k z}$ dominate both $A_j e^{b_j z}$ and $A_{j+1} e^{b_{j+1} z}$.

Suppose $k \notin \{j, j+1\}$. We want to check the behaviour of $\left| \frac{A_k e^{b_k z}}{A_j e^{b_j z}} \right|$ for $m \to \infty$, as well as for $m \to -\infty$. If this ratio is decaying to $-\infty$ as $m \to \infty$, it means that all other terms are decaying relative to $A_j e^{b_j z} + A_{j+1} e^{b_{j+1} z}$ since k is arbitrary. The same applies to $m \to -\infty$. Hence, this will tell us whether the equation of the ray uses $m \to -\infty$ or $m \to \infty$.

We note from Lemma 3.0.1 that

$$\overline{b_k} = \overline{b_j} + (\overline{b_{j+1}} - \overline{b_j})(r + is)$$

for $r, s \in \mathbb{R}, s > 0$. Hence, we have

$$\left|e^{b_k z}\right| = \left|e^{b_j z}\right| \left|e^{[(b_{j+1}-b_j)(r-is)]z}\right|.$$

Substituting the above, we get

$$\begin{aligned}
\left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right| &= \left|\frac{A_k}{A_j}\right| \left|e^{[(b_{j+1}-b_j)(r-is)]z}\right| \\
&= \left|\frac{A_k}{A_j}\right| \left|e^{[(b_{j+1}-b_j)(r-is)]\frac{i\pi(2m+1)+\log\frac{A_{j+1}}{A_j}}{b_j-b_{j+1}}}\right| \\
&= \left|\frac{A_k}{A_j}\right| \left|e^{-(r-is)\left[i\pi(2m+1)+\log\frac{A_{j+1}}{A_j}\right]}\right| \\
&= \left|\frac{A_k}{A_j}\right| \left|e^{(r-is)\log(\frac{A_j}{A_{j+1}})-2mr\pi i-r\pi i-(2m+1)s}\right|
\end{aligned}$$

Because $s > 0$, $(2m+1) \to \infty$ implies $\left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right| \to 0$, and $(2m+1) \to -\infty$ causes $\left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right| \to \infty$. Hence, $(2m+1) > 0$, which implies $m \in \mathbb{N}^0$. $\square$

3.2 Side of polygon with 3 collinear points

Next, we look at the case of 3 equally spaced collinear points on one side. That means that 2 points form the corners, and the third point is exactly at the center. This situation frequently comes up. For example, it occurs in the boundary value problem

$$\begin{aligned}
y^{(4)}x &= \lambda y, \\
y(0) &= 0 = y(1), \\
y'(0) &= 0, \\
y''(0) - y'(1) &= 0.
\end{aligned}$$

For a worked example of this problem to understand why this creates a convex polygon hull, with sides with 3 equally spaced collinear points, refer to Appendix B.

Firstly, we prove a lemma about the zeros of

$$f(z) = Ae^{az} + Be^{bz} + Ce^{cz}, \tag{3.2}$$

where $A, B, C, a, c \in \mathbb{C}, A, B, C \neq 0$ and $b = \frac{a+c}{2}$, i.e. b is the midpoint of a and c.

If we can obtain an analytic solution to 3.2, it means that we can try obtaining an asymptotic locus of zeros arising from a side with 3 equally spaced collinear points, in a similar way to how we used Proposition 3.1.1 to prove Theorem 3.1.2.

Lemma 3.2.1. The zeros of 3.2 lie on the lines

$$z = \frac{2}{a-c}\left(\pm \operatorname{arcosh}\left(-\frac{B}{2A}e^{-\frac{1}{2}\log\frac{C}{A}}\right) + \frac{1}{2}\log\frac{C}{A} + 2m\pi i\right)$$

for $m \in \mathbb{Z}$.

Proof. We calculate

$$\begin{aligned}
0 &= Ae^{az} + Be^{\frac{a+c}{2}z} + Ce^{cz} \\
&= Ae^{\frac{a+c}{2}z+\frac{1}{2}\log\frac{C}{A}}\left(e^{\frac{a-c}{2}z-\frac{1}{2}\log\frac{C}{A}} + \frac{B}{A}e^{-\frac{1}{2}\log\frac{C}{A}} + e^{-(\frac{a-c}{2}z-\frac{1}{2}\log\frac{C}{A})}\right) \\
&= Ae^{\frac{a+c}{2}z+\frac{1}{2}\log\frac{C}{A}}\left(2\cosh\left(\frac{a-c}{2}z - \frac{1}{2}\log\frac{C}{A}\right) + \frac{B}{A}e^{-\frac{1}{2}\log\frac{C}{A}}\right)
\end{aligned}$$

which is equivalent to

$$2\cosh\left(\frac{a-c}{2}z - \frac{1}{2}\log\frac{C}{A}\right) = -\frac{B}{A}e^{-\frac{1}{2}\log\frac{C}{A}},$$

which holds if and only if

$$z = \frac{2}{a-c}\left(\pm \operatorname{arcosh}\left(-\frac{B}{2A}e^{-\frac{1}{2}\log\frac{C}{A}}\right) + \frac{1}{2}\log\frac{C}{A} + 2m\pi i\right)$$

for $m \in \mathbb{Z}$. $\square$

Now, we can follow the same process we used for proving Theorem 3.1.2.

Theorem 3.2.2. Suppose $\overline{b_1}, \overline{b_2}, \cdots, \overline{b_n} \in \mathbb{C}$ is the convex polygon hull of

$$f(z) = \sum_{k=1}^{N} A_k e^{b_k z},$$

anti-clockwise, $n \leq N$, and

1. $\overline{b_j}, \overline{b_{j+2}}$ are the corners of one side, forming a side with exactly 3 collinear points;
2. $\overline{b_{j+1}}$ is the midpoint of $\overline{b_j}, \overline{b_{j+2}}$.

Then, the zeros of $f(z)$ arising from that particular side lie asymptotically on the rays

$$\frac{2}{b_j - b_{j+2}} \left(\pm \operatorname{arcosh} \left(-\frac{A_{j+1}}{2A_j} e^{-\frac{1}{2}\log \frac{A_{j+2}}{A_j}} \right) + \frac{1}{2} \log \frac{A_{j+2}}{A_j} + 2m\pi i \right)$$

for $m \in \mathbb{N}$.

Proof. We note that

$$\overline{b_k} = \overline{b_j} + (\overline{b_{j+2}} - \overline{b_j})(r + is)$$

for $r, s \in \mathbb{R}, s > 0$ from Lemma 3.0.1. Hence, we have

$$b_k = b_j + \left(b_{j+2} - b_j\right)\left(r - is\right).$$

From Lemma 3.2.1, the zeros of $A_j e^{b_j z} + A_{j+1} e^{b_{j+1} z} + A_{j+2} e^{b_{j+2} z}$ are at

$$z = \frac{2}{b_j - b_{j+2}} \left(\pm \operatorname{arcosh} \left(-\frac{A_{j+1}}{2A_j} e^{-\frac{1}{2}\log \frac{A_{j+2}}{A_j}} \right) + \frac{1}{2} \log \frac{A_{j+2}}{A_j} + 2m\pi i \right)$$

for $m \in \mathbb{Z}$.

Let $\alpha, \beta \in \mathbb{R}$ such that

$$2\left(\pm \operatorname{arcosh}\left(-\frac{A_{j+1}}{2A_j}e^{-\frac{1}{2}\log\frac{A_{j+2}}{A_j}}\right)+\frac{1}{2}\log\frac{A_{j+2}}{A_j}\right)=\alpha+i\beta.$$

We are going to compare the ratio $\left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right|$ for $m \to \infty$ and $m \to -\infty$, for the same reason as stated in the proof of Theorem 3.1.2.

$$\begin{aligned}\left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right| &= \left|\frac{A_k}{A_j}\right|\left|e^{\left(b_{j+2}-b_j\right)(r-is)\left(\frac{\alpha+i\beta}{b_j-b_{j+2}}+\frac{4m\pi i}{b_j-b_{j+2}}\right)}\right| \\ &= \left|\frac{A_k}{A_j}\right|\left|e^{-(r-is)(\alpha+i\beta+4m\pi i)}\right| \\ &= \left|\frac{A_k}{A_j}\right|\left|e^{-(r-is)(\alpha+i\beta)-4mr\pi i}\right|\left|e^{-4ms\pi}\right| \\ \implies \left|\frac{A_k e^{b_k z}}{A_j e^{b_j z}}\right| \to 0 &\iff m \to \infty.\end{aligned}$$

Hence, $m \in \mathbb{N}$. □

For the case where b in (3.2) is not the midpoint, I was unable to obtain an analytic solution for $f(z) = 0$. This is because if we follow the proof of Lemma 3.2.1, we end up with $2\cosh\left(\frac{a-c}{2}z - \frac{1}{2}\log\frac{C}{A}\right) = -\frac{B}{A}e^{z\left(b-\frac{a+c}{2}\right)-\frac{1}{2}\log\frac{C}{A}}$, and when we compare real and imaginary parts, we end up with a system of equations that can only be solved numerically. In order to solve it, one would have to obtain the Jacobian matrix by hand, estimate the locations of solutions, and then enter it into a numerical solver, which is too much work.

However, all is not lost.

Proposition 3.2.3. Suppose z_0 is such that for

$$f(z) = Ae^{az} + Be^{bz} + Ce^{cz},$$

where $A, B, C, a, b, c \in \mathbb{C}, A, B, C \neq 0$, a, b, c are collinear on the complex plane, $f(z_0) = 0$. Then, $f(z_0 + 2\omega\pi i) = 0$ where $\omega, \omega a, \omega b, \omega c \in \mathbb{Z}$.

Proof. We evaluate

$$\begin{aligned} f(z_0 + 2\omega\pi i) &= Ae^{az_0}e^{2a\omega\pi i} + Be^{bz_0}e^{2b\omega\pi i} + Ce^{cz_0}e^{2c\omega\pi i} \\ &= Ae^{az_0} + Be^{bz_0} + Ce^{cz_0} \\ &= 0. \end{aligned}$$ □

In fact, we can still obtain an asymptotic locus of zeros by first locating all the zeros within a $2\omega\pi i$ range using a numerical root finder (such as the one I developed [4]), $\{r_1, \cdots, r_m\}$. Then, the asymptotic locus of zeros corresponding to that side of a convex polygon hull is $\{r_j + 2\omega\pi i\}$ for $j \in \{1, \cdots, m\}$, and only for positive values of ω, following the logic of Theorem 3.2.2.

3.3 Conclusion

Corollary 3.3.1. Suppose $\overline{b_1}, \overline{b_2}, \cdots, \overline{b_n} \in \mathbb{C}$ is the convex polygon hull of

$$f(z) = \sum_{k=1}^{N} A_k e^{b_k z},$$

anti-clockwise, $n \leq N$, and each side of the polygon either has

1. only 2 points forming the corners,
2. 3 collinear points, with 2 points forming the corners and the third point exactly at the center of the side.

Then we can obtain asymptotic formulae for *all* zeros of $f(z)$ with $|z|$ sufficiently large using Theorems 3.1.2 and 3.2.2.

This result is significant, because many exponential sums fit the description above.

If one wishes to see the results from Theorems 3.1.2 and 3.2.2 in action, they can be checked with a numerical root finder as well.

Chapter 4

Numerical Method

4.1 Overview of method

I implemented a package in Julia ([4]) that uses the argument principle to obtain numerical approximations for complex zeros of analytic functions.

The code has two main functions,

1. `countZeros`: count the number of zeros within a given rectangular domain

2. `findZerosWithSubdivision`: obtain rectangular locations of sufficiently small perimeters (determined by a user-defined error tolerance), containing zeros of the function within a given rectangular domain.

4.2 Derivation of algorithm

Given a rectangular domain on the complex plane, we will approximate the locations of the zeros of an analytic function. To achieve this end, we will be making use of the argument principle.

Theorem 4.2.1 (Argument Principle)**.** If f is a meromorphic function inside and on some closed contour C, and f has no zeros on C, then

$$\frac{1}{2\pi i}\oint_C \frac{f'(z)}{f(z)}\mathrm{d}z = Z - P$$

where Z and P denote the number of zeros and poles of $f(z)$ inside the contour C, with each zero counted as many times as its multiplicity and order. This statement of the theorem assumes that the contour C is simple, that is, without self-intersections, and that it is oriented anti-clockwise.

Since we are solely looking at exponential sums, which are holomorphic (and therefore meromorphic) functions, the number of poles will be zero.

Hence, for the purposes of this paper, we can simplify the argument principle to

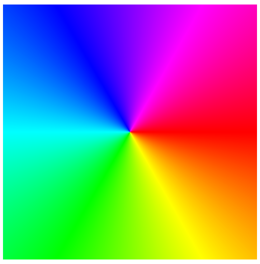
$$\frac{1}{2\pi i}\oint_C \frac{f'(z)}{f(z)}\mathrm{d}z = Z.$$

4.2.1 Seeing the argument principle

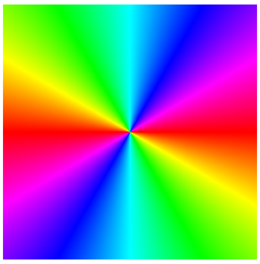
We can refer to rectangular domains on the complex plane by their upper left and lower right corners. For example, $(-1+i, 1-i)$ is a rectangular domain with $-1+i$ as the upper left corner, and $1-i$ as the lower right corner.

We can visually count the number of zeros in each of the following boxes by using the argument principle.

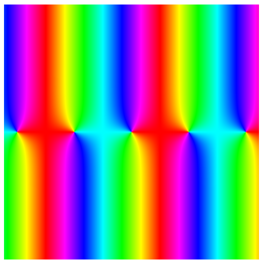
In the context of our algorithm, every time the colour on the edge of the box goes from dark blue across cyan to green in the anti-clockwise direction (indicating that there is a jump in $\arg f(z)$ from π to $-\pi$), the count of zeros is incremented by 1.



(A) $f(z) = z$



(B) $f(z) = z^2$



(C) $f(z) = sin(z)$

FIGURE 4.1: Phase portraits of sample functions; the boxes are $(-1+i, 1-i)$ for 4.1a and 4.1b, and $(-7+7i, 7-7i)$ for 4.1c

If the colour goes from green across cyan to dark blue (indicating that there is a jump in $\arg f(z)$ from $-\pi$ to π), the count of zeros is decremented by 1.

In Figure 4.1, the count of zeros for the 3 phase portraits is 1, 2 and 5, from left to right.

4.2.2 Proof of approach

The function `countZeros` is based off Theorem 4.2.1 .

We are interested in the points on the border where there is a jump of 2π in $\arg f(z)$, i.e. the colour changes from dark blue across cyan to green, or green across cyan to dark blue. We shall refer to these points as *jump points*.

`countZeros` detects jump points around a box. It does that by evaluating the angle $\arg(f(z_i))$ at each point z_i, separated by a step size. When the difference between 2 points is larger than $2\pi - \epsilon$, a jump point is detected. Then, the number of jump points can be used to determine the number of zeros a box contains.

Let C be the closed contour of a box that contains zeros, without any zeros on the border, Z be the number of zeros within C, and f be a holomorphic function. From Theorem 4.2.1,

$$\begin{aligned} Z &= \frac{1}{2\pi i} \oint_C \frac{f'(z)}{f(z)} \mathrm{d}z \\ &= \frac{1}{2\pi i} \oint_C \frac{d}{dz} \log(f(z)) \mathrm{d}z. \end{aligned}$$

Because $f(z)$ evaluates to a complex number, it can be expressed in its polar form. Hence, we have,

$$\begin{aligned} \log(f(z)) &= \log(|f(z)|e^{i \arg f(z)}) \\ &= \log(|f(z)|) + i \arg f(z) \\ \implies \quad \frac{1}{2\pi i} \frac{d}{dz} \log(f(z)) &= \frac{1}{2\pi i} \frac{d}{dz} log(|f(z)|) + \frac{1}{2\pi} \frac{d}{dz} \arg f(z) \\ \implies \quad Z &= \frac{1}{2\pi i} \oint_C \frac{d}{dz} log(|f(z)|) \mathrm{d}z + \frac{1}{2\pi} \oint_C \frac{d}{dz} \arg f(z) \mathrm{d}z \\ &= \frac{1}{2\pi} \oint_C \frac{d}{dz} \arg f(z) \mathrm{d}z \\ &\text{(since } \frac{d}{dz} \log(|f(z)|) \text{ is continuous, the closed} \\ &\text{contour integral of it is 0).} \end{aligned}$$

Suppose there are n points on the border where $\arg f(z) = \pi$ or $-\pi$; meaning there are n jump points.

The contour starts at z_0, travels in a counter-clockwise direction, passes through each jump point z_i, and ends at z_{n+1}, where $z_0 = z_{n+1}$.

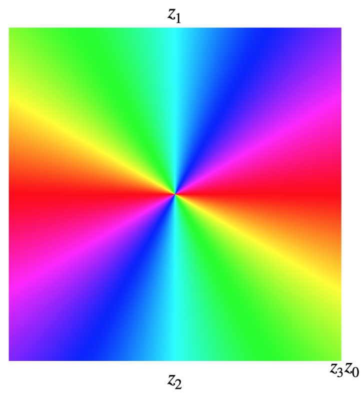


FIGURE 4.2: Sample box where $n = 2$

Without loss of generality, suppose z_0 is not a jump point. (If z_0 is a jump point, we can simply choose another starting point.)

$$\begin{aligned}
Z &= \frac{1}{2\pi} \oint\limits_C \frac{d}{dz} \arg f(z) \mathrm{d}z \\
&= \frac{1}{2\pi} \sum_{j=1}^{n+1} \int_{z_{j-1}}^{z_j} \frac{d}{dz} \arg f(z) \mathrm{d}z \\
&= \frac{1}{2\pi} \sum_{j=1}^{n+1} \left(\lim_{z \uparrow z_j} \arg f(z) - \lim_{z \downarrow z_{j-1}} \arg f(z) \right) \\
&= \frac{1}{2\pi} \left(-\arg f(z_0) + \arg f(z_{n+1}) \right) + \frac{1}{2\pi} \sum_{j=1}^{n} \left(\lim_{z \uparrow z_j} \arg f(z) - \lim_{z \downarrow z_j} \arg f(z) \right) \\
&= \frac{1}{2\pi} \sum_{j=1}^{n} \left(\lim_{z \uparrow z_j} \arg f(z) - \lim_{z \downarrow z_j} \arg f(z) \right) \\
&= (\text{number of jumps of } \pi \text{ to } -\pi) - (\text{number of jumps of } -\pi \text{ to } \pi).
\end{aligned}$$

4.3 Subdivision algorithm

The generic subdivision algorithm for root finding subdivides an initial range of interest, discards regions guaranteed not to contain zeros, and repeats this process until satisfactorily small area(s) can be verified to contain root(s) [6]. Hence, a list of rather small regions that contain roots can be obtained, which allows us to determine the roots with a certain degree of precision.

Subdivision methods have the advantage of being able to restrict computational effort to a given region, and may terminate quickly if there are no zeros [13].

I adapted the generic subdivision algorithm for `findZerosWithSubdivision`. I define

1. an inclusion predicate Inclusion(A, f) which holds only if the region A contains roots of f, and the sum of the length and width of A falls within a user-defined error margin;

2. an exclusion predicate Exclusion(A, f) which holds only if A does not contain a root of f.

The algorithm is recursive and written with tail-call optimisation. The structure of my algorithm is as follows:

Algorithm 1: Subdivision algorithm

Input : initial rectangular domain A_0, function f, error

Output: array R of isolating rectangular regions for the zeros of f

```
R → {} ;
Q → {A_0} ;
while Q is not empty do
    A = dequeue Q ;
    if Inclusion(A, f) holds then
        append A to R ;
    else if Exclusion(A, f) holds then
        discard A ;
    else
        subdivide A ;
        enqueue subdivisions to Q ;
end
```

4.4 Handling edge cases

The handling of roots lying very near or on the common boundary of some subdivided regions is a delicate issue in the design of subdivision algorithms. This issue is generally glossed over by existing literature [5]. Two suggestions that I came across in [5] are

1. "Subdivide the region to the maximal extent possible, and then coalesce unresolved boxes (these will include boxes that share a root on

their boundary). Coalescing is performed by grouping together unresolved boxes that share a common edge, and by constructing a minimal bounding box around these groups. We can then run our subdivision algorithm with these bounding boxes as starting regions to isolate roots that they are suspected to contain."

2. "Perturb boxes during the subdivision process. If we suspect that a box shares a root on its boundary with another, during its next subdivision we adjust its boundaries outward by a fixed ϵ, while at the same time adjusting the boundaries of the corresponding neighbours inwards."

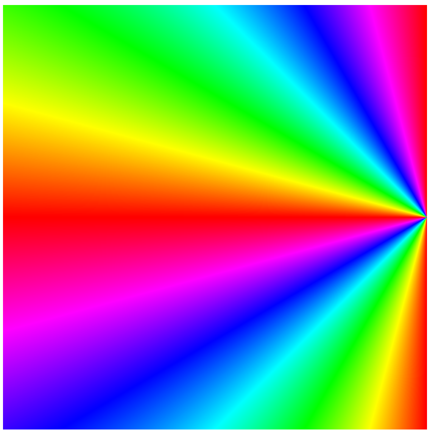


FIGURE 4.3: $f(z) = (z - 1.999)^4$, in the box $(-2 + 2i, 2 - 2i)$

The method I propose is to first count the total zeros in the current region being examined, C_1. In the 4 regions that it is divided into, we sum up the count of zeros in each subregion, C_2. If $C_1 \neq C_2$, it indicates that during subdivision, there is a zero lying very near or on the contour, causing `countZeros` not to work as intended. I handle this with an exception, `ZerosNearContourException`.

```
struct ZerosNearContourException <: Exception
biggerBox::Tuple
```

```
count:: Int
potentialLocations::Queue{Tuple{Tuple{Any,Any},Any}}
end
```

This exception contains information on

1. the current region being examined that caused the exception to be raised,
2. the total count of zeros it contains,
3. other regions that contain zeros, and the count of zeros they each contain.

When this happens, there are some options to try obtaining better results, including changing the upper left/lower right corners of the initial region and making the step size smaller. The error message for this exception comes with this tip for users.

The rationale for having this exception is so that users can at least have a narrower range of regions to examine when the algorithm terminates. Furthermore, checking whether $C_1 = C_2$ is a fast way to verify that the algorithm is working as intended, since we need to use `countZeros` on each region anyways. Lastly, it also appears to be more efficient for the user to change the values of the input of `findZerosWithSubdivision`, compared to performing coalescing or peturbation like suggested above, which could slow the algorithm down significantly.

4.5 Testing

I used both unit-tests and randomised property-based testing for `countZeros` and `findZerosWithSubdivision`. Tests can be found at [4].

4.6 Conclusion

In conclusion, a complex root finding algorithm for analytic functions is presented, with a distinct approach in addressing edge cases. The accuracy of the numerical approximations of zeros can be controlled.

Chapter 5

Further Work

The analytical methods described in Chapter 3 can be made more easily accessible through implementing a dedicated library to solving exponential sums, such as in Julia. The outline of what such a library would do is as follows:

1. Given an exponential sum, construct the convex polygon hull from the conjugate of its exponents
2. Check if each side of the convex polygon hull fits the criteria described in Corollary 3.3.1
3. If the side does, include the asymptotic locus of zeros arising from it as part of the returned result, using Theorem 3.1.2 and Theorem 3.2.2, represented with symbolic computation such as SymPy [11]
4. If it does not, bring it to the user's attention when returning the result

Obtaining a convex polygon hull is simple, with Julia packages such as QHull [10] and Polyhedra [9]. Though these pacakges ignore collinear points of each side, implementing a function that includes the collinear points is fairly straightforward as well, by checking whether unincluded points fit in with the equation of the line formed by the 2 corner points. I have previously implemented such a function as part of another project.

Therefore, implementing a library for obtaining analytical solutions of exponential sums appears very much feasible. This library could combine numerical and analytical approaches as well, by including a numerical root finder.

Having such a libary would make obtaining asymptotic solutions of exponential sums much more convenient, compared to doing it by hand, which is laborious and mechanical.

For the numerical solver, more work could be done on making it more precise for zeros of higher multiplicities. At the moment, it works best for single zeros. It can still work for multiple zeros, but to a lower degree of precision. For our intended purpose, which is to find the zeros of exponential sums, it is largely fine as they usually have single zeros. But for examples such as $\cos(x) - 1$, it would be helpful.

Higher computation speed, for industrial purposes, through parallel computing can also be explored, since subdivided regions are processed independently.

Glossary

commensurable Two non-zero real numbers are said to be commensurable if their ratio is a rational number. 4

complex phase portrait Refer to Appendix A . 5

holomorphic A holomorphic function is a complex-valued function of one or more complex variables that is, at every point of its domain, complex differentiable in a neighborhood of the point. 19

locus In geometry, a locus (plural: loci) is a set of all points whose location satisfies or is determined by one or more specified conditions. 1

meromorphic A meromorphic function on an open subset D of the complex plane is a function that is holomorphic on all of D except for a set of isolated points, which are poles of the function. 19

monomial symbol differential operator The symbol of a linear differential operator is a polynomial that represents it, obtained by replacing each partial derivative with a new variable. If the symbol is monomial, that means there is only one kind of partial derivative. 1

ray In geometry, a ray is a part of a line that has a fixed starting point but no end point. It can extend infinitely in one direction. 3

Bibliography

[1] G. D. Birkhoff. "On the Asymptotic Character of the Solutions of Certain Linear Differential Equations Containing a Parameter". In: *Trans. Amer. Math. Soc.* 9 (1908), pp. 219–231.

[2] "ComplexPhasePortrait.jl". https://github.com/JuliaHolomorphic/ComplexPhasePortrait.jl.

[3] "ComplexPortraits.jl". https://github.com/yanhua-wang/ComplexPortraits.jl.

[4] "FindComplexZeros.jl". https://github.com/yanhua-wang/FindComplexZeros.jl.

[5] Narayan Kamath, Irina Voiculescu, and Chee K Yap. "Empirical study of an evaluation-based subdivision algorithm for complex root isolation". In: *Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation*. 2012, pp. 155–164.

[6] Alexander Kobel. "Certified numerical root finding". In: *Master's thesis, Universität des Saarlandes, Saarbrücken, Germany* (2011).

[7] Rudolph E Langer. "On the zeros of exponential sums and integrals". In: *Bulletin of the American Mathematical Society* 37.4 (1931), pp. 213–239.

[8] Beatrice Pelloni. “The spectral representation of two-point boundary-value problems for third-order linear evolution partial differential equations”. In: *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 461 (2005), pp. 2965–2984.

[9] "Polyhedra.jl". https://github.com/JuliaPolyhedra/Polyhedra.jl.

[10] "QHull.jl". https://github.com/JuliaPolyhedra/QHull.jl.

[11] "SymPy.jl". https://github.com/JuliaPy/SymPy.jl.

[12] Elias Wegert. *Visual complex functions: an introduction with phase portraits*. Springer Science & Business Media, 2012.

[13] Chee K Yap and Michael Sagraloff. “A simple but exact and efficient algorithm for complex root isolation”. In: *Proceedings of the 36th international symposium on Symbolic and algebraic computation*. 2011, pp. 353–360.

Appendix A

Complex Phase Portraits

Complex phase portraits is a special coloring technique which visualizes functions as images.

The phase of a complex number is $cos\phi + isin\phi$. It is well-defined for all complex numbers except for zero [12].

Because of the above property, phase can be colour-coded. Phase portraits depict the color-coded values of the phase on the domain of the function. The zeros of analytic functions may be identified as the points of discontinuity on their phase portraits.

In this paper, I used phase portraits as a visual aid to locate zeros of equations, and to demonstrate how my algorithm works.

FIGURE A.1: The phase portrait of $f(z) = z$

Appendix B

Worked Example

Consider the boundary value problem

$$y^{(4)}x = \lambda y, \tag{B.1}$$

$$y(0) = 0 = y(1), \tag{B.2}$$

$$y'(0) = 0, \tag{B.3}$$

$$y''(0) - y'(1) = 0. \tag{B.4}$$

Let $\lambda = k^4, k \in \mathbb{C} \backslash \{0\}$.

Using the 4 fourth roots of 1, we get the equation

$$Y(x) = Ae^{kx} + Be^{-kx} + Ce^{ikx} + De^{-ikx}$$

that solves (B.1).

Hence, we have

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ e^k & e^{-k} & e^{ik} & e^{-ik} \\ 1 & -1 & i & -i \\ 1-e^k & 1-e^{-k} & -(1-e^{ik}) & -(1-e^{-ik}) \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}. \tag{B.5}$$

For the sake of simplifying calculations, on the right hand side of (B.5), we can switch the second and third column of the first matrix, and B and C

in the second matrix to get an equivalent expression.

In order for (B.5) to hold and to obtain a non-trivial solution, we must have

$$
\begin{aligned}
0 &= \det \begin{pmatrix} 1 & 1 & 1 & 1 \\ e^{k} & e^{ik} & e^{-k} & e^{-ik} \\ 1 & i & -1 & -i \\ 1-e^{k} & -(1-e^{ik}) & 1-e^{-k} & -(1-e^{-ik}) \end{pmatrix} \\
&= \sum_{j=1}^{4} i^{j} \det \begin{pmatrix} 1 & i & -1 \\ e^{i^{j}k} & e^{i^{j+1}k} & e^{i^{j+2}k} \\ 1-e^{i^{j}k} & -\left(1-e^{i^{j+1}k}\right) & 1-e^{i^{j+2}k} \end{pmatrix} \\
&= 2ie^{-k} + (2-2i)e^{(-1-i)k} - 4e^{-ik} + 4e^{ik} + (-2-2i)e^{(i-1)k} \\
&\quad + (2i-4)e^{(1-i)k} + (-2-2i)e^{(1+i)k} + (-4i)e^{k}.
\end{aligned}
$$

(The in-between working steps are left as an exercise for the reader.)

We see that we get the exponential sum

$$
\begin{aligned}
\Phi(k) &= 2ie^{-k} + (2-2i)e^{(-1-i)k} - 4e^{-ik} + 4e^{ik} + (-2-2i)e^{(i-1)k} \\
&\quad + (2i-4)e^{(1-i)k} + (-2-2i)e^{(1+i)k} + (-4i)e^{k}. \qquad \text{(B.6)}
\end{aligned}
$$

When we plot the conjugates of the exponents of (B.6) in the complex plane, we notice that we get a square with vertices at $-1+i, 1+i, -1-i, 1-i$, and all 4 sides have 3 equally spaced collinear points.

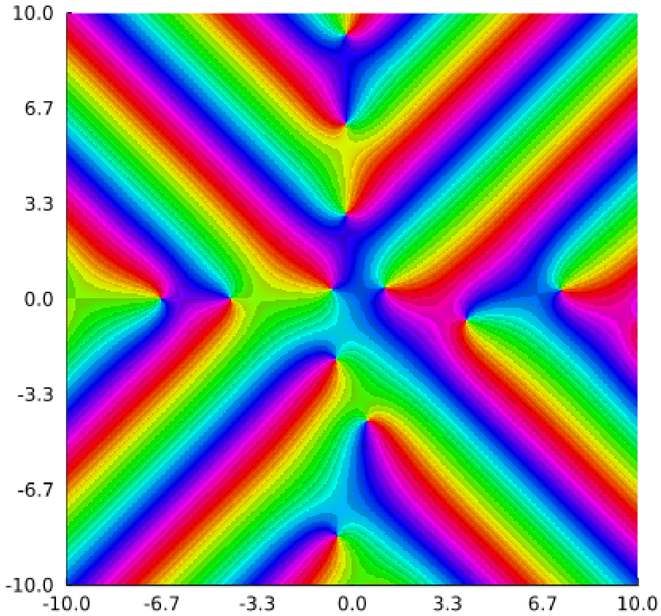


FIGURE B.1: The phase portrait (B.6)

Hence, we can use Theorem 3.2.2 to get the asymptotic locus of zeros arising from each side.