

2007

Nonlinear approximation using Blaschke polynomials

Daniel Van Vliet
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Van Vliet, Daniel, "Nonlinear approximation using Blaschke polynomials" (2007). *Graduate Theses, Dissertations, and Problem Reports*. 2598.
<https://researchrepository.wvu.edu/etd/2598>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Nonlinear Approximation Using Blaschke Polynomials

Daniel Van Vliet

Dissertation submitted to the
Eberly College of Arts and Sciences
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Mathematics

Sherman Riemenschneider, Ph.D., Chair
Weifu Fang, Ph.D.
Dening Li, Ph.D.
Natalia Schmid, Ph.D.
Jerzy Wojciechowski, Ph.D.

Department of Mathematics

Morgantown, West Virginia
2007

Keywords: approximation, empirical mode decomposition,
instantaneous frequency, analytic signals

Copyright 2007 Daniel Van Vliet

ABSTRACT

Nonlinear Approximation Using Blaschke Polynomials

Daniel Van Vliet

This dissertation, entitled Nonlinear Approximation Using Blaschke Polynomials, is motivated by questions arising from Empirical Mode Decomposition (EMD). EMD is a signal processing method which decomposes input signals into components called intrinsic mode functions (IMFs). These IMFs often have the desirable property that the instantaneous frequency of their analytic signals is positive. However, this is not always the case.

The first two chapters are introductions to approximation in general, and Empirical Mode Decomposition, respectively.

The third chapter presents a characterization of which analytic signals have the property of non-negative instantaneous frequency. These 'analytic signals with non-negative instantaneous frequency' (ASNIFs) are described using Hardy spaces on the unit disc. Analogous results are also found for analytic signals which are boundary values of elements of Hardy spaces on the half-plane.

The fourth chapter describes in general terms how one might construct an approximation method using ASNIFs, or possibly some restricted subclass of ASNIFs, as approximants.

The fifth chapter introduces a special set of ASNIFs: Blaschke polynomials. These are functions which are linear combinations of Blaschke products, which are a special kind of ASNIF. Blaschke polynomials are a natural extension of other classical approximating sets, and have some interesting properties of their own. Some of these properties are explored.

The sixth chapter contains an implementation of a signal decomposition method using Blaschke polynomials, and a discussion of the results.

The seventh chapter further explores the conclusions of this work, and lays out future research questions.

Acknowledgements

I am deeply thankful to my advisor, Dr. Sherman Riemenschneider, for his patience and guidance through this adventure. By encouraging my better insights, questioning my wayward thoughts, and always having a suggestion when needed, Sherm has not only guided this dissertation to completion, but also helped establish my intuition as a researcher. For this I will thank him for the rest of my career.

The members of my examining committee have also helped me immensely through this process. Weifu Fang, Natalia Schmid, Dening Li, and Jerzy Wojciechowski have contributed valuable time, energy, and perspective which has helped to shape this dissertation, and myself. Weifu has been an inspiration both in the classroom and in research. By his energy and enthusiasm, he has helped me to see the task of setting these thoughts in order as a joy rather than work. Natalia has provided insightful comments and direction at critical junctures throughout the process. Dening and Jerzy were both indispensable resources to me, especially in establishing the background I needed to approach this project.

Dr. Andy Liu will always hold a special place in my heart. Without his early mentorship I doubt I would have ended up in mathematics, and certainly not so well prepared. Andy's example of enthusiasm for mathematics, his love of teaching, and his boundless generosity will always serve as an inspiration to me, as well as the many others whose lives he has touched. From the early days of the SMART math club, Andy's efforts gave me a solid foundation in mathematics, a social haven, and a wealth of experiences which will continue to shape my life. I can only hope to have such an amazing impact on my own students in the future.

I am forever indebted to my lovely wife Nicole for her patience and support. Also to my parents who have been integral in preparing me for every challenge I face, including (of course) this one.

Contents

1	Introduction	1
1.1	Preliminaries, Definitions, and Notation	2
1.1.1	Order of Approximation	4
1.2	Fourier Series	5
1.2.1	The Thresholding Fourier Series	6
1.3	Wavelets and Frames	7
1.4	Highly Nonlinear Methods	8
2	The Empirical Mode Decomposition	9
2.1	Introduction	9
2.1.1	Definitions and Notation	9
2.1.2	The sifting algorithm	10
2.1.3	The Hilbert Spectrum	11
2.2	Research Problems	12
3	ASNIFs	15
3.1	Preliminaries, Definitions, and Notation	16
3.2	Instantaneous Frequency of Analytic Signals	18
3.2.1	Periodic Analytic Signals	19
3.2.2	Nonperiodic Analytic Signals	22
3.3	Constructing ASNIFs	24
3.3.1	Outer Functions with Locally Bounded Instantaneous Frequency . . .	25
3.3.2	Outer Functions whose Instantaneous Frequency is not Locally Bounded	26
3.4	Conclusions	27
3.5	Consequences for an “EMD-like” ASNIF decomposition	28
4	An ASNIF-based Signal Decomposition	31
4.1	Introduction	31
4.2	An energy estimate for the $(\mathcal{Y} \subset \mathcal{X})$ decomposition	33

4.3	Examples of J Functional Based Decompositions	34
4.4	Using the Bergman Norm $\ \cdot\ _{0,2}$ as $\ \cdot\ _y$	35
4.4.1	Using the Bergman J -Functional with the Fourier basis functions	35
4.4.2	An algorithm for use with Blaschke products	38
5	Blaschke Polynomials	41
5.1	Blaschke Polynomials	42
5.2	Spaces of Blaschke Polynomials	44
5.2.1	$\mathbb{C}[\zeta]$	45
5.2.2	$\mathbb{C}[\zeta_1, \dots, \zeta_n]$	45
5.2.3	Spaces of Blaschke polynomials in infinitely many variables	46
5.3	Approximation Using Blaschke Polynomials	47
5.4	A Possible Extension to Singular Inner Functions	48
6	J-functional Decomposition Using Blaschke Polynomials	51
6.1	Introducing the Algorithms	51
6.1.1	The IBA Algorithm	52
6.1.2	The JFD Algorithm	53
6.1.3	A Smaller Constraint Set	53
6.2	Minimizing the IBA functional	54
6.2.1	Consequences for Stopping Criteria	56
6.3	Minimizing the JFD functional	56
6.3.1	Consequences for Stopping Criteria	59
6.4	Implementing the Algorithms	60
6.4.1	IBA Pseudocode	61
6.4.2	JFD Pseudocode	61
6.4.3	A further note on stopping criteria	62
6.5	Examples and Discussion	62
6.5.1	Calibration	62
6.5.2	Reconstruction Accuracy	63
6.5.3	Identification of localized frequencies	67
7	Conclusions and Future Research	69
7.1	ASNIFs	69
7.1.1	Future research	70
7.2	Blaschke polynomials	71
7.2.1	Future research	72

8	Appendix	77
8.1	Proofs	77
8.1.1	Proof of equation 3.16	77
8.1.2	Proof of equation 3.17	78
8.1.3	Proof of the convexity of J_2 w.r.t. a and b	79
8.2	MATLAB Code	80
8.2.1	makelists.m	80
8.2.2	IBA.m	82
8.2.3	JFD.m	83
8.2.4	Examples	84

List of Figures

6.1	log(reducedresiduals) for IBA algorithm with input signals $f_{m,r}(z)$	64
6.2	log(reducedresiduals) for JFD algorithm with input signals $f_{m,r}(z)$	65
6.3	log(reducedresiduals) for IBA algorithm with input signals $g_{m,r}(z)$	66
6.4	log(reducedresiduals) for IBA algorithm with input signals $g_{m,r}(z)$	66

Chapter 1

Introduction

The fields of signal processing, approximation theory, wavelets, time-frequency analysis, and other related areas have undergone a rapid evolution since the development of wavelet theory. Wavelets brought with them powerful tools for analysis, compression, encoding, and pretty much any other task one might ever need to do with signals or images. As wavelet methods have evolved and matured, it has become clear that they are not without limitations. Due to their inherently linear structure, they may not be the best tool for analyzing highly nonlinear problems. Recently, new methods have been proposed which show some promise in dealing with highly nonlinear signals. The Empirical Mode Decomposition (EMD), developed by Norden Huang[13], is one such method. EMD represents a bold attempt to address some of the shortcomings that linear signal processing methods have when dealing with nonlinear nonstationary signals. EMD has produced some promising results, but is hampered by what might be its greatest strength: it is (currently) only defined algorithmically. There is no analytic representation of how EMD produces its output.

In some important ways, EMD serves as a prototype of the next generation¹ of signal processing methods:

- Components are selected by a nonlinear process.
- The set of potential component functions is much larger than that of any wavelet (or other linear) method.

These characteristics are typical of methods being studied currently in the field of nonlinear approximation theory (see [6]). Interesting research problems have presented themselves regarding EMD. In Chapter 2, some of these problems are outlined. In this work, we address some of these problems in a way that will hopefully benefit not only EMD research, but other aspects of nonlinear approximation theory as well.

¹after wavelet type methods

1.1 Preliminaries, Definitions, and Notation

Hilbert spaces provide the simplest setting within which to explore the problems of approximation theory. The results obtained for Hilbert spaces are rich enough that they illustrate many important aspects of modern approximation theory, and the results for Hilbert spaces often extend quite naturally to more general settings when necessary. Unless stated otherwise, the results presented in this work are developed within the context of separable Hilbert spaces. This introduction closely parallels chapter 2 of [6], with some omissions. For a more thorough introduction, see [6].

Let X be a separable Hilbert space, with inner product $\langle *, * \rangle_X$, and the norm induced by the inner product given by $\|*\|_X = \sqrt{\langle *, * \rangle_X}$. (When there is no ambiguity, the subscript X may be dropped from the norm and inner product.) Let $\{\eta_k\}_{k=1}^\infty$ be an orthonormal basis for X .

Definition 1.1.1. *Let X be a separable Hilbert space, $\mathcal{Y} \subset X$, and M be a surjective mapping $M : X \rightarrow \mathcal{Y}$. Such a mapping M is called an approximation method. The set \mathcal{Y} is called the set of approximants.*

Definition 1.1.2. *If $M : X \rightarrow \mathcal{Y}$ is a linear map, then it is called a linear approximation method. If M is not a linear map, then M is called a nonlinear approximation method.*

For linear approximation, we use the notation $\mathcal{Y} = X_n := \text{span}(\{\eta_k : 1 \leq k \leq n\})$ for the n -dimensional subspace of X used to approximate an element $f \in X$. The error of the best approximation in X_n is given by

$$E_n(f)_X := \inf_{g \in X_n} \|f - g\|_X. \tag{1.1}$$

When performing nonlinear approximation, we construct the set of n -term approximations which replaces X_n by the space $\mathcal{Y} = \Sigma_n$ consisting of all elements $g \in X$ which can be expressed as

$$g = \sum_{k \in \Lambda} c_k \zeta_k, c_k \in \mathbb{C}$$

where $\Lambda \subset \mathbb{N}$ is a set of indices with $|\Lambda| \leq n$.² The ζ_k 's will initially be elements of the orthonormal basis $\{\eta_k\}$. Later, when looking at highly nonlinear approximation methods the ζ_k 's will be chosen from a set which is much larger than $\{\eta_k\}$. Notice that the space Σ_n is not linear. While Σ_n is closed under scalar multiplication, a sum of two elements of Σ_n may require up to $2n$ terms in its representation by the ζ_k . Analogous to the linear space E_n , we

²We use $\#\mathcal{S}$ to denote the cardinality of the set \mathcal{S} .

define the *error of the best n -term approximation* for a nonlinear approximation method as:

$$\sigma_n(f)_X := \inf_{g \in \Sigma_n} \|f - g\|_X.$$

Definition 1.1.3. *For a linear approximation method, an n -term best approximation to $f \in X$ is an element $f^* \in X_n$ which satisfies*

$$\|f - f^*\| = E_n(f)_X$$

For a nonlinear approximation method, an n -term best approximation to $f \in X$ is an element $f^ \in \Sigma_n$ which satisfies*

$$\|f - f^*\| = \sigma_n(f)_X$$

Basic questions concerning the best approximation are: existence, uniqueness, characterization (Can one recognize f^* without comparing it to every other element of X_n or Σ_n), and construction. Some approximant sets lend themselves to the easy computation of the best approximation. For others it may be impractical to calculate the best approximation; for these methods, we are often interested in comparing the resulting approximation with the best approximation in order to see how “good” the method is. Even if the best approximation is not available, it may be possible to obtain useful bounds on the distance between a given approximation and the best approximation.

Definition 1.1.4. *An iterative approximation method is a sequence $\{M_n\}$ of approximation methods.*

A simple result which characterizes the structure of iterative best approximation methods is:

Theorem 1.1.5. *If the family of maps $\{M_n\}_{n=1}^\infty$ is an iterative approximation method for which $M_n(f) = f^*$ is given by the best approximation from \mathcal{Y}_n , and the error of approximation $\|f - f^*\|_X$ is non-increasing in n for all $f \in X$, then*

$$\mathcal{Y}_1 \subseteq \mathcal{Y}_2 \subseteq \mathcal{Y}_3 \subseteq \dots$$

Proof. Choose an element $y \in \mathcal{Y}_m$. The error of the m^{th} approximation is 0. If $\|y - M_n(y)\|_X$ is non-increasing in n , then $\|y - M_n(y)\|_X = 0 \forall n \geq m$. Therefore $y \in \mathcal{Y}_n \forall n \geq m$. \square

This result states that if the error of approximation is monotonically decreasing for all f , then the individual iterations of the approximation method will have a lot in common structurally. This is true of most commonly used approximation methods. After all, if each

iteration of an approximation method has very little in common with each other, then there is often not much to be gained by considering them together as a family.

In this work, one goal is characterizing more precisely in what way various nonlinear approximation methods are nonlinear. To aid in exploring this perspective, we introduce the following notation and terminology.

Definition 1.1.6. *$R - S - T$ Factorization.* Let \mathcal{Z} be a linear space. Choose operators $T : X \rightarrow \mathcal{Z}$, $S : \mathcal{Z} \rightarrow \mathcal{Z}$ and $R : \mathcal{Z} \rightarrow \mathcal{Y}$ which satisfy

$$R \circ S \circ T = M,$$

and $\forall z \in \mathcal{Z}$, there exists a linear projection operator $P_z : \mathcal{Z} \rightarrow \mathcal{Z}$ such that $S(z) = P_z(z)$. Then T is called a decomposition operator, S is called a selection operator, and R is called a reconstruction operator for the approximation method M .

In approximation literature, the decomposition operator may also be called a *analyzing operator*, or a *frame operator*.

The parameter space \mathcal{Z} may be a Hilbert space, or it may have some other structure³. In the case that \mathcal{Z} is countable, the elements of \mathcal{Z} are called coefficients. If \mathcal{Z} is uncountable and equipped with a non-discrete topology, the elements of \mathcal{Z} are often simply called variables or parameters.

Given an approximation method with mapping M , there always exists a factorization as in Definition 1.1.6. For a mapping M , a trivial factorization is given by letting $\mathcal{Z} = \mathcal{Y}$, $T = M$, and R and S both be the identity map on \mathcal{Y} . Then T is an decomposition operator for M , S is a selection operator, and R is a reconstruction operator for M . Of course, this factorization is not very useful or instructive. Thankfully, the factorization of M into decomposition and reconstruction operators is far from unique. Often, there is great latitude in choosing this factorization; this will be exploited to help illustrate the common structure of many approximation methods.

1.1.1 Order of Approximation

If we have a sequence of subsets $\{\mathcal{Y}_n\}_{n=1}^{\infty}$, Questions about order of approximation consider the behavior of $E_n(x) = \text{dist}(x, \mathcal{Y}_n)$ as a function of the parameter n . The first question is whether or not $\lim_{n \rightarrow \infty} E_n(x) = 0$. This cuts to the heart of whether the sets $\{\mathcal{Y}_n\}$

³Definition 1.1.6 may even be modified so that Z is not a Banach space. However, defining the selection operator appropriately becomes somewhat tricky in the absence of a linear structure. It can be done in terms of the image of the composition $R \circ S$, but the purpose becomes less clear. Also, this is not needed for any of the results in this work, as the parameter space Z will always have at least a linear structure.

are a reasonable choice for doing approximation at all. (Of course, if $\mathcal{Y}_n = X_n$, then this convergence is obvious. However, when considering other alternatives for \mathcal{Y}_n , this question may become nontrivial.) The next question is how fast or slow this convergence is for particular subsets of X . Usually the rate of convergence is compared with some standard function, e.g. $n^{-\alpha}$. For a particular class of functions $\mathcal{Z} \subset X$, there are two major classes of results regarding the degree of approximation:

1. Jackson type results: $x \in \mathcal{Z} \Rightarrow E_n(x) \in O(g(n))$
2. Bernstein type results: $E_n(x) \in O(g(n)) \Rightarrow x \in \mathcal{Z}$

These two types of results give different insights into how suitable the class \mathcal{Z} is for doing approximation in the space X .

1.2 Fourier Series

Any approximation method which uses an orthonormal basis is a generalized Fourier series. Such methods include orthonormal wavelet decompositions, the “standard” sine and cosine series, and many other orthogonal decompositions.

For a function $f \in X$, the Fourier series of f is given by:

$$f = \sum_{k=1}^{\infty} \langle f, \eta_k \rangle \eta_k.$$

This summation converges absolutely in $\|\cdot\|_X$ (see [15]).

We consider first the case of linear approximation based on the Fourier series. The best approximation to f from X_n is given by the projection

$$P_n(f) := \sum_{k=1}^n \langle f, \eta_k \rangle \eta_k$$

onto X_n and the approximation error is given by

$$E_n(f)_X^2 = \sum_{k=n+1}^{\infty} |\langle f, \eta_k \rangle|^2.$$

This linear approximation method admits a natural $R - S - T$ factorization:

$$T : f \mapsto \{\langle f, \eta_k \rangle_{k=1}^{\infty}\} \quad (1.2)$$

$$S_n : \{c_k\} \mapsto \{c_1, c_2, c_3, \dots, c_n, 0, 0, \dots\} \quad (1.3)$$

$$R : \{c_k\} \mapsto \sum_{k=1}^{\infty} c_k \eta_k \quad (1.4)$$

This is not the only factorization for M , but this factorization will help emphasize the similarities between this linear Fourier series approximation method and the nonlinear (thresholding) Fourier series which will be introduced next. While it may seem pedantic to split an operator as simple as $P_n(f)$ into three factors, each of the factors in the $R - S - T$ factorization has a particular role: T evaluates the coefficient values, S selects which coefficients will be used in the reconstruction of the approximant, and R performs the actual reconstruction. Notice that only the selection operator S_n depends on the parameter n ; the decomposition and reconstruction operators are identical for all values of n .

1.2.1 The Thresholding Fourier Series

Nonlinear approximation methods arise from the Fourier series as well. It is quite natural to ask what happens if one selects a subset of the Fourier coefficients not according to a prescribed cutoff in index value, but according to some prescribed cutoff in coefficient magnitude, i.e.

$$f \approx \sum_{|c_n| \geq K} c_n \eta_n$$

The approximation method given by $M_K(f) = \sum_{|c_n| \geq K} c_n \eta_n$ is nonlinear. This is easily shown:

$$M_K(K\eta_1) = K\eta_1 \neq 0 = KM_K(\eta_1)$$

Despite being nonlinear, the thresholding Fourier series still has a lot in common with the truncated Fourier series. The mapping M_K can still be factored into a form that is almost identical to the truncated Fourier series. The coefficients that end up being selected in the reconstruction are clearly identical to the coefficient values in the truncated Fourier series. The only difference is which coefficients are selected. This suggests a factorization of the form

$$M = R \circ S_K \circ T$$

where T and R are the decomposition and reconstruction operators given in equations 1.2

and 1.4 respectively, and S is a selection operator given by

$$S_K(\{c_k\}) = \{\tilde{c}_k\}, \quad \tilde{c}_k = \begin{cases} 0 & , |c_k| < K \\ c_k & , |c_k| \geq K \end{cases}$$

The thresholding Fourier series is prototypical of many nonlinear approximation methods. Some common methods which are similar to (or examples of) the thresholding Fourier series are greedy algorithm based methods (see [22]), and even the wavelet shrinkage method (see [7]). Even though the thresholding Fourier series is nonlinear, there is a lot about it that is linear. The (nonzero) coefficient values are all calculated by means of a linear projection. The reconstruction is also accomplished by a linear map.

1.3 Wavelets and Frames

An introduction to modern approximation methods would not be complete without mentioning of wavelets and frames. Wavelets have been an area of very active research, with many exciting results in linear and nonlinear approximation.

In the realm of linear approximation, wavelets have brought to the forefront the general setting within which they reside: frames. A *frame* is a set $\mathcal{F} = \{f_j\}$ of vectors for which there are constants $0 < A \leq B < \infty$ such that the *frame condition*

$$\forall x \in X \quad A \|x\|_X^2 \leq \sum_j \langle f_j, x \rangle^2 \leq B \|x\|^2$$

is satisfied. If $A = B$, the frame is said to be tight. This property allows frames to be used for approximation in a manner similar to a basis. Frames may be used for both linear and nonlinear approximation. The most commonly used frame-based approximation methods (including wavelet methods) use a common decomposition operator, called the frame operator.

Definition 1.3.1. Let X be a separable Hilbert space, and let $\mathcal{F} = \{f_j\}_{j=1}^\infty$ be a frame for X . The frame operator $T_{\mathcal{F}} : X \rightarrow l^2(\mathbb{C})$ is given by

$$T_{\mathcal{F}}(x) := \{\langle f_j, x \rangle\}_{j=1}^\infty$$

Because of the frame condition, the frame operator $T_{\mathcal{F}}$ is a bounded linear operator, and has a bounded inverse. This inverse operator is used as a reconstruction operator for frame based approximation methods.

1.4 Highly Nonlinear Methods

There are many ways to construct nonlinear approximation methods. The simplest is to apply some kind of nonlinear component selection to a linear approximation method (like the thresholding Fourier Series). Such methods are nonlinear, but they are not radically different from linear approximation methods. They use a linear operator to evaluate the coefficient values, then impose a nonlinear component selection operator, and finally a linear reconstruction operator. The only real difference between such a nonlinear approximation method and the linear method from which it is derived is the component selection step - which is a decidedly simple process. In fact, there is very little about the process that is nonlinear at all. To aid in this discussion, we introduce the following terminology:

Definition 1.4.1. *A nonlinear approximation method M is weakly nonlinear if there is an $R - S - T$ factorization for M in which both R and T are linear. An approximation method M is highly nonlinear if it does not admit an $R - S - T$ factorization for M in which both R and T are linear.*

Weakly nonlinear approximation methods have some compelling advantages over highly nonlinear methods:

- Linear decomposition and reconstruction operators typically translate into very fast computer implementations.
- It is relatively easy to compare a given approximant to the best approximation, when the best approximation is available.
- It is easy to compare the properties of a set of weakly nonlinear methods which share common decomposition and reconstruction operators.
- There are many powerful theoretical tools available to study the general properties of algorithms that employ linear operators.

Notwithstanding all of these benefits, there are some problems for which weakly nonlinear methods are unsatisfactory. This may be because the problem being addressed does not have a linear structure which can be exploited easily, or because the most natural way of attacking a problem with linear operators leads to ill-posedness or instability of some kind.

There is very little general theory pertaining to nonlinear methods - especially for highly nonlinear methods.

Chapter 2

The Empirical Mode Decomposition

2.1 Introduction

The empirical mode decomposition (EMD) is a new signal decomposition method developed by Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung and Henry H. Liu. The empirical mode decomposition as laid out in [13] consists of a two stage process. First, the input signal is decomposed into component signals called *intrinsic mode functions* or IMFs using a “sifting algorithm”. Second, the IMFs are used to create analytic signals (by adding their Hilbert transform as a complex conjugate function) whose instantaneous frequencies are used to generate a “Hilbert spectrum”. This “Hilbert spectrum” is used to interpret the original input signal, in a manner similar to a spectrogram, windowed Fourier transform, or wavelet transform. The EMD method seems quite promising, as it yields results which seem very good in cases where more conventional methods (like wavelets) have trouble. However, it is hard to say precisely what is meant by “good results”, as it is unclear what it is the EMD algorithm is actually doing.

In this chapter we present a brief summary of the EMD method, with an emphasis on the motivation behind the elementary definitions of the method. The complete details of the method and its implementation, and the relevant computer code are available in [13] and [10].

2.1.1 Definitions and Notation

“Time” refers to a real-valued variable, denoted by t . Let $g(t)$ be a real-valued function of the variable t . $H\{f\}(t)$ denotes the Hilbert transform of the function $f(t)$. If $f : \mathbb{R} \rightarrow \mathbb{R}$,

then $H\{f\}(t)$ is given by

$$H\{f\}(t) = \frac{1}{\pi} P.V. \int_{-\infty}^{\infty} \frac{f(s)}{t-s} ds,$$

where $P.V.$ denotes the Cauchy principal value of the integral. The Hilbert transform exists for functions in any class L^p (see [23]).

The functions $f(t)$ and $H\{f\}(t)$ are then used to form a complex pair

$$F(t) = f(t) + iH\{f\}(t) = A(t)e^{i\theta(t)}, \text{ where} \quad (2.1)$$

the resulting signal $F(t)$ is called the *analytic signal* of $f(t)$. For more detail on the uses of the analytic signal in time-frequency analysis, see [3].

The *instantaneous frequency* of an analytic signal $F(t)$ is given by

$$\omega_F(t) = \frac{d}{dt}\theta(t), \quad (2.2)$$

where $\theta(t)$ is the phase angle of $F(t)$.

2.1.2 The sifting algorithm

The first stage of the EMD method is a signal decomposition algorithm called the *sifting algorithm*. In this step, the input function is decomposed into component functions called *intrinsic mode functions*, or IMFs. The sifting algorithm is a fairly intuitive process. It is only defined algorithmically - making it fairly simple to implement, and difficult to comment on from a theoretical point of view.

The basic idea of the sifting algorithm is to identify the highest frequency oscillations within a signal $f(t)$ by subtracting a sort of moving average from the input. This moving average is found by interpolating the local maxima of the signal (typically using a spline curve) to construct an upper envelope¹ for $f(t)$, and likewise interpolating the local minima to yield a lower envelope for $f(t)$. The mean envelope (i.e. the mean of the upper and lower envelopes) is then subtracted from $f(t)$, and the same process is applied to the remainder. This ‘‘sifting’’ algorithm is iterated until the mean envelope is approximately zero everywhere (within some prescribed tolerance), and the resulting signal is called the first IMF, or $c_1(t)$. The sifting algorithm is then applied to the remainder $r_1(t) := f(t) - c_1(t)$ to yield the second

¹This is not an envelope in the strict sense, but only an approximation of an envelope. Notwithstanding the standard definition, we will use the term here without any ambiguity to mean the approximate envelope used in [13].

IMF, $c_2(t)$. This process is repeated until the residual is a monotone function:

$$\begin{aligned} f(t) &= c_1(t) + r_1(t) \\ &= c_1(t) + c_2(t) + r_2(t) \\ &\vdots \\ &= \sum_{k=1}^N c_k(t) + r_N(t) \end{aligned}$$

2.1.3 The Hilbert Spectrum

After the sequence of IMFs have been extracted from an input function, it is a simple matter to apply the Hilbert transform to each component, in order to obtain an analytic signal. Given a real-valued signal $x(t)$, the *analytic signal* of x , written as $\mathcal{A}[x]$, is given by:

$$\mathcal{A}[x] = f(t) + \frac{i}{\pi} P.V. \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau$$

P.V. denotes the Cauchy principle value of the integral. For more details on the analytic signal, and its significance in signal processing, the reader is referred to [3]. For simplicity of notation, we will use the corresponding capital letters to denote analytic signals i.e. for a real-valued function $g(t)$, the analytic signal is written as $\mathcal{A}[g](t) = G(t)$. The *instantaneous frequency* $\omega_G(t)$ of an analytic signal $G(t)$ is defined to be the time-derivative of its phase, wherever it exists:

$$\omega_G(t) := \frac{d}{dt} \theta(t),$$

where $\theta(t)$ is the phase angle of $G(t) = A(t)e^{i\theta(t)}$.

After sifting the input to yield a sequence of IMFs, the analytic signals of the IMFs are constructed. The *instantaneous frequency* of the IMFs is then used to study the input $f(t)$. The instantaneous frequency $\omega_F(t)$ of a complex-valued function $F(t)$ is defined as the time-derivative of its phase:

$$\omega_F(t) := \frac{d}{dt} \theta(t), \tag{2.3}$$

where $\theta(t)$ is the phase angle of $F(t) = A(t)e^{i\theta(t)}$.

This simple fact is the key reason that the analytic signal is used in the EMD method: the analytic signal of an IMF allows one to introduce a notion of frequency that is perfectly localized in time. The frequency content of a signal in a way that appears to defy one of the basic notions of signal processing: the uncertainty principle. This is one of the most exciting (and controversial) aspects of EMD. Claims have been made which hint that EMD

may have broken free from the constraint of the uncertainty principle:

Windowed Fourier and wavelet-signal-analysis techniques ...suffer from an uncertainty problem that can limit the accuracy of the frequency (scale for wavelets) and timing information they yield.

:

[EMD] signal separations are not subject to this limitation and provide both accurate frequency and accurate timing simultaneously. [EMD] has this unique advantage over conventional time-frequency analysis techniques[16].

Given the ubiquity of the uncertainty principle throughout signal processing, physics, and other fields which use applied mathematics, it is not surprising that claims like this raise some eyebrows. The fact remains that an EMD analysis of a signal definitely does not lie within the realm of any known setting for which an uncertainty principle applies. This may be because it is unclear what realm or setting EMD actually resides within!

In order to more fully understand exactly what is going on with EMD, it is important to explore the basic properties of the algorithm. While a conventional uncertainty principle may not apply, there is the possibility that some more general type of principle still relates the uncertainties in the time and frequency variables. There is clearly no uncertainty in the instantaneous frequency of a given IMF, but there is very little that is known about how certain one can be about the precision of that IMF. How sensitive is the output (the set of IMFs) to perturbations in the input signal? Is the algorithm stable? Answering these questions would go a long way towards understanding questions about uncertainty as they relate to EMD.

2.2 Research Problems

EMD presents some exciting possibilities. It allows “ugly” signals to be analyzed in a way that yields better looking results than more conventional methods. It appears not to be bound by the uncertainty principle - a principle that characterizes almost every signal processing method invented to date. However, these possibilities will remain unrealized until some important questions have been answered:

- Given that IMFs are not guaranteed to have positive instantaneous frequency, how should the Hilbert spectrum of a set of IMFs be understood? Are the counterexamples found in [20] freak occurrences, or are they likely to affect real-world EMD results?
- Is the current definition of an IMF desirable, or should it be changed to avoid the

problems in [20] - for example, are analytic signals with non-negative instantaneous frequency (ASNIFs) a preferable class of approximants?

- If the current definition of an IMF is chosen, can the sifting algorithm be reformulated in a way that is more theoretically precise?
- In what sense can the sifting algorithm be said to converge? Does the sifting algorithm converge everywhere, or just for a certain class of approximants?
- How are the resulting IMFs affected by perturbations in the input signal?

These are all very significant questions. Until most of them are answered, EMD will not be a viable option for real-world signal processing applications. This is especially true of the last question, regarding error analysis.

Some very interesting progress has been made regarding the nature of the sifting algorithm. In [5], the original sifting algorithm is replaced with a specialized nonlinear diffusion process for estimation of the mean envelope. In the future this may help settle questions of convergence of EMD, as well as error analysis.

In this work, the consequences of using ASNIFs as approximants for nonlinear approximation are explored. Starting by characterizing ASNIFs (or at least a sizeable subclass of them), we then explore various formulations of nonlinear approximation using classes of ASNIFs as approximants, using the intuition gained from their characterization.

In order to construct a technique similar to (or possibly more accurately: inspired by) the empirical mode decomposition, one is faced with a dilemma. Which aspect of EMD is more important to focus on keeping intact: the sifting algorithm, or the non-negative instantaneous frequency property of the components? This choice is an open one, and it is not obvious whether the choices are mutually exclusive. In order to make quick headway without doubling back too often to reconcile these two perspectives, we choose to pursue the second option. Thus, the particular goal of this work is to explore what nonlinear approximation methods we can build which use classes of ASNIFs as the component functions.

An optimization based decomposition technique is the basic framework for this investigation. Rather than delving into questions about the sifting algorithm itself, we replace it with this more well-known decomposition technique. An optimization has the advantage that the set of approximants can be tailored to any list of a priori characteristics by simply constraining the optimization to the desired set of functions. It is therefore a natural way to investigate decomposition algorithms which use ASNIFs as approximants.

The set of ASNIFs is a very large set to use as a starting point. It is probably best to build up to this set gradually. This is also justifiable by virtue of the fact that IMFs are not

representative of the entire set of ASNIFs. IMFs that arise from the sifting algorithm tend to have very little amplitude modulation compared to their instantaneous frequency. This is suggested by the ASNIF condition that the instantaneous frequency resulting from the outer part must be compensated for by an inner factor whose instantaneous frequency is “big enough”. However, the IMFs which EMD produces have much less amplitude modulation than the “worst case” ASNIFs. In fact, most of the IMFs appearing in published EMD applications appear to be very close to constant amplitude functions. (See [13], [14], and [16].)

This investigation into ASNIF-based approximation methods will use only functions with unit modulus as approximants. This has an added benefit of allowing natural comparisons to be made with Fourier analysis. Recall that the traditional Fourier component functions are a subclass of inner functions: functions analytic on the open unit disc, and whose boundary values have modulus 1 almost everywhere. Expanding the set of approximants to a larger class of inner functions is, in a sense, simply relaxing the orthogonality property of the Fourier modes.

Chapter 3

ASNIFs

The definition of an intrinsic mode function is designed to ensure that these signals are narrow-band. Therefore their analytic signals will (hopefully) only have non-negative instantaneous frequency. It has been demonstrated in [20] that there are IMFs which have negative instantaneous frequency. This presents a conundrum: should the definition of an IMF be modified to ensure that the resulting analytic signal has nonnegative instantaneous frequency, or should the expectation of nonnegative instantaneous frequency be relaxed in order to preserve the original definition of an IMF? Another perspective on this issue is to ask the question: what are the tradeoffs involved in using IMFs to perform EMD, as opposed to using components whose analytic signals actually have non-negative instantaneous frequency?

To answer to these questions, it is necessary to investigate some basics regarding EMD. If the definition of an IMF is not sufficient to ensure that the resulting analytic signals are, in fact, *analytic signals with non-negative instantaneous frequency* (or ASNIFs) then how are ASNIFs characterized? If they can be characterized in a sensible way, we may then be able to understand the tradeoffs involved in using IMFs as opposed to using ASNIFs for a signal decomposition method.

In this chapter we consider analytic signals as being boundary value functions of elements of a Hardy space. (They may also be defined for more general settings, i.e. in the sense of distributions.) Periodic analytic signals can be mapped to the unit circle. These functions have natural analytic continuations to the disc which are elements of a Hardy space for the disc. Nonperiodic analytic signals can similarly be extended to elements of a Hardy space for the upper half-plane. These spaces are a natural setting in which to consider analytic signals. They are one of the most general settings in which one might encounter analytic signals, and the body of theory available contains powerful tools. Using the tools from these Hardy spaces, we can obtain a simple yet informative characterization of ASNIFs in the

periodic and nonperiodic case.

Section 1 lays out some definitions, notation, and preliminaries. (The notation used in this chapter may differ slightly from that used in previous chapters.) Section 2 explores the consequences of Hardy spaces on the disc for instantaneous frequency. A characterization of instantaneous frequency is found for sufficiently smooth elements of $H^p(D)$. Parallel results of this analysis are then stated for the Hardy spaces on the upper half plane.

3.1 Preliminaries, Definitions, and Notation

In this section we introduce the notation used and state classical results which are needed to develop our results. Let z denote a complex number, whose real and imaginary parts are given by x and y , respectively. i.e., $z = x + iy$. Whenever polar notation is used, the complex variable z will be expressed as either $z = re^{i\theta}$ or as $z = \rho e^{i\phi}$ to indicate the argument and output value of a function, respectively. i.e. $\rho e^{i\phi} = f(re^{i\theta})$.

D denotes the unit disc $D = \{z : |z| < 1\}$.

\mathbb{C}_+ denotes the upper half of the complex plane: $\mathbb{C}_+ = \{z = x + iy \in \mathbb{C} : y > 0\}$.

$H^p(D)$ denotes the Hardy space of functions f , which are holomorphic on D and for which the following holds:

$$\sup_{1>r>0} \frac{1}{2\pi} \int |f(re^{i\theta})|^p d\theta = \|f\|_{H^p}^p < \infty.$$

$H^p(\mathbb{C}_+)$ ($0 < p < \infty$) denotes the Hardy space of functions f , which are holomorphic on \mathbb{C}_+ and for which the following holds.

$$\sup_{y>0} \int_{-\infty}^{+\infty} |f(x + iy)|^p dx = \|f\|_{H^p}^p < \infty. \tag{3.1}$$

$H^\infty(\mathbb{C}_+)$ denotes the Hardy space of functions f , which are holomorphic and uniformly bounded on \mathbb{C}_+ . The norm is given by

$$\|f\|_{H^\infty} = \sup_{z \in \mathbb{C}_+} |f(z)| < \infty. \tag{3.2}$$

The results found here apply to functions in $H^p(\mathbb{C}_+)$, $0 < p \leq \infty$.

A *Blaschke product* for $H^p(D)$ is a function of the form

$$B(z) = z^m \prod_k \frac{|z_k|}{z_k} \frac{z_k - z}{1 - \overline{z_k}z}. \tag{3.3}$$

Here, k is a nonnegative integer, and $\sum_k(1 - |z_k|) < \infty$.

A *Blaschke product* for $H^p(\mathbb{C}_+)$ is a function of the form

$$B(z) = \left(\frac{z - i}{z + i} \right)^m \prod_k \frac{|z_k^2 + 1|}{z_k^2 + 1} \frac{z - z_k}{z - \bar{z}_k}. \quad (3.4)$$

Here, k is a nonnegative integer, and $\sum_k \frac{y_k}{1 + |z_k|} < \infty$, where $z_k = x_k + iy_k$.

A *singular inner function* for $H^p(D)$ is a function of the form

$$S(z) = \exp \left(- \int_0^{2\pi} \frac{e^{it} + z}{e^{it} - z} d\mu(\theta) \right), \quad (3.5)$$

where $\mu(\theta)$ is a bounded nondecreasing function, and $\mu'(\theta) = 0$ almost everywhere.

A *singular inner function* for $H^p(\mathbb{C}_+)$ is a function of the form

$$S(z) = \exp \left(i \int_{-\infty}^{\infty} \frac{1 + tz}{t - z} d\nu(t) \right), \quad (3.6)$$

for some nondecreasing function $\nu(t)$ of bounded variation on \mathbb{R} with $\nu'(t) = 0$ almost everywhere.

An *outer function* for $H^p(D)$ is a function of the form

$$G(z) = e^{i\gamma} \exp \left(\frac{1}{2\pi} \int_0^{2\pi} \frac{e^{it} + z}{e^{it} - z} \log \psi(t) dt \right), \quad (3.7)$$

where $\gamma \in \mathbb{R}$, $\psi(t) \geq 0$, $\log \psi(t) \in L^1$, and $\psi(t) \in L^p$.

An *outer function* for $H^p(\mathbb{C}_+)$ is a function of the form

$$G(z) = e^{i\gamma} \exp \left(\frac{1}{\pi i} \int_{-\infty}^{\infty} \frac{(1 + tz) \log \psi(t)}{(t - z)(1 + t^2)} dt \right), \quad (3.8)$$

where $\gamma \in \mathbb{R}$, $\psi(t) \geq 0$, $\int_{-\infty}^{\infty} \frac{\log \psi(t)}{1 + t^2} > -\infty$, and $\int_{-\infty}^{\infty} \frac{(\psi(t))^p}{1 + t^2} dt < \infty$.

One well-known result from Hardy spaces will be used. For the spaces $H^p(D)$, we have:

Theorem 3.1.1. *If $f \in H^p(D)$, $p > 0$, and f not identically 0, then $f(z)$ has a unique decomposition*

$$f(z) = B(z)S(z)G(z),$$

where $B(z)$ is a Blaschke product in $H^p(D)$, and $S(z)$ is a singular inner function in $H^p(D)$, and $G(z)$ is an outer function in $H^p(D)$.

On the half plane, an analogous result holds:

Theorem 3.1.2. *If $f(z) \in H^p(\mathbb{C}_+)$, $p > 0$, then $f(z)$ has a unique decomposition*

$$f(z) = e^{i\alpha z} B(z) S(z) G(z), \quad (3.9)$$

where $\alpha \geq 0$, $B(z)$ is a Blaschke product, $S(z)$ is a singular function, and $G(z)$ is an outer function in $H^p(\mathbb{C}_+)$.

For further details about Blaschke products, singular inner functions, and outer functions for the classes $H^p(D)$ and $H^p(\mathbb{C}_+)$, and for the canonical factorization theorem, see [9, 11].

3.2 Instantaneous Frequency of Analytic Signals

Definition 3.2.1. *Time will denote the arc-length parameterization of the boundary of the Hardy domain in question. i.e. for $f \in H^p(\mathbb{C}_+)$ time denotes the real valued coordinate $x = \Re(z)$; for $f \in H^p(D)$, time is given by the angle $\theta = \text{Arg}(z)$ (ignoring for now the optional multiples of 2π).*

Definition 3.2.2. *The instantaneous frequency of a complex valued function f is defined to be the time-derivative of its phase, whenever it exists. It is referred to here as $\omega_f(*)$, or as $\omega[f](*)$. This may also be written in terms of the complex logarithm:*

$$\omega_f(z) = \frac{d}{dt} \text{Im}(\text{Log}(f)).$$

For $H^p(D)$, time is parameterized by $t = \theta$; for $H^p(\mathbb{C}_+)$, time is parameterized by the real axis $t = x$.

Note that this expression may be evaluated on the interior of the Hardy domain, as well as on the boundary. To avoid questions about what the time parameter t means on the interior of the domain, we only consider the meaning of this expression on the boundary.

The following result applies to instantaneous frequency of a complex valued function f independent of what space f is in.

Theorem 3.2.3. *If f is analytic on an open set $O \subset \mathbb{C}$, and has a representation $f = \prod_n g_n(z)$ on O with convergence in the sense of uniform convergence on compacta of the partial products, then the series representation*

$$\omega_f(z) = \sum_n \omega_{g_n}(z)$$

converges pointwise on O , and uniformly on compacta in $O \setminus \{z : f(z) = 0\}$.

Proof. Let A be a compactum contained in $O \setminus \{z : f(z) = 0\}$. Let $f_m = \prod_{n=1}^m g_n$ on A . Clearly $\text{Log}(f_m)$ is analytic on A , and

$$\text{Log}(f_m) \rightarrow \text{Log}(f) + 2k\pi i \quad \text{for some } k \in \mathbb{Z}$$

uniformly on A . The result then follows from the Analytic Convergence Theorem. \square

Corollary 3.2.4. *If $f(z) = Cg(z)$, C constant, then $\omega_f(z) = \omega_g(z)$.*

Note that these results apply only on the open set $O \setminus \{z : f(z) = 0\}$. ($O = D$ or \mathbb{C}_+ as appropriate.) Establishing strong results for the boundary values is beyond the scope of this work. The boundary value results used here require that f belong to some smoothness class within H^p .

3.2.1 Periodic Analytic Signals

Using the Cauchy-Riemann equations and Definition 3.2.2, we have

$$\begin{aligned} \lim_{|z| \rightarrow 1^-} \omega_f(z) &= \lim_{|z| \rightarrow 1^-} \text{Im} \left(\frac{\partial}{\partial \theta} \text{Log} f(z) \right) \\ &= \lim_{|z| \rightarrow 1^-} \text{Im} \left(iz \frac{d}{dz} \text{Log} f(z) \right) \\ &= \lim_{|z| \rightarrow 1^-} \text{Re} \left(z \frac{f'(z)}{f(z)} \right). \end{aligned}$$

(The limits here are understood to be taken non-tangentially. See [9] and [11].) This limit does not necessarily converge for $f \in H^p$. General conditions for boundary convergence of the first derivative are still an open question for the Hardy spaces. To avoid these questions, further conditions will be imposed when this formula is used to ensure convergence.

We restrict ourselves to functions $f \in H^p(D)$ for which the boundary value function $f(\theta)$ is differentiable, and satisfies

$$\omega_f(\theta) = \lim_{|z| \rightarrow 1^-} \omega_f(z). \tag{3.10}$$

The sufficient conditions used to guarantee this are given in Theorem 3.2.9.

Lemma 3.2.5. *The instantaneous frequency of a finite Blaschke product is non-negative on ∂D .*

This result is not new (see [8]). It is easily shown for Blaschke products that the instantaneous frequency function $\omega_B(\theta)$ is given by a summation of Poisson kernels:

$$\omega_B(\theta) = \sum_n \frac{1 - r_n^2}{r_n^2 - 2r_n \cos(\theta_n - \theta) + 1} > 0.$$

Here $z_n = r_n e^{i\theta_n}$ are the zeros of $B(z)$. A proof is shown below, as the setting of the calculations and the notation in [8] are quite different, and also to illustrate how instantaneous frequency can be defined on the interior of D .

Proof. The instantaneous frequency of a finite Blaschke product $B(z) \in H^p(D)$ is given by

$$\begin{aligned}\omega_B(z) &= \omega \left[z^m \prod_n^N \frac{\bar{z}_n}{|z_n|} \frac{z_n - z}{1 - \bar{z}_n z} \right] (z) \\ &= \omega[z^m](z) + \sum_n^N \omega \left[\frac{z_n - z}{1 - \bar{z}_n z} \right] (z) \\ &= m + \sum_n^N \operatorname{Re} \left[z \left(\frac{1}{z - z_n} + \frac{\bar{z}_n}{1 - \bar{z}_n z} \right) \right]\end{aligned}$$

Observe that allowing z_n to be 0 no longer presents a problem, so there is no need to isolate the instantaneous frequency of the z^m term. Thus, incorporating the zeros at the origin into the list $\{z_n\}_{n=1}^M$, $M = N + m$ gives

$$\begin{aligned}\omega_B(z) &= \sum_{n=1}^M \operatorname{Re} \left[z \left(\frac{1 - \bar{z}_n z + z \bar{z}_n - r_n^2}{z - z_n - \bar{z}_n z^2 + r_n^2 z} \right) \right] \\ \omega_B(z) &= \sum_{n=1}^M \operatorname{Re} \left[\frac{1 - r_n^2}{1 - \frac{z_n}{z} - \bar{z}_n z + r_n^2} \right] \\ \omega_B(z) &= \sum_{n=1}^M \frac{1 - r_n^2}{1 - 2\operatorname{Re} \left[\frac{z_n}{z} \right] + r_n^2} \\ \omega_B(z) &= \sum_{n=1}^M \frac{1 - r_n^2}{1 - 2\frac{r_n}{r} \cos(\theta_n - \theta) + r_n^2} > 0\end{aligned}\tag{3.11}$$

$\omega_B(z)$ is defined for all points in D , except at the points $\{z_n\}$. For finite Blaschke products, the boundary values can be evaluated directly. (The result also extends directly to infinite Blaschke products for which the cluster points of the zero set $\{z_n\}$ have measure zero on ∂D .)

$$\begin{aligned}\omega_B(\theta) &= \lim_{|z| \rightarrow 1^-} \omega_B(z) = \omega_B(z)|_{|z|=1} \\ &= \sum_n \frac{1 - r_n^2}{r_n^2 - 2r_n \cos(\theta_n - \theta) + 1} > 0\end{aligned}$$

□

Lemma 3.2.6. *For a singular function $S(z)$, whose singular support $\text{supp}(d\nu)$ is a discrete set, instantaneous frequency is non-negative almost everywhere on ∂D .*

Proof. The instantaneous frequency of a singular function in $H^p(D)$ is given by

$$\begin{aligned}
\omega_S(z) &= \text{Re} \left(z \frac{S'(z)}{S(z)} \right) \\
&= \text{Re} \left(- \int_{-\pi}^{\pi} \frac{2ze^{iu}}{(e^{iu} - z)^2} d\mu(u) \right) \\
&= - \int_{-\pi}^{\pi} \frac{2\text{Re}(ze^{-iu} - 2|z|^2 + \bar{z}|z|^2 e^{iu})}{|e^{iu} - z|^4} d\mu(u) \\
&= \int_{-\pi}^{\pi} \frac{4|z|^2 + 2(1 + |z|^2)|z|\cos(\theta - u)}{|e^{iu} - z|^4} d\mu(u) \tag{3.12}
\end{aligned}$$

If $\text{supp}(d\nu)$ is a discrete set, then boundary limits can be taken almost everywhere, and we have:

$$\begin{aligned}
\omega_S(\theta) &= \lim_{|z| \rightarrow 1^-} \int_{-\pi}^{\pi} \frac{4|z|^2 + 2(1 + |z|^2)|z|\cos(\theta - u)}{|e^{iu} - z|^4} d\mu(u) \\
&= \int_{-\pi}^{\pi} \frac{4(1 + \cos(\theta - u))}{(|e^{iu}|^2 + |z|^2 - 2|e^{iu}z|\cos(\theta - u))^2} d\mu(u), \quad z \notin \text{supp}(\mu) \\
&= \int_{-\pi}^{\pi} \frac{4(1 + \cos(\theta - u))}{(2 - 2\cos(\theta - u))^2} d\mu(u), \quad z \notin \text{supp}(\mu) \\
\omega_S(\theta) &= \int_{-\pi}^{\pi} \frac{1 + \cos(\theta - u)}{(1 - \cos(\theta - u))^2} d\mu(u) \geq 0, \quad z \notin \text{supp}(\mu) \tag{3.13} \\
\omega_S(\theta) &\geq 0 \quad a.e.
\end{aligned}$$

□

Lemma 3.2.7. *An outer function $G(z)$ which has a meromorphic continuation to some open set S containing $D \cup \partial D$ has mean instantaneous frequency given by:*

$$\int_0^{2\pi} \omega_G(\theta) d\theta = \pi \sum_{z_m \in \partial D} \text{Re} \left[\text{Res} \left(\frac{G'(z)}{G(z)}; z_m \right) \right].$$

Proof. The existence of a meromorphic continuation for $G(z)$ ensures that $\omega_G(z)$ is well-defined almost everywhere on ∂D . The result then follows from the Residue Theorem. □

Corollary 3.2.8. *An outer function $G(z)$ which is non-zero on ∂D , and which has an analytic continuation to some open set S containing $D \cup \partial D$ has zero mean instantaneous frequency. Particularly, any non-constant outer function which has such an analytic continuation has negative instantaneous frequency at some point on ∂D .*

Theorem 3.2.9. *If $f \in H^p(D)$, and f has a meromorphic continuation to some open set S containing $D \cup \partial D$, then $\omega_f(\theta)$ can be decomposed into*

$$\omega_f(\theta) = \omega_B(\theta) + \omega_S(\theta) + \omega_G(\theta),$$

where B , S , and G are determined by the canonical factorization of f .

Proof. The existence of a meromorphic continuation is a very strong condition. It implies that the singular support $\text{supp}(d\nu)$ of $S(z)$ is discrete, and also that the Blaschke product is finite. The result then follows from the Canonical Factorization Theorem and Theorem 3.2.3. \square

Corollary 3.2.10. *A signal satisfying the conditions of Theorem 3.2.9 is an ASNIF if and only if*

$$-\omega_G(t) \leq \omega_B(t) + \omega_S(t) \quad \forall t$$

3.2.2 Nonperiodic Analytic Signals

Nonperiodic analytic signals are boundary functions of elements of a Hardy space for the upper half-plane $H^p(\mathbb{C}_+)$. The space $H^p(\mathbb{C}_+)$ shares many characteristics with the spaces $H^p(D)$. (The most obvious difference is that non-zero constant functions are not elements of $H^p(\mathbb{C}_+)$, while they are elements of $H^p(D)$.) Many of the proofs in this section are very similar to the results for $H^p(D)$. These proofs are in the appendix. Those proofs which differ significantly from the results for $H^p(D)$ have their proofs stated here.

We restrict ourselves to functions $F(z) \in H^p(\mathbb{C}_+)$ which are “nice” on the boundary (as a result of having a meromorphic continuation across the boundary), giving:

$$\omega_F(x) = \lim_{\text{Im}(z) \rightarrow 0^+} \text{Im} \left(\frac{F'(z)}{F(z)} \right). \quad (3.14)$$

This is not true in general; the boundary value function $F(x)$ may not be differentiable, meaning the left hand side of equation 3.14 is not defined. In order to use equation 3.14, it is sufficient to assume that $F(z)$ has a meromorphic continuation to an open domain containing $\mathbb{R} \cup \mathbb{C}_+$.

The canonical factorization theorem (Theorem 3.1.2) for $H^p(\mathbb{C}_+)$ is slightly different from that for $H^p(D)$. (See [11] and [9].)

Following the terminology for $H^p(D)$, we will refer to the parts of instantaneous frequency in $H^p(\mathbb{C}_+)$ given by the factors in equation 3.9 as the Fourier, Blaschke, singular, and outer parts. Under similar conditions to $H^p(D)$, the instantaneous frequency of the Fourier,

Blaschke, and singular parts in $H^p(\mathbb{C}_+)$ are all non-negative on \mathbb{R} . There are some differences that are worth noting:

1. It is sufficient to require that the zero set $\{z_n\}$ of the Blaschke product be a discrete set (i.e. $\{z_n\}$ has no cluster points). Because the zero set $\{z_n\}$ has no cluster points on \mathbb{R} , $\omega_B(z)$ is well defined on \mathbb{R} . Note that in the case of $H^p(D)$, the discreteness condition implies finiteness of the Blaschke product. Thus, it is more appropriate to think of discreteness as the principle which gives the result.
2. Similarly, it is sufficient to require that the singular support $\text{supp}(d\nu)$ be a discrete set.

Under these conditions we have

$$\omega[e^{i\alpha z}](x) = \alpha \geq 0 \quad (3.15)$$

$$\omega_B(x) = \sum_n \frac{2}{y_n} \frac{1}{\left(\frac{x_n-x}{y_n}\right)^2 + 1} \geq 0 \quad (3.16)$$

$$\omega_S(x) = \int_{-\infty}^{\infty} \frac{u^2 + 1}{(u-x)^2} d\nu(u) \geq 0 \quad (3.17)$$

When evaluating $\omega_B(x)$ above, we admit the possibility that $z_n = i$. Proofs of equation 3.16 and 3.17 are in Appendix 8.

For outer functions in $H^p(\mathbb{C}_+)$, we have the following result.

Theorem 3.2.11. *Let $G(z)$ be an outer function in $H^p(\mathbb{C}_+)$ which has a meromorphic continuation to some open set S containing $\mathbb{C}_+ \cup \mathbb{R}$, and for which $\text{Res}(G'(z)/G(z); \infty) = 0$.*

Then

$$P.V. \int_{\mathbb{R}} \omega_G(x) dx = \pi \sum_{z_k \in \mathbb{R}} \text{Re} \left[\text{Res} \left(\frac{G'(z)}{G(z)}; z_k \right) \right].$$

Proof. Let $G(z)$ be an outer function in $H^p(\mathbb{C}_+)$ which has a meromorphic continuation to some open set S containing $\mathbb{C}_+ \cup \mathbb{R}$. Let $\{z_k\}$ be the set of points in $\mathbb{C}_+ \cup \mathbb{R}$ at which $G(z)$ is not analytic. In fact, $\{z_k\} \subset \mathbb{R}$. Thus, G is analytic on $\mathbb{C}_+ \setminus \{z_k\}$, which is contained in S .

Let Γ_n be the boundary of the set $(B(0, n) \cap \mathbb{C}_+) \setminus (\cup_k B(z_k, \frac{1}{n}))$, where $B(p, r) = \{z \in \mathbb{C} : |z - p| < r\}$. Then

$$\int_{\Gamma_n} \omega_G(z) dz = \text{Im} \int_{\Gamma_n} \frac{G'(z)}{G(z)} dz = 0$$

for any n . Letting $n \rightarrow \infty$, we have:

$$\begin{aligned}
0 &= P.V. \int_{\mathbb{R}} \omega_G(x) dx - \text{Im} \left(\pi i \text{Res} \left(\frac{G'(z)}{G(z)}; \infty \right) + \pi i \sum_k \text{Res} \left(\frac{G'(z)}{G(z)}; z_k \right) \right) \\
P.V. \int_{\mathbb{R}} \omega_G(x) dx &= \text{Re} \pi \left(\text{Res} \left(\frac{G'(z)}{G(z)}; \infty \right) + \sum_k \text{Res} \left(\frac{G'(z)}{G(z)}; z_k \right) \right) \\
&= \pi \sum_{z_k \in \mathbb{R}} \text{Re} \left[\text{Res} \left(\frac{G'(z)}{G(z)}; z_k \right) \right]
\end{aligned}$$

□

Corollary 3.2.12. *An outer function $G(z)$ which is non-zero on \mathbb{R} , and which has an analytic continuation to some open set S containing $\mathbb{C}_+ \cup \mathbb{R}$ has zero mean instantaneous frequency. Particularly, any outer function which is non-zero on \mathbb{R} which has such an analytic continuation has negative instantaneous frequency at some point $x_0 \in \mathbb{R}$.*

For nonperiodic signals, an analogous result to 3.2.9 then follows:

Theorem 3.2.13. *If $f \in H^p(\mathbb{C}_+)$, and f has a meromorphic continuation to some open set S containing $\mathbb{C}_+ \cup \mathbb{R}$, then $\omega_f(t)$ can be decomposed into*

$$\omega_f(t) = \omega[e^{i\alpha t}] + \omega_B(t) + \omega_S(t) + \omega_G(t),$$

where B , S , and G are determined by the canonical factorization of f .

Proof. The existence of a meromorphic continuation ensures that $\omega[e^{i\alpha t}]$, $\omega_B(z)$, $\omega_S(z)$, and $\omega_G(z)$ all exist and are well-defined on $\mathbb{C}_+ \cup \mathbb{R}$. The result then follows from the Canonical Factorization Theorem and Theorem 3.2.3. □

Corollary 3.2.14. *A signal satisfying the conditions of Theorem 3.2.13 is an ASNIF if and only if*

$$-\omega_G(t) \leq \alpha + \omega_B(t) + \omega_S(t) \quad \forall t$$

3.3 Constructing ASNIFs

In recent literature ([24], [8]) it is of particular interest how, given an amplitude function $a(x)$, one can construct an analytic signal with non-negative instantaneous frequency (ASNIF) with prescribed amplitude. Using the results found here, we can give a more complete answer to this problem. In this section, we only give each result for only one of $H^p(\mathbb{C}_+)$ and $H^p(D)$. Analogous results are readily available with appropriate modification.

From Theorem 3.1.2 it is clear that an analytic signal $F(x)$ has amplitude $|F(x)| = a(x)$ if and only if its outer factor $G(z)$ is equal to

$$G(z) = \exp \left\{ \frac{1}{i\pi} \int_{-\infty}^{\infty} \frac{(1+uz) \log a(u)}{(u-z)(1+u^2)} du \right\} \quad (3.18)$$

Thus, choosing a given amplitude function $a(x)$ completely determines the outer factor of any ASNIF $F(x)$ for which $|F(x)| = a(x)$. (In the rest of this section, any reference to $G(z)$ or $G(x)$ will refer to the function given by equation 3.18. i.e. the outer function with amplitude equal to $a(x)$.)

We are then free to choose three other factors in the canonical factorization as needed in order to ensure that $F(z) = e^{i\alpha z} B(z) S(z) G(z)$ has non-negative frequency on the whole real line. It remains to be shown under what conditions this is possible.

3.3.1 Outer Functions with Locally Bounded Instantaneous Frequency

The easiest outer function to make into an ASNIF is one which satisfies the hypotheses of Theorem 3.2.11. For such an outer function, instantaneous frequency is negative somewhere, but it is locally bounded for all x . If it is uniformly bounded (by some M), all that is needed is a factor of the form e^{iMx} , giving $F(x) = e^{iMx} G(x)$ as an ASNIF with the required amplitude.

This is not the only way to construct an ASNIF $F(x)$, only the simplest. Multiplying two complex signals is equivalent to the pointwise addition of their instantaneous frequencies. Singular inner functions or Blaschke products may be used instead of e^{iMx} . There are many such choices available, and many different criteria that one might use to make that choice. Those criteria are not dictated by the theoretical results presented here. That choice would best be governed by application-specific concerns (such as minimizing the total variation of phase, for example).

If $\omega_G(x)$ is locally bounded but not uniformly bounded, we need to use an inner function $I(z)$ whose instantaneous frequency $\omega_I(x)$ is everywhere greater than $-\omega_G(x)$ ¹. Such a function can easily be shown to exist. One way is to iteratively “build” a Blaschke product or singular function by adding zeros or adding point masses to the singular measure, respectively, so that $\omega_B(x)$ or $\omega_S(x)$ is large enough on each successive pair of intervals $[-(k+1), -k] \cup [k, k+1]$. This can be done by a fairly simple induction.

¹This is both necessary and sufficient, as $\omega_I(x) \geq 0$ by definition. Thus $\omega_I(x) \geq -\omega_G(x) \Leftrightarrow \omega_I(x) \geq \min(0, -\omega_G(x))$

3.3.2 Outer Functions whose Instantaneous Frequency is not Locally Bounded

For signals with non-locally bounded instantaneous frequency, we look at signals from $H^p(D)$. If the outer function $G(z)$ has zeros and/or poles on ∂D , then $\omega_G(t)$ is not locally bounded. Even so, it may still be possible to construct an inner function $I(z)$ such that the analytic signal $I(t)G(t)$ has nonnegative instantaneous frequency almost everywhere.

We restrict ourselves to signals $G(t)$ for which $G(z)$ has an analytic continuation to some open set O containing $D \cup \partial D$, and for which the function $G'(z)/G(z)$ has a finite number of zeros and poles on ∂D . For such signals, $G(z)$ can be factorized into the form

$$G(z) = G_1(z)H(z), \quad (3.19)$$

where $G_1(z)$ is an outer function which satisfies Corollary 3.2.8, and $H(z)$ is of the form $H(z) = \prod_{k=1}^N (z - z_k)^{p_k}$, $N < \infty$, $p_k \in \mathbb{Z}$. ($H(z)$ cannot have any essential singularities as this would cause $G(z)$ not to be an outer function. See [9, 11].) We need only concern ourselves with the instantaneous frequency of $H(z)$, as $G_1(z)$ was addressed in the previous section. We then have

$$\begin{aligned} \omega_H(t) &= \sum_{k=1}^N \omega[(z - x_k)^{p_k}](t) \\ &= \sum_{k=1}^N p_k \pi \cdot \delta(t - x_k) = 0 \quad a.e. \end{aligned}$$

It is not clear how a jump in phase relates to instantaneous frequency. It is tempting to simply treat these jumps as being components of a “weak” instantaneous frequency. Upon closer inspection, it is clear that the value of a phase jump is not well-defined. A (non-zero) phase jump value is only unique up to addition of $2n\pi$, $n \in \mathbb{Z}$. Thus, it is not clear what is meant by a positive (or negative) jump in instantaneous frequency. If a more precise framework for treating phase jumps is desired, one might be tempted to define the phase jump as the non-tangential limit of phase jump as one approaches the point in question. However, this quantity is not well-defined by simply “taking the non-tangential limit of the phase jump”.²

We simply ignore the phase jumps, as there is currently no clear way to reconcile them

²There are problems with even stating the idea (hence the quotation marks) because there are two limits involved: one in evaluating the jump, and one in taking the non-tangential limit. Obviously, evaluating the jump first gives a value of 0. Taking the non-tangential limit first gives problems: the phase jumps for discontinuous inner functions are either not well-defined or equal to $-\infty$, depending on the order in which the limits are evaluated.

with the concept of instantaneous frequency. This simplifies things greatly, as $\omega_H(t) = 0$ at every other point. We continue with this simplified analysis, but it is an open question how the phase jumps should be properly understood.

We now have everything we need to construct an ASNIF whose amplitude is given by an outer function $G(z)$ which has a suitable meromorphic continuation. For any such $G(z)$, find the factorization $G(z) = G_1(z)H(z)$ prescribed in equation 3.19. Let $B^*(t)$ be a Blaschke product such that $\phi_{B^*}(t) \geq |\phi_{G_1}(t)|$. Then one ASNIF with outer part G is given by

$$F(t) = B^*(t)G(t)$$

3.4 Conclusions

It is important to note once more that the results presented in this chapter only apply to a fairly exclusive subclass of analytic signals: those elements of $H^p(D)$ or $H^p(\mathbb{C}_+)$ which admit an analytic continuation which is “nice enough.” The results as stated do not apply to the entire Hardy space. There is significant work required to establish results which properly address the questions of convergence for signals where the boundary values of the first derivative (and thus the concept of instantaneous frequency itself) are not readily available. However, for analytic signals which are generated by real-world time series data the results of this chapter are often sufficient to conduct a meaningful analysis.

As progress is made towards a complete characterization of ASNIFs, we see more clearly the parallels and differences between how one should understand “instantaneous frequency” (IF) and the classical concept of “Fourier frequency” (FF).

The components of instantaneous frequency in the IF setting have natural parallels in the FF setting. Blaschke products for $H^p(D)$ are the natural generalizations of the individual modes of the Fourier series³. With this perspective, inner functions are a single family of “pure instantaneous frequency” functions which is a natural generalization of the “pure Fourier frequency” functions for periodic and non-periodic signals.

The notion of what a “pure amplitude” function should be illustrates a divergence between the IF and FF perspectives. In the FF understanding, an amplitude function is understood to be purely real (and non-negative). In the IF world, amplitude functions correspond to outer functions, which are never purely real valued⁴. In fact, they always have

³The Fourier modes for periodic functions are in fact Blaschke products which have only one zero at the origin. For nonperiodic signals, the Fourier transform modes correspond to singular inner functions which have only one point mass at infinity. “At infinity” is used loosely here. The meaning is clear when looking at a conformal mapping of $D \rightarrow H$ given by the Cayley transform. The singular inner functions for D which have a point mass at 1 become the Fourier modes $e^{i\omega t}$.

⁴In the $H^p(D)$ setting, outer functions may be constant. Thus $G(z) = c \in \mathbb{R}$ is an outer function for

non-zero instantaneous frequency if they have any amplitude modulation. At first glance, this seems like a stark contrast between the IF and FF perspectives. A more careful read on the situation brings to light a surprising parallel: Outer functions (that are reasonably nice) have zero mean instantaneous frequency; The Fourier transform of a purely amplitude modulated (AM) signal has symmetric support. This leads to analogous procedures for making signals with positive frequency in both the IF and FF realms. In the FF realm, a band limited signal is made to have positive Fourier frequency by modulating it with a carrier frequency (a pure FF function) high enough to shift the Fourier frequencies above zero. (This is the well known procedure for making an analytic signal out of an arbitrary band-limited signal.) In the IF realm, an analytic signal is made to have positive instantaneous frequency by modulating it with an inner function (a pure IF function) appropriately chosen to shift the instantaneous frequencies above zero. The analogy is quite clear: ASNIFs are to analytic signals as analytic signals are to “left band-limited signals”⁵.

3.5 Consequences for an “EMD-like” ASNIF decomposition

In order to construct a technique similar to (or possibly more accurately: inspired by) the empirical mode decomposition, one is faced with a dilemma. Which aspect of EMD is more important to focus on keeping intact: the sifting algorithm, or the non-negative instantaneous frequency property of the components?

If the property of non-negative frequency is the most important characteristic for the component set of an EMD-like decomposition method, the results of this chapter have some significant consequences for the design and implementation of such an algorithm. We will use the term *ASNIF decomposition* to denote an algorithm which decomposes an input signal into a sequence of component functions which are each ASNIFs, as well as a residual (which may not be an ASNIF).

A general ASNIF decomposition algorithm would take an analytic signal $F(t)$ as an input,

$H^p(D)$. In $H^p(\mathbb{C}_+)$, outer functions may not be constant, ruling out this possibility. In either case, there are no non-trivial real-valued outer functions.

⁵That is to say, signals whose Fourier transform vanishes for $\omega < K \in \mathbb{R}$

and decompose it into a series of ASNIFS iteratively. i.e.

$$\begin{aligned}
 F(t) &= C_1(t) + R_1(t) \\
 &= C_1(t) + C_2(t) + R_2(t) \\
 &\quad \vdots \\
 &= \sum_{j=1}^N C_j(t) + R_N(t),
 \end{aligned}$$

where $C_j(t)$ are ASNIFSs, and R_N is the N^{th} residual. (In order to apply such a method to a signal which is not analytic, we must first construct the analytic signal using the Hilbert transform.) The component selection may be done any number of ways. One very general way to represent a component selection step is to specify a functional $J(u, v)$, whose minimizer (selected over all possible components u) yields the best component u , and the residual v . The choice of functional then determines the decomposition.

This functional must be constrained to the set of candidate component functions. The entire set of ASNIFSs is a very large set, for the purposes of constrained optimization. Instead of attacking this large problem head on, we will look at a simpler class of ASNIFSs which should still offer some insight into what an ASNIFS decomposition does. The simplest class of ASNIFSs whose span generates the entire space L^2 is the set of Fourier modes. Using this set for approximation would yield approximation methods which don't differ radically from the Fourier series. Clearly a larger set is needed in order to gain any insight into an ASNIFS decomposition. A natural generalization of the set of Fourier modes has been suggested in section 3.4. In the classical setting, the Fourier modes represent "pure frequency"; in the instantaneous frequency setting, inner functions represent "pure instantaneous frequency". Thus, it is of interest to investigate what approximation methods arise when we use inner functions as approximants.

In practice, one typically only looks at finitely sampled signals, as computers cannot represent continuous functions directly. Thus, we only need to consider a class of inner functions which is general enough to include each finitely sampled inner function. A classical interpolation result for Hardy spaces states that any inner function sampled at n points may be interpolated by a finite Blaschke product of degree $\leq n$. (See [9], [11].) It is sufficient to use the set of finite Blaschke products to investigate the problem of approximation using inner functions as approximants. This simplification is also available if one makes the assumption that the analytic signals being investigated are continuously differentiable. The existence of a complex derivative precludes any singular inner factors, infinite Blaschke products, or singularities in the outer factor.

Chapter 4

An ASNIF-based Signal Decomposition

4.1 Introduction

EMD decomposes signals iteratively. The algorithm selects the “next” IMF from the previous residual independent of the previous IMFs selected, and also independent of the IMFs yet to be selected. While the choice of the n^{th} IMF depends implicitly on the previous $n - 1$ IMFs (so that it would be incorrect to say that the IMF choices are independent of each other), the sifting algorithm itself is blind to the previous IMFs which have been selected in the course of a particular decomposition; the output of the sifting algorithm depends only on the input (and the stopping criteria).

As a surrogate for the sifting algorithm, we will construct an optimization based decomposition which will decompose an input signal into a component ASNIF, and a residual. This approach is chosen in order to investigate what happens when one tries to implement the basic idea of EMD - decomposing a signal into components with positive instantaneous frequency - while strictly enforcing the requirement that the component functions have positive instantaneous frequency. This requirement may be enforced easily using a constrained optimization wherein the components are restricted to a desired class of functions: ASNIFs, or some subset of the set of ASNIFs.

It bears repeating that this is not the only way to construct a signal processing algorithm which captures the ideas of EMD. One might opt to preserve the basic idea of the sifting algorithm, while abandoning the strict positive instantaneous frequency requirement for the component functions. Exciting work has been done in this direction using a nonlinear diffusion process instead of the sifting algorithm (see [5]). This research direction is vital to understanding exactly what the sifting algorithm and its many variants are actually doing.

After all, it is clear that *something* interesting is happening, even if positive instantaneous frequency is not strictly enforced by these algorithms. However, this is not the question we are investigating in this work.

Definition 4.1.1. *We will use the notation $\|\cdot\|_{\mathcal{V}}$ to mean a (nonnegative, real-valued) weight function with domain \mathcal{V} . This might be a norm, seminorm, or some other non-negative function.*

There is no risk of confusion in redefining the conventional notation for a norm; the only cases for which $\|\cdot\|_{\mathcal{V}}$ is redefined to be something other than a norm are when \mathcal{V} does not have a norm. This seems appropriate, as $\|\cdot\|_{\mathcal{V}}$ will act as a surrogate for a norm when none is available.

We endeavor to develop an algorithm which adaptively decomposes a function $f \in \mathcal{X}$ into component functions $u_j \in \mathcal{Y} \subset \mathcal{X}$ so that

$$f = \sum_{j=0}^n u_j + v_n,$$

where v_n is a residual, which is small (in some sense). Component selection is done using the algorithm

$$\begin{aligned} v_{-1} &:= f \\ u_j &:= \operatorname{argmin}_{\in \mathcal{Y}} J(v_{j-1}, \cdot, \lambda_j), \quad j = 0, 1, 2, \dots, n \end{aligned} \tag{4.1}$$

where J is a functional of the form

$$J(f, u, \lambda) = \lambda \|f - u\|_{\mathcal{X}} + \|u\|_{\mathcal{Y}}. \tag{4.2}$$

A completeness result is needed for (4.1) to be meaningful. The specific form of this result will depend on the nature of \mathcal{X} , \mathcal{Y} , and the weights $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{Y}}$. Uniqueness of the minimizer in (4.1) is also an important question if a complete characterization of this technique is desired. Fortunately, uniqueness is not necessarily required in order for this algorithm to yield a useful decomposition. It is of course required to guarantee a unique decomposition. However, given the high degree of redundancy in the sets of approximants we will use, it is not yet clear if uniqueness is desirable or even attainable in most circumstances.¹

¹Whenever a uniqueness result is easily obtainable, we will use it. When we look at larger approximant sets, we will not be focused on questions of uniqueness for the two reasons stated: uniqueness results are harder to come by, and it is not clear that uniqueness is a reasonable expectation. This is an issue that is important for future research. However, it falls outside the scope of this work.

After k steps, this yields a partial decomposition of f :

$$\begin{aligned}
f &= u_0 + v_0 \\
&= u_0 + u_1 + v_1 \\
&\quad \vdots \\
&= u_0 + \cdots + u_k + v_k
\end{aligned} \tag{4.3}$$

This gives a decomposition $f \approx u_1 + \cdots + u_k$, with a residual v_k . As k increases, the decomposition captures more and more of the behavior that $\|\cdot\|_{\mathcal{Y}}$ measures.

In cases where \mathcal{Y} is a subspace of \mathcal{X} , the J -functional technique yields decompositions which lie in interpolation spaces between \mathcal{Y} and \mathcal{X} . Such an interpolation space method is often referred to as a ‘ $(\mathcal{Y}, \mathcal{X})$ decomposition’. In order to avoid unnecessary abuse of notation, we will refer to any method for which \mathcal{Y} is a subset of \mathcal{X} , but not a subspace of \mathcal{X} , as a ‘ $(\mathcal{Y} \subset \mathcal{X})$ decomposition’.

Using this framework, we will develop tools for adaptive nonlinear decompositions. Further assumptions about \mathcal{X} , \mathcal{Y} , and J will be made when appropriate.

4.2 An energy estimate for the $(\mathcal{Y} \subset \mathcal{X})$ decomposition

Let us compare the decomposition of $v_j = u_{j+1} + v_{j+1}$, which comes from the minimizer of $J(v_j, \cdot, \lambda_{j+1})$, with the trivial decomposition $v_j = 0 + v_j$. If $0 \in \mathcal{Y}$, and $\|0\|_{\mathcal{Y}} = 0$, we have

$$\lambda_{j+1} \|v_{j+1}\|_{\mathcal{X}} + \|u_{j+1}\|_{\mathcal{Y}} \leq \lambda_{j+1} \|v_j\|_{\mathcal{X}}. \tag{4.4}$$

If $0 \notin \mathcal{Y}$, we can still obtain the same result if some further conditions are satisfied.

Lemma 4.2.1. *If $\exists \{m_n\}_{n=1}^{\infty} \subset \mathcal{Y}$ such that:*

1. $\|m_n\|_{\mathcal{Y}} \rightarrow 0$, and
2. $\|g + m_n\|_{\mathcal{X}} \rightarrow \|g\|_{\mathcal{X}} \forall g \in \mathcal{X}$

then (4.4) holds at the minimizer of $J(v_j, \cdot, \lambda_{j+1})$.

Proof. Compare the value of the J functional at the optimizer $(u_j, u_{j+1}, \lambda_{j+1})$ with the points $(u_j, m_n, \lambda_{j+1})$, $m = 1, 2, \dots$:

$$\begin{aligned}
J(u_j, u_{j+1}, \lambda_{j+1}) &\leq J(u_j, m_n, \lambda_{j+1}) \\
\lambda_{j+1} \|u_j - u_{j+1}\|_{\mathcal{X}} + \|u_{j+1}\|_{\mathcal{Y}} &\leq \lambda_{j+1} \|u_j - m_n\|_{\mathcal{X}} + \|m_n\|_{\mathcal{Y}} \\
\lambda_{j+1} \|v_{j+1}\|_{\mathcal{X}} + \|u_{j+1}\|_{\mathcal{Y}} &\leq \lambda_{j+1} \|u_j - m_n\|_{\mathcal{X}} + \|m_n\|_{\mathcal{Y}}
\end{aligned}$$

Taking the limit as $n \rightarrow \infty$ gives the desired result:

$$\lambda_{j+1} \|v_{j+1}\|_{\mathcal{X}} + \|u_{j+1}\|_{\mathcal{Y}} \leq \lambda_{j+1} \|u_j\|_{\mathcal{X}}.$$

□

It then follows that

$$\begin{aligned} \sum_{j \geq 0} \frac{1}{\lambda_j} \|u_j\|_{\mathcal{X}} &= \frac{1}{\lambda_0} \|u_0\|_{\mathcal{X}} + \sum_{j \geq 0} \frac{1}{\lambda_{j+1}} \|u_{j+1}\|_{\mathcal{X}} \\ &\leq \|f\|_{\mathcal{Y}} - \|v_0\|_{\mathcal{Y}} + \sum_{j \geq 0} [\|v_j\|_{\mathcal{Y}} - \|v_{j+1}\|_{\mathcal{Y}}] \leq \|f\|_{\mathcal{Y}} \end{aligned} \quad (4.5)$$

This bound on the energy of the components $\{u_j\}$ is completely generic. It requires no particular assumptions to be made about the weight functions $\|\cdot\|_{\mathcal{X}}$ or $\|\cdot\|_{\mathcal{Y}}$.

4.3 Examples of J Functional Based Decompositions

When constructing a decomposition for members of a space \mathcal{X} in terms of a subspace of functions $\mathcal{Y} \subset \mathcal{X}$, one typically constructs a functional using the norms of the two respective spaces, such as the functional used by [19] and [21]:

$$J = \lambda \|u\|_2^2 + \|v\|_{BV}, \quad f = u + v.$$

If \mathcal{Y} is not a subspace of \mathcal{X} , the norm $\|\cdot\|_{\mathcal{X}}$ may be used for the weight function $\|\cdot\|_{\mathcal{Y}}$ as well. Examples of this are the Fourier series, and the orthogonal wavelet decomposition. A Fourier series decomposition can be obtained using a J -functional of the form:

$$J = \lambda \|u\|_2^2 + \|v\|_2^2, \quad f = u + v, \quad v = e^{ikx}, \quad k \in \mathbb{Z}$$

As long as $\lambda > 1$, this J -functional will yield the standard Fourier series decomposition, with components arranged in descending order of magnitude². This extends naturally to generalized Fourier series (including orthogonal wavelet decomposition) by replacing the set $\mathcal{Y} = \{e^{ikx}, k \in \mathbb{Z}\}$ with an orthonormal basis of choice.

²Of course, this is not the traditional method of constructing the Fourier series. However, it is instructive to highlight the fact that the J -functional framework includes many standard decompositions which may not be thought of as being optimization based.

4.4 Using the Bergman Norm $\|\cdot\|_{0,2}$ as $\|\cdot\|_{\mathcal{Y}}$

For reasons to be given later, we will use a J -functional which uses the norm from the Bergman space $\mathcal{A}_{0,2}$ as the weight function for the set \mathcal{Y} . We will analyze the behavior of the decomposition that results from the functional

$$J(u; v, \lambda) = \lambda \frac{\sqrt{\pi}}{2} \|v - u\|_2^2 + \|u\|_{0,2}, \quad \|u\|_{0,2}^2 = \int_{\mathbb{D}} |f(z)|^2 dA \quad (4.6)$$

Within this section, the functional above will be referred to as the “Bergman J -functional.”

4.4.1 Using the Bergman J -Functional with the Fourier basis functions

As a first experiment, consider the set of functions which are analytic on the unit disc, and whose boundary value functions are in L^2 . For such a function f , define

$$\begin{aligned} f &= v_0 \\ \text{for } k &= 1, 2, \dots \\ u_k &= \operatorname{arginf}_{u_k = bz^m} J(u_k; v_{k-1}, \lambda) \\ v_k &= v_{k-1} - u_k \qquad \qquad \qquad \text{end} \end{aligned}$$

Let $\sum_{n=0, n \neq m}^{\infty} d_n z^n$ be the Taylor polynomial of v_k about the origin. The J -functional then works out to be

$$\begin{aligned} J &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \|v_{k-1} - cz^m\|_2^2 + \|cz^m\|_{2,0} \\ &\text{by Parseval's equality :} \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\left(\sum_{n=0, n \neq m}^{\infty} d_n^2 \right) + (d_m - c)^2 \right] + |c| \sqrt{\frac{\pi}{m+1}} \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) [\|v_{k-1}\|_2^2 + c^2 - 2cd_m] + |c| \sqrt{\frac{\pi}{m+1}} \end{aligned}$$

For a given m value, the optimum c value (call it c_m) can be found by setting $\frac{\partial}{\partial c} J = 0$.

$$\begin{aligned} 0 &= \frac{\partial}{\partial c} J(cz^m; v_{k-1}, \lambda) = \lambda \sqrt{\pi} (c_m - d_m) + \operatorname{sign}(c_m) \sqrt{\frac{\pi}{m+1}}, \quad c_m \neq 0 \\ d_m &= c_m + \frac{\operatorname{sign}(c_m)}{\lambda \sqrt{m+1}} \end{aligned}$$

Note that d_m and c_m must have the same sign. Thus $c_m = d_m - \frac{\text{sign}(d_m)}{\lambda\sqrt{m+1}}$. Now, if $c_m \neq 0$, then $|d_m| > \frac{1}{\lambda\sqrt{m+1}}$. We can then deduce that $|d_m| \leq \frac{1}{\lambda\sqrt{m+1}} \Rightarrow c_m = 0$. This gives a complete solution for c_m :

$$c_m = \begin{cases} 0 & , |d_m| \leq \frac{1}{\lambda\sqrt{m+1}} \\ d_m - \frac{\text{sign}(d_m)}{\lambda\sqrt{m+1}} & , |d_m| > \frac{1}{\lambda\sqrt{m+1}} \end{cases} \quad (4.7)$$

Evaluating the J -functional at c_m (assuming $c_m \neq 0$) gives:

$$\begin{aligned} J(c_m z^m; f, \lambda) &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\left(\sum_{n=0, n \neq m}^{\infty} d_n^2 \right) + \frac{1}{\lambda^2 (m+1)} \right] + |b_m| \sqrt{\frac{\pi}{m+1}} \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\|f\|_2^2 - d_m^2 + \frac{1}{\lambda^2 (m+1)} \right] + \left(|d_m| - \frac{1}{\lambda\sqrt{m+1}} \right) \sqrt{\frac{\pi}{m+1}} \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\|f\|_2^2 - d_m^2 + \frac{1}{\lambda^2 (m+1)} + \frac{2|d_m|}{\lambda\sqrt{m+1}} - \frac{2}{\lambda^2 (m+1)} \right] \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\|f\|_2^2 - \left(d_m^2 - \frac{2|d_m|}{\lambda\sqrt{m+1}} + \frac{1}{\lambda^2 (m+1)} \right) \right] \\ &= \lambda \left(\frac{\sqrt{\pi}}{2} \right) \left[\|f\|_2^2 - \left(|d_m| - \frac{1}{\lambda\sqrt{m+1}} \right)^2 \right] \end{aligned}$$

The global minimizer of $J(c_m z^m; f, \lambda)$ occurs at the index m for which $\left(|d_m| - \frac{1}{\lambda\sqrt{m+1}} \right)$ is the greatest. i.e. those coefficients d_m whose absolute values exceed the threshold $\frac{1}{\lambda\sqrt{m+1}}$ correspond to the coefficients which are extracted first.

Equation 4.7 gives some valuable insight into the nature of the decomposition that results from the Bergman J -functional. Any function f whose Taylor coefficients $\{d_m\}$ satisfy $|d_m| \leq \frac{1}{\lambda\sqrt{m+1}}$ is essentially “noise.” i.e. applying the Bergman J -functional decomposition will simply return the function f as a residual, with no components extracted. An immediate consequence of this is that the residuals do not approach zero. Therefore, the partial sums $S_n = \sum_{k=1}^n v_k$ do not converge to f , and therefore $\{S_n\}$ are not truly approximants for f . It is more proper to think of this algorithm as decomposing a signal f into two distinct pieces. The “nice” part is given by $\tilde{f} = \lim_{n \rightarrow \infty} S_n$, while the “noise” part is given by $\lim_{n \rightarrow \infty} u_n$.

This brings us to the final problem of deducing the significant characteristics of the “nice” part $\tilde{f} = \lim_{n \rightarrow \infty} S_n$ and the residual $\lim_{n \rightarrow \infty} u_n$. One simple result sets the stage for this discussion:

Lemma 4.4.1. *If $f \in H^2(D)$ with $f = \sum_{n=0}^{\infty} d_n z^n$, and $\{d_n\} \in O(n^{-1/2})$, then there is a constant C such that applying the Bergman J -functional decomposition (BJFD) to Cf yields $u_1 = f$ (and thus $u_n = f \forall n$).*

Proof. $\{d_n\} \in O(n^{-1/2}) \Rightarrow \{d_n\} \in O((\lambda\sqrt{n+1})^{-1})$, and thus $\exists n_0 \geq 0, M_1 > 0$ such that $|d_n| \leq M_1 \left| \frac{1}{\lambda\sqrt{n+1}} \right| \forall n > n_0$. Choose M_2 such that $M_2 > \max_{n \leq n_0} (|d_n| \lambda\sqrt{n+1})$. Let $C < \min(\frac{1}{M_1}, \frac{1}{M_2})$. The function $Cf = \sum_{n=0}^{\infty} (Cd_n)z^n$, and the coefficients $\{Cd_n\}$ satisfy $|Cd_n| < \frac{1}{\lambda\sqrt{n+1}}$. Therefore applying the BJFD algorithm to the input signal Cf yields $u_1 = f$. \square

The decay condition on f in Lemma 4.4.1 is not very restrictive. Any signal f whose boundary value function is a member of a “nice” smoothness class has Taylor coefficients which decay at a corresponding rate: $f \in C^m(\partial D) \Leftrightarrow \{c_n\} \in O(n^{-m})$ (see [12]). This means that for any $f \in C^m(\partial D)$, $m \geq 1$ there is a multiple Cf for which the BJFD algorithm yields $\tilde{C}f = 0$.

For any signal f whose Taylor series coefficients are $O\left(\frac{1}{\lambda\sqrt{n+1}}\right)$, the BJFD algorithm must terminate after finitely many steps. This gives a decomposition into a polynomial part \tilde{f} , and a residual $(f - \tilde{f})$. The Taylor series of the residual can be thought of as being the product of giving the Taylor series of f a “haircut”:

$$\tilde{f} = \sum_{n=1}^{\infty} \tilde{d}_n z^n, \quad \tilde{d}_n = \begin{cases} d_n & , |d_n| \leq \frac{1}{\lambda\sqrt{n+1}} \\ \frac{\text{sign}(d_n)}{\lambda\sqrt{n+1}} & , |d_n| > \frac{1}{\lambda\sqrt{n+1}} \end{cases}$$

In the subsequent sections, we will be performing approximation using an algorithm similar to the BJFD algorithm in this section, but using approximants from the set of Blaschke products. Blaschke products are a natural generalization of $\{1, z, z^2, \dots\}$, but they lack many of the tools that were available in this section. Blaschke products are not orthogonal to each other (in general), and thus finding the minimizer of the J -functional in general is impractical at best. The simplicity of the calculations in this section will definitely not translate well, as this was a result of the wealth of results that are available for $\{1, z, z^2, \dots\}$. Some valuable insights from this simple exercise ought to be highlighted.

The approximating sequence derived from the BJFD algorithm does not converge to the input signal. Therefore it is not actually an approximation technique. However, this is not entirely disastrous. The EMD technique yields a decomposition into IMF components, and leaves a residual which may still contain a lot of energy (in the sense of L^2). In general, the only thing that can be said about the residual from EMD is that it is monotone. It is whatever is left after all the significant oscillations have been removed from the signal. Therefore, it may contain many interesting features - as long as they don't create oscillations. When generalizing the BJFD algorithm to a larger component set, we do not expect the component sequence to approach the input precisely. Hopefully the residual will be small (in some sense), and contain only the features we do not want to emphasize with the BJFD algorithm, but

we should not expect the residual to approach zero as the algorithm progresses.

4.4.2 An algorithm for use with Blaschke products

Definition 4.4.2. Define \mathcal{B}_A to be the set of all finite Blaschke products whose zeros come from some fixed set $A = \{a_0, a_1, \dots, a_n\}$. (Typically, it is desirable to have $0 \in A$, so we'll assume $a_0 = 0$ in this section.) i.e.

$$\mathcal{B}_A = \left\{ B(z) : B(z) = z^{m_0} \prod_k = 1^{m_k} \frac{|a_k|}{a_k} \frac{a_k - z}{1 - \overline{a_k}z} \right\}$$

In the previous subsection the Bergman J -functional decomposition (BJFD) algorithm was applied to the component set $\{1, z, z^2, \dots\}$. The next natural generalization is to perform decompositions of input functions using a larger set of component functions. A simple, incremental generalization of $\{1, z, z^2, \dots\}$ is a set like \mathcal{B}_A . As long as we include the Blaschke products which are zero at the origin, the set of component functions includes the monomial terms we used before.

The JFD algorithm for an input function f is defined as follows.

$$\begin{aligned} v_0 &:= f \\ \text{for } j &= 1, 2, \dots, \text{ iterate :} \\ u_j &:= \operatorname{argmin} J(v_{j-1}, \cdot) \\ v_j &:= v_{j-1} - u_j \end{aligned}$$

where

$$\begin{aligned} J(v, cB(z)) &= \lambda \frac{\sqrt{\pi}}{2} \|v - cB(z)\|_2^2 + |c| \sqrt{\frac{\pi}{1 + \sum_{k=0}^n m_k}} \\ \text{and } B(z) &\in \mathcal{B}_A, c \in \mathbb{C} \end{aligned}$$

The major difference between the BJFD algorithm and the JFD algorithm given above is the second term in the J -functional. If the JFD algorithm is applied to the smaller set of approximants $\{1, z, z^2, \dots\}$, the two functionals are identical. Rather than evaluating the Bergman norm, the direct dependance on the degree of the Blaschke product is used instead. In the simple case of polynomial components, the two are identical. It was a simple matter to evaluate the Bergman norm of any monomial. However, it is not easy to do in general for Blaschke products, and would require a numerical integration *for each evaluation of the J -functional*. This calculation is performed many times for each iteration of the JFD

algorithm, because finding the minimizer of the J -functional requires an explicit search. For this reason, simplifying the evaluation of the J -functional is of the utmost importance.

As a substitute for the Bergman norm, we use a term which is a function of the degree of a Blaschke product. We use the same function of degree as in the BJFD algorithm. When evaluated for the polynomial terms $\{1, z, z^2, \dots\}$, it is identical to the Bergman norm. When applied to the degree of a Blaschke product, this term performs much the same function as the Bergman norm does in the BJFD algorithm: it characterizes exactly how many oscillations the component in question has. The degree of a Blaschke product is much easier to compute than its Bergman norm, as it only requires (at most) n additions.

Chapter 5

Blaschke Polynomials

The optimization based component selection algorithm introduced in chapter 4 leaves only one question to be settled before embarking on numerical experiments: How should we choose the constraint set \mathcal{Y} ? The natural first response to this question is that \mathcal{Y} should be the set of ASNIFs contained in the space \mathcal{X} (typically $L^2[0, 2\pi]$, as this would give the most general ASNIF decomposition available. Unfortunately this is impractical as a first effort, as the set of ASNIFs is too large to be searched efficiently without first establishing some further approximation estimates.

Various subclasses of the set of ASNIFs are possible choices for a set of approximants to use as a constraint set for the J -functional. The simplest choice is the set of powers of z , $\{z^k; k = 0, 1, 2, \dots\}$ which was investigated in section 4.4.1. The next natural step is to look at the set of finite Blaschke products. This is a significant set of approximants for practical applications. Most practical data sets are modeled as a finite number of sample points. There is a well known result which states that any inner function sampled at a finite number of points on the boundary can be interpolated by a finite Blaschke product (see [9]). Thus, finite Blaschke products are sufficient to represent decompositions over the set of inner functions; one need not consider infinite Blaschke products or singular inner functions to achieve a unique representation using inner functions as approximants, as each finite Blaschke product can be identified with the set of inner functions which it interpolates over the given set of sample points.

After investigating decompositions over finite Blaschke products, one may want to consider approximants which are ASNIFs with non-constant outer functions. This would be a mature *ASNIF Decomposition*¹ in every sense. A possible simplification for finitely sampled signals would be to consider ASNIFs whose inner part is a finite Blaschke product. An

¹The capitalized term *ASNIF Decomposition* will be used to denote a J -functional based decomposition which uses a set of approximants which are ASNIFs whose outer parts may be non constant.

ASNIF Decomposition will be easier to understand if some results are first established using inner functions (or more practically, finite Blaschke products) as approximants.

Throughout this chapter unless otherwise stated, the terms Blaschke product and Blaschke polynomial will be used to refer only to Blaschke products for the unit disc, and linear combinations of Blaschke products for the unit disc, respectively.

5.1 Blaschke Polynomials

Recall a Blaschke product for the space $H^p(D)$ is a function of the form

$$B(z) = z^m \prod_{k=1}^{\nu} \frac{|a_k|}{a_k} \frac{a_k - z}{1 - \overline{a_k}z}, \quad \nu \leq \infty$$

provided that

$$\sum_{k=1}^{\nu} (1 - |a_k|) < \infty. \quad (5.1)$$

Alternatively, one could rewrite the list $\{a_k\}$ of the roots of $B(z)$ without duplication, and instead incorporate repeated roots with a multiplicity index to account for repeated roots:

$$B(z) = z^m \prod_{r=1}^{\nu} \left(\frac{|a_r|}{a_r} \frac{a_r - z}{1 - \overline{a_r}z} \right)^{m_r} \quad (5.2)$$

If we let $\zeta_0 = z$, $m_0 = m$, and $\zeta_r = \left(\frac{|a_r|}{a_r} \frac{a_r - z}{1 - \overline{a_r}z} \right)$, we have

$$B(z) = \prod_{r=0}^{\nu} (\zeta_r)^{m_r} \quad (5.3)$$

Blaschke products appear to be monomials in multiple variables $\{\zeta_r\}$ with coefficients equal to 1. The J -functional decomposition returns a linear combination of Blaschke products as an approximant, along with a residual. Linear combinations of Blaschke products appear to be polynomials in many (possibly infinitely many) variables.

In order to further explore these unique polynomials we introduce some terminology.

Definition 5.1.1. A Blaschke factor *zeta* is a rational function of the form $\zeta = \frac{a_r}{|a_r|} \frac{a_r - z}{1 - \overline{a_r}z}$, or $\zeta = z$. The root of the Blaschke factor ζ is a_r or 0, respectively.

Definition 5.1.2. A Blaschke product $B(z) = \prod_{r=0}^{\nu} (\zeta_r)^{m_r}$ has exponent $\alpha = (m_0, m_1, \dots)$, and has total degree $d = \sum_{r=0}^{\nu} m_r$.

Notice that an infinite Blaschke product has degree $d = \infty$. Also notice that any finite product of Blaschke factors is a Blaschke product, but not all infinite products of Blaschke factors are Blaschke products (because of the admissibility condition 5.1). Some details need to be addressed when defining polynomial spaces which admit infinite degrees, as this is not something that is ordinarily done.

When deriving some results dealing with Blaschke polynomials, it is important to remember that they are meromorphic functions of \mathbb{C} . When dealing with the Hardy spaces $H^p(D)$ we often think of Blaschke products as being only functions defined on \mathcal{D} , but the fact that Blaschke products are meromorphic on \mathbb{C} comes in handy from time to time.

Definition 5.1.3. *A finite linear combination of Blaschke products for the space $H^p(D)$ is called a Blaschke polynomial for the space $H^p(D)$.*

Using the term Blaschke polynomial casually requires some justification. It is not obvious that the polynomial ring generated by taking linear combinations of Blaschke products (and treating the individual Blaschke factors as the indeterminants of the polynomial ring) is in fact isomorphic to the space of Blaschke polynomial functions of the variable z . For an “ordinary” polynomial ring over an infinite field there is a bit of housekeeping to do before we can identify polynomials with their corresponding polynomial functions. Recall:

Definition 5.1.4. *Let $X = (x_1, x_2, \dots, x_n)$. If $f(X) \in k[X]$ define its polynomial function $f^b : k^n \rightarrow k$ by evaluation: If $(y_1, \dots, y_n) \in k^n$, then*

$$f^b : (y_1, \dots, y_n) \rightarrow f(y_1, \dots, y_n).$$

This is a linear mapping, because it is a homomorphism of rings.

For polynomial rings $k[X]$ over an infinite field k , the problem of identifying the polynomial function $f^b : k^n \rightarrow k$ with the polynomial $f \in k[X]$ is a simple matter (see [18]). Polynomial functions have a one to one correspondence with the corresponding elements of the polynomial ring. This allows one to drop the $(\cdot)^b$ notation for polynomial functions in practice, and refer to both the polynomials and the polynomial functions as “polynomials”.

In the case of Blaschke polynomials the correspondence between Blaschke polynomials and Blaschke polynomial functions does not follow immediately from the standard result. This is because the variables $\{\zeta_r\}$ are not independent of each other. Rather, they are all functions of the single complex variable z . Surprisingly, this does not complicate matters that much. We start by redefining the $(\cdot)^b$ notation to suit our purposes.

Definition 5.1.5. *Let $Z = (\zeta_1, \zeta_2, \dots, \zeta_n)$, where $\zeta_r = \frac{|a_r|}{a_r} \frac{a_r - z}{1 - \bar{a}_r z}$. If $f(Z) \in k[Z]$ define its Blaschke polynomial function $f^b : \mathbb{C} \rightarrow \mathbb{C}$ by evaluation, with all removable singularities*

removed:

$$f^b : z \rightarrow \lim_{\zeta \rightarrow z} f(\zeta)$$

The identification result then follows quite simply.

Theorem 5.1.6. ² Let $Z = \{\zeta_1, \zeta_2, \dots, \zeta_\nu\}$ be Blaschke factors for $H^p(D)$ ($\nu \leq \infty$). If $f(Z), g(Z) \in \mathbb{C}[Z]$ satisfy $f^b = g^b$, then $f(Z) = g(Z)$.

Proof. Assume $f \neq g$. Then $(f - g) = \sum_{k=1}^N c_k Z_k^{\alpha_k}$. Now $(f - g)^b(z) = 0 \forall z \in \mathbb{C}$. Therefore $(f - g)$ cannot have any poles in \mathbb{C} . Thus the degree of $(f - g)$ must be zero, meaning $(f - g)$ is constant, and thus $(f - g) = 0$. \square

The proof above is quite simple. However it has significant implications for approximation using Blaschke polynomials. The one to one correspondence of Blaschke polynomials with Blaschke polynomial functions means that *any set of distinct Blaschke products is linearly independent*. The importance of this for approximation cannot be understated.

The Fourier series approximation for boundary functions in $H^p(D)$ uses a basis consisting of the linearly independent functions $\{1, z, z^2, \dots\}$. This basis is complete in $L^2(\partial D)$ (as well as many other norms), and thus is capable of generating approximating sequences for any element of $L^2(\partial D)$. The set of Blaschke products for $H^2(D)$ includes the Fourier basis functions, and is still linearly independent. It is only in the sense of L^2 that there is any dependency.

5.2 Spaces of Blaschke Polynomials

Blaschke polynomials form the basis for many different linear spaces, each amenable to different types of applications. One might want to consider the space of all finite Blaschke polynomials whose Blaschke factors come from a prescribed set when doing approximation using an iterative decomposition algorithm. Alternatively, one might consider the space of all Blaschke polynomials in order to explore questions of functional analysis using Blaschke polynomials. These settings (and many others) correspond naturally to various linear spaces of Blaschke polynomials.

²This author was unable to find any literature containing any similar results. The literature on Blaschke products typically uses them to characterize multiplier spaces, as factors of certain ideals, or extracts them as factors from other functions - like the canonical factorization theorems. This author was left with the impression that most of the interest in Blaschke products pertained to multiplying them with other objects; adding Blaschke products together apparently wasn't very interesting to other authors. This is somewhat understandable, as Blaschke products are typically presented as a special topic in books on complex function spaces. For the methods being developed there, another polynomial space might seem rather irrelevant. Undoubtedly, if Blaschke products were covered in introductory books on complex analysis, the proof of theorem 5.1.6 would be one of the simpler homework problems.

Ordinarily, polynomial spaces are defined as including only polynomials of finite degree. The notion of Blaschke polynomials admits the possibility of defining a polynomial space which includes polynomials of infinite total degree, because there are well-defined Blaschke products with infinite degree.

5.2.1 $\mathbb{C}[\zeta]$

The trivial space $\mathbb{C}[\zeta]$ warrants mention for two reasons: it serves as an explicit reminder of the connection between Blaschke polynomials and ordinary polynomials $\mathbb{C}[z]$, and it also provides a convenient setting in which to demonstrate a completeness result without any unnecessary complications.

Clearly the choice of the root of the Blaschke factor ζ is the only difference between any two different Blaschke polynomial spaces $\mathbb{C}[\zeta_1]$ and $\mathbb{C}[\zeta_2]$. If one chooses $\zeta = z$, one is left with the ordinary polynomial space $\mathbb{C}[z]$. It is well known that the polynomials $\mathbb{C}[z]$ are dense in $H^2(D)$. Using the mapping $T(z) = \zeta(z)$, it is simple to show that for any Blaschke factor ζ , the set $\mathbb{C}[\zeta]$ is also dense in $H^2(D)$.

Theorem 5.2.1. *If $\zeta = \frac{|a|}{a} \frac{a-z}{1-\bar{a}z}$ for some $0 < |a| < 1$, then $\mathbb{C}[\zeta]$ is dense in $H^2(D)$.*

Proof. Observe that $z(\zeta) = \frac{a}{|a|} \frac{|a|-\zeta}{1-|a|\zeta}$. The Taylor series for z about the point $\zeta = 0$ is

$$\begin{aligned} z &= \sum_{k=0}^{\infty} \frac{z^{(k)}(0)}{k!} \zeta^k \\ &= a + a(|a|^2 - 1) \sum_{k=1}^{\infty} |a|^{k-2} \zeta^k. \end{aligned}$$

This series converges absolutely for $|\zeta| < \frac{1}{|a|}$. Now, ζ is a conformal mapping from D to D , with $\zeta : \partial D \rightarrow \partial D$, so the power series above converges absolutely for every $z \in D \cup \partial D$. Clearly then, for any polynomial $p(z)$ there exists a power series $\sum_{k=0}^{\infty} c_k \zeta^k$ which is absolutely convergent for all $z \in D \cup \partial D$. Because the polynomials $\mathbb{C}[z]$ are dense in $H^2(D)$, the Blaschke polynomials $\mathbb{C}[\zeta]$ must be dense in $H^2(D)$ as well. \square

This result shows that any single Blaschke factor ζ generates a polynomial space $\mathbb{C}[\zeta]$ which is dense in $H^2(D)$. A simple consequence is that any set which contains $\mathbb{C}[\zeta]$ (for any Blaschke factor ζ) is dense in $H^2(D)$. For example, $\mathbb{C}[\zeta_1, \zeta_2, \dots, \zeta_n]$ is dense in $H^2(D)$.

5.2.2 $\mathbb{C}[\zeta_1, \dots, \zeta_n]$

The simplest non-trivial class of Blaschke polynomial spaces is constructed by selecting a finite set of distinct points $\{a_k\}_{k=1}^n \subset \mathcal{D}$, and using the corresponding Blaschke factors

$\zeta_k = \frac{a_k \cdot \overline{a_k - z}}{|a_k| 1 - \overline{a_k} z}$ (or $\zeta_k = z$ if $a_k = 0$) to generate the polynomial ring $\mathbb{C}[\zeta_1, \dots, \zeta_n]$.

There are no infinite Blaschke products that can be constructed using a finite number of Blaschke terms. Thus the polynomial space $\mathbb{C}[\zeta_1, \dots, \zeta_n]$, defined in the usual way as all finite linear combinations of finite products of $\{\zeta_1, \dots, \zeta_n\}$, includes every Blaschke product of the form $\prod_{k=1}^n \zeta_k^{p_k}$.

5.2.3 Spaces of Blaschke polynomials in infinitely many variables

There are many different Blaschke polynomial spaces that may be of interest which include infinitely many Blaschke factors.

Definition 5.2.2. *Given a set of Blaschke roots $\{a_\nu\} \subset D$ where ν are elements of an index set for $\{a_\nu\}$, $\mathbb{C}[\{\zeta_\nu\}]$ denotes the set of all finite linear combinations of finite Blaschke products involving the Blaschke factors $\{\zeta_\nu\}$.*

This is the ordinary definition of a space of polynomials in infinitely many variables. Every polynomial $f \in \mathbb{C}[\{\zeta_\nu\}]$ has finite degree.

If the set $\{a_\nu\}$ contains a sequence of points which satisfy the Blaschke condition 5.1, then it is possible to construct infinite Blaschke products using $\{\zeta_\nu\}$. The space $\mathbb{C}[\{\zeta_\nu\}]$ does not include these infinite Blaschke products.

Definition 5.2.3. *Given a set of Blaschke roots $\{a_\nu\} \subset D$ where ν are elements of an index set for $\{a_\nu\}$, $\mathbb{C}_B[\{\zeta_\nu\}]$ denotes the set of all finite linear combinations of Blaschke products involving the Blaschke factors $\{\zeta_\nu\}$.*

The two polynomial spaces $\mathbb{C}[\{\zeta_\nu\}]$ and $\mathbb{C}_B[\{\zeta_\nu\}]$ may or may not be identical, depending on the set $\{a_\nu\}$. This dependence is characterized quite simply.

Lemma 5.2.4. *Given a set of Blaschke roots $\{a_\nu\} \subset D$, the polynomial spaces $\mathbb{C}[\{\zeta_\nu\}]$ and $\mathbb{C}_B[\{\zeta_\nu\}]$ are identical if and only if $\{a_\nu\}$ contains no infinite subsequence $\{a_k\}$ which satisfies equation 5.1.*

Proof. \Rightarrow : Assume not. i.e. $\mathbb{C}[\{\zeta_\nu\}]$ and $\mathbb{C}_B[\{\zeta_\nu\}]$ are identical, and $\exists \{a_k\}_{k=1}^\infty \subset \{a_\nu\}$ which satisfies equation 5.1. Then $B(z) = \prod_{k=1}^\infty \zeta_k \in \mathbb{C}_B[\{\zeta_\nu\}]$. However $B(z) \notin \mathbb{C}[\{\zeta_\nu\}]$, so $\mathbb{C}[\{\zeta_\nu\}] \neq \mathbb{C}_B[\{\zeta_\nu\}]$, a contradiction.

\Leftarrow : If $\{a_\nu\}$ contains no sequence of points which satisfy equation 5.1, then $\mathbb{C}_B[\{\zeta_\nu\}]$ contains only finite Blaschke products. Therefore, $\mathbb{C}_B[\{\zeta_\nu\}] = \mathbb{C}[\{\zeta_\nu\}]$. \square

Corollary 5.2.5. *If $\mathbb{C}_B[\{\zeta_\nu\}] \neq \mathbb{C}[\{\zeta_\nu\}]$, then $\{a_\nu\}$ is not closed.*

Proof. If $\mathbb{C}_B[\{\zeta_\nu\}] \neq \mathbb{C}[\{\zeta_\nu\}]$, then there is a sequence $\{a_k\} \subset \{a_\nu\}$ which satisfies 5.1. Any sequence which satisfies 5.1 must satisfy $\lim_{k \rightarrow \infty} |a_k| = 1$. Now, $\{a_k\}$ is a bounded sequence, so there is a convergent subsequence $\{a_{k_l}\} \subset \{a_k\}$. Let $\lim_{l \rightarrow \infty} a_{k_l} = a_0$. Observe that $|a_0| = 1$, so $a_0 \notin D$, and thus $a_0 \notin \{a_\nu\}$. Therefore $\{a_\nu\}$ is not closed. \square

The converse of this result is false. i.e. it is not true that if $\{a_\nu\}$ is not closed, that $\mathbb{C}_B[\{\zeta_\nu\}] \neq \mathbb{C}[\{\zeta_\nu\}]$. For example, choose a set $\{a_\nu\} \subset D$ such that $\{a_\nu\}$ is not closed, and $\{a_\nu\} \cup \partial\{a_\nu\} \subset D$. There is no sequence $\{a_k\} \subset \{a_\nu\}$ for which $\lim_{k \rightarrow \infty} |a_k| = 1$, so no sequence $\{a_k\} \subset \{a_\nu\}$ can satisfy 5.1. Therefore $\mathbb{C}_B[\{\zeta_\nu\}]$ contains only finite Blaschke products, and $\mathbb{C}_B[\{\zeta_\nu\}] = \mathbb{C}[\{\zeta_\nu\}]$.

5.3 Approximation Using Blaschke Polynomials

Definition 5.3.1. A Blaschke series in $H^p(D)$ is a series of the form:

$$\sum_{k=1}^{\infty} c_k B_k,$$

where each B_k is a Blaschke product for $H^p(D)$.

An immediate consequence of the size of the set of Blaschke products is that any non-trivial³ set of Blaschke polynomials cannot be used to give a unique approximating sequence for any element of $L^2(\partial D)$. For example, take any Blaschke product $B(z) = \zeta^m$ for which $\zeta \neq z$. $B(z)$ has a unique Blaschke series expansion in the space $\mathbb{C}[\zeta]$, namely $B(z) = \zeta^m$. However, in the slightly larger space $\mathbb{C}[\zeta, z]$, there are two readily available Blaschke series expansions (and infinitely many others as well). $B(z)$ may still be represented trivially as $B(z) = \zeta^m$. It may also be represented by its Taylor series about the origin:

$$B(z) = \sum_{k=0}^{\infty} \frac{B^{(k)}(0)}{k!} z^k.$$

The nonuniqueness of Blaschke polynomial approximations means that the choice of algorithm plays a key role in the decomposition one obtains. What makes one Blaschke polynomial decomposition better than another may depend entirely on the particular application at hand. Possible considerations include:

- sparsity of the representation,

³Nontrivial here means that more than one Blaschke factor is used, i.e. the set of Blaschke polynomials being considered is not of the form $\mathbb{C}[\zeta]$, or $\mathbb{C}[z]$.

- orthogonality of the components,
- the distribution of the roots of the component Blaschke products,
- the informational content of the residual.

Clearly there is no one best way to set up a Blaschke series decomposition technique.

5.4 A Possible Extension to Singular Inner Functions

It is only natural to generalize Blaschke polynomial spaces to include singular inner functions as well. After all, the inner functions - Blaschke products, and singular inner functions - are the “pure instantaneous frequency” functions in $H^2(D)$. A series representation which uses approximants from this larger set is a natural generalization of The Fourier series. For lack of better notation, we will call the linear span of all inner functions $\mathbb{C}[\mathcal{IF}]$

It is not clear that polynomials are the best setting to study this function space. Blaschke products translate naturally into the language of polynomials because the exponents on each Blaschke factor are non-negative integers. For singular inner functions, the “exponents” involved do not share this property. More precisely, the exponents on the monomials in the space $\mathbb{C}[\zeta_1, \zeta_2, \dots, \zeta_n]$ are elements of \mathbb{N}^n (here $\mathbb{N} = \{0, 1, 2, \dots\}$), and form a monoid under the usual vector addition. The addition of two exponent vectors corresponds to the multiplication of the corresponding two Blaschke products. The two operations are homomorphic; in this sense, the exponent vectors provide a natural parametrization of the set of Blaschke products. The monoid which consists of the set of Blaschke products in n Blaschke factors, along with the operation of multiplication is thus naturally parameterized by the set \mathbb{N}^n . Thus it is natural to consider this a set of monomials - and thus the linear span of Blaschke products may naturally be considered a polynomial space.

In the case of singular inner functions, the set of singular measures forms a similar monoid with the usual addition operation, which is homomorphic to the monoid of singular inner functions under the operation of multiplication. The homomorphism between these two operations (multiplication of the singular inner functions, and addition of the singular measures) is identical to the homomorphism used to parameterize the operations in the case of Blaschke products (multiplication of Blaschke products, and addition of the exponents). The critical difference is that the space of exponents⁴ is not discrete for singular inner functions. The exponents of singular inner functions are not “nicely” parameterized by any monoid of the form \mathbb{N}^ν - not even if ν is (countably) infinite.

⁴*Exponent* here is used to mean the set of objects in the monoid which, under the logarithm map, is homomorphic to multiplication of the singular inner functions.

Characterizing the set of singular inner functions as monomials in order to shoehorn them into the language of polynomials seems premature at best, and disingenuous at worst. There are certainly some interesting questions regarding the linear span of all inner functions. It is a set which holds promise for approximation, but does not translate simply into the language of polynomials.

Chapter 6

J-functional Decomposition Using Blaschke Polynomials

Blaschke polynomials provide intriguing opportunities for new approximation and decomposition algorithms. There is a plethora of different Blaschke polynomial spaces available to use for these techniques, and an infinite number of algorithms one could potentially use to do approximation or decomposition over these spaces. In this chapter, we begin to explore the possibilities with some simple examples.

6.1 Introducing the Algorithms

We will be using two algorithms to perform decompositions of analytic signals: the *J-functional decomposition* (JFD) algorithm, and the *iterative best approximation* (IBA) method. For both algorithms, some common notation will be used.

The set $X = \{\exp(in\pi/12)\}, n = 0, 1, \dots, 23$ will be used as sample points. Input functions will be of the form $f : X \rightarrow \mathbb{C}$, where $f : D \rightarrow \mathbb{C}$ is a member of $H^2(D)$, having an analytic continuation to an open set $O \supset (D \cup \partial D)$.

For both algorithms, approximants will be elements of $\mathbb{C}[\{\zeta_n\}]$, where $\{a_n\}$ is given by $a_0 = 0, a_n = R \exp(in\pi/3), n = 1, 2, \dots, 6$ and $R = \frac{\sqrt{5}-1}{2}$, the conjugate golden ratio. Thus, $\{\zeta_n\} = \left\{ z \cup \left\{ \frac{|a_n|}{a_n} \frac{a_n - z}{1 - \bar{a}_n z}, n = 1, 2, \dots, 6 \right\} \right\}$.¹ Only Blaschke products of total degree ≤ 12 will be used, for reasons given in Section 6.1.3.

The JFD and IBA algorithms are very similar to each other. They are both iterative

¹There is no clear choice for a best set $\{a_n\}$ of Blaschke roots to construct a set of approximants. The set constructed here was chosen for some geometric properties that it has. These properties may or may not make for a good set of approximating functions. There is certainly much work to be done in order to establish what makes a good space of Blaschke polynomial approximants. The problem is similar in nature to establishing a sampling theorem.

algorithms. At each step, a Blaschke monomial $cB(z) \in \mathbb{C}[\{\zeta_n\}]$ is found which minimizes a J -functional.² Both algorithms may be written in the form below.

$$\begin{aligned} v_0 &:= f \\ \text{for } j = 1, 2, \dots : \\ u_j &:= \operatorname{argmin} J(v_{j-1}, \cdot) \\ v_j &:= v_{j-1} - u_j \end{aligned}$$

For the IBA, $J(v, cB(z)) = J_1(v, cB(z))$; for the JFD, $J(v, cB(z)) = J_2(v, cB(z))$. These are:

$$\begin{aligned} J_1(v, cB(z)) &= \frac{\sqrt{\pi}}{2} \|v - cB(z)\|_2^2, \text{ and } B(z) \in \mathbb{C}[\{\zeta_n\}], c \in \mathbb{C} \\ J_2(v, cB(z)) &= \frac{\sqrt{\pi}}{2} \|v - cB(z)\|_2^2 + |c| \sqrt{\frac{\pi}{1 + \deg B(z)}}, \text{ and } B(z) \in \mathbb{C}[\{\zeta_n\}], c \in \mathbb{C} \\ &= J_1(v, cB(z)) + |c| \sqrt{\frac{\pi}{1 + \deg B(z)}} \end{aligned}$$

Because the set of approximants is so large, there may be more than one minimizer for the J -functional being used. When this occurs, the first minimizer found will be chosen as the next component in the representation.

For both of these algorithms, the optimal coefficient c can be solved for explicitly in terms of the other inputs $v, B(z)$.

6.1.1 The IBA Algorithm

The IBA algorithm for the constraint set $\mathbb{C}_B[Z_P]$ is given by:

$$\begin{aligned} v_0 &:= f \\ \text{for } j = 1, 2, \dots : \\ u_j &:= \operatorname{argmin} J_1(v_{j-1}, \cdot) \\ v_j &:= v_{j-1} - u_j \end{aligned}$$

where $J_1(v, cB(z)) = \frac{\sqrt{\pi}}{2} \|v - cB(z)\|_2^2$, and $B(z) \in \mathbb{C}[\{\zeta_n\}], c \in \mathbb{C}$.

This algorithm requires a search to be done in order to find the minimizer of $J_1(v_{j-1}, \cdot)$. For each possible Blaschke product $B(z) \in \mathbb{C}[P]$, an optimal value c must be found, and

²It is a slight abuse of terminology to refer to one algorithm as *the* JFD algorithm, but there is no risk of ambiguity within this work. It is more accurate to think of both algorithms as being JFDs, and the term IBA simply highlights a special property which that particular algorithm has.

the best resulting candidate $cB(z)$ selected as the minimizer of the J -functional. There is a problem here: $\mathbb{C}[\{\zeta_n\}]$ is an infinite set. This problem will be partially addressed in Section 6.1.3.

6.1.2 The JFD Algorithm

The JFD algorithm was introduced in general in Section 4.4.2. In order to use this algorithm in practice the parameter λ needs to be chosen. The parameter λ may be thought of as a regularization parameter. (See[2].) With this perspective, it is clear that there may not be one “best” value of λ . In this work, we set $\lambda = 1$ for simplicity.

The JFD algorithm we will use is given by:

$$\begin{aligned} v_0 &:= f \\ \text{for } j &= 1, 2, \dots : \\ u_j &:= \operatorname{argmin} J_2(v_{j-1}, \cdot) \\ v_j &:= v_{j-1} - u_j \end{aligned}$$

where $J_2(v, cB(z)) = \frac{\sqrt{\pi}}{2} \|v - cB(z)\|_2^2 + |c| \sqrt{\frac{\pi}{1 + \sum_{k=0}^n m_k}}$, and $B(z) \in \mathbb{C}[\{\zeta_n\}]$, $c \in \mathbb{C}$.

Notice that $J_2(v, cB(z)) = J_1(v, cB(z)) + |c| \sqrt{\frac{\pi}{1 + \sum_{k=0}^n m_k}}$. The problem of minimizing J_2 is almost identical to that of minimizing J_1 . Once again, $\mathbb{C}[\{\zeta_n\}]$ is an infinite set.

6.1.3 A Smaller Constraint Set

In both the IBA and JFD algorithm, a functional must be minimized over a space of Blaschke polynomials set $\mathbb{C}[\{\zeta_n\}]$, with zeros chosen from the set $\{a_n, n = 0, 1, \dots, 6\}$. This presents a problem, as $\mathbb{C}[\{\zeta_n\}]$ is an infinite set. It is necessary to either restrict the search to a (finite) subset of $\mathbb{C}[\{\zeta_n\}]$, or to develop results which can guarantee that a minimum has been found.

Using the Nyquist rate as a rough guide, we restrict our search domain for the minimization step to those Blaschke products whose zeros are contained in the set $\{a_n\}$, and whose total degree is no greater than 12.

Definition 6.1.1. Let $B_{\{a_n\}}^{\leq 12}$ be the set of all Blaschke products of the form:

$$B(z) = z^{m_0} \prod_{n=1}^6 \left(\frac{|a_n|}{a_n} \frac{a_n - z}{1 - \overline{a_n} z} \right)^{m_n},$$

where $\deg(B(z)) = \sum_{k=0}^6 m_k \leq 12$.

This reduces the problem to a finite one, but it is by no means trivial. When performing IBA using approximants from $\mathbb{C}[z]$ of degree ≤ 12 , we only need to compare 13 values at each step. This is further simplified by orthogonality, which means we don't really need to perform any iterations. Once the optimal coefficients are found for each power of z in the first iteration, the IBA is finished, as the coefficients are independent. The only thing IBA does is rearrange them in descending order of magnitude.

Performing IBA or JFD using $B_{\{a_n\}}^{\leq 12}$ as a constraint set complicates things. $B_{\{a_n\}}^{\leq 12}$ is a very large set: $|B_{\{a_n\}}^{\leq 12}| = \binom{19}{7} = 50388$. To make matters worse, the Blaschke products we are using are not orthogonal to each other in general. This has two related consequences. The first is that we must evaluate the optimal coefficient c for each Blaschke product when searching for the minimizer of $J_1(f, \cdot)$ independently. (There is no obvious FFT-like acceleration available - yet.) The other effect of nonorthogonality is that after the first component has been extracted, a new search must be performed to find the next one. The net effect bears repeating: we must evaluate 50388 different possible coefficients *at each iteration of the algorithm*. Suddenly expanding the constraint set to allow Blaschke products with six more possible zero locations doesn't seem so simple.

It is important to re-emphasize that stopping the search at 50388 is essentially arbitrary. Without any solid results to use as a guide, we are simply using the consequences of Nyquist sampling as a *rough* guideline. There is not yet a coherent understanding of how instantaneous frequency relates to the standard concept of frequency as used in Fourier series (and thus Nyquist sampling). However, the two quantities *do* mesh perfectly for the Blaschke products $\{1, z, z^2, \dots\}$.

Clearly using any Blaschke polynomial space larger than the ordinary polynomial space $\mathbb{C}[z]$ introduces many orders of magnitude more computation. In order for such a method to be usable, acceleration techniques are desperately needed. Such results could take many forms: a condition that allows certain classes of Blaschke products to be ignored in the optimization step, a generalization of orthogonality which allows optimizers to be found directly for (subclasses of) Blaschke products, a method of combining calculations analogous to the FFT, or some other type of acceleration.

6.2 Minimizing the IBA functional

The heart of the IBA algorithm (and also the JFD algorithm) is the minimization of the J -functional. This step requires finding the Blaschke monomial $cB(z)$, $c \in \mathbb{C}$, $B(z) \in B_{\{a_n\}}^{\leq 12}$, such that $J_1(v_{j-1}, cB(z))$ is minimized. Due to the simple structure of J_1 , this does not require an explicit search algorithm such as steepest descent, conjugate gradients, etc.

The J_1 functional is quadratic in c , the coefficient of the Blaschke monomial. This means that for each candidate Blaschke product $B(z) \in B_{\{a_n\}}^{\leq 12}$, the best coefficient $c = a + ib$ can be solved for explicitly by two simple conditions:

$$\begin{aligned}\frac{\partial}{\partial a} J_1 &= 0 \\ \frac{\partial}{\partial b} J_1 &= 0\end{aligned}$$

For these calculations we introduce one more bit of notation.

Definition 6.2.1. *The inner product $\langle \cdot, \cdot \rangle : \mathbb{C}^{24} \times \mathbb{C}^{24} \rightarrow \mathbb{C}$ is defined to be $\langle u, v \rangle = \frac{1}{24} \sum_{n=1}^{24} u_n \overline{v_n}$.*

The norm induced by this inner product will be written without a subscript as $\|\cdot\|$. This norm is clearly identical to the standard l^2 norm on \mathbb{C}^{24} , when normalized appropriately.

$$\begin{aligned}J_1(v, cB(z)) &= \frac{\sqrt{\pi}}{2} \|v - cB(z)\|^2 \\ &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}(\langle v, cB \rangle) + \|cB\|^2) \\ &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}(\overline{c} \langle v, B \rangle) + |c|^2) \\ &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}((a - ib) \langle v, B \rangle) + a^2 + b^2) \\ &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2a\operatorname{Re}(\langle v, B \rangle) - 2b\operatorname{Im}(\langle v, B \rangle) + a^2 + b^2) \\ 0 = \frac{\partial J_1}{\partial a} &= \frac{\sqrt{\pi}}{2} (-2\operatorname{Re}(\langle v, B \rangle) + 2a) \\ a &= \operatorname{Re} \langle v, B \rangle \\ 0 = \frac{\partial J_1}{\partial b} &= \frac{\sqrt{\pi}}{2} (-2\operatorname{Im}(\langle v, B \rangle) + 2b) \\ b &= \operatorname{Im} \langle v, B \rangle \\ c = a + ib = \langle v, B \rangle &= \frac{1}{24} \sum_{n=1}^{24} v_n \overline{B_n}\end{aligned}\tag{6.1}$$

Solving for J_1 at this optimum c value gives:

$$\begin{aligned}
J_1(v, cB(z)) &= \frac{\sqrt{\pi}}{2} \|v - cB(z)\|^2 \\
&= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}\langle v, cB \rangle + \|cB\|^2) \\
&= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}(\bar{c}\langle v, B \rangle) + |c|^2) \\
&= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2|c|^2 + |c|^2) \\
&= \frac{\sqrt{\pi}}{2} (\|v\|^2 - |c|^2)
\end{aligned} \tag{6.2}$$

This expression for J_1 allows for a significant simplification of IBA decomposition algorithm. The IBA algorithm requires the evaluation of $\operatorname{argmin}(J_1)$ over all candidate Blaschke products and coefficients. Clearly, equation 6.2 implies that the minimum J_1 value is achieved by the Blaschke product whose coefficient c has the largest absolute value. Thus there is no need to evaluate the J_1 functional value explicitly in order to find its minimizer. One need only evaluate the optimum c value for each candidate Blaschke product, and pick the one with the largest absolute value.

6.2.1 Consequences for Stopping Criteria

Error minimizing algorithms like IBA are among the oldest and most thoroughly studied approximation methods. Convergence is guaranteed (asymptotically at least), as long as the component set being used forms a complete basis of the space being approximated. This is certainly the case here, as there are many more than 48 linearly independent functions being used to do approximation in \mathbb{C}^{24} , a 48 \mathbb{R} -dimensional space. Sensible stopping criteria are easy to come by. Some examples are:

- a prescribed number of components are extracted.
- the norm/energy of the residual is less than some fixed tolerance.
- the norm/energy of the residual is less than some proportion of the norm/energy of the input signal.

6.3 Minimizing the JFD functional

Much like the IBA algorithm, the JFD algorithm basically boils down to finding the minimizer J_2 . This requires finding the Blaschke monomial $cB(z)$, $c \in \mathbb{C}$, $B(z) \in B_{\{a_n\}}^{\leq 12}$, such that

$J_2(v_{j-1}, cB(z))$ is minimized. However, it turns out that the extra term in the J_2 functional makes this a slightly more subtle endeavor than for J_1 .

The best coefficient $c = a + ib$ can still be found directly for each Blaschke product $B(z)$ because J_2 is also convex in each of a and b . (The proof of this is in Appendix 8.1.3.) For this calculation, the Cartesian form and the polar form of c will be used interchangeably: $c = a + ib = r(\cos \theta + i \sin \theta)$.

$$\begin{aligned}
J_2 &= \frac{\sqrt{\pi}}{2} \|v - cB\|_2^2 + |c| \sqrt{\frac{\pi}{1+D}}, \text{ where } D = \deg(B(z)) \\
&= \frac{\sqrt{\pi}}{2} (\|v\|_2^2 - 2\operatorname{Re}(\langle v, cB \rangle) + |c|^2) + |c| \sqrt{\frac{\pi}{1+D}} \\
&= \frac{\sqrt{\pi}}{2} (\|v\|_2^2 - 2\operatorname{Re}(\bar{c}\langle v, B \rangle) + |c|^2) + |c| \sqrt{\frac{\pi}{1+D}} \\
&= \frac{\sqrt{\pi}}{2} (\|v\|_2^2 - 2\operatorname{Re}((a-ib)\langle v, B \rangle) + a^2 + b^2) + \sqrt{a^2 + b^2} \sqrt{\frac{\pi}{1+D}} \\
&= \frac{\sqrt{\pi}}{2} (\|v\|_2^2 - 2a\operatorname{Re}(\langle v, B \rangle) - 2b\operatorname{Im}(\langle v, B \rangle) + a^2 + b^2) + \sqrt{a^2 + b^2} \sqrt{\frac{\pi}{1+D}}
\end{aligned}$$

$$\begin{aligned}
0 &= \frac{\partial J_2}{\partial a} = \frac{\sqrt{\pi}}{2} (-2\operatorname{Re}(\langle v, B \rangle) + 2a) + \frac{a}{\sqrt{a^2 + b^2}} \sqrt{\frac{\pi}{1+D}} \\
&= -2\operatorname{Re}(\langle v, B \rangle) + 2a + \frac{2a}{\sqrt{a^2 + b^2}} \frac{1}{\sqrt{1+D}} \\
&= -\operatorname{Re}(\langle v, B \rangle) + r \cos \theta + \frac{\cos \theta}{\sqrt{1+D}} \\
r &= \frac{\operatorname{Re}(\langle v, B \rangle)}{\cos \theta} - \frac{1}{\sqrt{1+D}} \tag{6.3}
\end{aligned}$$

$$\begin{aligned}
0 &= \frac{\partial J_2}{\partial b} = \frac{\sqrt{\pi}}{2} (-2\operatorname{Im}(\langle v, B \rangle) + 2b) + \frac{b}{\sqrt{a^2 + b^2}} \sqrt{\frac{\pi}{1+D}} \\
&= -2\operatorname{Im}(\langle v, B \rangle) + 2b + \frac{2b}{\sqrt{a^2 + b^2}} \frac{1}{\sqrt{1+D}} \\
&= -\operatorname{Im}(\langle v, B \rangle) + r \sin \theta + \frac{\sin \theta}{\sqrt{1+D}} \\
r &= \frac{\operatorname{Im}(\langle v, B \rangle)}{\sin \theta} - \frac{1}{\sqrt{1+D}} \tag{6.4}
\end{aligned}$$

Equating 6.4 and 6.3 gives:

$$\begin{aligned}\frac{\operatorname{Re}(\langle v, B \rangle)}{\cos \theta} - \frac{1}{\sqrt{1+D}} &= \frac{\operatorname{Im}(\langle v, B \rangle)}{\sin \theta} - \frac{1}{\sqrt{1+D}} \\ \frac{\operatorname{Re}(\langle v, B \rangle)}{\cos \theta} &= \frac{\operatorname{Im}(\langle v, B \rangle)}{\sin \theta} \\ \tan \theta &= \frac{\operatorname{Im}(\langle v, B \rangle)}{\operatorname{Re}(\langle v, B \rangle)} \\ \theta &= \arg(\langle v, B \rangle)\end{aligned}$$

As we are interested in the complex value c , the nonuniqueness of θ is not a concern. The optimal c value is a real multiple of the quantity $\langle v, B \rangle$. Substituting θ into equation 6.4 we have

$$c = \left(1 - \frac{1}{|\langle v, B \rangle| \sqrt{1+D}}\right) \langle v, B \rangle \quad (6.5)$$

Evaluating J_2 at the optimum c value gives

$$\begin{aligned}J_2 &= \frac{\sqrt{\pi}}{2} \|v - cB\|^2 + |c| \sqrt{\frac{\pi}{1+D}} \\ J_2 &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}(\langle v, cB \rangle) + |c|^2) + |c| \sqrt{\frac{\pi}{1+D}} \\ J_2 &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2\operatorname{Re}(\bar{c}\langle v, B \rangle) + |c|^2) + |c| \sqrt{\frac{\pi}{1+D}} \\ J_2 &= \frac{\sqrt{\pi}}{2} \left(\|v\|^2 - 2 \left(1 - \frac{1}{|\langle v, B \rangle| \sqrt{1+D}}\right) |\langle v, B \rangle|^2 \right. \\ &\quad \left. + \left(1 - \frac{1}{|\langle v, B \rangle| \sqrt{1+D}}\right)^2 |\langle v, B \rangle|^2 \right) + \left(1 - \frac{1}{|\langle v, B \rangle| \sqrt{1+D}}\right) |\langle v, B \rangle| \sqrt{\frac{\pi}{1+D}} \\ &= \frac{\sqrt{\pi}}{2} \left(\|v\|^2 - 2|\langle v, B \rangle|^2 + 2\frac{1}{\sqrt{1+D}} |\langle v, B \rangle| + |\langle v, B \rangle|^2 - 2\frac{|\langle v, B \rangle|}{\sqrt{1+D}} + \frac{1}{1+D} \right) \\ &\quad + \sqrt{\frac{\pi}{1+D}} |\langle v, B \rangle| - \frac{\sqrt{\pi}}{1+D} \\ &= \frac{\sqrt{\pi}}{2} \left(\|v\|^2 - \frac{1}{1+D} + 2\frac{|\langle v, B \rangle|}{\sqrt{1+D}} - |\langle v, B \rangle|^2 \right) \\ &= \frac{\sqrt{\pi}}{2} \left(\|v\|^2 - \left(|\langle v, B \rangle| - \frac{1}{\sqrt{1+D}} \right)^2 \right) \quad (6.6)\end{aligned}$$

One possible problem is what happens if $\langle v, B \rangle = 0$. If this is the case, then equation 6.5 is undefined. In this case, we must evaluate J_2 directly. Comparing the value of $J_2(v, 0)$ to

that of $J_2(v, cB)$ for $c \neq 0$ gives:

$$\begin{aligned} J_2(v, cB) - J_2(v, 0) &= \left(\frac{\sqrt{\pi}}{2} \|v - cB\|^2 + |c| \sqrt{\frac{\pi}{1 + \deg B}} \right) - \left(\frac{\sqrt{\pi}}{2} \|v\|^2 \right) \\ &= \frac{\sqrt{\pi}}{2} \left(\|v\|^2 + |c|^2 + |c| \sqrt{\frac{\pi}{1 + \deg B}} - \|v\|^2 \right) \\ &= \frac{\sqrt{\pi}}{2} \left(|c|^2 + |c| \sqrt{\frac{\pi}{1 + \deg B}} \right) > 0 \end{aligned}$$

Thus if $\langle v, B \rangle = 0$, the coefficient c which minimizes $J_2(v, cB)$ is $c = 0$.

6.3.1 Consequences for Stopping Criteria

The JFD algorithm is not guaranteed to converge (i.e. the residual does not approach 0). However, equation 6.6 sheds some light on what is going on with the JFD algorithm, and also on how to pick sensible stopping criteria.

Consider a constant input signal given by $v = K$. For any nonconstant finite Blaschke product $B(z)$, $\langle v, B \rangle = 0$. Therefore, the only Blaschke product for which $J_2(v, cB)$ has a non-zero minimizer is $B(z) = 1$. The first JFD iteration would produce a component $u_1 = \left(1 - \frac{1}{|K|}\right) K$, and a residual equal to $v_1 = \frac{K}{|K|}$. The second iteration (and every subsequent iteration) would produce a component equal to zero, and a residual equal to v_1 . Essentially the JFD algorithm terminates after one iteration yielding an decomposition of K into a component: $\left(1 - \frac{1}{|K|}\right) K$, and an error term: $\frac{K}{|K|}$.

This situation isn't too bad if $|K| \gg 1$. Clearly this decomposition is essentially meaningless if $|K| < 1$. If $|K| < 1$, the resulting error is greater than the input signal!

In order to get a better feel for what is going on, let's look at a more complicated example. Consider an input signal v which is analytic on D with a Taylor series given by $v = \sum_{k=1}^{\infty} d_k z^k$, for which the minimizer of $J_2(v, cB)$ is cz^m . If this is the case, then the optimal coefficient c is given by

$$c = \left(1 - \frac{1}{|d_m| \sqrt{1 + m}} \right) d_m.$$

The first iteration of the JFD algorithm will return a component u_1 and a residual v_1

given by

$$\begin{aligned}
u_1 &= \left(1 - \frac{1}{|d_m| \sqrt{1+m}}\right) d_m z^m \\
v_1 &= v - u_1 \\
&= \left(\sum_{k=1, k \neq m}^{\infty} d_k z^k\right) + \frac{1}{\sqrt{1+m}} \frac{d_m}{|d_m|} z^m
\end{aligned}$$

It is now clear that the JFD algorithm is doing something very similar to the simpler example in Section 4.4.1. At each iteration, a coefficient is truncated down to the level $\frac{1}{\sqrt{1+m}}$.

This suggests a stopping criterion for the JFD algorithm: terminate the algorithm after the n^{th} iteration if the Blaschke product B which minimizes $J_2(v_n, cB)$ (where c is the optimal coefficient) is such that:

$$|\langle v_n, B \rangle| \leq \frac{1}{\sqrt{1 + \deg(B)}} \quad (6.7)$$

At every step in the decomposition, this guarantees that that the act of extracting a component is actually removing energy from the signal. If an input signal were such that $\forall B(z)$, $|\langle v_n, B \rangle| \leq \frac{1}{\sqrt{1 + \deg(B)}}$, then “extracting” a component using the JFD algorithm would actually be adding energy to the signal. For example, take $K < 1$ in the earlier example with the constant signal.

It is important to note that this stopping criterion does not guarantee that after the algorithm terminates, that all inner products $|\langle v_n, B \rangle|$ will be less than their respective threshold values $\leq \frac{1}{\sqrt{1 + \deg(B)}}$. The only guarantee is that when the algorithm terminates, that the next component selected would have satisfied this condition.

6.4 Implementing the Algorithms

We now have all the tools in place to implement the JFD and IBA algorithms.

Single and double subscript indices will be used to refer to particular entries of vectors or matrices, respectively. In keeping with Matlab’s conventions, a colon will be used to refer to an entire row or column of a matrix. e.g. $M_{2,:}$ refers to the entire second row of the matrix M .

A bar over a quantity denotes complex conjugation.

$\mathbf{f} \in \mathbb{C}^{24}$ is a column vector whose entries are the boundary values of the input function, sampled at the points $z_n = e^{n\pi i/12}$, $n = 0, 1, \dots$

$\mathbf{B} \in M_{50388 \times 24}(\mathbb{C})$. Each row of \mathbf{B} contains boundary values of a unique element of $B_{\{a_n\}}^{\leq 12}$, sampled at z_n . Code used to produce \mathbf{B} is in Appendix 8.2.1.

$\mathbf{d} \in \mathbb{Z}^{50388}$ is a vector containing the degrees of the Blaschke products in the corresponding rows of \mathbf{B} . i.e. \mathbf{d}_7 is the degree of the Blaschke product whose boundary values are given by $\mathbf{B}_{7,:}$ - the seventh row of \mathbf{B} .

6.4.1 IBA Pseudocode

```

r := f
for  $n = 1, \dots, N$ 
    k :=  $\langle f, B \rangle$ 
     $l := \text{find}(\max |k|)$ 
     $\mathbf{U}_{:,n} := k_l \mathbf{B}_{l,:}$ 
    r := r -  $\mathbf{U}_{:,n}$ 
end

```

Matlab code for the IBA algorithm is in Appendix 8.2.2.

6.4.2 JFD Pseudocode

```

r := f
for  $n = 1, \dots$ 
    k :=  $\langle f, B \rangle$ 
     $l := \text{find} \left( \max \left| |k| - \frac{1}{\sqrt{1 + \text{deg} B}} \right| \right)$ 
    if  $|\langle f, B \rangle| \leq \frac{1}{\sqrt{1 + \text{deg}(B_l)}}$  then break
    k :=  $\left( 1 - 1/(2\pi |\mathbf{k}| \sqrt{1 + \mathbf{d}}) \right) \mathbf{k}$  note
     $\mathbf{U}_{:,n} := k_l \mathbf{B}_{l,:}$ 
    r := r -  $\mathbf{U}_{:,n}$ 
end

```

note: The operations in this line are all evaluated componentwise, not with the usual vector operations. These componentwise operations are performed using the Matlab componentwise operations which are prefixed by a period. This line could be replaced with a loop which would be formally more correct, however the componentwise operations correspond to the

Matlab implementation used in this work. There is no ambiguity or problem when written in Matlab code.

Matlab code for this algorithm is in Appendix 8.2.3.

6.4.3 A further note on stopping criteria

When implementing the IBA and JFD algorithms on a computer, machine errors will inevitably arise. It would be nonsensical to force an algorithm to continue a decomposition if the input signal for a given iteration is indistinguishable from machine error. Thus, an appropriate stopping criterion for use on Matlab is: If $\|\text{residual}\| < \|\text{original input}\| 10^{-14}$ then stop.

The purpose of this investigation is to compare the IBA and JFD algorithms to each other. This can be used to give another stopping criterion for the IBA algorithm. It makes sense to consider at least as many terms for IBA as the JFD returns. Also, because the residual of the JFD algorithm does not converge to 0, JFD will typically stop when there is still a non-trivial amount of energy in the residual. Thus, we should continue IBA for at least a few terms more than the JFD algorithm returns, in order to extract some information from the residual. The stopping criterion we will use for the IBA algorithm is: IBA will stop if that the number of IBA terms is equal to (number of JFD terms)+8.

As a further safety measure, both the IBA and JFD algorithms will be terminated if the norm of the residual increases.

6.5 Examples and Discussion

6.5.1 Calibration

Example 1a

It seems appropriate to begin investigating new decomposition techniques with a calibration exercise. We start by performing IBA and JFD decomposition on a polynomial, given by

$$f(z) = 1000 + 100z^{10} + 10z^7 + z^2 + 0.1z^6$$

All the components of $f(z)$ are orthogonal to each other. It is reasonable to expect both methods to resolve $f(z)$ into the monomials above. Of course, JFD should only resolve those terms whose coefficients are large enough to pass the stopping criterion. However, for the larger terms the IBA and JFD results should appear very similar to each other. Matlab code for this example is in Appendix 8.2.4, as ‘Example 1a’.

The IBA decomposition performs as expected, producing:

$$f(z) = 1000 + 100z^{10} + 10z^7 + z^2 + 0.1z^6.$$

The JFD algorithm did almost as good, recovering the three terms with the largest coefficients:

$$f(z) \approx 1000 + 100z^{10} + 10z^7.$$

The residual from the JFD decomposition has norm $\|\text{residual}\| = 1.0202$, while the IBA's residual is essentially zero ($\|\text{residual}\| = 1.14 \times 10^{-14}$).

Before performing any more actual decompositions, it would be helpful to get a better idea of exactly what the IBA and JFD algorithms are doing, and how they relate to each other.

6.5.2 Reconstruction Accuracy

With these examples, we investigate how well the JFD and IBA decompositions approximate certain input signals. We start by constructing polynomials of the form

$$f_{m,r}(z) = m \sum_{n=0}^9 r^n z^{12-n}$$

By varying m and r , the task of separating different modes can be made easier or harder. If $r \gg 1$, the monomial components of $f(z)$ should be very easy to extract. If $r \approx 1$, the task of separating the individual modes may become more difficult. The parameter m ideally shouldn't affect the outcome of the IBA algorithm. However, small values of m should cause the JFD algorithm to terminate after extracting only a few (or possibly none) of the components. As m increases, the results of the JFD algorithm should begin to resemble those of the IBA algorithm.

In figures 6.1 and 6.2, the weighted residual $\log_{10}(\text{residual}) - \log_{m,r,9}(c)$ is plotted against m and r . The upper right regions of these plots are empty due to NaN and Inf results for the residuals. This is not surprising given the extremely large values of m and r in the upper right region of the plots. The missing residual values can be thought of as being essentially zero.

As predicted, the IBA algorithm is unaffected by the m value chosen. There is a clear dependence on r , but m has no visible effect on the residual of IBA. There is a distinct r value at which the character of the IBA algorithm changes dramatically. A more precise calculation reveals that this cutoff value is $\log(r) \approx 2.9$ for IBA, and $\log(r) \approx 3.36$ for JFD.

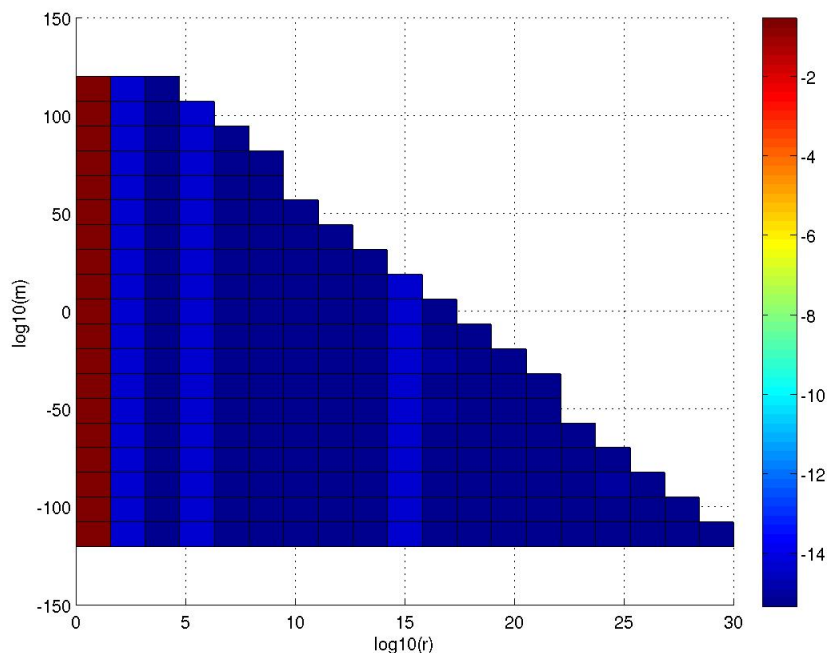


Figure 6.1: $\log(\text{reduced residuals})$ for IBA algorithm with input signals $f_{m,r}(z)$

(Code for this calculation is in Appendix 8.2.4. The IBA code is ‘example2c.m’; the JFD code is ‘example2d.m’.)

There are two clearly distinct regions³ of the graph in figure 6.2. The region $\log(m) > 0$ appears almost identical in character to figure 6.1. In this region, the JFD algorithm behaves in a quasi-linear fashion. Recall that at every iteration, the JFD algorithm selects a Blaschke product as a component, and extracts most - but not all - of the correlation between the signal and the new component. In the quasi-linear ($\log(m) > 0$) region, the constant m is large enough that these little remainders are negligible.

In the region $\log(m) < 0$, the JFD algorithm takes on a completely different character from the IBA algorithm. The residuals depend strongly on both variables m and r .

We wish to further explore the question of reconstruction accuracy, and how it can be used to illustrate the nonlinearity of JFD. It is possible that the observations made from figures 6.1 and 6.2 were a special consequence of the orthogonality of the components of the input signal. It is important to ask what such results look like when we choose an input signal from $B_{\mathcal{P}}^{\leq 12}$ that is not an ordinary polynomial. For this exercise, we use input signals

³One might prefer to partition the m, r plane into two partitions according to the size of the reduced residual (i.e. the color of the graph in figure 6.2). However, one loses the distinction between one quasi-linear and one nonlinear region, as both regions’ boundaries depend on r .

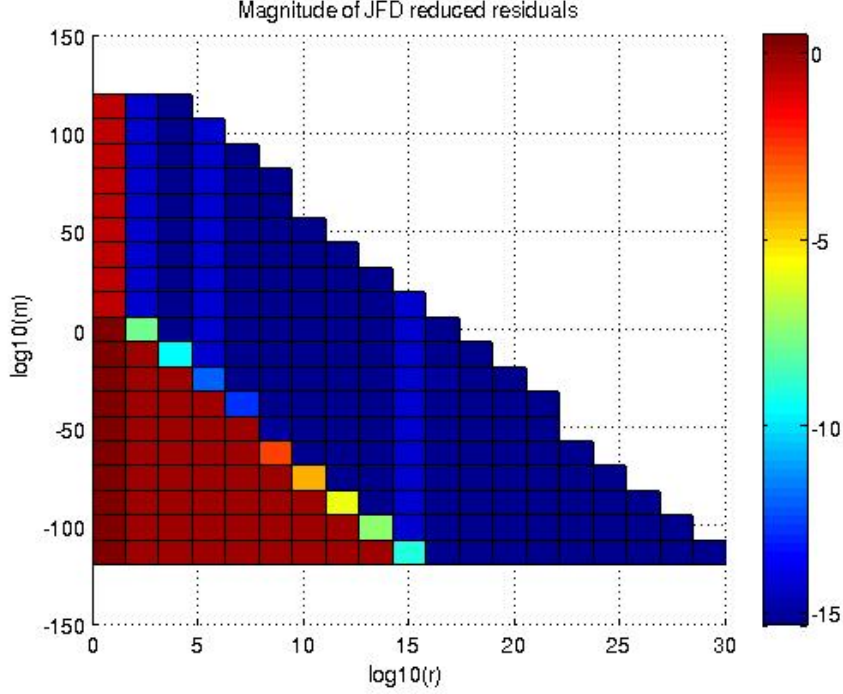


Figure 6.2: $\log(\text{reducedresiduals})$ for JFD algorithm with input signals $f_{m,r}(z)$

of the form:

$$g_{m,r}(z) = m \left(r^1 z^3 \zeta_1 \zeta_2 \zeta_3^3 \zeta_4^2 \zeta_5 \zeta_6 + r^2 z \zeta_1 \zeta_2^2 \zeta_3^2 \zeta_4^3 + r^3 \zeta_1 \zeta_2^2 \zeta_3 \zeta_4^3 \right. \\ \left. + r^4 \zeta_1 \zeta_2^2 \zeta_4 \zeta_5 + r^5 \zeta_1 \zeta_2 \zeta_6 + r^6 \zeta_4 \right)$$

The analysis for $\{g_{m,r}\}$ is the same as for $\{f_{m,r}\}$. The reduced residuals are plotted as a function of m and r in figure 6.3 and 6.4. There are some strong similarities, and also differences in the behavior of the IBA algorithm in figures 6.1 and 6.3. In both cases, there is virtually no dependence on m . This is as expected, due to the linearity of IBA. However, in the first case (figure 6.1), there was a very sharp demarcation between the regime of bad approximation, $\log(r) < 2.9$, and the region of good approximation, $\log(r) > 2.9$. In figure 6.3, there is a gradual improvement in the approximation obtained from IBA as r increases. For any given accuracy level, there is a value of r that will achieve satisfactory results. However, the location of this threshold clearly depends upon the desired r value, and not on some obvious feature of the residuals themselves. This is somewhat disappointing; it would be nice to have some a priori knowledge about the character of the results. However, it is not all that surprising that when analyzing a signal with non-orthogonal components, that some features of the analysis might become “smeared”. This is typical of complex signals.

The JFD algorithm’s results are also interesting. It might be hard to definitively spot the

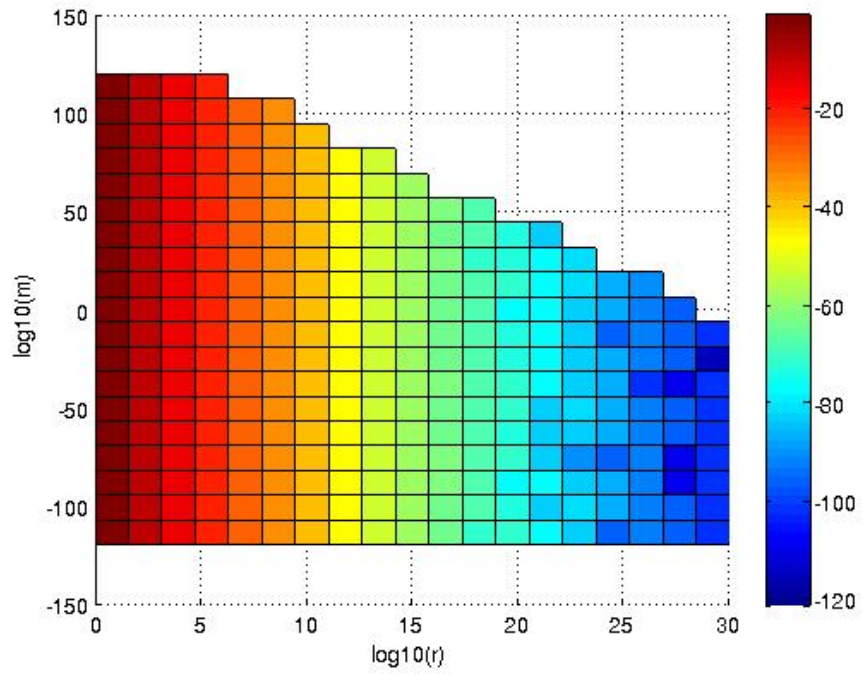


Figure 6.3: $\log(\text{reducedresiduals})$ for IBA algorithm with input signals $g_{m,r}(z)$

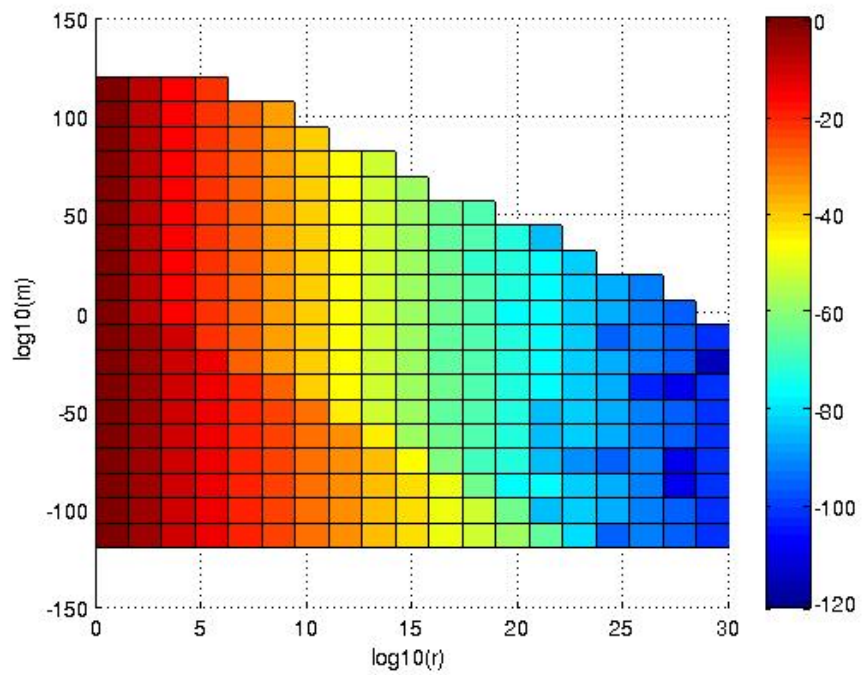


Figure 6.4: $\log(\text{reducedresiduals})$ for IBA algorithm with input signals $g_{m,r}(z)$

two regions on the graph in figure 6.4, had we not already seen the same analysis done for the polynomial signals $f_{m,r}$. It is fairly clear that there is a similar demarcation between a pseudo-linear region and a nonlinear region in figure 6.4. In the region given by $\log(m) > 0$, there is essentially no dependence on m

6.5.3 Identification of localized frequencies

In the final example, we take a look at how IBA and JFD algorithms perform with signals that are not sums of elements of $B_{\{a_n\}}^{\leq 12}$. Specifically, if an input signal is a Blaschke polynomial which is close to another Blaschke polynomial (in the sense that the roots are close to each other), will the IBA and JFD algorithms extract those components in the constraint set which are close to the modes contained in the original signal?

We start by looking at a single Blaschke monomial: $f(z) = 10000\tilde{\zeta}_3^4\tilde{\zeta}_1$, where the factors $\{\tilde{\zeta}_n\}$ have roots which have been slightly perturbed from the roots of $\{\zeta_n\}$. Matlab code for this example is in the file ‘example4a.m’, in Appendix 8.2.4.

The IBA and JFD algorithm returned very similar decompositions for the perturbed Blaschke product. They agree up to the first six terms. (The two decompositions differ slightly on the sixth term.) Those terms are:

$$f(z) \approx (9997+31i)\zeta_3^4\zeta_1 - (120+131i)\zeta_1\zeta_3^3\zeta_4^3 - (118+72i)\zeta_3^4\zeta_4\zeta_7^3 - (35-73i)\zeta_3^4\zeta_4^2 + (77-i)\zeta_3^4\zeta_4^3\zeta_5\zeta_7$$

The decomposition above is very encouraging. Of course, a shorter series might have been nicer (only some of the terms are shown above), but the result we obtained is not that surprising. Perturbing a Blaschke product essentially introduces aliasing to the frequency content of the boundary function. This can be seen by looking at the instantaneous frequency of a Blaschke product. It is a sum of Poisson kernels, each depending on one of the Blaschke root locations. Thus, shifting a root corresponds to shifting one of those Poisson kernels along the boundary ∂D (as well as possibly squeezing or widening it). Whenever aliasing is introduced, it is reasonable to expect that many (small) terms will be needed in order to clean up the signal.

We finish with an example that is only slightly more complicated than the previous one: $f(z) = 10000\tilde{\zeta}_3^4\tilde{\zeta}_1 - 800z\tilde{\zeta}_1$. The IBA and JFD algorithms return identical decompositions. (This example was run with a maximum of 20 terms.) The first few terms are shown below.

$$f(z) \approx (15111 - 2953i)\zeta_3^4\zeta_1 - (7640 + 632i)\zeta_1\zeta_2\zeta_3 - (1811 + 4042i)\zeta_2\zeta_3\zeta_4 + \dots$$

The first term in the series is the Blaschke product we would have expected. The input signal’s leading term was a perturbation of the Blaschke product in the leading term. However

that is where our luck runs out. Because the next term in the input signal has a smaller coefficient, that information gets swamped by the residual from the first term's extraction. The first term of the approximant is not a perfect fit for the leading term in the input signal, leaving a remainder of $\left(10000\tilde{\zeta}_3^4\tilde{\zeta}_1 - (15111 - 2953i)\zeta_3^4\zeta_1\right)$. Clearly, this is much larger than the second input term $-800z\tilde{\zeta}_1$, and this frequency information cannot be recovered cleanly.

While it is possible to set up more thorough experiments on this matter, it does not seem a wise application of resources. After all, with only 24 data points, aliasing artifacts will no doubt have a major impact on any experiments done with more than two perturbed terms. There is a need for a thorough understanding of sampling effects with Blaschke polynomials before trying to pack much more information into such a short data set.

Chapter 7

Conclusions and Future Research

The research presented in this work is motivated by a desire to better understand the empirical mode decomposition (EMD). As it turns out, the original results presented here do not directly pertain to EMD. However, it is important to keep the larger goal in mind when looking at these results in order to better see how they fit together. EMD presents the mathematical researcher with an ages old riddle: something seems to work without any apparent explanation of why or how. By investigating some of the questions that this riddle poses, we have come a little bit closer to possibly understanding how EMD works. As is so often the case with mathematical research, we have stumbled upon answers to other questions as well.

7.1 ASNIFs

The first problem posed by EMD addressed in this work a foundational one. The IMF components produced by EMD are supposed to have positive instantaneous frequency[13]. This is certainly true much of the time, but counterexamples have been shown to exist[20]. This leads to two questions:

- What functions are characterized by the property of non-negative instantaneous frequency?
- Why is it that, in spite of the counterexamples produced, the IMFs resulting from real-world applications tend to have non-negative instantaneous frequency?

In Chapter 3, the first question has been answered for functions which admit an analytic continuation to an open set which is “big enough”. A function which admits such an analytic continuation is an *analytic signal with non-negative instantaneous frequency* (ASNIF) if and only if its canonical factorization satisfies precise conditions. (See corollaries 3.2.10 and

3.2.14.) These results are an extension of previous results which specified smaller classes of ASNIFs ([8],[24]). In obtaining these results, a natural setting was found for further investigation into larger classes of ASNIFs: Hardy spaces.

The results of chapter 3 also shed some light on the second question. The conditions of theorems 3.2.9 and 3.2.13 are strikingly similar to the experimental outcomes of the sifting algorithm: the rate of amplitude modulation of a given IMF is typically much lower than the instantaneous frequency of the FM part of the IMF([13]). Of course this is not guaranteed, otherwise there wouldn't be any counterexamples! However it is interesting to see that the typical output of the sifting algorithm matches quite nicely with the ASNIF criteria found in Chapter 3.

7.1.1 Future research

Hardy spaces contain a rich assortment of functions which do not satisfy the criteria of theorems 3.2.9 and 3.2.13, but which may be considered ASNIFs in a more general setting. Further research into instantaneous frequency for these functions would require careful consideration of boundary convergence. This is no small task; ensuring the convergence of the boundary values of the first derivative is a difficult problem. (See [9], [11]) However, at least it is clear how to proceed in order to obtain more general results.

Another research problem of more direct practical importance has arisen from Chapter 3: a new potential stopping criterion¹ for the sifting algorithm. This criterion can be stated most directly as: Continue sifting until the signal is an IMF. The results of Chapter 3 indicate that this would not be very difficult. However, it would require looking not only at the real data values (as EMD currently does), but at the analytic signal of the candidate IMF. Currently, HHT only constructs the analytic signals after all the sifting is completed as an a posteriori analysis tool. The counterexamples in [20] show that this is insufficient; simple criteria based solely on the real valued signal are insufficient to guarantee non-negative instantaneous frequency. Introducing a new stopping criterion to the sifting algorithm (and the fundamental change to the algorithm itself that it implies) obviously presents many new avenues for research, as it would create an entirely new signal processing method.

The result presented in the later chapters of this work represent the first small steps on another research program suggested by the ASNIF results. The factorization of ASNIFs into component building blocks suggests a systematic way to build incrementally more general ASNIF based signal decomposition/analysis methods. Various classes of ASNIFs may be

¹The suggestion being made here is not necessarily that this condition would be a sufficient stopping criterion. Rather that the criterion proposed here could be used to guarantee that an IMF is indeed an ASNIF. Other criteria may still be desired in order to ensure that IMFs have other desirable properties, such as relatively smooth amplitude modulation.

defined. These may be used as sets of component functions for approximation methods whose complexity increases with the size of the component set. Some such sets are:

- The set of all Fourier basis functions. ($\{z^n\}$ for $H^p(D)$, and $\{e^{inz}\}$ for $H^p(\mathbb{C}_+)$)
- The set of all Blaschke products over a given set of roots.
- The set of all Blaschke products.
- The set of all singular inner functions.
- The set of all inner functions.
- The set of all ASNIFs which satisfy theorem 3.2.9 or 3.2.13.
- The set of all ASNIFs.

While these sets are not well-ordered by the containment relation, there is a general progression suggested by this list. Larger classes of ASNIFs allow decompositions which impose fewer a priori restrictions on the component functions generated. Each class of ASNIFs has distinct strengths and weaknesses which may make it suitable to a particular signal processing application.

A decomposition which used all ASNIFs as approximants (or even those which satisfy Theorem 3.2.9 or 3.2.13) would be very interesting to compare directly to EMD/HHT. This would go a long way to settling many of the outstanding theoretical questions about EMD/HHT.

On a less applied note, a decomposition or approximation method using the set of all inner functions (for either $H^p(\mathbb{C}_+)$ or $H^p(D)$) would present some fascinating problems for functional analysis. Such a method would be a direct extension of the Fourier series/transform methods. Such an *inner function decomposition* would be especially fascinating, as this set of functions is the largest set of pure frequency (or “pure FM”) functions that exists within the Hardy spaces. Clearly this presents many avenues of potentially fruitful research.

7.2 Blaschke polynomials

Theorem 5.1.6 (the fact that the linear span of the set of all Blaschke products (in $H^p(D)$ or $H^p(\mathbb{C}_+)$) is a polynomial space) is not a difficult result. It is not even a surprising result. However, it is significant because it makes it explicitly clear that spaces such as $\mathbb{C}[Z_P]$ are natural generalizations of ordinary polynomials $\mathbb{C}[z]$. This is very important, as it suggests

that many tools and methods from ordinary polynomials may translate quite naturally to Blaschke polynomials.

One of the most obvious characteristics of Blaschke polynomial spaces is their sheer size. Using only seven Blaschke roots, and degrees ≤ 12 , the set $B_{\{a_n\}}^{\leq 12}$ contains 50388 elements. This was quite manageable using modern (2007) computer hardware. It might not even be a problem to use a few orders of magnitude more Blaschke products. However, any given computer will be very easy to overburden, as the size of such a set of Blaschke products grows super-exponentially with respect to the number of Blaschke roots². The obvious implication is that the method used in this work - choosing a set of Blaschke roots, creating a list of all possible Blaschke products over those roots, and calculating all the $\langle v, B \rangle$ quantities - is only practical for data sets of a known (short) length.

For any application where the data set is of unknown length, it is impossible to control the size of the set of Blaschke products without inappropriately restricting the degree of the Blaschke polynomials used³. Thus, the method used in this work is probably only useful for simple test cases, or to obtain calibration results.

7.2.1 Future research

The IBA and JFD algorithms represent a small first step into the world of Blaschke polynomial approximation methods. The IBA method is barely a departure from a classical Fourier series. The only difference is that the set of basis functions has been expanded. The JFD algorithm has much more in common with contemporary methods. It is nonlinear, but is still well behaved enough that it may be thought of as pseudo linear, at least in some cases.

As implemented, these algorithms are unlikely to be of any practical use. The brute force approach used (building a list of Blaschke products, and searching through the list for each coefficient) does not scale well. The super-exponential growth rate of a set $B_{\mathcal{S}}^{\leq n}$ with respect to $|\mathcal{S}|$ means new methods are needed to minimize any functional used in a J -functional decomposition. For the J_1 and J_2 functionals used in this work, there is a way to write an algorithm that would scale much better with $|\mathcal{S}|$ and n . Rather than evaluating all possible $\langle v, B \rangle$ values to find the minimizer of $J(v, cB(z))$ (which requires constructing a very long list of Blaschke products), it should be possible to search for the optimum Blaschke monomial of a given degree. This would be done as follows:

1. Solve for the optimum coefficient c as a function of the Blaschke product $B(z)$ and

²The size of a set of Blaschke products is given by $|B_{\mathcal{S}}^{\leq n}| = \binom{|S|+n}{n}$. The situation gets even worse under the reasonable assumption that both n and $|\mathcal{S}|$ increase proportionally with the length of the data set.

³This assumes that the appropriate number of Blaschke roots and the maximum degree of the Blaschke products are increasing functions of the length of the data set. This may or may not be true

input function v .

2. Write the J -functional as a function of v , and the zero locations of $B(z)$.
3. Use a search method to minimize $J(v, \zeta_1, \zeta_2, \dots, \zeta_n)$.

This seems like a simple proposition, but there are some significant problems to be resolved in order to realize such an algorithm. For a given J -functional, a continuity result is needed of the form:

$$d((\zeta_1, \zeta_2, \dots, \zeta_n) - (\xi_1, \xi_2, \dots, \xi_n)) < \delta \Rightarrow |J(v, \zeta_1, \zeta_2, \dots, \zeta_n) - J(v, \xi_1, \xi_2, \dots, \xi_n)| < \epsilon$$

For functionals like J_1 and J_2 used in this work, this might not be too difficult. However, it is important to establish such a result in order to proceed. The most difficult part of such an algorithm may be minimizing the functional $J(v, \zeta_1, \zeta_2, \dots, \zeta_n)$. If $J(v, cB(z))$ is nonlinear in $cB(z)$ and/or v , the functional $J(v, \zeta_1, \zeta_2, \dots, \zeta_n)$ may turn out to be incredibly complicated. The difficulty of this optimization would increase with n . However even if this technique is incredibly difficult, it may still allow for a significant improvement over the simpler approach used in this work. The list of Blaschke products grows super-exponentially in n . If the search algorithm proposed here is even slightly faster than that, the rewards will become clear as n increases.

Ironically, the potential acceleration due to this more general approach requires using a larger Blaschke polynomial space than the slower brute force method of enumerating all Blaschke products. Performing a search over Blaschke root locations means there would be no prescribed set of Blaschke polynomials. The size of the approximating set is therefore determined by the precision of the search.

A cautionary note must be made regarding the precision of a search for approximants. Determining how precisely one needs to, or is able to, determine an approximant, is analogous to having a sampling theorem and an uncertainty principle, respectively. For example, if one were performing a Blaschke J -functional decomposition on a data set of 10000 samples, it would be important to know what the maximum degree of approximants should be. The Nyquist rate was used as a rough guide in this work, but a more precise result would be valuable for such a large data set. A rule which selects the maximum degree necessary to analyze a signal would essentially be a generalized sampling theorem. Once a maximum degree were established, a procedure for selecting Blaschke roots is needed. The precision with which a set of Blaschke roots is determined implicitly determines how well the set of component functions will be able to localize frequency. This is clearly equivalent (in some sense) to an uncertainty principle. Knowing the nature of (whatever is analogous to) the

uncertainty principle at work would be very valuable, as it would allow one to know how much precision is needed in the calculation of the Blaschke roots. Needless to say, it would also be invaluable for the interpretation of the results.

Bibliography

- [1] C. Bennett and R. Sharpley. *Interpolation of Operators*. Academic Press Inc., Boston, MA, 1988.
- [2] Tony F. Chan, Jianhong Shen. *Image Processing and Analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
- [3] Leon Cohen. *Time-Frequency Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [4] Ingrid Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
- [5] Eric Delchelle, Jacques Lemoine, and Oumar Niang, Empirical Mode Decomposition: An Analytical Approach for Sifting Process. *IEEE Signal Processing Letters*, Vol. 12, NO. 11, 2005.
- [6] Ronald A. DeVore. Nonlinear approximation. *Acta Numerica*, pp. 51-150. Cambridge Univ. Press, Cambridge, 1998.
- [7] D.L. Donoho & I.M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. Technical Report, Department of Statistics, Stanford University. 1992.
- [8] M. Doroslovački, On nontrivial analytic signals with positive instantaneous frequency, *Signal Processing*, 83 (2003) pp. 655-658.
- [9] P. L. Duren. *Theory of H^p Spaces*. Academic Press, New York. 1970.
- [10] <http://perso.ens-lyon.fr/patrick.flandrin/emd.html>
- [11] J. B. Garnett. *Bounded Analytic Functions*. Academic Press, New York. 1981.
- [12] C. Gasquet and P. Witomski. *Fourier Analysis and Applications*. Springer-Verlag, New York. 1999.

- [13] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. London, Ser. A*, **454**, 903-995. 1998.
- [14] N. E. Huang, and S. Shen (Ed.s). *Hilbert-Huang Transform and its Applications*. World Scientific Publishing, 2005.
- [15] Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, New York, 1989.
- [16] Reginald M. Meeson, Jr. HHT Sifting and Filtering. *Hilbert-Huang Transform and its Applications*. World Scientific, Hackensack, NJ. 2005.
- [17] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*. University Lecture Series, 22, AMS, 2002.
- [18] Joseph J. Rotman. *Advanced Modern Algebra*. Pearson Education, Upper Saddle River, 2002.
- [19] Leonid Rudin, Stanley Osher, Emad Fatemi. *Nonlinear total variation based noise removal algorithms*. *Physica D*, 60 (1992) 259-268.
- [20] R. Sharpley and V. Vatchev. Analysis of the intrinsic mode functions. *Constructive Approximation* VOL 24, number 1, pp. 17-47. Springer-Verlag, 2006.
- [21] E. Tadmor, S. Nezzar, L. Vese. A multiscale image representation using hierarchical (BV, L^2) decompositions, UCLA CAM Report 03-32, Department of Mathematics, University of California, Los Angeles, CA, 2003.
- [22] V.N. Temlyakov. Nonlinear methods of approximation. Technical Report 0109, Dept. of Mathematics, University of South Carolina, Columbia, SC 29208, 2001.
- [23] E. C. Titchmarsh, *Introduction to the theory of Fourier integrals*. Oxford University Press. 1948.
- [24] X.-G. Xia, L. Cohen, "On Analytic Signals with Nonnegative Instantaneous Frequency." Proceedings of the ICASSP-99, Phoenix, March 1999, Paper 1483.

Chapter 8

Appendix

8.1 Proofs

8.1.1 Proof of equation 3.16

Proof. Let $B(z)$ be a Blaschke product for $H^p(\mathbb{C}_+)$, with a discrete zero set $\{z_k\}$. (i.e. $\{z_k\}$ has no cluster points.) Such a function is meromorphic on the entire complex plane. Wherever $B(z)$ is analytic, we have

$$\begin{aligned}\omega_B(z) &= \operatorname{Im} \left[\frac{B'(z)}{B(z)} \right] \\ &= \operatorname{Im} \left[\sum_k \frac{z - \bar{z}_k}{z - z_k} \frac{d}{dz} \left(\frac{z - z_k}{z - \bar{z}_k} \right) \right] \\ &= \operatorname{Im} \left[\sum_k \frac{z - \bar{z}_k}{z - z_k} \left(\frac{z_k - \bar{z}_k}{(z - \bar{z}_k)^2} \right) \right] \\ &= \operatorname{Im} \left[\sum_k \frac{z_k - \bar{z}_k}{|z - z_k|^2} \right] \\ &= \sum_k \frac{2y_k}{|z - z_k|^2}\end{aligned}$$

Now, $\{z_k\}$ has no cluster points in \mathbb{C} , and thus no cluster points on \mathbb{R} . Therefore $B(z)$ is analytic on the real line, and there is no problem taking the limit

$$\begin{aligned}\omega_B(x) &= \lim_{z \rightarrow x \in \mathbb{R}} \omega_B(z) \\ &= \sum_k \frac{2y_k}{|x - z_k|^2} > 0\end{aligned}$$

□

8.1.2 Proof of equation 3.17

Proof. Let $S(z)$ be a singular inner function for $H^p(\mathbb{C}_+)$, whose singular measure $d\nu$ has discrete support. Such a function is analytic on $\mathbb{C} \setminus (\text{supp } d\nu)$. Wherever $S(z)$ is analytic, we have

$$\begin{aligned}\omega_S(z) &= \text{Im} \left[\frac{S'(z)}{S(z)} \right] \\ &= \text{Im} \left[i \frac{d}{dz} \int_{-\infty}^{\infty} \frac{1+tz}{t-z} d\nu(t) \right] \\ &= \text{Re} \left[\frac{d}{dz} \int_{-\infty}^{\infty} \frac{1+tz}{t-z} d\nu(t) \right]\end{aligned}$$

Wherever $S(z)$ is analytic, its derivative exists, so there is no problem switching the order of the derivative and the integration above. (This becomes an issue if we wish to evaluate the limit of $\omega_S(z)$ as z approaches a cluster point of the singular support.)

$$\begin{aligned}\omega_S(z) &= \text{Re} \left[\frac{d}{dz} \int_{-\infty}^{\infty} \frac{1+tz}{t-z} d\nu(t) \right] \\ &= \text{Re} \left[\int_{-\infty}^{\infty} \frac{t(t-z) + (1+tz)}{(t-z)^2} d\nu(t) \right] \\ &= \text{Re} \left[\int_{-\infty}^{\infty} \frac{1+t^2}{(t-z)^2} d\nu(t) \right]\end{aligned}$$

Because the singular support is discrete, there is no problem taking the limit of $\omega_S(z)$ as $z \rightarrow x \in \mathbb{R}$ almost everywhere, to give:

$$\begin{aligned}\omega_S(x) &= \text{Re} \left[\int_{-\infty}^{\infty} \frac{1+t^2}{(t-x)^2} d\nu(t) \right] \\ &= \int_{-\infty}^{\infty} \frac{1+t^2}{(t-x)^2} d\nu(t) \geq 0 \text{ a.e.}\end{aligned}$$

□

8.1.3 Proof of the convexity of J_2 w.r.t. a and b

Proof.

$$\begin{aligned}
 J_2(a, b; v, B) &= \frac{\sqrt{\pi}}{2} (\|v\|^2 - 2a\operatorname{Re}(\langle v, B \rangle) - 2b\operatorname{Im}(\langle v, B \rangle) + a^2 + b^2) \\
 &\quad + \sqrt{a^2 + b^2} \sqrt{\frac{\pi}{1 + \deg(D)}} \\
 \frac{\partial^2}{\partial a^2} J_2 &= \sqrt{\pi} + \frac{2a^2 + b^2}{(a^2 + b^2)^{3/2}} \sqrt{\frac{\pi}{1 + \deg B}} \geq 0 \\
 \frac{\partial^2}{\partial b^2} J_2 &= \sqrt{\pi} + \frac{a^2 + 2b^2}{(a^2 + b^2)^{3/2}} \sqrt{\frac{\pi}{1 + \deg B}} \geq 0
 \end{aligned}$$

Furthermore, $\frac{\partial^2}{\partial a^2} J_2$ and $\frac{\partial^2}{\partial b^2} J_2$ are only equal to zero when both a and b are equal to zero. i.e. $\frac{\partial^2}{\partial a^2} J_2, \frac{\partial^2}{\partial b^2} J_2 \neq 0 \forall (a, b) \neq (0, 0)$. Thus J_2 is strictly convex in a and b . \square


```
exponents(8,:) = sum(exponents,1);
BPlist(25,:) = complex(exponents(8,:));
exponents = sortrows(exponents',8)';
BPlist = sortrows(BPlist',25)';
degree = exponents(8,:);
exponents = exponents(1:7,:);
BPlist = BPlist(1:24,:);
save exponents exponents; save degree degree; save BPlist BPlist;
clear
```

8.2.2 IBA.m

```
function [c,e,res]=IBA(f,m)

load BPlist
load exponents
R=f;

for n=1:m
    k = (BPlist')*R(:,n)/24;
    [C,1] = max(abs(k));
    R(:,n+1) = R(:,n) - k(1)*(BPlist(:,1));
    if (sum(abs(R(:,n+1).^2)) > sum(abs(R(:,n)).^2))
        break
    end
    if (sqrt(sum(abs(R(:,n)).^2))/24) ...
        / sqrt(sum(abs(R(:,1)).^2))/24 < 10^-14)
        break
    end
    c(1,n) = k(1);
    e(1,n) = 1;
end
res = sqrt(sum(abs(R).^2,1)/24);
```

8.2.3 JFD.m

```
function [c,e,res]=JFD(f)

load BPlist
load degree
c(1,1) = 0;
e(1,1)=0;
f=f(:);
R(:,1) = f;

for n=1:100
    k= (BPlist') * R(:,n) / 24;
    [C,l] = max(abs(abs(k) - sqrt(1+degree')));
    if (abs(k(l)) <= (1+degree(l))^( -0.5))
        break
    end
    kk=(1-(2*pi*abs(k(l))*sqrt(1+degree(l)))^-1).*k(l);
    R(:,n+1) = R(:,n) - kk*BPlist(:,l);
    if (sum(abs(R(:,n+1)).^2) > sum(abs(R(:,n)).^2))
        break
    end
    if (sqrt(sum(abs(R(:,n)).^2)/24) ...
        / sqrt(sum(abs(R(:,1)).^2)/24) < 10^-14)
        break
    end
    c(1,n) = kk;
    e(1,n) = l;
end

res = sqrt(sum(abs(R).^2,1)/24);
```

8.2.4 Examples

example1.m

```
%Example 1

clear
load Bplist
f = 1000*Bplist(:,1) + 100 *Bplist(:,11441) + 10 * Bplist(:,1717) + ...
    Bplist(:,9) + 0.1*Bplist(:,793);
[c1,e1,JFDres]=JFD(f);
[c2,e2,IBares]=IBA(f,2*length(c1)+8);
JFDdecay = -diff(log(JFDres))
IBAddecay = -diff(log(IBares))
[JFDres(length(JFDres)) IBares(length(JFDres))]
JFDdraw = conj([complex(e1)' c1'])
IBAdraw = conj([complex(e2)' c2'])
```

example2a.m

```
%Example 2a

clear
z = exp(i * (0:23) * pi / 12); z=z(:);
k1=20;k2=20;
m=logspace(-120,120,k1);
r=logspace(0,30,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1= 1:k1
    for n2=1:k2
        f=zeros(24,1);
        for k=0:9
            f=f+m(n1)*r(n2).^k * z.^(12-k);
        end
        [c1,e1,IBAres]=IBA(f,20); ll=length(IBAres);
        res(n1,n2)=log10(IBAres(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
h=surf(R,M,res)
xlabel('log10(r)'); ylabel('log10(m)'); colorbar;
view(0,90); camproj('orthographic')
save ex2a res record
```


example2b.m

```
%Example 2b

clear
z = exp(i * (0:23) * pi / 12); z=z(:);
k1=20;k2=20;
m=logspace(-120,120,k1);
r=logspace(0,30,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1=1:k1
    for n2=1:k2
        f=zeros(24,1);
        for k=0:9
            f=f+m(n1)*r(n2)^k * z.^(12-k);
        end
        [c1,e1,JFDres]=JFD(f,20); ll=length(JFDres);
        res(n1,n2)=log10(JFDres(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
h=surf(R,M,res)
xlabel('log10(r)'); ylabel('log10(m)'); colorbar;
view(0,90); camproj('orthographic')
save ex2b res record
```

example2c.m

```
%Example 2c

clear
z = exp(i * (0:23) * pi / 12); z=z(:);
k1=1;k2=200;
m=40;
r=logspace(1,2,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1= 1:k1
    for n2=1:k2
        f=zeros(24,1);
        for k=0:9
            f=f+m(n1)*r(n2).^k * z.^(12-k);
        end
        [c1,e1,IBAres]=IBA(f,20); ll=length(IBAres);
        res(n1,n2)=log10(IBAres(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
h=plot(R,res)
xlabel('log10(r)'); ylabel('log of reduced residuals');
save ex2c res record
```

example2d.m

```
%Example 2d

clear
z = exp(i * (0:23) * pi / 12); z=z(:);
k1=1;k2=200;
m=40;
r=logspace(1,2,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1= 1:k1
    for n2=1:k2
        f=zeros(24,1);
        for k=0:9
            f=f+m(n1)*r(n2).^k * z.^(12-k);
        end
        [c1,e1,JFDres]=JFD(f,20); ll=length(JFDres);
        res(n1,n2)=log10(JFDres(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
h=plot(R,res)
xlabel('log10(r)'); ylabel('log of reduced residuals');
save ex2d res record
```

example3a.m

```
%Example 3a

clear
load Bplist
index = [40000 7000 2000 500 100 6 1];
BPs = Bplist(:,index); clear Bplist;
k1=20;k2=20;
m=logspace(-120,120,k1);
r=logspace(0,30,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1=1:k1
    for n2=1:k2
        f=zeros(24,1);
        for j=1:7
            f=f+m(n1)*r(n2).^(j-1) * BPs(:,j);
        end
        [c1,e1,IBares]=IBA(f,20); ll=length(IBares);
        res(n1,n2)=log10(IBares(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
surf(R,M,res)
xlabel('log10(r)'); ylabel('log10(m)'); colorbar;
view(0,90); camproj('orthographic')
save ex3a res record
```

example3b.m

```
%Example 3b

clear
load Bplist
index = [40000 7000 2000 500 100 6 1];
BPs = Bplist(:,index); clear Bplist;
k1=20;k2=20;
m=logspace(-120,120,k1);
r=logspace(0,30,k2);r=r(:);
[R,M]=meshgrid(r,m);R=log10(R);M=log10(M);
res=zeros(k1,k2);
t=cputime;
record=zeros(k1*k2,4);
for n1=1:k1
    for n2=1:k2
        f=zeros(24,1);
        for j=1:7
            f=f+m(n1)*r(n2).^(j-1) * BPs(:,j);
        end
        [c1,e1,JFDres]=JFD(f,20); ll=length(JFDres);
        res(n1,n2)=log10(JFDres(ll)/(m(n1)*r(n2)^9));
        record(n1*(k2-1)+n2,:)= [n1 n2 ll res(n1,n2)];
    end
end
end
t=cputime-t
surf(R,M,res)
xlabel('log10(r)'); ylabel('log10(m)'); colorbar;
view(0,90); camproj('orthographic')
save ex3b res record
```

example4a.m

```
%Example 4a

clear

a = 0.5*(sqrt(5)-1)* exp(i * pi * (0:5) /3 );
z = exp(i*(0:23)' * pi / 12);
zzeta = angle( (ones(24,1)* (abs(a)./a)).* ...
    (( (ones(24,1)*a) - (z*ones(1,6)) ) ...
    ./(1- (ones(24,1)*conj(a)) .* (z*ones(1,6)) )));
zzeta = [zzeta angle(z)];
nota = [a 0];nota = nota + 0.002-0.001i;
notzeta = angle( (ones(24,1)* (abs(nota)./nota)).* ...
    (( (ones(24,1)*nota) - (z*ones(1,7)) ) ...
    ./(1- (ones(24,1)*conj(nota)) .* (z*ones(1,7)) )));
BP2 = (notzeta(:,3)*4)+(notzeta(:,4)*3);
f = 10000*exp(i*BP2) ;
[c1,e1,JFDres]=JFD(f,20);
[c2,e2,IBAres]=IBA(f,50);
M1 = [conj(c1') e1'];
M2 = [conj(c2') e2'];
[M1 M2(1:size(M1,1),:)]
```

example4b.m

```
%Example 4b

clear

a = 0.5*(sqrt(5)-1)* exp(i * pi * (0:5) /3 );
z = exp(i*(0:23)' * pi / 12);
zzeta = angle( (ones(24,1)* (abs(a)./a)).* ...
    (( (ones(24,1)*a) - (z*ones(1,6)) ) ...
    ./(1- (ones(24,1)*conj(a)) .* (z*ones(1,6)) )));
zzeta = [zzeta angle(z)];
nota = [a 0];nota = nota + 0.002-0.001i;
notzeta = angle( (ones(24,1)* (abs(nota)./nota)).* ...
    (( (ones(24,1)*nota) - (z*ones(1,7)) ) ...
    ./(1- (ones(24,1)*conj(nota)) .* (z*ones(1,7)) )));
BP2 = (notzeta(:,3)*4)+(notzeta(:,4)*3);
BP3 = z+(notzeta(:,1));
f = 10000*exp(i*BP2) - 800*exp(i*BP3);
[c1,e1,JFDres]=JFD(f,20);
[c2,e2,IBARES]=IBA(f,50);
M1 = [conj(c1') e1'];
M2 = [conj(c2') e2'];
[M1 M2(1:size(M1,1),:)]
```