

11-18-2008

Orthogonal Filters and the Implications of Wrapping on Discrete Wavelet Transforms

Sarah K. Bleiler

University of South Florida

Follow this and additional works at: <https://scholarcommons.usf.edu/etd>

 Part of the [American Studies Commons](#)

Scholar Commons Citation

Bleiler, Sarah K., "Orthogonal Filters and the Implications of Wrapping on Discrete Wavelet Transforms" (2008). *Graduate Theses and Dissertations*.

<https://scholarcommons.usf.edu/etd/144>

This Thesis is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Orthogonal Filters and the Implications of Wrapping on Discrete Wavelet Transforms

by

Sarah K. Bleiler

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Arts
Department of Mathematics
College of Arts and Sciences
University of South Florida

Major Professor: Catherine Beneteau, Ph.D.
Thomas Bieske, Ph.D.
Dmitry Khavinson, Ph.D.

Date of Approval:
November 18, 2008

Keywords: filter, matrix transform, Fourier series, smoothness, zero moments

©Copyright 2008, Sarah K. Bleiler

Acknowledgements

I would like to thank my advisor and mentor, Dr. Catherine Beneteau, for all of her help and support throughout my time in the mathematics department at University of South Florida. Her enthusiasm, guidance, and advice were invaluable in the writing of this thesis and in my academic journey thus far.

I would also like to thank both Dr. Thomas Bieske and Dr. Dmitry Khavinson for all their advice and suggestions during the formulation of this thesis. Dr. Bieske was particularly wonderful in assisting me with the technical formatting of this paper, and Dr. Khavinson's review and suggestions were invaluable in keeping the subject matter focused and informative.

Special thanks to my mother and father, without whom I would never be in the great place I am today. Your love and support have been my greatest motivation.

Lastly, I want to thank my best friend in this world, Ray. Without you, I surely would never have made it to this point. I love you.

Table of Contents

| | |
|--------------------------------------------------------------------------|-----|
| List of Figures | i |
| Abstract | iii |
| 1 Introduction | 1 |
| 2 Orthogonal Filters | 14 |
| 2.1 Haar Filter | 14 |
| 2.2 Daubechies Filters | 15 |
| 2.3 Coiflet Filters | 19 |
| 3 Smoothness | 23 |
| 3.1 Smoothness of $ H(w) ^2$ | 23 |
| 3.2 Smoothness of $ H(w) $ | 25 |
| 4 Wrapping | 33 |
| 4.1 Non-wrapping Rows | 35 |
| 4.2 Wrapping Rows | 36 |
| 4.2.1 Non-wrapping Portion | 36 |
| 4.2.2 Wrapping Portion | 37 |
| 4.3 Effects of Wrapping on Zero Orthogonality Conditions | 39 |
| 4.4 A New Filter Construction | 43 |
| 5 Conclusion | 48 |
| 5.1 Future Research and Study | 48 |
| 5.2 Mathematica Program for Construction of Daubechies Filters | 49 |

List of Figures

| | | |
|-----|----------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | $H(w)$ (from Example 1.0.12) on $[0, \pi]$, h is a lowpass filter | 8 |
| 1.2 | $G(w)$ (from Example 1.0.12) on $[0, \pi]$, g is a highpass filter | 9 |
| 1.3 | Image before and after the application of the Haar wavelet transform | 11 |
| 3.1 | $H(w)$ on $[0, \pi]$ for Daubechies length 4 filter | 30 |
| 3.2 | $H(w)$ ² on $[0, \pi]$ for Daubechies length 4 filter | 30 |
| 3.3 | $H(w)$ and $H(w)$ ² on $[0, 4\pi]$ for Daubechies length 4 filter | 31 |
| 3.4 | $H(w)$ for Daubechies and Coiflet filters (length 6) | 31 |
| 3.5 | $H(w)$ for Daubechies and Coiflet filters (length 12) | 32 |
| 4.1 | Lowpass portion of a 14×14 DWT (the top half of the matrix) | 34 |
| 4.2 | Lowpass portion of a 10×10 DWT (the top half of the matrix) with $c > L$ | 37 |
| 4.3 | Length 12 modified filters | 45 |
| 4.4 | Length 18 modified filters | 46 |
| 4.5 | Modulus comparison of Daubechies length 12, Coiflet length 12, and modified length 12 ($K = 2, c = 3$) filters | 47 |
| 4.6 | Modulus comparison of Daubechies length 18, Coiflet length 18, and modified length 18 ($K = 3, c = 5$) filters | 47 |

Orthogonal Filters and the Implications of Wrapping on Discrete Wavelet Transforms

Sarah K. Bleiler

ABSTRACT

Discrete wavelet transforms have many applications, including those in image compression and edge detection. Transforms constructed using orthogonal filters are extremely useful in that they can easily be inverted as well as coded. We review the major properties of three well-known orthogonal filters, namely, the Haar, Daubechies, and Coiflet filters. Subsequently, we analyze the Fourier series that corresponds to each of those filters and recall some important results about the smoothness of the modulus of those Fourier series. We consider a specialized case in which the length of the discrete wavelet transform is not much longer than the length of the filter used in its construction. For this case, we prove the existence of additional degrees of freedom in the system of equations used in the construction of the aforementioned orthogonal filters. We suggest a modified Coiflet filter which takes advantage of the extra degrees of freedom by imposing further conditions on the derivative of the Fourier series.

1 INTRODUCTION

In mathematics it is often beneficial to transform a given problem into another setting, where it is easier to work with. For instance, it is sometimes more straightforward to work in polar coordinates than in Cartesian coordinates, in finite dimensional spaces instead of in infinite dimensional spaces, or in radian measures instead of in degrees. This idea of transforming objects to a better suited environment is an important theme that runs throughout all of the sciences.

Discrete wavelet transforms (DWTs) are matrices used to transform sets of discrete data into data that is easier to work with and manipulate. This transformation, or processing, of data is done through matrix multiplication. One example of such processing is the application of a DWT to an image. For instance, we can use DWTs to aid in the process of image compression. To do this, we would need to choose an appropriate DWT, one which is known in practice to be useful in image compression, and then apply it to the matrix containing the pixel intensities of the image we wish to compress. The resulting (transformed) image should then be better suited to the process of image compression. Another such image processing domain for which DWTs have proven useful is edge detection. We will explore these image processing applications in more detail at the end of this section (see Figure 1.3).

We take some time to discuss the history of wavelets and why they came about. Fourier analysis has been used for centuries as a tool for analyzing and transforming signals and images. Fourier series are built using sine and cosine functions, which are periodic waves that continue in both directions forever. Therefore, when processing data which is not itself periodic and/or time-independent, but rather is transient and contains abrupt changes, Fourier transforms are not very efficient. Wavelets were constructed as the tool needed to successfully work with such transient data. In 1946, Gabor examined Fourier series which were taken on a finite interval, such that both time and frequency could be considered [11].

This formed the basis of modern wavelet analysis [14]. The first “true” wavelets can be found in a 1983 paper written by Morlet entitled *Sampling Theory and Wave Propagation* (see [16]) and in a 1984 paper by Morlet and Grossman entitled *Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape* (see [12]). Most early work with wavelets was conducted in either France or America; in France, some of the seminal works on wavelets were [7] and [15], and in America, [18] and [17]. However, the major paper that influenced many of the applications of wavelets that we see today, *Orthonormal Bases of Compactly Supported Wavelets*, was written by Ingrid Daubechies in 1988 [8]. Part II of that paper [9], published in 1993, was equally as influential and forms the background for much of the information in this thesis.

We now outline the structure of this thesis. First note that the building blocks of DWTs are called filters (essentially the rows of the DWT), and they will be our main mathematical objects of discussion. We will begin, in Chapters 1 and 2, by introducing some terminology and the basics behind the construction of some of the most well-known filters, namely the Haar, Daubechies, and Coiflet filters. Then we will see how they can be used in applications such as image compression and edge detection. We will classify the construction of these filters in terms of their corresponding Fourier series and discuss why this is useful. In Chapter 3, we will consider the “smoothness” of the graph of the modulus of the Fourier series for some general filters. This will help us to better understand the implications of each filter construction as well as to approximate how close a filter is to being “ideal.” We notice that for any DWT composed of filters of length greater than two, there will always be what is called “wrapping” throughout the matrix. Our main question for this thesis is to find out what implications the process of wrapping has on the construction conditions that we highlight in Chapters 1 and 2. Therefore, in Chapter 4 our focus will be on exploring wrapping, identifying interesting outcomes for DWTs containing significant amounts of wrapping, and considering a new filter construction that results from those outcomes. This new filter produces a corresponding graph which is “smoother”, thus implying that the filter is closer to “ideal.”

We begin with the definition of a filter.

Definition 1.0.1 A **filter** is a bi-infinite sequence $(\dots, h_{-3}, h_{-2}, h_{-1}, h_0, h_1, h_2, \dots)$.

Definition 1.0.2 A **finite length filter** is a filter that has only a finite number of non-zero coefficients.

Filters are used to “process” data; this occurs by means of convolution.

Definition 1.0.3 Let h and x be two bi-infinite sequences. Then the **convolution product** y of h and x , denoted by $h * x$, is the bi-infinite sequence $y = h * x$, whose n th component is given by

$$y_n = \sum_{k=-\infty}^{\infty} h_k x_{n-k}.$$

Thus, if we think of x as the input data or signal, and h as the filter, then the “processed” data is represented by $h * x$. As a simple example, consider the filter

$h = (\dots, 0, \frac{1}{2}, \frac{1}{2}, 0, 0, \dots)$, where $h_k = \frac{1}{2}$ for $k = 0, 1$, and $h_k = 0$ otherwise. If x is any sequence, then the components of the convolution product can be written as

$y_n = \frac{1}{2}x_n + \frac{1}{2}x_{n-1}$. Therefore, the “processing” of data in this case can be thought of as a simple averaging of consecutive entries of the input data sequence x . Some filters are useful for identifying homogeneous (i.e., locally constant) portions of the input data, while others are useful for identifying large differences in the input data. Therefore, the filter one chooses depends greatly on the application, and the type of processing that is desired.

Throughout this thesis, we will only be concerned with finite length filters. Such a filter can be written as $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$, where $h_l \neq 0$, $h_L \neq 0$. This means that each component of the convolution product will be a convergent series, represented by

$$y_n = \sum_{k=l}^L h_k x_{n-k}.$$

In this thesis we will also assume that the coefficients of the filters are all real-valued, unless otherwise noted.

In order to gain a better understanding of how different filters will process data, it is often useful to consider their corresponding Fourier series.

Definition 1.0.4 Given a filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$, its **corresponding Fourier series** is given by $H(w) = \sum_{k=l}^L h_k e^{ikw}$, $w \in \mathbb{R}$.

It is beneficial to consider a filter in the “Fourier domain” for several different reasons, such as the following:

- We can characterize filters by the properties of their corresponding Fourier series.

- It is easier to generalize filter constructions by using their Fourier series.
- The modulus $|H(w)|$ can be used as a good predictor of the filter's properties in certain applications.
- There are many well-known results and powerful tools from the field of Fourier Analysis that are at our disposal.

Let us investigate the importance of the modulus $|H(w)|$ that is associated with a filter h . How can $|H(w)|$ be used to predict a filter's properties?

Proposition 1.0.5 *Given filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ (with real-valued entries) the corresponding function $|H(w)|$ is even.*

Proof.

$$H(-w) = \sum_{k=l}^L h_k e^{ik(-w)}.$$

$$\overline{H(w)} = \sum_{k=l}^L h_k e^{-ikw} \quad (\text{since } h_k \text{ is real}).$$

So $H(-w) = \overline{H(w)}$. Therefore $|H(-w)| = |\overline{H(w)}| = |H(w)|$ and $|H(w)|$ is an even function. ■

By Euler's formula, we know $e^{ikw} = \cos kw + i \sin kw$. Because $H(w) = \sum_{k=l}^L h_k e^{ikw} = \sum_{k=l}^L h_k (\cos kw + i \sin kw)$, it is easy to see that $H(w)$ can be written as a polynomial in terms of $\cos kw$ and $\sin kw$, and hence $|H(w)|$ is 2π -periodic. Given the fact that $|H(w)|$ is both even and 2π -periodic, we can gather all important information of the graph by considering only the portion defined on $[0, \pi]$.

By examining $|H(w)|$ on that interval, we can get a good idea of the nature of the processing that the corresponding filter h will perform. For instance, if the value of $|H(w)|$ is large at $w = 0$, then when we process the data using filter h , if there are portions of the data that are largely homogeneous, the filter will preserve those portions of the data. As an example, consider the filter $h = (h_0, h_1, h_2) = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. The corresponding Fourier series is $H(w) = \frac{1}{4} + \frac{1}{2}e^{iw} + \frac{1}{4}e^{2iw}$, and thus $|H(0)| = 1$. If we use this filter to process the signal $x = (\dots, 1, 1, 1, \dots)$, then $h * x = (\dots, 1, 1, 1, \dots)$, thus preserving the homogeneity of the signal.

In addition, if the value of $|H(w)|$ is small at $w = \pi$, then when we apply the filter h to our data, if there are portions of the data that have many differences (i.e., they are highly oscillatory), the filter will annihilate (i.e., make less significant) those portions of the data. As an example of this, we use the same filter $h = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ as above, noting that $|H(\pi)| = 0$. If we use this filter to process the signal $x = (\dots, -1, 1, -1, 1, \dots)$, then $h * x = (\dots, 0, 0, 0, \dots)$, thus annihilating the oscillation of the signal.

We characterize this type of filter (i.e., one which preserves homogeneity while simultaneously annihilating large amounts of oscillation) as a *lowpass* filter. Example 1.0.12 will help to clarify this notion even further. We give the precise definitions for lowpass (and highpass) filters below (see [20], p. 143, 147):

Definition 1.0.6 Let h be a filter. Let $0 < w_p \leq w_s < \pi$ and suppose that there exists $0 < \delta < 1/2$, with $\sqrt{2} - \delta \leq |H(w)| \leq \sqrt{2} + \delta$ for $0 \leq w \leq w_p$ and a $0 < \lambda < 1/2$, so that for $w_s \leq w \leq \pi$, $|H(w)| \leq \lambda$. Then we call h a **lowpass filter**.

For example, if h is a filter such that $|H(w)| = \sqrt{2}$ for $w \leq a$ and $|H(w)| = 0$ for $w > a$, for some $a \in (0, \pi)$, then h is a lowpass filter. In fact, this h is called the *ideal* lowpass filter.

Remark 1.0.7 In Chapter 3 we will discuss the smoothness of $|H(w)|$ and show which filters are better at approximating the behavior of the ideal lowpass filter for values of w close to 0 and close to π .

The second type of filter that we want to define is called a *highpass* filter. It acts in exactly the opposite way in that it preserves large oscillation and annihilates homogeneity. Therefore, if we have a highpass filter g , the modulus of its corresponding Fourier series would be large at $w = \pi$ and small at $w = 0$.

Definition 1.0.8 Let g be a filter. Let $0 < w_p \leq w_s < \pi$ and suppose that there exists $0 < \lambda < 1/2$ so that $|G(w)| \leq \lambda$ for $0 \leq w \leq w_p$ and a $0 < \delta < 1/2$ with $\sqrt{2} - \delta \leq |G(w)| \leq \sqrt{2} + \delta$ for $w_s \leq w \leq \pi$. Then we call g a **highpass filter**.

The definition of the *ideal* highpass filter is analogous to the definition given above.

In general, we consider a simplification of the definitions of lowpass and highpass filters given above by requiring only the following conditions:

1. If h is a lowpass filter then we must have $|H(0)| = \sqrt{2}$ and $|H(\pi)| = 0$.
2. If g is a highpass filter then we must have $|G(0)| = 0$ and $|G(\pi)| = \sqrt{2}$.

The value $\sqrt{2}$ may seem a bit arbitrary but as we will see later, it is important to maintain orthogonality.

Remark 1.0.9 *All of the filters introduced in Chapter 2 will in fact satisfy the more stringent requirements of Definition 1.0.6 and Definition 1.0.8.*

We have now defined the two types of filters (lowpass and highpass) that we will use in order to process our data. We will use these filters as the building blocks of the discrete wavelet transform.

Definition 1.0.10 For a lowpass filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ and a highpass filter $g = (g_l, g_{l+1}, \dots, g_{L-1}, g_L)$, the corresponding **length** N (N even) **discrete wavelet transform (DWT)** is denoted by W_N and is written as an $N \times N$ matrix as follows:

$$\begin{pmatrix} h_L & h_{L-1} & \cdots & \cdots & \cdots & \cdots & h_{l+1} & h_l & 0 & 0 & 0 & 0 \\ 0 & 0 & h_L & h_{L-1} & \cdots & \cdots & \cdots & \cdots & h_{l+1} & h_l & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \ddots \\ h_{l+1} & h_l & 0 & 0 & 0 & 0 & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & h_{l+1} & h_l & 0 & 0 & 0 & 0 & \ddots & \ddots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & h_{l+1} & h_l & 0 & 0 & 0 & 0 & h_L & h_{L-1} \\ g_L & g_{L-1} & \cdots & \cdots & \cdots & \cdots & g_{l+1} & g_l & 0 & 0 & 0 & 0 \\ 0 & 0 & g_L & g_{L-1} & \cdots & \cdots & \cdots & \cdots & g_{l+1} & g_l & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots & \ddots & \ddots \\ g_{l+1} & g_l & 0 & 0 & 0 & 0 & \ddots & \ddots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & g_{l+1} & g_l & 0 & 0 & 0 & 0 & \ddots & \ddots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & g_{l+1} & g_l & 0 & 0 & 0 & 0 & g_L & g_{L-1} \end{pmatrix}$$

Note 1.0.11 *We wish to make two observations.*

- *For obvious reasons we consider the top half of the matrix as the lowpass portion of the DWT and the bottom half of the matrix as the highpass portion.*

- *Different filters lead to different DWTs. However, once the lowpass and highpass filters h and g have been selected, the corresponding (length N) DWT is automatically defined, and we call it W_N .*

A DWT is always split in half into highpass and lowpass portions. We assume throughout the paper that the dimensions of the matrix are even so that this construct is possible. In a DWT, the first row contains the lowpass filter h followed by an appropriate number of zeros which fill in the remaining places in the row. The remaining rows in the lowpass portion are simply shifts of row 1, each by two places to the right. (We call these *2-translates* of row 1.) The top half of the matrix contains the lowpass row h and its 2-translates, while the bottom half of the matrix contains the highpass row g and its 2-translates.

A DWT is used to process data in much the same way as was described for a single filter. Looking closer, if x is a vector that represents, say, a signal, then in multiplying x by W_N , we are essentially computing every other component of the convolution product. For more information on the rationale behind the construction of a DWT and its relation to convolution, see [20].

We work through an example below to show a specific case of a DWT.

Example 1.0.12 Suppose $h = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $g = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. We can easily check that h is a lowpass filter (using the simplified definition):

$$H(w) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}e^{iw} = \frac{\sqrt{2}}{2}(1 + e^{iw}) = \frac{\sqrt{2}}{2}(1 + \cos w + i \sin w).$$

Therefore, $H(0) = \frac{\sqrt{2}}{2}(1 + 1) = \sqrt{2}$ and $H(\pi) = \frac{\sqrt{2}}{2}(1 + (-1)) = 0$. Similarly, g is a highpass filter. The graphs of $|H(w)|$ and $|G(w)|$ are given in Figures 1.1 and 1.2.

Now suppose we want to process the vector $v = (1, 1, 1, 1, -1, 1, -1, 1)$ using a DWT constructed with lowpass filter $h = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and highpass filter $g = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. The vector v is of length 8 so the DWT we will use to process the data in v should be a matrix of

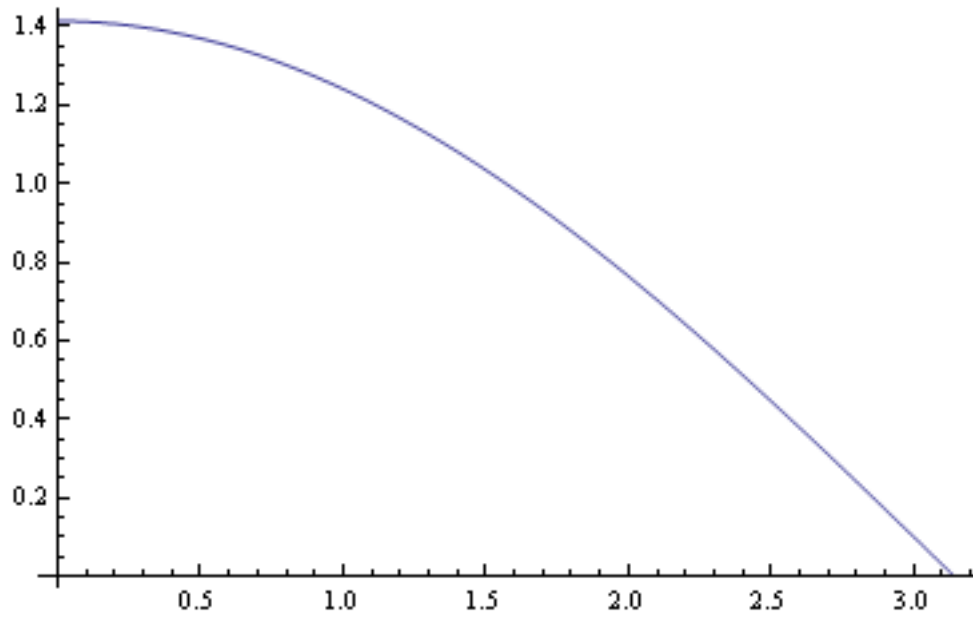


Figure 1.1: $|H(w)|$ (from Example 1.0.12) on $[0, \pi]$, h is a lowpass filter

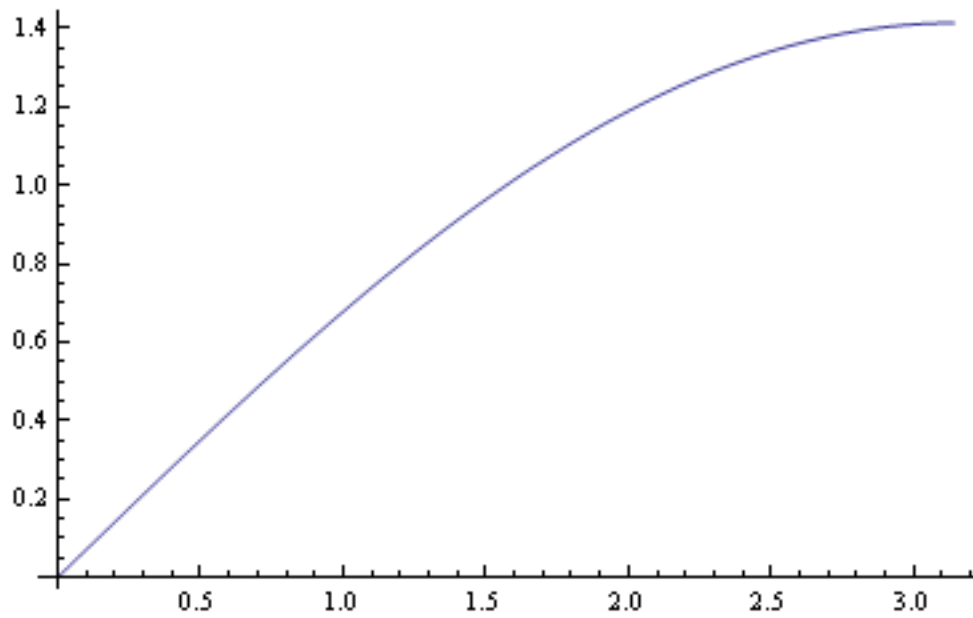


Figure 1.2: $|G(w)|$ (from Example 1.0.12) on $[0, \pi]$, g is a highpass filter

dimension 8×8 . We write the matrix as follows (see Definition 1.0.10):

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

Now let us consider what happens when we use the DWT to process our data in vector $v = (1, 1, 1, 1, -1, 1, -1, 1)$. To do this, we use matrix multiplication as follows:

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} \sqrt{2} \\ \sqrt{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{2} \\ \sqrt{2} \end{pmatrix}$$

Notice that the first half of v is very homogeneous, while the second half varies between 1 and -1 . We interpret the above matrix multiplication as follows:

- Row 1 (a lowpass row) has zero entries everywhere except the first and second entries. Hence, it is used to compare the entries v_1 and v_2 of vector v . Because these entries of v are the same, it should preserve that homogeneity. We see that $\text{row } 1 \cdot v = \sqrt{2}$ and therefore preserves homogeneity (by a seemingly arbitrary factor of $\sqrt{2}$).
- Row 2 acts in the same manner as row 1, comparing v_3 and v_4 .
- Now consider row 3 which has nonzero entries in the 5th and 6th column, and hence is used to compare the entries $v_5 = -1$ and $v_6 = 1$. Because these entries of v are

different and since row 3 is lowpass, it should annihilate their differences. We see this is the case, since $\text{row } 3 \cdot v = 0$.

- Row 4 acts in the same manner as row 3, comparing v_7 and v_8 .

We could easily check that the highpass portion of the matrix acts in exactly the opposite manner, preserving differences and annihilating similarities.

Remark 1.0.13 *In the above example, the lengths of the filters h and g are two. However, for DWTs built with filters h and g of lengths greater than two, some rows of the DWT will wrap around to the front of the matrix (as shown in Definition 1.0.10). We will talk more about the implications of this “wrapping” of rows in Chapter 4.*

Other than processing a one dimensional signal, we can also use a DWT to process a two dimensional array of data, such as an image. The image itself is represented by a matrix, say M , where the entries of M represent the pixel intensities of the image. In order to process a matrix M , we need to process both the rows and columns of the matrix. To perform this 2-dimensional transform, we compute $W_N M W_N^T$. Figure 1.3 shows the result when we process an image using the DWT formed by lowpass vector $h = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and highpass vector $g = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$. We will learn in the next chapter that the name of this particular DWT is the *Haar wavelet transform*.

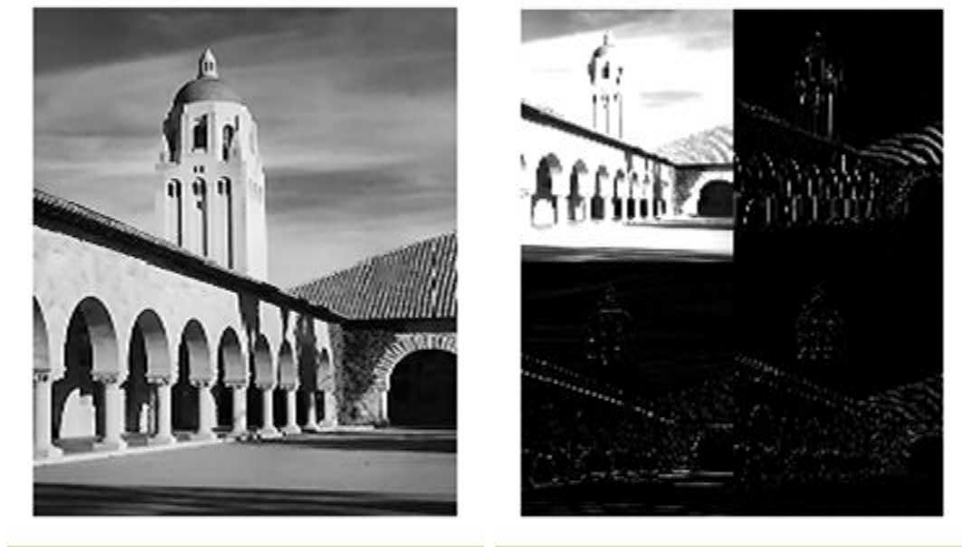


Figure 1.3: Image before and after the application of the Haar wavelet transform

Consider W_N as two submatrices such that the top half is the $\frac{N}{2} \times N$ lowpass submatrix H , and the bottom half is the $\frac{N}{2} \times N$ highpass submatrix G . Then, the top left corner of the processed image is the result of processing both the rows and columns of the image with the lowpass submatrix H (i.e., HMH^T). Therefore, the top left corner represents an approximation of the original image. Alternatively, the top right corner of the processed image is the result of processing the columns of the image with the lowpass submatrix (i.e., HM) and then processing that result by the transpose of the highpass submatrix (i.e., HMG^T). Therefore, the top right corner represents the *vertical* differences between the original image M and the approximation image HMH^T . Similarly, the bottom left corner (GMH^T) represents the *horizontal* differences between the original and the approximation images, while the bottom right corner (GMG^T) represents the *diagonal* differences between the original and the approximation images.

We see immediately how this process can aid in both edge detection and image compression. As stated above, after we perform the transform, an approximation of the original image appears in the top left corner while the other three blocks represent the “differences” among pixel intensities in that image. If there is a large difference in value between two consecutive pixels in the original image, then the resulting value in the “difference” blocks will also be large. The pixel intensities for a typical 8-bit grayscale image range from 0 to 255 where 0 is black and 255 is white. Therefore the edges of an image are easily detected by looking for the largest values (i.e., the whitest pixels) in the resulting three “difference” blocks.

Also, note that the resulting transformed matrix has significantly less detail than the original image. In fact, many of the pixel values have been converted to 0 (black) or close to 0. This is likely to occur with most natural images because they usually have large homogeneous regions, and therefore the differences in those areas will be minimal. Performing what is called lossy compression, we can convert to 0 all of those pixel intensities which are close to 0. This way, we significantly reduce the amount of storage space needed in order to store the image or send it as a file over the internet. We call this lossy compression because when we restore the image (by performing an inverse transform), some of the original information will be lost due to the slight conversion of those pixel intensities.

Next, we specify the particular type of DWT that we will consider in this thesis, namely the *orthogonal* DWT.

Definition 1.0.14 A square matrix M is called **orthogonal** if $M^{-1} = M^T$.

Definition 1.0.15 A set of vectors $\{u_i\}$ is called **orthonormal** if

$$u_i \cdot u_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

In other words, the rows of an orthogonal matrix M must form an orthonormal set.

The significant benefit of dealing with orthogonal DWTs is that we can easily invert the transform process since the inverse of W_N is simply its transpose. In addition, instead of computing the entire matrix multiplication, we can take advantage of the sparseness of W_N and write an algorithm which efficiently computes the DWT. Having orthogonality in our DWTs makes it easy to write a nearly identical algorithm for the inverse transform process (see [20] for an example).

In the next chapter, we show how to construct some specific finite length filters which come together to form an orthogonal DWT.

2 ORTHOGONAL FILTERS

In this chapter, we discuss some of the most well known orthogonal filters, namely the Haar, Daubechies, and Coiflet filters. Like many mathematical developments, these filters were originally built to suit the needs of a particular application.

2.1 Haar Filter

Although the official study of wavelets had not yet begun, in 1910 Haar introduced the filters that are used in what we now know as the Haar wavelet transform [13]. He originally proposed the system as an example of a set of coefficients for a pair of functions that form a compactly supported orthonormal basis for $L^2(\mathbb{R})$. In fact, the length two filters $h = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $g = (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$ which we discussed in Chapter 1 are actually those coefficients. These two filters form the building blocks for the *Haar wavelet transform*. Recall that h is a lowpass filter and g is a highpass filter. Therefore, we can place these filters (and their 2-translates) in their respective positions of the lowpass and highpass portions of a DWT. The (*length* N) *Haar wavelet transform* W_N is written as follows:

$$\begin{array}{c} \frac{N}{2} \\ \text{rows} \end{array} \left(\begin{array}{cccccccc} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{N}{2} \\ \text{rows} \end{array} \right) \begin{array}{c} 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{array}$$

We can interpret the lowpass filter as an “averaging” filter which is scaled by a factor of $\sqrt{2}$. As this filter is used to process data, it results in giving an approximation of the original data. On the other hand, the highpass filter computes differences, and when applied to data, gives a weighted average of the difference between consecutive values in the data. As was discussed in Chapter 1, another way to think of this, in terms of image processing, is that the highpass filter computation gives information detailing how different the approximation image is from the original image. Putting these concepts of averaging and differencing together gives us a transform which is completely invertible.

Next we check for orthogonality of the Haar wavelet transform. Remember, W_N is orthogonal if and only if $W_N W_N^T = I_N$. In other words, the dot product of row i with row j must be 1 if $i = j$ and 0 if $i \neq j$. It is enough to consider the dot product of only row 1 with all other rows of W_N . Given the structure of h , g , and W_N it is easy to see that all other dot products will give the same set of results.

- row 1 \cdot row 1 = $\frac{\sqrt{2}}{2} \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \frac{\sqrt{2}}{2} = \frac{1}{2} + \frac{1}{2} = 1$.
- row 1 \cdot row $k = 0$ ($k = 2, \dots, \frac{N}{2}$) since all other rows in the lowpass portion are 2-translates of row 1.
- row 1 \cdot row $(\frac{N}{2} + 1) = \frac{\sqrt{2}}{2} \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} (-\frac{\sqrt{2}}{2}) = 0$.
- row 1 \cdot row $j = 0$ ($j = \frac{N}{2} + 2, \dots, N$) since all other rows in the highpass portion are 2-translates of row 1.

Therefore, we have shown that the Haar wavelet transform is orthogonal.

2.2 Daubechies Filters

The Haar filter is useful in some applications, particularly because it is so short and thus decreases computation time. However, the shortness of the filter can also hinder its effectiveness in some situations. For instance, since the filter is only of length 2, it may not process enough information at a time to catch all the detail that we desire for a particular edge detection or image processing application. A Daubechies filter can be constructed for any arbitrary even length, hence overcoming the limited applications of the Haar filter.

Suppose we want to construct filters h and g of arbitrary even length such that h is lowpass, g is highpass, and the resulting DWT is orthogonal. In this section we assume

that our filter h is of the form $h = (h_0, h_1, \dots, h_L)$ where L is an odd positive integer. In order for a DWT W_N to be orthogonal, we need $W_N W_N^T = I_N$. Each row in the lowpass (correspondingly, highpass) portion of W_N is a 2-translate of the first row of the lowpass (highpass) portion. Therefore, if we have the conditions $\sum_{k=0}^L h_k^2 = 1$ and $\sum_{k=2m}^L h_k h_{k-2m} = 0$ for $m = 1, 2, \dots, \frac{L-1}{2}$ (correspondingly, $\sum_{k=0}^L g_k^2 = 1$ and $\sum_{k=2m}^L g_k g_{k-2m} = 0$ for $m = 1, 2, \dots, \frac{L-1}{2}$), then the set of lowpass (highpass) rows form an orthonormal set.

Remark 2.2.1 *For now, we take the previous statement to be true, and reference [20]. In Chapter 4, we will prove the statement in general by focusing on a more specific case (i.e., when “wrapping” occurs among the elements in the rows of a DWT).*

The following proposition gives an equivalent condition in terms of Fourier series (see [20], p. 286):

Proposition 2.2.2 *Suppose $H(w)$ is the Fourier series corresponding to the real-valued filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$, then*

$$|H(w)|^2 + |H(w + \pi)|^2 = 2$$

if and only if

$$\sum_{k=l}^L h_k^2 = 1$$

and

$$\sum_{k=l+2m}^L h_k h_{k-2m} = 0$$

for $m = 1, 2, \dots, \frac{L-l-1}{2}$.

Note 2.2.3 *For our case in this section, we set $l = 0$.*

Thus, we can write all the conditions that are needed in order to form a lowpass filter which satisfies orthonormality in the lowpass portion of a DWT solely in terms of its corresponding Fourier series:

- $|H(0)| = \sqrt{2}$ (lowpass).
- $|H(\pi)| = 0$ (lowpass).

- $|H(w)|^2 + |H(w + \pi)|^2 = 2$ (orthonormality).

If we solve this system for a filter with $L = 1$ (i.e., $h = (h_0, h_1)$), then one of our solutions will be the Haar lowpass filter. If we solve the system above for any greater odd value of L , then we obtain an infinite number of solutions [8]. Therefore, we need to add more conditions to our system of equations. The equations we will add can be considered “smoothness” conditions. They come from taking derivatives of the Fourier series at $w = \pi$ and hence flatten (or smooth) the graph of $|H(w)|$ at that point. If $L = 3$, we only need to add one of these derivative conditions to the system, that is, we require $H'(\pi) = 0$. For $L = 5$, we need two derivative conditions, namely $H'(\pi) = 0$ and $H''(\pi) = 0$. In general, for filter $h = (h_0, h_1, \dots, h_L)$ of length $L + 1$, we need the derivative conditions $H^{(m)}(\pi) = 0$ for $m = 1, 2, \dots, \frac{L-1}{2}$, [8]. In fact, Daubechies shows that this is the maximal number of derivatives that can be taken.

Remark 2.2.4 *We will see in Chapter 3 how these derivative conditions at π play an important role not only in the smoothness of $|H(w)|$ at $w = \pi$, but also at $w = 0$.*

What we have described above is essentially the characterization of the lowpass filters that we call *Daubechies filters*. That is, Daubechies filters are lowpass, even length filters of the form $h = (h_0, h_1, \dots, h_L)$ which satisfy the orthonormality conditions given in Proposition 2.2.2 as well as the $\frac{L-1}{2}$ derivative conditions described above. Therefore, the system of equations we need to solve in order to get a finite number of solutions for possible Daubechies filters is the following:

- $|H(0)| = \sqrt{2}$ (lowpass).
- $|H(\pi)| = 0$ (lowpass).
- $|H(w)|^2 + |H(w + \pi)|^2 = 2$ (orthonormality).
- $H^{(m)}(\pi) = 0$ for $m = 1, 2, \dots, \frac{L-1}{2}$ (derivative conditions).

Once we obtain a finite set of solutions to this system, we have to narrow it down to one particular solution that we will call *the* Daubechies filter. Daubechies [8] explains her rationale for picking *the* Daubechies filter from this finite set of solutions. For the interested reader, we have included a *Mathematica* [22] program which outlines the process for finding

the appropriate filter (see Section 5.3). Given any odd positive integer L , the program will return the Daubechies filter of length $L + 1$.

So far in this section we have described how to construct the Daubechies lowpass filter. Once we construct this filter, the only remaining problem is to construct the corresponding highpass filter $g = (g_0, g_1, \dots, g_L)$ that completes the orthogonal DWT. It is easy to check that if we let $g_k = (-1)^k h_{L-k}$, then g is highpass (i.e., satisfies $G(0) = 0, |G(\pi)| = \sqrt{2}$), and the following conditions hold: $\sum_{k=2m}^L h_k g_{k-2m} = 0$ for $m = 0, 1, \dots, \frac{L-1}{2}$. These conditions give orthonormality between rows in the highpass and the lowpass portions of a DWT. The following proposition gives an equivalent condition for orthonormality (between lowpass and highpass portions) in terms of the Fourier series $H(w)$ and $G(w)$ (see [20], p. 292):

Proposition 2.2.5 *Suppose $H(w)$ is the Fourier series corresponding to the filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ and $G(w)$ is the Fourier series corresponding to the filter $g = (g_l, g_{l+1}, \dots, g_{L-1}, g_L)$, then*

$$H(w)\overline{G(w)} + H(w + \pi)\overline{G(w + \pi)} = 0$$

if and only if

$$\sum_{k=l+2m}^L h_k g_{k-2m} = 0$$

for $m = 0, 1, \dots, \frac{L-l-1}{2}$.

Note 2.2.6 *For our case in this section, we set $l = 0$.*

It is easy to check that for the filter g satisfying $g_k = (-1)^k h_{L-k}$, its corresponding Fourier series can be written as $G(w) = -e^{iLw}\overline{H(w + \pi)}$. The following proposition gives (1) an equivalent condition for orthonormality (among the highpass rows) in terms of the Fourier series $G(w)$, and (2) an equivalent condition for orthonormality (between lowpass and highpass rows) in terms of the Fourier series $H(w)$ and $G(w)$:

Proposition 2.2.7 *Given a lowpass Daubechies filter h and corresponding Fourier series $H(w)$, if $G(w) = -e^{iLw}\overline{H(w + \pi)}$ then*

1. $|G(w)|^2 + |G(w + \pi)|^2 = 2$.
2. $H(w)\overline{G(w)} + H(w + \pi)\overline{G(w + \pi)} = 0$.

Proof.

1.

$$\begin{aligned}
|G(w)|^2 + |G(w + \pi)|^2 &= |-e^{iLw}|^2 |\overline{H(w + \pi)}|^2 + |-e^{iL(w+\pi)}|^2 |\overline{H(w)}|^2 \\
&= |H(w + \pi)|^2 + |H(w)|^2 \\
&= 2.
\end{aligned}$$

2.

$$\begin{aligned}
H(w)\overline{G(w)} + H(w + \pi)\overline{G(w + \pi)} &= H(w)\overline{-e^{iLw}\overline{H(w + \pi)}} \\
&\quad + H(w + \pi)\overline{-e^{iL(w+\pi)}\overline{H(w)}} \\
&= H(w)(-e^{-iLw})H(w + \pi) \\
&\quad + H(w + \pi)(-e^{-iL(w+\pi)})H(w) \\
&= H(w)H(w + \pi)(-e^{-iLw} - e^{-iL(w+\pi)}) \\
&= H(w)H(w + \pi)(-e^{-iLw})(1 + (-1)^L) \\
&= 0.
\end{aligned}$$

The last equality holds because L is odd.

■

2.3 Coiflet Filters

In this section, we will consider what are called Coiflet filters. Daubechies designed these filters in response to a suggestion by Coifman [9]. He needed this particular construction to use for applications in numerical analysis [1]. The indices of these filters differ from Daubechies filters in that they have negative as well as positive values. Therefore, we will write the filter as $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$, where l is some negative integer. We still want to ensure the lowpass (highpass) nature of the filters along with their orthonormality. However, we will not require the maximal number of derivative conditions on $H(w)$ at $w = \pi$ as was the case with Daubechies filters. We will see exactly how the derivative conditions differ after a few preliminary results.

Proposition 2.3.1 *Suppose we have a finite length filter $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ with $h_l \neq 0$ and $h_L \neq 0$ and corresponding Fourier series satisfying $|H(w)|^2 + |H(w + \pi)|^2 = 2$. Then l and L have different parity.*

Proof. By Proposition 2.2.2, we know that $|H(w)|^2 + |H(w + \pi)|^2 = 2$ implies that the lowpass rows (of any DWT constructed using h) form an orthonormal set. Suppose that l and L have the same parity. If l and L are both odd, then h has $L - l + 1$ (i.e., an odd number of) entries. The same is true if l and L are both even.

For the orthonormality of the lowpass rows to hold for all N (i.e., for all transform matrices W_N), every row translated by two entries and dotted with row 1 must give zero. However, since there are an odd number of entries in h , when we translate for the $(\frac{L-l}{2})$ th time, we get $h_L h_l = 0$. This implies that $h_L = 0$ or $h_l = 0$, which is a contradiction. ■

We will see later that Coiflet filters can be defined by the Fourier series

$$H(w) = \sqrt{2} \cos^{2K}\left(\frac{w}{2}\right) \left(\sum_{j=0}^{K-1} \binom{K-1+J}{j} \sin^{2j}\left(\frac{w}{2}\right) + \sin^{2K}\left(\frac{w}{2}\right) \sum_{l=0}^{2K-1} a_l e^{ilw} \right)$$

for $K \in \mathbb{Z}^+$.

Here we present a formula for the resulting starting and stopping indices of the filter corresponding to such a Fourier series (see [9] or [20], p. 307):

Proposition 2.3.2 *Suppose $h = (h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ is a finite length filter with corresponding Fourier series*

$$H(w) = \sqrt{2} \cos^{2K}\left(\frac{w}{2}\right) \left(\sum_{j=0}^{K-1} \binom{K-1+J}{j} \sin^{2j}\left(\frac{w}{2}\right) + \sin^{2K}\left(\frac{w}{2}\right) \sum_{l=0}^{2K-1} a_l e^{ilw} \right),$$

then $l = -2K$ and $L = 4K - 1$ so that the length of h is $6K$.

The main difference between the construction of Daubechies and Coiflet filters (other than the range of the indices) is the number of derivative conditions imposed on the Fourier series $H(w)$ at $w = \pi$ compared to the number of derivative conditions imposed on the Fourier series at $w = 0$. Recall that for Daubechies filters, the maximal number of derivative conditions are required at $w = \pi$ (i.e., $H^{(m)}(\pi) = 0$ for $m = 1, 2, \dots, \frac{L-1}{2}$)

while requiring no derivative conditions at $w = 0$ (i.e., no required value of m such that $H^{(m)}(0) = 0$). With Coiflet filters, however, an equal number of derivative conditions are required at both $w = \pi$ and $w = 0$. The length $6K$ Coiflet filter is characterized by the Fourier series

$$H(w) = \sqrt{2} \cos^{2K}\left(\frac{w}{2}\right) \left(\sum_{j=0}^{K-1} \binom{K-1+J}{j} \sin^{2j}\left(\frac{w}{2}\right) + \sin^{2K}\left(\frac{w}{2}\right) \sum_{l=0}^{2K-1} a_l e^{ilw} \right).$$

In [9], Daubechies shows that if $H(w)$ is defined as above, then

- $H^{(m)}(\pi) = 0$ for $m = 1, 2, \dots, 2K - 1$.
- $H^{(m)}(0) = 0$ for $m = 1, 2, \dots, 2K - 1$.

See [3], Section 6.9, for a discussion of some of the benefits and applications of using a filter that has a balance of derivative conditions between $w = 0$ and $w = \pi$.

Remark 2.3.3 *We will discuss the implications of these derivative conditions on the modulus $|H(w)|$ in greater depth in Chapter 3.*

Daubechies also shows in [9] that if

$$H(w) = \sqrt{2} \cos^{2K}\left(\frac{w}{2}\right) \left(\sum_{j=0}^{K-1} \binom{K-1+J}{j} \sin^{2j}\left(\frac{w}{2}\right) + \sin^{2K}\left(\frac{w}{2}\right) \sum_{l=0}^{2K-1} a_l e^{ilw} \right),$$

then $H(0) = \sqrt{2}$ and $|H(w)|^2 + |H(w + \pi)|^2 = 2$.

So, in review, the characteristics which define a Coiflet filter are as follows:

1. $|H(0)| = \sqrt{2}$ (lowpass).
2. $|H(\pi)| = 0$ (lowpass).
3. $|H(w)|^2 + |H(w + \pi)|^2 = 2$ (orthonormality).
4. $H^{(m)}(\pi) = 0$ for $m = 1, 2, \dots, 2K - 1$ (derivative conditions at π).
5. $H^{(m)}(0) = 0$ for $m = 1, 2, \dots, 2K - 1$ (derivative conditions at 0).

As we have seen, the Fourier series corresponding to the Coiflet filter is a long and complicated one. We can simplify the process of solving for Coiflet filters by using the starting

and stopping indices from Proposition 2.3.2 and then solving the system of equations given in the list above. Just as with the system of equations used to solve for Daubechies filters, this system gives a finite set of possible solutions. We can consult [9] to see the exact process for choosing *the* Coiflet filter, but we will not concern ourselves with that process in this thesis.

Note 2.3.4 *In a similar manner as with Daubechies filters, we can show that if the high-pass filter g is built from the Coiflet filter h by $G(w) = -e^{i(2K+1)w}\overline{H(w+\pi)}$, then the following properties hold:*

1. $|G(w)|^2 + |G(w + \pi)|^2 = 2.$
2. $H(w)\overline{G(w)} + H(w + \pi)\overline{G(w + \pi)} = 0.$

3 SMOOTHNESS

In Chapter 2, we introduced several different filter constructions, each one differing mainly in the number and type of derivative conditions imposed on the filter. In this chapter, we will consider those different derivative conditions in a more general setting, in hopes of gaining a better understanding of their direct implications on the properties of a filter. More specifically, we will compare the graphs of the modulus of the Fourier series corresponding to different filter constructions, and consider how the derivative conditions affect the smoothness of those graphs at $w = 0$ and $w = \pi$.

We are interested in the smoothness of $|H(w)|$ because it gives us a good idea of how close our filter is in comparison to the *ideal* filter (see Chapter 1). That is, the smoother the graph of $|H(w)|$ near $w = \pi$ and $w = 0$, the closer its representative filter is to being ideal. This is important for image processing applications such as edge detection. For instance, a highpass filter which is close to ideal will do a good job of preserving oscillatory (i.e., different) and annihilating homogeneous (i.e., similar) portions of the data. Therefore, the edges (largest differences) will be clearly visible in the transformed image.

In [9], Daubechies considers the smoothness results introduced below. We note that the results as written here are our own proofs.

Note 3.0.5 *In this chapter, we do not assume that we have filters with real coefficients, but rather we specify when they are necessary for a particular result.*

We begin our analysis of the smoothness of $|H(w)|$ by first considering the smoothness of $|H(w)|^2$.

3.1 Smoothness of $|H(w)|^2$

Lemma 3.1.1 *Given a finite length filter satisfying $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$, then*

$$\left(\frac{d^p}{dw^p} |H(w)|^2 \right) \Big|_{w=\pi} = 0 \text{ for } p = 0, 1, \dots, 2m + 1.$$

Proof. We will show that $(\frac{d^{2m+1}}{dw^{2m+1}}|H(w)|^2)|_{w=\pi} = 0$. (The smaller derivative conditions follow similarly.)

$$|H(w)|^2 = H(w)\overline{H(w)}.$$

By the Leibniz product rule, $(\frac{d^{2m+1}}{dw^{2m+1}}H(w)\overline{H(w)})|_{w=\pi}$

$$= \left(\sum_{k=0}^{2m+1} \binom{2m+1}{k} H^{(k)}(w)\overline{H^{(2m+1-k)}(w)} \right) \Big|_{w=\pi}. \quad (3.1.1)$$

For $k = 0, \dots, m$, our assumption conditions $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$ imply that the corresponding terms in the summation (3.1.1) are zero. We now only need to consider the terms of the summation which correspond to $k = m+1, \dots, 2m+1$, namely,

$$\binom{2m+1}{m+1} H^{(m+1)}(w)\overline{H^{(2m+1-(m+1))}(w)}, \dots, \binom{2m+1}{2m+1} H^{(2m+1)}(w)\overline{H^{(2m+1-(2m+1))}(w)}.$$

At $w = \pi$, these are $\binom{2m+1}{m+1} H^{(m+1)}(\pi)\overline{H^{(m)}(\pi)}, \dots, \binom{2m+1}{2m+1} H^{(2m+1)}(\pi)\overline{H(\pi)}$ which are all equal to zero (again by the assumptions $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$).

Therefore, $(\frac{d^{2m+1}}{dw^{2m+1}}|H(w)|^2)|_{w=\pi} = 0$. ■

Recall Proposition 2.2.2 which states that if we have a real-valued filter

$h=(h_l, h_{l+1}, \dots, h_{L-1}, h_L)$ whose Fourier series satisfies $|H(w)|^2 + |H(w + \pi)|^2 = 2$, then the orthonormality conditions

$$\sum_{k=0}^L h_k^2 = 1$$

and

$$\sum_{k=2m}^L h_k h_{k-2m} = 0$$

for $m = 1, 2, \dots, \frac{L-l-1}{2}$ are satisfied. We will need to assume that orthonormality for the next few results.

Lemma 3.1.2 *If a finite length filter with real-valued coefficients has a Fourier series satisfying the following:*

1. $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$
2. $|H(w)|^2 + |H(w + \pi)|^2 = 2$

then $(\frac{d^p}{dw^p}|H(w)|^2)|_{w=0} = 0$ for $p = 1, 2, \dots, 2m + 1$.

Proof. As in Lemma 3.1.1, we will only show that $(\frac{d^{2m+1}}{dw^{2m+1}}|H(w)|^2)|_{w=0} = 0$ but note that the smaller derivative conditions follow similarly.

Notice that $(\frac{d^{2m+1}}{dw^{2m+1}}(|H(w)|^2 + |H(w + \pi)|^2 = 2))|_{w=0}$ is equivalent to

$$\left(\frac{d^{2m+1}}{dw^{2m+1}}|H(w)|^2\right)\Big|_{w=0} + \left(\frac{d^{2m+1}}{dw^{2m+1}}|H(w + \pi)|^2\right)\Big|_{w=0} = 0. \quad (3.1.2)$$

We know from Lemma 3.1.1 that the second term in (3.1.2) must be 0. Therefore, we can conclude that $(\frac{d^{2m+1}}{dw^{2m+1}}|H(w)|^2)|_{w=0} = 0$. ■

3.2 Smoothness of $|H(w)|$

We use the above Lemmas 3.1.1 and 3.1.2 to help us to come to the following results about the smoothness of $|H(w)|$ at both $w = 0$ and $w = \pi$.

Theorem 3.2.1 *If a finite length filter with real-valued coefficients has a Fourier series satisfying the following:*

1. $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$
2. $|H(w)|^2 + |H(w + \pi)|^2 = 2$

then $(\frac{d^p}{dw^p}|H(w)|)|_{w=0} = 0$ for $p = 1, 2, \dots, 2m + 1$.

Note 3.2.2 *The lowpass condition $|H(0)| = \sqrt{2}$ follows directly from assumptions (1) and (2) above.*

Proof. (Induction on m)

Basis: Assume $H(\pi) = 0$. We want to show that $(\frac{d}{dw}|H(w)|)|_{w=0} = 0$.

We know from Lemma 3.1.2 that $(\frac{d}{dw}|H(w)|^2)|_{w=0} = 0$.

We can write this as $(2|H(w)|\frac{d}{dw}|H(w)|)|_{w=0} = 0$ and since $|H(0)| = \sqrt{2}$, this implies that $(\frac{d}{dw}|H(w)|)|_{w=0} = 0$.

Inductive Hypothesis: Given a finite length filter with real-valued coefficients satisfying the conditions $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m - 1$ and assumption (2) above, we assume that $(\frac{d^p}{dw^p}|H(w)|)|_{w=0} = 0$ for $p = 1, 2, \dots, 2m - 1$.

Inductive Step: We need to show that if we are given the hypotheses (1) and (2) above, then

- $(\frac{d^{2m}}{dw^{2m}} |H(w)|)|_{w=0} = 0.$
- $(\frac{d^{2m+1}}{dw^{2m+1}} |H(w)|)|_{w=0} = 0.$

We will only show $(\frac{d^{2m+1}}{dw^{2m+1}} |H(w)|)|_{w=0} = 0$ and note that the other condition follows similarly.

Consider

$$\begin{aligned} \frac{d^{2m+1}}{dw^{2m+1}} |H(w)|^2 &= \frac{d^{2m}}{dw^{2m}} \left(\frac{d}{dw} |H(w)|^2 \right) \\ &= \frac{d^{2m}}{dw^{2m}} \left(2|H(w)| \frac{d}{dw} |H(w)| \right) \\ &= 2 \frac{d^{2m}}{dw^{2m}} \left(|H(w)| \frac{d}{dw} |H(w)| \right). \end{aligned}$$

Now, by the Leibniz product rule we can write the above as

$$2 \sum_{k=0}^{2m} \binom{2m}{k} (|H(w)|)^{(k)} \left(\frac{d}{dw} |H(w)| \right)^{(2m-k)} = 2 \sum_{k=0}^{2m} \binom{2m}{k} (|H(w)|)^{(k)} (|H(w)|)^{(2m-k+1)}.$$

We know (from the inductive hypothesis) that for $k = 1, 2, \dots, 2m - 1$, the corresponding terms in the summation will equal 0 at $w = 0$. For $k = 2m$, the corresponding term is $\binom{2m}{2m} (|H(w)|)^{(2m)} (|H(w)|)^{(1)}$, which is also equal to 0 at $w = 0$. The only term which remains is the one corresponding to $k = 0$, that is, $\binom{2m}{0} (|H(w)|) (|H(w)|)^{(2m+1)}$.

Recall, by Lemma 3.1.2, we know that

$$\left(\frac{d^{2m+1}}{dw^{2m+1}} |H(w)|^2 \right) \Big|_{w=0} = 0$$

and so by all of the above it follows that

$$\left(2(|H(w)|) (|H(w)|)^{(2m+1)} \right) \Big|_{w=0} = 2\sqrt{2} \left(\frac{d^{2m+1}}{dw^{2m+1}} |H(w)| \right) \Big|_{w=0} = 0.$$

Hence, $(\frac{d^{2m+1}}{dw^{2m+1}} |H(w)|)|_{w=0} = 0.$

■

Theorem 3.2.3 Given a finite length filter satisfying $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$, then $\lim_{w \rightarrow \pi} (\frac{d^j}{dw^j} |H(w)|) = 0$ for $j = 0, 1, \dots, m$.

Proof. (Induction on m)

Basis: Assume $H(\pi) = 0$ and $H'(\pi) = 0$. We want to show that

$\lim_{w \rightarrow \pi} (\frac{d}{dw} |H(w)|) = 0$. (It is trivial that $\lim_{w \rightarrow \pi} |H(w)| = 0$.)

Consider $\frac{d}{dw} |H(w)|^2$. We can write this two different ways, as follows:

- $\frac{d}{dw} |H(w)|^2 = 2|H(w)| \frac{d}{dw} |H(w)|$.
- $\frac{d}{dw} |H(w)|^2 = H(w) \overline{H'(w)} + \overline{H(w)} H'(w)$.

Therefore, $\lim_{w \rightarrow \pi} (2|H(w)| \frac{d}{dw} |H(w)|) = \lim_{w \rightarrow \pi} (H(w) \overline{H'(w)} + \overline{H(w)} H'(w))$.

Because we are taking a limit as $w \rightarrow \pi$, we can divide through by $2|H(w)|$ to get

$$\lim_{w \rightarrow \pi} \left(\frac{d}{dw} |H(w)| \right) = \lim_{w \rightarrow \pi} \left(\frac{H(w)}{2|H(w)|} \overline{H'(w)} \right) + \lim_{w \rightarrow \pi} \left(\frac{\overline{H(w)}}{2|H(w)|} H'(w) \right).$$

We know that for any (nonzero) complex number z , $\frac{z}{|z|} = 1$ and $\frac{\bar{z}}{|z|} = 1$, so it follows that $\frac{H(w)}{2|H(w)|}$ and $\frac{\overline{H(w)}}{2|H(w)|}$ are equal to $\frac{1}{2}$.

Therefore we have $\lim_{w \rightarrow \pi} (\frac{d}{dw} |H(w)|) = \lim_{w \rightarrow \pi} (\frac{1}{2} \overline{H'(w)}) + \lim_{w \rightarrow \pi} (\frac{1}{2} H'(w)) = 0 + 0 = 0$.

Inductive Hypothesis: Given a finite-length filter satisfying $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m-1$, we assume that $\lim_{w \rightarrow \pi} (\frac{d^j}{dw^j} |H(w)|) = 0$ for $j = 0, 1, \dots, m-1$.

Inductive Step: We need to show that if $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$, then

$\lim_{w \rightarrow \pi} (\frac{d^m}{dw^m} |H(w)|) = 0$.

Consider $\frac{d^m}{dw^m} |H(w)|^2$. We can write this two different ways, as follows:

- $\frac{d^m}{dw^m} |H(w)|^2 = \frac{d^{m-1}}{dw^{m-1}} (2|H(w)| \frac{d}{dw} |H(w)|) = 2 \frac{d^{m-1}}{dw^{m-1}} (|H(w)| \frac{d}{dw} |H(w)|)$.
- $\frac{d^m}{dw^m} |H(w)|^2 = \frac{d^m}{dw^m} (H(w) \overline{H(w)})$.

Using the Leibniz product rule, we can rewrite these as follows:

- $\frac{d^m}{dw^m} |H(w)|^2 = 2 \sum_{k=0}^{m-1} \binom{m-1}{k} (|H(w)|)^{(k)} (\frac{d}{dw} |H(w)|)^{(m-1-k)}$
 $= 2 \sum_{k=0}^{m-1} \binom{m-1}{k} (|H(w)|)^{(k)} (|H(w)|)^{(m-k)}$.
- $\frac{d^m}{dw^m} |H(w)|^2 = \sum_{k=0}^m \binom{m}{k} (H^{(k)}(w)) (\overline{H^{(m-k)}(w)})$.

We rewrite the two equations above in a slightly different form, as follows:

•

$$\begin{aligned} \frac{d^m}{dw^m} |H(w)|^2 &= 2 \binom{m-1}{0} (|H(w)|) (|H(w)|)^{(m)} \\ &\quad + 2 \sum_{k=1}^{m-1} \binom{m-1}{k} (|H(w)|)^{(k)} (|H(w)|)^{(m-k)}. \end{aligned}$$

•

$$\begin{aligned} \frac{d^m}{dw^m} |H(w)|^2 &= \binom{m}{0} (H(w)) (\overline{H^{(m)}(w)}) \\ &\quad + \sum_{k=1}^{m-1} \binom{m}{k} (H^{(k)}(w)) (\overline{H^{(m-k)}(w)}) + \binom{m}{m} (H^{(m)}(w)) \overline{H(w)}. \end{aligned}$$

It follows that

$$\begin{aligned} &\lim_{w \rightarrow \pi} (2(|H(w)|) (|H(w)|)^{(m)}) + \lim_{w \rightarrow \pi} \left(2 \sum_{k=1}^{m-1} \binom{m-1}{k} (|H(w)|)^{(k)} (|H(w)|)^{(m-k)} \right) \\ &= \lim_{w \rightarrow \pi} \left(\binom{m}{0} (H(w)) (\overline{H^{(m)}(w)}) \right) + \lim_{w \rightarrow \pi} \left(\sum_{k=1}^{m-1} \binom{m}{k} (H^{(k)}(w)) (\overline{H^{(m-k)}(w)}) \right) \\ &\quad + \lim_{w \rightarrow \pi} \left(\binom{m}{m} (H^{(m)}(w)) \overline{H(w)} \right). \end{aligned}$$

By the induction hypothesis, we know that

$$\lim_{w \rightarrow \pi} \left(2 \sum_{k=1}^{m-1} \binom{m-1}{k} (|H(w)|)^{(k)} (|H(w)|)^{(m-k)} \right) = 0.$$

By the assumptions that $H^{(j)}(\pi) = 0$ for $j = 0, 1, \dots, m$, we know that

$$\lim_{w \rightarrow \pi} \left(\sum_{k=1}^{m-1} \binom{m}{k} (H^{(k)}(w)) (\overline{H^{(m-k)}(w)}) \right) = 0.$$

Therefore,

$$\begin{aligned} &\lim_{w \rightarrow \pi} (2(|H(w)|) (|H(w)|)^{(m)}) \\ &= \lim_{w \rightarrow \pi} \left(\binom{m}{0} (H(w)) (\overline{H^{(m)}(w)}) \right) + \lim_{w \rightarrow \pi} \left(\binom{m}{m} (H^{(m)}(w)) \overline{H(w)} \right). \end{aligned}$$

Now since we are taking a limit as $w \rightarrow \pi$, we can divide through by $2|H(w)|$ to get

$$\lim_{w \rightarrow \pi} (|H(w)|)^{(m)} = \lim_{w \rightarrow \pi} \left(\binom{m}{0} \frac{H(w)}{2|H(w)|} (\overline{H^{(m)}(w)}) \right) + \lim_{w \rightarrow \pi} \left(\binom{m}{m} (H^{(m)}(w)) \frac{\overline{H(w)}}{2|H(w)|} \right).$$

In the same manner as in the basis, we see that each of the limits on the right side tends to 0 and therefore,

$$\lim_{w \rightarrow \pi} \left(\frac{d^m}{dw^m} |H(w)| \right) = 0.$$

■

Example 3.2.4 Consider the Daubechies length 4 filter:

$$h = (h_0, h_1, h_2, h_3) = \left(\frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}} \right).$$

The graphs of $|H(w)|$ and $|H(w)|^2$ on $[0, \pi]$ are shown in Figures 3.1 and 3.2, respectively (see next page). The graphs of both $|H(w)|$ and $|H(w)|^2$ on $[0, 4\pi]$ are shown in Figure 3.3.

The Daubechies filter satisfies all the conditions in Lemma 3.1.1, Lemma 3.1.2, Theorem 3.2.1, and Theorem 3.2.3. Therefore, in the figures, we see that the “smoothness” of $|H(w)|^2$ is the same at $w = 0$ as it is at $w = \pi$. Even more interesting, the graph of $|H(w)|$ is about twice as “smooth” at $w = 0$ as it is at $w = \pi$, even though the derivative conditions are taken at $w = \pi$.

Now let’s consider what happens if we have the derivative conditions $H^{(j)}(0) = 0$ for $j = 1, 2, \dots, m$, along with orthonormality, as is the case for half the derivative conditions imposed on Coiflet filters. What implications do these conditions have on the smoothness of $|H(w)|$ and $|H(w)|^2$? It is easy to show (similar to the arguments in Lemma 3.1.2 and Theorem 3.2.1) that with these conditions, we only get $(\frac{d^j}{dw^j} |H(w)|)|_{w=0} = 0$ for $p = 1, 2, \dots, m$. In other words, the derivative conditions taken at $w = \pi$ have a greater impact on the smoothness of $|H(w)|$ at $w = 0$ than the derivative conditions taken at $w = 0$. We can also see, simply by experiment, that the conditions $H^{(j)}(0) = 0$ for $j = 1, 2, \dots, m$ have little effect on the smoothness of $|H(w)|$ at $w = \pi$.

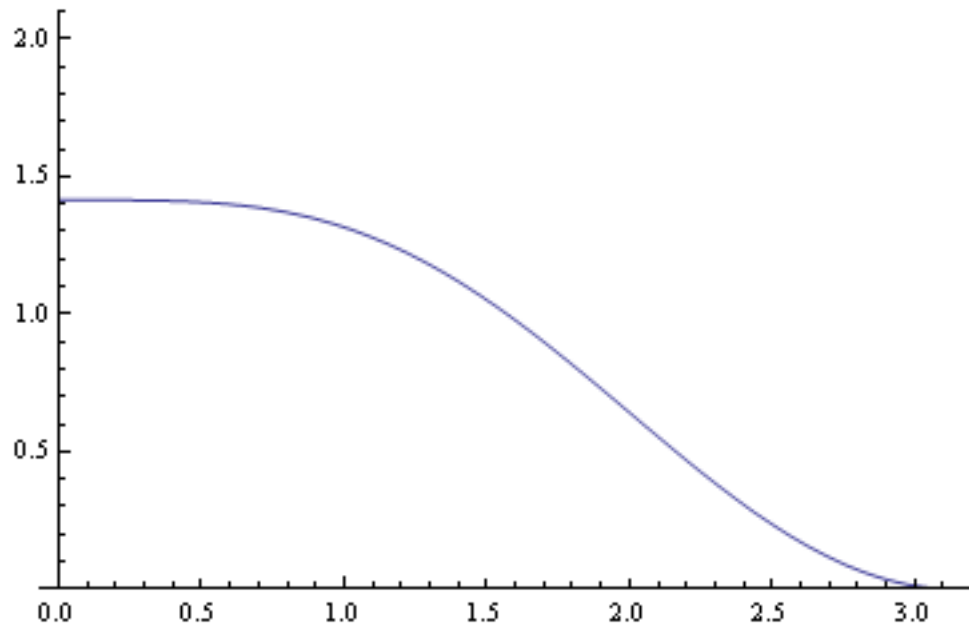


Figure 3.1: $|H(w)|$ on $[0, \pi]$ for Daubechies length 4 filter

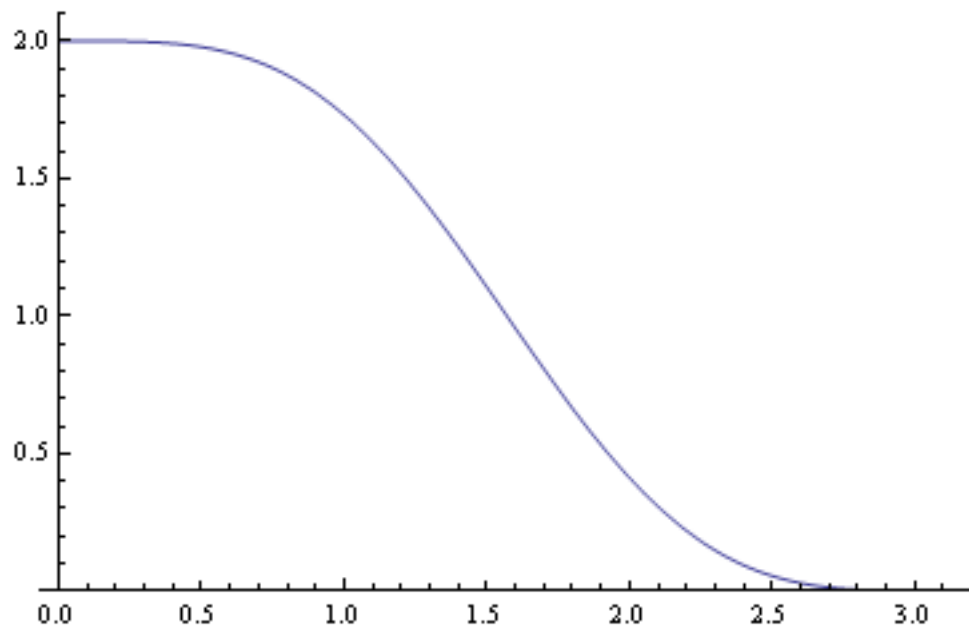


Figure 3.2: $|H(w)|^2$ on $[0, \pi]$ for Daubechies length 4 filter

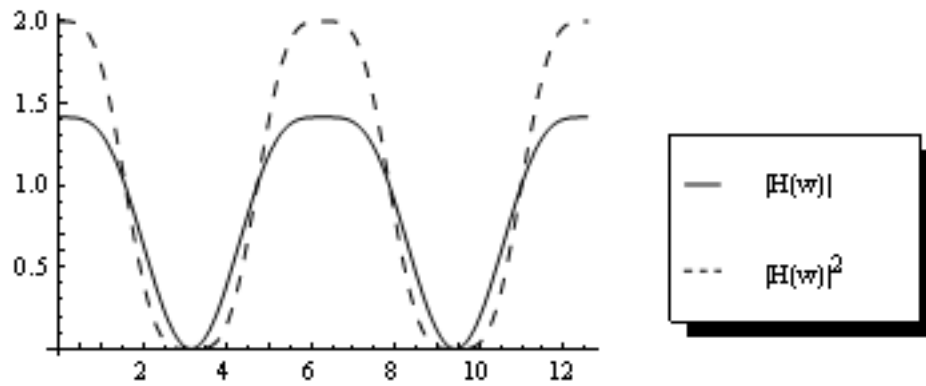


Figure 3.3: $|H(w)|$ and $|H(w)|^2$ on $[0, 4\pi]$ for Daubechies length 4 filter

Thus, for the filters we consider in this paper, the derivative conditions taken at $w = 0$ are irrelevant when considering smoothness of $|H(w)|$. This is because the number of derivative conditions that we take at π is always greater than (Daubechies) or equal to (Coiflet) the number of derivative conditions that we take at 0.

When comparing Daubechies filters and Coiflet filters of the same length, we would expect the graph of the modulus for the Daubechies filter to be smoother at both $w = 0$ and $w = \pi$ than the graph of the modulus of the Coiflet filter would be near those points. Figure 3.4 and Figure 3.5 compare the modulus of the Fourier series corresponding to the Daubechies and Coiflet filters of length 6 and length 12, respectively. We will come back to this comparison at the end of Chapter 4 when we suggest a new filter construction that blends the characteristics of both the Daubechies and Coiflet filter constructions.

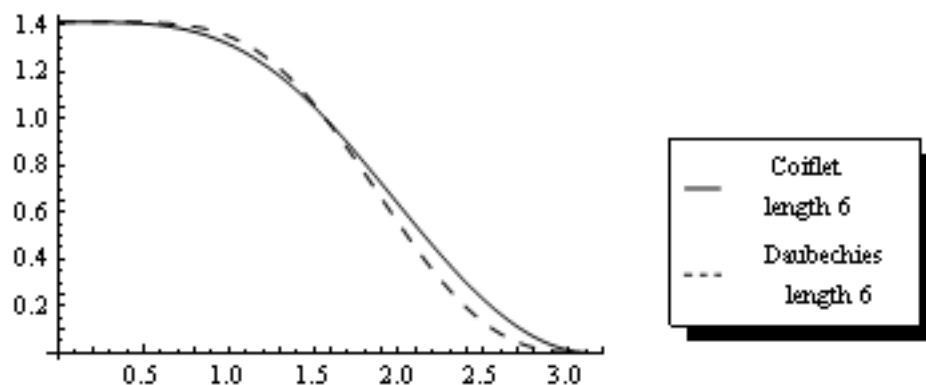


Figure 3.4: $|H(w)|$ for Daubechies and Coiflet filters (length 6)

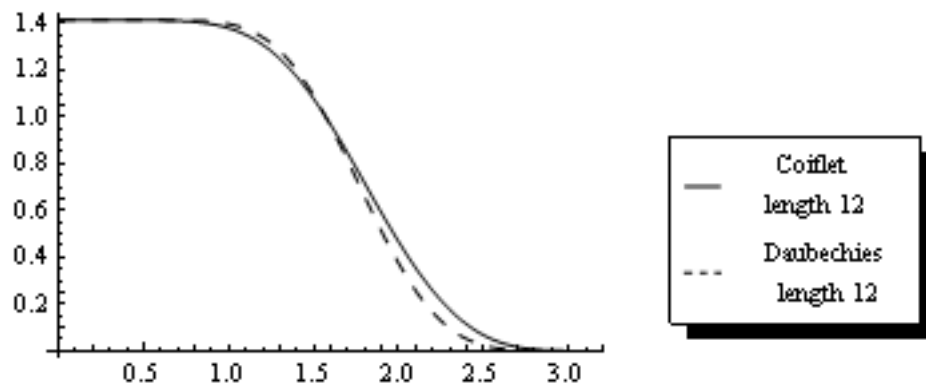


Figure 3.5: $|H(w)|$ for Daubechies and Coiflet filters (length 12)

4 WRAPPING

In any DWT, defined as in Definition 1.0.10, that is, constructed using filters of length greater than two, we automatically see wrapping occur among the rows of the DWT. That is, some of the rows are translated (by 2's) so far that the (nonzero) entries of the filter must be wrapped around to the first few columns of the matrix. For example, in Figure 4.1, wrapping occurs in rows 4 through 7.

We now have a thorough understanding of the filter constructions for some of the most well-known filters, namely the Daubechies and Coiflet filters. Hence, we thought it would be interesting to consider why wrapping, which occurs in the majority of DWTs, does not seem to affect the construction conditions of these filters. More specifically, since these filters were constructed in a way which assured orthogonality of their corresponding DWTs, we question if and how wrapping affects the orthogonality conditions as they were originally stated in Chapter 2.

Suppose that we consider the lowpass portion of $N \times N$ transform matrices which are formed using filters of length $L + 1$. In this section, we assume that N and $L + 1$ are both even. For simplicity, we will use filters of the form $h = (h_0, h_1, \dots, h_L)$ (same as Daubechies filters), but we note that everything that follows in this chapter can easily be extended to even length filters with *any* indices (as with Coiflet filters whose indices are both negative and positive). Figure 4.1 gives an example of the lowpass portion of a DWT of length $N = 14$ with filter length $L + 1 = 9 + 1 = 10$. Take note that in row 1 there are 4 zeros following the filter coefficients of h . In general (i.e., for filter length $L + 1$ and matrix length N), we say that there are $c - 1$ zeros placed in the remaining entries of row 1 so that $N = L + c = (L + 1) + (c - 1)$.

Definition 4.0.5 A **non-wrapping row** in a DWT W_N is one in which the entries of the row have been 2-translated $\frac{c-1}{2}$ or fewer times (i.e., $c - 1$ or fewer spaces to the right) from their original position in row 1, where $c = N - L$, and $L + 1$ is the length of the filter.

$$\begin{pmatrix} h_9 & h_8 & h_7 & h_6 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_9 & h_8 & h_7 & h_6 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_9 & h_8 & h_7 & h_6 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & h_9 & h_8 & h_7 & h_6 & h_5 & h_4 & h_3 & h_2 \\ h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & h_9 & h_8 & h_7 & h_6 & h_5 & h_4 \\ h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & h_9 & h_8 & h_7 & h_6 \\ h_7 & h_6 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & h_9 & h_8 \end{pmatrix}$$

Figure 4.1: Lowpass portion of a 14×14 DWT (the top half of the matrix)

In other words, a non-wrapping row does not have any nonzero entries which are wrapped around to the first column of the transform matrix.

Definition 4.0.6 A **wrapping row** in a DWT W_N is one in which the entries of the row have been 2-translated $\frac{c+1}{2}$ or more times (i.e., $c+1$ or more spaces to the right) from their original position in row 1, where $c = N - L$, and $L + 1$ is the length of the filter.

In other words, a wrapping row has at least two nonzero entries wrapped around to the first columns of the transform matrix.

Note 4.0.7 Row 1 (abbreviated R1) is considered its own entity, neither a wrapping nor a non-wrapping row. Therefore, in Figure 4.1, row 2-row 3 are non-wrapping rows while rows 4 through 7 are wrapping rows.

We will denote the k th non-wrapping row by R_k^{NW} and the j th wrapping row by R_j^W . Then, in Figure 4.1, row 2 and row 3 are denoted R_1^{NW} and R_2^{NW} , respectively, and rows 4 through 7 are denoted $R_1^W - R_4^W$.

It is clear from the definition of non-wrapping rows that there are $\frac{c-1}{2}$ non-wrapping rows in the lowpass portion of any transform matrix. In the following Lemma, we show that the number of wrapping rows in the lowpass portion of a DWT depends solely on the length of the filter.

Lemma 4.0.8 The number of wrapping rows in the lowpass portion of a DWT (matrix length N , filter length $L + 1$) is $\frac{L-1}{2}$.

Proof. The lowpass portion of a DWT has $\frac{N}{2}$ rows total. From this, we subtract the number of non-wrapping rows ($\frac{c-1}{2}$) and another 1 (since we do not include R1). Therefore, there

are

$$\frac{N}{2} - \frac{c-1}{2} - 1 = \frac{L+c}{2} - \frac{c-1}{2} - \frac{2}{2} = \frac{L-1}{2}$$

wrapping rows in the lowpass portion of the transform matrix. ■

We recall the zero orthogonality conditions which characterize the solution of Daubechies filters of length $L + 1$:

$$\sum_{k=2m}^L h_k h_{k-2m} = 0 \quad \text{for } m = 1, 2, \dots, \frac{L-1}{2}. \quad (4.0.1)$$

We can think of these conditions as dot products of the original filter with all possible 2-translates of the filter.

Note 4.0.9 *We will refer to the equations in (4.0.1) as the set of standard zero orthogonality conditions.*

Because the standard zero orthogonality conditions are equivalent to the dot products of the original filter and all possible 2-translates (without any wrapping), then we pose the following question: can we use the same set of orthogonality conditions for matrices which involve a large number of wrapping rows? In the next few paragraphs, we will generalize the form of (1) the dot product of $R1$ with non-wrapping rows and (2) the dot product of $R1$ with wrapping rows, in hopes that we may answer the question above in the affirmative.

4.1 Non-wrapping Rows

It is easy to see that the dot products of the $\frac{c-1}{2}$ non-wrapping rows (of a transform matrix) with $R1$ are the same as the first $\frac{c-1}{2}$ standard zero orthogonality conditions. In other words, we can generalize the dot product of R_n^{NW} with $R1$ as follows:

$$R_n^{NW} \cdot R1 = \sum_{k=2n}^L h_k h_{k-2n} \quad \text{for } n = 1, 2, \dots, \frac{c-1}{2}. \quad (4.1.2)$$

Example 4.1.1 Consider the lowpass portion of the DWT shown in Figure 4.1. The value of $c - 1$ for this matrix is 4. Therefore, there are $\frac{c-1}{2} = \frac{4}{2} = 2$ non-wrapping rows in this

matrix, namely row 2 and row 3. The dot products of these two rows with $R1$ are

$$h_9h_7 + h_8h_6 + h_7h_5 + h_6h_4 + h_5h_3 + h_4h_2 + h_3h_1 + h_2h_0$$

and

$$h_9h_5 + h_8h_4 + h_7h_3 + h_6h_2 + h_5h_1 + h_4h_0,$$

respectively. These are the same conditions as in (4.1.2).

Next we will generalize a formula for the dot product of the wrapping rows with $R1$.

4.2 Wrapping Rows

In order to generalize the formula for the dot product of wrapping rows

$R_1^W, R_2^W, \dots, R_{\frac{L-1}{2}}^W$ with $R1$, we must consider both the *wrapping portion* and the *non-wrapping portion* of each dot product.

Definition 4.2.1 The **wrapping portion** of $R1 \cdot R_n^W$ is the portion of the dot product obtained from the entries in R_n^W which have been wrapped around.

Definition 4.2.2 The **non-wrapping portion** of $R1 \cdot R_n^W$ is the portion of the dot product obtained from the entries in R_n^W which have not been wrapped around.

4.2.1 Non-wrapping Portion

Consider the first wrapping row. For this row, we have shifted the filter $\frac{c+1}{2}$ times (each time shifting two spaces) so that each filter element has been shifted to the right $c + 1$ spaces from its original position (as in $R1$). Therefore, the non-wrapping portion of the dot product $R1 \cdot R_1^W$ is really just the $(\frac{c+1}{2})$ th standard zero orthogonality condition. It is clear that this extends for the non-wrapping portion of each of the dot products $R1 \cdot R_2^W, R1 \cdot R_3^W, \dots, R1 \cdot R_{\frac{L-1}{2}}^W$. That is, the non-wrapping portion of $R1 \cdot R_n^W$ for $n = 1, 2, \dots, \frac{L-1}{2}$ is given by

$$\sum_{k=2m}^L h_k h_{k-2m} \quad \text{for } m = n + \frac{c-1}{2} \tag{4.2.3}$$

(i.e., for $m = \frac{c+1}{2}, \frac{c+3}{2}, \dots, \frac{c+L-2}{2}$).

Note 4.2.3 *We wish to make two observations.*

- *From the previous formula, it is easy to see that for $m = \frac{L+1}{2}, \dots, \frac{c+L-2}{2}$, there will be no resulting orthogonality conditions. In other words, the non-wrapping portions of $R1 \cdot R_{\frac{L-c+2}{2}}^W, \dots, R1 \cdot R_{\frac{L-1}{2}}^W$ are all zero.*
- *Also, if $c > L$, then there will be no resulting orthogonality conditions stemming from the non-wrapping portion of any of the wrapping rows. For an example, see Figure 4.2, the lowpass portion of a 10×10 matrix with $L = 3$ and $c = 7$.*

$$\begin{pmatrix} h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 \\ h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 \end{pmatrix}$$

Figure 4.2: Lowpass portion of a 10×10 DWT (the top half of the matrix) with $c > L$.

We rewrite (4.2.3) so that it is indexed by the numeric value of the wrapping row. That is, we want to rewrite (4.2.3) so that it is indexed from 1 to $\frac{L-1}{2}$. To do this, we substitute $m = n + \frac{c-1}{2}$ to get

$$\sum_{k=2m}^L h_k h_{k-2m} = \sum_{k=2(n+\frac{c-1}{2})}^L h_k h_{k-2(n+\frac{c-1}{2})} \quad \text{for } n = 1, 2, \dots, \frac{L-1}{2}. \quad (4.2.4)$$

4.2.2 Wrapping Portion

The equation in (4.2.4) gives the generalization of the non-wrapping portion of $R1 \cdot R_n^W$ for $n = 1, \dots, \frac{L-1}{2}$. Next, we consider the wrapping portion of the dot product of $R1$ with each of the wrapping rows $R_1^W, R_2^W, \dots, R_{\frac{L-1}{2}}^W$. We observe that R_1^W has precisely two filter elements (h_1 and h_0) which are wrapped around to the first columns of the DWT, R_2^W has four filter elements (h_3, h_2, h_1 , and h_0) which are wrapped around, and so on in increments of two until $R_{\frac{L-1}{2}}^W$ has $L-1$ filter elements (h_{L-2}, \dots, h_0) wrapped around. Therefore, the orthogonality condition that we obtain from the wrapping portion of $R1 \cdot R_1^W$ is $h_1 h_L + h_0 h_{L-1}$, from the wrapping portion of $R1 \cdot R_2^W$ is $h_3 h_L + h_2 h_{L-1} + h_1 h_{L-2} + h_0 h_{L-3}$, and so on until the wrapping portion of $R1 \cdot R_{\frac{L-1}{2}}^W$ which is $h_{L-2} h_L + h_{L-3} h_{L-1} + \dots + h_1 h_3 + h_0 h_2$.

Thus, we can liken the orthogonality conditions obtained by the wrapping portion of $R1 \cdot R_n^W$ (for $n = 1, 2, \dots, \frac{L-1}{2}$) to those that we get by shifting of rows in the standard zero orthogonality conditions, except in reverse order. That is, the wrapping portion of $R1 \cdot R_n^W$ is given by the following:

$$\sum_{k=2(\frac{L+1}{2}-n)}^L h_k h_{k-2(\frac{L+1}{2}-n)} \quad \text{for } n = 1, 2, \dots, \frac{L-1}{2}. \quad (4.2.5)$$

Now, by combining the results in (4.2.4) and (4.2.5), the complete generalization of $R1 \cdot R_n^W$ can be written as follows:

$$\sum_{k=2(\frac{L+1}{2}-n)}^L h_k h_{k-2(\frac{L+1}{2}-n)} + \sum_{k=2(n+\frac{c-1}{2})}^L h_k h_{k-2(n+\frac{c-1}{2})} \quad \text{for } n = 1, 2, \dots, \frac{L-1}{2}. \quad (4.2.6)$$

In order to have orthogonality among all the rows of the lowpass portion of a DWT, all of the expressions in (4.2.6) must be set to zero. (Remember, we also need the condition $R1 \cdot R1 = 1$ to have an orthonormal set, but we will concern ourselves only with the zero orthogonality conditions at this point.)

Example 4.2.4 Again, consider the lowpass portion of the DWT shown in Figure 4.1. The value of L for the filter in this matrix is 9. Therefore, there are $\frac{L-1}{2} = \frac{8}{2} = 4$ wrapping rows in this matrix (see Lemma 4.0.8), namely row 4, row 5, row 6, and row 7. The dot products of these four rows with $R1$ are

$$h_9 h_1 + h_8 h_0 + h_3 h_9 + h_2 h_8 + h_1 h_7 + h_0 h_6,$$

$$h_9 h_3 + h_8 h_2 + h_7 h_1 + h_6 h_0 + h_1 h_9 + h_0 h_8,$$

$$h_9 h_5 + h_8 h_4 + h_7 h_3 + h_6 h_2 + h_5 h_1 + h_4 h_0,$$

and

$$h_9 h_7 + h_8 h_6 + h_7 h_5 + h_6 h_4 + h_5 h_3 + h_4 h_2 + h_3 h_1 + h_2 h_0,$$

respectively. These are the same conditions as in (4.2.6).

The complete generalization formula given in (4.2.6) shows that the wrapping portion and the non-wrapping portion are each equivalent to one of the standard zero orthogonality

conditions. Therefore, we see that having all of the standard orthogonality conditions in our system is sufficient, even in a DWT involving wrapping, to satisfy the orthogonality necessary for the lowpass portion of a DWT. Notice the word *sufficient* just used. We now ask, do we *need* all $\frac{L-1}{2}$ of the standard orthogonality conditions in order to obtain orthogonality in the lowpass portion of a DWT? That is, we know the standard zero orthogonality conditions are sufficient for an orthogonal DWT containing wrapping, but are they necessary?

4.3 Effects of Wrapping on Zero Orthogonality Conditions

In this section, we will first show that some of the orthogonality conditions from (4.2.6) will repeat themselves if we have a significant amount of wrapping in a DWT. Then we will conclude that the set of standard zero orthogonality conditions are in fact not all necessary in order to maintain orthogonality in this case.

Lemma 4.3.1 *The equality $R1 \cdot R_m^{NW} = R1 \cdot R_{\frac{L-1}{2}-(m-1)}^W$ holds for $m = 1, 2, \dots, \frac{c-1}{2}$.*

In other words, the orthogonality condition we obtain from the dot product of the first non-wrapping row with row 1 is the same as the orthogonality condition that we obtain from the dot product of the last wrapping row with row 1. Similarly this follows for the second non-wrapping row and the second-to-last wrapping row, third non-wrapping row and third-to-last wrapping row, and so on for all possible non-wrapping rows.

Note 4.3.2 *If $c > L$, then there will be more non-wrapping rows than there are wrapping rows (see Figure 4.2). We noted before in Note 4.2.3 that this is the trivial case when the non-wrapping portion of the wrapping rows are all zero. For the remainder of this section, we will focus on the non-trivial case (i.e., $c \leq L$).*

Proof. Recall from Note 4.2.3 that the non-wrapping portions of $R1 \cdot R_{\frac{L-c+2}{2}}^W, \dots, R1 \cdot R_{\frac{L-1}{2}}^W$ are all zero. So for the last $\frac{L-1}{2} - \frac{L-c+2}{2} + 1 = \frac{c-1}{2}$ wrapping rows, we obtain no orthogonality conditions stemming from the non-wrapping portion of the dot product. Therefore, we only need to compare the orthogonality conditions from the $\frac{c-1}{2}$ non-wrapping rows to the orthogonality conditions coming from the *wrapping portion* of the last $\frac{c-1}{2}$ wrapping rows.

The conditions from the $\frac{c-1}{2}$ non-wrapping rows are given by

$$\sum_{k=2m}^L h_k h_{k-2m} \quad \text{for } m = 1, 2, \dots, \frac{c-1}{2}. \quad (4.3.7)$$

The conditions from the last $\frac{c-1}{2}$ wrapping rows are given by

$$\sum_{k=2(\frac{L+1}{2}-n)}^L h_k h_{k-2(\frac{L+1}{2}-n)} \quad \text{for } n = \frac{L-1}{2} - (m-1) \quad \text{where } m = 1, 2, \dots, \frac{c-1}{2}. \quad (4.3.8)$$

We substitute the value for n into (4.3.8) to get

$$\begin{aligned} \sum_{k=2(\frac{L+1}{2}-n)}^L h_k h_{k-2(\frac{L+1}{2}-n)} &= \sum_{k=2(\frac{L+1}{2}-(\frac{L-1}{2}-(m-1)))}^L h_k h_{k-2(\frac{L+1}{2}-(\frac{L-1}{2}-(m-1)))} \\ &= \sum_{k=2m}^L h_k h_{k-2m} \quad \text{for } m = 1, 2, \dots, \frac{c-1}{2}. \end{aligned}$$

■

Now we must consider the remaining wrapping rows (i.e., those not considered in Lemma 4.3.1). There are $\frac{L-1}{2} - \frac{c-1}{2} = \frac{L-c}{2}$ of these remaining wrapping rows.

Lemma 4.3.3 *The equality $R1 \cdot R_m^W = R1 \cdot R_{\frac{L-c}{2}-(m-1)}^W$ holds for $m = 1, 2, \dots, \frac{L-c}{2}$.*

Proof.

1.

$$\begin{aligned} R1 \cdot R_m^W &= \sum_{k=2(\frac{L+1}{2}-m)}^L h_k h_{k-2(\frac{L+1}{2}-m)} + \sum_{k=2(m+\frac{c-1}{2})}^L h_k h_{k-2(m+\frac{c-1}{2})} \\ &= \sum_{k=L+1-2m}^L h_k h_{k-(L+1-2m)} + \sum_{k=2m+c-1}^L h_k h_{k-(2m+c-1)}. \end{aligned}$$

2.

$$\begin{aligned}
R1 \cdot R_{\frac{L-c}{2}-(m-1)}^W &= \sum_{k=2(\frac{L+1}{2}-(\frac{L-c}{2}-(m-1)))}^L h_k h_{k-2(\frac{L+1}{2}-(\frac{L-c}{2}-(m-1)))} \\
&+ \sum_{k=2((\frac{L-c}{2}-(m-1))+\frac{c-1}{2})}^L h_k h_{k-2((\frac{L-c}{2}-(m-1))+\frac{c-1}{2})} \\
&= \sum_{k=2m+c-1}^L h_k h_{k-(2m+c-1)} + \sum_{k=L+1-2m}^L h_k h_{k-(L+1-2m)}.
\end{aligned}$$

■

The following theorem uses Lemma 4.3.1 and Lemma 4.3.3 to show how many zero orthogonality conditions are *necessary* in the lowpass portion of a DWT.

Theorem 4.3.4 *For a DWT with length $N = L + c$ (composed of a lowpass/highpass filter pair of length $L + 1$), there are only $\lceil \frac{N-2}{4} \rceil$ zero orthogonality conditions which are necessary in order to satisfy orthogonality of the lowpass portion of W_N .*

Proof. Out of the $c - 1$ rows considered in Lemma 4.3.1 (i.e., $\frac{c-1}{2}$ non-wrapping rows and $\frac{c-1}{2}$ wrapping rows), only $\frac{c-1}{2}$ unique orthogonality conditions result from their dot products with $R1$.

Now, out of the $\frac{L-c}{2}$ remaining wrapping rows considered in Lemma 4.3.3, we obtain the following:

- If $\frac{L-c}{2}$ is even, then $\frac{L-c}{4}$ unique orthogonality conditions result from their dot products with $R1$.
- If $\frac{L-c}{2}$ is odd then we get $\lfloor \frac{L-c}{4} \rfloor + 1 = \lceil \frac{L-c}{4} \rceil$ unique orthogonality conditions that result from their dot products with $R1$.

We know that for the first case, when $\frac{L-c}{2}$ is even, then $\frac{L-c}{4}$ is an integer and hence $\frac{L-c}{4} = \lceil \frac{L-c}{4} \rceil$. Therefore, putting together the results from Lemma 4.3.1 and Lemma 4.3.3, we have $\frac{c-1}{2} + \lceil \frac{L-c}{4} \rceil = \lceil \frac{c-1}{2} + \frac{L-c}{4} \rceil = \lceil \frac{L+c-2}{4} \rceil = \lceil \frac{N-2}{4} \rceil$ *necessary* zero orthogonality conditions.

■

Thus, Theorem 4.3.4 implies that if W_N has a significant amount of wrapping (i.e., $c \leq L$), then we do not need all $\frac{L-1}{2}$ of the standard zero orthogonality conditions to satisfy orthogonality in the lowpass portion of the matrix.

Corollary 4.3.5 *There are $\lfloor \frac{L-c}{4} \rfloor$ degrees of freedom resulting in the system (compared to the standard system containing $\frac{L-1}{2}$ zero orthogonality conditions).*

Note 4.3.6 *Again, we are focusing on the case with significant amounts of wrapping, that is when $c \leq L$. Otherwise, if $c > L$, it is easy to see that we would obtain no degrees of freedom in the system.*

Proof. Recall that there are $\frac{L-1}{2}$ standard zero orthogonality conditions and only $\lceil \frac{N-2}{4} \rceil$ necessary zero orthogonality conditions. To obtain the degrees of freedom in our system, we subtract the number of necessary conditions from the number of standard conditions, as follows:

$$\frac{L-1}{2} - \lceil \frac{N-2}{4} \rceil = \frac{2L-2}{4} - \lceil \frac{L+c-2}{4} \rceil. \quad (4.3.9)$$

Note that $L+c-2$ is always even. Therefore, when we divide by 4 we get either an integer or an integer and a half. Thus,

$$(4.3.9) = \begin{cases} \frac{2L-2}{4} - (\frac{L+c-2}{4}), & \text{if } \frac{L+c-2}{4} \in \mathbb{Z} \\ \frac{2L-2}{4} - (\frac{L+c-2}{4} + \frac{1}{2}), & \text{if } \frac{L+c-2}{4} \notin \mathbb{Z} \end{cases}$$

$$= \begin{cases} \frac{L-c}{4}, & \text{if } \frac{L+c-2}{4} \in \mathbb{Z} \\ \frac{L-c}{4} - \frac{1}{2}, & \text{if } \frac{L+c-2}{4} \notin \mathbb{Z}. \end{cases}$$

We note, as with $L+c-2$ above, that $L-c$ is always even. In addition, we can show that when $\frac{L+c-2}{4}$ is odd, then $\frac{L-c}{4}$ is also odd. To show this, we subtract the even number $\frac{2c-2}{2}$ from the odd number $\frac{L+c-2}{2}$ to get a (necessarily odd) result of $\frac{L-c}{2}$. Therefore, $\frac{L-c}{4}$ is always an integer or an integer and a half (and so occurs concurrently at the respective times when $\frac{L+c-2}{4}$ is an integer or an integer and a half).

Thus, $(4.3.9) = \lfloor \frac{L-c}{4} \rfloor$. ■

4.4 A New Filter Construction

We know that there are certain benefits of using filters having the maximal number of derivative conditions at $w = \pi$ (see [20, 8]). On the other hand, we know that there are also benefits of working with filters having equal numbers of derivative conditions at $w = \pi$ and $w = 0$ (see [1, 3, 9]). We use this information and Corollary 4.3.5 (which allows additional conditions in the system of equations used for filter constructions) to propose a new filter construction which could be used in transform matrices containing significant amounts of wrapping. This new construction modifies the standard Coiflet filter construction in that it adds derivative conditions at $w = \pi$ while maintaining a close balance between derivative conditions at $w = \pi$ and $w = 0$. That is, the number of derivative conditions at $w = \pi$ is increased and thus moves towards the maximal number of conditions (the benefit of Daubechies filters) while the symmetry of the derivative conditions is as optimal as possible (the benefit of Coiflet filters).

Note 4.4.1 *A DWT is the most computationally efficient when it is “sparse” in nature (i.e., there are many more 0 entries than filter coefficient entries). Therefore, a disadvantage of using the newly proposed filter construction is that the matrix would not be very sparse, since the construction relies on a large amount of wrapping.*

We now outline the system of equations used to construct this new filter. The orthonormality conditions include those that were stated in equations (4.1.2) and (4.2.6); they look a little different because we have extended them to the case for Coiflet filters in which we can have negative and positive filter indices. Notice that the last two orthonormality conditions depend on the value of c .

Lowpass conditions:

1. $|H(\pi)| = 0$.
2. $|H(0)| = \sqrt{2}$.

Orthonormality conditions:

1. $\sum_{k=l}^L h_k^2 = 1$.
2. $\sum_{k=l+2m}^L h_k h_{k-2n} = 0$ for $n = 1, 2, \dots, \frac{c-1}{2}$.

3. Let $\mathcal{C} = \frac{c-1}{2}$ and $\mathcal{L} = \frac{L-l+1}{2}$ then

$$\sum_{k=l+2(\mathcal{L}-n)}^L h_k h_{k-2(\mathcal{L}-n)} + \sum_{k=l+2(n+\mathcal{C})}^L h_k h_{k-2(n+\mathcal{C})} = 0$$

for $n = \frac{c+1}{2}, \dots, \frac{L-l-1}{2}$.

Derivative conditions (The values of α and β change depending on the number of degrees of freedom in the system):

1. $H^{(p)}(\pi) = 0$ for $p = 1, 2, \dots, \alpha$.
2. $H^{(p)}(0) = 0$ for $p = 1, 2, \dots, \beta$.

For the standard Coiflet filter, the values of α and β are both $2K - 1$. Therefore, for our modified filter, we first increase the value of α to $2K$, then, if possible, we increase the value of β to $2K$. After that we would continue one by one increasing the value of α and then increasing the value of β . From our empirical observations, we form the following conjecture:

Conjecture 4.4.2 *If the system of equations has t degrees of freedom (from wrapping), then at most $\lfloor \frac{t}{2} \rfloor$ derivative conditions can be added to the system.*

When we solved the above system, the majority of the time we obtained multiple real solutions. In Figures 4.3 and 4.4, we have given some examples of the filters obtained by solving this system. Note that there is no solution for a modified filter with $K = 3$ and $c = 7$, but there is a solution for a modified filter with $K = 3$ and $c = 9$. We include the cases where there is only one degree of freedom so that the reader can see how we came to the conjecture stated above.

In Figure 4.5, we provide a graphical comparison between the modulus graphs of the standard Daubechies filter, the standard Coiflet filter, and the modified filter ($K = 2$, $c = 3$); all of these are length 12 filters. In Figure 4.6, we compare the modulus of the Daubechies, Coiflet, and modified ($K = 3$, $c = 5$) filters of length 18. In these graphical representations, it is interesting to see that the graph of $|H(w)|$ corresponding to the modified filter construction “lies between” the graphs of $|H(w)|$ for the Daubechies and Coiflet filters of the same length.

| K | c | Deg. Of Freedom | Conditions Added | Filter |
|---|---|-----------------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 | 1 | 2 | 1 | { $h_0 \rightarrow 0.7871269814$, $h_3 \rightarrow -0.027724786355$, $h_1 \rightarrow 0.33171799012$, $h_1 \rightarrow 0.4971719869$, $h_4 \rightarrow 0.007899839164$, $h_2 \rightarrow -0.03975464795$, $h_2 \rightarrow -0.08053110444$, $h_4 \rightarrow 0.04052100435$, $h_6 \rightarrow -0.008155291297$, $h_3 \rightarrow 0.11098387021$, $h_5 \rightarrow 0.0170774851423$, $h_7 \rightarrow -0.000152024418154$ } |
| 2 | 3 | 2 | 1 | { $h_0 \rightarrow 0.7871269814$, $h_3 \rightarrow -0.027724786355$, $h_1 \rightarrow 0.33171799012$, $h_1 \rightarrow 0.4971719869$, $h_4 \rightarrow 0.007899839164$, $h_2 \rightarrow -0.03975464795$, $h_2 \rightarrow -0.08053110444$, $h_4 \rightarrow 0.04052100435$, $h_6 \rightarrow -0.008155291297$, $h_3 \rightarrow 0.11098387021$, $h_5 \rightarrow 0.0170774851423$, $h_7 \rightarrow -0.000152024418154$ } |
| 2 | 5 | 1 | 1 | No solutions |

Figure 4.3: Length 12 modified filters

| K | c | Deg. Of Freedom | Conditions Added | Filter |
|---|----|-----------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3 | 1 | 4 | 2 | { $h_0 = 0.7229903001, h_5 = 0.002091795071,$ $h_3 = -0.02942296042, h_1 = 0.3069038071,$ $h_4 = -0.001752785165, h_2 = -0.0009035056930,$ $h_6 = 0.0004704643518, h_{10} = -0.001540465297,$ $h_4 = 0.04403653401, h_6 = -0.02905652078,$ $h_1 = 0.5734789560, h_3 = -0.2070400810,$ $h_5 = 0.07681186583, h_9 = 0.0007876712402,$ $h_7 = -0.01678345431, h_{11} = 0.0002791816358,$ $h_2 = -0.0374500330878, h_8 = 0.0103127927281$ } |
| 3 | 3 | 3 | 1 | { $h_0 = 0.723889358846, h_5 = 0.00212674571617,$ $h_3 = -0.0296627958080, h_1 = 0.307603470879,$ $h_6 = 0.000434614107419, h_8 = 0.0102323398501,$ $h_4 = -0.00151893321897, h_{10} = -0.00152393140409,$ $h_2 = -0.0381147879608, h_2 = -0.00153722834604,$ $h_1 = 0.572360280349, h_7 = -0.0166484895336,$ $h_{11} = 0.000273796637646, h_9 = 0.000789849711196,$ $h_5 = 0.0762578891504, h_3 = -0.205993965915,$ $h_4 = 0.0442022148203, h_6 = -0.0289568655071$ } |
| 3 | 5 | 3 | 1 | { $h_0 = 0.723682728895, h_5 = 0.00211864901013,$ $h_3 = -0.0296072550075, h_1 = 0.307441594231,$ $h_6 = 0.000442917464779, h_8 = 0.0102512275994,$ $h_4 = -0.00157299983119, h_2 = -0.00139105488021,$ $h_2 = -0.0379631437828, h_{10} = -0.00152778710699,$ $h_7 = -0.0166782897017, h_1 = 0.572618598964,$ $h_9 = 0.000788786306098, h_3 = -0.206234567867,$ $h_5 = 0.0763841312856, h_{11} = 0.000275133965867,$ $h_4 = 0.0441658117133, h_6 = -0.0289809188849$ } |
| 3 | 7 | 2 | 1 | No solutions here (?) |
| 3 | 9 | 2 | 1 | { $h_0 = 0.723990943969, h_5 = 0.00213077283313,$ $h_{10} = -0.00152194819409, h_{11} = 0.000272756227374,$ $h_6 = 0.000430256524561, h_9 = 0.000792486608343,$ $h_3 = -0.0296905640742, h_4 = -0.00149103220677,$ $h_8 = 0.0102221961051, h_7 = -0.0166384351838,$ $h_1 = 0.307684470610, h_2 = -0.00161114196926,$ $h_6 = -0.0289420080304, h_5 = 0.0762004511868,$ $h_1 = 0.572231897004, h_4 = 0.0442150288434,$ $h_3 = -0.205877054025, h_2 = -0.0381855138549$ } |
| 3 | 11 | 1 | 1 | No solutions |

Figure 4.4: Length 18 modified filters

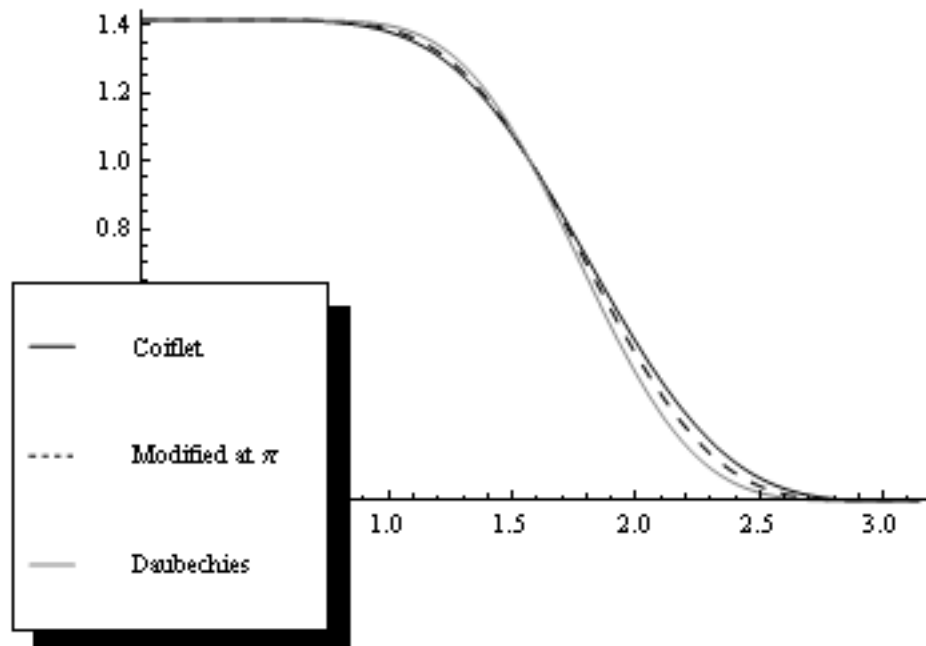


Figure 4.5: Modulus comparison of Daubechies length 12, Coiflet length 12, and modified length 12 ($K = 2, c = 3$) filters

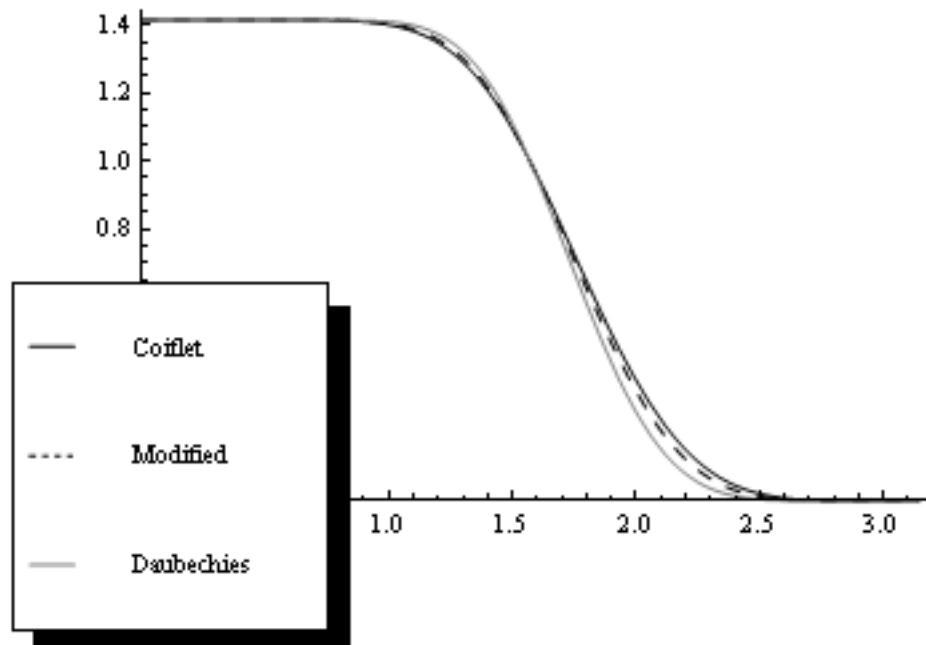


Figure 4.6: Modulus comparison of Daubechies length 18, Coiflet length 18, and modified length 18 ($K = 3, c = 5$) filters

5 CONCLUSION

5.1 Future Research and Study

There are many avenues that can be explored for future research on this topic. For instance, consider the following:

- How could we alter, or what extra conditions could we impose on the Daubechies filter construction, in instances with large amounts of wrapping, so that we obtain real-valued filter coefficients?
- Can we prove the conjecture stated in Section 4.4? Namely, we would like to show that if our system has t degrees of freedom (from wrapping), then we can add at most $\lfloor \frac{t}{2} \rfloor$ derivative conditions to the modified system.
- In our discussions, the length of a Coiflet filter is restricted to multiples of six. In [4] and [19], there are Coiflet constructions for length $6K-2$ and $6K+2$ filters, respectively. Would the modified construction presented in Section 4.4 work equally well for the Coiflet filters of different length?

In this thesis, we have discussed only orthogonal filters. Since the construction of these filters requires a large number of orthogonality conditions, there are not many degrees of freedom remaining in our system to better suit the application we are working towards. There are however many other types of filters that have been constructed, including those called biorthogonal. Biorthogonal filters are constructed so that their corresponding DWT is invertible, but not orthogonal [6]. These filters can be extremely useful in image processing applications. Actually, one of the most powerful and frequently used image compression tools is called JPEG2000, and it uses biorthogonal filters in its wavelet transforms. For further reading on the construction of biorthogonal filters, see [6], and for further reading on JPEG2000 and its applications, see [5].

The work done in this thesis focuses solely on “discrete” wavelet transforms, in which we work only with discrete sets of data. On the other hand, classical wavelet analysis deals with data of a continuous nature, and continues to be the focus of numerous research projects. The interested reader should see [10], [2], or [21].

5.2 Mathematica Program for Construction of Daubechies Filters

In this section, we provide the *Mathematica* [22] code for finding the Daubechies Filters.

```

Start of code
-----
Daub[L_]:=Module[{H,w,k,orthogsquare,orthog,m,i,
remainingorthog, lowpasszero, lowpasspi, derivspi,
derivpi, Solutions, R, n, P,z,zroot,mytable,
mytable2},
(* Defines a module and all of its local variables *)

H[w_]:=Sum[Subscript[h, k]*E^(I*k*w), {k,0,L}];
(* The Fourier series for a Daubechies filter
of length L+1 *)

orthogsquare=Sum[Subscript[h, k]^2, {k,0,L}]==1;
(* The first orthonormality condition from Proposition 2.2.1 *)

orthog[m_]=
Sum[Subscript[h, k]Subscript[h, k-2m],{k,2m,L}]==0/.{ m->i};
remainingorthog=Table[orthog[m],{i,1,(L-1)/2}];
(* The next (L-1)/2 orthonormality conditions
from Proposition 2.2.1 *)

lowpasszero=H[0]==Sqrt[2];
(* The lowpass condition H(0)=Sqrt[2] *)

lowpasspi=H[Pi]==0;

```

```

(* The lowpass condition  $H(\pi)=0$  *)

derivpi[m_]=D[H[w],{w,m}]==0/.{m->i};
derivspi=Table[derivpi[m],{i,1,(L-1)/2}]/.{w->Pi};
(* The  $(L-1)/2$  derivative conditions at  $w=\pi$  *)

Solutions=NSolve[Join[{orthogsquare},remainingorthog,
{lowpasszero},{lowpasspi},derivspi],10];
(* The complete set of solutions to the system of
equations (above) which define Daubechies filters *)

For[{n=1;R={}};,n<=Length[Solutions],n++,
If[MatchQ[Solutions[[n,1,2]],_Real], AppendTo[R,n]]];
(* Makes a list called 'R' which contains the index
of the real solutions in the set 'Solutions' *)

P[z_]:=Sum[Subscript[h, k]*z^k,{k,0,L}];
(* Defines a polynomial  $P(z)$  which is equal to  $H(w)$ 
if substituting  $z=e^{iw}$  *)

Do[Subscript[P, k]=P[z]/.Solutions[[R[[k]]]],{k,1,Length[R]}];
(* Substitutes each of the real solutions from 'Solutions'
into  $P(z)$  and names each one as a new equation *)

Do[Subscript[zroot, k]=Solve[Subscript[P, k]==0,z],
{k,1,Length[R]}];
(* Assigns a list 'Subscript[zroot, k]' of solutions for each
of the equations from the previous step *)

mytable=Table[Abs[z]/.Subscript[zroot, k][[j]],
{k,1,Length[R]},{j,1,L}];
(* Creates a table of  $|z|$  for each list of solutions

```

found in the previous step *)

```
mytable2=Table[Sort[mytable[[k]],{k,1,Length[R]}];
```

```
(* Sorts each of the lists in 'mytable' so that the  
values of |z| are listed in increasing order *)
```

```
For[k=1, k<=Length[R],k++,
```

```
If[mytable2[[k,1]]>=1, Break[]];
```

```
(* Finds the one (see [8]) list in 'mytable2' which has  
all entries greater than 1 *)
```

```
Solutions[[R[[k]]]]
```

```
(*Returns the Daubechies filter. That is, the corresponding  
solution to that one list found in the previous step *)
```

End of code

References

- [1] G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms, *Communications on Pure and Applied Mathematics* **44** (1991), 141-183.
- [2] A. Boggess and F. J. Narcowich, *A First Course in Wavelets with Fourier Analysis*, Prentice Hall, Upper Saddle River, N.J., 2001.
- [3] C. S. Burrus and R. A. Gopinath, *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, Upper Saddle River, N.J., 1998.
- [4] C. S. Burrus and J. E. Odegard, Coiflet systems and zero moments, *IEEE Transactions on Signal Processing* **46** (1998), no. 3, 761-766.
- [5] C. Christopoulos, A. Skodras, and T. Ebrahimi, The JPEG2000 still image coding system: An overview, *IEEE Transactions on Consumer Electronics* **46** (2000), no. 4, 1103-1127.
- [6] A. Cohen, I. Daubechies, and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics* **45** (1992), no. 5, 485-560.
- [7] J. M. Combes, A. Grossman, and P. Tchamitchian, Eds., *Wavelets, Time-Frequency Methods and Phase Space*, Springer-Verlag, Berlin, 1989.
- [8] I. Daubechies, Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics* **41** (1988), 909-996.
- [9] I. Daubechies, Orthonormal bases of compactly supported wavelets, *SIAM Journal on Mathematical Analysis* **24** (1993), no. 2, 499-519.
- [10] M. W. Frazier, *An Introduction to Wavelets Through Linear Algebra*, Springer-Verlag, New York, N.Y., 1999.

- [11] D. Gabor, Theory of communication, Journal of the Institution of Electrical Engineers, Part III **93** (1946), no. 26, 429-457.
- [12] A. Grossman and J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM Journal on Mathematical Analysis **15** (1984), 723-736.
- [13] A. Haar, Zur theorie der orthogonalen funktionensysteme, Mathematische Annalen **69** (1910), 331-371.
- [14] P. C. J. Hill, Dennis Gabor-Contributions to communication theory & signal processing, EUROCON, 2007. The International Conference on "Computer as a Tool" (2007), 2632-2637.
- [15] Y. Meyer, Ed., *Wavelets and Applications*, Springer-Verlag, Berlin, 1992.
- [16] J. Morlet, Sampling theory and wave propagation, Issues on Acoustic Signal/Image Processing and Recognition, C.H. Chen, Ed., NATO ASI, Springer-Verlag, New York, N.Y., 1983.
- [17] O. Rioul and M. Vetterli, Wavelet and signal processing, IEEE Signal Processing Magazine **8** (1991), no. 4, 14-38.
- [18] M. B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, Eds., *Wavelets and their Applications*, Jones and Bartlett, Boston, M.A., 1992.
- [19] J. Tian and R. O. Wells, Vanishing moments and wavelet approximation, Technical Report Computational Mathematics Lab, Rice University, January 1995.
- [20] P. J. Van Fleet, *Discrete Wavelet Transformations: An Elementary Approach with Applications*, John Wiley & Sons Inc., Hoboken, N.J., 2008.
- [21] D. F. Walnut, *An Introduction to Wavelet Analysis*, Birkhauser, Cambridge, M.A., 2002.
- [22] Wolfram Research, Inc., *Mathematica Edition: Version 6.0*, Champaign, I.L., 2007.