2010

Statistical Idealities and Expected Realities in the Wavelet Techniques Used for Denoising

Eric-Jan D. DeNooyer

University of South Florida

Follow this and additional works at: http://scholarcommons.usf.edu/etd

Part of the American Studies Commons

Scholar Commons Citation


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.
Statistical Idealities and Expected Realities
in the Wavelet Techniques Used for Denoising

by

Eric-Jan D. DeNooyer

A thesis 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 Bénétou, Ph.D.
Arthur A. Danielyan, Ph.D.
Yuncheng You, Ph.D.

Date of Approval:
April 12, 2010

Keywords: expected value, filters, Fourier series, mean square error, wavelet shrinkage

© Copyright 2010, Eric-Jan D. DeNooyer
I owe a great deal of gratitude and thanks to one person in particular, my devoted mother, Greetje H. Roos-Brown. She, more than any other person in my life, has given me a great deal of love and patience. She took the time to understand me, when I needed to be understood. She remained faithful to my potential, even when my progress in life seemed uncertain. I also need to give her a tremendous amount of credit for being one of the most emotionally honest and principled people I know. It is by these qualities, I believe, that her children grew up to become the self-respecting individuals they are today. Most importantly, it allowed me to find the faith that 'honesty' and 'goodness' is something that truly can be believed in. Through all of this, I've become a stronger and more patient person. One who is better able to endure those difficult times. Mom, I thank you, and I love you.

I want to give a special thanks to my advisor, Dr. Catherine Bénéeau. Her patient support and help, as we worked together on this thesis, will always be greatly appreciated. I will take from this experience one important value that she has consistently reinforced throughout: always endeavor to truly understand the depth of what we were studying. Dr. Bénéeau, thank you for making me think more critically.

I am also grateful for all the enjoyable learning experiences that I’ve had with the mathematics professors here at the University of South Florida (USF). I took my first math class here as a non-degree seeking student back in 1995. At that time, I was refreshing my math skills that I had not practiced for six years (since receiving my undergraduate degree in Mechanical Engineering from Rensselaer Polytechnic Institute in Troy, New York). The class that I took back then was Calculus I, with Dr. Yuncheng You. Dr. You suggested to me, already then, that I should think about working towards a
career in teaching mathematics for a community college. After finishing my non-degree goals at USF in 1996, it would not be for another nine years that would I returned to USF, in 2005, ready to see the wisdom of Dr. You's vision through. Upon my return, the first class that I would take would be *Bridge to Abstract Mathematics*, with Dr. Arthur A. Danielyan. It seems quite fateful that I would invite these two professors to be committee members on my thesis. I thank them kindly for their willingness to give their valuable time towards this last step in helping me realize an important professional accomplishment.

Finally, I would like to thank Dr. Anya Sebastien, and the rest of the administration at St. Petersburg College's Seminole Campus in Seminole, Florida, for offering me that first opportunity to teach preparatory mathematics in a college environment back in 2004. It opened the door for a renewed chance at achieving something I truly can be proud of.
# Table of Contents

List of Tables iii

List of Figures iv

Abstract vi

1. Preliminaries: Mathematical Spaces, Transformations, and the Fourier Series 1
   1.1 Mathematical Spaces 1
   1.2 Transforming Data 3
   1.3 The Fourier Series 5

2. Wavelets and Their Origins 14
   2.1 Alfréd Haar, and the Theory of the Systems of Orthogonal Functions 14
   2.2 Jean Morlet: Looking for Oil, and Finding Wavelets 15
   2.3 Yves Meyer & Stéphane Mallat: Fostering the Father of a Theory 24
   2.4 Wavelets as Filters: An Introduction to Signal Processing 27

3. The Statistics of Noise, Error, & Estimation 41
   3.1 Statistics: The Essentials 41
   3.2 Noise: A Statistical Characterization 53
   3.3 Estimation: Decisions that Consider the Statistical Errors of Loss & Risk 55
4. Denoising, Thresholding, and an Idealistic Oracle 62
   4.1 Denoising: An Overview. 62
   4.2 Lowpass Filtering: Muffling the Noise. 63
   4.3 Highpass Filtering: The Highlights, and Less Noise too. 66
   4.4 Wavelet Shrinkage, Part I: Of Ideals, Oracles, & Great Expectations. 67
   4.5 Wavelet Shrinkage, Part II: A Tale of Two Errors. 86
   4.6 Illustrative Example: Denoising a Corrupted Sine Function. 92
   4.7 Summary & Conclusions. 101

References 103

Bibliography 105

Appendices 109
   Appendix A: Selected Mathematica Code 110
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Wavelet shrinkage thresholding functions.</td>
<td>69</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of <em>ideal</em> and <em>non-ideal</em> thresholding decisions.</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>MSE calculations for various individual signal strengths, $g_i$.</td>
<td>77</td>
</tr>
</tbody>
</table>
List of Figures

1.1 A graph of the sawtooth function, a periodic linear function ($p = 2$).
1.2 The Fourier series approximations of the sawtooth function ($n = 1, 3, 5$).
2.1 The wavelet series approximations of the sawtooth function ($n = -1, 0, 1$).
2.2 Modulus plots of the Haar lowpass and highpass filters.
2.3 Comparing an original picture to some of its lowpass and highpass data.
2.4 Complete image before and after a Haar wavelet transformation.
3.1 Probability distribution function of the standard normal distribution.
3.2 Linear-log plots of some colored noise spectral power densities.
4.1 Noise level reductions of the Haar lowpass filter.
4.2 Noise before and after a Haar wavelet transformation.
4.3 $\text{MSE}(\hat{g}_i)$ from ideal thresholding, normalized to noise level $\sigma$.
4.4 Signal vector $g_i = 1.5\sigma$ overlaid with a distribution of noisy observations $y_i$.
4.5 Comparison of expected $\text{MSE}(\hat{g}_i)$ to ideal $\text{MSE}(\hat{g}_i)$, thresholding at $\lambda = \sigma$.
4.6 Effects on expected $\text{MSE}(\hat{g}_i)$ when changing threshold: $\lambda = 0\sigma, 0.8\sigma, 1.0\sigma, 1.20\sigma, \infty\sigma$.
4.7 Comparison of two differently defined oracles.
4.8 Comparison of oracle$_2$’s ideal $\text{MSE}(\hat{g}_i)$ to oracle$_1$ and previous expectations.
4.9 Oracle$_2$ overlaid with the expectations of a hard thresholding rule, $\lambda = \frac{1}{2}$.
4.10 Close-up view of oracle$_2$, hard thresholding, and two $g_i$’s, $2g_j < \lambda$ and $2g_k > \lambda$. 
4.11 Effects of an increasing $\lambda$ on different strengths of $g_i$'s.

4.12 Close up view of Figure 4.6, showing the effects of different $\lambda$'s on $g_i$.

4.13 The highpass data of $\theta_j$ and $\theta_k$ will be thresholded by the oracle.

4.14 The highpass data of $\theta_j$ and $\theta_k$ will not be thresholded by the oracle.

4.15 $y = \sin x$, $0 \leq x \leq 2\pi$.

4.16 PDF of the sine function

4.17 PDFs of various noisy sine functions.

4.18 The denoising of a corrupted sine function, before and after.

4.19 Finding the best $\lambda$ for denoising a corrupted sine function with noise $\sigma$.

4.20 The lowpass and highpass data of a clean sine function.

4.21 Plots of the MSE for various levels of noise $\sigma$, as $\lambda$ varies from 0 to $5\sigma$.

4.22 PDF of the cosine function. The expected $\hat{\text{MSE}}(\tilde{g}_i)$ of the highpass data.
In the field of signal processing, one of the underlying enemies in obtaining a good quality signal is noise. The most common examples of signals that can be corrupted by noise are images and audio signals. Since the early 1980's, a time when wavelet transformations became a modernly defined tool, statistical techniques have been incorporated into processes that use wavelets with the goal of maximizing signal-to-noise ratios. We provide a brief history of wavelet theory, going back to Alfréd Haar's 1909 dissertation on orthogonal functions, as well as its important relationship to the earlier work of Joseph Fourier (circa 1801), which brought about that famous mathematical transformation, the Fourier series. We demonstrate how wavelet theory can be used to reconstruct an analyzed function, ergo, that it can be used to analyze and reconstruct images and audio signals as well. Then, in order to ground the understanding of the application of wavelets to the science of denoising, we discuss some important concepts from statistics. From all of these, we introduce the subject of wavelet shrinkage, a technique that combines wavelets and statistics into a "thresholding" scheme that effectively reduces noise without doing too much damage to the desired signal. Subsequently, we discuss how the effectiveness of these techniques are measured, both in the ideal sense and in the expected sense. We then look at an illustrative example in the application of one technique. Finally, we analyze this example more generally, in accordance with the underlying theory, and make some conclusions as to when wavelets are an effective technique in increasing a signal-to-noise ratio.
Chapter One

Preliminaries: Mathematical Spaces, Transformations, and the Fourier Series

1.1 Mathematical Spaces

The most basic notion of a space comes from our three dimensional view of the world we live in. It is from these views that we can begin to understand that a space, in the mathematical sense, is not just a loose set of objects, but a set of objects with structure. One can construct a description of a very simple space by using the rudimentary Euclidean objects of points. Beyond this it is quite impressive to realize the sheer number and complexities of the many different spaces that can be imagined, or rather, discovered. So we quickly move through the hierarchies, and head towards a mathematical space where the objects contained in it are 'functions'.

We can start off structuring a set of k points by ordering them under the Cartesian coordinate system. These points can be equivalently considered to be vectors, i.e. \( \mathbf{v} = (v_1, v_2, \ldots, v_k) \), each \( v_j \in \mathbb{R} \). The familiar operations of vector addition and scalar multiplication are defined, which gives us a vector space over \( \mathbb{R} \) (equivalently, a linear space). We also define the important inner product operation.

**Definition 1.1.** Let \( \mathbf{v} = [v_1, v_2, \ldots, v_k]^T \) and \( \mathbf{w} = [w_1, w_2, \ldots, w_k]^T \) be two vectors in \( \mathbb{R}^k \). The inner product \( \langle \mathbf{v}, \mathbf{w} \rangle \) is defined by

\[
\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v} \cdot \mathbf{w} = \sum_{j=1}^{k} v_j w_j.
\]

With this inner product we now have an inner product space. From here, we can define the idea of a norm for that space.
**Definition 1.2.** Let \( v \) be a vector in \( \mathbb{R}^k \). The *norm* of \( v \) is given by

\[
|v| = (v \cdot v)^{\frac{1}{2}} = \left( \sum_{j=1}^{k} v_j^2 \right)^{\frac{1}{2}}
\]

With this norm we now have the classic *Euclidean space*. Further, this norm allows us to define the concept of distance between two points, \(|v - w|\), giving us the added characterization of qualifying a Euclidean space as a *metric space*.

We now move on from using simple points as our objects and work our way into including functions as our objects. To evolve a structured space for functions, we need to introduce the concept of a *\( \sigma \)-algebra*.

**Definition 1.3.** A collection \( \mathcal{M} \) of subsets of a set \( X \) is called a *\( \sigma \)-algebra* in \( X \) if \( \mathcal{M} \) has the following properties [18, p. 8]:

(i) \( X \in \mathcal{M} \),

(ii) If \( A \in \mathcal{M} \), then \( A^C \in \mathcal{M} \), where \( A^C \) is the complement of \( A \) relative to \( X \),

(iii) If \( A = \bigcup_{n=1}^{\infty} A_n \) and if \( A_n \in \mathcal{M} \) for \( n = 1, 2, 3, \ldots \), then \( A \in \mathcal{M} \).

If \( \mathcal{M} \) is a *\( \sigma \)-algebra* of subsets of \( X \), then this pair \((X, \mathcal{M})\) represents a *measurable space*. If a subset \( A \) of \( X \) is also an element in the *\( \sigma \)-algebra* \( \mathcal{M} \), then \( A \) is called a *measurable set*. If a function \( f \) is a mapping from a measurable space \( X \) into a measurable space \( Y \), such that the pre-image \( f^{-1}(V) \) is a measurable set in \( X \) for every measurable set \( V \) in \( Y \), then \( f \) is called a *measurable function*.

These conditions above will give us a space that is measurable, but how do we measure the size of sets in \( \mathcal{M} \)? We next define a special function, we call it a *measure*, that will relate a nonnegative number to any measurable set. That related number will indicate the size of the measured set.
Definition 1.4. A measure is a nonnegative function \( \mu \) defined [18, p. 16] on a \( \sigma \)-algebra \( \mathcal{M} \) satisfying \( \mu(\emptyset) = 0 \) and

\[
\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i)
\]

for any sequence \( \{A_i\} \) of disjoint measurable sets in \( \mathcal{M} \).

The property of \( \mu \) in (1) is referred to as countable additivity. If the \( \sigma \)-algebra \( \mathcal{M} \) has such a measure \( \mu \) defined on it, then the triple \((X, \mathcal{M}, \mu)\) will be called a measure space.

In a way that is analogous to how we defined the inner product between two vectors, we can also define an inner product between two measurable functions:

\[
\langle f, g \rangle = \int_X fg \, d\mu.
\]

Given this inner product, we can now define [18, p. 65] an \( L^2 \)-norm of a measurable function \( f \) as:

\[
\|f\|_2 = \left\{ \int_X |f|^2 \, d\mu \right\}^{1/2}, \text{ provided that } \|f\|_2 < \infty.
\]

This norm gives us the function space \( L^2(X) \). This space is important to our discussion for denoising since the statistical measures that we will be using later are defined in terms of the \( L^2 \) metrics.

1.2 Transforming Data

Transformations stand out for their ability to preserve a problem's underlying structure, while at the same time simplifying that problem's analysis and manipulation. One example of a simplification would be transform the graphic representation of logarithmically dependent data, from a linear scale to a logarithmic scale. In such an example we are essentially changing the domain of the graph. For a base-10 logarithm, that would mean mapping 1 to 0 (i.e., \( 10^0 \)), 10 to 1 \((10^1)\), 100 to 2 \((10^2)\), etc. This really
points to the central purpose of transformations, and that is to represent a particular problem in the most appropriate basis.

We don't have to limit ourselves to just bases for exponents, we also have the bases for vector spaces.

**Definition 1.5.** Let \( V \) be a linear space, and \( f_1, f_2, \ldots, f_n \in V \). If \( f_1, f_2, \ldots, f_n \) span \( V \), (meaning every \( f \) in \( V \) can be expressed as a linear combination of \( f_1, f_2, \ldots, f_n \)), and if \( f_1, f_2, \ldots, f_n \) are linearly independent (meaning the equation \( c_1f_1 + c_2f_2 + \ldots + c_nf_n = 0 \) is possible only if \( c_1 = c_2 = \ldots = c_n = 0 \)), then we can say that the elements \( f_1, f_2, \ldots, f_n \) form a basis for \( V \) [1, pp. 157-158].

With such a basis we can represent every \( f \in V \) uniquely as a linear combination \( f = c_1f_1 + c_2f_2 + \ldots + c_nf_n \). The coefficients \( c_1, c_2, \ldots, c_n \) are called the coordinates of \( f \) with respect to the basis \( \mathfrak{B} = (f_1, f_2, \ldots, f_n) \). The vector \( [c_1 \ c_2 \ldots \ c_n]^T \) in \( \mathbb{R}^n \) is called the \( \mathfrak{B} \)-coordinate vector of \( f \), denoted \( [f]_\mathfrak{B} \). A transformation \( T \) that transforms any \( f \) into its \( \mathfrak{B} \)-coordinate vector, \( T(f) = [c_1 \ c_2 \ldots \ c_n]^T \), is called a \( \mathfrak{B} \)-coordinate transformation, \( T : V \rightarrow \mathbb{R}^n \). The \( \mathfrak{B} \)-coordinate transformation is invertible: \( T^{-1}[c_1 \ c_2 \ldots \ c_n]^T = c_1f_1 + c_2f_2 + \ldots + c_nf_n = f \) [1, pp. 157-158]. So we can recover our original function's representation by an inverse transformation. If our basis is orthonormal (meaning \( \|f_i\| = 1 \) for all \( i \), and \( f_i \cdot f_j = 1 \) if \( i = j \), and 0 otherwise [1, p. 187]), then our transformation will preserve the magnitude of the original function (e.g., length, energy).

For the purpose of this discussion, when we process a signal, we should look upon our signal as an original function to be transformed. If we think about the familiar ideas of digitally taking a picture, or recording audio, we are, in a sense, transforming a real-world representation into a numerical representation (our eyes and brain do something similar). Once we've recorded our signal, we hopefully have the best representation of the signal possible. Sometimes, though, maybe because of noise, we don't get the best
signal. That is where the mathematics and statistics of transformations and data manipulation can make measurable improvements.

The process of taking our original function (our signal) and transforming its representation from one domain into another is called the analysis of the signal. Its complement, the process of taking the function back into its original domain by inverse transformation, is called the synthesis of the signal.

A particularly famous example of analyzing and synthesizing functions through mathematical transformations is the Fourier series. We will make a formal presentation of this method in the next section. Before we do, we would like to briefly conclude this section with a discussion to how the Fourier series makes things easier for us. The Fourier series has the ability to take an arbitrarily complicated function and transform its representation into sums of much simpler functions. Additionally, it will be shown at the end of Chapter 2, the Fourier transforms allow us to equivalently replace the complicated operation of convolution in one domain, with the simple operation of multiplication in the other. It is by such mathematical transformations that we can accomplish beneficial mathematical manipulation of data that would otherwise be tremendously difficult, if not, impossible.

1.3 The Fourier Series

Every text investigated for this thesis invariably precedes the exposition of wavelets with a discussion on the Fourier series. Fourier series is part of the family of transformations of a similar name, the Fourier transforms. This proper name, "Fourier", gives honor to the French scholar Joseph Fourier (1768 - 1830), who gets much of the credit for starting what would eventually develop into harmonic analysis and functional analysis [10, p. 1].

The actual focus of Fourier's work that would bring about his series was 'the propagation of heat'. He began his work on this subject in 1801. His first attempt to publish a monograph on the matter was rejected in 1807. It would not be until 1822 that
Fourier would get a more refined work published under the title "Théorie analytique de la chaleur" (translated, "Analytical theory of heat"). What Fourier presented eventually became the method of decomposing periodic phenomena into a series of harmonic (oscillatory) terms.

Before we present the formal definition of a Fourier series, some preliminary definitions and proofs need to be stated.

We have already mentioned that Fourier series are helpful to describe phenomena of a periodic nature. We now define the notion of a periodic function.

**Definition 1.6.** A complex-valued function \( f \) defined on \( \mathbb{R} \) is said to be a *periodic function* with period \( p \) if

\[
f(x + p) = f(x), \quad \forall x \in \mathbb{R}.
\]

The best examples of a periodic function are the trigonometric functions of *sine* and *cosine*. We now define the concept of a trigonometric polynomial.

**Definition 1.7.** A *trigonometric polynomial* is a function that can be written as

\[
f(x) = a_0 + \sum_{n=1}^{N}(a_n \cos nx + b_n \sin nx),
\]

where \( x \) is real and \( a_0, \ldots, a_N, b_1, \ldots, b_N \) are complex [17, p. 185].

We now quickly refer to a famous result from Swiss mathematician Leonhard Euler (1707 - 1783). Euler's formula, \( e^{ix} = \cos x + i \sin x \), allows us to transform a trigonometric polynomial into a finite sum of complex exponential functions. The formula itself will not be proved here, but it will be used just below to give the some
trigonometric identities. Let $x$ be a real number, $i$ the imaginary unit, and $e$ the base of the natural logarithm. Then [17, p. 182]

\[
\cos x = \frac{e^{ix} + e^{-ix}}{2} \quad \text{and} \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i}.
\]

These identities allow us to rewrite a trigonometric polynomial into the more convenient form of

\[
f(x) = \sum_{n=-N}^{N} c_ne^{inx}.
\]

It should be obvious that these polynomials are periodic functions with a period of $2\pi$.

Now consider the integral

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{inx} e^{-inx} \, dx,
\]

where $m$ and $n$ are integers. Using the additive rule of exponents, and then applying Euler's formula to substitute out the exponential functions, we get

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{inx} e^{-inx} \, dx = \frac{1}{2\pi} \left( \int_{-\pi}^{\pi} \cos((m-n)x) \, dx + i \int_{-\pi}^{\pi} \sin((m-n)x) \, dx \right).
\]

Considering two cases, $m-n = 0$ and $m-n \neq 0$, the integration shows us that

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(m-n)x} \, dx = \begin{cases} 
1 & \text{if } m = n \\
0 & \text{if } m \neq n.
\end{cases}
\]

Next, if we multiply both sides of (2) by $e^{-inx}$, and then integrate both sides, we get

\[
\int_{-\pi}^{\pi} f(x)e^{-inx} \, dx = \sum_{n=-N}^{N} c_ne^{inx} e^{-inx} \, dx.
\]

Interchanging the integral and sum we then have

\[
\int_{-\pi}^{\pi} f(x)e^{-inx} \, dx = \sum_{n=-N}^{N} c_ne^{i(m-n)x} \, dx.
\]

Using the previous result from (3) gives us the opportunity to simplify (4) into

\[
\int_{-\pi}^{\pi} f(x)e^{-inx} \, dx = c_n \int_{-\pi}^{\pi} e^{i(0)x} \, dx = c_n \cdot 2\pi,
\]

allowing us to conclude that
\[(5)\]
\[c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)e^{-inx} \, dx\]

for \(|n| \leq N\). If \(|n| > N\), the integral in (5) is 0.

We can slightly adjust the earlier definition of the *inner product* to this result, and state that

\[c_n = \langle f(x), \exp(-inx) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)e^{-inx} \, dx.\]

We do not have to limit ourselves to analyzing functions that only have a period \(p = 2\pi\). We can easily adapt this to work for any periodic function \(g(t)\) with an arbitrary period \(p = 2S\). Let \(x = (\pi t / S)\), and \(f(x) = g(t) = g(Sx/\pi)\). Then \(f\) has period \(2\pi\) and \(t = \pm S\) corresponding to \(x = \pm \pi\). By substitution we get the following:

\[(6)\]
\[c_n = \langle g(t), \exp(-in\pi t / S) \rangle = \frac{1}{2S} \int_{-S}^{S} g(t)e^{-inx/S} \, dt.\]

**Definition 1.8.** A *trigonometric series* is a series of the form

\[(7)\]
\[g(t) = \sum_{n=-\infty}^{\infty} c_n e^{inx/S},\]

where \(t\) is a real number [17, p.186]. (Note: unless there are assumptions made about the function \(g\), the series may not converge.)

**Definition 1.9.** If \(g\) is an integrable function on \([-S, S]\), the numbers \(c_n\) defined by (6) for all integers \(n\) are called the *Fourier coefficients* of \(g\).

**Definition 1.10.** The series presented in (7) formed with the Fourier coefficients defined in (6) is called the *Fourier series* of \(g\).
Together, the two formulas presented in (6) and (7) respectively represent the complementary aspects of the *analysis* and *synthesis* of a function. They are the essence to understanding the Fourier series.

If a Fourier *analysis* of a function $g$ is applicable, then we can *synthesize* the same function from its Fourier series (meaning the series converges). It should be mentioned that not all functions surrender to such an analysis. First, the function in question needs to be integrable under some definition of integrability (Riemann, Lebesgue, etc.). And second, despite a function $g$ being integrable, the convergence of its Fourier series is never a certainty. In any regard, for the purposes of this brief introduction, it will be enough to say that $g$ is well-behaved if it is at least a piecewise continuous function with a finite number of jump discontinuities in the interval $[-S, S]$ [21, p. 110].

An important aspect to note about the Fourier series is that it provides us with a set of complex exponential equations $\{e_n = \sqrt{p}^{-1}\exp(i2\pi p^{-1}t)\}$ that form a complete orthonormal basis for the space of $L^2[-p/2, p/2]$, where $p$ is the period of the periodic function being analyzed. Thus, the analysis of the function, which gives us the Fourier coefficients, is in fact a projection of the given function onto the space spanned by those basis functions, via an inner product integral.

One should also see that this family of orthonormal basis functions are all related to a single complex exponential function, $e^{it}$, via dilations. Simply multiplying the $t$ occurring in the exponent of $e^{it}$ by a factor of $n$ gives us a new basis function of a different frequency, $e^{int}$.

Remember, in the Fourier series, $n$ is an integer. Such a fact is only natural in the analysis of a periodic function, since we want the cycles of our basis functions to be in phase (i.e., in sync) with the period of our analyzed function. However, restricting our analysis to only the integers leaves out an infinite number of frequencies in between. For example, if our domain is time, and our period is one second, then $n = 1$ gives us a basis function working at 1 Hertz, $n = 2$ gives us 2 Hertz, etc. To analyze all the frequencies,
including those in between 1 Hertz and 2 Hertz, we do have the more complete tool, the Fourier transform.

**Definition 1.11.** The Fourier transform of a function $g \in L^1(\mathbb{R})$ is defined by

$$ \hat{g}(\omega) = \langle g(t), e^{i\omega t} \rangle = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} g(t) e^{-i\omega t} dt $$

**Theorem 1.1.** If $\hat{g} \in L^1(\mathbb{R})$ is the Fourier transform of $g \in L^1(\mathbb{R})$, then

$$ g(t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{g}(\omega) e^{i\omega t} d\omega $$

at every point of continuity of $g$ [13, p. 660].

Understandably, the Fourier transform looks quite similar to the Fourier series. This transformation is typically used in the analysis of phenomena that are measured in respect to their relationships with time and frequency. For example, as we experience a signal $g(t)$, we do so in the time domain. If we analyze that signal with (8), we would transform that representation into the frequency domain.

This method, as will be explained later, does have its limitations. So, for this discussion, the theory of Fourier analysis will not be developed any further than this. We will make some important use of what we do have in our next chapter on wavelets.

To conclude this first chapter, we now present a popular function that is commonly used to illustrate an application of the Fourier series. This function will be referred to again as a tie-in to our understanding of wavelets.

**Example 1.1.** Application of the Fourier series to the sawtooth function (also called the sawtooth wave). Let $g$ be a function defined by

$$ g(t) = 2 \left( \frac{t}{p} - \left\lfloor \frac{t+p/2}{p} \right\rfloor \right), $$
where \( \lfloor \cdot \rfloor \) is the floor function. This function is periodic with period \( p \). It ranges between \(-1\) and 1. If we set the period \( p \) to be \( 2\pi \), then it would have the same phase as the sine function. Its graph in the following figure makes it clear why it is called the "sawtooth" function.

![Figure 1.1](image)

**Figure 1.1.** A graph of the sawtooth function, a periodic linear function (\( p = 2 \)).

For the future purposes of this discussion, it will be convenient for us to use the period \( p = 2 \). Then it can be noted that \( S = 1 \), and that we will analyze our sawtooth function over the domain \([-1, 1]\), where it can be defined by \( g(t) = t \). Using that fact we apply (6) to compute the Fourier coefficients of our periodic sawtooth function:

\[
(9) \quad c_n = \frac{1}{2} \int_{-1}^{1} e^{-in\pi t} \, dt = \frac{1}{2} \left( \int_{-1}^{1} t \cos(n\pi t) \, dt - i \int_{-1}^{1} t \sin(n\pi t) \, dt \right).
\]

Using several additional facts, that \( g(t) \) and \( \sin \) are odd functions, that \( \cos \) is an even function, and that our integral is symmetric over the interval \([-1, 1]\), we can simplify (9) to be

\[
(10) \quad c_n = -i \int_{0}^{1} t \sin(n\pi t) \, dt.
\]

Setting \( t = u \) and \( \sin(n\pi t) \, dt = dv \), we use integration by parts, \( uv - \int udv \), to solve (10):

\[
c_n = -i \left( -\frac{t}{n\pi} \cos(n\pi t) \bigg|_{0}^{1} + \frac{1}{n\pi} \int_{0}^{1} \cos(n\pi t) \, dt \right)
\]

\[
c_n = -i \left( -\frac{1}{n\pi} \cos(n\pi) + \frac{1}{(n\pi)^2} \sin(n\pi t) \bigg|_{0}^{1} \right).
\]
where $n$ is a nonzero integer. Since $\sin(n\pi \cdot 1) = 0$ and $\cos(n\pi) = (-1)^n$ for all integers $n$, we get

$$c_n = \left(\frac{(-1)^n}{n\pi}\right)i$$

for all nonzero integers $n$. In determining the coefficient for $n = 0$ we can apply (9) in its complex exponential form and determine that

$$c_0 = \frac{1}{2} \int_{-1}^{1} e^{it} \cdot 1 \, dt = \frac{1}{2} \int_{-1}^{1} t \, dt = 0,$$

giving us the following Fourier coefficients

$$c_n = \begin{cases} 0 & \text{if } n = 0 \\ \left(\frac{(-1)^n}{n\pi}\right)i & \text{if } n \neq 0, \end{cases}$$

which provides us with the following Fourier series for $g$

$$(11) \quad g(t) = \sum_{n=1}^{\infty} \frac{(-1)^n}{n\pi} i \cdot e^{in\pi t}.$$  

The Fourier coefficients, $c_n$, in (11) do not converge absolutely, but Kammler [11, p. 43] demonstrates that this series does converge for $t \in \mathbb{R}$. Knowing that the series does converge allows us to do a little algebraic manipulation of the terms. We group them in such a way that we can simplify this Fourier series using the earlier trigonometric identities:

$$g(t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n\pi} i \cdot e^{in\pi t} = \sum_{n=0}^{\infty} \frac{(-1)^n}{n\pi} \cdot \frac{e^{in\pi t}}{2i} = -\frac{2}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{n} \cdot \frac{e^{in\pi t}}{2i}$$

$$= -\frac{2}{\pi} \left( -\frac{e^{i\pi t} - e^{-i\pi t}}{1 \cdot 2i} + \frac{e^{i2\pi t} - e^{-i2\pi t}}{2 \cdot 2i} - \frac{e^{i3\pi t} - e^{-i3\pi t}}{3 \cdot 2i} + \ldots \right)$$

$$= -\frac{2}{\pi} \left( -\sin(1\pi t) + \sin(2\pi t) - \sin(3\pi t) + \ldots \right)$$

$$(11) \quad g(t) = -\frac{2}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \sin(n\pi t).$$

We now have the simplest expression for the Fourier series of the sawtooth function.
If we define a sequence of partial sums based on (11),

\[ g_n(t) = -2 \sum_{k=1}^{n} \frac{(-1)^k}{k} \sin(k \pi t), \]

we can illustrate (see Figure 1.2) how the given Fourier series converges to the sawtooth function.

![Figure 1.2](image)

(a) Fourier series partial sum, \( n = 1 \).

(b) Fourier series partial sum, \( n = 3 \).

(c) Fourier series partial sum, \( n = 5 \).

**Figure 1.2.** The Fourier series approximations of the sawtooth function \((n = 1, 3, 5)\).

In the next chapter, we will develop the basic theoretical background of wavelets. Chapter 3 will cover some basic statistical concepts necessary for this discussion. Then Chapter 4 will be the crux of the discussion, analyzing the key principals that validate the use of wavelets in denoising corrupted signals. The last section in Chapter 4 will summarize our conclusions.
Chapter 2
Wavelets and Their Origins

2.1 Alfréd Haar, and the Theory of the Systems of Orthogonal Functions

Fourier series aside, researchers trace the origins of wavelets back to Alfréd Haar (1885 - 1933), a Hungarian mathematician who studied under David Hilbert at the University of Göttingen in Germany. In July 1909, he proposed in his "Inauguraldissertation" a sequence of functions as an example of a countable orthonormal system for the space of square-integrable functions on the real line. Similar to the Fourier series, Haar's representation was a series of terms involving coefficients multiplied by their respective basis functions,

\[ f(x) = \sum_{n=0}^{\infty} \langle f, \xi_n \rangle \cdot \xi_n(x). \]

The coefficients are determined by calculating the following inner product integral:

(1) \[ \langle f, \xi_n \rangle = \int f(x) \xi_n(x) \, dx. \]

The integral here is presented in its indefinite form for the moment while we continue to discuss the historical development of the wavelet transform.

Haar's thesis work would later be published in 1910 for the German publication Mathematische Annalen under the title "Zur Theorie der orthogonalen Funktionensysteme," [8]. Translated, it means "On the Theory of the Systems of orthogonal Functions." But nowhere in the article will one find the term "wavelet". That term would come much later, in 1982, out of the research of Jean Morlet (1931 - 2007), a French engineer who developed the wavelet transform to study the layering of sediments in the geophysics of oil exploration.
2.2 Jean Morlet: Looking for Oil, and Finding Wavelets

Jean Morlet's problem was to try to find the influence of each sedimentary layer when reflected acoustic waves generated at the earth's surface were recorded. Since some waves get trapped inside a layer and others do not, the sedimentary influence could be found in the different instantaneous frequencies reflected off the different layers [10, p. 179].

To bridge the development of wavelets from Haar through Morlet one has to mention the work of another Hungarian scholar, Dennis Gabor (1900 - 1979). He had a paper published in 1946, "Theory of Communication," [6] that described a new method of analyzing signals in which time and frequency play symmetrical parts. Gabor was trying to overcome a problem with Fourier analysis. Its inherent problem is that the analysis of signals in the time domain produces results that exist purely in the frequency domain. These frequencies are treated as being constant over time, so a Fourier transform cannot determine the particular point in time at which a particular frequency is occurring. It only determines to what degree a particular frequency exists in the entire global picture of the data. This makes the Fourier series exceptionally weak at detecting discontinuities in data as well as working poorly with phenomena that really do not exhibit regular periodicity.

What Gabor did to try to overcome this was to modulate the signal with a window function before performing the Fourier transform. It effectively adapted the traditional Fourier transform, by placing a recurring window around subsets of the data to be analyzed. Gabor's transform uses a Gaussian function for its windowing function [10]. That technique creates uniform windows where each window is associated with a wave shape of invariant envelope with a carrier of variable frequency [7]. Doing this allows the influence of frequencies occurring outside the window to be excluded from the calculations. Today, Gabor's transform is known as a special case of the "short-time Fourier transform" (STFT). The end result gives us a time-frequency analysis of the signal.
The essence of the underlying problem here, as described by German physicist Werner Heisenberg (1901 - 1976), is the "uncertainty principle." This principle states that certain pairs of physical properties (in our concern, time and frequency) cannot both be known to arbitrary precision. Heisenberg realized this principle in 1926 out of his research on the mathematical foundations of quantum mechanics (he would later be awarded the Nobel Prize for Physics in 1932 for the creation of quantum mechanics).

Initially Morlet used Gabor's Fourier-based transform, but found the results unsatisfactory. What Morlet thought of next was to change what was to be invariant. He decided that it was the wave shape that should be invariant (instead of the wave's envelope), which would give uniform resolution to the entire plane [7]. To accomplish this, he would adapt the sampling rate to the frequency. This effectively creates a changing time-scale producing a dilation of the wave shape. The end result gives us a time-scale analysis of the signal (as opposed to time-frequency). This new idea still respected the constraints imposed by Heisenburg's uncertainty principle, but provided much greater flexibility in balancing them.

So, what Morlet had developed was the first discrete wavelet transform. His initial work was published in 1982 in a two part article titled "Wave propagation and sampling theory" [14 & 15]. However, the mathematical soundness of his creation was still a question to him. To find an answer for this, he worked with Alex Grossman, a theoretical physicist at the Centre de Physique Theorique in Marseille, France. What Grossman and his team at Marseille did from here was to take Morlet's discrete use of dilations and extrapolated them into the continuous realm, where all dilations were possible. That is where theoretical introductions to continuous wavelets begin.
Definition 2.1. A wavelet basis of $L^2(\mathbb{R})$ is a family of functions,

$$\{\psi_{a,b}(t) = \sqrt{a} \cdot \psi(a \cdot t - b) : (a,b) \in \mathbb{R}^2, a > 0\},$$

where $a$ is a dilation parameter defining the scale transformation, $D_a f(x) = \sqrt{a} f(a \cdot x)$, and $b$ is a translation parameter defining the shift (location) transformation, $T_b f(x) = f(x - b)$.

In other words, this family of functions consists of all the translations and re-scales of a single function $\psi$. A colloquial note here about wavelets, $\psi_{1,0} = \psi$ is called the mother wavelet, and all other $\psi_{a,b}$'s are called the daughter wavelets.

You should notice that the wavelet's "translation parameter" mentioned above is something that the Fourier series does not have. This highlights an important difference between using wavelets as a basis function versus the complex exponentials (of the Fourier series). Unlike complex exponential functions, wavelet functions do not continue on indefinitely (hence the name "wavelet", as compared to "wave"). Thus, this "localized" wave needs to be translated to other areas of the domain in order to generate the analyzed function. The localization is the key to the power of the wavelets, and it brings about many of their benefits over the Fourier series.

As mentioned, the wavelet representation presented in (1) is one of a "continuous" nature. However, there is much redundancy to be found in an analysis that is done continuously. There is, though, a critical sampling rate that we do need to achieve. The foundation of this sampling rate can be interpreted from The Sampling Theorem, which is stated below without proof. (Note, the literature acknowledges many people with regards to the development of this theorem: D. Gabor, V. A. Kotelnikov, Karl Küpfmüller, Harry Nyquist, H. Raabe, Claude E. Shannon, E. T. Whittaker, and J. M. Whittaker.)

**Theorem 2.1 The Sampling Theorem** [19]. If a function $f(t)$ contains no frequencies higher than $W$ hertz, it is completely determined by giving its ordinates at a series of points $1/(2W)$ seconds apart.
Thus, if we sample our signal at twice the rate of the highest frequency present, then we can completely reconstruct the entire signal. In our discrete wavelet transformations, this necessitates the idea of the dyadic (binary) scale.

Let the dilation parameter be \( a \in \{2^j : j \in \mathbb{Z}\} \), \( D_x f(x) = 2^{j/2} f(2^j \cdot x) \), and the translation parameter \( b \in \{k : k \in \mathbb{Z}\} \), \( T_k f(x) = f(x - k) \) [23, pp. 79-80], then \( f_{j,k}(x) = D_x T_k f(x) = 2^{j/2} f(2^j x - k) \) gives us our discrete wavelet basis:

\[
\left\{ \psi_{j,k}(t) = \sqrt{2^j} \psi(2^j t - k) : (j, k) \in \mathbb{Z}^2 \right\}.
\]

The dyadic scale effectively partitions the domain into intervals of equal length at each level \( j \) [23, p. 115]. Each successive finer scale divides those intervals from the previous (and coarser) scale in half. For example, if one scale's partitions are \( \{..., [0,1), [1,2), ...\} \), then the next finer scale's partition would be \( \{..., [0,\frac{1}{2}), [\frac{1}{2},1), [1,\frac{3}{2}), [\frac{3}{2},2), ...\} \). We can carry these finer and finer scales onto infinity, but for all practical applications we will reach an upper (lower) limit to what we need in detecting the highest (lowest) frequencies.

So, a wavelet basis, together with some coefficients derived from an analyzed function, gives us the wavelet series representation:

\[
f(t) = \sum_{j,k=-\infty}^{\infty} g_{jk} \psi_{j,k}(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_{jk} \psi_{j,k}(t).
\]

Going back to Alfréd Haar’s original wavelet, we can now present a discrete Haar wavelet basis:

\[
\psi_{j,k}(t) = \begin{cases} 
1 & t \in \left[2^{-j} (k), 2^{-j} (k + \frac{1}{2})\right) \\
-1 & t \in \left[2^{-j} (k + \frac{1}{2}), 2^{-j} (k + 1)\right) \\
0 & \text{otherwise}
\end{cases}
\]

Using our original indexing procedure, we have \( a = 2^{j/2} \). So, if \( a = 1 \), then \( j = 0 \). Then, for the purposes of having a convenient index, our mother wavelet will actually be indexed as \( \psi_{0,0} \) instead of as \( \psi_{1,0} \). This wavelet, \( \psi \), is defined on the interval \([0,1)\). Since we now have the interval for which these basis functions are defined, we can determine
the coefficients for Haar's wavelet series. Remember, this is a projection of the analyzed function onto the space spanned by this orthonormal basis. We calculate the inner product integral of \( f \) and \( \psi_{j,k} \):

\[
g_{j,k} = \langle f, \psi_{j,k} \rangle = \int_{2^{-j}(k)}^{2^{-j}(k+\frac{1}{2})} f(x) \psi_{j,k}(x) \, dx = \sqrt{2^j} \left[ \int_{2^{-j}(k)}^{2^{-j}(k+\frac{1}{2})} f(x) \, dx + \int_{2^{-j}(k+\frac{1}{2})}^{2^{-j}(k+1)} -f(x) \, dx \right].
\]

This equation should be recognizable as Haar's representation shown in (1) at the beginning of this chapter. Then by taking \( F \) as the anti-derivative (primitive) of \( f \) we get

\[
g_{j,k} = \sqrt{2^j} \left[ \left( F \left( 2^{-j} \left( k + \frac{1}{2} \right) \right) - F \left( 2^{-j} (k) \right) \right) - \left[ F \left( 2^{-j} (k + 1) \right) - F \left( 2^{-j} \left( k + \frac{1}{2} \right) \right) \right] \right] = \sqrt{2^j} \left[ 2F \left( 2^{-j} \left( k + \frac{1}{2} \right) \right) - F \left( 2^{-j} (k) \right) - F \left( 2^{-j} (k + 1) \right) \right].
\]

And so we can now analyze a function with respect to the Haar wavelet series.

So then, let's apply this new technique to our earlier example from the first chapter of our discussion, the sawtooth function.

**Example 2.1.** Let \( f(t) = t - 2n \) for \( t \in [2n-1, 2n+1) \), \( n \in \mathbb{Z} \). Without loss of generality, we will restrict our analysis to the interval \([-1, 1)\), where \( F(t) = t^2/2 \). Further, it should be noted that our integrals above need to be applied upon intervals where the function is continuous. So, since the sawtooth function has discontinuities at \((..., -3, -1, 1, 3, ...)\), we do need to break up any analyzed interval that includes any discontinuities. This would occur for any \( j < 0 \). Before we go any further about this issue, we can at least determine the coefficients of the analyzed sawtooth function for \( j \geq 0 \):

\[
g_{j,k} = \sqrt{2^j} \left[ 2 \cdot \frac{\left( (k + \frac{1}{2}) 2^{-j} \right)^2}{2} - \frac{\left( k 2^{-j} \right)^2}{2} - \frac{\left( (k + 1) 2^{-j} \right)^2}{2} \right] = \sqrt{2^j} \left[ (k^2 + k + \frac{1}{2}) 2^{-2j} - k^2 2^{-2j-1} - (k^2 + 2k + 1) 2^{-2j-1} \right] = \sqrt{2^j} \left[ 2k^2 2^{-2j-1} + k2^{-2j} + 2^{-2j-2} - k^2 2^{-2j-1} - k^2 2^{-2j-1} - k2^{-2j} - 2^{-2j-1} \right]
\]
You can see that in this case the coefficients do not depend on any translation parameter $k$, it depends only on the dilation parameter $j$. Now, so far, we know this much about the sequence of coefficients, $\{g_{jk}\}_{j\geq 0} = \{-\gamma_1, -\gamma_2, \ldots\}$. Then what about the coefficients for $j < 0$? For $j = -1$ we are working on intervals $[2(k), 2(k+1))$, and so the coefficient $g_{-1,k}$ is determined by the following integral:

$$
g_{-1,k} = \sqrt{2^{-1}} \left[ \int_{2(k)}^{2(k+1/2)} (x-2k) \, dx - \int_{2(k+1/2)}^{2(k+1)} (x-2(k+1)) \, dx \right]
$$

$$
\begin{align*}
&= \frac{1}{\sqrt{2}} \left[ \left( \frac{x^2}{2} - 2kx \right) \bigg|_{2k}^{2k+1/2} - \left( \frac{x^2}{2} - 2(k+1)x \right) \bigg|_{2k+1}^{2k+1/2} \right] \\
&= \frac{1}{\sqrt{2}} \left[ \left( \frac{(2k+1)^2}{2} - 2k(2k+1) \right) - \left( \frac{(2k)^2}{2} - 2k(2k) \right) \right] \\
&\quad - \left( \left( \frac{(2k+2)^2}{2} - 2(k+1)(2k+2) \right) - \left( \frac{(2k+1)^2}{2} - 2(k+1)(2k+1) \right) \right) \\
&= \frac{1}{\sqrt{2}} \left[ \left( (2k^2 + 2k + \gamma_2) - (4k^2 + 2k) \right) - \left( 2k^2 - 4k^2 \right) \right] \\
&\quad - \left( \left( (2k^2 + 4k + 2) - (4k^2 + 8k + 4) \right) - \left( (2k^2 + 2k + \gamma_2) - (4k^2 + 6k + 2) \right) \right) \\
&= \frac{1}{\sqrt{2}} \left[ k^2(2-4-2+4-2+4-2) + k(2-2-4+8+2-6) + (\gamma_2 - 2+4 - \gamma - 2) \right] \\
&= \frac{1}{\sqrt{2}} \left[ k^2(0) + k(0) + (1) \right] = \frac{1}{\sqrt{2}}.
\end{align*}
$$
Next, we go onto \( j = -2 \), where the interval we are working on is \([4(k), 4(k+1))\), and so the coefficient \( g_{-2,k} \) is determined by the following integral:

\[
g_{-2,k} = \sqrt{2^{-2}} \left[ \int_{4(k)}^{4(k+\frac{1}{2})} (x - 2k) \, dx + \int_{4(k+\frac{1}{2})}^{4(k+1)} (x - 2(k+1)) \, dx \\
- \int_{4(k+\frac{1}{2})}^{4(k+1)} (x - 2(k+1)) \, dx - \int_{4(k+1)}^{4(k+\frac{1}{2})} (x - 2(k+2)) \, dx \right]
\]

\[
= \frac{1}{2} \left[ \left( \frac{x^2}{2} - 2kx \right)_{4k+1}^{4k+2} + \left( \frac{x^2}{2} - 2(k+1)x \right)_{4k+1}^{4k+2} \\
- \left( \frac{x^2}{2} - 2(k+1)x \right)_{4k+2}^{4k+3} - \left( \frac{x^2}{2} - 2(k+2)x \right)_{4k+2}^{4k+3} \right]
\]

\[= 0.\]

To keep going lower than \( j = -2 \) will only produce the same result.

So our wavelet series for the sawtooth function can be represented as

\[
f(t) = \sum_{j=-1}^{\infty} \sum_{k=-\infty}^{\infty} 2^{-j} \cdot \begin{cases} 
1 & t \in \left[ 2(k), 2(k + \frac{1}{2}) \right) \\
-1 & t \in \left[ 2(k + \frac{1}{2}), 2(k+1) \right)
\end{cases}
\]

\[+ \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} -2^{-j-2} \cdot \begin{cases} 
1 & t \in \left[ 2^{-j}(k), 2^{-j}(k + \frac{1}{2}) \right) \\
-1 & t \in \left[ 2^{-j}(k + \frac{1}{2}), 2^{-j}(k+1) \right)
\end{cases}\]

As we did similarly with the Fourier series representation of sawtooth at the end of chapter one, we can define a partial sum of this sequence, call it \( f_n(t) \), where we give the infinite summation involving the index \( j \) a finite upper limit of \( n \) instead (note: if \( n < 0 \), then the infinite summation involving \( j \) becomes zero). The next figure illustrates how a Haar wavelet series of the sawtooth function converges.
What if our function was not oscillating around the mean value of zero? What if our sawtooth function is instead ranging between $0$ and $2$ instead of $-1$ to $1$? Then $f$ on the interval $[-1, 1)$ will be defined by $f(t) = t + 1$, and the primitive is $F(t) = \frac{t^2}{2} + t$. We now calculate the coefficients of this new sawtooth function. First for $j \geq 0$:

$$g_{jk} = \sqrt{2^j} \left[ 2 \cdot \left( \frac{\left( \frac{k + \frac{1}{2}}{2} \right)^2}{2} + \left( \frac{k + \frac{1}{2}}{2} \right) 2^{-j} \right) - \left( \frac{(k 2^{-j})^2}{2} + k 2^{-j} \right) - \left( \frac{(k + 1) 2^{-j})^2}{2} + (k + 1) 2^{-j} \right) \right]$$
\[ g_{jk} = \sqrt{2^j} \left[ 2 \cdot \frac{(k + \frac{j}{2}) 2^{-j}}{2} - \frac{(k 2^{-j})^2}{2} - \frac{(k + 1) 2^{-j})^2}{2} \right. \]
\[ \quad + \left( 2 \cdot \frac{(k + \frac{j}{2}) 2^{-j}}{2} - k 2^{-j} - (k + 1) 2^{-j} \right) \]
\[ = -2^{\frac{j-2}{2}} + \sqrt{2^j} \left[ 2 \cdot (k + \frac{j}{2}) 2^{-j} - k 2^{-j} - (k + 1) 2^{-j} \right] \]
\[ = -2^{\frac{j-2}{2}} + \sqrt{2^j} \left[ 2 \cdot 2^{-j} k + 2^{-j} - 1 \cdot 2^{-j} k - 2^{-j} \right] \]
\[ = -2^{\frac{j-2}{2}} + 0 = -2^{\frac{j-2}{2}}. \]

So, these coefficients have not changed. Next, what about for \( j = -1 \)?

\[ g_{-1,k} = \sqrt{2} \int_{2}^{2(k+\frac{j}{2})} ((x+1) - 2k) dx - \int_{2}^{2(k+\frac{j}{2})} ((x+1) - 2(k+1)) dx \]
\[ = \sqrt{2} \int_{2k}^{2k+1} (x - 2k) dx + \int_{2k+1}^{2k+2} (x - 2(k+1)) dx \]
\[ = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \left[ (x)_{2k+1}^{2k+2} - (x)_{2k}^{2k+1} \right] \]
\[ = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \left[ (2k+1)(2k+2) - (2k+2)(2k+1) \right] \]
\[ = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} [0] = \frac{1}{\sqrt{2}}. \]

That coefficient has not changed either. Finally, looking at the coefficient for \( j = -2 \), we will get the same coefficient there as well if the following integral is zero,

\[ \int_{4(k+\frac{j}{2})}^{4(k+\frac{j}{2})} 1 dx + \int_{4(k+\frac{j}{2})}^{4(k+\frac{j}{2})} 1 dx - \int_{4(k+\frac{j}{2})}^{4(k+\frac{j}{2})} 1 dx - \int_{4(k+\frac{j}{2})}^{4(k+\frac{j}{2})} 1 dx \]
\[ = (x)_{4k+1}^{4k+2} + (x)_{4k+1}^{4k+2} - (x)_{4k+2}^{4k+3} - (x)_{4k+3}^{4k+4} \]
\[ = (4k+1 - 4k) + (4k + 2 - 4k - 1) - (4k + 3 - 4k - 2) - (4k + 4 - 4k - 3) \]
\[ = 1 + 1 - 1 - 1 = 0. \]
So that didn't change either, and we can see that this new sawtooth function has the exact same wavelet series representation as our original sawtooth function. Clearly this is not going to be a good approximation of our new function. So, we need something more to work with in order to analyze functions that do not only oscillate around zero.

### 2.3 Yves Meyer & Stéphane Mallat: Fostering the Father of a Theory

This brings us to the next piece of wavelet theory. There is a companion function to the wavelet function called the scaling function. One can think of the scaling function as taking the average of the analyzed function on an interval. It can be looked upon as replacing all wavelets existing at frequencies below its own detail level $j$.

The idea of a scaling function grew out of the work of Yves Meyer, a mathematician at the École Polytechnique in Paris, France (finally a pure mathematician enters the fray), and Stéphane Mallat, who, at the time, was a Ph.D. candidate in electrical engineering in Philadelphia. In 1985 Meyer had postulated the existence of a scaling function in order to complete his theoretical construction of a wavelet basis that could be extended to a multi-dimensional setting. With this development, Mallat soon would connect wavelet theory to similar algorithms already used in his field. These included the pyramidal algorithms of computer vision, the subband coding schemes of signal processing, and the perfect reconstruction quadrature mirror filters. Mallat and Meyer would eventually collaborate to develop the formalism that includes this important idea of a scaling function. That formalism is called a *Multiresolution Analysis* [10, pp. 183-185].

Definition 2.2. A Multiresolution Analysis (MRA) decomposes the entire function space into a sequence of closed subspaces $V_n$, $n \in \mathbb{Z}$, in $L^2(\mathbb{R})$. These subspaces satisfy:

(i) a nested hierarchy, $V_j \subset V_{j+1}$, where the spaces

(a) have a trivial intersection, $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$, and

(b) a dense union in $L^2(\mathbb{R})$, $\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R})$;

(ii) the $V$-spaces are self-similar, $f(t) \in V_j \iff f(2t) \in V_{j+1}$; and

(iii) $V_j$ has an orthonormal basis \( \{ \varphi_{j,k}(t) = \sqrt{2^j} \varphi(2^j t - k) : (j,k) \in \mathbb{Z}^2 \} \).

These functions $\varphi_{j,k}$ are called the scaling functions associated with $V_j$. Since $V_j \subset V_{j+1}$, the function $\varphi_{j,k}(t) \in V_j$ can be represented as a linear combination of functions from $V_{j+1}$,

$$\varphi_{j,k}(t) = \sum_{k \in \mathbb{Z}} h_k \varphi_{j+1,k}(t-k) = \sum_{k \in \mathbb{Z}} h_k \sqrt{2} \varphi_{j,k}(2t-k).$$

for some coefficients $h_k, k \in \mathbb{Z}$ [22, p. 52].

There is an interesting relationship to develop here between the scaling function $\varphi$ and the wavelet function $\psi$. Given a sequence of subspaces that satisfy the properties of an MRA, we can define another subspace based on the differences between two successive subspaces of the MRA, $W_j = V_{j+1} \ominus V_j$. It so happens that the wavelet basis for $L^2(\mathbb{R})$, described earlier in (1), provides for an orthonormal basis of $W_j$ [22, p. 57],

$$\{ \psi_{j,k}(t), j \text{ fixed}, k \in \mathbb{Z} \}.$$

In turn, with $\psi_{j,k}(t) \in W_j \subset V_{j+1}$, we can actually derive the wavelet function $\psi_{j,k}$ from the scaling function $\varphi_{j+1,k}$:

$$\psi_{j,k}(t) = \sum_{k \in \mathbb{Z}} g_k \varphi_{j+1,k}(t-k) = \sum_{k \in \mathbb{Z}} g_k \sqrt{2} \varphi_{j,k}(2t-k).$$

for some coefficients $g_k, k \in \mathbb{Z}$. The scaling function is called the "father wavelet".

For example, in Haar's discrete wavelet basis we define the scaling function to be

$$\varphi_{j,k}(t) = \sqrt{2^j} \begin{cases} 1 & t \in [2^{-j} (k), 2^{-j} (k+1)] \\ 0 & \text{otherwise}. \end{cases}$$
Projecting the analyzed function onto the space generated by this orthonormal basis, again by taking the inner product:

$$\begin{align*}
  h_{jk} &= \int_{2^{-j}(k)}^{2^{-j}(k+1)} f(t) \varphi_{j,k}(t) \, dt = \sqrt{2^j} \int_{2^{-j}(k)}^{2^{-j}(k+1)} f(t) \, dt.
\end{align*}$$

Taking $F$ as the anti-derivative (primitive) of $f$, then

$$h_{jk} = \sqrt{2^j} [F(2^{-j}(k+1)) - F(2^{-j}(k))].$$

We now have a much more convenient theory to work with: a double family of orthogonal projection operators [10, p. 183]. Therefore, an analyzed function can now be synthesized by the following series:

$$f(t) = \sum_{k=-\infty}^{\infty} h_{nk} \varphi_{n,k}(t) + \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} g_{jk} \psi_{j,k}(t),$$

for some coefficients $h_{nk}$ and $g_{jk}$.

We can now go back to that altered sawtooth function example and complete its analysis using our new scaling function. Looking at $j = -2$, we are working on the interval $[4(k), 4(k+1))$. Discontinuities exists here, and so we calculate the coefficient $h_{-2,k}$ by the following integral:

$$\begin{align*}
  h_{-2,k} &= \frac{1}{2} \left[ \int_{4k}^{4k+1} (t - (4k - 1)) \, dt + \int_{4k+1}^{4k+3} (t - (4k + 1)) \, dt \\
  &\quad + \int_{4k+3}^{4k+4} (t - (4k + 3)) \, dt \right] \\
  &= \frac{1}{2} \left[ \int_{4k}^{4k+1} t \, dt + \int_{4k+1}^{4k+3} t \, dt \right].
\end{align*}$$
$$h_{2,k} = \frac{1}{2} \left[ \left( \frac{t^2}{2} - (4k-1)t \right)_{4k}^{4k+1} + \left( \frac{t^2}{2} - (4k+1)t \right)_{4k}^{4k+1} \right]_{4k}^{4k+4}$$

$$= 2.$$  

Applying this to the scaling function representation we get

$$f_{-2}(t) = \sum_{k=-\infty}^{\infty} 2 \cdot \sqrt{2^{-2}} \cdot \begin{cases} 1 & t \in [2^{-j}(k), 2^{-j}(k+1)) \\ 0 & otherwise \end{cases}$$

$$\equiv 1,$$

which is the average value of our altered sawtooth function.

### 2.4 Wavelets as Filters: An Introduction to Signal Processing

We just discussed how the scaling function, $\phi$, effectively takes the "average" of the function over its analyzed interval, $[2^{-j}(k), 2^{-j}(k+1))$. If we take this observation and then look at the companion wavelet function, $\psi$, we should also see that a similar use of averaging is going on, but with a twist. What the wavelet function essentially does is to find the "difference" between the respective average values of the function on its two adjacent subintervals, $[2^{-j}(k), 2^{-j}(k + \frac{1}{2}))$ and $[2^{-j}(k + \frac{1}{2}), 2^{-j}(k + 1))$. These companion ideas of "average" and "difference" will be important concepts to appreciate as we develop our rationale for connecting wavelet theory to filter theory.

So far, the use we've made of the Haar wavelet has really been in the analysis of piecewise continuous functions. We can easily make an analogous jump to discrete applications, and do so now.
To begin, consider a typical discrete sequence of numbers, \( \{x_k\} \). You already know that to find the average of two numbers, \( x_i \) and \( x_j \), you just simply take their sum and divide by two. This is analogous to the scaling function’s averaging effect, and can be written equivalently as

\[
\frac{1}{2} x_i + \frac{1}{2} x_j.
\]

Continuing the analogy, we can relate \( x_i \) and \( x_j \) to the two subintervals of \([2^{-j}(k), 2^{-j}(k + \frac{1}{2}))\) and \([2^{-j}(k + \frac{1}{2}), 2^{-j}(k + 1))\), and then a simple association would suggest that the analogy of the wavelet function onto \( x_i \) and \( x_j \) would look like

\[
\frac{1}{2} x_i - \frac{1}{2} x_j.
\]

So then, what does this offer us?

What we have actually done is to transform the two original numbers \( x_i \) and \( x_j \) into two new numbers, and in doing so we really have not lost any information about the two original numbers. For we can still recover \( x_i \) and \( x_j \) from the two new numbers perfectly:

\[
x_i = \left( \frac{1}{2} x_i + \frac{1}{2} x_j \right) + \left( \frac{1}{2} x_i - \frac{1}{2} x_j \right), \quad \text{and} \quad x_j = \left( \frac{1}{2} x_i + \frac{1}{2} x_j \right) - \left( \frac{1}{2} x_i - \frac{1}{2} x_j \right).
\]

We can express this process equivalently through linear algebra. First, let’s just take a simple look at the initial transformation for the case of a two element vector \([x_i \ x_j]^T\),

\[
\sqrt{2} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \sqrt{2} \begin{bmatrix} \frac{1}{2} x_i + \frac{1}{2} x_j \\ \frac{1}{2} x_i - \frac{1}{2} x_j \end{bmatrix}.
\]

And then look at the inverse transformation that brings us back to the original vector,

\[
\sqrt{2} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \sqrt{2} \begin{bmatrix} \frac{1}{2} x_i + \frac{1}{2} x_j \\ \frac{1}{2} x_i - \frac{1}{2} x_j \end{bmatrix} = \begin{bmatrix} \frac{1}{2} x_i + \frac{1}{2} x_j + \left( \frac{1}{2} x_i + \frac{1}{2} x_j \right) + \left( \frac{1}{2} x_i - \frac{1}{2} x_j \right) \\ \frac{1}{2} x_i + \frac{1}{2} x_j - \left( \frac{1}{2} x_i - \frac{1}{2} x_j \right) \end{bmatrix} = \begin{bmatrix} x_i \\ x_j \end{bmatrix}.
\]

We include the factor \( \sqrt{2} \) in order to preserve energy. What we have here is called the \textit{Haar Transform}. We now define it formally for the simple case of a 2-element vector.
Definition 2.3. Let $\mathbf{x} = [x_1, x_2]^T$ be a 2-element vector. The Haar transform of $\mathbf{x}$ is given [12] by

$$
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} = W
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix},
$$
where $W = \sqrt{2} \cdot
\begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}
\end{bmatrix}$.

What does this transformation of the original numbers offer us? The transformed numbers tell us something about how much the value from $x_i$ to $x_j$ has changed. If $x_i$ and $x_j$ are really close to the same value, then their resulting difference (analogous to the wavelet function) will end up being very close to zero. As for its companion, the average (analogous to the scaling function), this new number will be obviously very close to both of the two original numbers. On the other hand, if there is a large difference between $x_i$ and $x_j$, then, of course, their difference will be a large value, and their average will no longer be near either of the two original numbers. In essence, through the use of these two calculations, we can say that the original numbers are being "filtered" for two opposing qualities: stability versus instability.

If you don't quite see this filtering comparison, then let's think again about the two scenarios described in the preceding paragraph, stability and instability, but only at their respective extremes. First, we consider the two extremes for the case of the averaging filter. If we take two numbers that are exactly the same (i.e. stable), then their resulting average will be the exact same value as well, and thus the one number will pass through the averaging filter unchanged. If we next take two un-equal numbers through the averaging filter, then the further apart they are, the more divergent their average value will be from their two original values. So the averaging filter better preserves the values that have a low rate of change between them, and thus we call it a lowpass filter. On the other hand, when we consider the two extremes for the case of the difference filter, we get something else. When we take the two equal numbers, their resulting difference is zero, thus the difference filter will produce a zero as well. When we take the two un-equal numbers through the difference filter, the further apart those numbers are (i.e.
unstable), the larger the filter's resulting value will be. So the difference filter gives larger values for the higher rates of change, and we thus call it a highpass filter.

There is another way to illustrate these highpass and lowpass filtering concepts, and that is to use the coefficients of the two filtering functions as the coefficients in two respective finite-length Fourier series. First we describe the general representation. Given a finite sequence of coefficients \( \{ h_k \}, \) \( k = 0, 1, 2, \ldots, N, \) we will construct a finite-length Fourier series from these coefficients as follows:

\[
H(\omega) = \sum_{k=0}^{N} h_k e^{ik\omega}.
\]

Then, in the case of the Haar transform, its lowpass filter, whose coefficients are \( \{ \sqrt{2}/2, \sqrt{2}/2 \} \), will give us the following finite-length Fourier series:

\[
H(\omega) = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} e^{j\omega}.
\]

For Haar's highpass filter, those coefficients are \( \{ \sqrt{2}/2, -\sqrt{2}/2 \} \), and gives us

\[
G(\omega) = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} e^{j\omega}.
\]

Remember that these functions are \( 2\pi \) periodic. In Figure 2.2 we plot the modulus of these two functions onto the interval \([-\pi, \pi]\).

![Figure 2.2. Modulus plots of the Haar lowpass and highpass filters.](image)
What these plots show us is that the functions $|H(\omega)|$ and $|G(\omega)|$ are the frequency response functions of the lowpass and highpass filters, respectively. You can see that $|H(\omega)|/\sqrt{2}$ is near 1 when $\omega$ is a low frequency (near 0), and $|G(\omega)|/\sqrt{2}$ is near 1 when $\omega$ is a high frequency (near $-\pi$ or $\pi$). We now formally define the **Haar Transform Filters** [21, p. 166].

**Definition 2.4.** The filter

$$h = (h_0, h_1) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

is a lowpass filter called the **Haar filter**, and

$$g = (g_0, g_1) = \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right)$$

is a highpass filter called the **Haar wavelet filter**.

The filters we’ve looked at here are based on the Haar transform. The coefficients used to create these filters are referred to as *filter taps*. With only a finite number of taps in these examples, such filters are considered to be a **finite impulse response filter** (FIR). The concept of filtering helps us to theoretically connect wavelet theory to the field of signal processing, which is the dominant area for the application of discrete wavelets.

A signal, in general, is a time-varying or spatial varying quantity, such as an audio signal or a photographic image. These signals can be codified into streams and/or arrays of quantified data. Thus, we can look at an arbitrary bi-infinite sequence $\{s_j\}$ as a signal representation $\{\ldots, s_{-2}, s_{-1}, s_0, s_1, s_2, \ldots\}$, or equivalently as an infinite vector $[\ldots s_{-2} \ s_{-1} \ s_0 \ s_1 \ s_2 \ldots]^T$.  

31
Earlier, when we illustrated how the Haar transform operates as a combination of lowpass and highpass filters, we used the trivial example of a two element vector. In order to get a more robust use out of these filters we will need to understand a special binary operation called convolution.

**Definition 2.5. Convolution** [21, p. 128]. Let $h$ and $x$ be two bi-infinite sequences. Then the convolution product $y$ of $h$ and $x$, denoted $h \ast x$, is the bi-infinite sequence $y = h \ast x$, whose $n^{th}$ component is given by

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

The definition requires two bi-infinite sequences. In general, there is no guarantee that this series converges. We will mostly be dealing with FIR filters, and therefore will not be worried about convergence. We can easily make a bi-infinite sequence out of any FIR filter by adding zeros on either side of the coefficients, \{... , 0, 0, $h_0$, $h_1$, ... , $h_N$, 0, 0, ...\}. We can then apply any FIR filter onto a signal \{s_j\} by using the convolution operation. What we will get from that is the following sequence, which represents a filtered signal

$$\{t_j\} = \{... (h_0 s_n + h_1 s_{n-1} + \ldots + h_N s_{n-N}), \ldots, (h_0 s_2 + h_1 s_1 + \ldots + h_N s_{2-N}), \ldots\}$$

You'll notice that the sequence \{s_j\} appears to be processed in reverse order. That comes from how convolution has been defined. When $n = 0$, $x_{n-k} = x_{-k}$, and clearly shows how the bi-infinite sequence \{x_j\} is processed from $+\infty$ to $-\infty$ as $k$ goes from $-\infty$ to $+\infty$.

We can translate this into linear algebra, but we will reverse the taps of \{h_k\} instead of the original signal \{s_j\}:

32
So, we now have a convention for processing a large signal through a filter. We do need to develop this some more (remember there are two filters to deal with), but before we do, there is an important result that characterizes convolution in the Fourier domain. That result is the convolution theorem.

**Theorem 2.2. Convolution Theorem** [21, pp. 133-134]. Let \( h, x, \) and \( y \) be bi-infinite sequences with \( y = h \ast x \). Let \( H(\omega), X(\omega), \) and \( Y(\omega) \) denote the Fourier series of \( h, x, \) and \( y \), respectively. Then

\[
Y(\omega) = H(\omega)X(\omega).
\]

This result allows us to substitute the complicated operation of convolution with a simple multiplication in the Fourier domain. It provides us with an important tool in the design of filters.

Before we conclude this chapter, there is one more aspect of signal processing we need to finalize. The convolution we just explained above was only demonstrated for one filter. You should remember, though, that the Haar transform works as two combined filters, a lowpass and a highpass. We do need them both, and there is a way to combine these filters into a single process using the *block matrix arithmetic* of linear algebra. Let \( \{h_k\} \) be the taps of a lowpass filter, and \( \{g_k\} \) be the taps of a highpass filter. Then the translation of this process into linear algebra will look like this:
In the case of the Haar Transform, and remember that we're reversing the taps, we get a process that looks like this:

\[
\begin{bmatrix}
\vdots & \vdots & \vdots & & \vdots \\
0 & h_L & \cdots & h_1 & h_0 & 0 & 0 & 0 & \cdots \\
0 & 0 & h_L & \cdots & h_1 & h_0 & 0 & 0 & \cdots \\
0 & 0 & 0 & h_L & \cdots & h_1 & h_0 & 0 & \cdots \\
\vdots & \vdots & \vdots & & \vdots \\
0 & g_L & \cdots & g_1 & g_0 & 0 & 0 & 0 & \cdots \\
0 & 0 & g_L & \cdots & g_1 & g_0 & 0 & 0 & \cdots \\
0 & 0 & 0 & g_L & \cdots & g_1 & g_0 & 0 & \cdots \\
\vdots & \vdots & \vdots & & \vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
t_{-1} \\
t_0 \\
t_1 \\
t_2 \\
\vdots \\
s_{-1} \\
s_0 \\
s_1 \\
s_2 \\
\vdots \\
u_{-1} \\
u_0 \\
u_1 \\
u_2 \\
\vdots \\
\end{bmatrix}
= \sqrt{2}
\begin{bmatrix}
\vdots \\
\frac{1}{2} s_{-1} + \frac{1}{2} s_0 \\
\frac{1}{2} s_0 + \frac{1}{2} s_1 \\
\frac{1}{2} s_1 + \frac{1}{2} s_2 \\
\frac{1}{2} s_2 + \frac{1}{2} s_3 \\
\vdots \\
\frac{1}{2} s_{n-1} + \frac{1}{2} s_n \\
\frac{1}{2} s_0 + \frac{1}{2} s_1 \\
\frac{1}{2} s_1 + \frac{1}{2} s_2 \\
\frac{1}{2} s_2 + \frac{1}{2} s_3 \\
\vdots \\
\frac{1}{2} s_{n-1} + \frac{1}{2} s_n \\
\end{bmatrix}
\]
When we think about inverting this transform, remember our goal is for perfect reconstruction of the original signal \( \{s_j\} \). Looking carefully above, it can be noticed that we actually have more information than we really need in order to accomplish this. Observe that for any \( s_j \) above we have \( \frac{1}{2}(s_{j-1} + s_j) \), \( \frac{1}{2}(s_j + s_{j+1}) \), \( \frac{1}{2}(-s_{j-1} + s_j) \), and \( \frac{1}{2}(-s_j + s_{j+1}) \). That means we can reconstruct \( s_j \) in two ways:

\[
\begin{align*}
  s_j &= \frac{1}{2}(s_{j-1} + s_j) + \frac{1}{2}(-s_{j-1} + s_j), \text{ or, } \\
  s_j &= \frac{1}{2}(s_j + s_{j+1}) - \frac{1}{2}(-s_j + s_{j+1}).
\end{align*}
\]

We only need to choose one of them to make this work. That is why we are going to downsample our processing by truncating the matrix. To accomplish the truncation we will get rid of every other row in the Haar transformation matrix:

$$
\begin{pmatrix}
\vdots & \vdots \\
\cdots & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \cdots \\
\vdots & \vdots \\
\end{pmatrix}
= \sqrt{2}
\begin{pmatrix}
\vdots \\
\frac{1}{2}s_{j-1} + \frac{1}{2}s_j \\
\frac{1}{2}s_j + \frac{1}{2}s_{j+1} \\
\vdots \\
\frac{1}{2}s_{j-1} + \frac{1}{2}s_j \\
\frac{1}{2}s_j + \frac{1}{2}s_{j+1} \\
\vdots
\end{pmatrix}
$$

What we have above is the **Haar Wavelet Transformation**, generalized to \( N \) dimensions. We now formalize its definition just below.
Definition 2.6. The Haar Wavelet Transformation [21, p. 166] is given by

\[
W_N = \sqrt{2} \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
-\frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & -\frac{1}{2} & -\frac{1}{2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & -\frac{1}{2}
\end{bmatrix}.
\]

Then, if \( S_N \) represent a one-dimensional signal matrix of length \( N \), where \( N \) is an even positive integer,

\[
V = W_N \cdot S_N
\]

is the transformed signal.

It is important to note here that the resulting matrix has two blocks in it. The upper block is a result of the lowpass filter and is considered to be the lowpass data. The lower block is a result of the highpass filter and is considered to be the highpass data.

The concise mathematical notation to represent this is as follows. Let the one-dimensional signal be represented by the matrix \( S = [\ldots \ s_{-2} \ s_{-1} \ s_0 \ s_1 \ s_2 \ldots]^T \), and let \( W \) be the matrix of the entire wavelet transformation, and \( H \) and \( G \) be the respective lowpass and highpass blocks of \( W \). Then our processed signal matrix, call it \( V \), will be the result of the following block matrix multiplication:

\[
V = W \cdot S = \begin{bmatrix} H \\ G \end{bmatrix} \cdot S = \begin{bmatrix} H \cdot S \\ G \cdot S \end{bmatrix}.
\]
The inverse of our transformation will work as follows:

\[
\begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots \\
\ddots & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
\sqrt{2} \\
\end{bmatrix}
= \begin{bmatrix}
\ddots & \vdots & \vdots & \vdots & \vdots \\
\ddots & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
\frac{1}{2} s_{-1} + \frac{1}{2} s_0 \\
\frac{1}{2} s_1 + \frac{1}{2} s_2 \\
\vdots \\
\frac{1}{2} s_n + \frac{1}{2} s_{n+1} \\
\vdots \\
\end{bmatrix}
\begin{bmatrix}
\sqrt{2} \\
\end{bmatrix}.
\]

We can represent this concisely as follows:

\[
S = W^T \cdot V = \begin{bmatrix} H^T \vert G^T \end{bmatrix} \cdot \begin{bmatrix} H \cdot S \\ G \cdot S \end{bmatrix} = \begin{bmatrix} H^T \cdot H \cdot S + G^T \cdot G \cdot S \end{bmatrix}.
\]

This is all satisfactory for a one-dimensional signal, but what do we do for the case of a two-dimensional signal, like a digital photograph? We could use the wavelet transformation \( W \) on the two dimensional data in just the same way, but it would only filter the data through its columns. It would not filter any of the data through its second dimension, meaning it would not analyze how the data is changing along each rows. We can still accomplish this by including one additional step of matrix multiplication. We only need to multiply the other side of \( S \) by the transpose of \( W \) to get the rows filtered.

We can thus define a 2-dimensional Haar Wavelet Transform.

**Definition 2.7.** Let \( S \) represent a two-dimensional signal matrix, and let \( W \) be the matrix of the entire wavelet transformation. Then the 2-dimensional Haar Wavelet Transform of \( S \) is given by

\[
V = W \cdot S \cdot W^T.
\]
Looking at the block matrix arithmetic details of this 2-dimensional transform we have:

\[
W \cdot S \cdot W^T = \left[ \frac{H}{G} \cdot S \left[ \begin{array}{c} H^T \\ G^T \end{array} \right] \right] = \left[ \frac{H \cdot S \cdot H^T}{G \cdot S \cdot H^T} \left| \frac{H \cdot S \cdot G^T}{G \cdot S \cdot G^T} \right. \right].
\]

This is the matrix of our fully transformed two-dimensional signal. It is important to note that a two-dimensional transformation produces four blocks instead of two. The one block in the upper left is a single lowpass block, whereas the other three are considered to be all highpass blocks.

To illustrate graphically what the differences between lowpass and highpass data amount to, we will give an example shortly of a photograph transformed by the Haar wavelet. First, it needs to be noted that we've taken some liberties in adjusting the data for these illustrations. In the rendering of any image, we are required to stay within the limits of the grayscale format, [0, 255]. But the Haar transform’s normalizing factor makes the potential range of the lowpass data become [0, 510], and the highpass data, [−255, 255]. So, in order to fit this data into a range of [0, 255], the lowpass values have been cut in half, and the highpass values were increased by 255 before cutting those in half as well. A particular point to be aware of is that what use to a value of zero in the highpass block, is now 127. In any regard, these manipulations allow us to really appreciate the relative changes occurring between adjacent values.

In Figure 2.3 we present a picture of a clown before and after transformation. In (a) we have an original picture (160 × 160 pixels). In (b) we have the lowpass block of the clown data (80 × 80 pixels). Notice how the averaging the adjacent data points (pixel values) has blurred the image. In (c) we show one of the highpass blocks (80 × 80 pixels). Here it is visually dramatic what calculating the differences between adjacent pixel values is doing. It's mostly preserving the locations where the picture has large amounts of contrast between adjacent pixels, i.e., where our eyes would see edges (for example, the eyeglasses).
In Figure 2.4, we show the entire picture before and after transformation. It can be seen that only one block has lowpass data, whereas the other three are all highpass data. To inverse the transformation, we simply calculate $W^T \cdot V \cdot W = S$.

So, we now have the basic mechanics for using discrete wavelet transformations in the field of signal processing. These techniques by themselves are not our main goal: There is a point to this signal processing other than breaking down and then perfectly reconstituting a signal. In between the forward and the inverse transformations we are going to make some changes to the processed signal with the intention of trying to
improve it in some idealized way. In this discussion our stated goal is to denoise a
 corrupted signal. The denoising will be accomplished in between the two
 transformations, through the use of statistics. In the next chapter we introduce the
 necessary statistical concepts used in the art of denoising.
3.1 Statistics: The Essentials.

When we use statistics to analyze data, we're not really trying to be objectively accurate in a traditional mathematical way. Rather, we are trying to impose a system of estimation that is somewhat subjective, yet still quite accurately defined by mathematics. A simple explanation would be to say that we are "trying to estimate the truth." Unlike a purely physical phenomena that follow an exact law of nature to an $n^{th}$ degree of accuracy, more complicated events, whether physical or man-made, tend to have too many unknown dependencies to come up with an exact answer. Under such circumstances we'll be happy to have, however uncertain, decent information on what the majority of data might look like. Statistical techniques offer us the tools to accomplish this.

To start with, two prominent descriptive measures used in statistics are the estimations of the data's central tendency (e.g., mean, median, and mode) and the estimations of the data's degree of dispersal (e.g., variance and standard deviation). We will develop these notions through the conventional mathematical model of randomness, the probability space.

**Definition 3.1.** Let $\Omega$ be a set (which we think of as the set of all possible outcomes of an "experiment"), $S$ be the $\sigma$-algebra of subsets of $\Omega$ (any subset $S$ is a set of events), and $P$ be the measure on $(\Omega, S)$ such that $P(\Omega) = 1$ (such a measure is called a probability measure). Then the triple $(\Omega, S, P)$ is called a probability space [16, p. 3, 7, 8].
A probability space is effectively a measure space where its measure, the probability measure $P$, is a measure with total measure equal to one.

In order to make a probability space conducive to mathematical analysis, we often need to define functions that assign a numerical value to our experimental events. Such functions are called random variables. That label may actually sound a little confusing at first, with the word 'variable' in there and the word 'function' not. But it does make sense if we think about how functions are typically defined by a mathematical expression that includes a variable. It just happens that the variable in this kind of function is a random event, not a number.

**Definition 3.2.** Let $\Omega$ be the set of all possible outcomes, and $S$ be the $\sigma$-algebra of subsets of $\Omega$. A finite, single-valued, measurable function $X$ that maps $\Omega$ into $\mathbb{R}$ is called a random variable (RV) [16, p. 41].

A relevant example of a RV for this discussion is connected to the familiar idea of taking a photograph. In this process of recording an image digitally, our camera assigns numerical values to the physical events of color and location. In particular, each specific pixel will be assigned some value that indicates the intensity of its light source. The light will be measured relative to some scale. For example, a grayscale image has pixel intensities ranging from 0 to 255, where 0 is black, and 255 is white.

**Theorem 3.1.** The random variable $X$ defined on the probability space $(\Omega, S, P)$ induces a probability space $(\mathbb{R}, \mathcal{B}, Q)$ by means of the correspondence:

$$Q(\mathcal{B}) = P(X^{-1}(\mathcal{B})) = P\{\omega: X(\omega) \in \mathcal{B}\} \text{ for all } \mathcal{B} \in \mathcal{B}.$$  

We write $Q = PX^{-1}$, and call it the distribution of $X$ [16, p. 43].
Random variables may be discrete functions, like in the camera example, or they may be functions that map their results continuously over some range. In either case, we will have varying degrees of dispersal around some central tendency. In order to characterize these dispersals we introduce the definition function.

**Definition 3.3.** A real-valued function $F$ defined on $(-\infty, \infty)$ that is non-decreasing, right-continuous, and satisfies

$$F(-\infty) = 0 \text{ and } F(+\infty) = 1,$$

is called a distribution function (DF) [16, p. 44].

**Definition 3.4.** Let $X$ be a random variable defined on $(\Omega, S, P)$. Define a point function $F(\cdot)$ on $\mathbb{R}$ by

$$F(x) = Q(-\infty, x] = P\{\omega: X(\omega) \leq x\}$$

for all $x \in \mathbb{R}$.

The function $F$ is called the distribution function of RV $X$ [16, p. 45].

What a distribution function accomplishes is to measure how probable some set of events is. Each event has been related by a random variable function $X$ to a real number. Now, you can imagine that there are many non-decreasing, right-continuous, real-valued functions that range from 0 to 1. In order to understand the connection between such a real-valued function $F$ and the probability it's associated with, we examine two different situations, discrete and continuous. For discrete situations we have various probability mass functions, $P\{X = x_i\}$, and for continuous situations we have probability density functions, $f(x)$. 
Definition 3.5. A random variable $X$ defined on $(\Omega, S, P)$ with distribution function $F$ is said to be a \textit{discrete random variable} if there exists a countable set $E \subseteq \mathbb{R}$ such that $P\{ X \in E \} = 1$. The collection of numbers $\{p_i\}$ satisfying $P\{ X = x_i \} = p_i \geq 0$, for all $x_i \in E$ and $\sum_{i=1}^{\infty} p_i = 1$, is called a \textit{probability mass function} (PMF) of RV $X$. The DF $F$ of $X$ is given [16, p. 48-49] by

$$F(x) = P\{X \leq x\} = \sum_{x_i \leq x} p_i.$$ 

Definition 3.6. A random variable $X$ defined on $(\Omega, S, P)$ with distribution function $F$ is said to be a \textit{continuous random variable} if there exists a nonnegative function $f(x)$ such that for every real number $x$ we have

$$F(x) = P\{X \leq x\} = \int_{-\infty}^{x} f(t) \, dt,$$

i.e., $F$ is absolutely continuous. The function $f$ is called the \textit{probability density function} (PDF) of the random variable $X$ [16, p. 50].

Theorem 3.2. Let $X$ be an RV of the continuous type with PDF $f$. Then for every Borel set $B \in \mathcal{B}$ [16, p. 50],

$$P(B) = \int_{B} f(t) \, dt.$$ 

If $F$ is absolutely continuous, and $f$ is continuous at $x$, we have

$$F'(x) = \frac{dF(x)}{dx} = f(x).$$

Given an RV $X$ with a known distribution, we can easily incorporate $X$ into a function to create a function of a random variable, $Y = f(X)$. We then can extend the theory of distributions to be able to determine the distribution of $Y$. 

44
Theorem 3.3. Let $X$ be an RV defined on $(\Omega, S, P)$ with known distribution $F$. Let $f$ be a Borel-measurable function on $\mathbb{R}$. Then $f(X)$ is also an RV and the distribution of the RV $Y = f(X)$ is determined [16, p. 57].

When we consider the signals that we will be working with (digital photographs, audio signals), what we have is a collection of several random variables, which, itself, is a RV, $X = (X_1, X_2, \ldots, X_n)$. We can describe the probabilities of the various possibilities of RV $X$ with a joint distribution function, $F$, and joint probability density function, $f$.

Definition 3.7. Let $X_1, X_2, \ldots, X_n$ be $n$ random variables. We say that these random variables have a joint distribution [16, p. 103, 106] if there exists a non-negative function $P\{X_1 = x_i, X_2 = x_j, \ldots, X_n = x_l\} = p_{i,j,\ldots,l}$, discrete case, or $f(t_1, t_2, \ldots, t_n)$, continuous case, such that

$$P\{X_1 \leq x_p, X_2 \leq x_q, \ldots, X_n \leq x_s\} = F(x_p, x_q, \ldots, x_s) = \begin{cases} \sum_{x_i \leq x_p, x_j \leq x_q, \ldots, x_l \leq x_s} p_{i,j,\ldots,l} & \text{discrete case} \\ \int_{-\infty}^{x_q} \cdots \int_{-\infty}^{x_p} f(t_1, t_2, \ldots, t_n) dt_1 dt_2 \ldots dt_n & \text{continuous case} \end{cases}$$

for all $(x_p, x_q, \ldots, x_s) \in \mathbb{R}^n$. The function $F$ is called the joint distribution function of $X = (X_1, X_2, \ldots, X_n)$, $p_{i,j,\ldots,l}$ the joint probability mass function, and $f(t_1, t_2, \ldots, t_n)$ the joint probability density function of $X$.

There are some theoretical conditions that need to be placed on $F$ for it to truly be a joint distribution function. Two of them look similar to the conditions for the earlier definition of a DF for a single RV $X$. 

45
Theorem 3.4. A function $F(x_1, x_2, \ldots, x_n)$ is the joint DF of some $n$-dimensional RV if and only if $F$ is non-decreasing and right-continuous with respect to all arguments $x_1, x_2, \ldots, x_n$ and satisfies the following conditions:

(a) $F(-\infty, x_2, \ldots, x_n) = F(x_1, -\infty, \ldots, x_n) = \ldots = F(x_1, x_2, \ldots, -\infty) = 0$ for all $x_k$,

(b) $F(+\infty, +\infty, \ldots, +\infty) = 1$,

(c) For every $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ and all $\varepsilon_i > 0$, $i = 1, 2, \ldots, n$, the following inequality holds:

$$F(x_1 + \varepsilon_1, x_2 + \varepsilon_2, \ldots, x_n + \varepsilon_n) - \sum_{i=1}^{n} F(x_1 + \varepsilon_1, \ldots, x_{i-1} + \varepsilon_{i-1}, x_i, x_{i+1} + \varepsilon_{i+1}, \ldots, x_n + \varepsilon_n) + \sum_{i,j=1, i < j}^{n} F(x_1 + \varepsilon_1, \ldots, x_{j-1} + \varepsilon_{j-1}, x_j, x_{j+1} + \varepsilon_{j+1}, \ldots, x_n + \varepsilon_n) + \ldots + (-1)^n F(x_1, x_2, \ldots, x_n) \geq 0.$$ 

For the proof of the case for two variables, see Rohatgi [16, pp. 103-105] and Tucker [20, p. 26].

There is one more general kind of distribution to mention, and that is the case where we have a joint distribution of an $n$-dimensional multiple RV $X$, but we would like to identify the DF of only one (maybe more, but less than $n$) of the RVs. Such a distribution is called a marginal distribution.
Definition 3.8. Let \( X_1, X_2, \ldots, X_n \) be \( n \) random variables, and suppose that \( p_{i,j,\ldots,l} \) is the joint probability mass function (in the discrete case), or \( f(x_1, x_2, \ldots, x_n) \) is the joint probability density function (in the continuous case), for these RVs. We say that the random variable \( X_m, 1 \leq m \leq n \), has a marginal distribution if there exists a non-negative function \( p_{\cdot,\ldots,k,\ldots,\cdot} \) or \( f_m(x_m) \) defined [16, p. 108-111] by

\[
p_{\cdot,\ldots,k,\ldots,\cdot} = \sum_{i,j,\ldots,l=1 \atop i,j,\ldots,l \neq k}^{\infty} p_{i,j,\ldots,l} = P\{X_m = x_k\}
\]

\[
f_m(x_m) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(t_1, t_2, \ldots, t_m, \ldots, t_n) dt_{i\neq m} dt_{j\neq m} \cdots dt_{l\neq m}
\]

such that

\[
P\{X_m \leq x_r\} = F_m(x_r) = \begin{cases} \sum_{x_k \leq x_r} p_{\cdot,\ldots,k,\ldots,\cdot} & \text{discrete case} \\ \int_{-\infty}^{x} f_m(t_m) dt_m & \text{continuous case} \end{cases}
\]

for all \( x_r \in \mathbb{R} \). The function \( F_m \) is called the marginal distribution function of \( X_m \), the function \( p_{\cdot,\ldots,k,\ldots,\cdot} \) is called the marginal probability mass function, and \( f_m(x_m) \) is called the marginal probability density function. (Note: these definitions can be generalized to a \( d \)-dimensional, \( 1 \leq d \leq n - 1 \), marginal distribution.)

When it comes to the noise that we will be dealing with, we will be treating it as a random variable. Additionally though, we will be assuming that the probable strength of any one instance of noise will not influence the strength of any other instance of noise. How we say this statistically is to call these random variables independent.

Definition 3.9. Let \( F(x_1, x_2) \) and \( F_1(x_1) \) and \( F_2(x_2) \), respectively, be the joint DF of \( (X_1, X_2) \) and the marginal DFs of \( X_1 \) and \( X_2 \). Then we say \( X_1 \) and \( X_2 \) are independent [16, p. 119] if and only if

\[
F(x_1, x_2) = F_1(x_1)F_2(x_2) \quad \text{for all} \ (x_1, x_2) \in \mathbb{R}^2.
\]
The definition of *independence* can be generalized to a sequence \( \{X_n\} \) of RVs, meaning, for every \( n = 2, 3, 4, \ldots \) the RVs \( X_1, X_2, \ldots, X_n \) are independent. One other statistical quality we can assume about noise is that any instance of noise is *identically distributed*, which we can combine with independence to describe random variables that are both *independent & identically distributed*.

**Definition 3.10.** We say that RVs \( X_1 \) and \( X_2 \) are *identically distributed* if \( X_1 \) and \( X_2 \) have the same DF [16, p. 123], that is,

\[
F_{X_1}(x) = F_{X_2}(x) \quad \text{for all } x \in \mathbb{R}.
\]

**Definition 3.11.** We say that \( \{X_n\} \) is a sequence of independent, identically distributed (i.i.d.) RVs with common law \( \mathcal{L}(X) \) if \( \{X_n\} \) is an independent sequence of RVs and the distribution of \( X_n \) (\( n = 1, 2, \ldots \)) is the same as that of \( X \) [16, p. 123].

By themselves, these distribution functions and probability mass/density functions give us the ability to determine the likelihood of any set of events that we can measure. This is not all that we can measure with these functions. We can now also calculate the idea of an *expected value*. The most simple example of an expected value is our common notion of *arithmetic mean* (i.e., average). There are many more mathematical expectations that can be calculated. If we are given a random variable \( X \), its probability mass (or density) function, \( \{p_i\} \) (or \( f \)), and a function \( g \), then we can calculate such expected values as \( X^2 \) or \( g(X) \). First, let's formalize the concept of expected value with a definition.
Definition 3.12. Let $X$ be a discrete random variable with probability mass function $p_i = P[X = x_i], i = 1, 2, \ldots$. If
\[ \sum_{i=1}^{\infty} |x_i| p_i < \infty, \]
then we say that the expected value of $X$ exists [16, p. 69] and write
\[ \mu = E(X) = \sum_{i=1}^{\infty} x_i \cdot p_i. \]

Similarly, let $X$ be a continuous random variable with probability density function $f$. If
\[ \int_{-\infty}^{\infty} |x| f(x) dx < \infty, \]
then we say that the expected value of $X$ exists [16, p. 70] and write
\[ \mu = E(X) = \int_{-\infty}^{\infty} x \cdot f(x) \, dx. \]

We can see that for discrete RVs, the expected value is the probability-weighted sum of the possible values. For continuous RVs with a density function, it is the probability density-weighted integral of the possible values. Now the definition above was written in the context of finding the expected value of $X$. If we instead actually wanted to find the expected value of $X^2$, with density function $f$, we would need to calculate
\[ E(X^2) = \int_{-\infty}^{\infty} x^2 f(x) \, dx. \]
And if we wanted to find expected value of $g(X)$, where $g$ is an arbitrary function, then we would calculate
\[ E(g(X)) = \int_{-\infty}^{\infty} g(x) f(x) \, dx, \]
which is the inner product of $f$ and $g$, $\langle f, g \rangle$.

If we add the results of two or more random variables, then we have one kind of function of several random variables. When we consider how the Haar transform works, we note that we will be adding two random variables at a time, as well as multiplying.
them by some constant factor of $\sqrt{2}$. Thus will need to know some properties of expected values under basic mathematical operations.

**Theorem 3.5.** (Addition Properties of Expected Value) [21, p. 507]. Let $X_1$ and $X_2$ are two random variables, and $a$ and $b$ be arbitrary real constants. Then

(a) $E(a) = a$,
(b) $E(aX_1 + b) = aE(X_1) + b$,
(c) $E(X_1 + X_2) = E(X_1) + E(X_2)$.

More generally, if $X_1, X_2, \ldots, X_n$ are $n$ random variables, then

(d) $E\left(\sum_{k=1}^{n} X_k\right) = \sum_{k=1}^{n} E\left(X_k\right)$.

Of special importance are the expectations of $X^n$, where $n$ is a positive integer. $E(X^n)$ is called the $n^{th}$ moment of $X$ about the origin [16, p. 72], and the $1^{st}$ moment gives us a good estimate of the data's central tendency, $\mu = E(X)$. We also need to a way of determining the character of the data's range of dispersal around its central tendency. In addressing that, we will first mention the general concept of a moment of order $k$ about a point $c$, $E(X - c)^k$ [16, p. 79]. Then, if we take $c = E(X) = \mu$, we get what we need, the central moment of order $k$, $E(X - \mu)^k$. Of particular importance is the central moment of order 2, which is called the variance.

**Definition 3.13.** If $E(X^2)$ exists, we call $E(X - \mu)^2$ the variance of $X$, and we write

$$\sigma^2 = \text{Var}(X) = E(X - \mu)^2 = \begin{cases} \sum_{i=1}^{n} (x_i - \mu)^2 p_i & \text{discrete case} \\ \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx & \text{continuous case.} \end{cases}$$

The quantity $\sigma$, the square root of the variance, is called the standard deviation of $X$. 

50
We need to mention here some of the properties of variance. These properties will be important to the understanding and appreciation of how we use the Haar transform to accomplish denoising.

**Theorem 3.6.** (Properties of Variance) [21, p. 510]. Let \( X \) be a RV, \( X_1, X_2, \ldots, X_n \) be \( n \) independent RVs, \( a \) and \( b \) be arbitrary real constants, and \( \mu = E(X) \). Then

(a) \( \text{Var}(aX + b) = a^2 \text{Var}(X) \),

(b) \( \text{Var}(X) = E(X^2) - \mu^2 \),

(c) \( \text{Var}(X_1 + X_2 + \ldots + X_n) = \text{Var}(X_1) + \text{Var}(X_2) + \ldots + \text{Var}(X_n) \).

It was mentioned a little bit ago that there are many probability distributions to consider. One distribution familiar to most is the normal distribution (also commonly referred to as a Gaussian distribution). This particular distribution is tremendously important to our discussion, particularly to our study of noise, and the measurement of the errors caused by noise.

**Definition 3.14.** A random variable \( X \) is said to have a normal distribution with mean parameter \( \mu \) and variance parameter \( \sigma \) if its PDF is given by

\[
f(x) = \varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).
\]

If we set \( \mu = 0 \), and \( \sigma = 1 \), then we have the standard normal distribution

\[
\varphi(t) = \varphi_{0,1}(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right).
\]

You can see that \( \varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma} \varphi\left(\frac{x-\mu}{\sigma}\right) \).
If a random variable $X$ is normally distributed with mean $\mu$ and variance $\sigma^2$, then we write $X \sim \mathcal{N}(\mu, \sigma^2)$. Given a normally distributed random variable $X$, we can, for convenience, define a new random variable, $Z = (X - \mu)/\sigma$, which is a function of the original RV $X$. We can then say that the distribution function $F$ of a $\mathcal{N}(0, 1)$ random variable is given by

$$F(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-t^2/2} dt.$$ 

Figure 3.1 shows the probability density function $f$ (the familiar bell curve) of a normally distributed random variable $Z$, $Z \sim \mathcal{N}(0, 1)$.

![Probability density function of the standard normal distribution.](image)

**Figure 3.1.** Probability distribution function of the standard normal distribution.

The importance of the normal distribution is that it is the most commonly used tool for the statistical analysis of complex phenomena, because any random variable that is the sum of a large number of independent factors is likely to be normally distributed. This distribution provides us with a very simple model for an otherwise unmanageable problem, which is why it is frequently used as a model for noise.

It should be of no surprise that not all noises are created equal. However, it's interesting to note that there are various statistical characterizations for the different types of noise. One such attribution that scientists give to noise is the description of its spectral density. Noise does not necessarily affect the entire frequency spectrum of a given signal uniformly. It may actually have a varying amount of intensity at the different frequencies. If we consider the noise to be operating as a random variable, then this means the noise is not identically distributed from frequency to frequency. One of the spectral density characterizations involves assigning a color to noise. In Figure 3.2 we selected three different noise colors for graphical representation. *Pink noise* has a spectral density that is proportional to the inverse of the frequency, meaning it has a stronger intensity on the lower frequencies. On the other hand, blue noise has a spectral density that is in direct proportion to the frequency, giving a stronger effect at the higher frequencies. Finally, with what may be a familiar reference to some, we also have *white noise*. White noise is a noise that pervades the entire frequency spectrum with equal effect. This distinguishes it as being the noise most difficult to remove.

![Graph showing spectral power densities of pink, white, and blue noises.](image)

**Figure 3.2.** Linear-log plots of some colored noise spectral power densities.
Here is a quick note about the decibel (dB) calculations involved in Figure 3.2. The decibel formula is given by

\[ L_{dB} = 10 \cdot \log_{10} \left( \frac{P_1}{P_0} \right), \]

where \( P_1/P_0 \) is a ratio that compares one power value, \( P_1 \), to some reference power value, \( P_0 \).

Decibels are a dimensionless measurement. They do not indicate the actual power that could be measured by some absolute unit of measurement (like watts or volts). For example, consider the pink noise shown in our graph. It has been representatively defined by the formula \( L_{dB} = 10 \cdot \log_{10} \left( \frac{100}{2\omega} \right) \). This returns a value of 0 dB at \( \omega = 50 \) Hz, making our reference power, \( P_0 \), equal to whatever the power of the noise may be at 50 Hz. If we go up one octave, to \( \omega = 100 \) Hz, then we have \( P_{100\text{Hz}} \) equal to half of \( P_{50\text{Hz}} \); that is a loss of 3.01 dB between the two frequencies. In general, the spectral power density for pink noise, given some reference frequency \( \omega_0 \), will decrease inversely with \( \omega_1 \),

\[ \frac{P_1}{P_0} = \frac{1}{2 \log_2 \left( \frac{\omega_1}{\omega_0} \right)}, \]

which is about \(-3\)dB per octave.

Here is one final note about the graph: For the purpose of conciseness we've substituted \( \omega \) for \( \log_2(\omega_1/\omega_0) \), which would actually make \( \frac{1}{2} \) Hz the reference frequency.

In addition to the noise's spectral density, we also need to consider one more statistical characterization to be able to work with noise mathematically. Again, since noise operates like a random variable, we can describe the range and dispersal of the varying occurrences of noise with a probability density function. To be clear, this is a different consideration than the noise's spectral density. We are now talking about characterizing how the 'instances' of noise vary in intensity around any arbitrary point on the frequency spectrum. It may sound similar to the idea of spectral density, but it's not. So let's contrast this a little more, with some statistical language. Spectral density
describes the noise's changing "variance", i.e., strength, at a specific point on the frequency spectrum. But the PDF for the same noise will describe its generic dispersion, i.e., the shape of its curve, for any instance of noise no matter where it happens on the frequency spectrum. This second characterization essentially gives our noise a two-dimensional description.

Fit for our purposes, the most common PDF characterization given to noise is Gaussian (i.e., normally distributed). This should be understandable since normal distributions, as described earlier, are the preferred tool for the analysis of events resulting from a large number of independent factors. If it were otherwise, meaning there was something systematic to our noise (like a motor running in the background), then we could likely characterize the noise more closely with a more suitable PDF (like a Cauchy distribution, for example). In any regard, the noise we are going to work with will be assumed to be Gaussian white noise.

3.3 Estimation: Decisions that Consider the Statistical Errors of Loss & Risk.

Noise can easily be thought of as causing errors in our attempts at measuring the true signal. We will eventually present a technique that reduces the effects of noise-induced error. Before we do, we first examine components of two underlying processes, estimation and decision, that will form the basis of our technique.

When we observe (i.e., record) a signal, whether an audio signal or digital photograph, we are observing a large set of data points. Each one of those observed data points will be mapped to a numerical value, which, with all accuracy, has only one 'true' value. A deviation in our recording of the true value is called an observational error. Let \( \theta \) represent the perfect vector resulting from the sampling of our true signal, let \( \varepsilon \) represent the noise vector, \( \varepsilon \sim \mathcal{N}(0, 1) \), that is impacting our true signal, and let \( \sigma \)
represent the noise variance (i.e., strength of the noise). Then we can represent the actual observations of our signal with the following model:

\[ x = \theta + \sigma e, \]

where \( x \) is the observation vector.

Each vector component \( \theta_i \) in \( \theta \) is one sampled parameter of the true signal. The set of all admissible values of the parameters makes a parameter space, denoted \( \Theta \). The goal in our denoising technique will be to come up with the best estimate for each \( \theta_i \), which in turn gives us the best estimate for \( \theta \). We will denote the respective estimates of \( \theta_i \) and \( \theta \) as \( \hat{\theta}_i \) and \( \hat{\theta} \). To make these estimates, we will need to make decisions that are based not only on our observations, \( x \), but also on a well-reasoned decision function, \( \delta \).

**Definition 3.15.** Let \( \Theta \) be a parameter space, and let \( \{ f_{\theta_i} : \theta_i \in \Theta \} \) be a family of PDFs (or PMFs). Further, let \( (X_1, X_2, \ldots, X_n) \) be the RV obtained from sampling the parameter space \( n \) times and let \( (x_1, x_2, \ldots, x_n) \) be the observed sample point. Finally, let \( \mathcal{A} \) be a set, which we think of as the set of all decisions that may be taken based on those observations. Then a decision function, \( \delta \), is a mapping \( \delta : \mathbb{R}^n \rightarrow \mathcal{A} \) that relates observations of \( X \) to prescribed decisions in \( \mathcal{A} \), \( \delta(X) \in \mathcal{A} \) [16, p. 424].

Our decisions won't be perfect. There will still be some degree of error in our estimates. If we can measure the effectiveness of our decisions, then we will be able to determine which particular decision function works best. So, to accomplish that, we are going to measure what is called the loss, which is a numerical value that indicates how wrong the result from the decision, \( \delta(X) = \hat{\theta} \), is compared to the perfect vector \( \theta \). There are many different formulations for calculating a loss, and each one of them is associated with its own loss function.
Definition 3.16. Let $\mathfrak{A}$ be an arbitrary space of decisions. Then a loss function, $L$, is a mapping $L : (\Theta, \mathfrak{A}) \to \mathbb{R}^+$ that relates any decision about a parameter to a nonnegative real number. The resulting value is an indication of the amount of loss coming from that decision, $L(\theta_i, \hat{\delta}(x_i)) = L(\theta_i, \hat{\delta})$ [16, p. 425].

One loss function to consider is the absolute-difference loss function,

$$L(\theta_i, \delta(x_i)) = |\theta_i - \hat{\theta}_i|.$$ 

Another loss function is the quadratic loss function,

$$L(\theta_i, \delta(x_i)) = (\theta_i - \hat{\theta}_i)^2.$$

For any estimation process we need to realize that our losses will not always be the same every time. Consider our noise for instance. Most of the time the loss due to noise will be close to zero, but sometimes that noise will cause bigger losses. For our discussion, we've assumed that noise will be normally distributed. Recalling the statistical concept of expected value, we can define an expectation for our losses. The expected loss on $\theta_i$ will be called the risk of the estimator $\hat{\theta}_i$.

Definition 3.17. Let $\mathfrak{D}$ be a class of decision functions, $\delta : \mathbb{R}^n \to \mathfrak{A}$, and let $L$ be a loss function defined on $(\Theta, \mathfrak{A})$. Then a risk function, $R$, is a mapping $R : (\Theta, \mathfrak{D}) \to \mathbb{R}^+$ that relates a decision function $\delta$ and an unknown parameter $\theta_i$ to a nonnegative real number. The resulting value is an indication of the expected loss coming from that decision, $R(\theta_i, \delta) = E_\theta L(\theta_i, \delta(x_i))$ [16, p. 424].

A risk that is of particular interest to us is the one associated with the quadratic loss function.
Definition 3.18. Let \( \theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \) be the true function vector of a sampled signal. Let \( \hat{\theta} \) be its estimator, and let the quadratic loss, \( L(\theta_i, \hat{\theta}_i) = (\theta_i - \hat{\theta}_i)^2 \) be the loss function for each \( \hat{\theta}_i \). Then the mean square error of \( \hat{\theta} \) is defined to be [16, p. 354] the expected loss of \( L \), given the actual sampling \( s_i = 1, 2, \ldots, n \):

\[
\text{MSE}(\hat{\theta}) = R(\theta, \hat{\theta}) = E_{\theta \sim \theta}(\theta_i - \hat{\theta}_i)^2 = \frac{1}{n} \sum_{i=1}^{n} (\theta_i - \hat{\theta}_i)^2 = \frac{1}{n} \left\| \theta - \hat{\theta} \right\|_2^2.
\]

The quantity \( \sum_{i=1}^{n} (\theta_i - \hat{\theta}_i)^2 = \left\| \theta - \hat{\theta} \right\|_2^2 \) is known as the total quadratic loss (or \( \ell^2 \)-loss).

The mean square error of the estimator \( \hat{\theta} \), with respect to the estimated parameter \( \theta \), is the second moment of the measured loss, essentially, the "average of the square of the error". Taking the square root of the MSE yields the root mean square error (RMSE), which is analogous to standard deviation.

These definitions of loss and risk are quite theoretical. Meaning, we need to appreciate that coming up with the true and exact values requires much real knowledge that we don't actually have. In the case of a our corrupted signal, \( x = \theta + \sigma \varepsilon \), we do not know where the true signal \( \theta \) is exactly, nor do we know how strong any particular instance of noise \( \sigma \varepsilon \) is. Thus our estimation of \( \theta, \hat{\theta} \), is going to be based on a prescribed decision function, \( \delta \), that is partly operating on our observations \( x \), i.e., \( \hat{\theta} = \delta(x) \). That means in our larger discussion of denoising, we will eventually come to an idealized decision function, \( \delta^* \), that provides for, by some measure, the optimal denoising technique.

As we formulate this decision function, \( \delta^* \), we will base it on some well-reasoned theoretical considerations. To evaluate this reasoning, we will need some statistical structure. We would like our decision function to be valid for all parameters \( \theta_i \) in \( \Theta \) (or at least, for all \( \theta_i \) in a family of parameters, \( \Theta_\delta \)). Obviously, our concern will be the minimization of error. The particular error that we will work to minimize, considering all
the $\theta_i \in \Theta$, is the mean square error. That means we are working to minimize the
expected risk.

**Definition 3.19.** Let $\Theta$ be a parameter space, and $\theta = [\theta_1 \ \theta_2 \ldots \ \theta_n]^T$ be a true but
unknown value of a sampled parameter, $\theta \subset \Theta$. Assume the model, $x = \theta + \sigma \varepsilon$, where
$x$ is the observation of the sampled parameter, and $\sigma \varepsilon$ is the error on that observation.
Further, let $\delta(x) = \hat{\theta}$ be the estimator of $\theta$ that is dependent on the observation $x$, and
finally, let $\text{MSE}(\hat{\theta})$ be the risk on $\hat{\theta}$. Then the expected risk of $\hat{\theta}$ is defined
as:

$$E_\theta \left( \text{MSE}(\hat{\theta}) \right) = E_\theta \left( R(\theta, \delta(x)) \right) = E_{\theta_i \in \Theta} \left( E_i \left( \theta_i - \delta(x_i) \right)^2 \right)$$

$$= E_{\theta_i \in \Theta} \left( \frac{1}{n} \sum_{i=1}^{n} \left( \theta_i - \delta(x_i) \right)^2 \right) = E_{\theta_i \in \Theta} \left( \frac{1}{n} \left\| \theta - \delta(x) \right\|_2^2 \right).$$

This is the basic problem of decision theory: to find the particular decision
function, $\delta^*$, that minimizes, in some measure, the expected risk over all $\theta_i \in \Theta$.
For us to determine which particular $\delta$ will be our optimal decision function, $\delta^*$, we
do need one more thing, a criterion called the principle of minimax.

**Definition 3.20.** Given a space of possible decisions, $\mathfrak{A}$, a class of decision functions, $\mathcal{D}$,
and a loss function, $L(\theta_i, \delta(x_i))$, the principle of minimax is to choose $\delta^* \in \mathcal{D}$ so that

$$\max_{\theta_i} R(\theta_i, \delta^*) \leq \max_{\theta_i} R(\theta_i, \delta)$$

for all $\delta$ in $\mathcal{D}$. Such a rule $\delta^*$, if it exists, is called a minimax (decision) rule [16, p. 425].

Since our problem is one of estimation, we call $\delta^*$ satisfying (1) a minimax
estimator of $\theta_i$. Essentially, what we are looking to find is the particular rule whose
worst outcome is least among all the worst outcomes possible of any rule. We are not
going the best outcome necessarily. We are just assuring ourselves that when we do
encounter the worst possible outcome, then we've chosen a rule that minimizes the error we would get from it.

So, if we were to use the absolute-difference loss function, $|\Theta - \hat{\Theta}|$, then it would be the observed median that would minimize our risk.

**Theorem 3.7.** Let $f(\theta_i)$ denote the PDF of $\Theta$, and let the loss function be given by $|\Theta - \hat{\Theta}|$. Then the estimator which minimizes the expected value of the loss function is the median of the distribution of $\Theta$ [9, p. 31].

**Proof:** Let $m$ be an arbitrary parameter in $\Theta$. Then

$$E_\Theta |\Theta - m| = \int_{-\infty}^{m} (m - \theta) f(\theta) d\theta + \int_{m}^{\infty} (\theta - m) f(\theta) d\theta.$$  

Using integration by parts, $uv - \int v du$, to evaluate the integrals, we take the antiderivatives of $f$ to be either $F(\theta_i)$ or $F(\theta_i) - 1$, as appropriate (i.e., $f(\theta_i)d\theta_i = F(\theta_i) - C$). Then letting $\Delta$ represent the primitive of $F$, we get

$$E_\Theta |\Theta - m| = \int_{-\infty}^{m} (m - \theta) f(\theta) d\theta + \int_{m}^{\infty} (\theta - m) f(\theta) d\theta$$

$$= \left[ (m - \theta) F(\theta) \right]_{-\infty}^{m} + \int_{-\infty}^{m} F(\theta) d\theta + \left[ (\theta - m)(F(\theta) - 1) \right]_{m}^{\infty} - \int_{m}^{\infty} (F(\theta) - 1) d\theta$$

$$= \int_{-\infty}^{m} F(\theta) d\theta + \int_{m}^{\infty} (1 - F(\theta)) d\theta$$

$$= \left[ \Delta(m) - \Delta(-\infty) \right] + \left[ (\infty - \Delta(\infty)) - (m - \Delta(m)) \right].$$

Because $|\Theta - \hat{\Theta}|$ is a convex function, if we simply take the derivative of (2) with respect to $m$, and set that result equal to zero,

$$F(m) - (1 - F(m)) = 0$$

and then solve for $F(m)$,

$$F(m) = \frac{1}{2},$$

we get the result that the "$m$" that minimizes absolute-difference loss is the median.
If we were to use the quadratic loss function, \( (\Theta - \hat{\Theta})^2 \), then it would be the observed mean, \( \mu \), that would minimize our quadratic loss.

**Theorem 3.8.** Let \( f(\theta_i) \) denote the PDF of \( \Theta \), and let the loss function be given by \( (\Theta - \hat{\Theta})^2 \). Then the estimator which minimizes the expected value of the loss function is the *mean* of the distribution of \( \Theta \) [9, p. 28-29].

**Proof:** We are seeking the value \( \hat{\Theta} \) that minimizes \( E_\theta (\Theta - \hat{\Theta})^2 \).

\[
E_\theta (\Theta - \hat{\Theta})^2 = E_\theta (\Theta_i^2 - 2\Theta_i \hat{\Theta} + \hat{\Theta}_i^2)
\]

(3)

\[
= E_\theta \Theta^2 - 2\hat{\Theta}E_\theta \Theta + \hat{\Theta}^2.
\]

Because \( (\Theta - \hat{\Theta})^2 \) is a convex function, if we simply take the derivative of (3) with respect to \( \hat{\Theta} \), and set that result equal to zero,

\[-2E_\theta \Theta + 2\hat{\Theta} = 0,
\]

then solve for \( \hat{\Theta} \),

\[
\hat{\Theta} = E_\theta \Theta
\]

we get the result that \( E_\theta \Theta \), the mean of \( \Theta \), is the value that minimizes the quadratic loss.

We've now laid out the underlying statistical structure that we will use to evaluate our denoising techniques. So we head onto the next chapter, and towards the main focus of this discussion, denoising.
Chapter 4

Denoising, Thresholding, and an Idealistic Oracle

4.1 Denoising: An Overview

We are now ready to look at the actual mechanics of denoising. We assume the following model for our observed signal:

\[ \mathbf{x} = \theta + \sigma \mathbf{\varepsilon}, \]

where \( \mathbf{x} \) is the observation vector, \( \theta \) is the perfect vector from sampling our true signal, and \( \sigma \mathbf{\varepsilon} \) is the noise vector (with \( \sigma \) being the level of the noise). Component-wise, each \( x_i = \theta_i + \sigma \varepsilon_i, \ i = 1, \ldots, n \). Remembering that \( \varepsilon \sim \mathcal{N}(0, 1) \), thus the \( E(\varepsilon_i)^2 = 1 \), it should be obvious that the mean square error between our observations and the true signal will be

\[ \text{MSE}(\mathbf{x}) = E_{\theta} \left( \theta - (x_i) \right)^2 = E_{\theta} \left( \theta - (\theta_i + \sigma \varepsilon_i) \right)^2 \]

\[ = E_{\theta} \left( \sigma \varepsilon_i \right)^2 = \sigma^2 E_{\theta} (\varepsilon_i)^2 = \sigma^2. \]

We are going to try to reduce this error by a denoising technique that incorporates wavelets into its process. Our signal will essentially be the function that we are going to analyze. There are many different wavelet transforms that one can use to perform the analysis, but for this discussion we are going stick with the rather simple Haar wavelet transform, so as to easily illustrate some other important details that we want to point out.

In its raw form, there is not much we can do to a noisy signal directly. The process of denoising is a multi-step program that begins with the transformation of the signal, via the Haar wavelet transform, into its lowpass and highpass components (as illustrated back in Chapter 2). This first step does not do too much to denoise the signal. It will be explained shortly that the lowpass portion of the data, by itself, is a denoised version of the original data. That alone is not a very sophisticated result. Sophistication comes in the next step, when we look at the highpass data, and apply a process that we
have not even mentioned yet. That process is called "wavelet shrinkage" and incorporates the technique of thresholding. We will explain this in a short while too. It is after this that we can reconstitute the signal, via a synthesis of the modified highpass and un-modified low pass, into a newly revised estimate of the true signal.

There is an intricate balancing act of choices being made here. We are soon going to see how the choice to threshold, or not to threshold, is a choice between two kinds of error. We start the explanation by looking at how the low pass filter accomplishes its noise reduction.

4.2 Lowpass Filtering: Muffling the Noise.

When we are sampling a signal impacted by noise, \( x = \theta + \sigma \varepsilon \), we are, in one sense, taking a sample of the noise. In order to illustrate how lowpass filtering muffles this noise, we are going to, for the moment, consider a true signal that is zero everywhere. That is, \( \theta_i = 0 \) for all \( i \), thus \( x = \sigma \varepsilon \). Then given a normally distributed noise that has a mean \( \mu = 0 \) and variance \( \sigma^2 \), we will have a random variable \( X_i \sim \mathcal{N}(0, \sigma) \). We can then use the properties of variance (mentioned in chapter 3) to see what happens to the expected error from noise after lowpass filtering.

Before we look at Haar's specific lowpass algorithm, let's first look at what happens to the noise-induced variance on the calculated average of \( n \) samples of a RV \( X_i \sim \mathcal{N}(0, \sigma) \) that is i.i.d. Remember, variance has the property of, \( \text{Var}(aX_i + b) = a^2 \text{Var}(X_i) \), and \( \text{Var}(\sum X_i) = \sum \text{Var}(X_i) \). Thus,

\[
\text{Var} \left( \frac{X_1 + X_2 + \cdots + X_n}{n} \right) = \frac{1}{n^2} \text{Var} \left( X_1 + X_2 + \cdots + X_n \right)
\]

\[
= \frac{1}{n^2} \left( \text{Var} \left( X_1 \right) + \text{Var} \left( X_2 \right) + \cdots + \text{Var} \left( X_n \right) \right)
\]

\[
= \frac{1}{n^2} \cdot n \cdot \text{Var}(X_i) = \frac{\sigma^2}{n}.
\]
If the variance \( \sigma^2 \) was associated with Gaussian white noise, then a noise level of \( \sigma \) would be cut in half after four samples.

Now let’s look at the algorithm for the low-pass portion of the Haar wavelet transform. Remember that the Haar transform involves multiplying by the \( \sqrt{2} \) so as to normalize the transformation. Using \( j \), instead of \( n \), to indicate the iterative level, we get the following lowpass portion of the transformed data for the first, second, and third iterations:

\[
\begin{align*}
  j = 1: & \quad \frac{X_1 + X_2}{\sqrt{2}}; \\
  j = 2: & \quad \frac{X_1 + X_2 + X_3 + X_4}{\sqrt{2}} = \frac{X_1 + X_2 + X_3 + X_4}{2}; \\
  j = 3: & \quad \frac{X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8}{2\sqrt{2}}.
\end{align*}
\]

So, in general, the lowpass portion of the transformed data after \( j \) iterations is:

\[
\frac{1}{(\sqrt{2})^j} \sum_{i=1}^{2^j} X_i.
\]

Then, given a noise-related variance of \( \sigma^2 \) on our blank signal, we can determine that in the low pass portion of the Haar transformed data the variance due to noise is:

\[
\text{Var} \left( \frac{1}{(\sqrt{2})^j} \sum_{i=1}^{2^j} X_i \right) = \frac{1}{2^j} \cdot \text{Var} \left( \sum_{i=1}^{2^j} X_i \right) = \frac{1}{2^j} \cdot 2^j \cdot \sigma^2 = \sigma^2.
\]

So, it looks like the noise is unchanged. That is an illusion due in part to how an orthonormal transform preserves distances. But it must be remembered that at each iteration of the Haar wavelet transform, the range of the data in the low pass portion, compared to the range of data on the original domain, has been increased by a factor of \( \sqrt{2} \). Thus, the new noise level, relative to the scale of the original, is lower:
\[ \sigma_j = \frac{\sigma}{(\sqrt{2})^j}, \] where \( j \) is the iterative level.

So each iterative level \( j \) involves \( 2^j \) independent identically distributed data points. Since \( j = \log_2 2^j \), we can rewrite the previous equation as:

\[ \sigma_j = \sigma_{\log_2 n} = \frac{\sigma}{(\sqrt{2}) \log_2 n} = \frac{\sigma}{(2^{\frac{1}{2}} \log_2 n)} = \frac{\sigma}{(2^{\frac{1}{2}} n) \log_2 n} = \frac{\sigma}{(n)^\frac{1}{2}} = \frac{\sigma}{\sqrt{n}}, \]

where \( n = 2^j \) for \( j = 0, 1, 2, 3, \ldots \).

Thus, if \( \sigma \) is the level of normal noise impacting our blank signal, then the relative noise \( \sigma_j \) on the low pass portion decreases with each iterative level \( j \) by a factor of \( \sqrt{2} \).

The following figure graphs the noise reducing effect of the low pass filter (the line relates \( n \) as a real number, the bold points relate \( j \) as an integer).

![Figure 4.1. Noise level reductions of the Haar lowpass filter.](image)

It should be clear that putting a real signal back into our model will not change the effective error the noise causes. Then if we square \( \sigma_j \) we will get the \( \text{MSE}(\hat{\theta}) \) that exists in the lowpass portion of the data. Obviously the comparative noise level in lowpass data is less than the original noise level \( \sigma \):

\[ \text{MSE}_{\text{lowpass}}(\theta) = \sigma_j^2 = \left( \frac{\sigma}{\sqrt{2}^j} \right)^2 = \frac{\sigma^2}{2^j}, \]

\( j \) being the number of iterative levels taken with the Haar transform.
4.3 Highpass Filtering: The Highlights, and Less Noise too.

Now let's discuss what the high pass portion of the Haar wavelet transform does with the same noise. Take a look at the highpass portion of the data after one iteration:

\[ N = 1: \frac{X_1 - X_2}{\sqrt{2}}. \]

If we go back to discussing just the noise, \( x = \sigma \varepsilon \), and remembering that the noise is normally distributed about a mean \( \mu = 0 \), then we should understand that any \( X_i \) is equally likely to be negative as it is positive. This fact was equivalently true for the low pass algorithm as well. What this effectively means is, with regards to noise, these algorithms are really indistinguishable. The normalized noise will be at the same level, \( \sigma \), in the highpass as it is in the lowpass.

That does not mean that the relative MSE(\( \hat{\theta} \)) in the highpass has been reduced as it was in the lowpass. When it comes to real signal data, the highpass is not a muffler on the original signal like the lowpass was. If we go back to the discussion we had at the end of section 2.3, we'll recall that the difference between the highpass and lowpass filters is that the highpass filtering preserves the differences between data points, whereas the lowpass takes the averages between data points. If we remember how bland the highpass block appeared (most of the highpass values being close to zero), it should be clear that if we were to compare the highpass data to the original image, it would have a much larger MSE(\( \hat{\theta} \)).

In any regard, all the lowpass and highpass blocks of the Haar transform return the same independent, identically distributed noise, (mean \( \mu = 0 \) and a noise level equal to the original noise level of \( \sigma \)). Further, the noise level in the highpass data has also had the same 'relative' decrease that the lowpass had (a factor of \( \sqrt{2} \) at each iterative level). Figure 4.2 illustrates some noise before and after a Haar transformation. We've taken a neutral gray image, color value 127 (on a grayscale of 0 to 255) and added noise to it, \( \sigma \varepsilon_i \sim (0, 32) \). We need to note, considering the limitations in rendering grayscale images, that we've taken the same liberties we did before with the clown photo, in adjusting the range of the transformed data so as to best communicate what is going on. You should
agree that the noise does appear to be reduced in all four blocks of the transformed data. It has actually been cut in half, $\sigma/2$.

![Image](image.png)

(a) Original noise  
(b) Haar transformed noise

**Figure 4.2.** Noise before and after a Haar wavelet transformation.

We move now onto the next section, where we finally introduce the all important process of *wavelet shrinkage*.

### 4.4 Wavelet Shrinkage, Part I: Of Ideals, Oracles, & Great Expectations.

Now that we understand how noise generally behaves after a Haar transformation, and how lowpass filtering reduces noise, the final piece of the process, from a denoising perspective, is to explain how we are going to reduce this error in the highpass data. The method that we will use here is called *wavelet shrinkage*. The method was developed in large part by Stanford University professor David Donoho in collaboration with his colleague, professor Jian Johnstone, as well as many others. The 1992 early work of Donoho and Johnstone, *Ideal spatial adaptation by wavelet shrinkage* [4], was the primary paper referenced in the development of this thesis.
We are now going to adapt our notation to reflect the current context of working with the transformed data. We assume the following signal model for our highpass data:

\[ y = g + \sigma \varepsilon, \]

where \( y \) is the highpass observation vector, \( g \) is the perfect highpass vector from sampling our true signal, and \( \sigma \varepsilon \) is the noise vector, \( \varepsilon \sim \mathcal{N}(0, 1) \). Note that this transformed noise has the same variance, \( \sigma^2 \), as the original noise. Component-wise, each \( y_i = g_i + \sigma \varepsilon_i, \ i = 1, \ldots, n. \)

The rationale for calling the method "wavelet shrinkage" is that, (a), the data that will be manipulated, the highpass data, are the coefficients that resulted from applying the wavelet algorithm, and (b), these wavelet coefficients are going to be shrunk down in magnitude, which is the real action that actually accomplishes the denoising. How much these wavelet coefficients will be shrunk, and in what manner they will be shrunk, is determined by a thresholding rule (i.e., a decision function/rule, \( \delta \)).

Thresholding rules can be defined in many ways. There is hard thresholding, soft thresholding, non-negative garrote thresholding, and firm thresholding. These four thresholding rules are defined and graphed in Table 4.1. Let the variable \( y_i \) represents the value of the wavelet coefficient, and \( \lambda \) the value where the thresholding rule takes effect. All of the rules, in some manner, take the wavelet coefficients and reduce their magnitude by some nonnegative amount (maybe zero).

Their subtle differences need to be appreciated. For example, both hard and soft thresholding will take a wavelet coefficient \( y_i \) to zero if \( y_i \leq \lambda \). But if \( y_i > 0 \), then where hard thresholding leaves \( y_i \) alone, soft thresholding decreases \( y_i \)'s magnitude by \( \lambda \). Such differences may be significant in some cases. For example, a hard thresholding treatment of a photograph may cause artifacts. By having such a blunt dropoff will exclude the smaller detail coefficients from the data. Consequently, the softer changes existing in the picture will tend to be homogenized by the hard thresholding process, leaving a gap in the range of potential values for those coefficients. The transitions that will remain will thus jump from nothing to something, looking harsher than before. Soft thresholding won't do this. That does not mean that hard thresholding does not have its applications. Each
Table 4.1: Wavelet shrinkage thresholding functions [5, p. 2].
technique has its strengths and weaknesses which need to be evaluated considering its intended use.

For the purposes of this discussion, hard thresholding will be the rule that we will use, since it's very easy to apply. To be concise, the rule we would simply state that we are either keeping or killing the observed wavelet coefficient $y_i$:

$$
\delta_i^H (y_i) = \begin{cases} 
  y_i & \text{if } |y_i| > \lambda \\
  0 & \text{if } |y_i| \leq \lambda 
\end{cases}
$$

What should be clear in all of these techniques is that the smaller coefficients are being shrunk more so than the larger coefficients. This is a key fact to appreciate. For if we have noise corrupting our signal, it is in the killing of these small wavelet coefficients that we have the best chance of improving our signal. Why? To answer that we have to really appreciate the subtleties of (a) what the highpass data tends to look like, and (b) how noise is affecting our data in general.

Remembering our clown photograph from earlier, we understand that the majority of the highpass data is near zero. Those zeros represent areas where there is much homogeneity in our color. If we lost these small wavelet coefficients, we would really have a difficult time noticing any difference. We really should care mostly about keeping the larger wavelet coefficients. That is where the stark changes in the details hit our eyes.

When it comes to the noise, we need to remember that regardless of whether it is the highpass data or the lowpass data, it is still the same Gaussian white noise. We refer again to our original signal model, $x = \theta + \sigma \epsilon$. Without a signal, the noise has a mean of $\mu = 0$. With a signal, the noise has a mean of $\mu = g_i$. So, if most of our transformed $g_i$ in the highpass data is near zero, it should be easy to connect the fact that most of the noise in the highpass data will have a mean near zero as well. This means that most of the noise in the highpass data is very likely to be eliminated by thresholding. If we can select the right $\lambda$, then we should be able to kill more bad noise than good data between $-\lambda$ and $\lambda$. The question then is: "What should the value of $\lambda$ be?"
Our quadratic loss function using the hard thresholding rule will become

\[ L(g_i, \delta^H_{\lambda \sigma^2}(y_i)) = (g_i - \delta^H_{\lambda \sigma^2}(y_i))^2, \]

where \( \delta^H_{\lambda \sigma^2}(y_i) = \hat{g}_i \) is our estimator of \( g_i \). We recall that the value of the noise variance, \( \sigma^2 \), after the transformation, has not changed in its absolute measure (only in its relative measure, \( \sigma^2/2^j \)). So we should understand that if we do nothing, our \( \text{MSE}(\hat{g}_i) \), for any particular \( g_i \), will just be \( \sigma^2 \). Therefore, when we choose to threshold, we should do so only if we are going to make improvements to that error of \( \sigma^2 \), not, if we should make things worse. Then, in an ideal way, we need the following hard thresholding rule,

\[
\delta^H_{\lambda \sigma^2}(y_i) = \begin{cases} 
  y_i & \text{if } (g_i - y_i)^2 > \sigma^2 \\
  0 & \text{if } (g_i - y_i)^2 \leq \sigma^2 
\end{cases} \quad \text{(keep } y_i) \]

for some \( \lambda \). This means that our mean square error, ideally, will be

\[
\text{MSE}_{\epsilon}(\hat{g}_i) = \begin{cases} 
  E_{\epsilon}(g_i - y_i)^2 = \sigma^2 & \text{if } g_i^2 > \sigma^2 \\
  E_{\epsilon}(g_i - 0)^2 = g_i^2 & \text{if } g_i^2 \leq \sigma^2 
\end{cases} \quad \text{(kill } y_i). 
\]

Thus the ideal \( \lambda \) that we should use, call it \( \lambda^* \), is the actual noise level \( \sigma \). Figure 4.3 graphs this ideal relationship, with \( g_i \) normalized to the noise \( \sigma \).

![Figure 4.3](image-url)  

**Figure 4.3.** MSE\((\hat{g}_i)\) from ideal thresholding, normalized to noise level \( \sigma \).
To achieve this level of quality, though, such an ideal minimization of our MSE, requires complete knowledge of the true highpass signal, $\mathbf{g}$. If we knew that, we wouldn't need to denoise at all. This is why we call this ideal decision function $\delta_\sigma$, with $\lambda^* = \sigma$, an oracle.

Since we really don't have an oracle, it would be good to figure out what our "expected" error will be. For a particular true highpass component $g_i$ (fixed), and all its possible observations, $y_i$, we will have, if we do not threshold, the following mean square error when observing $g_i$:

$$\text{MSE}(g_i - y_i) = \int_{-\infty}^{\infty} (g_i - y_i)^2 f_\sigma(g_i - y_i) d(g_i - y_i).$$

where the probability distribution function of $y_i - g_i$ is determined by the Gaussian white noise (noise level $\sigma$, mean $g_i$):

$$f_\sigma(g_i - y_i) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(g_i - y_i)^2}{2\sigma^2}\right).$$

When using the hard thresholding rule, the MSE$(\hat{g}_i)$ calculation has three regions: (a) below $-\lambda$, where the squared error at any point is $(g_i - y_i)^2$, (b) between $-\lambda$ and $\lambda$, inclusive, where the squared error at any point is $g_i^2$, and (c), above $\lambda$, where the squared error at any point is again $(g_i - y_i)^2$. Letting $t_i = g_i - y_i$ where $t_i$ represents the observation $y_i$'s relative position to the mean $g_i$, we can represent this as:

$$\text{MSE}(\hat{g}_i) = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{-\lambda - \delta_i} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i$$

$$+ \frac{1}{\sigma \sqrt{2\pi}} \int_{-\lambda - \delta_i}^{\lambda - \delta_i} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i$$

$$+ \frac{1}{\sigma \sqrt{2\pi}} \int_{\lambda - \delta_i}^{\infty} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i.$$ 

This calculation of MSE$(\hat{g}_i)$ assumes no "oracular" knowledge of the true $g_i$. An oracle would decide to keep or kill the observation not on the basis of the observed $y_i$, but on the basis of knowing how strong the true $g_i$ is. In the typical non-ideal circumstance,
there may be instances where the hard thresholding rule kills the observation $y_i$ even though the true signal component $g_i$ was above the threshold $\lambda$. Therefore, the above representation of $\text{MSE}(\hat{g}_i)$ is a value in the expected sense. To actually have oracular knowledge of $g_i$, would achieve the following ideal when calculating the $\text{MSE}(\hat{g}_i)$ for a particular $g_i$:

$$
\text{ideal MSE}(\hat{g}_i) = \begin{cases} 
\frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} g_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i & \text{if } g_i \leq \lambda^* \\
\frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{\infty} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i & \text{if } g_i > \lambda^*.
\end{cases}
$$

Since we have set the ideal threshold $\lambda^* = \sigma$, then either $g_i \leq \sigma$ or $g_i > \sigma$. This simply means that the ideal $\text{MSE}(\hat{g}_i)$ for a particular $g_i$ is:

$$\text{ideal MSE}(\hat{g}_i) = \min(g_i^2, \sigma^2).$$

This indicates that the oracle will, under all circumstances, never produce an ideal $\text{MSE}(\hat{g}_i) > \sigma^2$. In fact, if the entire true highpass signal $g$ is stronger than $\sigma$ for all instances $i$, then the ideal $\text{MSE}(\hat{g})$ will be exactly equal to $\sigma^2$. If there are some instances $i$ where the signal $g_i$ is weaker than the noise, then the ideal $\text{MSE}(\hat{g}_i)$ will definitely be less than $\sigma^2$.

The differences between the ideal $\text{MSE}(\hat{g}_i)$ and the expected $\text{MSE}(\hat{g}_i)$ can be seen mathematically by noting the particular regions of integration where these calculations are actually different under the two specific circumstances of the hard thresholding rule. When $g_i \leq \sigma$, over the region of $(-\infty, -\sigma - g_i) \cup (\sigma - g_i, \infty)$, we have $g_i$ in the ideal calculation being less than $t_i$ in the expected calculation, thus,

$$
\frac{1}{\sigma \sqrt{2\pi}} \left[ \int_{-\infty}^{-\sigma-g_i} g_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i + \int_{\sigma-g_i}^{\infty} g_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i \right] < \frac{1}{\sigma \sqrt{2\pi}} \left[ \int_{-\infty}^{-\sigma-g_i} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i + \int_{\sigma-g_i}^{\infty} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i \right].
$$
When $g_i > \sigma$, over the region of $[-\sigma-g_i, \sigma-g_i]$, we have $t_i$ in the ideal calculation being less than $g_i$ in the expected calculation, thus:

$$
\frac{1}{\sigma \sqrt{2\pi}} \int_{-\sigma - g_i}^{\sigma - g_i} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i < \frac{1}{\sigma \sqrt{2\pi}} \int_{-\sigma - g_i}^{\sigma - g_i} g_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i.
$$

Otherwise, the expected and ideal MSE($\hat{g}_i$) accumulate the same square errors. Table 4.2 quickly outlines a comparison of what makes for "ideal" results, and "non-ideal" results.

<table>
<thead>
<tr>
<th>True Signal</th>
<th>Observation</th>
<th>Observer?</th>
<th>Expected Loss</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>g_i</td>
<td>\leq \sigma$</td>
<td>$</td>
<td>y_i</td>
</tr>
<tr>
<td>$</td>
<td>g_i</td>
<td>\leq \sigma$</td>
<td>$</td>
<td>y_i</td>
</tr>
<tr>
<td>$</td>
<td>g_i</td>
<td>&gt; \sigma$</td>
<td>$</td>
<td>y_i</td>
</tr>
<tr>
<td>$</td>
<td>g_i</td>
<td>&gt; \sigma$</td>
<td>$</td>
<td>y_i</td>
</tr>
</tbody>
</table>

**Table 4.2:** Comparison of ideal and non-ideal thresholding decisions.

Shortly we will present a major conclusion that characterizes the expected MSE for all $\hat{g}_i$. First, we want to add a technique that gives us an additional mathematical perspective when looking at these ideal and non-ideal situations. We would like to be able to calculate the expected MSE($\hat{g}_i$) for a distinct subset of probable $y_i$ observations. Be aware that, for the moment, we are focusing more on $y_i$ than $g_i$. To correctly calculate the expected MSE($\hat{g}_i$) over a certain reduced domain, one needs to apply a statistical technique called conditional probability.
Definition 4.1. Let \((\Omega, S, P)\) be a probability space, and let \(B \in S\) with \(PB > 0\). For and arbitrary \(A \in S\) we shall write
\[
P\{A|B\} = \frac{P(A \cap B)}{P(B)},
\]
and call the quantity so defined the *conditional probability* of \(A\), given \(B\). Conditional probability remains undefined when \(P(B) = 0\) [16, p. 28].

Basically we are treating the population that exists in the reduced domain of \(B\) as if it were 100\% of the population. So far, we’ve been using calculations that consider the entire population to be existing between \(-\infty\) to \(\infty\). If we are going to calculate the MSE over a subset of this population, we’ll need to adjust our calculations by a specific factor so that it will treat the analyzed population existing between two limits as if it were everything. So we just simply divide our regular result by the actual percentage of the population that exists between those two limits. So, mathematically, let
\[
P\{a < T_i \le b\} = \tau_{(a,b)} = \frac{1}{\sigma \sqrt{2\pi}} \int_{a}^{b} \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i
\]
be the percent of the population that exists between \(a\) and \(b\). If we now incorporate \(\tau\) into our calculating of the expected MSE\((\hat{g}_i)\) between \(a\) and \(b\), we’ll get the correct value (otherwise we would only be getting the share of the total squared error that exists between \(a\) and \(b\)):
\[
MSE(\hat{g}_i)_{(a,b)} = \frac{1}{\tau_{(a,b)}} \cdot \frac{1}{\sigma \sqrt{2\pi}} \int_{a}^{b} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i.
\]
For example, in the domain of \((-\infty,-3.5\sigma) \cup (-1.5\sigma,\infty)\), we would get
\[
MSE(\hat{g}_i)_{(-\infty,-3.5\sigma) \cup (-1.5\sigma,\infty)}
= \frac{1}{\sigma \sqrt{2\pi}} \cdot \left(\int_{-\infty}^{-3.5\sigma} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i + \int_{-1.5\sigma}^{\infty} t_i^2 \exp\left(-\frac{t_i^2}{2\sigma^2}\right) dt_i\right) \cdot \frac{1}{\tau_{(-\infty,-3.5\sigma)} + \tau_{(-1.5\sigma,\infty)}}
= 0.795139\sigma^2.
\]
We now present one of our major examples that illustrates all the different perspectives on the expected MSE, further highlighting differences between the ideal or non-ideal results.

Example 4.1. Let \( g_i = 1.5\sigma \). Then the oracle would ideally keep this \( g_i \) regardless of \( y_i \). Realistically, we may have the result that the observed \( y_i \) is less than \( \sigma \), and so would be non-ideally thresholded. Figure 4.4 illustrates this pictorially with a noise distribution overlaying the signal vector. The dark gray areas indicate the region where \( y_i \) observations would be thresholded.

**Figure 4.4.** Signal vector \( g_i = 1.5\sigma \) overlaid with a distribution of noisy observations \( y_i \).
Letting

\[ f_\sigma(t) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-t^2}{2\sigma^2}\right), \]

then for \( g_i = 1.5\sigma \) the expected mean square error of \( \hat{g}_i \) is:

\[
\text{expected MSE}(\hat{g}_i) = \int_{-\infty}^{-1.5\sigma} t_i^2 f_\sigma(t_i) dt_i + \int_{1.5\sigma}^{\infty} (1.5\sigma)^2 f_\sigma(t_i) dt_i + \int_{1.5\sigma}^{\infty} t_i^2 f_\sigma(t_i) dt_i
\]

\[
= 0.050030\sigma^2 + 0.680238\sigma^2 + 0.515430\sigma^2
\]

\[
= 1.245700\sigma^2.
\]

So our expected MSE for \( \hat{g}_i \) is more than \( \sigma^2 \), which the oracle would never let happen (the ideal MSE(\( \hat{g}_i \)) being \( \sigma^2 \) since \( g_i > \sigma \)). This is the reality of expectation. If we instead use conditional probability to look at the individual ideal and non-ideal sub-regions, we would see results more representative of the theory. In the ideal thresholding region, \(|y_i| > \sigma \), the MSE(\( \hat{g}_i \)) is 0.810496\( \sigma^2 \). That is less than both \( \sigma^2 \) and \( g_i = 2.25\sigma^2 \).

Looking where the thresholding is not ideal, \(|y_i| \leq \sigma \), then the MSE(\( \hat{g}_i \)) is exactly 2.25\( \sigma^2 \).

Table 4.3 provides a summary of this and similar results for various values of \( g_i \).

| True Signal | Oracle? | ideal MSE | expected MSE | expected MSE, \(|y| \leq \sigma | \langle \rangle ? | expected MSE, \(|y| > \sigma |
|-------------|--------|-----------|--------------|-----------------|-----------------|
| 2.5\( \sigma \) | keep | 1.00\( \sigma^2 \) | 1.16\( \sigma^2 \) | 6.25\( \sigma^2 \) | > | 0.80\( \sigma^2 \) |
| 2.0\( \sigma \) | keep | 1.00\( \sigma^2 \) | 1.24\( \sigma^2 \) | 4.00\( \sigma^2 \) | > | 0.73\( \sigma^2 \) |
| 1.5\( \sigma \) | keep | 1.00\( \sigma^2 \) | 1.25\( \sigma^2 \) | 2.25\( \sigma^2 \) | > | 0.81\( \sigma^2 \) |
| 1.0\( \sigma \) | kill | 1.00\( \sigma^2 \) | 1.11\( \sigma^2 \) | 1.00\( \sigma^2 \) | < | 1.21\( \sigma^2 \) |
| 0.5\( \sigma \) | kill | 0.25\( \sigma^2 \) | 0.90\( \sigma^2 \) | 0.25\( \sigma^2 \) | < | 1.99\( \sigma^2 \) |
| 0.0\( \sigma \) | kill | 0.00\( \sigma^2 \) | 0.80\( \sigma^2 \) | 0.00\( \sigma^2 \) | < | 2.53\( \sigma^2 \) |

**Table 4.3:** MSE calculations for various individual signal strengths, \( g_i \).
We now present one of our primary results: an illustration of how the *expected* MSE($\hat{g}_i$) performs in comparison to the oracle's *ideal* MSE($\hat{g}_i$). We start by generally restating the equation used in Example 4.1, but now for an arbitrary $g_i$:

$$\text{expected MSE}(\hat{g}_i; g_i) = \int_{-\infty}^{-\sigma-g_i} t_i^2 f_\sigma(t_i)dt_i + \int_{-\sigma-g_i}^{\sigma-g_i} (g_i)^2 f_\sigma(t_i)dt_i + \int_{\sigma-g_i}^{\infty} t_i^2 f_\sigma(t_i)dt_i.$$  

Figure 4.5 graphs this result (normalizing the $g_i$'s to an arbitrary noise level $\sigma$). Note that the oracle's *ideal*, as defined, will always outperform the expected result for any $g_i$. The dotted lines have been added in as a representation of a continuation of the respective parts of the oracle's hard thresholding rule.

**Figure 4.5.** Comparison of expected MSE($\hat{g}_i$) to ideal MSE($\hat{g}_i$), thresholding at $\lambda = \sigma$.

It is interesting to see that for those $g_i$'s near zero and near $2\sigma$, our expectations are far from ideal. But near $1\sigma$ and greater than $4\sigma$, they are pretty close to ideal. We also would like to point out that this particular graph was based on choosing our threshold value for $\lambda$ to be equal to that theoretical ideal of noise strength, $\sigma$, being the best $\lambda$. Then a question to ask is, what would happen if we changed our $\lambda$ to a supposedly non-ideal value like $1.2\sigma$, or $0.8\sigma$?
We begin to answer this question by first generalizing the equation in (1) further, but now for arbitrary $\lambda$ as well as $g_i$:

\[
\text{expected } \text{MSE}(\hat{g}_i; g_i, \lambda) = \int_{-\lambda-g_i}^{\lambda-g_i} t_i^2 f_\sigma(t_i) dt_i + \int_{-\lambda-g_i}^{\lambda-g_i} (g_i)^2 f_\sigma(t_i) dt_i + \int_{-\lambda-g_i}^{\lambda-g_i} t_i^2 f_\sigma(t_i) dt_i.
\]

Figure 4.6 graphs this result for $\lambda = 1.2\sigma$ and $0.8\sigma$. You can notice that if we increase $\lambda$, we increase the expected error for those $g_i$'s with a strength above $0.7\sigma$, approximately. The same increase in $\lambda$ decreases the error for those $g_i$'s weaker than $0.7\sigma$. In the other direction, if we decrease $\lambda$, we decrease our errors on $g_i$ stronger than $0.7\sigma$, but increase the error on those weaker than $0.7\sigma$. You can extend these trends to where you would increase $\lambda$ to $\infty$ and kill everything ($\text{MSE}(\hat{g}) = \frac{1}{n} \sum g_i^2$), or decrease $\lambda$ to 0 and keep everything ($\text{MSE}(\hat{g}) = \sigma^2$). This brings up another question. Is the noise level $\sigma$ truly the ideal threshold to set $\lambda$ to?

![Figure 4.6](image)

**Figure 4.6.** Effects on expected MSE($\hat{g}_i$) when changing threshold:

$\lambda = 0\sigma, 0.8\sigma, 1.0\sigma, 1.20\sigma, \infty\sigma$. 

79
Ultimately, the performance of a particular $\lambda$ hard-thresholding rule will depend on two things: (a) the PDF of the $g_i$'s in the signal you are trying to improve, and (b) the overall strength of your signal in comparison to the noise. The implication of (b) should be clear. The stronger your signal is, the lower you should set your threshold, $\lambda$, in order to gain the best (i.e. lowest) overall expected $\text{MSE} (\hat{g})$. The implication of (a) is not so clear and will be discussed later.

Before we leave this section, we would like to parse the analysis of the errors a little bit more. We do so through the use of a more advanced oracle, one which was conceived by the author of this thesis. Whereas the original oracle presented does know where each component of the true signal is, it does not know the strength of any particular instance of noise. It only knows the general expected variance of the noise. We can design a stronger oracle than this by adding the knowledge of total noise condition. We would then have the following oracle, $O_2$, thresholding rule,

$$\delta^{O_2}(y_i) = \begin{cases} y_i & \text{if } (g_i - y_i)^2 \leq (g_i - 0)^2 \quad \text{(keep } y_i) \\ 0 & \text{if } (g_i - y_i)^2 > (g_i - 0)^2 \quad \text{(kill } y_i) \end{cases}$$

where giving a $\lambda$ really is longer necessary. Figure 4.7 illustrates, via the graph of a sine function, what is going on with the new oracle $O_2$, and how it compares to the original oracle $O_1$ (oracle $O_1$ is given a $\lambda = \frac{1}{2}$).
Figure 4.7. Comparison of two differently defined oracles.
We can calculate a new ideal MSE using our oracle\(_2\) (note: the following calculation is valid for all \(g_i \geq 0\). For \(g_i < 0\), we would need to change the signs on the bounds of integration that involve \(g_i\)):

\[
\text{ideal MSE}_{\text{oracle}_2}(\hat{\phi}_i; g_i) = \int_{-\infty}^{g_i} g_i^2 f(t_i) dt_i + \int_{-g_i}^{g_i} t_i^2 f(t_i) dt_i + \int_{g_i}^{\infty} g_i^2 f(t_i) dt_i.
\]

We can then include this result with our earlier graphs of ideal and expected MSE. You can see in Figure 4.8 how ideal the new oracle\(_2\) performs in comparison to oracle\(_1\).

\[\text{Figure 4.8.} \text{ Comparison of oracle}_2\text{'s ideal MSE(}\hat{\phi}_i\text{) to oracle}_1\text{ and previous expectations.}\]

Next, we take oracle\(_2\), and overlay the expected decisions of a hard thresholding rule, set to \(\lambda = \frac{1}{2}\) (see Figure 4.9). We can delineate zones where the expected thresholding of the \(y_i\) observations would give us a good result (ideal), and other zones where it would give us a bad result (not ideal).
Let us now take a close up look at two $g_i$'s, see Figure 4.10. One will be $g_i$ such that $2g_i < \lambda$, and the other one will $g_k$ such that $2g_k > \lambda$.

Figure 4.9. Oracle$_2$ overlaid with the expectations of a hard thresholding rule, $\lambda = \frac{1}{2}$.

Figure 4.10. Close-up view of oracle$_2$, hard thresholding, and two $g_i$'s, $2g_j \leq \lambda$ and $2g_k > \lambda$. 
Looking closely at these two $g_i$’s, think about how increasing the threshold will affect our expected MSE calculations. Does increasing the threshold $\lambda$ improve the error for a particular $g_i$, or not? To answer that question, consider the following two calculations of the expected MSE (one for each $g_i$), each of which are partitioned into the five regions depicted in Figure 4.10:

$$MSE(\hat{g}_k) = \int_{-\infty}^{-g_k} t_k^{-2} f(t_k) dt_k + \left\{ \int_{-g_k}^{-g_k+\lambda} g_k^{-2} dt_k + \int_{-g_k+\lambda}^{-g_j} g_j^{-2} dt_k \right\} + \int_{-\infty}^{-g_k} t_k^{-2} f(t_k) dt_k,$$

where $\lambda$ increases good results

$\begin{align*}
\text{increasing } \lambda \text{ increases good results} & \quad \text{increasing } \lambda \text{ increases bad results}
\end{align*}$

$$MSE(\hat{g}_j) = \int_{-\infty}^{-g_j} t_j^{-2} f(t_j) dt_j + \left\{ \int_{-g_j}^{-g_j+\lambda} g_j^{-2} dt_j + \int_{-g_j+\lambda}^{-g_i} g_i^{-2} dt_j \right\} + \int_{-\infty}^{-g_j} t_j^{-2} f(t_j) dt_j,$$

where $\lambda$ increases good results

$\begin{align*}
\text{increasing } \lambda \text{ increases good results} & \quad \text{increasing } \lambda \text{ increases bad results}
\end{align*}$

We should notice that for $g_k$, where $2g_k > \lambda$, an increase in the threshold trades "good" error for "bad" on one end (increases error), and trades "bad" error for "good" on the other (decreases error). We should also notice that once the threshold jumps past $2g_i$ (the case of $g_j$) any increase in $\lambda$ only reduces the error. In Figure 4.11 we present a graph of how the $MSE(\hat{g}_i)$ changes in response to an increasing $\lambda$. When $\lambda = 0$, it’s understandable that the $MSE(\hat{g}_i)$ for any $g_i$ is equal to $\sigma^2$, the noise. What we found is that for any $g_i > 0.7096\sigma$, increasing $\lambda$ first increases the MSE up to some peak amount, and then decreases it down to a value equal to $g_i^2$ (a complete quadratic loss of that $g_i$).
In Figure 4.12 we take a closeup look at one of our earlier graphs (Figure 4.6, which demonstrated how the expected $\text{MSE}(\hat{g}_i)$ is effected by a changing $\lambda$). Looking at a few marked locations for a $g_i = 0.77\sigma$, we point out the evidence of an increasing, and then decreasing error. Note how the error increases as $\lambda$ increases from $0.0\sigma$ to $1.0\sigma$, but then decreases as $\lambda$ increases from $1.0\sigma$ to $1.2\sigma$ (the peak actually occurs near $\lambda = 0.99\sigma$).
The implication of all this is that thresholding cannot help any \( g_i \) that is stronger than \( \sigma \). For any \( g_i \) less than or equal to \( \sigma \), but greater than \( 0.7096\sigma \), thresholding does reduce the error, but only if \( \lambda \) is sufficiently large enough. For anything below \( 0.7096\sigma \), any \( \lambda \) will reduce the error.

We'd like to mention that the determination of "0.7096" was made in Mathematica. Through recursive calculations we looked for the \( g_i \) whose maximum MSE did not exceed \( 1.0\sigma^2 \). No theoretical assessment of the exact value has been made. That this value is near \( 1/\sqrt{2} \), 0.7071, is a curiosity.

These particulars are being pointed out to bring across the idea that finding the best \( \lambda \) amounts to balancing the amount of "bad" error you would like to kill against the "good" error you would rather keep. As mentioned before, where the best \( \lambda \) may fall will be determined by the PDF of the particular signal's \( g_i \)'s. We will demonstrate this soon in section 4.6, but first we would like to present one more perspective on the theoretical ideals of wavelet shrinkage.

4.5 Wavelet Shrinkage, Part II: A Tale of Two Errors

We concluded the last section with a discussion of how increasing the value of the hard threshold \( \lambda \) can be looked upon as a trade-off between good errors and bad errors. We should be careful to realize that this was occurring strictly within the limited context of the highpass data. To fully assess the entire process, we also need to consider the errors that exist within the lowpass data as well. We recall that the noise level in the lowpass data is \( \sigma^2/2 \). If we were to replace our entire data set with just the lowpass data, then we would have reduced our noise down to \( \sigma^2/2 \). We also recall, from our first discussion on wavelets, that the entire original data set, including all the noise, is recoverable by a synthesis via an inverse Haar transform. That, of course, would only return us to the original noise level, \( \sigma^2 \). In between these two extremes lies the compromise of thresholding that we just introduced in the last section.
In David Donoho's paper, he refers to thresholding as "selective wavelet reconstruction". When we synthesize our modified data set back together, we do so by only using some of the highpass wavelet coefficients, i.e., not thresholding those values. In the other locations, where we do threshold the wavelet coefficients to zero, we are simply using the lowpass scaling coefficients to fill things in. Our selections will be based on our decision function, \( \delta^H_\lambda \), using some thresholding criteria, \( \lambda \). Intuitively, one can see that we are either accepting the original noise error by not thresholding, or accepting the reduced noise error by thresholding (lowpass data). i.e., \( \sigma^2 \) or \( \sigma^2/2 \).

This brings us to an interesting observation. We cannot actually get our \( \text{MSE}(\hat{\theta}_i) \) any lower than what the lowpass data can already offer. That implies our \( \text{MSE}(\hat{\theta}_i) \) is going to end up somewhere between \( \sigma^2 \) and \( \sigma^2/2 \). This is quite contrary to the idea that reducing our error would be the only goal here. What we need to realize is that a successful denoising process balances the desire of noise reduction against the need of information preservation. If we simply wanted to get the most noise reduction, we would just iterate the lowpass algorithm down to one data point, giving us the single value that represents the mean value of all the original data points. Theorem 3.8 suggests that this value would give us the least amount of MSE overall, but we surely understand that one data point cannot really tell us anything.

So, what value should we select for our thresholding value, \( \lambda \)? Based on our earlier discussion of an oracle, the ideal thresholding value should be the noise level \( \sigma \). Let’s analyze this suggestion from the perspective of the lowpass data. First we review what we have so far with regards to the error from noise in the original and transformed data. Given true data components \( \theta_i \) and \( \theta_j \), we have the following MSE's:

\[
\begin{align*}
\text{MSE}(x_i) &= E((\theta_i - x_i)^2) = \sigma^2, \text{ the original data error}, \\
\text{MSE}(h_i) &= E\left(\frac{\theta_j + \theta_k}{\sqrt{2}} - \frac{x_j + x_k}{\sqrt{2}}\right) = \sigma^2, \text{ the lowpass data error}, \\
\text{MSE}(g_i) &= E\left(\frac{-\theta_j + \theta_k}{\sqrt{2}} - \frac{-x_j + x_k}{\sqrt{2}}\right) = \sigma^2, \text{ the highpass data error}.
\end{align*}
\]
We know that if we do not threshold, that we would just get \( \text{MSE}(x_i) \) again. Then the question we come to now is this: What would our error be on \( \theta_i \) if we do threshold? To answer that we will need to evaluate the following:

\[
\text{MSE}(\hat{\theta}_i) = E\left(\hat{\theta}_i - \frac{x_i + x_j}{2}\right)^2.
\]

We do not want the above error to be any worse than our original \( \zeta^2 \). In order to see what situation would cause this \( \text{MSE}(\hat{\theta}_i) \) to exceed \( \zeta^2 \), we evaluate the following integral:

\[
E\left(\hat{\theta}_i - \frac{X_i + X_j}{2}\right)^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\theta_i - \frac{x_i + x_j}{2}\right)^2 f_{x_i, x_j}(x_i, x_j) \, dx_i \, dx_j,
\]

where \( f_{x_i, x_j}(x_i, x_j) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^2 \exp\left(-\frac{1}{2} \left[ \frac{(x_i - \theta)^2}{\sigma^2} + \frac{(x_j - \theta)^2}{\sigma^2} \right]\right) \),

is the joint PDF for two i.i.d. normal RVs, \( X_i \sim \mathcal{N}(\theta_i, \sigma^2) \) and \( X_j \sim \mathcal{N}(\theta_j, \sigma^2) \).

We start off by expanding the expression of the value being evaluated for expectation:

\[
\left(\hat{\theta}_i - \frac{x_i + x_j}{2}\right)^2 = \theta_i^2 - 2\theta_i \frac{x_i + x_j}{2} + \left(\frac{x_i + x_j}{2}\right)^2
= \theta_i^2 - \theta_i x_i - \theta_i x_j + \frac{x_i^2}{4} + \frac{x_i x_j}{2} + \frac{x_j^2}{4}.
\]

Next, because we have the independence of \( X_i \) and \( X_j \), we can split their joint PDF and evaluate the following integral, term by term:

\[
\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\theta_i^2 - \theta_i x_i - \theta_i x_j + \frac{x_i^2}{4} + \frac{x_i x_j}{2} + \frac{x_j^2}{4}\right) e^{-\frac{(x_i - \theta_i)^2}{2\sigma^2}} e^{-\frac{(x_j - \theta_j)^2}{2\sigma^2}} \, dx_i \, dx_j.
\]

Most of the work in evaluating this integral will be pretty straight forward. All of the integrals of the individual terms will jump out right way as being the expected value of...
that term. Many of those results we can use just as they are. Two of the terms, though, will require us to use a property that we stated back in Theorem 3.6. That

\[ Var(X) = E(X^2) - E(X)^2. \]

This will allow us to use the equality of \( E(X^2) = \sigma^2 + E(X)^2 \) in a substitution, (recalling that \( Var(X) = \sigma^2 \)).

We now evaluate, for purpose of limited exposition, just two of the terms. Afterwards, we’ll state the entire result. The first term we evaluate is \( \frac{x_i x_j}{2} \):

\[
\left( \frac{1}{\sqrt{2\pi\sigma}} \right)^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \frac{x_i x_j}{2} \right) \ e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} \ e^{-\frac{(x_j-\theta)^2}{2\sigma^2}} \ dx_j dx_i
\]

\[ = \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} (x_i) e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} \left( \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} (x_j) e^{-\frac{(x_j-\theta)^2}{2\sigma^2}} \ dx_j \right) \ dx_i
\]

\[ = \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} (x_i) e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} (\theta_j) \ dx_i
\]

\[ = \frac{1}{2} \cdot \theta_j \cdot \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} (x_i) e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} \ dx_i
\]

\[ = \frac{1}{2} \cdot \theta_j \cdot \theta_i. \]

The next term we evaluate is \( \frac{x_i^2}{4} \):

\[
\left( \frac{1}{\sqrt{2\pi\sigma}} \right)^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \frac{x_i^2}{4} \right) \ e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} \ e^{-\frac{(x_j-\theta)^2}{2\sigma^2}} \ dx_j dx_i
\]

\[ = \frac{1}{4} \cdot \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} (x_i^2) e^{-\frac{(x_i-\theta)^2}{2\sigma^2}} \left( \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} e^{-\frac{(x_j-\theta)^2}{2\sigma^2}} \ dx_j \right) \ dx_i
\]

\[ = \frac{1}{4} \cdot E(X_i^2) = \frac{1}{4} \cdot \left( \sigma^2 + E(X_i)^2 \right)
\]

\[ = \frac{1}{4} \left( \sigma^2 + \theta_i^2 \right). \]
We now present the final result of resolving the entire integral:
\[
\left( \frac{1}{\sqrt{2\pi\sigma}} \right)^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \theta_i^2 - \theta_j x_i - \theta_x j + x_i^2 + x_j^2 + x_j^2 \right) e^{-\frac{(x_i - \theta_j)^2}{2\sigma^2}} e^{-\frac{(x_j - \theta_i)^2}{2\sigma^2}} \, dx_j \, dx_i
\]
\[
= \theta_i^2 - \theta_j^2 - \theta_i \theta_j + \frac{1}{4} \left( \sigma^2 + \theta_j^2 \right) + \frac{1}{2} \theta_i \theta_j + \frac{1}{4} \left( \sigma^2 + \theta_i^2 \right)
\]
\[
= \frac{1}{4} \left( \sigma^2 + \theta_i^2 \right) - \frac{1}{2} \theta_i \theta_j + \frac{1}{4} \left( \sigma^2 + \theta_j^2 \right)
\]
\[
= \frac{1}{4} \left( \sigma^2 + \theta_i^2 - 2 \theta_i \theta_j + \sigma^2 + \theta_j^2 \right)
\]
\[
= \frac{1}{4} \left( 2\sigma^2 + (\theta_i - \theta_j)^2 \right).
\]

This last quantity is greater than \(\sigma^2\) when \(|\theta_i - \theta_j|\) is greater than \(\sqrt{2} \sigma\). If \(\theta_i - \theta_j = 0\), then this value achieves a minimum of \(\frac{1}{2} \sigma^2\), which happens to be the exact variance of the noise in the lowpass data. Thus
\[
E \left( \theta - \frac{X_i + X_j}{2} \right)^2 \leq \sigma^2 \text{ when } |\theta_i - \theta_j| \leq \sqrt{2} \sigma.
\]

This implies that the oracle can reduce the error by thresholding whenever
\[
|g_i| \leq \sigma,
\]
where \(g_i = (\theta_i - \theta_j) / \sqrt{2}\), the highpass coefficient. Thus the oracle's chooses \(\sigma\) as its ideal threshold \(\lambda^*\).

The two situations of "thresholding" and "not thresholding" are presented pictorially in Figures 4.13 and 4.14, respectively.
Within the framework of thresholding the highpass data, the oracle minimizes \( \text{MSE}(\hat{\theta}) \) at \( \sigma \). Setting \( \lambda \) any lower will only give us more \( \theta \)'s with an error equal to \( \sigma^2 \), instead of something less. Setting it any higher will actually give us errors greater than \( \sigma^2 \). In the end, the oracle will give us an \( \text{MSE}(\hat{\theta}) \) somewhere in between \( \sigma^2/2 \) or \( \sigma^2 \). We
must remember though that these results are ideal because their the oracle's results. They are not actually the expected result. We illustrate this issue with our example in the next section.

4.6 Illustrative Example: Denoising a Corrupted Sine Function.

We are now going to demonstrate our denoising techniques by trying to clean up a corrupted sine function. First, we discuss some of the probabilistic aspects of the sine function. We consider the standard sine function, $f(x) = \sin x$, defined on the unit circle, $0 \leq x < 2\pi$ and $-1 \leq \sin x \leq 1$. We treat the domain of sine, $X$, as a random variable that is uniformly distributed on $0$ to $2\pi$:

$$f(x) = \begin{cases} \frac{1}{2\pi} & 0 \leq x < 2\pi \\ 0 & \text{otherwise.} \end{cases}$$

Then $\sin X$ is a function of the RV $X : f(X) = Y = \sin X$. The DF of $Y$ is given by:

$$F(y) = P\{Y \leq y\} = P\{\sin X \leq y\}, \quad -1 < y < 1.$$

![Figure 4.15](image)

**Figure 4.15.** $y = \sin x$, $0 \leq x \leq 2\pi$.

There are two sub-regions in the domain where $\sin X$ satisfies the DF above for any $y$, see Figure 4.15. Also, we need to consider two different circumstances for the $y$'s: (a) when they are negative, and (b) when they are non-negative:
\[ F(y) = \begin{cases} 
P \left\{ 0 \leq X \leq \sin^{-1} y \right\} \cup \left\{ \left[ \pi - \sin^{-1} y \right] \leq X \leq 2\pi \right\} & \text{if } 0 \leq y \leq 1 \\
\frac{1}{2\pi} \int_0^{\sin^{-1} y} dx + \frac{1}{2\pi} \int_{\pi - \sin^{-1} y}^{2\pi} dx & \text{if } -1 \leq y < 0 
\end{cases} \]

We thus calculate:

\[
F(y) = \begin{cases} 
\frac{\sin^{-1} y}{2\pi} + \frac{\pi}{2\pi} - \frac{\pi - \sin^{-1} y}{2\pi} & \text{if } 0 \leq y \leq 1 \\
\frac{\sin^{-1} y + 2\pi}{2\pi} - \frac{\pi + \sin^{-1} y}{2\pi} & \text{if } -1 \leq y < 0 
\end{cases} \]

Applying Theorem 3.2, we determine that the probability density function of \( \sin X \) is:

\[
(1) \quad f(y) = \frac{d}{dy} \left( \frac{\pi + 2\sin^{-1} y}{2\pi} \right) = \frac{d}{dy} \left( \frac{\sin^{-1} y}{\pi} \right) = \frac{1}{\pi \sqrt{1 - y^2}} \quad \text{for } |y| \leq 1.
\]

Figure 4.16 graphs the result from (1).
Then the variance of the sine function is calculated as follows:

\[
\text{Var}(\sin X) = E(Y^2) = \int_{-1}^{1} y^2 f(y) dy
\]

\[
= \frac{1}{\pi} \int_{-1}^{1} \frac{y^2}{\sqrt{1-y^2}} dy = \frac{2}{\pi} \int_{0}^{1} \frac{y^2}{\sqrt{1-y^2}} dy
\]

Letting \( y = \sin u \), then \( u = \sin^{-1} y, y^2 = \sin 2u = (1 - \cos 2u)/2 \), and we get

\[
\text{Var}(\sin X) = \frac{2}{\pi} \int_{0}^{\frac{\pi}{2}} \frac{\sin u}{2} \int_{\sin^{-1}0}^{\sin^{-1}1} \frac{1-\cos 2u}{2} du
\]

\[
= \frac{1}{\pi} \cdot \frac{\pi}{2} - \frac{1}{\pi} \cdot \int_{0}^{\frac{\pi}{2}} \cos 2u \ du
\]

Letting \( v = 2u \), then \( dv = 2du \), and we conclude that

\[
\text{Var}(\sin X) = \frac{1}{\pi} \cdot \left( \frac{\pi}{2} \right) - \frac{1}{\pi} \cdot \int_{0}^{\pi/2} \cos v \ dv
\]

\[
= \frac{1}{2} - \frac{1}{\pi} \cdot \sin v \bigg|_{0}^{\pi/2} = \frac{1}{2}.
\]

Next we consider the sine function in Gaussian white noise. Let \( y(x) = \sin(x) + \sigma \varepsilon(x) \), where the noise \( \varepsilon \sim \mathcal{N}(0,1) \) with PDF \( \varphi(y) \). The probability density function, \( g(y) \), of this process is the convolution integral of the individual density functions \([2, \text{p. 61}]\).

We set the bounds to be \(-1\) to \(1\) since PDF of the sine function is zero outside these bounds:
\[ g(y) = [f * \varphi](y) = \int_{-1}^{1} f(x) \varphi(y-x) \, dx = \int_{-1}^{1} \frac{1}{\pi \sqrt{1-x^2}} \cdot \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(y-x)^2}{2\sigma^2} \right) \, dx \]

\[ = \frac{1}{\sigma \pi \sqrt{2\pi}} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \cdot \exp \left( -\frac{(y-x)^2}{2\sigma^2} \right) \, dx. \]

We let \( x = \cos u \). Then \( dx = -\sin u \, du \) and \( \cos^{-1}x = u \), thus \( \cos^{-1}(-1) = \pi \), and \( \cos^{-1}(1) = 0 \). Also, \( x^2 = \cos^2 u = 1 - \sin^2 u \), thus \( 1 - x^2 = \sin^2 u \). Substituting these equalities into the equation above, we get:

\[ g(y) = [f * \varphi](y) = \frac{1}{\sigma \pi \sqrt{2\pi}} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \cdot \exp \left( -\frac{(y-x)^2}{2\sigma^2} \right) \, dx \]

\[ = \frac{1}{\sigma \pi \sqrt{2\pi}} \int_{0}^{\pi} \frac{1}{\sqrt{\sin^2 u}} (-\sin u) \cdot \exp \left( -\frac{(y-\cos u)^2}{2\sigma^2} \right) \, du. \]

Because \( \sin u \geq 0 \) on \([0, \pi]\), we can simplify:

\[ g(y) = [f * \varphi](y) = \frac{1}{\sigma \pi \sqrt{2\pi}} \int_{0}^{\pi} \exp \left[ -\frac{(y-\cos u)^2}{2\sigma^2} \right] \, du. \]

In Figure 4.17, we graph this result for various noise levels \( \sigma \).

\[ \text{Figure 4.17. PDFs of various noisy sine functions.} \]
We now come to our feature example, where we take a discrete sampling of a noisy sine function ($n = 256$) and denoise it through our process of using the Haar Wavelet Transformation, the hard thresholding rule, and $\lambda$ set to the theoretical ideal of noise level $\sigma$. Figure 4.18 shows the before and after of noisy sine function.

![Figure 4.18](image)

(a) noisy sine function, $\text{MSE}(y) = \sigma^2 = 0.010$.

(b) denoised sine function, $\text{MSE}(\hat{y}) = 0.006$.

**Figure 4.18.** The denoising of a corrupted sine function, before and after.

What about other values of $\lambda$? Is $\sigma$ really the best $\lambda$? We now take the previous example and vary $\lambda$ for various levels of noise $\sigma$, looking for the one value that gives us the minimum MSE for each noise level. The results are graphed in Figure 4.19.
What we notice is that for low levels of noise, the best $\lambda$ for denoising is below the noise level $\sigma$. Then as the noise level increases, the necessary $\lambda$ increases relatively as well. It jumps above the noise level, to the point where the noise gets so strong that $\lambda$ needs to annihilate the entire highpass data. How can we explain this?

If we go back to our earlier result of the expected MSE, we can determine what the expected results for various $\lambda$'s should be. To accomplish this we take the inner product integral of the function that describes the expected MSE (for each $\lambda$), and the function that describes the density of the highpass data. That brings us to a question. What is the PDF of the highpass data of a sine function?

In Figure 4.20 we show the graph of the lowpass and highpass data of a clean sine function. What you should notice is that the highpass data looks like a graph of a cosine function, only with much smaller amplitude. It is a reasonable assumption that highpass algorithm operates like the derivative on our original data (which is a function). We just need to recall that the highpass algorithm calculates the differences between two adjacent data points. Taking that knowledge, along with the basic understanding of how the derivative of a continuous function is the result of a limiting process on the difference between close neighbors, then we should feel comfortable making this assumption. If so,
Figure 4.20. The lowpass and highpass data of a clean sine function.
then we can say that the highpass data is described by the PDF of the cosine function. It so happens that the cosine function has the same PDF as the sine function (we only need to adjust it for the smaller amplitude).

We are now ready to calculate the inner product of these two functions for various \( \lambda \)'s and noise level \( \sigma \)'s. A few of these results are plotted in Figure 4.21, along with corresponding plots of how the oracle would perform under a changing \( \lambda \) (dashed lines). The only assumption we make on the highpass data, so as to make the observations easy to see, is that the amplitude of the cosine function is 1.

![Figure 4.21](image.png)

**Figure 4.21.** Plots of the MSE for various levels of noise \( \sigma \), as \( \lambda \) varies from 0 to 4\( \sigma \).

What we notice now is that the best \( \lambda \) for those noise levels below \( \sigma^2 = \frac{1}{2} \) is zero, and that the best \( \lambda \) for those noise levels above \( \sigma^2 = \frac{1}{2} \) is \( \infty \). This is not quite in agreement with our experimental results from earlier, though it is somewhat close to it. In our experiment, the best \( \lambda \) does stay relatively low up onto the point that it reaches the variance of the highpass data. After that point, it does eventually take off to completely annihilate the highpass data. If we were to actually run our experiment for longer periods (i.e., more cycles of the sine function), our results would eventually approach our theoretical expected result. For what we have actually created here is a stochastic
process. It's just that we've only done one run of the process. If we do ever longer runs of the experiment, then the stochastic process will approach its theoretical expected value.

There is another observation to make here. It looks like our \( \lambda \) should never be set to the theoretical ideal of \( \sigma \). What we need to realize is that the PDF of the cosine function, see Figure 4.22 (a), is very heavy on the values that are close to its amplitude. Otherwise, it's very light, especially near zero. If we then look at how the expected MSE graph behaves, see Figure 4.22 (b), we should realize that we need the PDF of the highpass data to be heavier towards the values that are below 0.71\( \sigma \) in order to have a net reduction of the MSE.

![Figure 4.22](image-url). PDF of the cosine function. The expected MSE(\( \hat{g}_i \)) of the highpass data.

One final observation to make about this illustrative example is that once the variance of the noise (i.e., the noise level) is equal to the variance of the signal's highpass data, \( \sigma^2 = \text{Var}(\cos X) = \frac{1}{2} \), then the highpass signal has essentially been lost to the noise.
4.7 Summary & Conclusions

Through the course of researching for this discussion, there have been many interesting realizations made about the topics of statistics, wavelets, and their applications. The sheer volume and depth of knowledge that is available to read on these subjects is quite astounding.

The major observations made here are that ideals, general expectations, and stochastic processes do not necessarily lead to the same conclusions. In Section 4.4 we demonstrated through equation (1) and Figure 4.5 how the theoretical ideal MSE is often far lower than the generally expected MSE, when we applied the hard thresholding rule onto the highpass data. In equation (2) and Figure 4.6 we extended this result to observe that varying our choice of threshold $\lambda$ will affect the expected MSE in varying ways, depending on the particular $g_i$ being evaluated. In equation (3) and Figure 4.7 we introduced a new type of oracle for the simple purpose of illustrating how changing $\lambda$ will affect the expected MSE calculation for any particular $g_i$ (see Figures 4.10, 4.11, and 4.12). The larger point of all this was to communicate the idea that the effectiveness of "Wavelet Shrinkage," and the best threshold $\lambda$ to use, will be dependent upon the nature of the true signal's highpass data (i.e., the probability density function of a signal's highpass wavelet coefficients).

In Section 4.5 we validated the original oracle's theoretical choice of the best $\lambda$ being the noise level $\sigma$. We followed up on this in Section 4.6 with an experimental investigation. In that section we took a simple sine wave, corrupted it with some noise, and then applied our wavelet techniques to denoise it. Figure 4.18 illustrated one experimental run, showing that our denoising techniques can reduce the MSE. Then in Figure 4.19 we extended this single result, after numerous experimental runs of varying noise and thresholding levels, to demonstrate where the best $\lambda$ actually existed. What we found out was that $\sigma$ rarely was the best $\lambda$ to use. We then moved on to investigate whether the calculations of expected MSE for a denoised sine function would suggest a best $\lambda$ to use. Those results were presented in Figure 4.21 and indicated that the best $\lambda$
was either 0 or $\infty$, depending on the noise level. Like the oracle's theoretical ideal, these expected calculations did not agree with our experimental results either. This lead to the hypothesis that, for the given conditions of the experiment, the deeper complexities of a "stochastic process" needs to be considered in determining a theoretical best $\lambda$.

For future research considerations, one promising direction to take this discussion in is towards the challenge of obtaining some better algorithms for determining our $\lambda$ thresholding values. For example, can we simply look at a noisy signal's data set, and use the probability density of the highpass' raw data in a deterministic way?

Beyond that, we acknowledge that this discussion only touches the surface of what researchers have come up with in the field of denoising and wavelets. There are many other techniques that incorporate wavelets besides the Haar wavelet transform and hard thresholding. In the papers by David Donoho (a central figure in this field), three comprehensive techniques are mentioned: RiskShrink [4, p. 440], VisuShrink [4, p. 444], and SUREShrink [3, pp. 1203-1208]. Further, it is clear that to fully appreciate these papers will require a substantive study of "decision theory," in particular, the decision theory topic of "minimax." What was also observed, in the referenced research papers for this discussion, was frequent mention of certain important function spaces (beyond the fundamental ones of Lebesgue and Hilbert). In Brani Vidakovic's text, "Statistical Modeling by Wavelets" [22, pp. 36-37], three function spaces are listed: Sobolev, Holder, and Besov. In one of David Donoho's papers [3] a fourth is also identified, Triebel. Finally, it is clear that appreciating the "stochastic process" will be quite important to advancing this discussion further. Most of our experiences in life occur one event at a time. Thus we understand that normal expectations are typically not the norm.
References


Bibliography


Appendices
Appendix A: Selected Mathematica Code

Figure 1.2. The Fourier series approximations of the sawtooth function \( (n = 1, 3, 5) \).

\[ g_n(t) = -\frac{2}{\pi} \sum_{k=1}^{n} \frac{(-1)^k}{k} \sin(k\pi t) \]

```
FourierSawtooth1 = 
    Plot[-(2/Pi)*Sum[Sin[k*Pi*t]*(-1)^k/k,{k,1,1}],
         {t,-12/Pi,12/Pi}]
FourierSawtooth3 = 
    Plot[-(2/Pi)*Sum[Sin[k*Pi*t]*(-1)^k/k,{k,1,3}],
         {t,-12/Pi,12/Pi}]
FourierSawtooth5 = 
    Plot[-(2/Pi)*Sum[Sin[k*Pi*t]*(-1)^k/k,{k,1,5}],
         {t,-12/Pi,12/Pi}]
```

Figure 2.1. The wavelet series approximations of the sawtooth function \( (n = -1, 0, 1) \).

\[
f_n(t) = \sum_{j=-1}^{\infty} \sum_{k=-\infty}^{\infty} 2^{-j} \begin{cases} 1 & t \in [2k, 2k + \frac{1}{2}) \\ -1 & t \in [2k + \frac{1}{2}, 2k + 1) \end{cases} + \sum_{j=0}^{n} \sum_{k=-\infty}^{\infty} 2^{-j-2} \begin{cases} 1 & t \in [2^{-j}k, 2^{-j}(k + \frac{1}{2})) \\ -1 & t \in [2^{-j}k + \frac{1}{2}, 2^{-j}(k + 1)) \end{cases}
\]

```
HaarSawtoothNeg1
    = Plot[.5*SquareWave[t/2],
            {t,-12/Pi,12/Pi}]
HaarSawtoothZero
    = Plot[.5*SquareWave[t/2]
            + Sum[-2^(-j-2)*SquareWave[(2^j)*t],{j,0,0}],
            {t,-12/Pi,12/Pi}]
```

110
Appendix A: (Continued)

\[
\text{HaarSawtoothOne} = \text{Plot}[0.5 \times \text{SquareWave}[t/2] \\
+ \text{Sum}[-2^{-(j-2)} \times \text{SquareWave}[2^j \times t], \{j, 0, 1\}], \\
\{t, -12/\pi, 12/\pi\}]
\]

Figure 2.4. Complete image before and after a Haar wavelet transformation.

<<DiscreteWavelets`DiscreteWavelets`

\[
\text{gray} = \text{ImageNames}[\text{ImageType} \rightarrow \text{GrayScale}, \text{ListThumbnails} \rightarrow \text{True}]; \\
\text{clown} = \text{ImageRead}[\text{gray}[[6]]]; \\
\text{clown1} = \text{Take}[\text{clown}, \{41, 200\}, \{1, 160\}]; \\
\text{ImagePlot}[\text{clown1}]
\]

\[
\{\text{clownrows}, \text{clowncols}\} = \text{Dimensions}[\text{clown1}];
\]

\[
b = \{\sqrt{2}/2, \sqrt{2}/2\}; \\
d = \text{Table}[0, \{\text{clownrows}-2\}]; \\
f = \text{Join}[b, d]; \\
h = \text{Table}[\text{RotateRight}[f, 2k], \{k, 0, \text{Length}[f]/2-1\}];
\]

\[
a = \{\sqrt{2}/2, -\sqrt{2}/2\}; \\
c = \text{Table}[0, \{\text{clownrows}-2\}]; \\
e = \text{Join}[a, c]; \\
g = \text{Table}[\text{RotateRight}[e, 2l], \{l, 0, \text{Length}[e]/2-1\}];
\]

\[
\text{haar} = \text{ArrayFlatten}[\{(h), (g)\}];
\]

\[
\text{clown1transformed} = \text{haar}.\text{clown1}.\text{Transpose}[\text{haar}];
\]
Appendix A: (Continued)

clown1blur = Take[clown1transformed, {1, 80}, {1, 80}];
adjustclown1blur = .5*clown1blur;

gray127quarter = Table[255, {i, 1, 80}, {j, 1, 80}];

clo
[formula]

Figure 3.2. Linear-log plots of some colored noise spectral power densities.

\[ L_{dB} = 10 \cdot \log_{10} \left( \frac{P_1}{P_0} \right), \]

pink = LogLinearPlot[10*Log[10, 10^2/(2x)], {x, 100, 22000}];
blue = LogLinearPlot[10*Log[10, 2x/10^6], {x, 100, 22000}];
white = LogLinearPlot[10*Log[10, 1/(2^5)], {x, 100, 22000}];
colornoise = Show[pink, blue, white]
Appendix A: (Continued)

Figure 4.2. Noise before and after a Haar wavelet transformation.

<<DiscreteWavelets`DiscreteWavelets`

gray127 = Table[127, {i, 1, 160}, {j, 1, 160}];
gray127quarter = Table[255, {i, 1, 80}, {j, 1, 80}];

nd = NormalDistribution[0, 1];
SeedRandom[];
noise = Table[Random[nd], {i, 1, 160}, {j, 1, 160}];

sigma = 32; (*enter a choice for noise level sigma*)
sigmanoise = sigma*noise;

noisygray127 = gray127 + sigmanoise;
ImagePlot[noisygray127]

noisygray127transformed = haar.noisygray127.Transpose[haar];

noisygray127blur = Take[noisygray127transformed, {1, 80}, {1, 80}];
adjustnoisygray127blur = .5*noisygray127blur;

noisygray127hd = Take[noisygray127transformed, {81, 160}, {1, 80}];
adjustnoisygray127hd = .5*(noisygray127hd + gray127quarter);

noisygray127vd = Take[noisygray127transformed, {1, 80}, {81, 160}];
adjustnoisygray127vd = .5*(noisygray127vd + gray127quarter);
Appendix A: (Continued)

```math
noisygray127dd
    = Take[noisygray127transformed, {81, 160}, {81, 160}];
adjustnoisygray127dd = .5*(noisygray127dd + gray127quarter);

noisygray127HWT
    = ArrayFlatten[{{adjustnoisygray127blur, adjustnoisygray127vd},
                     {adjustnoisygray127hd, adjustnoisygray127dd}}];
imagenoisygray127HWT = ImagePlot[noisygray127HWT]
```

<table>
<thead>
<tr>
<th>Table 4.3: MSE calculations for various individual signal strengths, $g_i$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>sigma=1; (<em>enter a choice for noise level sigma</em>)</td>
</tr>
<tr>
<td>lambda= sigma; (<em>set threshold lambda</em>)</td>
</tr>
<tr>
<td>g=2.5*sigma; (<em>enter a choice for &quot;g&quot; to evaluate</em>)</td>
</tr>
</tbody>
</table>

\[
P(a < T_i \leq b) = \tau_{(a,b)} = \frac{1}{\sigma \sqrt{2\pi}} \int_a^b \exp \left( -\frac{t_i^2}{2\sigma^2} \right) dt_i
\]

\[
1/(\text{sigma}\*\text{Sqrt}[2*\text{Pi}])\*\text{NIntegrate}[1/\text{Exp}[t^2/(2*\text{sigma}^2)],\{t,a,b\}]
\]
Appendix A: (Continued)

\[ \text{MSE}(\hat{g}_i)_{(-\infty,-3\sigma)\cup(-1.5\sigma,\infty)} = \frac{1}{\sigma \sqrt{2\pi}} \left( \int_{-\infty}^{-3\sigma} t_i^2 \exp\left( -\frac{t_i^2}{2\sigma^2} \right) dt_i + \int_{-1.5\sigma}^{\infty} t_i^2 \exp\left( -\frac{t_i^2}{2\sigma^2} \right) dt_i \right) + \frac{1}{\tau_{(-\infty,-3\sigma)} + \tau_{(-1.5\sigma,\infty)}} \]

\[
\frac{1}{\text{sigma}\cdot\text{Sqrt}[2\cdot\text{Pi}]}
\frac{\text{NIntegrate}[t^2/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{Infinity},-\text{g-lambda}\}
+ \text{NIntegrate}[t^2/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{g-lambda,Infinity}\})}
\frac{\text{NIntegrate}[1/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{Infinity},-\text{g-lambda}\}
+ \text{NIntegrate}[1/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{g-lambda,Infinity}\})}{\text{NIntegrate}[1/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{Infinity},-\text{g-lambda}\}
+ \text{NIntegrate}[1/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{g-lambda,Infinity}\})}
\]

Figure 4.5. Comparison of expected MSE(\(\hat{g}_i\)) to ideal MSE(\(\hat{g}_i\)), thresholding at \(\lambda = \sigma\).

\[
\text{expected MSE}(\hat{g}_i; g_i) = \int_{-\infty}^{-\sigma g_i} t_i^2 f(t_i) dt_i + \int_{-\sigma g_i}^{\sigma g_i} (g_i)^2 f(t_i) dt_i + \int_{\sigma g_i}^{\infty} t_i^2 f(t_i) dt_i
\]

sigma=1; (*enter a choice for noise level sigma*)
lambda=sigma; (*set threshold lambda*)

Plot[1/(sigma*Sqrt[2*Pi])
\[
\frac{\text{NIntegrate}[t^2/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{Infinity},-\text{lambda-g}\}
+ \text{NIntegrate}[g^2/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,-\text{lambda-g,lambda-g}\}
+ \text{NIntegrate}[t^2/\text{Exp}[t^2/(2\cdot\text{sigma}^2)],}
\{t,\text{lambda-g,Infinity}\})]
\{(g,0,5)\}]

115
Appendix A: (Continued)

Figure 4.8. Comparison of oracle_2’s ideal MSE($\hat{g}_i$) to oracle_1 and previous expectations.

\[ \text{ideal MSE}_{O_2}(\hat{g}_i) = \int_{-\infty}^{g_i} g_i^2 f(t_i) dt_i + \int_{g_i}^{\infty} t_i^2 f(t_i) dt_i + \int_{-\infty}^{g_i} t_i^2 f(t_i) dt_i. \]

\[ \text{MSE}(\hat{g}_k) = \int_{-\infty}^{-g_k-\lambda} t_k^2 f(t_k) dt_k + \int_{-g_k-\lambda}^{-g_k+\lambda} t_k^2 f(t_k) dt_k + \int_{-\infty}^{-g_k+\lambda} g_k^2 dt_k. \]

sigma=1; (*enter a choice for noise level sigma*)

Plot[1/(sigma*Sqrt[2*Pi])]
(NIntegrate[g^2/Exp[t^2/(2*sigma^2)],{t,-Infinity,-g}]
+ NIntegrate[t^2/Exp[t^2/(2*sigma^2)],{t,-g,g}]
+ NIntegrate[g^2/Exp[t^2/(2*sigma^2)],{t,g,Infinity}]),
{g,0,5})

Figure 4.11. Effects of an increasing $\lambda$ on different strengths of $g_i$'s.

sigma=1; (*enter a choice for noise level sigma*)
g = 0.55*sigma; (*enter a choice for "g" to plot*)

Plot[1/(sigma*Sqrt[2*Pi])]
(NIntegrate[t^2/Exp[t^2/(2*sigma^2)],{t,-Infinity,-g-lambda}]
+ NIntegrate[g^2/Exp[t^2/(2*sigma^2)],{t,-g-lambda,-g+lambda}]
+ NIntegrate[t^2/Exp[t^2/(2*sigma^2)],{t,-g+lambda,Infinity}]),
{lambda,0,5*g})

116
Appendix A: (Continued)

Figure 4.17. PDFs of various noisy sine functions.

\[ g(y) = [f \ast \varphi](y) = \frac{1}{\sigma \sqrt{2\pi}} \int_{0}^{\pi} \exp \left[ -\frac{(y - \cos u)^2}{2\sigma^2} \right] du. \]

sigma = 0.1; (*enter a choice for noise level sigma*)
minrange = -8; (*enter lower limit of evaluation*)
maxrange = 8; (*enter upper limit of evaluation*)

Table[1/(sigma*Pi*Sqrt[2*Pi])*
    Integrate[Exp[-(y-Cos[x])^2/(2*sigma^2)],[x,0,Pi]],
    {y,minrange+0.01/2,maxrange-0.01/2,0.01}]

Figure 4.18. The denoising of a corrupted sine function, before and after.

<<DiscreteWavelets`DiscreteWavelets`
cleansine = Table[Sin[2Pi*(2x-1)/(2*256)],[x,1,256]];
CleanSampleSine = ListLinePlot[cleansine];

nd=NormalDistribution[0,1];
SeedRandom[];
noise=Table[Random[nd],[k,1,256]];

sigma=0.1; (*enter choice of noise level sigma*)
sigmanoise = sigma*noise;
noisysine = cleansine+sigmanoise;
NoisySampleSine = ListPlot[noisysine];
noisysinetransformed = HWT1D1[noisysine];
noisysinetransformedlist
    = WaveletVectorToList[noisysinetransformed];
noisysinelowpass = First[noisysinetransformedlist];
noisysinehighpass = Flatten[Drop[noisysinetransformedlist,1]]; lambda = sigma;
denoisedsinehighpass
    = Map[ShrinkageFunction[#,lambda]&,noisysinehighpass];
denoisedtransformedsine
    = Join[noisysinelowpass,denoisedsinehighpass];
denoisedsine = IHWT1D1[denoisedtransformedsine]; DenoisedSampleSine = ListPlot[denoisedsine];
Show[CleanSampleSine,NoisySampleSine]
Show[CleanSampleSine,DenoisedSampleSine]

Figure 4.19. Finding the best $\lambda$ for denoising a corrupted sine function with noise $\sigma$. 

<<DiscreteWavelets`DiscreteWavelets`

cleansine = Table[Sin[2Pi*(2x-1)/(2*256)],{x,1,256}];
cleansinetransformed = HWT1D1[cleansine];

setdetail = .001; (*set increments for table*)
Appendix A: (Continued)

\[
\text{MSE} = \text{Table[}
\text{Table[}
\text{Norm[}
\text{cleansinetransformed} \\
\text{- Join[ First[WaveletVectorToList}
\text{[HWT1D1[cleansine+\sigma*\text{noise}]]],}
\text{Map[ShrinkageFunction[#,\lambda]&,}
\text{Flatten[Drop[WaveletVectorToList}
\text{[HWT1D1[cleansine+\sigma*\text{noise}]],1]]]]}
\]^{2/256},
\{\sigma,0,0.1,\text{setdetail}\},
\{\lambda,0,0.3,\text{setdetail}\};
\]
\)

\{\text{rows, cols}\} = \text{Dimensions[MSE]};

\text{MSEp} = \text{Partition[Flatten[Transpose[MSE]],rows]};

\text{BestLambda} = \text{Flatten[}
\text{Table[Min[Position[}
\text{Flatten[Drop[Drop[MSEp,k],k-(\text{cols}-1)]]],}
\text{Min[Drop[Drop[MSEp,k],k-(\text{cols}-1)]]}
\]^{-1},\{k,0,\text{cols}-1\};
\]
LambdaAnnihilates = Table[Min[Position[
    Flatten[Drop[Drop[MSEp, k], k-(cols-1)]],
    Max[Drop[
        Flatten[Drop[Drop[MSEp, k], k-(cols-1)]],
        Min[Position[
            Flatten[Drop[Drop[MSEp, k], k-(cols-1)]],
            Min[Drop[Drop[MSEp, k], k-(cols-1)]]]
    ]-1]
  ]-1], {k, 0, (cols-1)}];

LambdaEqualSigma = Table[y, {y, 0, (cols-1)}];

BestLambdaPlot =
  ListLinePlot[{BestLambda, LambdaAnnihilates, LambdaEqualSigma}]

Figure 4.20. The lowpass and highpass data of a clean sine function.

<<DiscreteWavelets`DiscreteWavelets`

cleansine = Table[Sin[2Pi*(2x-1)/(2*256)], {x, 1, 256}];

cleansinetransformed = HWT1D1[cleansine];
cleansinetransformedlist = WaveletVectorToList[cleansinetransformed];
cleansinelowpass = First[cleansinetransformedlist];
cleansinehighpass = Flatten[Drop[cleansinetransformedlist, 1]];

ListLinePlot[cleansinelowpass]
ListLinePlot[cleansinehighpass]
ListLinePlot[cleansinetransformed]
Appendix A: (Continued)

Figure 4.21. Plots of the MSE for various levels of noise $\sigma$, as $\lambda$ varies from 0 to $5\sigma$.

Plot[1/(sigma*Sqrt[2*Pi])
    * NIntegrate[1/(Pi*Sqrt[amplitude^2-g^2])
        * (NIntegrate[t^2/Exp[t^2/(2*sigma^2)],
            {t,-Infinity,-g-lambda]]
        + NIntegrate[g^2/Exp[t^2/(2*sigma^2)],
            {t,-g-lambda,-g+lambda]]
        + NIntegrate[t^2/Exp[t^2/(2*sigma^2)],
            {t,-g+lambda,Infinity}]])],
    {g,-amplitude,amplitude}],
    {lambda,0,5.2}]