

An Application of the Continuous Wavelet Transform to Financial Time Series

KLAS ELIASSON

MASTER'S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY



An Application of the Continuous Wavelet Transform to Financial Time Series

Klas Eliasson

Department of Electrical and Information Technology
Lund University

Advisor: Stefan Höst

Examiner: Maria Kihl

March 26, 2018

Printed in Sweden
E-huset, Lund, 2018

Abstract

Wavelet theory, which shares fundamental concepts with windowed Fourier analysis, introduces the notion of scale in an effort to aid in joint time-frequency analysis. Having century-old roots, much of the essential research on the subject of wavelets was conducted during the 1970s and 1980s. Despite being a rather young toolset, wavelets have shown to be very useful when studying signals with transient, non-stationary, characteristics.

This thesis focuses on the continuous wavelet transform (CWT) in the one-dimensional case from a practical implementation standpoint. It also contains sections on wavelet history, development, and the theoretical fundamentals.

The presented implementation contains a computer software with a graphical user interface that was developed in the context of financial trading in the currency markets. More specifically, the implementation contains a C++ based code library developed to expose an application programming interface (API) that is called from a retail desktop forex trading software where it can aid in market analysis visualization.

Popular Science Summary

A Look at Volatility in Financial Time Series Through Wavelet Analysis

Each year, billions of dollars are spent finding an edge in the financial markets. At the forefront of this pursuit is Wall Street's elite – the Quants – bringing physics, signal processing and engineering to the world of finance and economics. With this backdrop of high-stake applied research and evaluation of new tools and theories, we develop a trading software indicator using wavelet analysis and bring it into practice by exploring volatility phenomenon in the currency markets.

In the field of signal processing, it is well known that the legendary Fourier transform help us move between the often empirically observed time domain and the theoretically fascinating frequency domain. By studying the transformed signal, we can easily reveal the set of frequencies it is composed of. However, operating globally on the signal content, the Fourier transform does not lend itself to exploration of *when* certain frequency events occur in time. In other words, traditional Fourier analysis falls short when used in a non-stationary, transient settings and this is where wavelets come into play. By introducing the notion of *scale* as a proxy to frequency, the wavelet framework approaches the problem of telling when an exact frequency occurs in time; a dilemma with its roots in the Heisenberg uncertainty principle.

One setting where these types of transient signals – or time series – are prevalent, is the world of finance and economics. In this thesis work, we implemented the highly theoretical wavelet framework in a desktop trading application for the currency markets and used it to explore market movements (or *volatility* in a broad sense). Volatility analysis is an invaluable concept for many traders, institutional and retail alike. Front-running in the sense of bringing wavelet based volatility analysis straight into the retail currency trading world, the implementation resulted in a C++ library, which was consumed and visualized in the form of a market indicator. The indicator was based on the scalogram output of the continuous wavelet transform. Similar to a heatmap, it can easily identify parts of the time series containing abrupt changes or clusters of high volatility. While implementing the technical solution we stumbled upon several challenges in bridging theory and practice. Despite this, both the general software library and practical indicator is fully functional and ready to be shared with an enthusiastic trading community for further exploration and battle testing.

Table of Contents

1	Introduction	1
1.1	Problem and Motivation	2
1.2	Approach	3
1.3	Report Outline	3
1.4	Scope and Limitations	4
2	Historical Background of Wavelets	5
2.1	Sinusoids Make the World Go Round	5
2.2	An Inevitable Trade-off	9
2.3	Adapting to Change	12
2.4	Beneath the Forest and the Trees	12
2.5	The Next Steps	15
3	Wavelet Theory	19
3.1	The Continuous Wavelet Transform	19
3.2	Computation and Interpretations	20
3.3	Requirements of the Wavelet Function	21
3.4	The Inverse Continuous Wavelet Transform	23
3.5	Scale, Frequency and Heisenberg Revisited	24
4	Implementation	29
4.1	The Forex Market, Trading Software and Financial Time Series	29
4.2	Bridging Theory and Practice	33
4.3	Implementing the Continuous Wavelet Transform	35
4.4	Implementation Approach	36
4.5	The Software Library Architecture	37
4.6	Consuming the Library in MetaTrader	38
4.7	The CWT Indicator Result Interpretation	42
4.8	Limitations Caused by the Cone of Influence	47
5	Discussion and Reflections	51
5.1	Discussion	51
5.2	Further Research	52
5.3	Reflections	53

List of Figures

2.1	Fourier and Sum of Sines	8
2.2	STFT Time-Frequency Plane	11
2.3	Wavelet Time-Frequency Plane	13
3.1	Wavelet Dilation and Translation	20
3.2	The Wavelet Transform Principle	22
3.3	Wavelet Frequency Mapping	26
4.1	Implementation Architecture	40
4.2	MetaTrader User Interface	42
4.3	CWT analysis of AUD/NZD January, 2015	46
4.4	The Cone of Influence	49
4.5	CWT analysis of EUR/CHF during "Francogeddon"	50

Dedicated to my father...

Introduction

In the field of signal processing, it is well known that the legendary Fourier transform help us move between the often empirically observed time domain and the theoretically fascinating frequency domain. Simply put, by applying this mathematical operator to a signal and studying the transformed result, we can easily reveal the set of frequencies which the signal is composed of. For example, a Fourier analysis of a signal's frequency content can be used to determine the most dominating frequencies, enabling us to construct a filter to enhance or remove these. The application of such filter to the signal, followed by an inverse transformation back to the time domain again, would affect the whole signal – from beginning to end – because in Fourier space there is no notion of time, only frequency. Thus, it could be said that the Fourier transform can tell us *which* frequencies the signal includes, but not *when* they actually occur. This is a fundamental drawback as it could, for example, be of importance to spot *when* a sudden change in the signal's frequency content occurs, or *at which point* noise is added or changes its characteristics. Using the foregoing analysis method however, these transient events in the signal would present themselves throughout the whole transformed result, tainting it regardless of their actual temporal location. In other words, traditional Fourier analysis falls short when used in a non-stationary setting and this is where wavelets come into play.

Wavelet theory constructs a framework where a set of mathematical functions, called wavelets, can be used to examine a signal's characteristics in a wider perspective by trying to preserve the time domain representation when performing the analysis. Sharing fundamental concepts with *windowed* Fourier analysis, wavelet theory introduces the notion of *scale* in an effort to approach – and work around – the problem of telling when an exact frequency occurs in time; a dilemma with its roots in the Heisenberg uncertainty principle. Though having century-old roots, much of the essential research on the subject of wavelets was conducted during the 1970s and 1980s. It has since evolved and worked its way up from being a limited academic research topic, to become widely accepted and implemented throughout a number of different industries. Numerous papers are published each year introducing and adapting the theories to new application areas, some of which can be found in digital image compression, seismology, and financial time series analysis.

The thesis work performed here contains two distinct components; a review and discussion of wavelet theory, as well as a practical implementation based upon the former. This text documents both parts.

1.1 Problem and Motivation

As mentioned earlier, an important property of wavelet analysis is its ability to operate on non-stationary signals and analyzing transient events. One context where these types of signals – or time series – are prevalent is the world of finance and economics. In financial trading in particular, price data of a financial instrument over time often exhibits these characteristics. As the price of for example a stock varies over time it constructs a time series (e.g. daily price observations). This price movement is sometimes modeled as a geometric Brownian motion with drift, and this seemingly random walk has been subject to much scrutiny, research, and even prediction by different market participants. Often credited as the starting point, Bachelier's work in the early 1900s has been continued and there is more research done on this general topic than we can even start to name here [1]. However one often common theme by the modern applied researchers, is to try to find an *edge* in ones analysis. This edge (or advantage) could help the analyst or trader to successfully execute a profitable investment or trade in the instrument. Billions of dollars are spent by large institutions such as hedge funds to find a competitive edge, but even in the so called retail segment composed of keen individual traders, more or less scientific research is conducted to gain some kind of advantage over others.

In the world of trading, new technologies, approaches and concepts are constantly explored and the latest academic research is put into practice by enthusiastic and profit chasing communities. With the background from our introduction to wavelet analysis, it would be very interesting to see what – if anything – this relatively new area could contribute to this pursuit. In this thesis work, we are exploring this question by implementing the theoretical framework in a retail desktop trading application for the currency markets. We use the created software to explore market movements (or *volatility* in a broad sense).

Volatility as a concept can be looked at and defined in several different ways, ranging from simple absolute values of the observed price returns, or as the statistical standard deviation of the time series. In our context, we explore it from the perspective of being present in the output of the continuous wavelet transform. We will for example investigate how our software reacts when analyzing certain price patterns such as clusters of high volatility and very abrupt changes in the data. Volatility analysis in different shapes and forms is an invaluable tool for many traders and investors, both institutional and individual. Some strategies work only in times of high volatility, yet others require very calm market conditions to be profitable. It can serve as input to risk models, or even be the basis of its own traded derivatives. In some contexts, volatility metrics are used as the "fear"-levels in the markets and some strictly equates it with statistical properties of a data set used in complex models.

Though many different software packages exist that either provide different volatility or wavelet analysis features, at the time of commencing this thesis work, no publicly available software package was found that brought the two concepts together in an effort to aid retail currency traders with a visualization of a wavelet-based joint time-frequency analysis directly from within their trading application. The main work of this thesis work became to explore and to a large extent solve

the problem of how it could be accomplished in a practical and concrete way. Formulated as "How can a system be implemented to use wavelet theory as an aid in the practical analysis of a financial time series with focus on volatility related phenomena?", this problem will occupy the reader for remainder of this text as it has done the author for much too long.

1.2 Approach

There are significant efforts made in bringing methods from the field of digital signal processing (DSP) into the world of finance. Some years ago, this contributed to the creation of the *quants*; Wall Street professionals that come from a physics and engineering background rather than finance and economics, developing models and strategies for the markets. As an engineering student, viewing financial data such as a price series as a signal or output from an information system is intriguing and exploring it in the frequency domain lies close to heart. With this perspective, volatility could loosely be seen as, for example, transient bursts of high frequency content or other very short-lived or local changes in the time domain.

The tool of choice for anything frequency related is often Fourier analysis. As briefly mentioned in the introduction, it has the drawback of poor temporal localization of transient events as it operates globally on the signal content. Different windowing techniques approach this precise problem, and as an extension to these, wavelet analysis suggests promising results in the trade-off between time and frequency resolution. The decision to use wavelets in this thesis work can thus be traced back to both a curiosity of the author in exploring beyond the techniques and concepts he previously has been exposed to, fueled by an interest in the edge seeking and competitive trading industry, as well as an implied theoretically well-aligned approach.

As for the technical context, though numerous trading software platforms exist, few were as available and open as the one chosen for this work, MetaTrader. This software offers the ability to write complex plugins in an advanced programming language using bespoke linked libraries. Combined with its simple user interface, it was the strongest candidate. The wavelet library itself was implemented using C++ as it has a good track-record in the DSP field and lends itself for advanced and fast manipulation of low-level data, yet using high-level syntax. The author has previous experience in currency trading and model development so to apply the work in these markets was a natural choice. It could, however, certainly be used in any other context such as stocks, bonds or commodities as well.

In the following paragraphs, we provide an overview of the subsequent chapters along with some of the more specific questions that each covers.

1.3 Report Outline

This report has the following outline. To provide the reader with a background and context, we start off by concerning ourselves with questions like; *What drove the development of wavelet theory?*, *Why do we need it?*, and *Why can't we have the best of two worlds – time and frequency – at the same time?* In this first

part of this thesis, the quest for "joint time and frequency localization" takes the reader on a journey starting in the early years of the 1800s with the research of Joseph Fourier, through times of great innovation and discoveries, touching down on Gabor's work of 1946, and finally arriving at the last decades of a century dominated by the digital revolution.

Following the historical background and overview, the continuous wavelet framework is presented and scrutinized in detail. This section serves as a presentation of the theory used in the implementation phase. We provide distinct definitions, formalism and answers to questions such as; *What defines a wavelet function?* and *How are wavelet scale and frequency related?*.

The final part concerns itself with our practical implementation and use of wavelet theory in a "real-world" setting. We introduce a chapter that explores questions like *How do we bridge theory and practice?*, *How can wavelets aid in the analysis of data?*, and *What challenges do we face in the implementations of the algorithms?* Here, we stumble upon an array of practical constraints and issues we have to adhere to and work around. We also briefly introduce the reader to the world of currency trading, which is the backdrop of our practical application. More specifically, the developed and presented software is part of a C++ based code library consumed from within a desktop trading platform where it could aid in market analysis and possibly even trading decision making. The implementation utilizes the continuous wavelet transform framework presented in previous chapters.

1.4 Scope and Limitations

In short, this thesis first and foremost aspires to serve as a background to, and presentation of, the practical implementation. Secondly, it serves as an introduction to the continuous wavelet framework and as an illustration of some of its inner workings in the one-dimensional case. The target audience considered in this paper is anyone interested in the subject, but to fully grasp and enjoy the content one should be familiar with basic college-level mathematics and have, at least some, knowledge of signal processing and the main concepts of that field.

With that said, the presentation will be aimed at providing a conceptual introduction and overview rather than being a mathematical thesis. Even though it lies within the subject-matter's nature to require some theoretical sections with rather dense mathematical theorems and definitions, the text is written for the engineer or practitioner seeking to get acquainted with topic, rather than one looking to explore mathematical proofs. I have purposely kept the theorems and formulae simple for better readability and apologize for any left-out definitions, limits, and more. It should also be mentioned that in quotations and citations using – or referencing – definitions or mathematical language, I have sometimes adapted the original author's notation (such as symbol representations or indexing schemes) to adhere to the convention used herein instead, for the purpose of consistency.

For a meticulously complete mathematical coverage of the topic we suggest [2–5], all of them being excellent sources written by admirable veterans of the field.

Historical Background of Wavelets

“Wavelet theory finds its origin in the recurrent need to develop a localized version of Fourier analysis, inasmuch as is possible within the Heisenberg principle constraint.”

Ingrid Daubechies

As mathematicians present the answer to a problem, they are often judged not only by its correctness, but also conciseness. This chapter’s opening quote is indeed both correct and concise enough, just as expected by one of the most recognized mathematicians and researcher on the subject of wavelets. In fact, the statement distinctly summarizes the next twenty or so pages. However, for the sake of context and broader perspectives, this thesis will stray into the field of modern wavelet theory first after seeking to account for where it all once began. The seemingly brief passage quickly turns out to be a 150-year long expedition with many stops on the way. It will also show, that there are as many ways to reach the different destinations, as there are pathfinders and navigational instruments. Let the journey begin...

2.1 Sinusoids Make the World Go Round

Making himself a name during the French Revolution and later serving side-by-side with Napoleon in his expedition to Egypt, Jean Baptiste Joseph Fourier (1768 - 1830) is nowadays more known as the French mathematician and physicist born in 1768 whose pioneering work on the propagation of heat resulted in one of the most employed mathematical analysis toolsets of modern times [6]. His work has later been generalized and made more abstract over time, but also adapted and refined to serve as a cornerstone in a vast set of disciplines, many of which Fourier himself never could have imagined. Just as an example, in the field of signal processing, when thinking of frequency analysis we instantly think of Fourier and the famous transform carrying his name.

In short, Fourier argued that any arbitrary periodic function could be decomposed into a series of simpler trigonometric functions. He initially needed this result to simplify the calculations in his efforts of solving the heat equation for which the solution was known in these particular simple cases, but not otherwise.

By applying a divide and conquer strategy, he could now solve the complex equations by dividing up the calculations and in 1807 Fourier was confident that his work was ready for the world. "An arbitrary function, continuous or with discontinuities, defined in a finite interval by an arbitrarily capricious graph can always be expressed as a sum of sinusoids" [7] he supposedly concluded while presenting his work entitled "On the Propagation of Heat in Solid Bodies" [8] to the Academy of Sciences in Paris. A review committee had been appointed that included the renowned mathematicians Lagrange and Laplace, among others. However, the paper immediately caused controversy and the criticism did not wait. The aforementioned mathematicians were some of the most noticeable objectors [9, 10].

A few years later, the same academy announced the "Grand Prix de Mathématiques de l'Institut" of 1812 and this year the prize would be awarded on the subject-matter of mathematical theory of heat. Fourier took the opportunity, revised and amended his work and submitted it. Apparently, only one other paper was entered into the contest and Fourier was finally set up for recognition. As customary, a committee was established to award the prize, but inconveniently for Fourier, it also included the two unimpressed objectors to his earlier work. Despite this disadvantage, Fourier was indeed awarded the prize but the triumph was short-lived. In the report, forefront mathematicians all over the world could read that "the manner in which the author arrives at these equations is not exempt of difficulties and that his analysis to integrate them still leaves something to be desired on the score of generality and even rigour" [6]. His theories had yet again been criticized, and being unable to impress the mathematical society in Paris, there was no convincing interest in publishing his work. In a few pages, we will see that Joseph Fourier is not the only scientist we will meet on this journey having formulated and presented their – nowadays considered keystone – theories just to be criticized by a judgmental establishment.

It is however – to both Fourier's and the establishment's defense – easy to recognize that trying to describe any arbitrary periodic function using only a subset of other select functions, Fourier had a very daunting and task at hand. The elementary functions he had chosen were sine and cosine, two fundamental trigonometric functions with their roots in ancient Indian astronomy which have since been the subjects of study by every noteworthy mathematician and present-day high school student alike. Using these functions, he formed a basis with which he hoped to represent, or at least approximate, any other periodic function of his interest. Most readers will recall the term *basis* from linear algebra, where it is defined as a set of linearly independent vectors in a vector space, that can represent every other vector in that same space [11]. A less strict, but perhaps more intuitive approach, would be to think of this basis as a coordinate system, e.g. the common Euclidian basis (x, y, z) , which spans the space and with which we can describe any point therein. In general mathematics however, we generalize our vectors into arbitrary *functions*. The analogous case is then, that a set of *basis functions* are said to make up the basis for a given *function space*, thus allowing every and any continuous function in that space to be described as a linear combination of these select functions.

Equipped with the sine/cosine basis, Fourier succeeded and like a prism breaks up light into different colors, he managed to decompose the functions into a (some-

times yet infinite) sum of the chosen basis constituents. In an IEEE journal article, the author Amara Graps helps us illustrating the principle by comparing $f(x)$ with a musical tone, for example the note A in a particular octave. As we know, one could construct A by means of adding combinations of sines and cosines with different amplitudes and frequencies together. In this case, the sines and cosines would be the *basis functions*, the elements of Fourier synthesis [12]. In mathematical terms, [12] accounts for the aforesaid by describing that Fourier asserted that any 2π -periodic function $f(x)$ is the sum

$$a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) \quad (2.1)$$

of its Fourier series, where the coefficients a_0 , a_k and b_k are calculated through

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx, \quad a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx \quad (2.2)$$

This played an essential role in the development of the mathematical understanding and view of functions. It could even be said to have opened up the door to a new functional universe [12].

If the Fourier *series* shows us how to decompose any *periodic* function into a sum of sinusoids, the famous Fourier *transform* could be seen as the process and extension of this idea to *non-periodic* functions. The Fourier transform, which in fact is closely related to an integral transform used by Laplace (now carrying *his* name), turns a function, f , originally in the time (or spatial) domain into a frequency dependent function \hat{f} . In mathematical notation this could be expressed as

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt \quad (2.3)$$

where ω represents the (angular) frequency and t represents time [3]. Though widely used in many applications throughout all fields of science and different areas of pure mathematics, in *signal processing* one often use the transformed result to explore how the energy is distributed among the different frequencies of a given signal. We therefor often refer to the transform of the signal as its *spectrum* and in particular analyze $|\hat{f}|$, called the *magnitude* (or $|\hat{f}|^2$ the *power*) spectrum. These spectra are often plotted with the frequencies represented on the horizontal axis and the magnitude (or power) varying vertically, making it easy to grasp the signal's characteristics in frequency space. As a side note, it should however not be forgotten that the transformed function \hat{f} is a complex function and that the signal's *phase* spectrum also is of interest depending on the application.

From the above it is clear that the Fourier transform can tell us *what* frequencies (and how much of them) the signal includes in a certain time interval but not *when* they actually occur. This arises from the mathematical definition of the transform since it is working with sinusoidal wave functions, $e^{i\omega t}$, which are infinitely long. The end result $\hat{f}(\omega)$ will therefore depend on all times, t , of the full duration of the signal function $f(t)$. Hence, [3] states that this global blend of information makes it very hard to analyze any local property of f from \hat{f} .

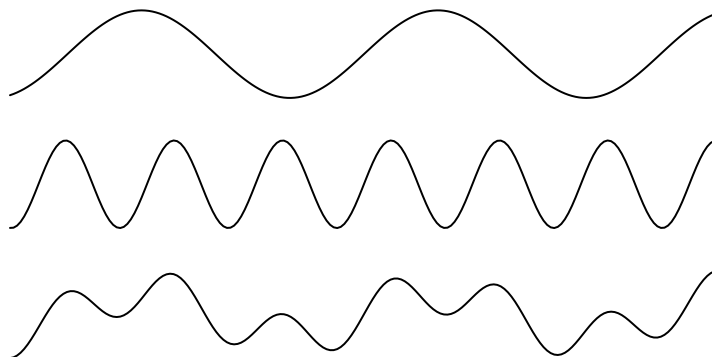


Figure 2.1: An illustration of Fourier's principle of adding simple sines (the top two) to compose a much more complex function (bottom).

Over-simplified one can say that, in the process of transforming the signal data, we have lost representation of the time-domain. In reality, we have not *lost* any information, we are just hiding it. Both the original function and the transformed one, contain all the information about the signal; it is just a matter of perspective. By using the so called *inverse* Fourier transform we can make the now frequency dependent function time dependent again and recreate the original signal through [3]:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega t} d\omega \quad (2.4)$$

It is easy to realize that the applications of Fourier analysis are next to limitless. The concept of moving from time to frequency representation – and back – is used almost everywhere in our daily lives; from the basic construction of the human ear, to signal transmission in cellular networks and in the processing and manufacturing of more or less all things around us. It is in other words a very fundamental concept, which has been studied by thousands of researchers, engineers, and scientists, along with millions of students, since we gave the phenomena its current name. Despite all these studies, dissertations, and a couple of centuries passing by, there is one fundamental property – or *limitation* – of time-frequency analysis that mocked them all since the get-go; it seemed nearly impossible to expose when an exact frequency occurred in time, and vice versa. As we will learn in subsequent sections, the dilemma is a Gordian knot tied by the Heisenberg uncertainty principle.

In many areas, just a *rough idea* of when certain frequency content exist would suffice, hence the total absence of time awareness results in a number of drawbacks of the Fourier transform's application. As an example, it can make an analysis especially error-prone when used with long-duration signals. As an illustration of this, consider the injection of a short burst of high frequency content at the very end of a long – otherwise "calm" – signal. This injection will now skew the whole

result of the analysis as it – in this particular case – after being transformed with the rest of the signal, would show up on the spectrum plots as if it dominated the *whole* data sample from start to finish.

Over time, the transform's application areas widened far beyond Fourier's heat equations and the mathematical undertakings of the nineteenth century. The demand for more sophisticated analysis increased and it was now crucial to find a way to cope with the enigma, and explore new methods based on Fourier's work that at least could reveal hints as to when a set of frequencies actually appear in the signal.

2.2 An Inevitable Trade-off

One scientist working on the aforementioned perplexity was the Hungarian-born British physicist Dennis Gabor who tried to tackle the dilemma of preserving time-domain representation by introducing what was later called the short-time Fourier transform (STFT), the windowed Fourier transform, or simply – the Gabor transform (the latter actually being a special case of the former) [13, 14]. In his paper entitled "Theory of Communication", which was published in 1946, he initiated the reader by emphasizing that our everyday life and human auditory sensations insist on a description in terms of *both* time and frequency [15]. By the tone of his paper, Gabor was evidently determined to shed new light on the Fourier transform and the established idealizations (as he called them) and provide means of describing his introductory point of view in quantitative language [15].

This is where we can return to the Gordian knot and the Heisenberg uncertainty principle – commonly known from Quantum mechanics – which basically states that one cannot know both the precise position *and* momentum of an elementary particle [16]. The uncertainty principle applied to time-frequency analysis, however, states that we cannot know the precise time at which a certain frequency occurs, and vice versa. As a matter of fact, speaking of *knowing* is misleading (hinting that we just have not found a way to represent the both domains simultaneously) when the truth is that a signal cannot simultaneously *have* a precise location in time and frequency [17]. This can be exemplified by the following exercise. Imagine the playing of a perfect tone, represented by a single sine wave, recorded over a period of time. Now, trim the ends of the recording over and over again, until all you can hear is a short "click". Like a Sorites paradox¹, at some point of the trimming process the recording stopped representing the tone, taking us from a well-defined single frequency during a time span, to a wide spectrum of frequencies at an exact point in time – the click [19]. In other words, there seems to be a trade-off between time and frequency whereas a more precise localization in one of the domains, gives a poorer precision in the other. As food for thought, the author of [17] rhetorically raises the question of how one can speak about

¹Also known as "little-by-little" arguments. Sorites comes from the Greek word "soros", meaning pile or heap, which is also the name of the puzzle it originally refers to; "Would you describe a single grain of wheat as a heap? No. Would you describe two grains of wheat as a heap? No... You must admit the presence of a heap sooner or later, so where do you draw the line?" [18]

frequencies at a precise instant, when frequencies need to have time to oscillate.

Gabor was purportedly a pioneer in this regard as he would insist that the uncertainty principle was relevant to communication theory, which he gave expression to by formulating a framework describing the *uncertainty relation* which acted in-between the two extremes of either time or spectral analysis [15, 17]. One might say that by doing so, Gabor now used the uncertainty principle he had been trying to battle, as a nifty tool in his new way of representing the two domains. He claimed that signals are represented in two dimensions and by using time and frequency as co-ordinates, such two-dimensional representations can be called *information diagrams*. Further, Gabor concluded that "the frequency of a signal which is not of infinite duration can be defined only with a certain inaccuracy, which is inversely proportional to the duration, and vice versa" [15].

Gabor's work on the uncertainty principle and how it relates to time-frequency analysis is today also known as the Gabor limit, which basically concludes that what we can know, are simply the time *intervals* in which a *band* of frequencies exist, all tied together with a trade-off approach in terms of precision. Seeming vague, this is in fact a very important realization and turning point, transforming the time-frequency enigma into a concrete *resolution* problem.

The approach Gabor chose when constructing his new representation and to manage the trade-off was to extend the Fourier transform using a *sliding window* while performing the analysis. Using a carefully chosen window function of fixed width, and by only analyzing the segment visible through the window before shifting it along and doing it all over again, until the signal was fully covered, he managed to create a representation that included information about both time and frequency. Like tiles placed on a floor one-by-one, Gabor's transform created a rectangular partitioning of an idealized time-frequency plane where each tile (better known as a Heisenberg box) had the same shape, governed by the window function used in the analysis. Even though the resolution can be changed using different window functions, it is always fixed for a given analysis set. Gabor could now retrieve an approximate time span in which a certain frequency event occurred.

Mathematically, the short-time Fourier transform can be expressed as

$$S(u, \omega) = \int_{-\infty}^{+\infty} f(t)g(t - u)e^{-i\omega t} dt \quad (2.5)$$

which indeed is a function of both time (u determines the location) and frequency [2, 3]. The simplest window function, g , is a rectangular one, constant during a certain (short) interval and zero outside. The window is multiplied with the original signal, leaving it with zero amplitude except at the window's position. As a side-effect, in the case of a rectangular function, the truncation will leave sharp "corners" at the beginning and end of the window, introducing unwanted frequencies which distort the final analysis result. To cope with this issue – and to adapt to the requirements of different applications – a wide range of window functions have been developed over the years. Some of the more famous are the *Hann* and *Hamming* functions. Gabor himself used a Gaussian window in his transform to "smooth" out the edges, actually making the Gabor transform a special case of

the more general STFT which – like we have seen – in itself does not direct what particular window to use [20].

Given a certain *shape* of a window, its *size* must also be carefully chosen since it will determine whether there will be a trade-off in time or frequency resolution. Now, recall the metaphor above about partitioning the time-frequency plane (where time varies horizontally and frequency vertically) like laying tiles on a floor. The size of each tile represents the precision, or resolution, and it cannot be arbitrarily small. In fact, the so-called time-bandwidth product, which – still metaphorically speaking – has the two sides of the tile as factors, dictates a theoretical lower bound of its area. The product can be proven to be distinctly defined by the Heisenberg principle, and this topic will be revisited and reviewed in-depth in the next chapter.

So, now given a set specification of how large of an area each tile must cover, there is also in practice a rather stringent constraint on their *dimensions*. Taking a narrow-and-tall approach leaves us with a fine partitioning and great precision along the time axis but due to the extensive "height" of the tile it will span a wide range of frequencies on the other axis. Conversely, choosing a wide-but-short measurement ratio will result in that pinpoint accuracy in the frequency domain, but then comes with the trade-off of widening the time interval instead. Once the desired trade-off approach is determined, the tool we use to finally cut the tiles to our preferred and chosen dimension is – as hinted above – the window size used in the transform. One can easily realize that a small window size corresponds to the "narrow-and-tall" decomposition approach with good temporal (however vague frequency) resolution. On the other hand, an ever-widening window – ultimately covering the full signal length – would result in an outcome with lost time representation, as is the case with the non-windowed Fourier transform.

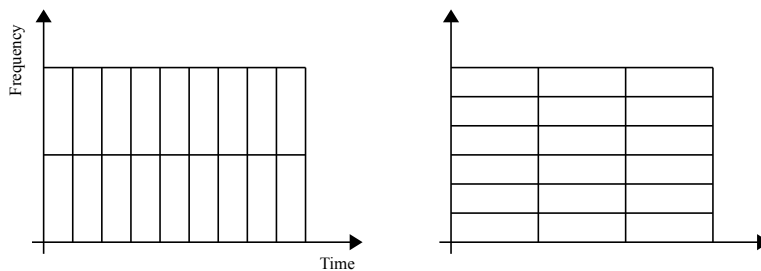


Figure 2.2: The illustrative effect of partitioning the time-frequency plane using the two different approaches narrow-and-tall (left) and wide-but-short (right).

With the set size of the window being such an imperative (and restricting) part of the analysis, it certainly raises questions on finding the *right* value. In practice, this naturally depends heavily on the application and desired properties of the product, but theoretically, what would happen if we let the size – or *scale* – of the window become a variable part of the equation? Seeking an answer to that

question will eventually take us to the final destination of our journey – multi-scale or *multi-resolution analysis* – the core of modern wavelet analysis.

2.2.1 A Quick Side Note on The Fast Fourier Transform

As a side note while on the topic of major contributions to Fourier analysis, the discovery of the fast Fourier transform (FFT) algorithm must be mentioned. It is a fast and efficient implementation of the discrete Fourier transform (DFT) and was published in the paper "An algorithm for the machine calculation of complex Fourier Series" by the American mathematicians James Cooley and John Tukey in 1965 [21]. It was however later discovered that the method actually was used more than a century earlier, but the timing of the publishing in 1965 – just at the beginning of the computer revolution – made the FFT practically useful and very popular. Today, it is this version that heavily dominates the practical use and implementations of the Fourier transform, many of which would not even have been possible without it.

2.3 Adapting to Change

Despite new findings and the introduction of the windowed approach, there were still problems the Fourier transforms could not solve. The fixed partitioning of the time-frequency plane generated by the STFT just did not give the level of detail needed to properly study all aspects of the signal. The trade-off between good resolution in *either* time or frequency simply made this tool blunt. In order to overcome this preset resolution problem, one had to dynamically change and adapt the window function whilst performing the analysis and this was exactly what became the next step. A variable-length sliding window was used and by keeping the window narrow when analyzing high frequencies for a better time resolution, and wider for lower frequencies (resulting in a better frequency resolution), the time-frequency plane partitioning now looked seemingly adaptive in nature. This is illustrated in Figure 2.3. Even though at each level (or row of tiles, to continue the analogy from the last section), the uncertainty trade-off still exists and the Heisenberg principle still holds true, this method provided the perfect answer to a commonly observed real-world signal processing dilemma; providing good temporal preciseness about high frequency content yet preserving the specifics of the lower frequencies making up the overall signal. This seemingly apparent work-around is, as we shall see later, actually the closest to a solution to the enigma we will come on our journey.

2.4 Beneath the Forest and the Trees

Due to its adaptive nature, a common analogy in the literature is that; with wavelets we can see both the forest *and* the trees. Interestingly enough, we find yet another dendrological reference in the literature on wavelet history; it is Ingrid Daubechies, one of the most prominent scientists in the field, who concludes in her paper called "Where do wavelets come from? – A personal point of view"

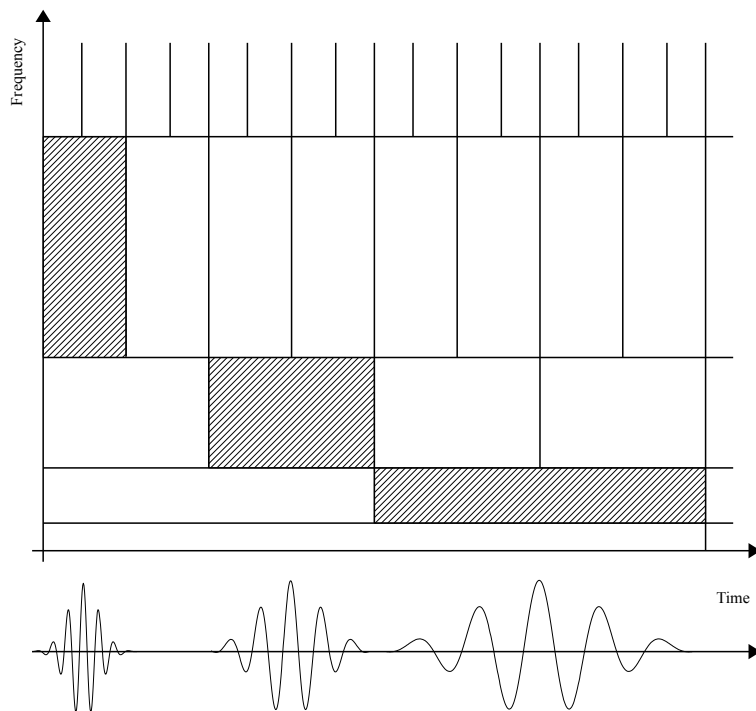


Figure 2.3: An illustration of the adaptive effect of partitioning the time-frequency plane using a wavelet transform with windows (wavelets) of variable length (scale) in contrast to the fixed partitioning in Figure 2.2.

[22] that the development of wavelet theory is connected and related to a vast number of different disciplines such as mathematics, physics, computer science, and engineering. She compares the history of wavelets with a tree whose roots reach deeply and in many directions and where its trunk represents the accelerated development of the wavelet tools in the second half of the 1980's. The crown of the tree has since branched out in many different directions, one for each field where wavelets are being commonly applied today.

Her point being that, it is hard to trace back a single line of events leading up to the state of the art today, and as we shall see later, that many findings actually are re-discoveries of previous work from other disciplines. Consequently, the story of the wavelet transform's origin will certainly be different depending on the storyteller. Nevertheless, we will make our next stop in the late 1970s and a certain Mr. Jean Morlet, a destination on a route coherent with Daubechies' own chronicle.

Jean Morlet, a pioneering geophysicist working at the French oil company Elf Aquitaine during a decade tainted by the "black gold" and one energy crisis after another, struggled to find new and better ways of detecting and analyzing oil fields for his employer. Traditionally, oil prospecting involved sending acoustic waves down through the ground and then analyzing the spectrum of the echoes, or backscatter, using Fourier analysis [23]. The returning signals contain an overwhelming amount of different frequencies and to separate and measure the geological layers and oil reservoirs, Morlet started using a windowed approach to divide up the work. At the time, computer systems were breaking more and more ground and – as computational power got more accessible – Morlet could afford to place these windows closer and closer together, finally even overlapping each other. In 1975, he was now in fact working with Gabor's ideas from 30 years back, however, instead of keeping the sliding window fixed in size and letting a wave oscillate inside it, he did the opposite and kept the oscillations constant and let the window width vary. He named his initiative "wavelets of constant shape" and it should soon prove to be a successful one. The signals he was interested in analyzing had the characteristics of containing very high frequency content during short time spans, and low frequency components with long durations. Looking back at the adaptive partitioning of the time-frequency plane proposed above, this new transform seemed to perform just the way he needed it to.

Motivated by the extraordinary results, he shared his findings with the world but was supposedly met with much skepticism and criticism about the lack of mathematical rigor. According to Daubechies, Morlet himself once paraphrased the attitude that met him and his results as "If it were true, then it would be in the math books. Since it isn't in there, it is probably worthless" [22]. Little did he know that he was now one of the main characters in a story that shared a remarkable resemblance to the one of Joseph Fourier himself – and his work – some 150 years earlier where innovative progress in a field was met by discouraging skepticism among the more established scientists. History does have a tendency to repeat itself, and this is only the first of many examples of that on this journey.

Determined to face the criticism, Morlet turned to a friend and former classmate to help him review his paper. The friend ended up referring him to Alex Grossmann, a Croatian-French theoretical physicist, active in the field of quantum

mechanics. Grossmann saw similarities in Morlet's transform to the methods he had previously used successfully in his own field and the work began with formalizing the framework mathematically. The two spent a year or so exploring many applications, frequently using their personal computers, at the time a rather new phenomenon which Grossmann attributed much of their initial success to [17]. Their endeavors were challenging albeit successful, and by 1984, Grossmann had not only managed to devise an inverse transform, they also had proven that one could perform the reconstruction with a single integral (as opposed to a double integral stemming from the two dimensional output of the transform) [17, 24]. This was a crucial step for practical reasons and real-world applications.

As mention earlier, history tends to repeat itself, and it would indeed later surface that Morlet's transform – perhaps *the* largest contribution to wavelet theory – was actually a rediscovery of the works of Alberto Calderón in the 1960's on the topic of harmonic analysis. In 1997, Jean Morlet received the Reginald Fessenden Award for his groundbreaking work.

2.5 The Next Steps

As previously revealed, Yves Meyer, a mathematician specializing in harmonic analysis, heard about Morlet and Grossmann's work (supposedly while waiting in line to a copy machine) and after getting acquainted with it in 1984, realized the similarity with theories in his own field [22]. He showed that the wavelets and their transform were related to the powerful Calderón-Zygmund theory [14]. Meyer, now intrigued by the new interpretation of this theory, contacted the two scientists and this would be the start of an era with interaction and knowledge sharing between many different disciplines, theoretical and applied researches alike. By connecting the dots, Meyer had expanded the universe of wavelet theory and brought it into a generalization framework of multiple dimensions [25].

After working closer with the results of Morlet and Grossmann, Meyer turned his attention to the level of *redundancy* in the transform. In the continuous case, the transform was – in theory – calculating an infinite number of wavelet coefficients turning a one-dimensional signal into a two-dimensional image, storing colossal amounts of information about the signal. Though this level of inefficiency could be useful in certain applications, a perfectly sparse and concise representation would pave the way to a whole new wavelet paradigm; orthonormal wavelet bases.

In the summer of 1985 Meyer created the first orthogonal wavelet basis the world had seen – or at least that is what he thought at the time. "These discoveries sprung out as a revolution", Meyer says and quickly adds, "But I soon found out that orthonormal wavelet bases already existed" [25]. Ironically, being the one who a few years earlier himself pointed out that the "invention" of Morlet et al. was a mere rediscovery of previous theories, Meyer now had the same achievement on his resume. A few years prior to Meyer, another harmonic analyst – this time a Swede by the name Jan-Olov Strömberg – had started working on the same wavelets and had indeed also presented an orthogonal basis with the same structure as Meyer's [25]. It should be noted however, that neither of the two actually were the first to

discover wavelets with this precious property. Technically, that accomplishment and honor dates back to the beginning of the century and the mathematician Alfréd Haar.

Haar was born 1885 in Budapest, Hungary, where he also started what would become a successful and lifelong academic career [26]. Even though his main interest was chemistry, the fascination for pure mathematics finally brought him to the University of Göttingen in Germany where he also received his doctorate with a dissertation entitled "Zur Theorie der Orthogonalen Funktionensysteme" (On the Theory of Orthogonal Function Systems) in 1909 [27]. It is in the very last section of this paper that a function that we today would call a wavelet, is mentioned in writing for the first time [12]. The function, now known as the Haar basis function or the *Haar wavelet*, is the simplest possible wavelet and is depicted in Figure 3.3 (bottom left).

The next major leap forward in the story of wavelet analysis also sprung out of a twist of fate, when one of Yves Meyer's graduate students was on vacation in the south of France and met an old friend, Stéphane Mallat, also taking a well-deserved break from his graduate work at the University of Pennsylvania. Upon returning to the United States, Mallat contacted Meyer, and the two met up at the University of Chicago, actually sharing Antoni Zygmund's office [17]. Together, Meyer and Mallat started discussing a theoretical framework which later became *multiresolution analysis (MRA)*. This framework not only aimed to explain wavelets and their inner workings, but also to serve as a recipe allowing the construction of new orthonormal wavelet bases in a very easy manner. Further, the multiresolution analysis framework led to a simple and recursive filtering algorithm that efficiently could compute the wavelet decomposition of a function or signal. This was how wavelet analysis really made it big in the field of signal processing. In 1988, Mallat earned his Ph.D. from the University of Pennsylvania through his hard work.

The next major contribution to wavelet theory came from a person hereto working in parallel with the other researchers, and with a somewhat lower profile. She was well connected to some of the most prominent names already mentioned and her name is Ingrid Daubechies, a Belgian physicist and mathematician, who had studied and earned her Ph.D. under Alex Grossmann in 1980 [28]. Daubechies had heard about Meyer's and Mallat's work very early on, and had gotten access to their unpublished findings. She took a special interest in the properties of the different wavelets themselves as well as the actual methods used to design them. Daubechies' work resulted in a whole new family of wavelets – the Daubechies wavelets – and among them one will find some of the most commonly used ones today. The unique iterative process she used to design this family of functions, along with her other research results were presented in her next to legendary paper "Orthonormal Bases of Compactly Supported Wavelets" [29] in 1988. Her publication "Ten Lectures on Wavelets" [2], cited by tens of thousands of researchers and read by far many more, is commonly referred to as one of the best selling mathematics books of the 1990s.

The story of wavelets does not end here. In fact, many of the groundbreaking scientists already mentioned throughout the paper have continued their research and are frequently attributed to further research, development and findings in the

field. One such example is Yves Meyer who together with Ronald Coifman, Victor Wickerhauser and Steven Quake presented *wavelet packets* in 1989 [30]. These are often called a natural extension to the multiresolution framework as they introduce even more flexibility in the analysis process via additional parameters.

For further reading about both the preceding and other developments and additions to wavelet theory I highly recommend "The World According to Wavelets" [17] by Hubbard for an inspirational and intuitive approach, and "A Wavelet Tour of Signal Processing" [3] by Mallat for the more stringent and mathematically minded reader. Both these publications have been excellent sources for this text. We now depart the historical chapter and leave it up to the reader to find the bearing to his own final destination within the world of modern wavelet theory.

In the previous chapter we saw how the roots of wavelet analysis originates from a need to achieve a notion of time localization in the frequency domain and how the Fourier transform falls short when analyzing non-stationary signals. In this chapter we will introduce the Continuous Wavelet Transform (CWT) and describe its inner workings.

3.1 The Continuous Wavelet Transform

Learning about the modern history of wavelets, one cannot miss the French connection. Much of the groundwork was done in France or by Frenchmen abroad, and the word they all used for wavelet was *ondelette* (meaning "small wave"). However, most wavelets we use today do not just have to be "small" to qualify, but rather adhere to a whole set of criteria. In fact, they are even quite often purposefully constructed to target specific features of the signal to be analyzed, and in other ways designed with certain properties in mind.

Since a wavelet merely is a mathematical building block, it does not make much sense to describe them further without putting them into a context. This context is the transform in which they – or more specifically descendants of them – are used to *analyze*, or *decompose* (or in the inverse case, *synthesize* or *reconstruct*) a signal. In the continuous case, which Morlet and Grossmann worked with, we have

$$W(u, s) = \int_{-\infty}^{\infty} f(t)\psi_{u,s}(t) dt \quad (3.1)$$

which is called the continuous wavelet transform (CWT)¹ [2, 3]. As we recall, their findings were a rediscovery, and related to the so called *Calderón's resolution of identity*, formulated in the sixties by the Argentine mathematician with the same name [25]. In the transform (3.1), the two-dimensional function $W(u, s)$ is obtained by projecting a signal $f(t)$ onto the wavelet

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right) \quad (3.2)$$

¹Following [2], we have here implicitly assumed that ψ is real. For complex ψ , we would instead use $\bar{\psi}$ in (3.1) which is useful in some applications. The function $\bar{\psi}$ is the complex conjugate of ψ .

which is a translated by u , and dilated by s ($s > 0$) version – sometimes called *daughter* wavelet – of the original function ψ , often referred to as the *mother* wavelet. In more detail, translating the function $\psi(t - u)$, simply shifts it u units to the right, and the dilation $\psi(t/s)$ stretches it depending on the value of the scaling factor s (see Figure 3.1 for an illustration of this concept). The factor $1/\sqrt{s}$ is simply to keep the energy of the (scaled) daughter wavelet equal to that of the mother. Keeping the energy constant makes it easier to compare the transformed signal across all the scales, but in some applications other functions are more suitable instead to e.g. weigh the result towards certain scales containing features of specific interest [31].

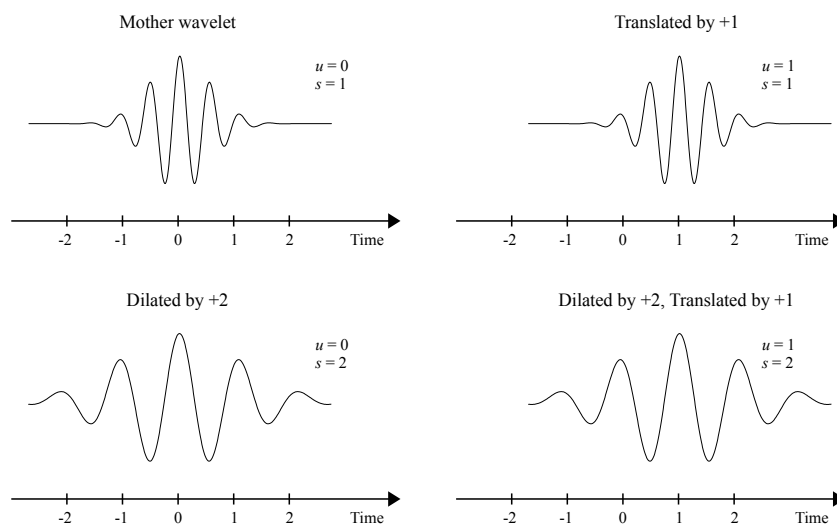


Figure 3.1: An illustration of a mother wavelet and three of its daughter wavelets generated by dilation and/or translation.

3.2 Computation and Interpretations

The CWT itself is seemingly straightforward but depending on which mathematical perspective or field of interest one has, it can have many different interpretations. For example, using the definition of an inner product we can write the transform as $W(u, s) = \int f(t) \psi_{u,s}(t) dt = \langle f, \psi_{u,s} \rangle$ which is common among mathematicians [3]. In this interpretation, it could be said to measure the cross-correlation between the analyzed function and the wavelets, i.e. the similarity between $f(t)$ and $\psi_{u,s}$ [20]. On the other hand, in the signal processing field, we can explain the CWT as a convolution of the input data sequence with a series

of functions derived from the mother wavelet, or as being the output of a certain bandpass filter [20]. The most intuitive explanation of the computation of the CWT is – however – perhaps the one in step-by-step (or algorithmic) form:

Assume that a mother wavelet is carefully chosen so that it meets the criteria for use in the CWT, and that the level of analysis needed, i.e. the number of scales, is determined. Following Figure 3.2, start with the most compressed function – representing the first, and lowest, scale (e.g. $s = 1$) but the highest frequencies – in the family of wavelets generated from the mother, and place it at the very beginning ($u = 0$) of the signal to be analyzed. Now, compute the wavelet coefficient value W by multiplying the current wavelet function ($s = 1, u = 0$) with the signal and integrate over its full duration. The result is then multiplied by the weighting function $1/\sqrt{s}$. In terms of the time-scale plane, we have now defined a value for the first point ($s = 1, u = 0$). Due to the inner product in the transform, the absolute value of $W(1, 0)$ can now be interpreted as how closely correlated that particular wavelet is with the signal at that specific position (large value means more similarity). To move on to the next point on the time-scale plane, keep the scale fixed but slide the wavelet – (since we are in the continuous realm) an infinitely small step Δu – to the right and compute the new coefficient $W(1, 0 + \Delta u)$. This is repeated until the full length of the signal has been covered and we have produced all coefficients for the given scale, $s = 1$. Finally, increase the scale parameter and repeat the above procedure as we move, scale-by-scale, with gradually increasing the scale towards the most stretched wavelet, capturing the lowest frequencies of the signal [32].

3.3 Requirements of the Wavelet Function

After having provided a context for the wavelets, it is now interesting to examine what mathematical properties of these functions actually make them earn their name. Although the Grossmann-Morlet definition of an *ondelette* was quite broad [4], they did include what is called the *admissibility condition* which can be written as

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < \infty \quad (3.3)$$

where $\hat{\psi}(\omega)$ is the (frequency dependent) Fourier transformation (2.3) of (time dependent) $\psi(t)$ [3, 24]. To ensure that the integral in (3.3) is finite we must have $\hat{\psi}(0) = 0$ [2, 3]. In fact, it can be shown that the spectrum $|\hat{\psi}(\omega)|^2$ must vanish at both extremes for the integral not to diverge. In other words, (3.3) implies that $\hat{\psi}(\omega) \rightarrow 0$ not only when $\omega \rightarrow 0$ but also as $\omega \rightarrow \infty$ and we can conclude that the wavelet function must have what we in signal processing would refer to as a bandpass like spectrum [33, 34].

This condition on the wavelet in the frequency domain obviously has consequences for its temporal representation. The zero at the zero frequency requires that the *average value* of the wavelet in the time domain must be zero, which is another way to express that wavelets cannot have non-zero DC components [20, 34].

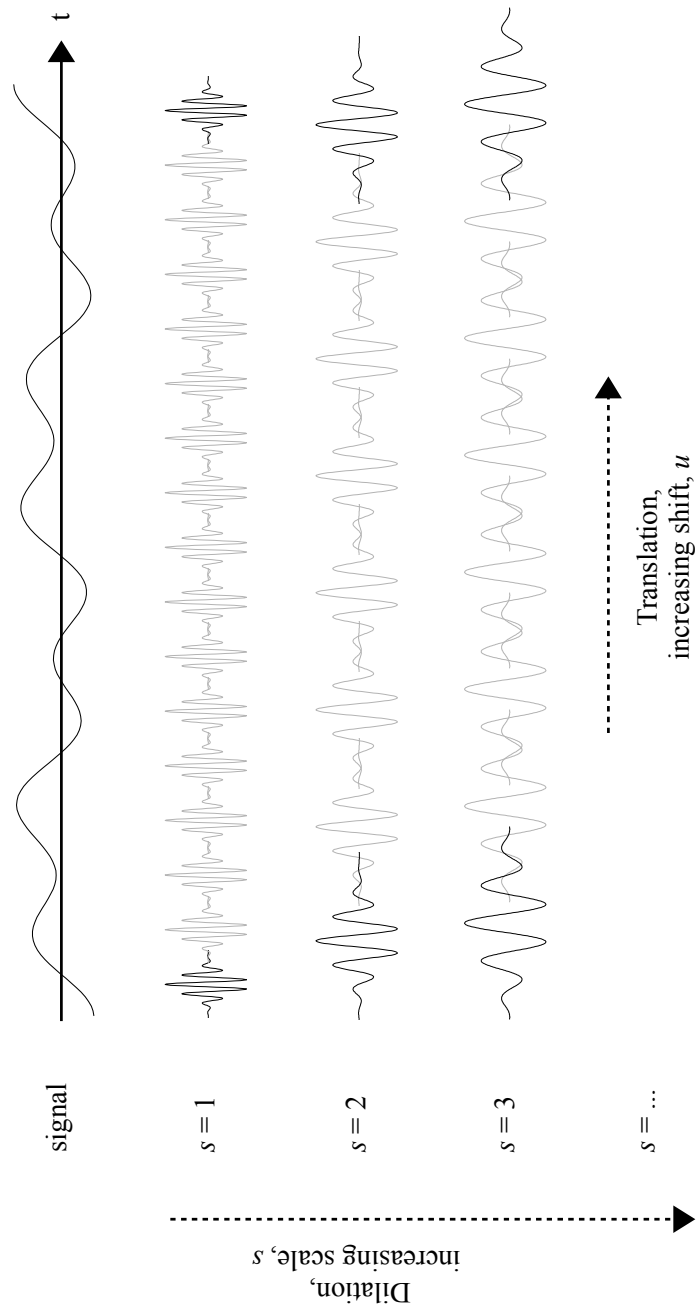


Figure 3.2: The general methodology of the wavelet transform.

This reasoning can be illustrated by the equality

$$\hat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (3.4)$$

which holds for functions oscillating around zero, hence said to explain the "wave" part in the word wavelet [2].

To fully ensure that a function is admissible as a wavelet, we must also show that $\hat{\psi}(\omega)$ is continuously differentiable, which according to Mallat in [3] can be done by verifying that the wavelet has sufficient time decay

$$\int_{-\infty}^{\infty} (1 + |t|)|\psi(t)| dt < \infty \quad (3.5)$$

Another time domain related property imposed on the mother wavelet – since it must belong to the $L^2(\mathbb{R})$ group of functions also called *square-integrable functions* – is that

$$E_{\psi} = \int_{-\infty}^{\infty} |\psi(t)|^2 dt = \|\psi(t)\|^2 < \infty \quad (3.6)$$

representing that the energy of the function simply must be finite [35]. These last two – albeit rather weak – decay properties can be used to support the fact that the wavelet must be well localized in time, contrary to e.g. an indefinitely oscillating sine wave used in the Fourier decompositions. This motivates the diminutive "let" in the word wavelet, and we realize that the word Morlet used for his functions – although in French – indeed suits them well.

In practice, there exist – in addition to the above basic (3.3) - (3.6) requirements – an array of supplementary conditions that one commonly imposes on the wavelet function in order to inherit desirable mathematical properties and adapt to specific applications. One such condition is that the wavelet must have *unit energy*, i.e. $E_{\psi} = 1$ in (3.6) [36].

3.4 The Inverse Continuous Wavelet Transform

Just like the other transforms we have visited so far – the (short-time) Fourier transform – the CWT also has an inverse under certain circumstances. When the admissibility condition (3.3) holds true, i.e. $C_{\psi} < \infty$, the *inverse* continuous wavelet transform is defined through Calderón's *resolution of identity* formula as

$$f(t) = \frac{1}{C_{\psi}} \int_{-\infty}^{\infty} \int_0^{\infty} W(u, s) \psi_{u,s}(t) \frac{1}{s^2} du ds \quad (3.7)$$

where $W(u, s)$ simply is the CWT as per (3.1) and $\psi_{u,s}(t)$ the wavelet function in (3.2) [2, 3]. Using this formula, a wavelet transformed signal can be synthesized again without any loss of information. Without its invertibility, the use of wavelets and the CWT would have been limited to the analysis (in a strict sense) of signals, losing much of its practical use. As an example, in certain smoothing techniques, one transforms the signal, removes all wavelet coefficients below a certain set threshold at some scale, before synthesizing it with the inverse transform again. This way, possibly even performed on multiple different scales, one have reconstructed a certain smooth version of the original signal.

3.5 Scale, Frequency and Heisenberg Revisited

It should now be well defined that the wavelet transform generates a function of two variables, *shift* and *scale*. However, we have so far mainly referred to the dimensions *time* and *frequency* throughout this thesis, and sometimes even incorrectly (for the sake of simplicity) almost implied that scale and frequency are interchangeable. Since we in this presentation tend to operate on signals – or functions – evolving over time, the *shift* corresponds to a location that is said to be *temporal* (as opposed to *spatial* when working with data in *space* e.g. in image analysis or data compression). Since we define the starting point of the analysis, say at time zero, a mapping between the temporal location and the time domain is rather evident. The relationship between scale and frequency is on the other hand, not as trivial. This has – not surprisingly – its roots in the already familiar Heisenberg uncertainty principle and the way our wavelets are constructed. Before going into details, let us first revisit the wavelet scaling parameter, s , and see how it affects the wavelet function's properties in the frequency domain in a broad sense.

When the scaling parameter increases;

- ⇒ the wavelet is *stretched* in time representing a *low* time resolution,
- ⇒ is *slowly* changing, analyzing *coarse* features,
- ⇒ overall *shifted towards lower* frequencies, and
- ⇒ obtaining *better* frequency precision.

Conversely, when decreasing the scale;

- ⇒ the wavelet is *compressed* in time representing a *high* time resolution,
- ⇒ *rapidly* changing, analyzing *fine* details,
- ⇒ overall *shifted towards higher* frequencies, and
- ⇒ obtaining *worse* frequency precision.

As we can see, scale seems to be – via the time domain – inversely proportional to a frequency interval, at least roughly speaking. The author of [20] chooses to illustrate this relationship by comparing it to the short-time Fourier transform where, at a fixed analyzing frequency of ω_0 , a change of window width will increase or decrease the number of cycles of ω_0 inside the window, whilst still keeping the frequency ω_0 fixed. On the other hand, in the wavelet case – where one could speak of wavelets having a *carrier frequency* of ω_0 – the same author states that the window width changes instead would mean dilation or compression, and that the carrier frequency now becomes ω_0/s for a window width change from T to sT . Note that here – in contrast to the STFT case – the number of cycles inside the window is still the same. In other words, the scaling parameter causes shifts of the *center frequency* along the frequency axis, hence making the wavelet's spectrum cover different bands at different scales as illustrated in Figure 2.3. As a matter of fact, the width of the frequency band it covers, better and formally known as the *bandwidth*, B , also changes. It is worth reiterating that a good time resolution, results in a poor frequency precision, that is, larger bandwidth.

The relationship between the central frequency and bandwidth is commonly known as the *fidelity factor*, or *quality factor* (Q), and is expressed as $Q =$

ω_c/B [37]. Later in this section, we will see that the relative bandwidth, B as later discussed in the context of frequency precision or *spread*, indeed like the wavelet's frequency also is inversely proportional to the scale. One now easily derives that if both bandwidth and central frequency are related to scale in the same manner, the yielded Q-factor becomes constant and independent of this scale. Hence, we say that wavelets generate a constant-Q analysis. Simply put, the constant-Q property is an expression of the fact that when the central frequency is increased as a result of decreasing scale in the family of wavelets, their individual bandwidth increases as well, yielding a poorer frequency resolution.

Though we above make references to a *carrier* or *center* frequency of the wavelet, it is important to note that no universal and precise mapping exists between scale and frequency in most cases. In fact, in the paper "Higher-Order Properties of Analytic Wavelets" [38], the authors emphasize that "it is critical to keep in mind that any assignment of frequency to scale is an interpretation, and there is in fact more than one valid interpretation". This can be further illustrated by the fact that some wavelets are highly irregular and some do not even have any dominant periodic components making it hard to speak of carrier – or center – frequencies at all in a general sense (see Figure 3.3 for four popular wavelets) [35]. For this reason, we will have to suffice with the comprehension that a wavelet's representative, or *characteristic* frequency is – if not in the eye of the beholder – at least a matter of definition specific to chosen wavelet function and application [31].

Albeit, if we conclude that this characteristic frequency, we call it ω_c , is determined, the relationship between frequency and scale can be expressed as $\omega_s = \frac{\omega_c}{s}$. Now, having that said, one of the elementary methods to define ω_c is to simply seek the frequency at which the maximum of the wavelet's Fourier transform magnitude, $|\hat{\psi}(\omega)|$, occurs, arguing that this is – in some sense – the "strongest" frequency in the wavelet. In [38], they call this interpretation the *peak frequency*. However, they also derive two other meaningful frequencies associated with a wavelet, one of which is called the *energy frequency*. This term is well-worth studying closer as such endeavor will not only further clarify this section's initial reasoning about the *time-scale* relationship, but also let us elaborate further on the Heisenberg dilemma and the uncertainty surrounding the *time-frequency* localization which we have touched on in previous chapters.

Let us start by revisiting an excellent, concise yet intuitive, presentation in the paper "The Continuous Wavelet Transform: A Primer" [35] already cited once before². The authors derive a pair of key quantities, the *center* and *standard deviation* of time and frequency respectively for the mother wavelet, which they use to define *Heisenberg boxes* in the same plane. The key here is to use the common interpretation and notion of probability density functions for which the center is the mean, and the spread the variance (or standard deviation). In the case of time, the center of the wavelet is defined as

$$\mu_{t;\psi} = \frac{1}{\|\psi\|^2} \int_{-\infty}^{\infty} t |\psi(t)|^2 dt \quad (3.8)$$

²In the following presentation we have switched indexes and variable names to match the conventions used in this thesis.

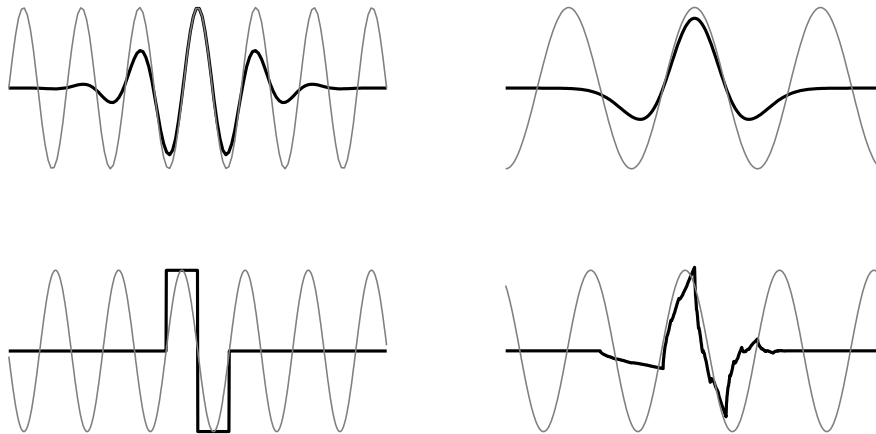


Figure 3.3: Four wavelets and their approximated center frequency. This illustrates the problem of a precise mapping. The wavelets are: Morlet (top left), Mexican hat (top right), Haar (bottom left), and Daubechies db2 (bottom right).

and the concentration around that center is described as the standard deviation

$$\sigma_{t;\psi} = \frac{1}{\|\psi\|} \left\{ \int_{-\infty}^{\infty} (t - \mu_{t;\psi})^2 t |\psi(t)|^2 dt \right\}^{\frac{1}{2}} \quad (3.9)$$

It then follows that the two quantities could be interpreted as the mean, μ_t , and standard deviation, σ_t , of the probability density function defined by $|\psi(t)|^2 / \|\psi\|^2$, hence creating an interval $[\mu_t - \sigma_t, \mu_t + \sigma_t]$ on the time axis where $\psi(t)$ should attain its most significant values. The corresponding quantities (and interval) for the frequency domain, $\mu_{\omega;\psi}$ and $\sigma_{\omega;\psi}$ are derived in an analogue way using the Fourier transform, $\hat{\psi}$, of the time dependent wavelet function ψ , instead. The two intervals now form the sides of a rectangle in the time-frequency plane

$$H_{\psi} : [\mu_t - \sigma_t, \mu_t + \sigma_t] \times [\mu_{\omega} - \sigma_{\omega}, \mu_{\omega} + \sigma_{\omega}] \quad (3.10)$$

which commonly is referred to as the Heisenberg box of the wavelet function ψ , which – as we have already seen – gives the time-frequency plane its characteristic rectangle-shaped partitioning. More specifically, we say that ψ is localized around the point (μ_t, μ_{ω}) in the (t, ω) -plane with *uncertainty* given by the product $\sigma_t \sigma_{\omega}$ which has a lower bound – the value $\frac{1}{2}$ in particular – governed by the uncertainty principle³. That is, the area of the box must remain constant, though the propor-

³The product is also commonly referred to in the literature in terms of the *variances* instead, which then yields $\sigma_t^2 \sigma_{\omega}^2 \geq \frac{1}{4}$.

tional relationship between its width and height may change. So far, the theory is rather general and could, for example, be applied to the window functions in the short-time Fourier transform instead of our mother wavelet (see [31]).

It is in the last part of the presentation in the above-mentioned paper, where our objective of relating scale to frequency, is truly aligned with the authors. They derive a scale and translation amended Heisenberg box

$$H_{\psi_{u,s}} = [u - s\sigma_t, u + s\sigma_t] \times \left[\frac{\mu_\omega - \sigma_\omega}{s}, \frac{\mu_\omega + \sigma_\omega}{s} \right] \quad (3.11)$$

by realizing that $\mu_{t;\psi_{u,s}} = u + s\mu_t$ and $\sigma_{t;\psi_{u,s}} = s\sigma_t$ in the time-domain and further that, $\mu_{\omega;\psi_{u,s}} = \frac{\mu_\omega}{s}$ and $\sigma_{\omega;\psi_{u,s}} = \frac{\sigma_\omega}{s}$ in the frequency domain. The now *daughter wavelet* specific box $H_{\psi_{u,s}}$ also assumes that the mother wavelet is centered around $t = 0$ (i.e. $\mu_t = 0$). Via the Parseval relation⁴, they finally conclude that a continuous wavelet transform $W(u, s)$ gives temporal information on $f(t)$ around the instant $t(u) = u$, with the precision $s\sigma_t$. Correspondingly, it gives frequency information about $\hat{f}(\omega)$ around the frequency $\omega(s) = \frac{\mu_\omega}{s}$, with the precision $\frac{\sigma_\omega}{s}$.

This elegantly modest mathematical detour has thus provided us with a better theoretical underpinning of the Heisenberg boxes and an expression of the uncertainty that surrounds the localization in both domains. Another outcome – although not as evident – is that we have derived the previously mentioned *energy frequency* which in our presentation goes under the symbol μ_ω and represents the mean of $|\hat{\psi}(\omega)|^2$.

As closing argument of this section, we discern that if a scale-frequency mapping is important to the application at hand, the choice of wavelet with respect to its frequency characteristics, will greatly determine the magnitude of the success in this regard. Further, the conclusions drawn from such analysis must emphasize the choice of frequency interpretation for the wavelet used in the mapping, and finally also account for the Heisenberg uncertainty imposed lack of precision.

⁴Loosely put, the domains time and frequency are equally complete representations of the same underlying signal, so they must have the same energy. Another way of putting it is that the sum (integral) of the square of a function is equal to that of its transform. This is true for all members of the Fourier transform family and commonly known as Parseval's relation [39].

Implementation

After first having explored where wavelets come from and what theoretical reasoning lie behind them, the previous chapter provided a brief exploration of some of the mathematical concepts involved as well as definitions and formalism. The first part of this chapter will provide a context along with an introduction to the domain in which we will let our wavelets operate – the analysis of a financial market. We will then present a software implementation based on the continuous wavelet transform.

The implementation is part of a C++ based code library developed to expose an application programming interface (API) to be called from a retail desktop trading software where it could aid in market analysis and possibly even trading decision making. In this part of the chapter, we will come across and discuss a set of practical issues and important implementation decisions. Here, we are interested in the difficulties and problems of transitioning from theory to application, rather than studying the performance of the wavelet based approach compared to other available methods in the industry. Finally, we will discuss some of the results and show examples of how the CWT software can be used, along with its drawbacks and benefits.

4.1 The Forex Market, Trading Software and Financial Time Series

Commonly referred to as the forex (or foreign exchange) market, this global financial system consisting of numerous distributed networks, is the marketplace where participants buy and sell – or exchange – currencies. These participants constitutes a diverse group, ranging from banks and large corporations maintaining positions for hedging purposes, to institutional actors and hedge funds speculating for profit, to name a few. Not only is the currency market decentralized and truly global in nature, it can be traded using a vast amount of vehicles such as cash, futures contracts, options or ETPs (exchange traded products) among others. Despite being the largest and most liquid financial market in the world, for most individuals, interaction with the forex market is hidden behind everyday transactions such as exchanging cash currency before making a trip abroad, a purchase from a foreign online retailer or paying for a hotel room with a credit card while vacationing in another country. What in these situations is referred to as the *exchange rate* –

the value of one country's currency expressed in terms of another currency – is in the forex trading realm known as the *spot price* of that specific currency pair. For example, one (1.00) Euro (EUR) could at one point in January 2014 be purchased for 1.36 US Dollar (USD), defining the price of 1.3600 for the EUR/USD instrument, commonly quoted in four decimals. When trading this pair in the spot market, the smallest price movement this instrument can make is usually called a *pip* (percentage in point) and is for most pairs 0.0001¹.

Though EUR/USD is by far the most traded pair – its trading volume alone exceeding the volume of all global equity markets added together – one sometimes speaks about the forex *majors* (EUR/USD, USD/JPY, GBP/USD, USD/CHF, and in some literature also AUD/USD and USD/CAD) which are the most liquid and together make up the lion's share of the traded volume in the currency market as a whole [40]. According to the Bank for International Settlements, the trading in foreign exchange markets averaged \$5.3 trillion per day in April 2013. This was up from \$4.0 trillion in April 2010 and \$3.3 trillion in April 2007 [41]. Looking back to 2001, the same figure is reported at \$1.2 trillion and it is not far-fetched to draw the conclusion that the globalization and an ever growing world economy and trade, has fueled this tremendous growth, emphasizing the need for – and importance of – the forex market and its well-being.

4.1.1 Retail Forex Trading

Part of this monumental financial system are the retail traders, individuals and smaller trading firms using the currency markets to trade pure speculative strategies, as investments in an alternative asset class pursuing diversification of their more traditional portfolios, or perhaps by placing bets as a long-term play of their macroeconomic predictions and views of the global economies. Together, these parties form the so called *retail forex market*, a growing group of traders representing a daily trading volume of more than 300 billion dollars in 2015 [42]. In the article "The Rise of Retail Foreign Exchange" [43], Betsy Waters – a director at Deutsche Bank at the time – summarizes the past decade of explosive retail trading growth by explaining that at first, the markets were simply not feasibly accessible to individual investors. Minimum trade amounts of as much as a million dollars, stacks of complex legal documents to be reviewed and signed along with extensive credit checks, were often all required before a financial institution would even consider trading with you [43]. Hence, *retail* trading was restricted to only a select few with enough resources and trading capital, something that would change with the next major milestone in the digital revolution – the advent of the Internet. As with so many other industries, when the Internet gained in popularity and became a household staple next to the phone line and cable TV, it created a money pipeline leading directly from the general investor's desktop. The opportunity was not only seized by the more traditional discount stock bro-

¹An exception is e.g. Yen-based currency pairs such as the USD/JPY, which is only quoted in two decimals making a *pip* worth 0.01. Note however, that some brokers have allowed trading using so called *fractional pips* (1/10th of a pip) meaning they introduced an extra digit of precision for a select range of currency pairs, adding to the special cases of quoting.

kerage firms, but also a variety of specialized broker firms dealing only in the forex market. These online forex trading brokers offers margin accounts with extreme leverage and minimum account deposits of only a few hundred dollars, enabling virtually anyone to trade the currency markets with the click of a button. Dubious brokerage firms, defrauded investors and misleading marketing strategies eventually caught the eye of the authorities and in recent years the industry has been heavily regulated followed by an even heavier consolidation. In the U.S. market, this lead to a much more mature online retail forex trading industry controlled by a few large brokerage houses offering more sensible leveraged accounts whilst being scrutinized and closely watched by the National Futures Association (NFA) by the order of the U.S. Commodity Futures Trading Commission (CFTC).

4.1.2 Trading Platforms

It is the aforementioned retail forex brokers – and their clients in particular – that interests us in the following sections of this chapter as we seek to apply an implementation of the wavelet transform as a tool among others, to aid in market analysis and perhaps ultimately, trading decisions. It lies within the broker's role to provide its clients with market data and connectivity, visualization software and analysis tools, as well as to facilitate the buy/sell orders and to overall act as a full-service interface to the market as a whole for its customers. Though varying between the brokers, one of the most commonly used software platforms managing all of the above is the MetaTrader system from the Russian based software vendor MetaQuotes Software Corporation [44]. Development began in 2000, but first after much evolution of the initial software and several iterations of version numbers and a name change, the MetaTrader 4 platform was released in 2005. Since then, it has become perhaps the most popular forex trading software on the market and an updated version, MetaTrader 5, was released in 2010.

On the broker's side (or server-side), the platform manages the customers' accounts, processes orders and serves as an interface and gateway to other trading systems, such as different electronic communication networks (ECNs) which actually facilitate the trading of currencies since it is a decentralized system lacking "regular" exchanges. On the end-user's side (or client-side), MetaTrader is a desktop (or mobile) application providing streaming quotes and charting of the market data, a user interface for order entry, and a library of analysis tools and features. The software also comes with a development environment which can be used to write custom trading signals – so called indicators – and automated trading strategies among other things. The platform specific programming language called MQL5 (MetaQuotes Language 5), is based on the concept of the C++ language. Hence, the MQL5 syntax is similar to any other modern object-oriented high-level programming language and has a wide set of basic functions as well as more application specific features such as order submission and management, graphical output to the charting engine, and manipulation of the market data. The custom development capabilities of the MetaTrader system has grown into a sophisticated environment supporting advanced tasks such as interfacing with other software systems (e.g. MATLAB, R, or other statistical packages) via linking of external libraries and the development of automated trading strategies including backtest-

ing and optimization of these using processor parallelism and a globally end-user powered distributed computing cloud called the MQL5 Cloud Network.

In a sense, these technological advancements together with the low latency 24-hour market access available to anyone, the retail trader now has the same tools at his disposal as only the most sophisticated Wall Street hedge funds had less than a decade ago. To some extent, this truly levels the field, setting it on a course destined for great independent research opportunities and more widespread interest in this pivotal financial market.

4.1.3 Financial Time Series and Technical Analysis

The most common way of observing a financial market is to examine the series of prices at which an instrument was traded or quoted during a time interval, at a certain time scale. For example, as briefly touched on previously, this could be the spot price of a given currency pair in the forex market. The near real-time stream of quotes delivered to the end-user's trading platform is often referred to as *tick data* where the time between the *ticks* can be measured in milliseconds. For many applications, this extremely high-frequency data puts unreasonable constraints on both storage space and computational efforts required to study, analyze, and make meaningful interpretations of the market. To overcome this, one usually samples or aggregates the tick data over more intuitively perceivable time intervals such as a minute, hour, day or week (though, it should be mentioned that a plethora of trading strategies and an entire industry sector is built upon the high frequency trading paradigm too).

The most popular aggregation technique takes four data points into account for each time interval, namely the beginning (Open) and ending (Close) values as well as the highest (High) and lowest (Low) observations during that same interval. Together, these four observations per time interval can comprise a so called OHLC time series. Strictly speaking, these are considered four different time series and in practice one usually refers to the one comprised of only close values since they imply how the time interval ended and is for example commonly used to compute the gain – or return – over a given time horizon. It is also this particular series of close values we will operate on in the following implementations, and conclude that it forms a discretely sampled, ordered and evenly spaced, time series with N real-valued variables (samples) which can be formally expressed as

$$\{X_t\} = X_0, X_1, \dots, X_{N-1} \quad (4.1)$$

where $t \in \mathbb{Z}$ and denotes the time index [5].

The broad and interdisciplinary field of time series analysis is well researched and in signal processing one is oftentimes particularly interested in *exploratory* analysis of the series' spectral properties. Though explanatory studies are undertaken on a regular basis in the trading realm as well, many retail traders equipped with trading software like the one described earlier, devote themselves to the time series analysis branches of *classification* and *forecasting* using so called *technical analysis*, or TA for short. Among "technical" traders in a broad sense, this term is the cause of much polarization with the so called *chartists* on one side, searching for repeatable patterns and predictive signals, and on the other side of the camp,

the *quants* relying on theoretical models with strict mathematical rigor and pure statistical testing to make their trading decisions. Though the two groups sometimes share some of the same tools from the toolbox of quantitative analysis, the latter group often stays far away from the TA label. The two aforementioned viewpoints are, however, likely the far extremes on a sliding scale of how quantitative sciences can be successfully applied to the analysis of financial markets. Professor David R. Aronson at Baruch College, New York makes an insightful reflection in the opening words of his book "Evidence-Based Technical Analysis" [45]. In this book, which both introduces and applies established statistical theory and rigor to the field of TA, Aronson proclaims that "[technical analysis] is comprised of numerous analysis methods, patterns, signals, indicators, and trading strategies, each with its own cheerleaders claiming that their approach works. Much of popular or traditional TA stands where medicine stood before it evolved from a faith-based folk art into a practice based on science. Its claims are supported by colorful narratives and carefully chosen (cherry picked) anecdotes rather than objective statistical evidence" [45]. The key takeaway of this section is that quantitative analysis, regardless of its level of sophistication, is a very popular undertaking – both used and abused – in an effort to find an edge in trading the markets.

Whichever side of the fence one stands on, this chapter will introduce wavelet theory as means to create a prototype of an additional tool in the technical analysis toolbox. The tool, best lending itself to a broad and subjective interpretation, aspires to add value in an overall classification of market "phases" by visualizing the intersection of temporal and spectral analysis through the continuous wavelet transform and its power spectra. After countless pages of historical review, theoretical discussion and most recently implementation context, the last and only thing remaining between us and the practical use of this knowledge is a shallow gap of theoretical adaptation to real world circumstances, which the following section seeks to overcome.

4.2 Bridging Theory and Practice

Bridging theory and practice is the recurring dilemma of the applied researcher. In the world of wavelets, there are a few fundamental constraints limiting the practical application of the theoretical framework. Even when operating inside these boundaries, rather innovative work-arounds have been developed over the years to enable practical implementations and produce meaningful interpretations. In this thesis, we bring wavelets into the realm of financial trading by implementing a software library applying the theory on a stream of discrete time series data, which raises the following high-level issues.

Firstly, in the preceding theoretical discussions, we have talked about how wavelets operate on *functions* or *signals*, which have been used almost interchangeably. Despite their unclear definitions it has not mattered much in the presentation so far. However, we can at this point conclude that the actual subject of analysis in the current context is – simply put – a discrete set of numbers. This plain realization does (to some surprise) *not* lead to a simplification of our task, but rather introduces a set of problems that we will now discuss.

On an almost philosophical level, we can still talk about operating on a function, namely the *market forces*. This function is however widely different from e.g. a sine or cosine as it is not well-defined and deterministic, but rather seemingly stochastic and even much unknown to us. Buyers and sellers meet in what is known as the *price discovery process*, governed by supply and demand. As an example, some participants such as banks or large corporations are seeking to hedge risk, which in turn is assumed by the pure speculators for a specific price, all in an efficient and continuous market. In other words, it is the observed output of this process, sampled at a specific rate over a given time interval, that forms the input to our wavelet computations.

This leads us to the perhaps most important issue as we realize that we have only a finite set of discrete data points. With a time series of fixed length we need to adapt our wavelet theory to operate on a finite interval, rather than the full function space $L^2(\mathbb{R})$ as in the previous chapter. The main problem when adapting the wavelets to live on the interval are the so called *boundary conditions* which concern the effects of applying the inner workings of the wavelet transform at the very beginning and end of the time series. The convolution scheme used in the transformation computation inherently uses data "outside" the data interval at the ends. A very commonly applied strategy to make up for these missing values, is to pad the data at the ends in order to extend the interval. Common choices of padding values are zeroes, the last/first data value, some other arbitrarily chosen constant, or a sequence generated by a more sophisticated method such as the fit from a polynomial model, etc.

Though the plain zero padding technique likely is the most straight-forward to implement (also our choice in the next coming section's example implementations), the classical approach is the one of circular, or periodic, handling. The approach is thoroughly described in the literature [5, 46], and in essence makes the data "wrap around" at the end-point, assuming that X_{N-1}, X_{N-2}, \dots are appropriate substitutes to X_{-1}, X_{-2}, \dots creating an N -periodic sequence. Though useful in theory and some special scenarios, this method impose a major drawback when used with real-world data where the beginning and end values of the data series are at a far distance from each other. As long as the data is not truly periodic, the introduced artificial singularities where the data is "glued" together will cause distortions among the coefficients representing these areas of the signal.

One way of handling the aforementioned problem is to reflect, or mirror, the data sequence at the end-point, hence creating a new signal of length $2N$. By adopting this strategy, we avoid the artificial discontinuities whilst also benefitting from keeping both the sample mean and variance of the original data [5]. This can be interpreted as we now analyze the series

$$X_0, X_1, \dots, X_{N-2}, X_{N-1}, X_{N-1}, X_{N-2}, \dots, X_1, X_0 \quad (4.2)$$

instead. As a result, X_0, X_1, \dots serves as the substitutes of X_{-1}, X_{-2}, \dots if we now put this new $2N$ long series back into the circular scheme above, creating a seamless joint between the two ends.

Though we have accounted for some of the more popular methods of handling the issues of the interval boundaries, one must keep in mind that these are merely ways of mitigating the problem and some edge effects will almost always taint

the analysis in one way or another. A common choice is to clearly highlight the (most heavily) affected wavelet coefficients in the presentation, or to remove them permanently from the results and further analysis.

This section would not be complete however, without the brief account of a completely different take on the conundrum. More specifically, apart from manipulating the input data to address the issues and constraints of "the interval", one can also try to modify the wavelets themselves to operate on a finite data series. This requires a significant effort as previous chapter showed that is hard enough to create new wavelets as it is, and applying an additional set of constraints on the wavelet making process brings a whole new level of complexity. In academia, the interval one often choses to discuss is the unit interval, $[0, 1]$, and without going into to details we will here simply mention two early and noteworthy efforts of constructing wavelets for this interval. The first one is by wavelet legend Meyer, who constructed *boundary wavelets* [47] based on Daubechies' compactly supported bases. The approach he used had a few significant disadvantages which were addressed by Daubechies herself about a year later, together with Cohen and Vial [48]. Since these two seminal efforts of constructing "wavelets on the interval", research have been made to approach it from new angles and to generalize the previous methods. Some of this research can be found in [49–51].

4.3 Implementing the Continuous Wavelet Transform

The primary goal of this section is to present and discuss a practical implementation of the continuous wavelet transform (CWT), and to demonstrate some of its interpretation in an empirical environment. As we will focus on the practical issues and the transform's application to a specific domain, many of the theoretical details will be presented without much rigor. Instead, this section will frequently refer back to the previous chapter and to Christopher Torrence and Gilbert P. Compo's seminal and indeed practical paper "A Practical Guide to Wavelet Analysis" [52]. Our implementation closely follows the one written by the authors and made available together with their text [53]. Though the methodology and approach is very similar, sometimes even identical to the aforementioned authors', it should be pointed out that it is *not* a pure line-by-line translation into a new programming language alongside their libraries [53] already written in MATLAB, Fortran and IDL. The main high-level differences are that our approach aspires to serve as a practical and object oriented C++ based framework rather than functional code snippets, utilize parallelism though threading at the transformation of multiple scales to increase performance, and finally to provide an application programming interface (API) to a third-party application where it is consumed, ultimately presenting graphical output to the end-user aiding in the interpretation of the wavelet coefficients. It should be further noted that, though the implementation is fully functional and achieves its goals well, it does have limitations and flaws and should not be seen as a production-ready system.

4.4 Implementation Approach

To implement the continuous version of the wavelet transform as defined in (3.1) we start by concluding that we are operating on a discrete sequence of input data as given by (4.1). As with all computer implementations, most operations will be performed on discrete data structures such as arrays, hence prompting a discussion as to what really is *continuous* about the CWT in this case. Technically, the CWT used herein is discretized, however to call it a *discrete wavelet transform* (DWT) would cause a conflict as this name is in the literature reserved for a very specific discretization scheme. Ideally, the CWT would be discretized *very* finely in terms of the scale parameter. In the DWT however, the scale parameter is discretized to integer powers of 2 ($2^j, j = 1, 2, \dots$) and the translation parameter is set to be proportional to the scale so that at scale 2^j , one would translate by $2^j m$ (m is positive integer). For an overview comparison and further treatment of the DWT, see [2, 3, 54]. We will now continue – as in most literature and software libraries – with the convention of using the name CWT as it operates under the finely discretized approach.

A consequence of working in a discrete and digital realm is that the use of integrals, such as the one found in definition (3.1), has to be replaced for practical reasons. Following the discussion in [52], we realize that the transform can be defined as a convolution operation of the input data and, translated and scaled, versions of the mother wavelet, ψ . Further, the authors deduce that in order to approximate the CWT, the convolution should be performed N times for each scale, where N is the number of data points in the time series [52]. Now, recall the *convolution theorem* stating that a convolution in the time-domain is the point-wise multiplication in Fourier space under certain circumstances [39]. Utilizing this, we speed up our computations by simply applying a discrete Fourier transform (DFT) to the input, perform a point-wise multiplication of that array with the wavelet function defined in frequency space, before applying the inverse Fourier transform on the result, bringing us back to the time domain. By using this trick we not only achieve more efficient code in terms of speed, we also gain an implementation advantage since thoroughly tested DFT algorithms are readily available to us. The Fourier library of choice used throughout this project is the "Kiss FFT" [55] library written in C by Mark Borgerding which is a "reasonable efficient" implementation of the fast Fourier transform (FFT) mentioned in the side-note under Section 2.2.1.

With the above overall approach in mind, we now turn our attention to the wavelet function itself. The authors of [52] discuss three common wavelet basis functions (Morlet, Paul, and Derivative of Gaussian [DOG]) and their properties. Dissected in numerous academic papers, and compared in a wide array of applied research areas, the choice of wavelet function is a broad topic to cover and is mostly considered out of scope for this presentation. However, to mention a few considerations, some wavelets have properties that highlight specific features of the analyzed data (wavelet's "shape"), others are more suitable for more precise time localization (wavelet's "width"), whilst all of them bringing trade-offs in other areas instead [52].

In any case, the perhaps simplest wavelet – which we also use in the following

examples – is the Morlet wavelet defined as

$$\psi(t) = \pi^{-1/4} e^{iw_0 t} e^{-t^2/2} \quad (4.3)$$

where w_0 is the parameter known as the central frequency (see discussion in Section 3.5). Arguably called the "original", the Morlet wavelet is in essence a sine wave multiplied (localized) with a Gaussian window creating its characteristic shape illustrated in Figure 3.1 [56]. Several versions of the Morlet wavelet exist in the literature and sometimes a correction term is added in (4.3) to satisfy the admissibility condition (3.3) (see discussion and proofs in e.g. [57, 58]). For large values of w_0 however, the correction is negligible and the wavelet is admissible for practical purposes [57]. The choice of $w_0 = 6$ is used in [52] and in our implementation as well.

Having defined the wavelet function in the time domain, we also recall that a definition of how the function behaves in Fourier space is necessary. This will be utilized in the nifty convolution theorem based implementation strategy described earlier. We again turn to [52] who in their "Table 1" summarize the three wavelet types and their properties in this regard. For the Morlet wavelet, the Fourier transform is given as

$$\hat{\psi}(sw) = \pi^{-1/4} H(w) e^{-(sw-w_0)^2/2} \quad (4.4)$$

where $H(w)$ is the heaviside step function defined as $H(w) = 1$ if $w > 0$ and $H(w) = 0$ otherwise [52]. To explicitly impose $\hat{\psi}(0) = 0$ is another way to ensure admissibility as per Section 3.3 [57]. Without providing further details, we now conclude that the Morlet wavelet as described above is indeed appropriate to use in our continuous wavelet analysis framework.

4.5 The Software Library Architecture

The main portions of the software solution was implemented in C++ utilizing the object orientated paradigm to as much extent as possible. The approach forms a shared library resulting in a dynamic linked library (DLL) file with "C style" exposed functions. When compiled as binary, this file can be linked from external applications and systems, hence bringing the wavelet analysis implementation to any third party platform supporting standard DLL imports. Thus, by confining most of the logic in the common library file, very little platform specific code needs to be written as opposed to implementing the wavelet algorithms for each software package of interest. The following gives an overview of the architecture as well as a few in-depth explorations at particularly interesting parts.

As we can see in Listing 4.1, the main entry point of the DLL is the function *fnCWTPowerSpectra(...)* which has the input data, a flag for wavelet type, and list of scales to perform the analysis on, as main arguments. Since we are in the realm of C/C++ we rely heavily on the concept of input/output buffers with pointers throughout the implementation. After some input and argument checks, this function initializes the appropriate wavelet object and instantiates the CWT engine with both the wavelet itself, and data to be analyzed, as input. The analysis

is performed at each scale independently, lending itself as a good candidate to parallelism. Thus, the transform at each scale is launched asynchronously using threaded tasks which are finally joined to bring the execution together before returning the full result to the end-user.

The inner workings of the CWT object are conceptually nearly identical to the [53] implementations. A mandatory zero padding of the input data to bring its number of samples up to the even next power of 2 is followed by a call to the FFT algorithms bringing the data into the frequency domain. We have already mentioned that the choice of wavelet is determined by a flag at the beginning of the execution, but only the Morlet wavelet has been implemented here. However, using a full-fledged object oriented approach with inheritance and polymorphism the architecture is very modular and the creation of a new wavelet type is as simple as deriving from the base class and finally providing only the implementation details specific to that wavelet (e.g. its definition in Fourier space, $\hat{\psi}$). For a thorough treatment of the implementation specifics of all the wavelet types, see [52, 53]. In their implementation – available in three different language flavors – they have made a great effort of commenting on details and to reference the actual definitions and theory sections in the paper directly alongside their source code in [53].

Once we have obtained an array representation of the wavelet in the frequency domain at a certain scale inside the function `fnTransformScale(...)` in the CWT object, we multiply it element-by-element with the padded and Fourier transformed input. To bring the final result back to the time domain, we simply perform an inverse Fourier transform after which we truncate the array to remove the samples added by the padding operation. The aforementioned illustrates how relatively simple the actual transform implementation has become using this approach, and an outline of this part of the source code is found in Listing 4.2. The overall architectural approach is documented in Figure 4.1 which shows the high-level schematics of the algorithm and library core.

4.6 Consuming the Library in MetaTrader

As previously described, MetaTrader is a popular trading platform with extensive functionality to trade and visualize financial markets. The built-in integrated development environment (IDE) enables the user to develop automated trading strategies and extensions to the software using the MQL programming language. The type of extension we will be utilizing is a so called *indicator*. These scripts can contain custom code for proprietary analysis methods and support interaction with the platform's user interface by means of altering the charts, adding variables and signals, as well as providing other visual output to aid the trader in his decision process. An indicator can, apart from using solely native MQL code, import the type of DLL files developed here, opening the platform for integration with practically any other software application and bespoke system, given they expose some means of external communication. By interacting with the developed DLL file, we can call our implemented wavelet functions with the financial data provided by MetaTrader as input, and later return the results to the trader in an visually

```

// +-----+
// |                                     WaveletIndicators.cpp |
// |      C++ DLL implementation of the Continuous Wavelet Transform |
// |      using the Morlet wavelet on a given number of scales. |
// |                                     Copyright 2014, Klas Eliasson |
// +-----+

// Entry point for the CWT in the DLL.
// Input buffer must be of size: signal-length * scales_size
// The first "signal_length" samples in "input" is the data to be transformed.
// Returns the CWT via the full "input" buffer on "scales_size" nbr of scales.
WAVELETINDICATORS_API int fnCWTPowerSpectra(const int whichTransform,
    double *input, const int input_size, const int signal_length,
    const double *scales, const int scales_size)
{
    // Ensure proper input and perform data checks.
    // ...

    // Fire up the engines...
    double dt = 1.0; // Sampling time
    vector<double> signal; signal.assign(input, input + signal_length);
    CMorlet *psi_morlet = new CMorlet(6.0); // Central frequency set to 6.0.
    CWavelet *psi = (CWavelet*) psi_morlet; // Any CWavelet derivative OK here.
    CCWT *cwt = new CCWT(signal, psi, dt);

    // Parallelized execution of a CWT at each scale
    vector<thread> threads;
    for(int i=0; i<scales_size; i++)
    {
        threads.push_back(thread(threadtaskTransformScale, cwt,
            scales[i], i, input));
    }

    // Wait for threads to finish up.
    // Delete pointers, clean up, and return.
    // ...
}

// Thread task function definition and pseudo code for illustration.
void threadtaskTransformScale(CCWT* cwt, double scale_val, int scale_idx,
    double *output)
{
    // Performing the CWT operation by calling cwt->fnTransformScale(scale_val);
    // Storing the result in the "output" buffer, clean up, and return.
}

```

Listing 4.1: C++ source code from the main entry point of the developed DLL showing the starting point of the execution and setup of the environment. Many implementation details are omitted to illustrate only the main line of execution.

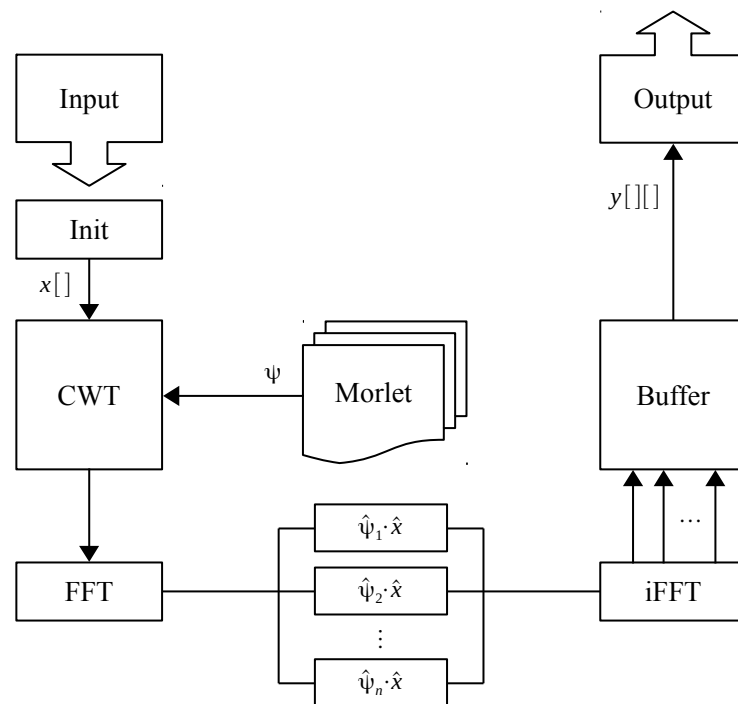


Figure 4.1: The high-level schematics of the implemented CWT algorithm and C++ software library. The one-dimensional input array x is transformed in parallel at each n scales, resulting in a two-dimensional output, y .

```
// +-----+
// |                                     CWT.cpp |
// |   Extract from the CWT class and the actual transform function. |
// |                                     Copyright 2014, Klas Eliasson |
// +-----+

vector<complex<float>> CCWT::fnTransformScale(double scale)
{
    // Initialize variables and obtain the frequency dependent daughter wavelet.
    // "x_hat" is of length "n" and contains FFT of padded input data.
    vector<complex<float>> daughter = psi->fnPsiHat(scale);
    // ...

    // Compute the transform by element-wise multiplication and perform iFFT.
    // Store the product back in daughter wavelet data structure to save memory.
    // ...
    kiss_fft_cfg cfg = kiss_fft_alloc(n, 1/*is_inverse_fft*/, NULL, NULL);
    for(int i=0; i<n; i++)
    {
        daughter[i] = daughter[i] * x_hat[i];
    }
    kiss_fft(cfg, (kiss_fft_cpx*) &daughter[0], (kiss_fft_cpx*) &coeffs[0]);

    // Truncate to "n_0" (original) length to remove padded data and
    // return the wavelet coefficients for this scale.
    coeffs.erase(coeffs.begin() + n_0, coeffs.end());
    return coeffs;
}
```

Listing 4.2: C++ source code extract from the CWT class showing the actual transform function. Some details are left out for better readability.

appealing way using MQL's native libraries for plotting and drawing onto the MetaTrader chart windows.

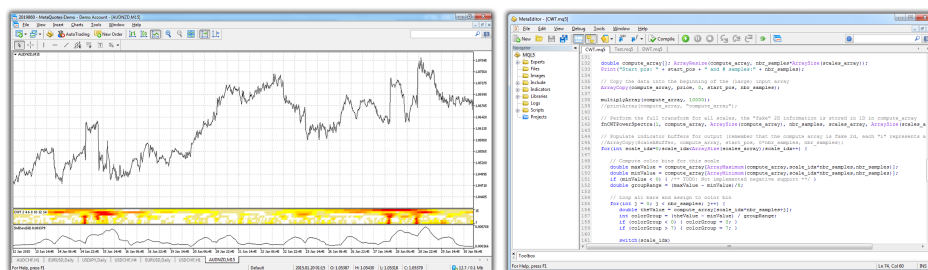


Figure 4.2: Screenshots of the MetaTrader main window (left) showing the price series graph and indicator window below, and the MQL development IDE (right).

Writing a custom indicator in MQL is rather straight-forward, especially when having some programming background knowledge. There are numerous templates, good documentation, and the product has a well-established community surrounding it. Listing 4.3 provides a skeleton on the script file highlighting only select code snippets illustrating some of the central concepts concerning the consumption of the developed wavelet library. From a technical perspective, one must first declare the interface and import the relevant DLL file, after which its functions are exposed and can be called as any other MQL function. The main code execution is performed in the indicator specific event function *OnCalculate(...)*, which is triggered by the MetaTrader platform as new data is streamed to the client from the server and is readily available for analysis [59]. At this point we invoke the *fnCWTPowerSpectra(...)* function from our DLL and send the input data along. As execution of the CWT is completed, our input buffer now serves as placeholder for the output as well, and we begin drawing the actual indicator onto the chart window. The end-result can be seen in Figure 4.2 (left) and we dedicate the next section to its exploration and interpretation.

4.7 The CWT Indicator Result Interpretation

In the field of signal processing, the study of *spectograms* are a common undertaking. These two-dimensional images are usually the preferred method for visual inspection of the result of a short-time Fourier transform (STFT) which was introduced early on in Chapter 2. The spectrogram can be plotted as a three-dimensional surface, or more commonly as a *heatmap* with frequency on the vertical axis and time on the horizontal axis. The intensity of the color in this representation can be said to reveal information about "how much" of a frequency the signal contains at a particular time. However, recall that it does so under all the constraints and drawbacks according to the discussion about fixed partitioning of the time-frequency plane and the Heisenberg uncertainty principle. Formally,

```
// +-----+
// |                                     CWT.mq5 |
// | MetaTrader Indicator performing a Continuous Wavelet Transform |
// | using the Morlet wavelet on a given number of scales. |
// |                                     Copyright 2014, Klas Eliasson |
// +-----+

#import "WaveletIndicators.dll"
int fnCWTPowerSpectra(int whichTransform, double &inpt[], int inpt_sz,
    int sig_len, double &scls[], int scls_sz);

#import

// Set indicator properties, allocate buffers, and implement OnInit().
// ...

// Define the scales used in the transform.
double scales_array[] = {1.0, 2.0, 4.0, 8.0, 16.0};

// This fnc is called whenever necessary to (re)calculate the indicator values.
// Arguments are the price data series and meta information about it.
int OnCalculate(const int rates_total, const int prev_calculated,
    const int begin, const double &price[])
{
    // Data checks and limiting the amount of data being analyzed.
    // ...

    // Copy financial data into the beginning of the input array "compute_array".
    ArrayCopy(compute_array, price, 0, start_pos, nbr_samples);

    // Perform CWT for all scales. Buffer "compute_array" will contain result.
    fnCWTPowerSpectra(1, compute_array, ArraySize(compute_array), nbr_samples,
        scales_array, ArraySize(scales_array));

    // Process output, render graphics, and return.
    // ...
}
```

Listing 4.3: MQL5 source code extract showing only a few central parts of the CWT indicator implementation. Specific MQL indicator infrastructure, buffer allocations, and graphics rendering are among parts left out for the sake of simplicity.

the spectrogram for a continuous function is defined as the squared modulus (magnitude) of its STFT as defined in (2.5), i.e. $|S(u, \omega)|^2$ [46].

As discussed earlier in this thesis, the output of the wavelet transform is a set of wavelet coefficients. Though there is room for more than one interpretation of these, the perhaps most intuitive one – in this context – is that each of the coefficients measure the similarity (or correlation) between a particularly scaled wavelet instance and the signal, at a certain time position. The result of the CWT can hence also be seen as two-dimensional image and it makes sense to find something similar to the spectrogram in the wavelet realm. The answer to this is the *scalogram* which is defined in an analogous fashion, namely the squared magnitudes of the CWT coefficients (as defined in (3.1));

$$|W(u, s)|^2 \tag{4.5}$$

which – as the name implies – is dependent on scale rather than frequency² [46]. The visualization techniques of the scalogram representation are the same as its Fourier based equivalent.

In our application, the heatmap approach was a good fit, rendering a clear visual distinction between – in relative terms – large and small values, consistent with the end-user's assumed intuition regarding the indicator's interpretation. By placing the indicator in a window below the actual price series graph, we allow the user to easily identify events of interest as the graphs visually line up along the time axis, see Figure 4.2.

In [52], the authors analyze time series generated in a meteorological context. In one of the examples, they study the scalogram (or power spectrum as they refer to it) generated by a Morlet wavelet based CWT of sea surface temperature data collected over more than a century. Among the observations, they are able to identify patterns of how the characteristics of their data set has changed over the years. Though their analysis goes deeper than this – and includes development of methods for significance testing and more – we are inspired to borrow some terminology from their meteorological setting and apply it to our analysis of the currency markets.

Rendering the power spectrum of the CWT of a currency pair's price data over a given time horizon could potentially reveal – or at least visually confirm – when there are transient events occurring in the ever-changing market. Recall that this is one of the major benefits of wavelet analysis; being able to – in the time domain – localize events expressed in the frequency domain, in an overall non-stationary setting. Analogous to a thermometer, the indicator could point the trader in the direction of the market being seemingly "hot" or "cold" at a certain time and on a particular scale (e.g. a span of minutes or number of hours). It would not be an indication of a specific *directional* price movement, but rather a gauge of sudden jump in either direction, creating discontinuities in the data series, or, temporary burst of high variance (or *volatility* to use the language of finance). It is well

²In the following implementations we use the Morlet wavelet where the scale-frequency mapping is particularly simple. The Morlet wavelet with $\omega_0 = 6$ gives $\lambda = 1.03s$, where λ is the Fourier period (i.e. the wavelet scale is almost equal to the Fourier period in this case) [52].

established in academia that these exact phenomena are some of the more famous stylized facts of financial time series which would speak for the indicator's usefulness. For example, the mathematician Benoit Mandelbrot, who is more known for his work on fractals and self-similarity (which curiously enough has strong ties to wavelets and multiresolution analysis), documented that the distributions of stock market return series often experience so called *fat tails*, or greater *kurtosis*. This means that large price jumps occur more often in these leptokurtic distributions than sometimes assumed using a model based on the gaussian normal distribution. The other characteristic, related to the former, is *volatility clustering*, also documented by Mandelbrot in 1963, which means that large changes tend to be followed by other large changes (of either sign) and, similarly, that small changes tend to be followed by more small changes [1]. Both these phenomena are the study of many researchers and industry professionals alike, and the CWT indicator could be an interesting tool aiding in the visualization of them a trading environment.

One can assert that the continuous wavelet transform is a good fit for the problem by the way it analyzes extreme changes in its input. "Intuitively, abrupt changes in a signal invariably produce high frequency components. Thus at small scales s , which correspond to high frequency regions, the scalogram will have high magnitudes." [20] Mathematically this can be exemplified by performing a wavelet transform on the *Dirac delta function*, $\delta(t)$, occurring at $t = t_0$, by inserting it in (3.1) with (3.2) as

$$W(u, s) = \frac{1}{\sqrt{s}} \int \psi \left(\frac{t-u}{s} \right) \delta(t-t_0) dt = \frac{1}{\sqrt{s}} \psi \left(\frac{t_0-u}{s} \right) \quad (4.6)$$

Here, as the wavelet is compressed when the scale becomes very small, or rather $s \rightarrow 0$, the modulus of the transform approaches a delta function, $|W(u, s)| \rightarrow \delta(t_0 - u)$. Thus, at small scales, the magnitude of the wavelet coefficients will be very large at $u = t_0$ which is the location of the original impulse [20]. Equipped with both intuition and theoretical underpinning, we now set out to study real market data using our implemented wavelet scheme.

As a first and illustrative example, we applied the CWT indicator to a price series based on the closing values of the AUD/NZD currency pair aggregated and sampled every 15 minutes during the month of January 2015. The pair, comprised of the Australian dollar and the New Zealand dollar, experienced a couple of interesting events generating two major features in the price data during the studied period. These were likely sparked by macroeconomic news releases in the two countries respectively, and generated a good data sample to illustrate both the price jump and to some extent, volatility clustering. The chart showing the price series and indicator can be found in Figure 4.3.

Looking at the scalogram generated by the CWT indicator we notice two groups of high activity, one to the far left and the other to the far right in Figure 4.3. The first one, starting just after the timestamp 22:45 on January 14th, contains a sharp upward movement followed a while later by an almost identical movement in the opposite direction. The CWT indicator gives high readings indicated by the darker color, and we can see that, though the first price jump gives a distinct short-lived indicator reading, the second one (after the downward movement which essentially reestablishes the price at the same level as prior to the event) rather

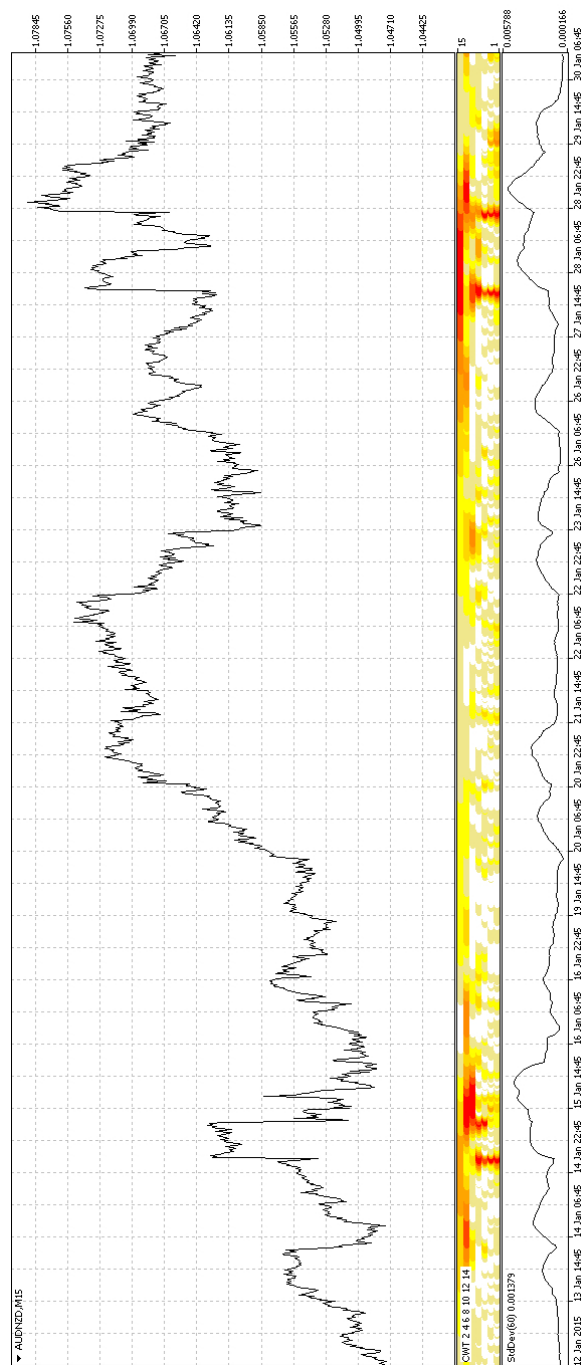


Figure 4.3: Screenshot from MetaTrader of the implemented CWT indicator applied to 15-minute data of the AUD/NZD pair during January, 2015. MetaTrader provided 60 sample windowed standard deviation indicator at the bottom.

consist of a somewhat prolonged period of large scalogram values in the higher scales. This tells us that the analysis has identified events on a broader time horizon (compared to the very sharp sample-to-sample jumps prior) during this period. Albeit, from Section 3.5 we know that the time resolution gets worse as the scales increase, so this is also an explanation as to why the pattern is seemingly smudged out in time at these higher scales. A visual inspection of the price chart confirms this, and we can indeed see an increased activity (wilder swings) in the currency pair after the major event. It is a well known fact that news releases can create these ferocious swings in the market, and it is easy to see how this can give rise to elevated volatility levels and a clustering effect while the new information is digested by the market participants. As a visual confirmation of volatility we have in this case also added a second indicator below the CWT scalogram. This is the standard deviation of the price returns using a sliding window of 60 samples. We can clearly see the "bump" during the event just discussed, although slightly shifted in time due to the lagging nature of the sliding window approach.

Continuing our example, after things calm down during January 16th, we see a long stretch of both calm volatility readings as well as a rather eventless CWT scalogram. However, things change drastically during the night between the 27th and 28th. An almost identical feature as in the first event presents itself in the data. A clear spike in the CWT indicator values can be seen on the sharp up-move, but a somewhat slower decline soon after, does not generate the same signal. The decline seems to be divided into two phases, spanning multiple data samples and hence not picked up as an equally sharp movement as prior jumps. This is a good example of how sensitive the analysis model is in regards to both data aggregation method, as well as scales chosen to be analyzed. Regardless, in the evening of the 28th however, we get another very distinct signal from the CWT analysis that yet another major price jump has occurred. Being in an already elevated volatility state, this swing also rhymes with the notion of volatility clustering and Mandelbrot's reasoning.

To summarize, despite being just a glimpse of a very limited example and a reasoning more intuitive than statistical, it is uplifting to see the CWT indicator in action, identifying price jumps and periods of increased volatility by signaling a "hot" market.

4.8 Limitations Caused by the Cone of Influence

Encouraged by the previous example of analyzing a historical segment of data, we now turn to some of the indicator's limitations, and start with one that limits its practical real-time use. Just as described in the discussion about bridging theory and practice above, we encounter a major obstacle when trying to analyze the most recent data points streamed to us (continuously plotted at the very end of the chart). The *edge effects* (that actually affects the beginning of our data set as well but is not usually of equal interest) introduce unfortunate artifacts at the perhaps most interesting point of the power spectrum, limiting the indicator's utility for near real-time analysis. Technically, it is the zero padding strategy chosen that produces a point of discontinuity at the very end, resulting in an

indication of the market constantly being very "hot" at the moment. As fresh data comes in, the prices at the previous points "cool off" more and more and, when at a certain distance from the end, has reduced its dependence enough from the zero-values beyond the edge, rendering them "safe" to include and trust in the analysis.

Though it can seem intuitive that the market always is "hotter" at the very edge, bordering into the unknown future where the market forces and price action try to settle for the next actual price level, there is actually a quantifiable measurement of the extent of this artificial effect. The authors of the paper which we commonly refer to herein, defined the *e-folding time*, a constant chosen for a particular wavelet so that the wavelet power for a discontinuity at the edge decreases by a factor e^{-2} [52]. This is done to ensure that the edge effects are negligible beyond that point in the analysis. For the particular wavelet we are using here, the Morlet wavelet, the *e-folding time* τ_s can be computed at each scale, s , as $\sqrt{2}s$.

As a matter of fact, the above reasoning does not only apply to the very edges. The scaling nature of the daughter wavelets creates a wider and wider set of affected samples centered around *any* event in the time domain as the scale parameter increases. Rising up through the scalogram as an expanding cone, the phenomena is commonly known as the *cone of influence* (CoI). In [3], Mallat discusses the CoI originating from an event at time v with a ψ that has a compact support equal to $[-C, C]$. He concludes that, since the support of $\psi((t - u)/s)$ would be equal to $[u - Cs, u + Cs]$, the cone of influence of v would be defined by $|u - v| \leq Cs$ [3]. This relationship is illustrated in Figure 4.4, and is as we can see, dependent on the choice of mother wavelet.

In other words, any major event – like a singularity midway into the time series – will heavily influence the wavelet coefficients in the scales above it, increasing its spread in time as it travels upward. The phenomena (which we have already gotten a glimpse of in the previous AUD/NZD analysis example) is clearly illustrated in Figure 4.5 where one can see the CoI and its characteristic shape³ caused by an abrupt event in the time series.

The event depicted in Figure 4.5 is "Francogeddon" (or at least so it was nicknamed by the trading community) and occurred on January 15th, 2015 after a surprise announcement from the Swiss National Bank. The central bank had for several years maintained a ceiling of the value of their Swiss Franc. The policy was maintained by buying up vast amounts of foreign currency to keep the value of the Franc below a rate of 1.20 Francs per Euro. The financial crisis and its aftermath had led to an enormous interest in the – traditionally considered – safe haven driving demand for, and valuation of, the Swiss Franc to levels where it had started to cripple the country's export industry. In a sudden move in early 2015, the central bank decided that the appreciation cap was no longer needed and the immediate market impact that ensued was next to indescribable [60]. Almost instantaneously gaining about 30 percent against the Euro, it created a market data event in the EUR/CHF pair that get as close to a theoretical singularity one can hope for in the context of our wavelet analysis illustrations. By studying this feature we can clearly illustrate how the CWT indicator reacts on the sudden price

³In this example, we have chosen the scales as a power of two, hence doubling the width of the affected samples (CoI) at each analyzed scale.

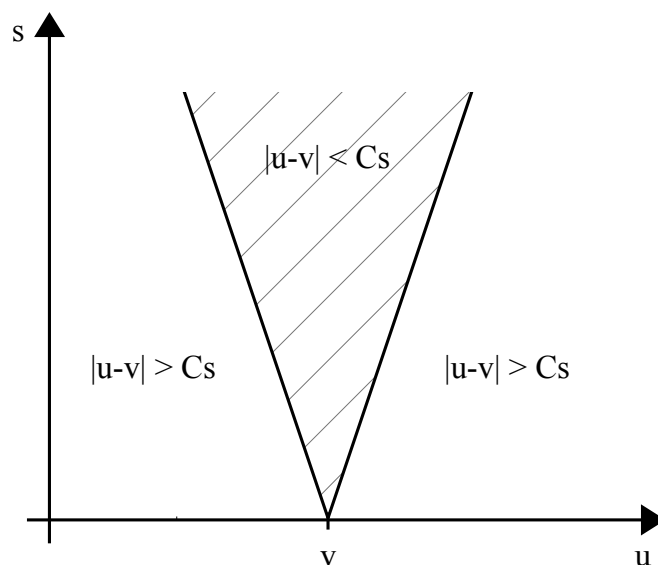


Figure 4.4: Illustration of the cone of influence of event v in the time-scale plane. Conceptually redrawn from [3].

jump at the lowest of scales, and how the cone of influence rises up through the higher scales, spreading out in the time domain⁴.

In the next chapter we will discuss the aforementioned results further and reflect back on the work performed in this thesis.

⁴As a side-note, we can also see traces of higher CWT readings scattered throughout the scales in the period following the event, reflecting the higher volatility that followed the "release" of the currency. Many forex traders went bust being on the wrong side of this trade and both trading and brokerage firms, even those partially hedged against the directional move as such, could not handle the increased volatility and negative client accounts that followed. Many well know names struggled, some were bailed out by loans, or even went bankrupt after not being able to meet and maintain the industry regulated minimum capital requirements.

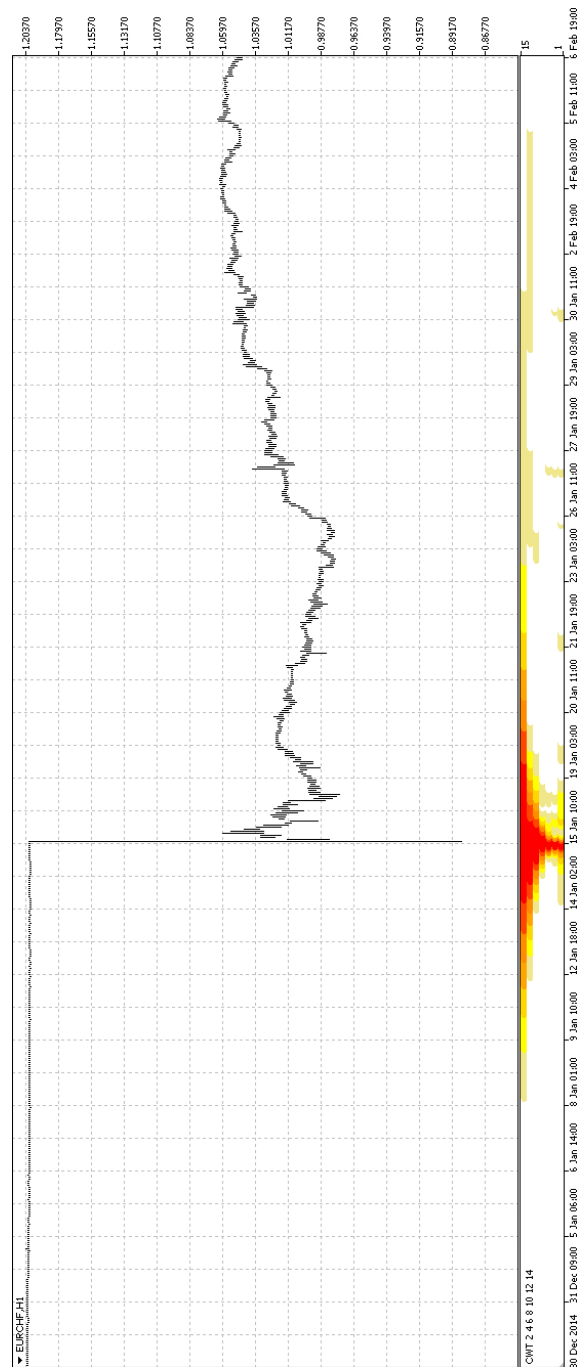


Figure 4.5: Screenshot from MetaTrader of the CWT indicator applied to EUR/CHF (sampled hourly) around the time of "Francogeddon" on January 15th, 2015. This event clearly illustrates the characteristics of the CWT power spectrum around a discontinuity in the data.

Discussion and Reflections

In the following section we will discuss the problem and results of the previous chapter and our implementation. We will also reflect back on the first chapters and touch on the questions that we raised in the introduction for these. Lastly, we will point to some interesting areas of further research.

5.1 Discussion

This thesis work set out to solve the problem of implementing a volatility exploring indicator for a desktop trading software using wavelet analysis. By approaching volatility from the frequency (scale) domain, we bring signal processing and a rigid analysis framework to the area of financial trading. After reviewing some of the vast theory behind wavelet analysis, we conclude that it is a theoretically viable method to approach to the problem due to its advantages over traditional Fourier analysis in the context of obtaining both a good temporal and spectral localization for transient events in non-stationary data. As we have explained, this is a result of the adaptive partitioning of the time-frequency plane.

While implementing the technical solution we stumbled upon several challenges in bridging theory and practice. The high degree of freedom that comes with the wavelet framework is also its own nemesis and the idiom "The Devil is in the details" certainly applies. Many of these issues were left for exploration in further research.

As we launched the application into the real world, we focused on illustrating the software's abilities with a couple of examples. We can conclude that the CWT indicator reacts visually on a set of interesting features in the data. In particular, we could see that large sudden price changes, creating jump discontinuities in the time domain of the data series, gave rise to large wavelet coefficient magnitudes. These were clearly visible in the scalogram in the shape of the cone of influence. It should be pointed out that our results here in themselves are no unique findings, and the use of wavelets in this regard is well-documented in the literature. For the interested reader, we recommend [61] or [62], the latter also including a brief overview of the considerable body of literature on discontinuity detectors that are not wavelet based.

We also reviewed and discussed volatility clustering, and could in the examples see that a large price movement indeed often was followed by a period of heightened

volatility as signaled by prolonged CWT indicator readings after these events. Though this effect is far more complex than what we have accounted for here, the application of wavelet theory to volatility analysis is indeed an interesting topic. A few further wavelet-based approaches to this exists in the literature, and [36] is an excellent book written by long-time finance and wavelet researchers.

Lastly and again, the results produced by our particular wavelet scheme heavily depends on a number of very specific implementation decisions. To generalize the analysis and interpret the results more scientifically, we could follow the efforts in [52] who show how to apply more rigorous statistical testing and derive e.g. significance levels in relation to a discussion of peak values in the power spectrum. Further quantitative analysis of the indicator readings of different price movement scenarios could further answer the question as to the contribution of this method to similar industry efforts.

Having successfully implemented a fully functional solution and used it in a real trading software with historical data, we have showed that wavelet analysis indeed is an interesting and viable area to explore in the context of trading and volatility. Though one could access and process the raw CWT output for further analysis, the visual market indicator lends itself better to a subjective, historical, and explorative analysis rather than, for example, creating a quantifiable and predictive trading signal in real-time.

As a remark, we further hope that the preceding implementation can serve as inspiration and starting point for further research to practitioners and financial software developers curious in the topic of applying wavelet analysis in a financial trading environment.

5.2 Further Research

We have already pointed out some of the shortcomings of our implementation. Many of these would have been interesting to investigate further, but due to time constraints, left for future work. In particular, the following areas of the implementation would greatly benefit from more research.

The first is the boundary handling of the time series data. We mentioned several methods available to us, and it would be interesting to explore this in more detail. With a somewhat philosophical reasoning, instead of the zero padding, we could for example implement a constant-value padding where we extend the data with its last known price point, to reflect that the equilibrium between buyers and sellers is assumed to continue as long as no new information enters the market. This would, perhaps, be more coherent with our understanding of the price discovery process or at least motivate a specific implementation. It is still to be determined which final method of data extension would result in the best mitigation of the boundary effects and their implications to the specific application at hand. To approach the problem from a different angle, a venture into "wavelets on the interval" and the modifications to both the functions and the transform to naturally adapt to the finite data set as briefly touched on, is certainly also an interesting study for the future. The importance of research in this area is particularly pressing in financial trading applications where the most recent data

usually is the focal point of study, and near real-time analysis is in high demand. Further, by implementing a larger set of mother wavelets in our software, we could explore other wavelets' properties, their strengths and weaknesses, along with the tradeoffs and sometimes conflicting relationships that they impose. The choice of wavelet function is always going to be an essential part of any implementation or analysis where the results are subject to (scientific) scrutiny.

Secondly, as our implementation time was limited, this is naturally an area that holds much room for further expansion. Without constraints, this implementation and the analysis of the CWT indicator's readings could be made with much more rigor. For example, it could be interesting to clarify which values are affected – and how much – by the cone of influence, or to provide a more extensive motivation behind the scales and data sampling intervals used. To its defense, the CWT indicator and its use within the MetaTrader platform, is intended to strengthen or confirm a trader's intuition about the market's behavior in a daily trading session, rather than quantitatively analyze the wavelet power spectrum. As long as it does so correctly within reasonable levels and under such assumptions, it serves its current purpose and an extension of the software to become more scientifically able is a separate undertaking of future research and development.

5.3 Reflections

In the first part of this paper, we came to realize that the story of wavelets is a story of extraordinary people working hard to advance the science and bringing it to practical application. From the outset, we aimed to provide a historical background that chronologically explores the development of wavelet theory by including these scientists, highlighting some of their contributions to the field as a tribute to their accomplishments. Initially driven by practical applications in the oil industry, modern wavelet theory was rapidly propelled through new advancements in computer science and vitalized by the dawn of the digital revolution which brought an exponential growth of sheer computing power. But apart from story-telling, the second chapter also tries to convey what fundamental problem the wavelet approach tries to solve, and that it provides a new perspective on an age-old dilemma. Simply put, we need wavelets to tackle the problem of joining the two dimensions, time and frequency, in a better way than was accomplished prior. By taking an adaptive approach under the constraints of the Heisenberg uncertainty principle – which could be said to separate the two – we can partition the joint-dimensional plane in a fashion that give us a good localization from both perspectives. This understanding was vital for the rest of the presentation, and great focus was put on it to lay a foundation for the next chapter.

In chapter three, we brought mathematical formalism rather than trivia and scrutinized the wavelet functions themselves as well as the continuous wavelet transform. In essence, the wavelet transform is similar to the windowed Fourier transform, however introducing the notion of scale by dilation of the wavelet – window – function during the analysis. This creates an adaptive – in contrast to a fixed – partitioning of the time-frequency plane which is beneficial in many applications. There is a general inverse proportional relationship between scale

and frequency, but no universal precise mapping exists and it depends on the mother wavelet function chosen for the transform. Though we touched on most of the central components of the core of the continuous wavelet transform, it should be pointed out that the branches of our wavelet tree only reach so far, and the roots only so deep, hence many parts have been left out of scope or merely just scratched on the surface.

In the final act we built a narrow but illustrative bridge between theory and practice in the setting of financial trading. After providing an introduction to the world's currency markets and its retail segment in particular, the continuous wavelet transform was implemented and the result turned into an indicator plugin for a popular desktop trading application to show how it can be used in this context. Hidden behind countless hours of C++ programming and debugging, the software came to reveal some interesting features and demonstrated via examples that it could serve its rudimentary purpose. By presenting our hands-on approach we additionally hope that the reader got a good glimpse of – and gained appreciation for – the process as such, of bringing a complex theoretical framework into a practical user-centered application.

As a final and ending remark, under the given constraints of the physical limitations of the time domain, I extend this thesis as a very small leaf, on a tiny twig, somewhere far out on a branch of Daubechies' wavelet tree, standing in the forest of functional analysis...

References

- [1] B. B. Mandelbrot. The variation of certain speculative prices. *The Journal of Business*, 36(4):394–419, 1963.
- [2] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [3] S. Mallat. *A Wavelet Tour of Signal Processing*. Wavelet Tour of Signal Processing. Elsevier Science, 1999. ISBN 9780080520834.
- [4] Y. Meyer and R.D. Ryan. *Wavelets: Algorithms and Applications*. Society for Industrial and Applied Mathematics, 1993.
- [5] D.B. Percival and A.T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge series on statistical and probabilistic mathematics. Cambridge University Press, 2000. ISBN 9780521640688.
- [6] J. J. O'Connor and E. F. Robertson. Jean Baptiste Joseph Fourier. <http://www-history.mcs.st-andrews.ac.uk/Biographies/Fourier.html>, 1997. [Online; accessed 21-March-2015].
- [7] R.X. Gao and R. Yan. *Wavelets: Theory and Applications for Manufacturing*. SpringerLink : Bücher. Springer US, 2010.
- [8] Jean Baptiste Joseph Fourier. Théorie de la propagation de la chaleur dans les solides. *Nouveau Bulletin des Sciences par la Société philomathique de Paris*, Tome I(6):112–116, March 1808.
- [9] Complete Dictionary of Scientific Biography. Fourier, Jean Baptiste Joseph. <http://www.encyclopedia.com/doc/1G2-2830901489.html>, 2008. [Online; accessed 21-March-2015].
- [10] Encyclopædia Britannica. Joseph, Baron Fourier. <http://www.britannica.com/EBchecked/topic/215097/Joseph-Baron-Fourier>, 2013. [Online; accessed 21-March-2015].
- [11] P.R. Halmos. *Finite Dimensional Vector Spaces*. Annals of mathematics studies. Princeton University Press, 1947.

-
- [12] A. Graps. An introduction to wavelets. *Computational Science Engineering, IEEE*, 2(2):50–61, Summer 1995. ISSN 1070-9924.
- [13] Nobelprize.org. *Dennis Gabor - Biographical*. http://www.nobelprize.org/nobel_prizes/physics/laureates/1971/gabor-bio.html, 2014. [Online; accessed 21-March-2015].
- [14] Ingrid Daubechies. The work of Yves Meyer. In Rajendra Bhatia, editor, *Proceedings of the International Congress of Mathematicians 2010 (icm 2010) (in 4 Volumes) - Vol. I: Plenary Lectures and Ceremonies, Vols. Ii-iv: Invited Lectures*, number v. 2-4. World Scientific Publishing Company Pte Limited, 2011.
- [15] D. Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93:429–441(12), November 1946.
- [16] Jan Hilgevoord and Jos Uffink. The Uncertainty Principle. <http://plato.stanford.edu/archives/spr2014/entries/qt-uncertainty/>, 2014. [Online; accessed 21-March-2015].
- [17] B.B. Hubbard. *The World According to Wavelets: The Story of a Mathematical Technique in the Making, Second Edition*. Ak Peters Series. Taylor & Francis, 1998.
- [18] Dominic Hyde. Sorites paradox. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2014 edition, 2014.
- [19] John Brignell. *The Uncertainty Principle*. http://www.numberwatch.co.uk/uncertainty_principle.htm, 2003. [Online; accessed 21-March-2015].
- [20] Y.T. Chan. *Wavelet Basics*. Springer US, 1994.
- [21] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [22] Ingrid Daubechies. Where do wavelets come from? a personal point of view. *Proceedings of the IEEE*, 84(4):510–513, 1996.
- [23] J. Morlet and A. Grossmann. Wavelets, Ten Years Ago. In Y. Meyer and S. Roques, editors, *Progress in Wavelet Analysis and Applications: Proceedings of the International Conference "Wavelets and Applications," Toulouse, France - June 1992*. Editions Frontieres, 1993.
- [24] Alexander Grossmann and Jean Morlet. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736, 1984.
- [25] Yves Meyer. *A Tribute to Jean Morlet*. 2008.

- [26] J. J. O'Connor and E. F. Robertson. *Alfréd Haar*. <http://www-history.mcs.st-andrews.ac.uk/Biographies/Haar.html>, 2006. [Online; accessed 22-March-2015].
- [27] Alfred Haar. Zur theorie der orthogonalen funktionensysteme [on the theory of orthogonal function systems]. *Mathematische Annalen*, 69:331–371, 1910.
- [28] Ingrid Daubechies. Ingrid Daubechies' Personal Biography. <http://web.math.princeton.edu/~ingrid/personalbio.html>. [Online; accessed 22-March-2015].
- [29] Ingrid Daubechies. Orthonormal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.
- [30] Ronald R Coifman, Yves Meyer, Steven Quake, and M Victor Wickerhauser. *Signal Processing and Compression with Wavelet Packets*. 1990.
- [31] P.S. Addison. *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance*. Taylor & Francis, 2002.
- [32] Robi Polikar. *The Engineer's Ultimate Guide To Wavelet Analysis – The Wavelet Tutorial*. <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>, 2001. [Online; accessed 28-March-2015].
- [33] V.M. Gadre. Lecture 23: Admissibility and Discretization of Scale, Lecture Notes in Advanced Digital Signal Processing - Wavelets and Multirate, Department of Electrical Engineering, Indian Institute of Technology Bombay. http://nptel.ac.in/courses/117101001/downloads/Lec-23_Script.pdf, 2011.
- [34] C. Valens. *A Really Friendly Guide To Wavelets*. 1999.
- [35] Luís Francisco Aguiar and Maria Joana Soares. The continuous wavelet transform: A primer. Technical Report 23/2010, NIPE - Universidade do Minho, 2010.
- [36] R. Gençay, F. Selçuk, and B.J. Whitcher. *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Academic Press, 2001.
- [37] Yunlong Sheng. Chapter 10. wavelet transform. In A.D. Poularikas, editor, *The Transforms and Applications Handbook, Second Edition*, Electrical Engineering Handbook. Taylor & Francis, 2000.
- [38] Jonathan M Lilly and Sofia C Olhede. Higher-order properties of analytic wavelets. *Signal Processing, IEEE Transactions on*, 57(1):146–160, 2009.
- [39] S.W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub., 1997.

-
- [40] *FXCM INC. FORM S-1, Securities Registration Statement, IPO Prospectus, Filed 09/03/10*. EDGAR Online, Inc., 2012.
- [41] Monetary and Economic Department. *Triennial Central Bank Survey 2013. Foreign exchange turnover in April 2013: preliminary global results*. Bank for International Settlements, 2013.
- [42] Patrick Graham and Mark Heinrich. *Daily retail FX volumes dip to \$317 bln in Q2 -industry site, Reuters*. <http://www.reuters.com/article/2015/07/30/markets-forex-volumes-idUSL5N10A3ZC20150730>, July 15 2015. [Online; accessed 30-August-2015].
- [43] Betsy Waters. *The Rise of Retail Foreign Exchange*. <http://www.nasdaq.com/investing/forex/the-rise-of-retail-foreign-exchange.aspx>, June 2009. [Online; accessed 06-April-2015].
- [44] *MetaQuotes Software Corp. Website*. <http://www.metaquotes.net/>. [Online; accessed 06-April-2015].
- [45] D.R. Aronson. *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. Wiley Trading. John Wiley & Sons, 2007.
- [46] R. Todd Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser Boston, 1997.
- [47] Yves Meyer. Ondelettes sur l'intervalle. *Revista Matemática Iberoamericana*, 7(2):115–133, 1991.
- [48] Albert Cohen, Ingrid Daubechies, and Pierre Vial. Wavelets on the interval and fast wavelet transforms. *Applied and Computational Harmonic Analysis*, 1(1):54–81, 1993.
- [49] Khaled Melkemi and Zouhir Mokhtari. Wavelet bases of the interval: A new approach. *International Journal of Mathematical Analysis*, 1:1019–1030, 2007.
- [50] Frank Koster and Michael Griebel. Orthogonal wavelets on the interval. 1998.
- [51] Laura Levaggi and Anita Tabacco. Wavelets on the interval and related topics. *Rend. Sem. Mat. Univ. Pol. Torino*, 57(2), 1997.
- [52] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.
- [53] C. Torrence and G. Compo. *Wavelet Software by C. Torrence and G. Compo*. <http://paos.colorado.edu/research/wavelets/software.html>. [Online; accessed 01-September-2015].

- [54] The MathWorks, Inc. *Continuous and Discrete Wavelet Transforms; Matlab, Wavelet Toolbox Documentation*. <https://www.mathworks.com/help/wavelet/gs/continuous-and-discrete-wavelet-transforms.html>, 2016. [Online; accessed 15-November-2016].
- [55] Mark Borgerding. *Kiss FFT - A mixed-radix Fast Fourier Transform Library Implemented in C*. <http://sourceforge.net/projects/kissfft/>, 2012. [Online; accessed 07-April-2015].
- [56] Jacques Lewalle. Tutorial on continuous wavelet analysis of experimental data. *Mechanical Aerospace and Manufacturing Engineering Dept., Syracuse University*, 1995.
- [57] Marie Farge. Wavelet transforms and their applications to turbulence. *Annual Review of Fluid Mechanics*, 24(1):395–458, 1992.
- [58] John Sadowsky. Investigation of signal characteristics using the continuous wavelet transform. *Johns Hopkins APL Technical Digest*, 17(3):258–269, 1996.
- [59] MQL5 Ltd. *Event Handling Functions; MQL5 Reference*. <https://www.mql5.com/en/docs/basis/function/events>, 2014. [Online; accessed 07-April-2015].
- [60] Ferdinando Giugliano. *Q&A: the Swiss central bank move explained, Financial Times*. <http://www.ft.com/intl/cms/s/0/36ee9504-9ccd-11e4-adf3-00144feabdc0.html#axzz3WgS8st2P>, January 15 2015. [Online; accessed 07-April-2015].
- [61] Yi Xue, Ramazan Gençay, and Stephen Fagan. Jump detection with wavelets for high-frequency financial time series. *Quantitative Finance*, 14(8):1427–1444, 2014.
- [62] Anestis Antoniadis and Irene Gijbels. Detecting abrupt changes by wavelet methods. *Journal of Nonparametric Statistics*, 14(1-2):7–29, 2002.



LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2018-625

<http://www.eit.lth.se>