

Wavelet Decomposition of a Function

Submitted in Partial Fulfillment of the Requirements for Graduation with Honors to the Department of Mathematics, Engineering, Physics, and Computer Science, Carroll College, Helena, Montana.

Michael L. Schauf

April 12, 1996

CORETTE LIBRARY CARROLL COLLEGE

CORETTE LIBRARY CARROLL COLLEGE



3 5962 00099 006

This thesis for honors recognition has been approved for the Department of
Mathematics, Engineering, Physics, and Computer Science by:

Anthony M. Szpilka
Director

John L. Schaf
Reader

[Signature]
Reader

12 APRIL 1996
Date

Table of Contents

Introduction.....	1
Fourier Analysis.....	3
The Fourier Series.....	3
The Fourier Integral.....	4
The Discrete Fourier Transform.....	5
Wavelets.....	6
Orthogonality.....	10
Haar Wavelet.....	12
Definition.....	12
Example.....	13
Mexican Hat Function.....	17
Definition.....	17
Frames.....	18
Voices.....	19
Analysis.....	22
Conclusion.....	28
Appendix.....	30
Haar Wavelet Break Program.....	30
Mexican Hat Break Program.....	33
Mexican Hat Build Program.....	36
Works Cited.....	38

Introduction

Computers today play a major role in practically every aspect of life, especially in the area of communications and signal processing. From e-mail and faxes to grocery market scanners and televisions, computers are relied upon to transfer and interpret almost every type information.

One such example is the transfer of fingerprints by the FBI (Brislaw 1278). Today the FBI has some 200 million fingerprint records, stored in the form of inked impressions on paper cards. As part of a modernizing process, the FBI began to digitize the records as 8-bit grayscale images. But fingerprints are so detailed that some 10 megabytes per card are required, making the current archive about 2000 terabytes in size. (Remember a 3.5" floppy disk holds only 1.5 megabytes). In addition to the current cards, the FBI also receives around 30,000 new cards per day. Therefore the FBI has joined the search with others to find a more simplified way of storing and transmitting complicated data. The use of wavelets to encode the digitized fingerprints requires significantly less stored data.

Since communication is often times composed of complex phenomena, engineers seek processes that will ease the transfer of information. They do this most commonly by decomposing the complex functions into simpler ones. This process originated in the early 1800s with Joseph Fourier (Haberman 75). Fourier found that complex phenomena could be represented as a sum of sines and cosines of various frequencies and amplitudes. Even though the trigonometric sines and cosines represent a decent basis and are well understood, they do have their drawbacks. Oftentimes a Fourier series requires a large

number of coefficients to be stored, along with the fact that it has troubles representing functions that have discontinuities. The difficulty arises from attempting to use smooth sine or cosine functions to model a discontinuous function.

In the past couple of decades a new method has been discovered by Jean Morlet and Alexander Grossman that further simplifies complex data (Graps 4). It is known as wavelet theory, and it uses the idea of a simplified orthonormal basis to represent complex functions. These basis functions are obtained by dilating and translating a particular starting function known as the "mother wavelet." Therefore they have the ability to focus in on specified parts of complex data. Like the Fourier series, wavelets can then be combined in a linear combination to represent the original complex data.

Two such examples of "mother wavelets" are the Haar wavelet and the Mexican Hat wavelet. In this paper I provide a description of the two wavelets. To show how they are used, I have written programs in Mathematica to break down a function into its sum of Haar or Mexican Hat wavelets. I then compare the wavelet sums with the original function to show the strengths and weaknesses of each approach. The goal was to see whether particular families of wavelets can accurately approximate localized or discontinuous functions by using significantly fewer coefficients than a Fourier series would require. In this way, I hoped to test the claims that wavelets offer a more economical method for encoding such functions.

Fourier Analysis

Fourier Series

The main branch of mathematics leading to wavelets began with Joseph Fourier in 1807 and his theories of frequency analysis (Haberman 75-92). Fourier asserted that a piecewise smooth, periodic function $f(t)$ can be represented as the sum

$$f(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad \text{eq(1)}$$

where $\omega = 2\pi / T$, T is the period of the function $f(t)$,

and the constant and coefficients are

$$a_0 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(n\omega t) dt \quad (n > 0)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(n\omega t) dt \quad (n > 0).$$

By using Euler's formulas,

$$\cos \theta = \frac{1}{2}(e^{i\theta} + e^{-i\theta}), \quad \sin \theta = -\frac{i}{2}(e^{i\theta} - e^{-i\theta}),$$

eq(1) can be written in its complex form as

$$f(t) = C_0 + \sum_{n=1}^{\infty} [C_n e^{in\omega t} + C_{-n} e^{-in\omega t}]$$

or

$$f(t) = \sum_{n=-\infty}^{\infty} C_n e^{in\omega t} = \frac{1}{T} \sum_{n=-\infty}^{\infty} \left[\int_{-T/2}^{T/2} f(\tau) e^{-in\omega \tau} d\tau \right] e^{in\omega t} \quad \text{eq(2)}$$

where $n \in Z$

and

$$C_0 = \frac{1}{T} \int_{-T/2}^{T/2} f(t) dt$$

$$C_n = \frac{1}{2} (a_n - ib_n) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-in\omega t} dt \quad (n > 0)$$

$$C_{-n} = \frac{1}{2} (a_n + ib_n) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{in\omega t} dt \quad (n > 0).$$

Instead of sines and cosines, the Fourier series is now represented as a sum of complex exponentials.

Fourier Integral

When a function is not periodic, we can express it in Fourier terms as a function with an infinite period. Therefore by eq(2),

$$f(t) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{-\infty}^{\infty} \left[\int_{-T/2}^{T/2} f(\tau) e^{-in\omega \tau} d\tau \right] e^{in\omega t}.$$

Since the only allowable frequencies are multiples of ω , then

$$\omega_n \equiv n\omega \quad \text{and} \quad \Delta\omega_n \equiv \omega_{n+1} - \omega_n = \frac{2\pi}{T}.$$

As $T \rightarrow \infty$, $\Delta\omega \rightarrow 0$ and the function is represented by a sum of waves of all possible wavelengths, giving

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(\tau) e^{-i\omega_n \tau} d\tau \right] e^{i\omega_n t} d\omega_n.$$

Based on the condition that $f(t)$ is piecewise smooth, we can then obtain the Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

and the Inverse Fourier Transform

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega.$$

Often times it is easier to analyze properties of a complex electrical signal by looking at the signal as a superposition of sinusoidal functions at different frequencies.

The Discrete Fourier Transform

In many instances it is more convenient to work in the discrete realm with Fourier analysis. The Fourier integral can then be represented as a summation

$$F(p\omega_o) = \sum_{n=0}^{N-1} f(n\Delta t) e^{-i\frac{2\pi}{T} p n \Delta t} \Delta t,$$

where

$$p = 0, 1, 2, 3, \dots, N-1, \text{ and } \omega_o = \frac{2\pi}{T} = \frac{2\pi}{N\Delta t} \text{ is the fundamental frequency.}$$

This is called the Discrete Fourier Transform and corresponds to samples of a continuous signal taken equally spaced in the frequency domain of the Fourier Transform.

Wavelets

Like sines and cosines of the Fourier series, wavelets are also basis functions used in representing other functions or data sets. In the wavelet "family", there exist a variety of different wavelets including smooth wavelets, compactly supported wavelets, wavelets with simple mathematical expressions, and many more. Each of these is called a "mother wavelet." To begin a wavelet transform, one "mother wavelet" must be chosen that best fits the data set to be analyzed.

The wavelet transform formula is defined as

$$(T^{wav} f)(a, b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \Psi\left(\frac{t-b}{a}\right) dt \quad \text{eq(3)}$$

or

$$(T^{wav} f)(a, b) = \int_{-\infty}^{\infty} f(t) \Psi^{a,b}(t) dt \quad \text{eq(4)}$$

where

$$\Psi^{a,b}(t) = |a|^{-1/2} \Psi\left(\frac{t-b}{a}\right) \quad \text{eq(5)}$$

(Daubechies 3). Here the function Ψ is the "mother wavelet" and the functions $\Psi^{a,b}$ are called wavelets. As $|a|$ changes, eq(5) covers different frequency ranges. Therefore large values of $|a|$ correspond to small frequencies and small values of $|a|$ correspond to high frequencies. Changing the parameter b moves the time localization center.

Often times for convenience a and b are restricted to discrete values. In doing so

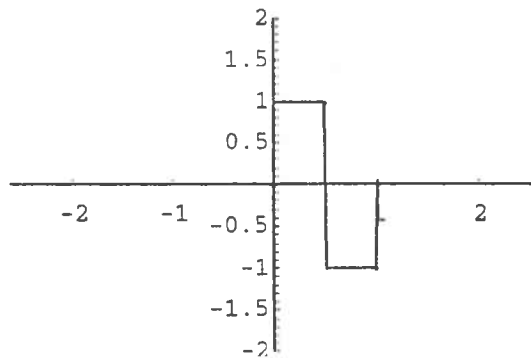
$$a = a_o^m, \quad b = nb_o a_o^m \quad \text{where } n, m \in Z \text{ and } a_o, b_o > 1$$

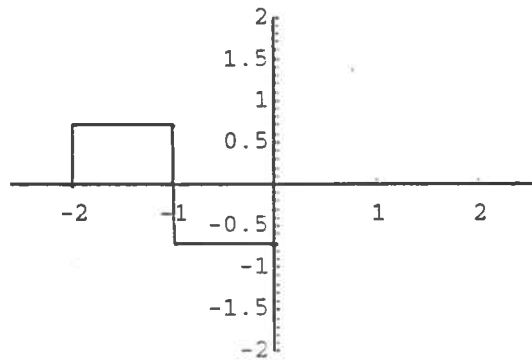
Substituting the given equivalents into equation (5), gives the discrete equation for $\Psi^{a,b}$ in terms of m and n

$$\Psi_{m,n}(x) = a_o^{-m/2} \Psi(a_o^{-m} x - n b_o). \quad \text{eq(6)}$$

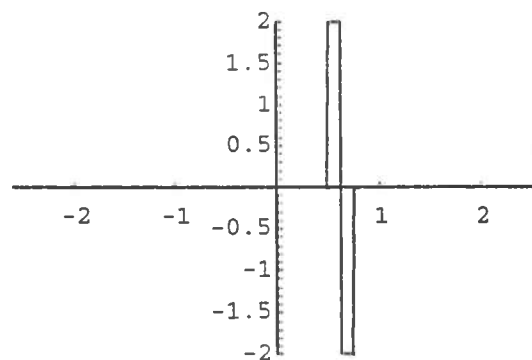
Wavelets differ from the Fourier series mainly in two important ways (Graps 1). The first is that wavelets are localized with respect to time and frequency, whereas the Fourier series is localized only with respect to frequency. Secondly, wavelets also have the ability to be dilated and translated, allowing a person to home in on particular sections of data. In equation (6), the value of m determines the horizontal compression and stretching, while the value of n translates the time localization center. To show examples of translation and dilation, the following are examples of Haar wavelets with $a_o = 2$ and $b_o = 1$.

Here is the plot of $\Psi_{0,0}$
The "mother wavelet"



$\Psi_{1,-1}$ 

As seen here the larger the m , the broader the graph. Also when n is negative it is shifted to the left.

 $\Psi_{-2,2}$ 

As seen here, the lower the m the thinner the graph. With n positive the graph is translated to the right.

This ability to translate and dilate is what makes wavelet analysis of a function superior to Fourier analysis. For example, to analyze broad features found throughout the data set, wavelets can use a large "window" to encompass all of the data. To analyze small, particular features, a small "window" encompassing only the feature would be used. Fourier analysis uses the same constant for all evaluations. As stated in a paper

introducing wavelets by Amara Graps, "The result in wavelet analyses is to see both the forest and the trees" (Graps 1).

Orthogonality

An important quality of both the sines and cosines of the Fourier series and of wavelets is orthogonality (Haberman 49-50). Orthogonal series play an important role in many areas of mathematics because they allow complicated operators to be simplified. If mathematicians are able to prove that a group of functions is orthogonal, they can then easily express other functions as a linear combination of the orthogonal ones.

Orthogonality is defined as:

A nontrivial sequence $\{f_n\}_{n=0}^{\infty}$ of real or complex functions is said to be orthogonal if

$$\langle f_n, f_m \rangle \equiv \int_a^b f_n(x) \overline{f_m(x)} dx = 0, \quad n \neq m, \quad n, m = 0, 1, 2, \dots \quad (\text{Haberman 49}).$$

Additionally, a sequence is orthonormal if also $\langle f_n, f_n \rangle \equiv 1, \quad n = 0, 1, 2, \dots$

A sequence that is orthonormal will prove to be important later on, because the orthonormality condition simplifies the process of computing the coefficients in a series expansion of the original function.

In the definition of orthogonality, $\langle f_n, f_m \rangle$ is called the "inner product" of two functions from the nontrivial sequence. The overbar denotes the complex conjugate. If a sequence of functions is found to be orthogonal, it represents a basis for a space of functions that satisfy appropriate admissibility conditions. By superposition, a function, $f(x)$, can be represented as an infinite sum of the basis functions $f_n(x)$:

$$f(x) = \sum_{n=0}^{\infty} a_n f_n(x). \quad \text{eq(7)}$$

To find the coefficients of the linear combination, we simply multiply both sides of eq(7) by an element f_m chosen from the orthonormal sequence $\{f_n\}$:

$$\langle f, f_m \rangle = \sum_{n=0}^{\infty} a_n \langle f_n f_m \rangle. \quad \text{eq(8)}$$

Because $\{f_n\}$ is an orthogonal system, $\langle f_n, f_m \rangle = 0$ whenever $n \neq m$. Therefore

$$\sum_{n=0}^{\infty} a_n \langle f_n, f_m \rangle = a_m \langle f_m, f_m \rangle.$$

Substituting the equality into eq(8) results in

$$\langle f, f_m \rangle = a_m \langle f_m, f_m \rangle.$$

Solving for a_m results in

$$a_m = \frac{\langle f, f_m \rangle}{\langle f_m, f_m \rangle}.$$

If the sequence $\{f_n\}$ is orthonormal, as well as orthogonal, the denominator is equal to 1,

leaving

$$a_m = \langle f, f_m \rangle.$$

These coefficients can be calculated easily by integrating and then stored for later reconstruction of the original function.

Haar Wavelet

Definition

The Haar wavelet, developed back in 1908, is the oldest and simplest of all wavelets. This "mother wavelet" is simply a step function taking on the values of 1 on the interval 0 to 1/2 and -1 on the interval 1/2 to 1:

$$\Psi(x) = \begin{cases} 1 & 0 \leq x \leq .5 \\ -1 & .5 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (\text{Daubechies 10}).$$

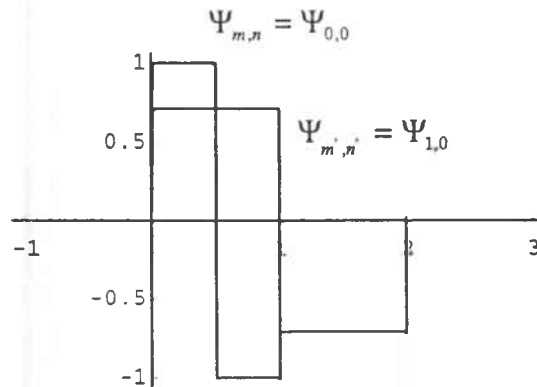
Referring back to equation (6), suppose $a_o = 2$ and $b_o = 1$. We then have

$$\Psi_{m,n} = 2^{-m/2} \Psi(2^{-m}x - n). \quad \text{eq(9)}$$

Proving that the Haar wavelets are orthonormal is quite easy (Daubechies 10-11).

Since $\Psi_{m,n}$ is non-zero only on the interval $[2^m n, 2^m(n+1)]$, then two Haar wavelets that have the same scale (m values) never overlap, therefore $\langle \Psi_{m,n}, \Psi_{m,n'} \rangle = 0$ for $n \neq n'$.

Overlappings do occur if the two wavelets have different scales (m values). For example, if $m < m'$ then $\Psi_{m,n}(x)$ lies entirely in the region where $\Psi_{m',n'}(x)$ is constant as seen below.



Therefore the inner product of $\Psi_{m,n}(x)$ and $\Psi_{m',n'}(x)$ is then proportional to the integral of $\Psi_{m,n}$ itself which is 0. In making this sequence orthonormal, the inner product of the function with itself must be 1. To achieve this, the "mother wavelet" must be multiplied by the constant, $2^{-m/2}$. That this constant, $2^{-m/2}$, makes the Haar wavelets orthonormal may be proved by noting

$$1 = (\text{const})^2 \int \Psi^2(2^{-m}x - n)dx = (\text{const})^2 * 2^m \int \Psi^2(t)dt = (\text{const})^2 * 2^m$$

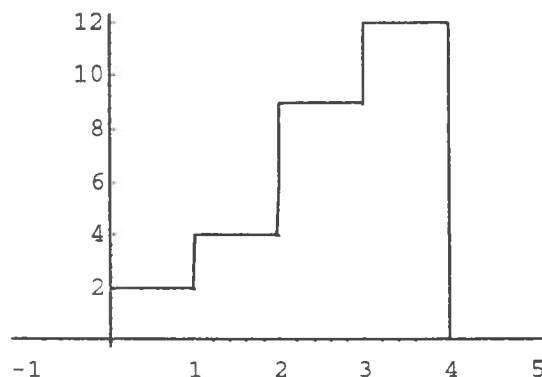
(Muller and Vidakovic 6).

Quite often for most wavelets there exists an alternative algorithm used to find the coefficients, rather than finding them by the inner product. The following is an example of the algorithm for Haar wavelets.

Example

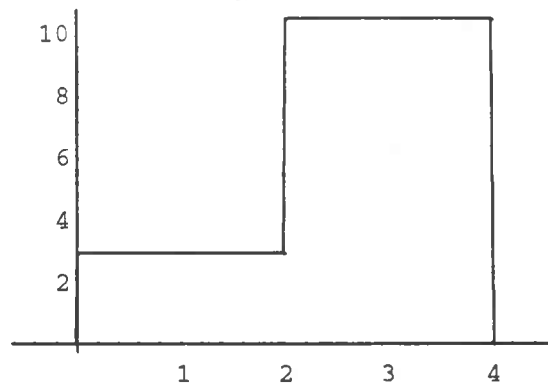
Let the function to be analyzed be a step function defined as follows:

$$f(x) = \begin{cases} 2 * \bar{x} & 0 \leq x < 2 \\ 3 * \bar{x} & 2 \leq x < 4 \\ 0 & \text{elsewhere} \end{cases} \quad (\text{the overbar here denotes the ceiling function})$$

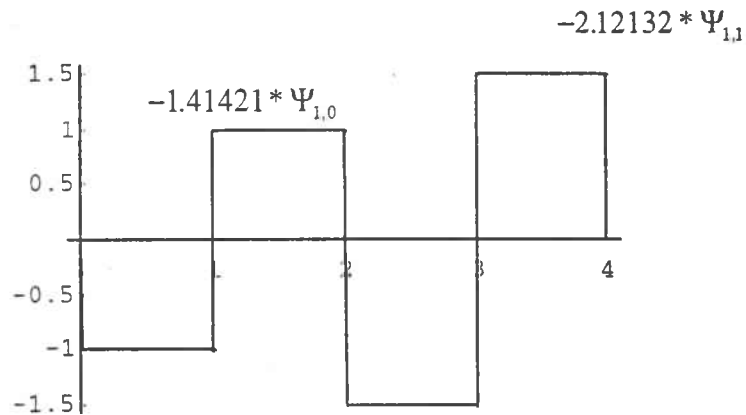


(For the Haar wavelets to be applied to the function, the number of steps must be 2^n .)

All coefficients corresponding to a particular value of m may be regarded as belonging to the same "generation." To evaluate one generation of coefficients, a temporary step function must first be calculated. This temporary function is just the average of the two function values over the entire range of both.

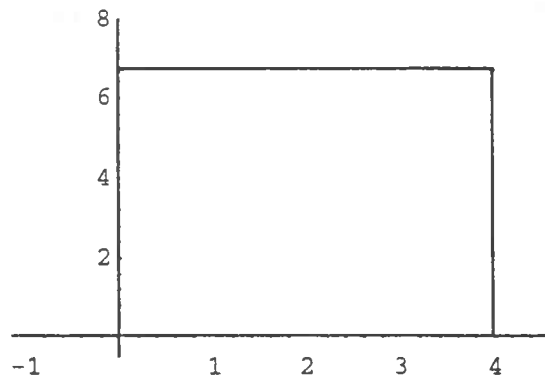


The coefficients of the linear combination are found starting with the $m = 1$ generation, which encompasses the narrowest wavelets that will be used. The first set is found by subtracting the temporary average from every other function value and multiplying by $2^{m/2}$. In this example there will be two coefficients for this generation with $m = 1$. The first coefficient is found by the process beginning $2 - 3 = -1$ and $-1 * 2^{1/2} = -1.41421$. The second coefficient is $9 - 10.5 = -1.5$ and $-1.5 * 2^{1/2} = -2.12132$. These two coefficients form two cycles of Haar wavelets which are plotted with respect to the range of each previous value.

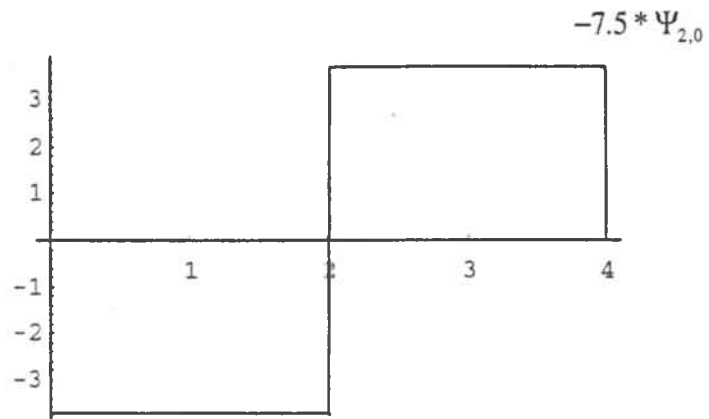


This looks exactly like two of the Haar wavelets seen earlier.

The next step would be to break down the temporary function of averages, and repeat this process until only one average value is left. Therefore, breaking down the function of averages as above, we obtain



The m value for this generation is 2 and the coefficient for this generation is computed from $3 - 6.75 = -3.75$ and $-3.75 * 2^{3/2} = -7.5$. This is the second family of coefficients and this coefficient forms one cycle of Haar wavelets, as plotted below with respect to the range of each previous value.



Once again the next step is to break down the temporary average function, but since the average function now has already only one value (here 6.75), our original function can be written as the sum $f(x) = 6.75 - 7.5 * \Psi_{2,0} - 1.41421 * \Psi_{1,0} - 2.12132 * \Psi_{1,1}$.

Mexican Hat Function

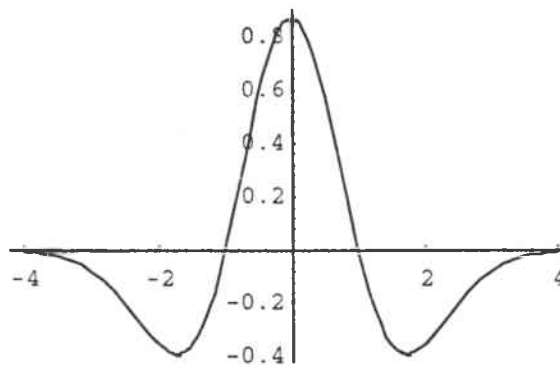
Definition

Even though the Haar wavelet is a well-localized and simple "mother wavelet," it does have its drawbacks. Since the Haar wavelet is a step function, it has discontinuities and is not continuously differentiable. Secondly, the Haar wavelet is a square wave. Therefore, when using it to represent a smooth function, many wavelets have to be used and still it only represents a function as "choppy."

One basis function that is found quite often is the second derivative of the Gaussian distribution, which is also known as the Mexican Hat function.

$$\Psi(x) = \frac{2}{\sqrt{3}} \pi^{-1/4} (1 - x^2) e^{-x^2/2} \quad (\text{Daubechies 75}).$$

A plot of this Mexican hat "mother wavelet" is shown below:



If one imagines rotating this graph around its symmetry axis, then one obtains a shape similar to a Mexican Hat. Since it is a continuously differentiable smooth function, it does a better job at representing a smooth function than the Haar wavelet.

Earlier I had stated how important orthogonality was. If one could develop a set of functions that are orthogonal, it permits a simple representation of the complicated

original function as a linear combination of these orthogonal basis functions. The problem with the Mexican Hat wavelet is that with certain values of a_0 and b_0 the different wavelets are not orthogonal. Because of this fact three mathematicians used the idea of "frames" to develop a linear combination that uses different "voices" of wavelets.

Frames

I have shown earlier in eq(7) that a function can be represented as an infinite sum of basis functions, but this holds true only when the basis functions are orthogonal.

Functions can be characterized by means of their wavelet coefficients $\langle f, \Psi_{m,n} \rangle$ if it is true that " $\langle f_1, \Psi_{m,n} \rangle = \langle f_2, \Psi_{m,n} \rangle$ for all m, n in Z " implies that " $f_1 \equiv f_2$ ". If the inner products in the sequence are close to each other in value, then in order for an algorithm such as eq(7) to be able to reconstruct f in a numerically stable way, f_1 and f_2 must be "close" also. This leads to the definition of a frame:

A family of functions $(\varphi_j)_{j \in J}$ in a Hilbert Space, H , is called a frame if

there exists $A > 0, B < \infty$ so that, for all f in H

$$A\|f\|^2 \leq \sum_{j \in J} |\langle f, \varphi_j \rangle|^2 \leq B\|f\|^2$$

where A and B are called the frame bounds (Daubechies 56).

If the two frame bounds are equal ($A = B$), the frame can then be characterized as a "tight frame." A tight frame is significant in the same way orthogonality was important. A tight frame allows us to simplify an original function by expressing it as a linear combination of functions φ_j (Daubechies 56).

With $A = B$,

$$\sum_{j \in J} |\langle f, \varphi_j \rangle|^2 = A \|f\|^2$$

which leads to

$$f = A^{-1} \sum_j \langle f, \varphi_j \rangle \varphi_j. \quad \text{eq(10)}$$

Eq(10) is almost identical to eq(7) except for the A^{-1} . The constant is needed to solve the problem that the φ_j are not orthonormal. If indeed φ_j are a tight frame with frame bound $A = 1$, and if $\|\varphi_j\| = 1$ for all $j \in J$, then φ_j constitute an orthonormal basis (Daubechies 57).

The Mexican Hat function leads to a frame with B/A close to 1 for $a_o \leq 2^{1/4}$, and consequently I used this wavelet to write a program capable of breaking down a function into Mexican Hat wavelets with $a_o = 2^{1/4}$. The problem with this function was that in order to include all Mexican Hat wavelets that significantly overlap the function, I needed to use a very large number of generations. For example in going from $m = 1$ to $m = 2$, the prefactor only increases by a factor of $2^{1/8}$. In addition, I must analyze the mother wavelet Ψ at smaller translations and smaller changes in dilations. Therefore I must take account of a large number of generations to approximate the function accurately.

Voices

A. Grossman, R. Kronland-Martinet, and J. Morlet discovered a method that remedies this problem by using "different voices per octave" (Daubechies 71). This amounts to using several different wavelets Ψ^1, \dots, Ψ^N for each set of m, n values: $\{\Psi_{m,n}^v\}$;

$m, n \in Z, v = 1, \dots, N$. What Grossman *et al.* did was to take "fractionally" dilated versions of a single wavelet,

$$\Psi^v(x) = 2^{-(v-1)/N} \Psi(2^{-(v-1)/N} x).$$

Tables 1 through 4 give the frame bounds for the Mexican Hat function with $a_0 = 2$ and various b_0 for different numbers of voices, N (Daubechies 77).

N=1			
b	A	B	B/A
0.25	13.091	14.183	1.083
0.50	6.546	7.092	1.083
0.75	4.364	4.728	1.083
1.00	3.223	3.596	1.116
1.25	2.001	3.454	1.726
1.50	0.325	4.221	12.986

Table 1

N=2			
b	A	B	B/A
0.25	27.203	27.278	1.002
0.50	13.673	13.639	1.002
0.75	9.091	9.093	1.002
1.00	6.768	6.870	1.015
1.25	4.834	6.077	1.257
1.50	2.609	6.483	2.485

Table 2

N=3			
b	A	B	B/A
0.25	40.914	40.914	1.000
0.50	20.457	20.457	1.000
0.75	13.638	13.638	1.000
1.00	10.178	10.279	1.010
1.25	7.530	8.835	1.173
1.50	4.629	9.009	1.947

Table 3

N=4			
b	A	B	B/A
0.25	54.552	54.552	1.000
0.50	27.276	27.276	1.000
0.75	18.184	18.184	1.000
1.00	13.586	13.690	1.007
1.25	10.205	11.616	1.138
1.50	6.594	11.590	1.758

Table 4

As soon as we take two or more voices, the frame may be considered tight for all $b_0 \leq .75$.

Therefore the function f can be represented as

$$f = A^{-1} \sum_{m,n,v} \langle f, \Psi_{m,n}^v \rangle \Psi_{m,n}^v \quad \text{eq(11)}$$

with, for example,

$m, n \in Z; \quad v = 1, 2; \quad A = 9.091;$

and

$$\Psi_{m,n}^v(x) = 2^{-\left(\frac{v+m-1}{2}\right)} \Psi\left(2^{-\left(\frac{m+v-1}{2}\right)} x - .75n\right). \quad \text{eq(12)}$$

This choice, drawn arbitrarily from Table 2, uses $b_o = .75$, $a_o = 2$, and $N = 2$. I then wrote a program in Mathematica using eq(11) and eq(12) that analyzed a function under these conditions.

Analysis

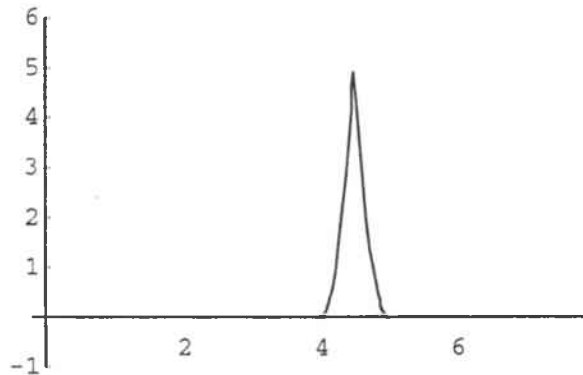
To illustrate how wavelets work, I have analyzed a well-localized function using the Fourier series and the Mexican Hat wavelets. For Fourier analysis, I simply loaded the Fouriertransform package in Mathematica and used the command "FourierTrigSeries". For wavelet analysis, I used the programs I designed in Mathematica that break down functions as sums of the Mexican Hat wavelet basis and then rebuild the original function from the sum of wavelets.

Example 1 Fourier Series and the Mexican Hat Wavelet

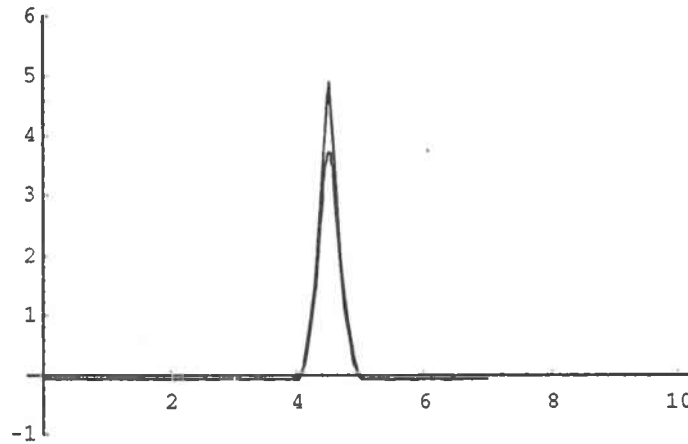
The function I chose to decompose is defined on the interval from 0 to 10 by

$$f(x) = \begin{cases} 20 * (x - 4)^2 & 4 \leq x < 4.5 \\ 20 * (x - 5)^2 & 4.5 \leq x \leq 5. \\ 0 & \textit{elsewhere} \end{cases}$$

Its graph is



Below is a plot comparing the original function to the Mexican Hat representation of the original function.

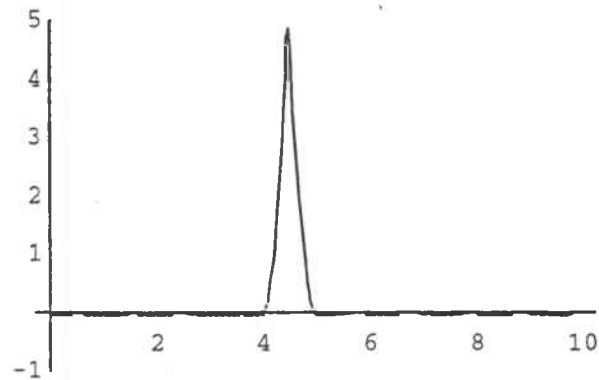


As you can see the Mexican Hat function faithfully reproduces almost all values of the original function. This decomposition used eight different generations of Mexican Hat wavelets (from $m = -4$ to $m = 3$). The reason I stopped at -4 is that the more negative m becomes, the smaller the translations become. For example if $m = -4$, then each time n is increased by 1, the Mexican Hat wavelet is translated by $2^{m/2} \cdot .75n$ or $.1875$. One can see that the more negative m is, the smaller the translations become. The result of small translations is an increase in the number of coefficients one must compute for a given m . The reason I took $m \leq 3$ is that values of m larger than 3 were insignificant to the wavelet series (i.e., the overlap of those wavelets with the original function was negligible).

Every calculated coefficient that was less than $.01$ I simply set equal to 0 . Coefficients of value less than $.01$ would have little contribution to the representation of the original function. Both the limited range of m values used and the discarding of small coefficients could account for why the two graphs are not identical.

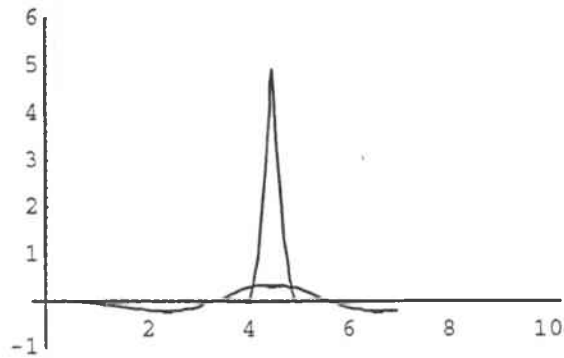
To make a comparison with what was obtained by the Mexican Hat, I also analyzed the original function by breaking it down into the Fourier series. In doing the

Fourier series analysis I calculated 200 coefficients (the same number as used for the Mexican Hat analysis). Below is a plot of the original function and its Fourier series:

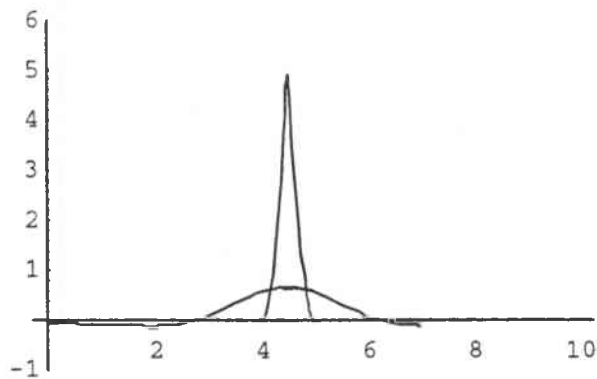


As you can see the Fourier series, for the most part, does a great job representing the original function. Over the interval where the function is non-negative it is practically impossible to tell the difference between the two. But if you look closely at the parts of the graph outside the range from 4 to 5, you can see little wiggles. The Fourier series, unlike the Mexican Hat wavelet, seems to have a difficult time representing the function at the points where the function is identically zero.

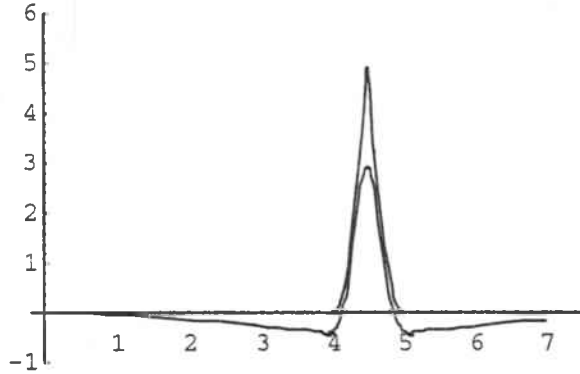
Since it required around 200 coefficients to rebuild the original function accurately using the Mexican Hat wavelets, I then tried to reduce the number of coefficients, yet still retain a close representation of the original function. To start out I arbitrarily chose only one generation, where the value of m is zero. For $m = 0$ there are only 23 non-negligible coefficients, but the plot is not representative of the original function. Below is a plot of the two.



I then decided to use $m = 0$ and the positive values of m that I had used earlier. This accounted for four generations ($m = 0, 1, 2, 3$). For this example there are 63 coefficients, but again this Mexican Hat series did not compare well with the original function. Below is a graph comparing them.

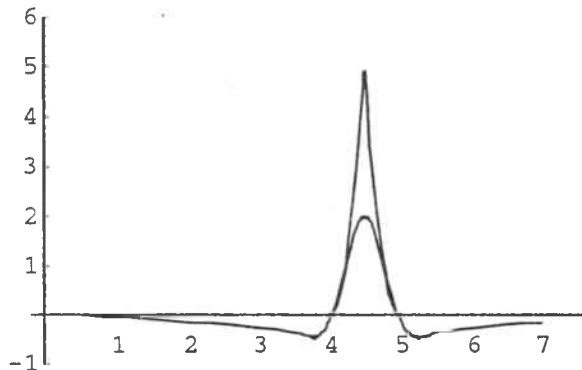


After observing that the generation for $m = -4$ contributes slightly, I decided to analyze the original function over four generations from $m = -3$ to $m = 0$. Even though differences could be seen, this resulted in a very close representation of the original function, as shown below:



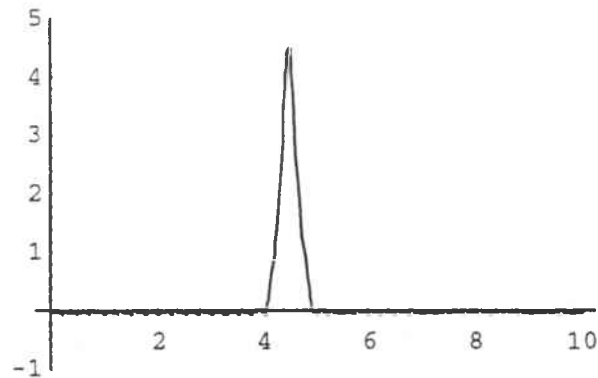
As you can see, the graphs are much closer than when using only the positive values of m . The more negative values of m appear to have a greater significance in the Mexican Hat wavelet series than the positive values did, because the wavelets with negative m values have narrower and sharper peaks, which are nearer in form to the original function. The only problem with this is that the series with negative values of m still required about 100 coefficients.

Trying many options for values of m revealed that the range with the lowest number of coefficients, but still representing the original function as closely as possible, was for values of $m = -2$ to $m = 0$. These 3 generations required around 70 coefficients and the graph comparing the original function and the Mexican Hat representation is seen below.



As you can see the Mexican Hat representation does not peak at even half the value of the original function and the representation also is negative where the original function was zero. However I felt that the Mexican Hat series does a decent job recreating the original function.

Since this is the best representation I could come up with, I once again decided to compare this fewer-coefficient Mexican Hat series to the Fourier series taken to 70 coefficients.



Even with only 70 coefficients, the Fourier series still does a better job than the Mexican Hat wavelets. It is still difficult to distinguish it from the original function. Once again the Fourier series is almost an exact representation, but again there are wiggles where the value of the function should be zero.

Conclusion

I have shown in my analysis that an original function can be represented as a sum of Mexican Hat wavelets. This is important since all that is required to reconstruct complex phenomena is the calculated coefficients of the series. Since the Mexican Hat wavelets do the same job as the Fourier series, an importance is placed upon the number of coefficients each series requires. It is optimal to minimize the number of coefficients to decrease time, for calculation purposes, and space, for storage purposes. Overall I have come to two conclusions.

First, Mexican Hat wavelets differ from Haar wavelets in that they are not orthonormal. To be able to use the Mexican Hat wavelet it is necessary that it consist of a "tight frame." A "tight frame" allows the function to be represented as a linear combination of infinitely many basis functions just as was the case for orthogonal functions. The fact that the wavelets did represent a "tight frame" for values of m less than $2^{1/4}$, was found to be not efficient. Using this value required the use of many generations of wavelets which in turn meant more coefficients and time. To solve this problem it was also necessary to take different "voices" of the wavelets.

Second, I discovered much to my disappointment, that the Mexican Hat wavelets did not do a better job representing a function with a minimal number of coefficients when compared to the Fourier series. In my analysis I compared a Mexican Hat wavelet series that accurately represented a function and one that well represented a function using a minimal amount of coefficients to the Fourier series with the same number of coefficients. After comparing the two, I found that in both cases the Fourier series did a better job than

the Mexican Hat wavelets. It was difficult to distinguish the Fourier series from the original function, but the Mexican Hat series had visible discrepancies.

The Mexican Hat wavelet is just one of the different "mother wavelets" that are available. Since the discovery of the Mexican Hat wavelet, there have been developed many more that do a better job of representing complex phenomena. Wavelets were only discovered in the 1980's and they are still fairly new and are continually being improved. As the example of the FBI fingerprints illustrates, people are finding a variety of uses for them.

Appendix

Haar Function - BREAK

This Program breaks a function into Haar Wavelets

```
Clear[wave,avg]
piece[start_,end_,n_]:=Module[{},
```

(*start and end is the range of transform, n is the number of steps*)

```
    step=(end-start)/n;
```

(*equat1 is function to be transformed*)

```
    equat1[a_]=-a^3+6*a^2+2*a;
    Plot[equat1[z],{z,start,end}];
```

(*b is the Table of values of the piece-wise constant function over*)
(*the range. They are found by integrating the function over each step*)
(*of the range*)

```
    b=Table[Integrate[equat1[c],{c,i,i+step}]/step//N,
            {i,start,end,step}];
```

(*equat2 is the constructed piece-wise constant function*)
(*It has been created by multiplying a ceiling function from 0 to 16*)
(*by its respective integrated value found over that range*)

```
    equat2[aa_]:=If[start<aa<end,b[[Ceiling[(aa-start)/step]]],0];
    Plot[equat2[z],{z,start,end}];
```

(*TRANSFORMING THE STEP FUNCTION INTO HAAR WAVELETS*)

(*m are the values of m in the discrete wavelet equation*)
(*and marray is developed to store the values*)

```
    Clear[m,nvalues,marray,narray,nnns];
    marray={ };
    narray=Array[nnns,Log[2,n]];
    loop=1;
```

(*This loop alternates through the different generations starting*)
(*from the highest to the lowest*)

```
While[loop<=(Log[2,n]),
```

(*m is calculated here according to the width of one complete evolution*)
(*of a Haar wavelet. As seen in this example, the width of one*)
(*will be 2*Range/steps. M will then be ln(width)/ln(2)*)
(*nvalues stores the values for n over for a generation. Each*)
(*n is equal to the values where the wavelet begins divided by 2^m*)

```
m=N[Log[(end-start)/2.^((Log[2,n])-loop)]/Log[2.]];
marray=Append[marray,m];
```

```
nvalues=Table[(start+i*2^m)/(2^m),{i,0,
(2.^((Log[2,n])-loop)-1)}];
nnns[loop]=nvalues;
```

(*wave here creates the table of coefficients for the Haar Wavelet*)
(*What it does is the average difference between 2 consecutive*)
(*values and multiply it by 2^(m/2)*)

```
wave1=Table[1/2*2^(m/2)*(b[[iii]]-b[[iii+1]]),
{iii,1,n/(2^(loop-1)),2}];
Print[wave1];
wave=Table[-alt*1/2*2^(m/2)*(b[[iii]]-b[[iii+1]]),
{iii,1,n/(2^(loop-1)),2},{alt,-1,1,2}];
wave=Flatten[wave];
```

(*equat3 is the step function of the Haar wavelets. As you can see*)
(*it is like the Ceiling function above but must be multiplied by *)
(*2^(-m/2) to get back the original function values*)

```
equat3[v_]:=If[start<v<end,
wave[[Ceiling[(v-start)/(step*2^(loop-1))]]], 0]*2^(-m/2);
Plot[equat3[z],{z,start,end}];
```

(*avg is the table of new values to be transformed into Haar Wavelets*)
(*Alternating each step, they are the average of the two consecutive*)
(*values*)

```
avg=Table[(b[[j]]+b[[j+1]])/2,{j,1,n/(2^(loop-1)),2}];
b=avg;
loop++;
equat4[vv_]:=If[start<vv<end,
avg[[Ceiling[(vv-start)/(step*2^loop)]]],0];
Plot[equat4[z],{z,start,end}];;
Print[CONSTANT];
```

```
Print[avg];  
Print[marray];  
Print[narray];
```

]

Mexican Hat Wavelet - BREAK

THIS PROGRAM USES THE MEXICAN HAT WAVELET (DERIVATIVE OF THE GAUSSIAN CURVE) TO ANALYZE A FUNCTION

This program analyzes a function using the discrete formula for wavelets $[a^{(m/2)}\Psi(a^{(-m)}x-.75*n)]$ with $a = 2$ and $b = .75$. Then for significant values of m and n , it uses voices to allow the representation of the function as a sum of wavelets

```
mexhat[step_,beg_,end_,mbeg_,mend_] := Module[{ },

(*step is the number of generations of mexican hat wavelets*)
  (*i.e. different values of m*)
(*beg is the beginning of where the function is analyzed*)
(*end is the end of where the function is analyzed*)
(*mbeg is the beginning m value*)
(*mend is the ending m value*)

  Clear[coeff,row,row2,totcoeff,totcoeff2,func,psiabv,coeffarray,
        mandnarray,new,n,m];

  pi=N[Pi];
  pref=2./N[Sqrt[3]]*pi^(-.25);
  func[xx_]:=0 /; -20 < xx < 4;
  func[xx_]:=20*(xx-4)^2 /; 4 <= xx < 4.5;
  func[xx_]:=20*(xx-5)^2 /; 4.5 <=xx <= 5;
  func[xx_]:= 0/; 5 < xx < 20;
  psiabv[a_]:=2.^-((v+m-1)/2)*psi[2.^-(m+((v-1)/2))*a-n*.75];
  psi[b_]:=pref*(1-b^2)*Exp[(-b^2/2)];

(*func is the function to be analyzed*)
(*psiabv is the definition of the wavelet with different voices v*)
(*psi is the Mother Wavelet (here the Mexican Hat Function)*)

  totcoeff=Array[row,{step,1}];
  totcoeff2=Array[row2,{step,1}];

(*These matrices store the values of the coefficients in totcoeff*)
(*and their corresponding values of m and n in totcoeff2*)
(*They will form step x 1 matrices*)

(*counter is used for formatting purposes*)

  counter=1;
```

```
m=mbeg;
```

(*This first loop increments the value of m over the range of m values*)

```
While[m <= mend,
```

(*For this value of m, n is set to the value where the mexican hat*)

(*function is just entering the range analyzed under the second voice*)

(*v = -3 is just an approximation to shift it to the left*)

```
n=Floor[beg/(.75*2^(m+.5))]-3;
```

(*coeffarray and mandnarray are temporary arrays that will store the*)

(*values of the coefficients & m and n values for incrementing*)

(*values of m*)

```
coeffarray={};
```

```
mandnarray={};
```

(*The second loop increments n from the value described above to the*)

(*value where the mexican hat function is just leaving the range to be*)

(*analyzed for the first voice. Again +4 is just an approximation*)

(*to shift it to the right*)

```
While[n<=Ceiling[end/(.75*2^m)]+4,
```

```
  vloop=1;
```

(*The third loop increments voices over the values of 1 & 2*)

```
  While[vloop<=2,
```

(*At this point the coefficients are calculated. A function called*)

(*new is the product of the wavelet psiabv and the analyzed function.*)

(*This new function is then integrated over the analyzed range from*)

(*beg to end. If the coefficient is less than .001 it is set to 0.*)

```
    v=vloop;
```

```
    Plot[psiabv[x],{x, 4, 6},PlotRange -> {-1,2}];
```

```
    new[g_]=func[g]*psiabv[g];
```

```
    Clear[coefficient];
```

```
    coefficient=NIntegrate[new[h],{h,beg,end}];
```

```
    If[Abs[coefficient] < .001,
```

```
      coefficient=0];
```

(*Simply adds each coefficient calculated to the end of coeffarray*)

(*each time a new value is calculated*)

```
coeffarray=Append[coeffarray,coefficient];  
vloop++;];  
mandnarray=Append[mandnarray,{m,n}];  
n++];
```

(*These parts of the program are for formatting purposes*)

```
row[counter,1]=coeffarray;  
row2[counter,1]=Flatten[mandnarray];  
totcoeff[[counter]]=Flatten[row[counter,1]];  
totcoeff2[[counter]]=Flatten[row2[counter,1]];  
counter++;  
m++;  
Print[totcoeff];  
Print[totcoeff2];
```

```
]
```

Mexican Hat Wavelet - BUILD

THIS PROGRAM REBUILDS A FUNCTION FROM COEFFICIENTS
AND VALUES OF M AND N

```
mexbuild[num_,coeffs_,mandns_]:=Module[{},
```

```
(*num is the number of generations*)
```

```
(*coeffs is the matrix containing the coefficients*)
```

```
(*mandns is the matrix containing the m and n values*)
```

```
(*example *)
```

```
Clear[newfunc,total,mexbuild,poodle,m,n,v,end];
```

```
pi=N[Pi];
```

```
pref=2/N[Sqrt[3]]*pi^(-.25);
```

```
psiabv[p_]:=N[2^-((v+m-1)/2)]*psi[N[2^-((m+((v-1)/2))]]*p-n*.75];
```

```
psi[q_]:=pref*(1-q^2)*Exp[(-q^2)/2];
```

```
(*psiabv is the wavelet over different voices v*)
```

```
(*psi is the Mother Wavelet (here the Mexican Hat wavelet)*)
```

```
end=0;
```

```
counter1=1;
```

```
(*This part of the program calculates the value for the number of*)
```

```
(*iterations that must be carried out when multiplying psiabv*)
```

```
(*by its corresponding coefficients. It is simply the sum of the*)
```

```
(*length of each row in the inputted coefficient matrix*)
```

```
While[counter1<=num,
```

```
end=end+Length[coeffs[[counter1]]];
```

```
counter1++];
```

```
(*doggy is an array of length equal to the number of coefficients.*)
```

```
(*It will store each product of the coefficient and its corresponding*)
```

```
(*function of psiabv*)
```

```
doggy=Array[poodle,{end,1}];
```

```
dogs=1;
```

```
loop3=1;
```

```
(*This loop increments through the different generations*)
```

```
While[loop3 <= num,
```

```
position2=1;
```

(*This sets m to the first value in each row of the mandns matrix*)

```
m=mandns[[loop3,1]];
loop4=1;
```

(*This loop increments n through values for each generation*)

```
While[loop4 <= (Length[coeffics[[loop3]]]/2),
      n=mandns[[loop3,loop4*2]];
      loop5=1;
```

(*this loop increments voices over values of 1 & 2*)

```
While[loop5 <= 2,
```

(*Here is the main part of the program. All it does is multiply*)
(*each function of psiabv at inputted values of m, n and v by the*)
(*corresponding coefficients calculated at the same values of*)
(*m, n and v. It then stores each product in the doggy array*)

```
      v=loop5;
      poodle[dogs,1] = coeffics[[loop3,position2]]*psiabv[r];
      position2++;
      loop5++;
      dogs++;
      loop4++;
      loop3++;
```

(*At this point it is taking all the terms in doggy array, summing*)
(*them together, and dividing by 9.090 (the value of A given in the table)*)

```
total[r_]=Sum[doggy[[i,1]],{i,1,end}]/9.091;
pr2=Plot[total[t],{t,0,7},PlotRange->{-2,10}];
```

```
]
```

Works Cited

- Brislawn, Christopher M. "Fingerprints Go Digital." Notices Nov. 1995: 1278-1282.
- Daubechies, Ingrid. Ten Lectures on Wavelets. Montpelier, VT: Capital City Press, 1992.
- Graps, Amara. An Introduction to Wavelets.
<http://www.amara.com/IEEEwave/IEEEwavelet.html>
- Haberman, Richard. Elementary Applied Partial Differential Equations. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1987.
- Muller, Peter and Vidakovic, Brani. Wavelets For Kids.
<ftp://ftp.isds.duke/pub/Users/brani/papers/wav4kidsA.ps.Z>