

(Notes Supplied by Dr Frank Stootman. UWS, 2001)

The Wavelet Transform

1.0 Basic Ideas

- The Discrete Fourier Transform (DFT) may be thought of in general terms as a matrix multiplication in which the original vector x_k is decomposed into a series of coefficients X_n . Both k and n are integers which range over the same value N .

$$\begin{pmatrix} x_0 \\ \cdot \\ \cdot \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & W_{kn} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} X_0 \\ \cdot \\ \cdot \\ X_{N-1} \end{pmatrix}$$

In the above we may derive the transformed coefficients X_n by inverting the matrix.

- The form of W_{kn} has many possibilities but physically we would like the option of forward and backward transforms ie., an inverse ought to exist.
- The Discrete Wavelet Transform (DWT) generates a matrix W_{kn} which is now widely used for image compression instead of the FT since it is able to localise preserve photographic detail such that many of the coefficients may be ignored (tantamount to filtering) and yet the reconstruction remains effective. For certain types of problems the filtering may be much more aggressive than corresponding FT coefficient filtering. (see “Numerical Recipes in C”, Prentice Hall, 2nd Ed. 1992, chapter 13).
- DWT’s are particularly effective in analysing waveforms which have spikes or pulses buried in noise. The noise may be more effectively removed than with FT filtering and the shape of the pulses preserved. Conservation of Energy similar to a Parseval theorem would also be nice.

2.0 The Haar Transform to get that Wavelet feel

- Suppose for simplicity we assume an input vector x_k with $0 < k < 7$. This is readily decomposed into an obvious basis set as shown.

$$(x_k) = x_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \dots + x_7 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- Other basis systems are of course possible (remember your QM and spinors?). In 1910 Haar proposed the following decomposition.

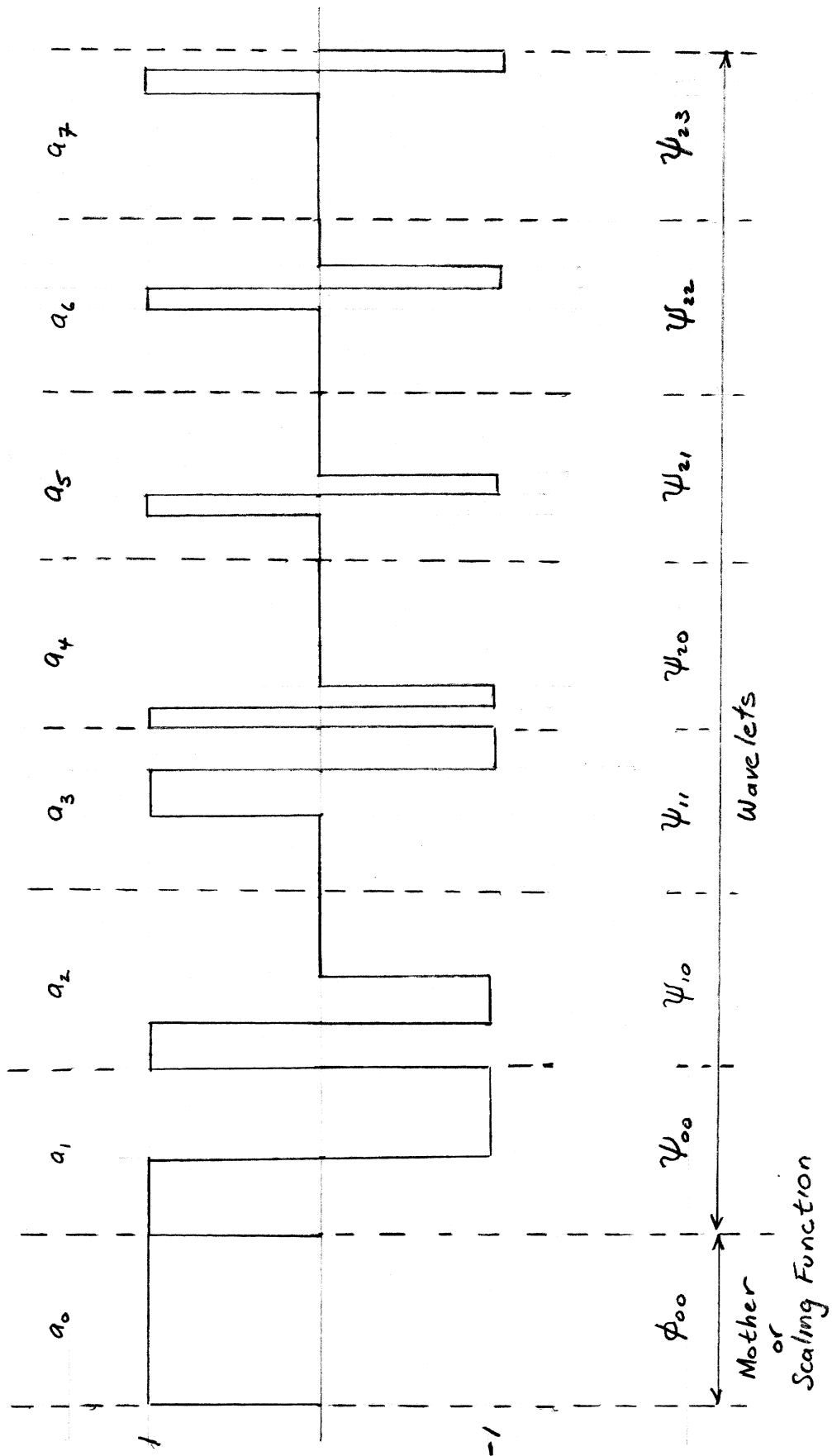
$$(x_k) = a_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + a_1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{pmatrix} + a_2 \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + a_3 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} + a_4 \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + a_5 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + a_6 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} + a_7 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

or $x_n = H_{nk} a_k$ with the columns of H being simply the above basis vectors and the a_k obtained by matrix inversion of H.

- These basis vectors have characteristic “shapes” when drawn on their side as shown in the figure on the next page and it is these shapes which show the essential features of what DWT decomposition does.

Notice:

- 1) A mother or scaling function at the start with a non-zero average. This will normally be normalised to 1.
- 2) Wavelet functions with zero average which are both compressed and translated. It is this compression and translation which finds peaks or pulses well.
- 3) The wavelet functions are orthogonal. You can see this directly by multiplying any two together.
- 4) The wavelet functions have compact support which means they are all localised. This is unlike the FT in which the basis functions $\exp(-2\pi nk / N)$ are continuous.



- How do we use other shapes and make a wavelet basis system out of them?

step 1: Mother functions

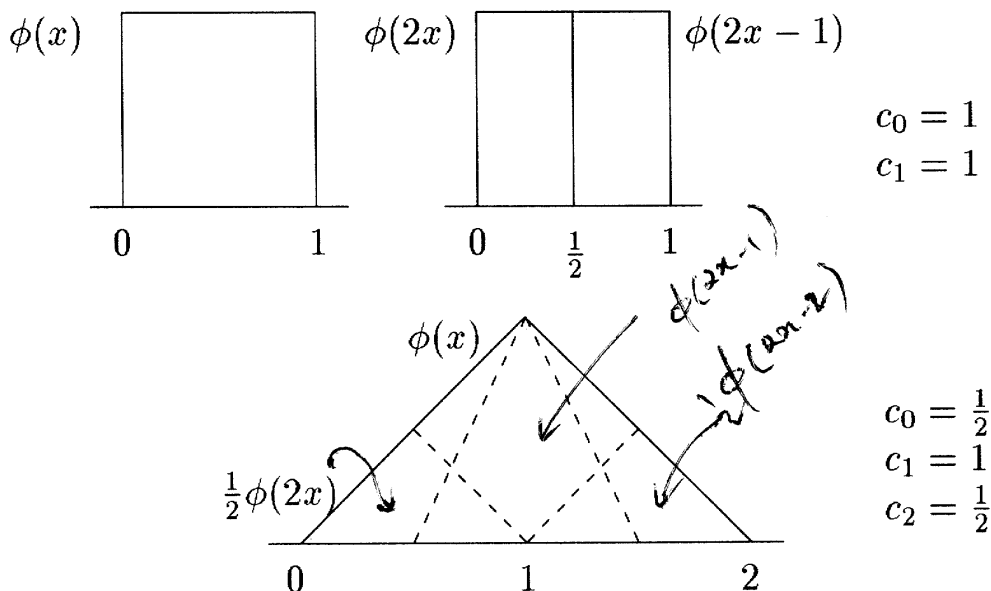
Let $\phi(x)$ be some mother function. The $\phi(2x)$ is the same function compressed by a factor of 2. Binary compression can therefore be denoted as $\phi_j = \phi(2^j x)$. Likewise $\phi(2x-1)$ is our compressed function translated by 1. Multiple translation and compression of the mother function can therefore be denoted as $\phi_{jk} = \phi(2^j x - k)$

We do **not** choose $\phi(x)$ arbitrarily but impose two conditions.

1) $\phi(x) = \sum_k c_k \phi(2x - k)$ or more generally $\phi(2^{j-1}x) = \sum_k c_k^j \phi(2^j x - k)$. That is it lends itself to a fractal like summing behaviour.

2) $\int \phi(x) dx = 1$, the normalisation condition. This leads to $\sum_k c_k = 2$. Alas, life is not easy and there is much confusion in the literature at this point. If you accept this as is then you will **NOT** get coefficients which produce a reversible transform. Since this is desirable in physics we need to do what Numerical Recipes suggests and force $\sum_k c_k^2 = 1$. This means reducing the coefficients by a further factor $1/\sqrt{2}$. The reason lies buried deep in matrix inversion.

- Here are two examples, our friend Haar and the “top hat”

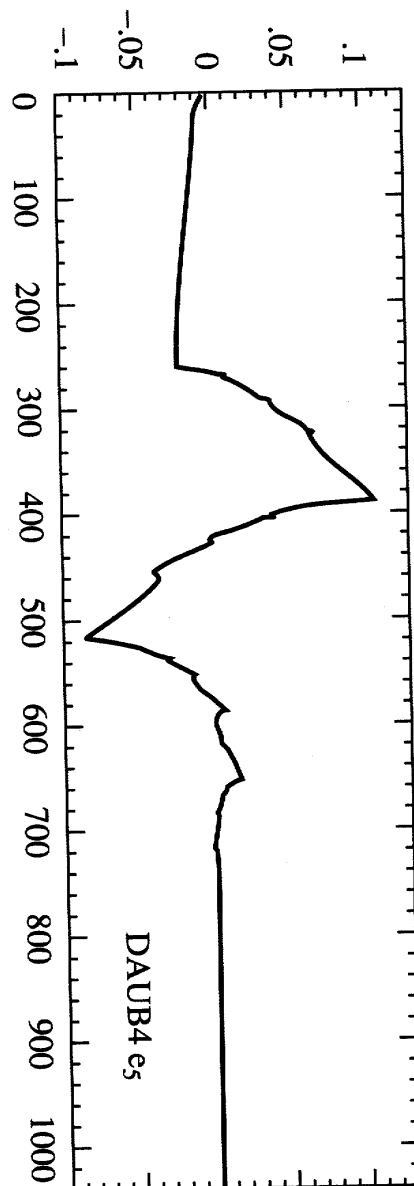


- Again a $1/\sqrt{2}$ multiplication factor ensures reversibility of the transform

- Ingrid Daubechies invented a four coefficient fractal which is not a simple mother function shape as above, but instead must be constructed by working backwards from the coefficients. They are:

$$c_0 = \frac{1}{4}(1 + \sqrt{3}), c_1 = \frac{1}{4}(3 + \sqrt{3}), c_2 = \frac{1}{4}(3 - \sqrt{3}), c_3 = \frac{1}{4}(1 - \sqrt{3})$$

- Again “Numerical Recipes” surreptitiously adds a further $1/\sqrt{2}$ and with good reason!



Constructed Mother function from the Daubechies coefficients. This function is not differentiable. see Daubechies I., Comm. on Pure and Applied Mathematics, **41**, 909, 1988

step2: Wavelet functions

- From the mother or scaling function and the coefficients we construct wavelet functions $\psi(x)$.

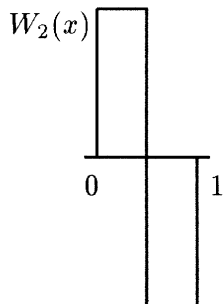
$$\psi(x) = \sum_k (-1)^k c_{M-k} \phi(2x - k) \text{ or at other compression levels}$$

$$\psi(2^{j-1}x) = \sum_k (-1)^k c_{M-k} \phi(2^j x - k) \text{ with generally}$$

$$\psi_{jk} = \sum_k (-1)^k c_{M-k} \phi(2^j x - k)$$

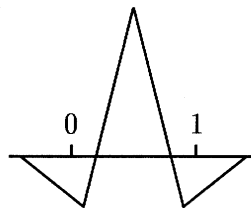
Three things to note:

- The introduction of an alternating negative sign on the coefficients.
 - The inversion of the order of coefficients assuming there are M coefficients.
 - If you have multiplied by the requisite fudge factor to get a reversible transform you don't need to do any more on these coefficients.
- Here are the basic wavelet shapes at the highest level. The Daubechies wavelet is a construction.



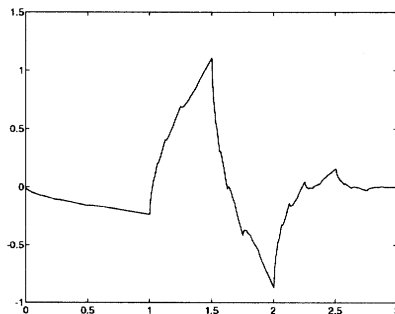
Haar wavelet from box function

$$W_2(x) = \phi(2x) - \phi(2x - 1)$$



“Wavelet” from hat function

$$W = \phi(2x) - \frac{1}{2}\phi(2x - 1) - \frac{1}{2}\phi(2x + 1)$$



$W_4(x)$ from $\phi = D_4$ Orthogonal wavelet

Please note the wavelet function for the top hat is strictly

$\psi(x) = \frac{1}{2}\phi(2x) - \phi(2x - 1) + \frac{1}{2}\phi(2x - 2)$ and inverted to the above diagram. The areas sum to zero as does the sum of the coefficients.

step3: Multi Resolution Analysis (MRA)

- Although we have quite general definitions for ϕ_{jk} and ψ_{jk} we need only use the $j=0$ level over and over again. This was a discovery by Mallat.

Here is the technique:

- 1) Multiply each a **pair** of input coefficients with the mother function coefficients on the top line and the wavelet coefficients in the bottom line.

eg For the non-reversible Haar transform this is

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \circ \begin{pmatrix} 3 \\ 2 \\ 4 \\ 1 \\ 4 \\ 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ 5 \\ 3 \\ 6 \\ 4 \\ 4 \\ 2 \end{pmatrix} \quad \begin{pmatrix} m \\ w \\ m \\ w \\ m \\ w \\ m \\ w \end{pmatrix}$$

- 2) Now sort (an effective permutation) the above column matrix and bring all the mother generated coefficients to the top.

$$\begin{pmatrix} 5 \\ 1 \\ 5 \\ 3 \\ 6 \\ 4 \\ 4 \\ 2 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 5 \\ 5 \\ 6 \\ 4 \\ 1 \\ 3 \\ 4 \\ 2 \end{pmatrix} \quad \begin{pmatrix} m \\ m \\ m \\ m \\ w \\ w \\ w \\ w \end{pmatrix}$$

3) Now repeat step 2 only on the coefficients labelled 'm'

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \circ \begin{pmatrix} 5 \\ 5 \\ 6 \\ 4 \\ 1 \\ 3 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \\ 10 \\ 2 \\ 1 \\ 3 \\ 4 \\ 2 \end{pmatrix}$$

4) repeat step 2) and 3) until only the top coefficient has the 'm' label.

- The complete sequence looks like:

$$\begin{pmatrix} 3 \\ 2 \\ 4 \\ 1 \\ 4 \\ 2 \\ 3 \\ 1 \end{pmatrix} \xrightarrow{T} \begin{pmatrix} 5 \\ 1 \\ 5 \\ 3 \\ 6 \\ 2 \\ 4 \\ 2 \end{pmatrix} \xrightarrow{P} \begin{pmatrix} 5 \\ 5 \\ 6 \\ 4 \\ 1 \\ 3 \\ 2 \\ 2 \end{pmatrix} \xrightarrow{T} \begin{pmatrix} 10 \\ 0 \\ 10 \\ 2 \\ 1 \\ 3 \\ 2 \\ 2 \end{pmatrix} \xrightarrow{P} \begin{pmatrix} 10 \\ 10 \\ 0 \\ 2 \\ 1 \\ 3 \\ 2 \\ 2 \end{pmatrix} \xrightarrow{T} \begin{pmatrix} 20 \\ 0 \\ 0 \\ 2 \\ 1 \\ 3 \\ 2 \\ 2 \end{pmatrix}$$

- Reversing the above procedure is used to compose the original vector. In this case the multiplying matrix has to be.

$$\begin{pmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix}$$

which is the inverse of the original coefficient matrix. But this is messy and could be fixed with a universal $1/\sqrt{2}$ to retain the symmetry of the mathematics and is why "Numerical Recipes" adds the factor to the coefficients.

$$\begin{bmatrix} c_0 & c_3 & & \dots & & & c_2 & c_1 \\ c_1 & -c_2 & & \dots & & & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & & & & \\ c_3 & -c_0 & c_1 & -c_2 & & & & \\ & & & \ddots & & & & \\ & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & c_3 & -c_0 & c_1 & -c_2 \\ & & & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & & & c_3 & -c_0 & c_1 & -c_2 \end{bmatrix}$$

- One of the intrinsic advantages of the wavelet transform is that only requires an order(N) computational effort and is much faster than the FFT at vector transformation..

3.0 Filtering the Coefficients

- Filtering of wavelet coefficients is done by removing the smaller coefficients in the difference terms. In these terms are the detail which can often be removed without making a large difference to the overall structure. Two types of filtering exist.

1. Hard Thresholding

A difference term is treated as follows:

$$d = 0 \text{ if } |d| < \lambda \text{ and otherwise is not touched.}$$

2. Soft Thresholding

As above but all other values for which $d > \lambda$ have the following operation done to them:

$$d \leftarrow \text{sgn}(d)(|d| - \lambda)$$

This repositions the remainder of the coefficients.

- The literature suggests that λ is best set to $\frac{\sigma\sqrt{2\ln n}}{\sqrt{n}}$. The standard deviation σ is taken over all the difference terms. 'n' is the number of difference terms.