**An-Najah National University**
**Faculty of Graduate Studies**

# Data Compression with Wavelets

**By**

**Rana Bassam Da'od Ismirate**

**Supervisor**

**Dr. Anwar Saleh**

**Submitted in partial fulfillment of the requirement for the degree of Master of Science in computational mathematics, faculty of graduate studies, at An-Najah National University, Nablus, Palestine**

**2009**

# Data Compression with Wavelets

By

**Rana Bassam Da'od Ismirate**

This thesis was defended successfully on 1/7/2009 and approved by:

| Committee Member | Signature |
|---|---|
| 1. Dr. Anwar Saleh (Supervisor) | |
| 2. Dr. Samir Matar (Internal examiner) | |
| 3. Dr. Saed Mallak (External examiner) | |

# Dedication

**I dedicate this work to my parents and my husband, also to my sister and my brothers.**

# Acknowledgment

I thank my thesis advisor, Dr. Anwar Saleh for introducing me to the subject of wavelets besides his advice, assistance, and valuable comments during my work on this thesis. My thanks, also, goes to Dr. Samir Matar and Dr. Saed Mallak. I am grateful to my husband for his encouragement and support during this work. Finally, I won't forget my parents, sister, and brothers for their care and support.

## إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

**Data Compression with Wavelets**

**ضغط البيانات بواسطة المويجات**

اقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل أية درجة علمية أو بحث علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

## Declaration

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

**Student's name:** **اسم الطالب**

**Signature:** **التوقيع:**

**Date:** **التاريخ:**

Table of Contents

**Data Compression with Wavelets**

**By**
**Rana Bassam Da'od Ismirate**
**Supervisor**
**Dr. Anwar Saleh**

## Abstract

There are two types of data compression; the first is lossless (exact) and the second is lossy (approximate). In lossless compression, all details are reserved but high compression ratios can not be achieved and this type is not considered in this thesis. The other type is the lossy compression where some details are lost in the process of compression. The size of the lost details is proportional with the the desired compression ratio which is controlled by the user. Using this type, high compression ratios can be achieved with acceptable resolution in the reconstructed data.

In this thesis, a computational study of the classical Fourier transform and the relatively new wavelet transform is done. In addition, a computational comparison between the two major transforms shows that the wavelet transform is more efficient than the classical Fourier transform. The high compression ratios that can be achieved by wavelet transform lead to the introduction of several wavelet-based lossy data compression software. Examples of these are the image compressor JPEG2000 and the text compressor DJVU.

# Chapter One

# Introduction

## 1.1 Approximation Theory

# Chapter One

## Introduction

Wavelets are functions that satisfy certain mathematical requirements and are used to analyze functions or to represent a set of sampled data by a function for further processing. Approximation of functions is not a new idea. In 1715, Brook Taylor discussed a power series expansion of analytic functions; the result is the well known Taylor series. In 1822, Joseph Fourier began the development of the theory of trigonometric series. In fact, Fourier paper in 1807 [5] about convergence of trigonometric series was rejected by other scientists and was not published until 1822.

Joseph Fourier, have used the orthogonal set $\{\cos(nx), \sin(nx)\}_{n=0}^{\infty}$ to represent periodic functions. The idea has been extended to non-periodic functions through Fourier integral. However, Fourier approximation is poor in approximating functions near discontinuities where Gibbs phenomenon persists. Search for methods to better approximate functions continued and wavelets become the answer for many applications.

While the origins of wavelets go back to the early years of 1900's, wavelet analysis and the extensive use of wavelets in applications is a relatively new subject. The term wavelet was first mentioned in 1909 by Haar in his doctoral thesis on orthogonal systems of functions [2]. Between 1960 and 1980, the mathematician Guido Weiss and Roland R. Coifman [2] studied the simplest elements of a function space, called atoms, the goal

was finding atoms for a common function and finding “assembly rules” that allow to reconstruct the function using these atoms. In 1980, Grossman and Morlet, defined wavelets in the context of quantum physics [2]. However, these two researchers provided a way of thinking of wavelets. The use of wavelets in applications goes back to the late 1980’s and early 1990’s; in 1985 there have been many research activities to construct wavelet with specified properties. The search was initiated by French mathematician Yves Meyer [7], who was sure that no wavelet could be infinitely often differentiable and decay exponentially. However, when he was trying to prove his claim he realized that he was wrong: in fact, he found a wavelet with properties he thought were impossible. Today, this wavelet is called Meyer’s wavelet. The first wavelet was constructed on some clever observations and tricks. For this reason there was a belief that wavelets are so special that the ways to find them are limited. But, around 1987 Stéphan Mallat and Yves Meyer [7] found conditions under which wavelets can be constructed. Their findings lead to multi-resolution analysis. In 1990, Ingrid Daubechies [7] boosted the theory of wavelets by constructing wavelets with compact support via multi-resolution analysis. Since then, much work, especially in applications of wavelets, has been done. Currently, research in both the theory and applications is an active field. The mathematical theory of wavelets is a challenging one but it is not subject of this thesis.

There are many fields where wavelets analysis can be used and give results superior to the classical Fourier analysis. For instance, the new JPEG2000 image format, which is used in digital cameras, compresses

images using wavelets transform. Data compression is an important field in archiving, processing, and transmitting data. In this thesis, we will review discrete wavelets transform, as well as, the discrete Fourier transform. Then a computational study of the differences between the two major transforms is carried out. In signal analysis, a signal is considered as a function of time. Computationally, the signal is sampled and is given as a discrete function of time or as a sequence of time. Many signals, exhibit a periodic behavior where Fourier transform can be used efficiently. There is a wide range of applications of wavelet. Some of such applications are listed below:

- **Denoising:** Old recordings, Long distance telephone messages, and images produced by electronic microscopes are often containing significant amounts of noise and wavelets are used to filter out such noise.The main idea is to transform the signal and remove the high frequency components which usally represents a noise in the signal.

- **Compression:** There are two categories of compression techniques. The first category is lossless compression. Examples of lossless compression methods are Huffman compression which is used in compression programs such as winzip files which is not the subject of this thesis. This type of compression achieves complete error free compressed signal. Unfortunately, compressions that can be obtained with lossless methods are rarely more than 2:1.

  The second category is lossy compression. A lossy compression method is one which produces inaccuracies in the signal. Lossy

techniques are used when these inaccuracies are too small to be imperceptible. The advantage of lossy techniques over lossless ones is that higher compression ratios can be achieved.

Raw images produced by digital cameras require a large storage space and high speed connections to transmit to another device or over the internet. Compressed images require less storage space and can be transmitted efficiently over low speed connections. Compression using wavelets proved to be more efficient. In fact, the late JPEG2000 uses wavelet transform as a compression method. The FBI with 25 million set of fingerprints found the solution to the storage of over 250 trillion bytes and the manual processing of such fingerprints by adoption of wavelets as their main storage and processing method.

DjVu is wavelet-based compression method for text files. It creats high quality scanned pages that are much smaller than PDF files.

- **Object detection:** This is another challenging application. Voice and image or character recognition becomes more important. Voice recognition can be used to replace the keyboard by dictating to the machine orally. Image recognition can be used in airports to pinpoint fugitives. Wavelets is a perfect tool for such applications.

## 1.1 Approximation Theory

In many applications of mathematics, we face functions which are far more complicated than the standard functions from classical analysis. Some of these functions can not be expressed in closed form via the standard functions, and some are only known implicitly or via their graph. Think of the integral $I = \int_a^b e^{x^2}\, dx$. There is no closed form for this integral and approximation is needed. Approximation is carried out in two stages: interpolate $e^{x^2}$ over the interval $[a,b]$, then integrate the polynomial from *a* to *b*. This is an example where approximation of a function is needed. An important theorem regarding the approximation of continuous functions by polynomials is given by Weierstrass which says that any continuous function on a closed interval of $\mathbf{R}$ can be approximated by a polynomial to within an arbitrarily $\varepsilon > 0$.

**Theorem 1.1** [6]**: Weierstrass Approximation Theorem**

Suppose that $f$ is defined and continuous on $[a,b] \in \mathbf{R}$. For each $\varepsilon > 0$, there exists a polynomial $P_\varepsilon$, with the property that

$$|f(x) - P_\varepsilon(x)| < \varepsilon, \quad \text{for all } x \in [a,b].$$

**Theorem 1.2** [6]**: Taylor Theorem**

Suppose $f \in C^n[a,b]$, that $f^{(n+1)}$ exists on $[a,b]$, and $x_0 \in [a,b]$. For every $x \in [a,b]$, there exists $\xi(x)$ between $x$ and $x_0$ such that

$$f(x)=\sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k+\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}=P_n(x)+R_{n+1}(x)$$

where $P_n(x)=\sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k$ and $R_{n+1}(x)=\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}$.

The polynomial $P_n$ is called the Taylor polynomial of degree $n$ of $f$ at the point $x_0$ and $R_{n+1}(x)$ is the remainder.

The remainder term in the Taylor theorem is not computable since the value $\xi$ is undetermined, however, the remainder term can be used to bound the error in Taylor polynomial. Taylor polynomial approximates a function near a point $x_0$. Accuracy of Taylor polynomial at a given point $x$ depends on two factors; the distance between $x$ and $x_0$ and the degree of the polynomial.

**Remark:** McLaurin polynomial is the Taylor polynomial about $x_0=0$.

**Example 1.1:**

The $5^{\text{th}}$ degree Taylor polynomial of the function $f(x)=\sin(x)$, $x\in\mathbf{R}$ about $x_0=0$ is given by $\mathrm{P}_5(\mathrm{x})=x-\frac{x^3}{3!}+\frac{x^5}{5!}$, with remainder $\mathrm{R}_7(\mathrm{x})=\frac{-\cos(\xi)}{7!}$ where $\xi$ lies between $x_0=0$ and $x$.

**Example 1.2:**

The function $f(x)=e^x$ has derivatives of all orders at any point $x\in\mathbf{R}$. The $n^{th}$-degree Taylor polynomial of $f$ about any $x_0\in\mathbf{R}$ is given by

$$P_n(x)=\sum_{k=0}^{n}\frac{e^{x_0}}{k!}(x-x_0)^k$$

Figure 1.1 shows the graphs of $f$ together with Taylor polynomials $P_1$, and $P_2$ about $x_0 = 1$. The graph of the 2nd-degree Taylor polynomial is closer to the graph of $f$ over a larger interval centered at $x_0 = 1$ when compared to the graph of $P_1$.

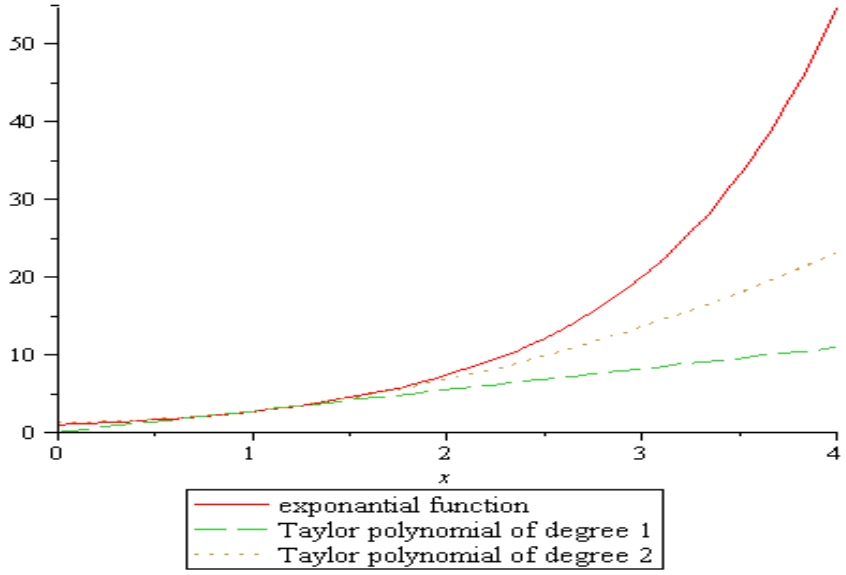



**Figure** 1.1

The remainder term in Taylor theorem shows that accuracy in Taylor polynomial depends on its degree and the distance from $x$ to $x_0$. In fact, the maximum errors in $P_1(2)$ and $P_2(2)$ equal 1.952492443 and 0.593351529 respectively. Moving away from $x_0$, we find the maximum errors $P_1(3)$ and $P_2(3)$ equals 11.93069144 and 6.49412778 respectively. This means the error becomes larger when moving away from $x_0$ and stays smaller for larger degree.

# Chapter Two

# Signal analysis

## 2.1 Signals

## 2.2 Sampling and Interpolation

## 2.3 Periodic Signals

## 2.4 Frequency Domain

## 2.5 Time and Frequency Domains

## 2.6 Special Types of Signals

## 2.7 Signals of Finite Energy

## 2.8 Signal Transmission

## 2.9 The Uncertainty Principle

# Chapter Two

# Signal Analysis

Signals are functions with their values depend on the phenomenon they represent. Gray scale images, for example, are signals in two dimensions with values equal to the light intensity at each point. In our daily life we communicate using different means. Audio (sound) messages and electrocardiogram are one-dimensional signals. Images, fingerprints, and video files are treated as two-dimensional signals. For computational purposes, such signals are sampled, producing digital signals, so they can be processed by digital computers. Processing digital signals is a wide range field known as Digital Signal Processing (DSP). Audio files are one-dimensional signals while images and text files are two-dimensional signals. Because of the importance of signal processing as a major application of wavelets, we give a brief review of signal analysis.

## 2.1 Signals

**Definition 2.1** [1]**: Analog Signals**

An analog signal is a function $f : \mathbf{R} \to \mathbf{R}$, where $\mathbf{R}$ is the set of real numbers, and $f(t)$ is the signal value at the independent variable $t$. A complex-valued analog signal is a function $f : \mathbf{R} \to \mathbf{C}$, where $\mathbf{C}$ is the set of complex numbers and $f(t) = u(t) + iv(t)$ with $u$ as the real part of $f$ and $v$ as the imaginary part $f$. Both parts are real valued signals and $i^2 = -1$.

**Example: 2.1**:

The function $g(t) = 11\sin(3\pi t) + 6\cos(5\pi t)$, where $t \in \mathbf{R}$, is an analog signal. It is a combination of two sinusoidal functions.

Definition (2.1) means that the theory of analog signals follows the theory of functions of real variables. Complex-valued signals appear in the study and analysis of the frequency contents of a signal. A generalization of this definition can be made to cover multi-dimensional signals from $\mathbf{R}^n \to \mathbf{R}$, for $n \geq 2$. For example, a photo or a text is a two-dimensional signal. Some special analog signals are listed below.

1. Unit step signal $u(t)$, see Figure 2.1(a), is defined by:

$$u(t) = \begin{cases} 1 & \text{if} \quad t \geq 0, \\ 0 & \text{if} \quad t < 0. \end{cases} \tag{2.1}$$

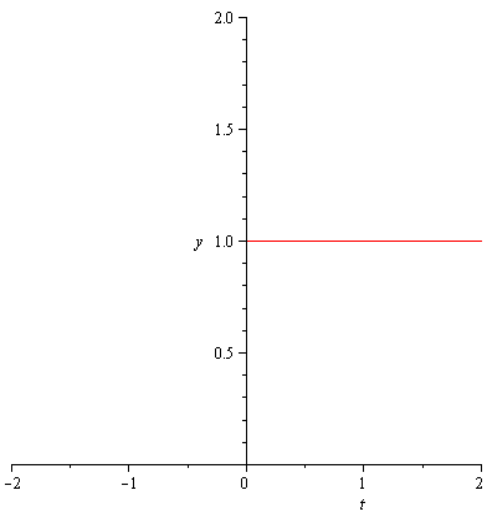
2. Sawtooth signal $f(t)$, see Figure 2.1(b), is the piecewise linear signal defined by:

$$f(t) = \begin{cases} t & \text{if} \quad t \geq 0, \\ 0 & \text{if} \quad t < 0. \end{cases} \tag{2.2}$$

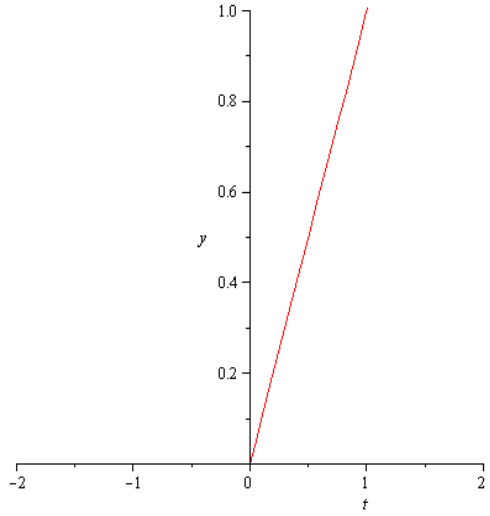
3. Dirac delta $\delta_n(t)$ , see Figure 2.1(c,d), is a sequence of analog signals $\delta_n(t)$ where $n > 0$ defined by:

$$\delta_n(t) = \begin{cases} \dfrac{n}{2} & \text{if} \quad t \in \left[\dfrac{-1}{n}, \dfrac{1}{n}\right], \\ 0 & \text{if} \quad \text{otherwise}. \end{cases} \tag{2.3}$$

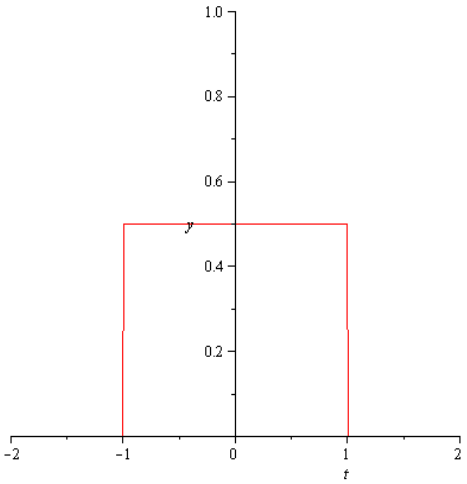
Sometimes, the values of a signal can be given by a formula or an expression of the independent variable as in the analytical solution of a differential equation. In many cases, however, such a closed formula can not be obtained and approximation is needed for processing such signals. This leads to discrete and digital signals.



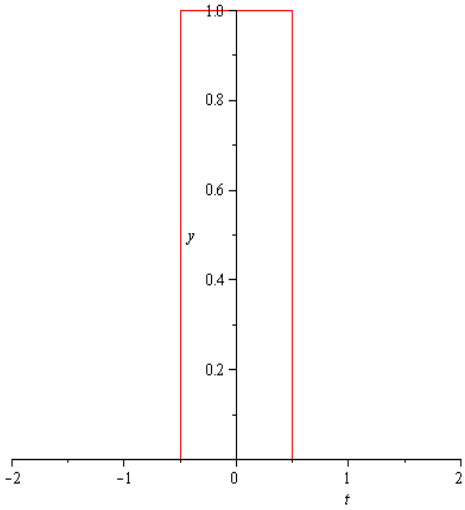

(a) Unit step signal $u(t)$




(b) Sawtooth signal




(c) $\delta_1(t)$




(d) $\delta_2(t)$

**Figure** 2.1

**Definition 2.2** [1]**: Discrete signals**

A discrete signal in one dimension is a real-valued function $x : \mathbf{Z} \to \mathbf{R}$ and $x(n)$ is the signal's value at instant $n$; $\mathbf{Z}$ is the set of integers. In other words, a discrete signal is a sequence of real numbers. A generalization can be easily made for multi-dimensional discrete signals. Some special discrete signals are listed below:

1. **Sinusoids:** discrete sinusoid signals such as $\sin(\omega n)$ or $\cos(\omega n)$. The function $\sin(\omega t + \phi)$ is the discrete sine function of radial frequency $\omega$ and phase $\phi$. Also $\cos(\omega t + \phi)$ is a sinusoid as well.

(a) $x(n) = \cos(\frac{\pi}{24} n + \frac{\pi}{3})$ (b) $x(n) = \cos(\frac{1}{7} n)$

**Figure** 2.2

2. **Exponentials**: a discrete exponential function is defined by:

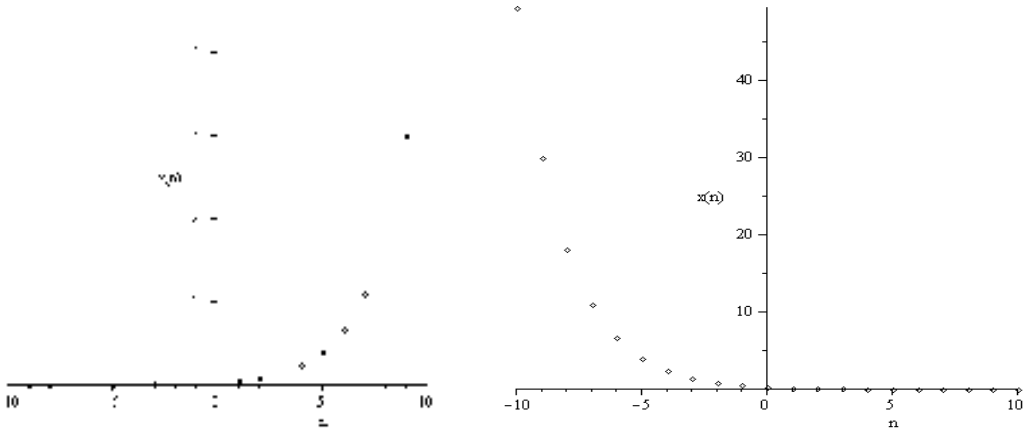
$$x(n) = ce^{an} \tag{2.4}$$

where $c$ and $a$ are constants. Discrete exponential are used in frequency domain signal analysis. See Figure 2.3(a,b).

3. **Discrete delta**: the discrete delta or impulse signal $\delta(n)$ is defined as:

$$\delta(n) = \begin{cases} 1 & \text{if } n = 0, \\ 0 & \text{if } n \neq 0. \end{cases} \tag{2.5}$$

It is zero everywhere except at the origin, which is one.




(a) $x(n) = \frac{1}{3}\exp(n/2)$ (b) $x(n) = \frac{1}{3}\exp(-n/2)$

**Figure** 2.3

4. **Discrete unit step**: the unit step signal is defined as:

$$\mathrm{u}(n) = \begin{cases} 1 & \text{if} \quad n \geq 0, \\ 0 & \text{if} \quad n < 0, \end{cases} \tag{2.6}$$

See Figure 2.4

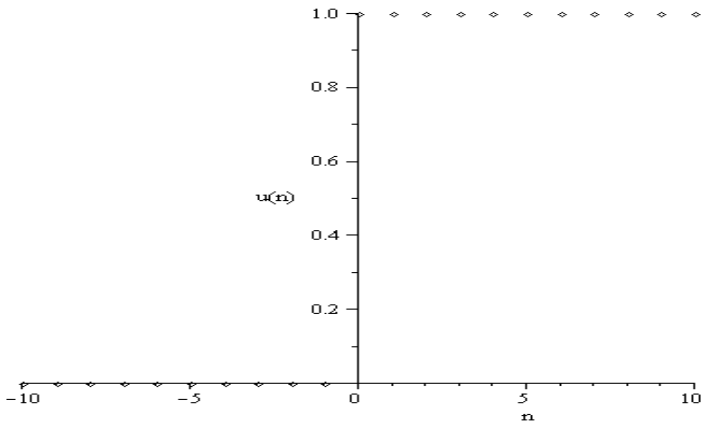



**Figure** 2.4

**Definition 2.3** [1]**: Digital signals**

A digital signal in one dimension is a discrete integer-valued signal $f : \mathbf{Z} \to \mathbf{Z}$, which means it is a sequence of integers.

We will express a discrete signal in the form $f = (f_1, f_2, ..., f_N)$, where $N$ is positive integer. For simplicity, we restrict $N$ to powers of 2; I mean $N = 2^n$ for $n$ integer. The components of $f$ are the $N$ real numbers: $f_1, f_2, ..., f_N$ representing an analog signal $g$ measured at equally spaced time values $t = t_1, t_2, ..., t_N$. i.e.

$$f_1 = g(t_1),\ f_2 = g(t_2), ...., f_N = g(t_N) \tag{2.7}$$

**Example 2.2:**

The signal $f = (2,6,9,4,3,8,0,5)$ is considered as a digital signal in 1D. Also, the signal $g = \begin{pmatrix} 5 & 7 & 4 & 3 \\ 4 & 3 & 6 & 8 \\ 7 & 5 & 4 & 3 \\ 6 & 7 & 9 & 5 \end{pmatrix}$ is considered to be a digital signal in 2D.

**Definition 2.4: The energy of an analog signal**

The energy of an analog signal $f$ is defined as the inner product:

$$< f, f >= \int_{-\infty}^{\infty} f^2(t)\, dt$$

**Definition 2.5: The energy of a discrete signal**

The energy $\varepsilon$ of a discrete signal $f = (f_1, f_2, ..., f_N)$ is the sum of the squares of its components or $\varepsilon_{(f)} = \sum_{k=1}^{N} f_k^2$ .

**Remark:** Any digital signal must be bounded, that is a number $M > 0$ exists such that $|f_k| \le M$ for all $k$ where $1 \le k \le N$. It is also true that signals have finite energy which means any signal satisfies $\varepsilon_{(f)} < \infty$.

## 2.2 Sampling and Interpolation

In nature, signals are analog. For processing signals on digital machines, digital signals are more convenient. Sampling converts an analog signal to a digital signal by taking values of the analog signal at discrete time intervals; usually, regular time intervals. Accurate representation of an analog signal depends on its sampling. On the other hand, interpolation converts a digital signal into an analog signal. In some applications, sampling is done manually, for instance, sampling the room temperature by measuring the temperature every one hour or even every 15 minutes. However, there are applications where much larger sample is to be taken. In such cases, digital sampling instruments are used. In digital photography, a 5 mega-pixel

digital camera samples a picture (a two-dimensional signal) by $5\times10^{10}$ pixels (picture element) where each pixel is assigned a color.

The sampling interval is the time (or other measure dimension) between samples. For a time signal, the sampling frequency is measured in hertz (Hz), which is one cycle per second. On the other hand, the sampling frequency for distance signal with sampling interval in meters is in units of $(\text{meter})^{-1}$.

For theoretical work, discrete signals are more convenient, but for computer processing, only a finite number of bits can represent a signal's value in binary form. The signal must be digitized, or the signal values must be quantized. Squeezing the signal value into an *N*-bit register, some fraction of the true signal is lost; the fraction is called quantization error. Also, the number of possible digital signal values is called the dynamic range of the conversion.

**Definition 2.6** [1]**: Sampling**

Sampling is the reduction of a continuous signal into a discrete signal. A common example is the conversion of a sound wave which is continuous-time signal to a sequence of samples representing a discrete time-signal.

**Definition 2.7** [1]**: Sampling rate**

The sampling rate (frequency) $f_s$ of a time-signal is the number of samples obtained in one second and $f_s = \frac{1}{T}$ where $T$ is the sampling

interval. Sampling rate is measured in hertz which is one sample per second.

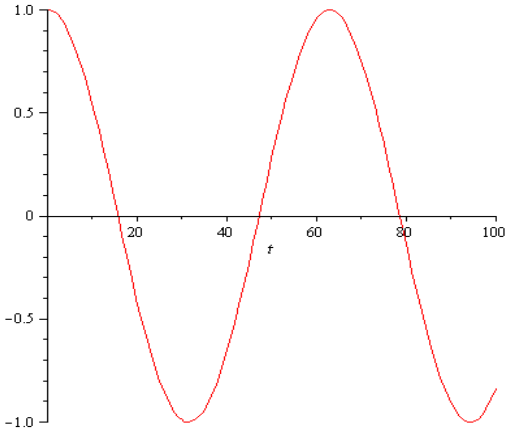
An *N*-bit register can hold non-negative values from 0 to $2^N$-1. When all bits are clear the smallest value is present, and when all bits are set, the largest value is present. Suppose there are $N$ bits available in the input register, and the quantized signal's bit values are $b_{N-1}, b_{N-2}, ..., b_1, b_0$. Then the digital value is

$$D = b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + ... + b_2 2^2 + b_1 2^1 + b_0 2^0 \qquad (2.8)$$

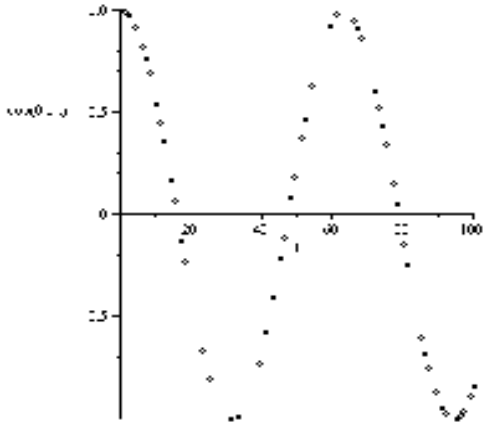
In this form, a register full of zero represents a digital zero value. The dynamic range of an $N$-bit register is $2^N$.

**Example 2.3:**

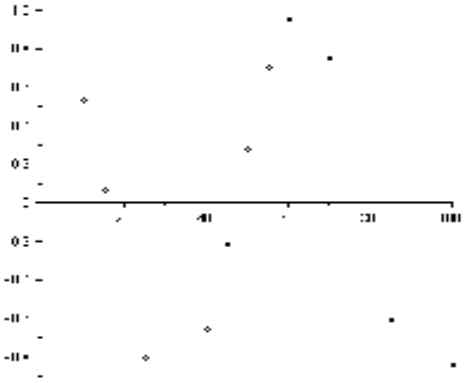
Consider the analog signal: $f(t) = \cos(0.1t), 1 \le t \le 100$. We sample $f$ using three different sampling intervals; $T = 1$, $T = 5$, and $T = 20$. The corresponding sampling rates are 100, 20, and 5 respectively. Figure 2.5(b) clearly shows that the signal with high sample rate approximates the analog signal more accurately than signals with less sampling rate as in figure 2.5 (c) and (d).



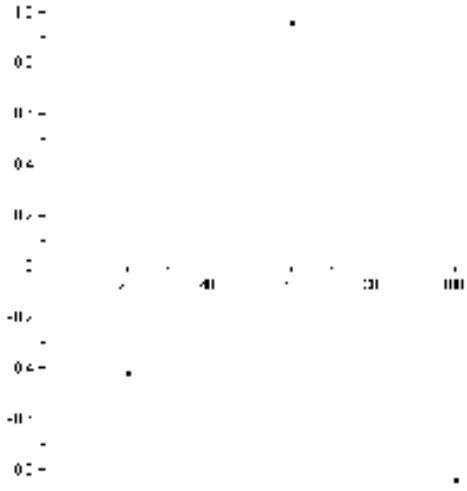

(a) $f(t) = \cos(0.1t)$



(b) $T = 1$



(c) $T = 5$



(d) $T = 20$

**Figure** 2.5

**Definition 2.8** [6]**: Algebraic polynomials**

An algebraic polynomial is a function in the form:

$$P_n(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t + a_0$$

Where $n$ is a non-negative integer, $a_i$ is a real constant for each $i = 0,1,\ldots,n$.

Weierstrass Approximation theorem guarantees the existence of a polynomial that approximates any continuous function defined on closed and bounded interval to within any $\varepsilon > 0.$ Also, the importance of polynomials in the approximation of functions is that the derivatives and indefinite integral of a polynomial are polynomials and are easy to determine. This is why polynomials are often used for approximating continuous functions. Taylor polynomial, given in theorem (1.2), approximates a function near a point. In some applications, a tabulated function over a long interval, need to be processed. Polynomial interpolation approximates the given tabulated function over the entire interval.

**Theorem 2.1** [6]**: Lagrange interpolation**

If $f$ be a function defined at the $n+1$ distinct numbers: $t_0, t_1, ..., t_n$, then there exists a unique polynomial $P(t)$ of degree at most $n$ with the property that

$$f(t_k) = P(t_k) \quad \text{for each } k = 0,1,...,n.$$

This polynomial is given by

$$P(t) = \sum_{k=0}^{n} f(t_k) L_k(t), \tag{2.9}$$

where

$$L_k(t) = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(t - t_i)}{(t_k - t_i)} \qquad \text{for each } k = 0,1,...,n. \tag{2.10}$$

There are other forms of the interpolating polynomials, like Newton forms, which have computational advantages over Lagrange form and they can be found in any numerical analysis text book.

**Example 2.4:**

The second-degree interpolating polynomial for $f(t) = 1/t$ using the numbers (nodes) $t_0 = 2, t_1 = 2.5, \text{and } t_2 = 4$ is:

$$P(t) = \sum_{k=0}^{2} f(t_k) L_k(t) = 0.05t^2 - 0.425t + 1.15$$

Figure 2.6 shows the graph of $f$ (solid) together with the graph of $P$ (dots)

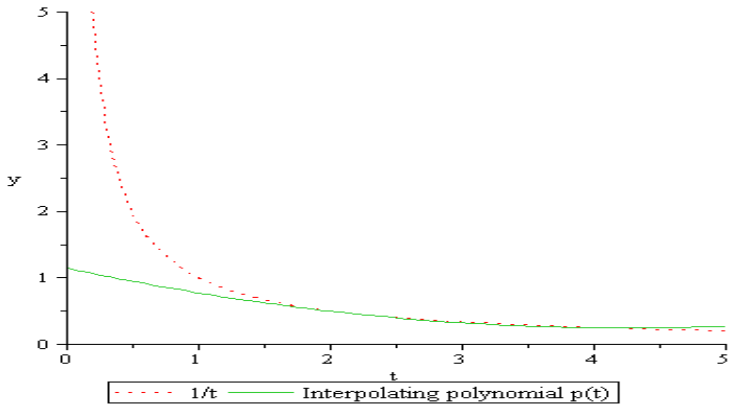



**Figure** 2.6

## 2.3 Periodic Signals

Periodic signals, whether analog or discrete, repeat their values over intervals. A sinusoid signal is a good example of periodic signals. The period of a function is the interval on which a signal repeats itself, and the reciprocal of its period is its frequency. If a signal repeats itself over an

interval, then it repeats itself over any positive integer multiple of that interval. We will characterize a periodic signal by the smallest interval of repetition.

**Definition 2.9** [1]**: Periodicity**

An analog signal $f$ is periodic if there is a number $T > 0$ with $f(t+T) = f(t)$ for all $t$. A discrete signal $x$ is a periodic if there is an integer $N > 0$ with $x(n+N) = x(n)$ for all $n \in \mathbf{Z}$ . The smallest value, for which a signal is periodic, is called the fundamental period.

**Definition 2.10** [1]**: Amplitude**

The amplitude of a signal is its maximum value.

**Example 2.5:**

The signal $f(t) = 3\cos(\frac{\pi}{2}t)$ is periodic with fundamental period 4 and amplitude $3$ as in figure 2.7.

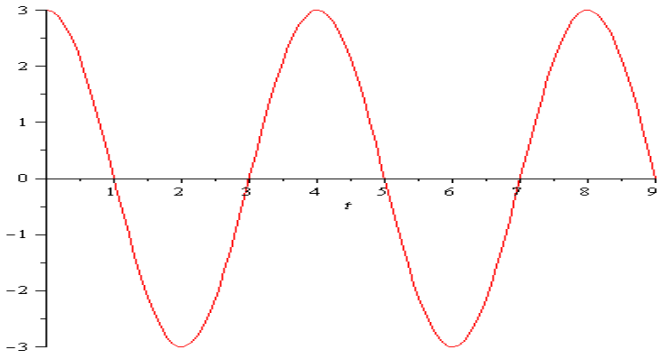


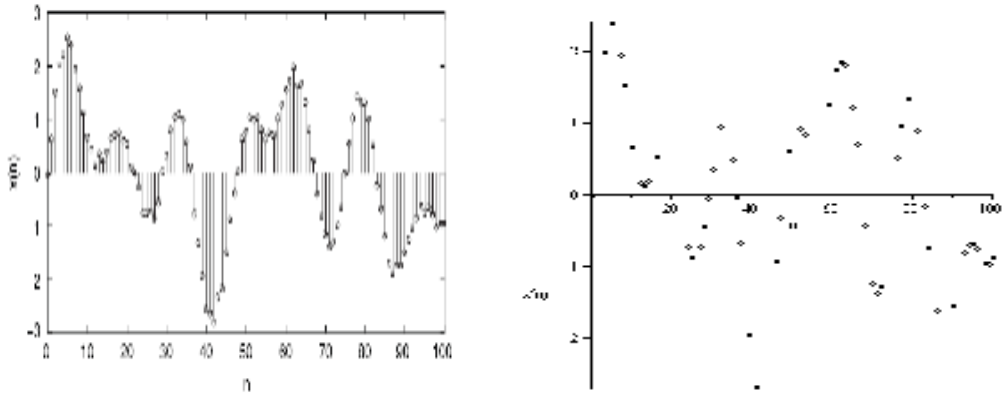

**Figure** 2.7

## 2.4 Frequency Domain

The signals; $\sin(t)$ and $\cos(t)$ are the most popular for identifying periodicities in signals. This is done by decomposing the signal in terms of $\sin(\omega t)$ and $\cos(\omega t)$ forming a trigonometric series. For periodic signals, convergence of a series means that only the first few terms are significant and the rest are considered noise in the signal. Therefore, the signal is approximated using few terms (partials sums) of the series representing the signal.

**Example 2.6:**

Consider the signal in figure 2.8(a), which consists of a series of irregular pulses.



(a) (b)

**Figure** 2.8

We can write the signal in figure 2.8(a) as a sum of three sinusoid and small noise component:

$$x(n) = \sin\left(\frac{2\pi n}{15}\right) + \sin\left(\frac{2\pi n}{25}\right) + \sin\left(\frac{2\pi n}{50}\right) + N(n)$$

Notice that the graphs of the three sinusoids in figure 2.8(b) are very close to the original signal in Figure 2.8(a).

## 2.5 Time and Frequency Domains

For signal in time-domain, which is a sequence of numbers, we can identify each value for this signal at a specified time, but we can't recognize its shape at a specified time. On the other hand, signal in frequency-domain we can identify its shape and recognize the part which contains high frequency but we don't know at any time.

Frequency-domain analysis is good and effective when we are dealing with regular signals. However, frequency-domain analysis is not as effective when dealing with irregular signals such as irregular heart beat and seismic signal interpretation. Our ultimate goal is to analyze a signal for its time and frequency.

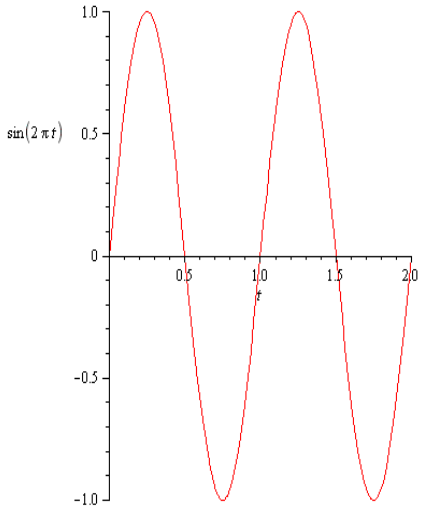
**Definition 2.11** [20]**: Frequency**

The frequency $\omega$ is the inverse of the period. It is measured in cycle per second or hertz (Hz).

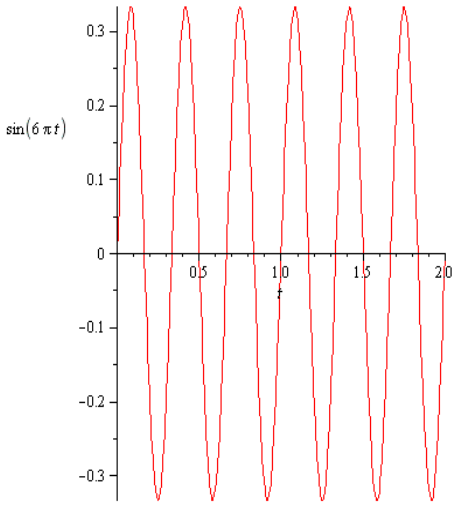
To understand the frequency domain, let's look at two simple examples.

**Example 2.7:**

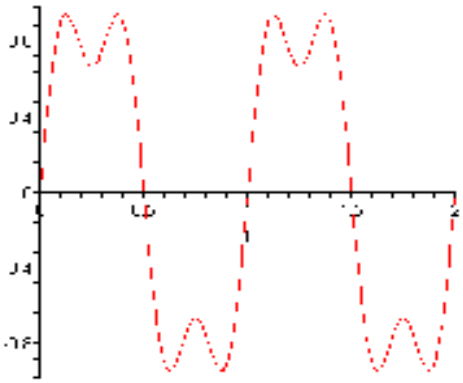
The function $g(t)=\sin(2\pi t)+\frac{1}{3}\sin(6\pi t)$ is a combination of two sine waves with amplitudes 1 and $\frac{1}{3}$, and frequencies 1 and 3, respectively.



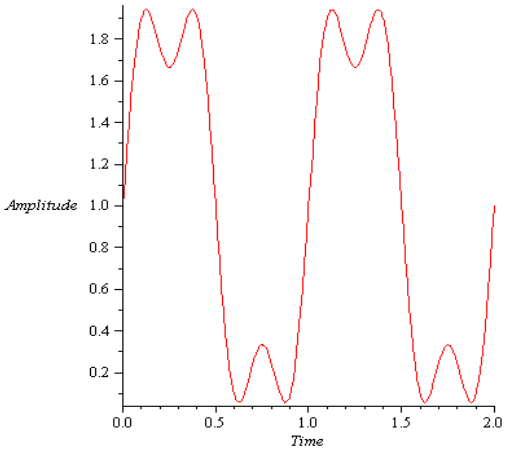

(a) $\sin(2\pi t)$




(b) $\frac{1}{3}\sin(6\pi t)$



(c) $\sin(2\pi t)+\frac{1}{3}\sin(6\pi t)$




(d) $\sin(2\pi t)+\frac{1}{3}\sin(6\pi t)+1$

**Figure** 2.9

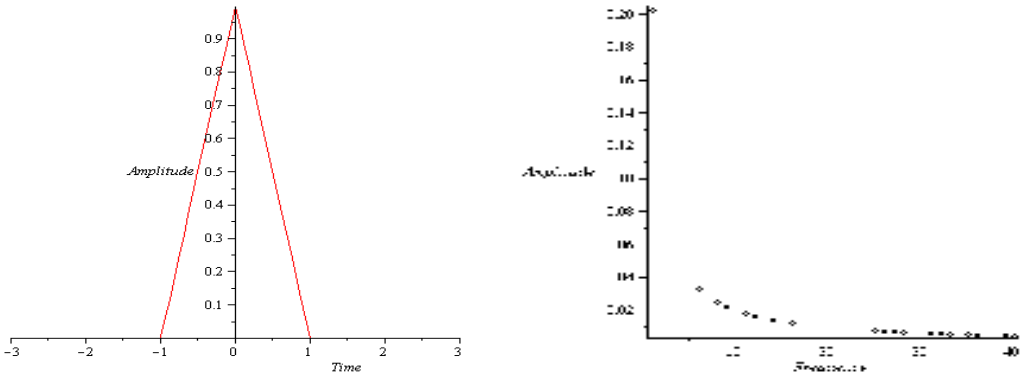
Their sum, shown in Figure 2.9(c), is periodic with frequency1 (the smallest). The frequency domain of $g(t)$ is a function consisting of just two points $(1,1)$ and $\left(3,\frac{1}{3}\right)$.

**Remark**: Every function has a simple frequency domain representation.

**Example 2.8:**

Consider the signal $g(t)$

$$g(t)=\begin{cases}1-|t|, & -1<t<1\\ 0\ , & otherwise\end{cases}$$




(a) (b)

**Figure** 2.10

The Fourier sine series of $g(t)$ is given by:

$$g(t)=\sum_{n=1}^{\infty} b_n \sin(n\pi t)$$

Where

$$b_n=\frac{2(n\pi-\sin(n\pi))}{\pi^2 n^2}$$

The function $g(t)$ is localized in time domain as we can see in Figure 2.10(a), but it is not localized in frequency domain as we can see in Figure 2.10(b). The signal $g(t)$ is decomposed as an infinite sum of sinusoids signals each of frequency $\omega = \frac{1}{n}$ and amplitude $b_n$. The above series is known as the Fourier sine series and will be discussed in chapter 3. Figure 2.10(a) shows the time domain of $g(t)$ and Figure 2.10(b) shows the frequency domain of $g(t)$. It is obvious that signal $g(t)$ is well localized in the time domain but not localized in the frequency domain.

Periodic functions can be represented in frequency domain as the sum of sine waves with frequencies that are integer multiples of some fundamental frequency. Note that the signal in Figure 2.10(a) is not periodic in time domain and we noticed that frequency domain concept can be applied to a nonperiodic function, but only if it is nonzero over a finite range (localized). Localized functions can be represented as the sum of sine waves with all kind of frequencies not only the harmonics.

**Definition 2.12** [20]**: The spectrum**

The spectrum of frequency domain (also called the frequency content of the function) is the range of frequencies it contains.

For instance the function in figure 2.9(c) has the spectrum $\{1,3\}$, while the spectrum of the function in Figure 2.10(b) is the entire range $[0,\infty)$.

**Definition 2.13** [20]**: The bandwidth**

Bandwidth of frequency domain is the width of its spectrum.

The bandwidth of the frequency domain of the function in figure 2.9(c) is $2$, but the bandwidth of figure 2.10(b) is $\infty$.

The concept of time and frequency domains is due to the French mathematician Joseph Fourier. He proved that any periodic function real or complex can be represented as the sum of sine and cosine functions. Moreover, he showed how to transform between the time and the frequency domains.

## 2.6 Special Types of Signals

In this section, we briefly discuss signals of special types that make processing such signals easier.

**Definition 2.14** [1]**: Even and odd part of signals**

A discrete signal $x(n)$ can be decomposed into an even part:

$$x_e(n) = \frac{x(n) + x(-n)}{2} \tag{2.11a}$$

and an odd part:

$$x_o(n) = \frac{x(n) - x(-n)}{2} \tag{2.11b}$$

with

$$x(n) = x_e(n) + x_o(n)$$

**Definition 2.15** [1]**: Finite support**

A discrete signal $x(n)$ is finitely supported if there are integers $M < N$ such that $x(n) = 0 \text{ for } n < M \text{ and } n > N.$ An analog signal is finitely supported if it is zero outside some interval $[a,b]$ on the real line.

**Definition 2.16** [1]**: Absolutely summable signals**

A discrete signal $x(n)$ is absolutely summable (or summable) if the sum of its absolute values is finite; i.e.

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \tag{2.12}$$

**Definition 2.17** [1]**: Absolutely integrable signals**

An analog signal $f(t)$ is absolutely integrable (or integrable) if the integral of its absolute value over $\mathbf{R}$ is finite; i.e.

$$\int_{-\infty}^{\infty} |f(t)|\, dt < \infty \tag{2.13}$$

## 2.7 Signals of Finite Energy

The most important signal classes are the discrete and analog signals of finite energy.

**Definition 2.18** [1]**: Finite-energy discrete signals**

A discrete signal $x(t)$ has a finite energy (or square-summable) if

$$\sum_{t=-\infty}^{\infty} |x(t)|^2 < \infty \tag{2.14}$$

An analog signal $f(x)$ is of finite-energy (or square integrable) if

$$\int_{-\infty}^{\infty} |f(x)|^2 \, dx < \infty \tag{2.15}$$

## 2.8 Signal Transmission

Modern technology often requires that information can be sent from one place to another, which is called signal transmission. It occurs, for example, in wireless communication, the internet, computer graphics, or transfer of data from CD-ROM to computer.

All types of signal transmission are based on transmission of a series of a numbers. One can sample the graph and send the sampled graph which consists of large number of sampled data. A more efficient way is to find a convergent series representation of the signal and to send the most significant coefficients then the receiver can reconstruct the signal from these significant coefficients. We illustrate the major steps to send a signal $f$ from a sender *S* to a receiver *R* by assuming a power series representation of $f$.

- *S* finds $f(x) = \sum_{n=0}^{\infty} a_n x^n$;
- *S* sends the coefficients $a_0, a_1, \cdots$,
- *R* receives the coefficients $a_0, a_1, \cdots$,
- *R* reconstructs the signal by multiplying the coefficients $a_n$ by $x^n$ and forming the infinite series $f(x) = \sum_{n=0}^{\infty} a_n x^n$.

In practice, *S* can not send an infinite sequence of coefficients $a_0, a_1, \cdots$: so, *S* sends a finite sequence of coefficients. This finite set of coefficients must consist of the most significant coefficients. For a convergent series, the most significant coefficients are the first few coefficients.

**Example 2.9:**

Assume that *S* wishes to send the function

$$f(t) = \sin t, \quad t \in [0,3]$$

Using Taylor theorem

$$\sin t = \sum_{n=0}^{\infty} (-1)^n \frac{t^{2n+1}}{(2n+1)!} \approx 0 + t + 0.t^2 - \frac{1}{6}t^3 + 0.t^4 + \frac{1}{120}t^5$$

With $\frac{1}{6} \approx 1.1666$ and $\frac{1}{120} \approx 0.0083$, S sends the numbers: 0; 1; 0; $-0.1666$; 0; 0.0083

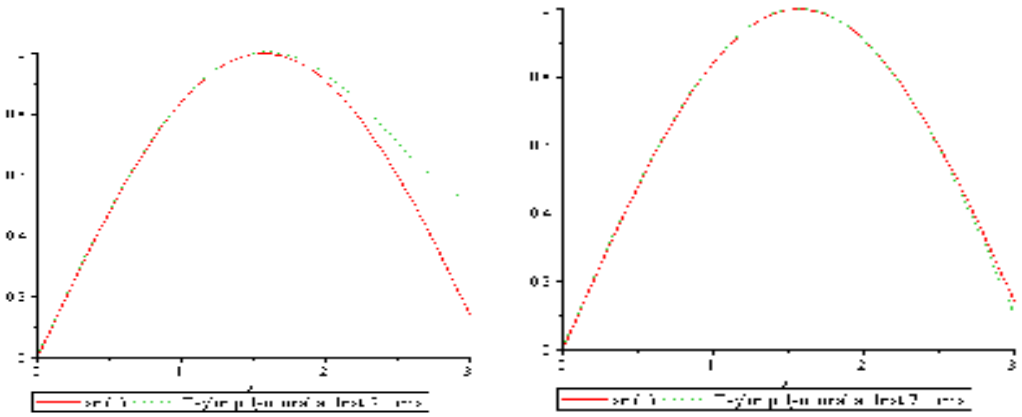
and *R* will reconstruct the signal from the received coefficients as

$$\tilde{f}(t) = 0 + t + 0.t^2 - 0.1666t^3 + 0.t^4 + 0.0083t^5$$

Figure 2.11(a) shows the original signal $f(t)$ and the reconstructed signal $\tilde{f}(t)$ over the interval $[0,3]$. From Figure 2.11(a) we can see that the reconstructed signal is close to the original signal when $x \in [0,2]$, but then the approximation starts to deviate from the signal. A better reconstruction can be reached if $S$ sends more coefficients. The next significant coefficient is $\frac{-1}{7!} = 0.000198$. Figure 2.11(b) shows the reconstructed signal

$$\tilde{f}(t) = 0 + t + 0.t^2 - 0.1666t^3 + 0.t^4 + 0.0083t^5 - 0.000189t^7$$

And the original signal over the interval $[0,3]$.



(a) (b)

**Figure** 2.11

Comparing Figure 2.11(a) with Figure 2.11(b) we see that Figure 2.11(b) is better since it is very close to the original signal. But, the problem here is that not every signal has a power series representation. So, we need to work with other types of series representation.

**2.9 The Uncertainty Principle** [20]

The relation between time and frequency domains is very important, they are complementary. Each of them complements the other, when one of them is localized the other must be global.

**Example 2.10:**

A pure sine wave is not localized in the time domain but it is well localized in the frequency domain. Example 2.8 shows a signal that is well localized in the time domain but not in in the frequency domain.

# Chapter Three

# Fourier Analysis

## 3.1 Fourier Series

## 3.2 Complex Form of Fourier Series

## 3.3 Fourier Transform

## 3.4 Fast Fourier Transform

### 3.4.1 Splitting Method

## 3.5 Two-Dimensional Fourier Transforms

## 3.6 Fourier Image Compression

# Chapter Three

# Fourier Analysis

Fourier analysis is a large classical area of mathematics. While Taylor series expands a function about a point $x_0 = 0$ using the basis $\{x^n\}_{n=0}^{\infty}$, Fourier series, uses the system $\{\sin(nx), \cos(nx)\}_{n=0}^{\infty}$ which is orthogonal over any period of length $2\pi$. The idea is to represent functions on **R** via trigonometric functions. This is suitable for periodic functions with periods $2\pi$. For a periodic function with period $2L$, the orthogonal system $\left\{\sin\left(\frac{n\pi x}{L}\right), \cos\left(\frac{n\pi x}{L}\right)\right\}_{n=0}^{\infty}$ is used. Non-periodic functions can be considered periodic with period $\infty$ and the result is the well known Fourier integral. In 1807 Fourier claimed that the Fourier series converges without any assumptions on the function, and later it was proved that the result does not hold as generally as Fourier believed. However, the theory is not part of this thesis.

## 3.1 Fourier Series

The functions $\sin(x)$ and $\cos(x)$ have periods $2\pi$. The functions $\sin(ax)$ and $\cos(ax)$, $a > 0$ have the period $2\pi / a$. If $f$ has period $p > 0$, it also has period $kp$ for $k \in \mathbf{N}$. Hence a periodic function can have many periods. The smallest period of $f$ is referred to as the period of $f$.

**Definition 3.1** [7]**: Fourier series expansion**

Let $f$ be a periodic function with period $p = 2\pi$, and square integrable on $[-\pi, \pi]$. Fourier series expansion of $f$ is given by:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}\left(a_n \cos nx + b_n \sin nx\right), \tag{3.1}$$

where the coefficients $a_n$, and $b_n$ are defined as:

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx, \; n = 0,1,2,\ldots \tag{3.2}$$

and

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx \; dx \;, \; n = 1,2,3,\ldots \tag{3.3}$$

The $N^{th}$ partial sum of the Fourier series is

$$S_N(x) = \frac{1}{2}a_0 + \sum_{n=1}^{N}(a_n \cos nx + b_n \sin nx). \tag{3.4}$$

**Definition 3.2** [16]**: The inner product**

Let $f$ and $g$ be two complex valued functions defined on the interval $[a,b]$. The inner product $< f, g >$ is defined by:

$$< f, g >= \int_a^b f(x)\overline{g(x)}dx$$

where $\overline{g}$ is the complex conjugate of $g$.

**Definition 3.3** [16]**: Orthogonality**

A set of functions $\{f_n(x)\}_{n=0}^{\infty}$ is orthogonal on an interval $I$, if $< f_n, f_m >= 0 \; \forall \; n \neq m$ on $I$.

**Theorem 3.1** [16]**:** the set $\{\sin(nx), \cos(nx)\}_{n=0}^{\infty}$ is orthogonal on $[0, 2\pi]$.

**Proof:** We want to show:

a. $\int_0^{2\pi} \sin(nx)\cos(mx)dx = 0 \qquad \forall\, n, m$

b. $\int_0^{2\pi} \sin(nx)\sin(mx)dx \quad = 0 \qquad \forall n \neq m$

c. $\int_0^{2\pi} \cos(nx)\cos(mx)dx \quad = 0 \qquad \forall n \neq m$

a. If $n = m$, then

$$\int_0^{2\pi} \sin(nx)\cos(mx)dx \quad = \quad \int_0^{2\pi} \sin(nx)\cos(nx)dx$$

$$= \quad \frac{1}{2}\int_0^{2\pi} \sin(2nx)dx == \quad \frac{1}{4n}\left[-\cos(2nx)\right]_0^{2\pi}$$

$$= \frac{-1}{4n}\left[\cos(4n\pi) - \cos(0)\right] = 0$$

If $n \neq m$, then

$$\int_0^{2\pi} \sin(nx)\cos(mx)dx = \frac{1}{2}\int_0^{2\pi} \cos(n-m)x - \cos(n+m)x \; dx$$

$$= \frac{1}{2}\left[\frac{1}{(n-m)}\sin(n-m)x - \frac{1}{(n+m)}\sin(n+m)x\right]_0^{2\pi}$$

Let $k = n - m$ and $r = n + m$. Then,

$$\int_0^{2\pi} \sin(nx)\cos(mx)dx = \frac{1}{2}\left[\frac{1}{k}\sin(kx) - \frac{1}{r}\sin(rx)\right]_0^{2\pi}$$

$$= \frac{1}{2}\left[\frac{1}{k}\sin(2\pi k) - \frac{1}{r}\sin(2\pi r)\right] = 0$$

Similarly we can prove (b) and (c).

**Theorem 3.2** [7]**:**

a. If $f$ is an even function, then $b_n = 0$ for all $n$, and

$$a_n = \frac{2}{\pi}\int_0^{\pi} f(x)\cos(nx)\,dx \quad , \quad n = 0,1,2,...$$

The result is a Fourier cosine series.

b. If $f$ is odd, then $a_n = 0$ for all $n$, and

$$b_n = \frac{2}{\pi}\int_0^{\pi} f(x)\sin(nx)\,dx \quad , \quad n = 1,2,3,...$$

The result is a Fourier sine series.

**Example 3.1:**

Consider the step function:

$$f(x) = \begin{cases} -1 & if \quad x \in [-\pi, 0), \\ 0 & if \quad x = 0, \\ 1 & if \quad x \in (0, \pi], \end{cases} \tag{3.5}$$

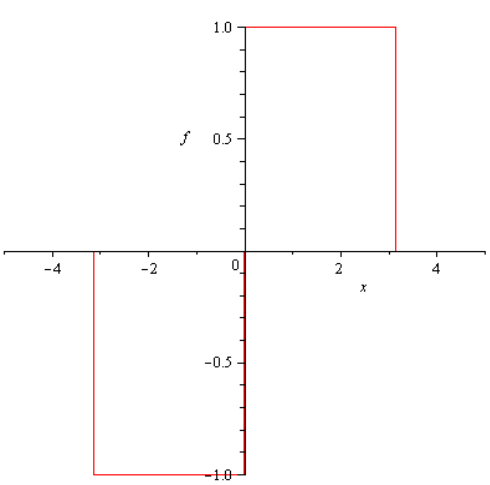
The periodic odd extension of the function $f$ is represented by a Fourier sine series with coefficients:

$$b_n = \frac{2}{\pi}\int_0^{\pi} f(x)\sin(nx)\,dx = \frac{2}{\pi}\int_0^{\pi} \sin(nx)dx = \begin{cases} \dfrac{4}{n\pi} & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases}$$

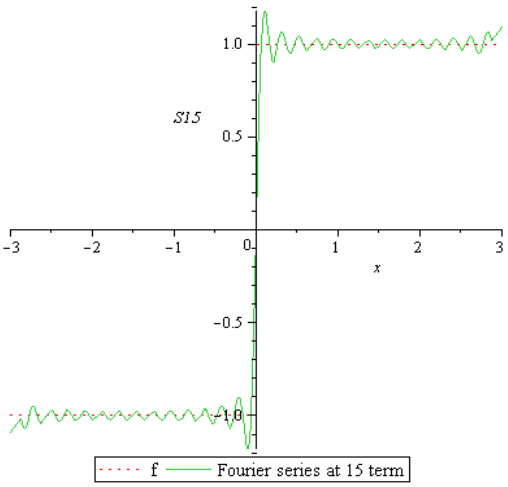
The Fourier sine series is:

$$f(x) = \sum_{n\,odd} \frac{4}{n\pi}\sin n\,x = \frac{4}{n\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \ldots\right)$$

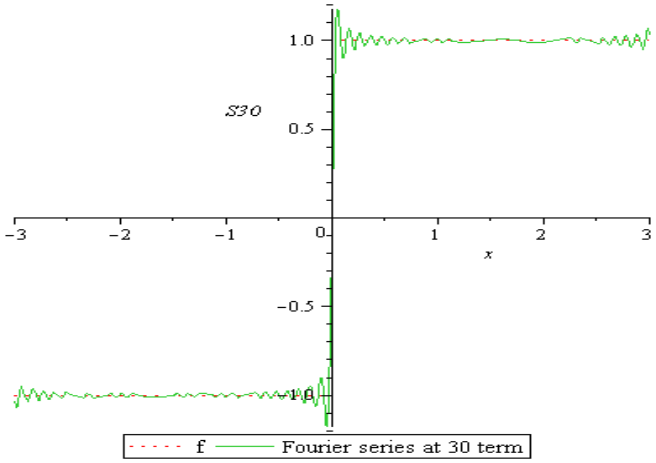
$$= \frac{4}{\pi}\sum_{n=1}^{\infty} \frac{1}{2n-1}\sin(2n-1)x \qquad (3.6)$$




(a)




(b)




(c)

**Figure** 3.1

Figures 3.1(b,c) show that the partial sums approximate $f$ well when $f$ is continuous, but that is not the case in the neighborhoods of points of discontinuities of $f$. This phenomenon is known as Gibb's phenomenon.

**Example 3.2:**

Consider the function

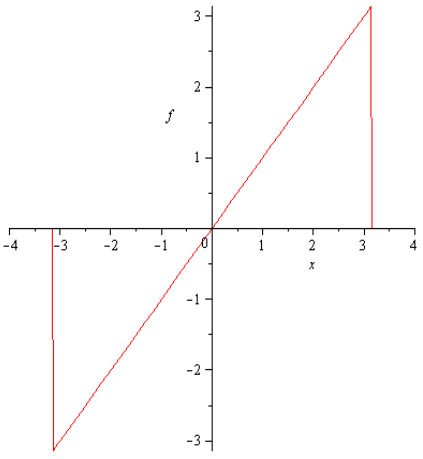
$$f(x) = x, \qquad x \in (-\pi, \pi)$$

The odd extension of the function $f$ is represented by a Fourier sine series with coefficients:

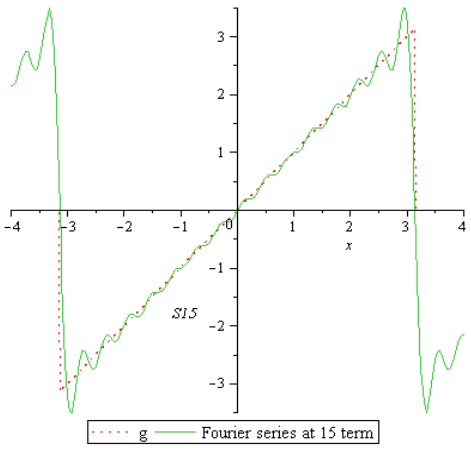
$$b_n = \frac{2}{\pi}\int_0^{\pi} f(x)\sin nx\, dx = = \frac{2}{\pi}\int_0^{\pi} x\sin n\,x\, dx = \frac{2}{n}(-1)^{n+1},$$

and the Fourier sine series is:

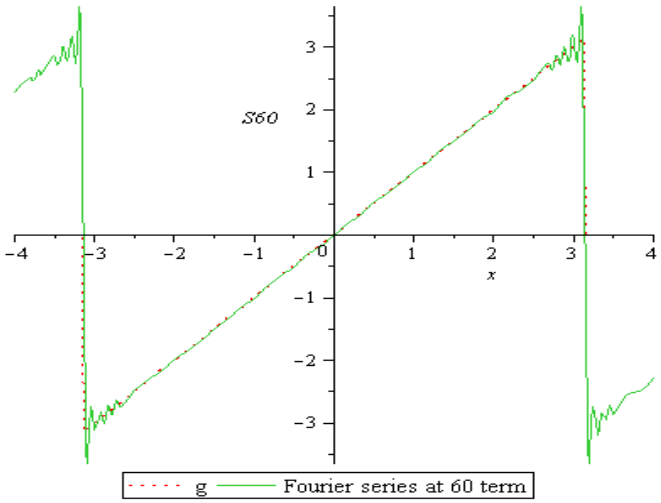
$$f(x) = \sum_{n=1}^{\infty}\frac{2}{n}(-1)^{n+1}\sin n\,x = 2\left(\sin x - \frac{1}{2}\sin 2x + \frac{1}{3}\sin 3x - \frac{1}{4}\sin 4x + \ldots\right) \qquad (3.7)$$




(a)




(b)




(c)

**Figure** 3.2

Figures 3.2(b,c) show that the partial sums approximate $f$ well, except around point where $f$ is not continuous.

**Example 3.3:**

Consider the function

$$f(x) = |x|, \qquad x \in [-\pi, \pi)$$

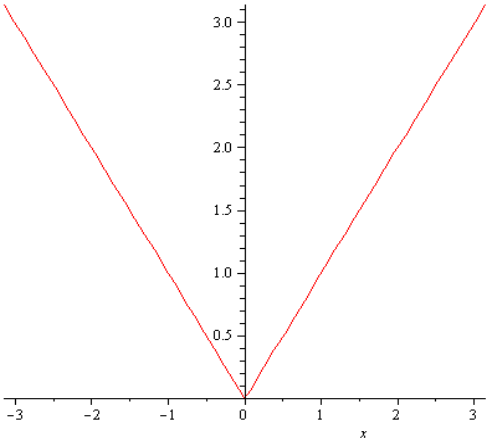
The even extension of the function $f$ is represented by a Fourier cosine series with coefficients:

$$a_0 = \frac{2}{\pi}\int_0^{\pi} x \, \mathrm{dx} = \pi$$

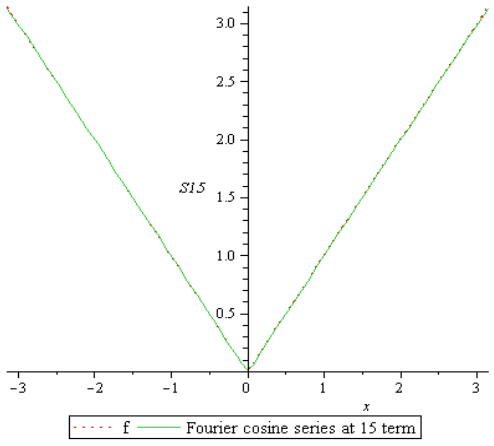
$$a_n = \frac{2}{\pi}\int_0^{\pi} f(x)\cos(nx)\, \mathrm{dx} = \frac{2}{\pi}\int_0^{\pi} x\cos(nx)\, \mathrm{dx} = \begin{cases} \dfrac{-4}{\pi n^2} & \text{if } n \text{ odd} \\ 0 & \text{if } n \text{ even} \end{cases}$$

and the Fourier cosine series is:

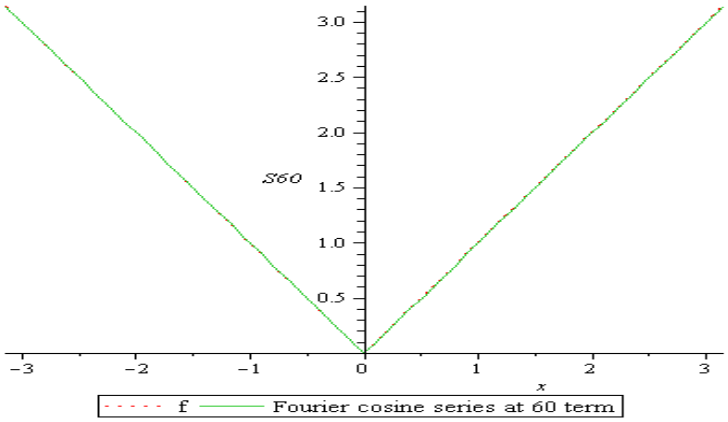
$$f = \frac{\pi}{2} - \sum_{n\,odd}^{\infty} \frac{4}{\pi n^2} \cos n x = \frac{\pi}{2} - \frac{4}{\pi}\left( \cos x + \frac{1}{9}\cos 3x + \frac{1}{25}\cos 5x + \ldots \right) \qquad (3.8)$$




(a)




(b)




(c)

**Figure** 3.3

Figure 3.3(b,c) show that the partial sums approtimates $f$ very well snice $f$ is continous.

## 3.2 Complex Form of Fourier Series

**Definition 3.4** [27]**:** Given $a > 0$ the set of functions

$$\left\{e^{2\pi i n x/a}\right\}_{n\in Z} \tag{3.9}$$

is called the period-$a$ trigonometric system.

**Remarks:**

- Euler's formula: $e^{ix} = \cos(x) + i\sin(x)$ implies that

$$e^{2\pi inx/a} = \cos(2\pi n x/a) + i\sin(2\pi n x/a)$$

This means that each element in trigonometric system has period-$a$.

- The period-$a$ trigonometric system is sometimes given in the form

$$\left\{1, \cos(2\pi n x/a), \sin(2\pi n x/a)\right\}_{n\in \mathbf{Z}} \tag{3.10}$$

Systems (3.9) and (3.10) can be obtained from each other by forming simple linear combinations. Specially, for $n \in \mathbf{Z}$,

$$e^{2\pi inx/a} = \begin{cases} \cos(2\pi n x/a) + i\sin(2\pi n x/a) & if \quad n \neq 0, \\ 1 & if \quad n = 0, \end{cases}$$

and for $n \in \mathbf{N}$,

$$\cos(2\pi n x/a) = \frac{e^{2\pi inx/a} + e^{-2\pi inx/a}}{2}$$

and

$$\sin(2\pi n x/a) = \frac{e^{2\pi inx/a} - e^{-2\pi inx/a}}{2i}$$

- A function that can be written as a finite linear combination of elements of the period-$a$ trigonometric system is called a period-$a$ trigonometric polynomial. That is, a trigonometric polynomial has the form

$$f(x)=\sum_{n=M}^{N} c(n)e^{2\pi i n x/a}$$

for some $M,N\in\mathbf{Z}$ and some coefficients $c(n)$.

**Theorem 3.3** [7]**:** The period-$a$ trigonometric system in 3.9 is orthogonal.

**Definition 3.5** [7]**:** Given an intagrable function $f(x)$ on $[0,a]$. Fourier series expansion of $f(x)$ is given by:

$$f(x)\sim\sum_{n\in\mathbf{Z}} c(n)e^{2\pi i n x/a} \tag{3.11}$$

where the coefficients $c(n)$ are defined as:

$$c(n)=\frac{1}{a}\int_0^a f(x)e^{-2\pi i n x/a}dx \quad \text{for} \quad n\in\mathbf{Z}. \tag{3.12}$$

provided that those integrals make sense. For example, if $f(x)$ is integrable on $[0\,,\,a]$, then the integral in (3.12) exists for each $n$.

**Remark:** The complex form of the Fourier series is just another form of the Fourier series. To show this, write

$$\begin{aligned}\sum_{n\in Z} c(n)e^{2\pi i n x/a} &= c(0)+\sum_{n\in\mathbf{N}} c(n)\left[\cos\left(\frac{2\pi n x}{a}\right)+i\sin\left(\frac{2\pi n x}{a}\right)\right]\\ &\quad+\sum_{n\in\mathbf{N}} c(-n)\left[\cos\left(\frac{2\pi n x}{a}\right)-i\sin\left(\frac{2\pi n x}{a}\right)\right]\end{aligned}$$

$$= \mathrm{c}(0) + \sum_{n\in\mathbf{N}} [c(n) + c(-n)] \cos\left(\frac{2\pi n x}{a}\right) + \sum_{n\in\mathbf{N}} i[c(n) - c(-n)] \sin\left(\frac{2\pi n x}{a}\right)$$

Conversely, a series of the form

$$a_0 + \sum_{n\in\mathbf{N}} a_n \cos(2\pi n x / a) + b_n \sin(2\pi n x / a)$$

Can be rewritten as

$$\sum_{n\in\mathbf{Z}} c(n) e^{2\pi i n x / a},$$

where

$$c(0) = a_0, \qquad c(n) = \frac{a_n - i b_n}{2}, n > 0, \qquad c(n) = \frac{a_{-n} + i b_{-n}}{2}, n < 0.$$

**Example 3.4:**

(a) Let $f(x)$ be the period-2 extension of the function $\chi_{[-1/2,1/2]}(x)$. The Fourier coefficients of $f(x)$ are:

$$c(n) = \frac{1}{2}\int_{-1}^{1} \chi_{[-1/2,1/2]}(x) e^{-2\pi i n x/2} dx$$

$$= \frac{1}{2}\int_{-1/2}^{1/2} e^{-\pi i n x} dx$$

$$c(n) = \frac{1}{2}\frac{2}{\pi n}\sin(\pi n/2)$$

$$= \begin{cases} 0 & \text{if } n \text{ is even}, n \neq 0, \\ \frac{1}{\pi n}(-1)^{(n-1)/2} & \text{if } n \text{ is odd}, \\ \frac{1}{2} & \text{if } n = 0. \end{cases}$$

The Fourier series associated to $f(x)$ is:

$$f(x) \sim \frac{1}{2} + \sum_{k=-\infty}^{\infty} \frac{1}{\pi(2k+1)} (-1)^k e^{\pi i(2k+1)x}$$

$$\sim \frac{1}{2} + \frac{2}{\pi}\cos(\pi x) - \frac{2}{3\pi}\cos(3\pi x) + \ldots$$

(b) Let $f(x)$ be the period-$\pi$ extension of the function $x \cdot \chi_{(0,\pi)}(x)$. Then

$$c(0) = (1/\pi)\int_0^{\pi} x dx = \frac{\pi}{2},$$

and for $n \neq 0$,

$$c(n) = \frac{1}{\pi}\int_0^{\pi} x\, e^{-2inx} dx = \frac{i}{2n}.$$

Therefore,

$$f(x) \sim \pi + \frac{i}{2}\sum_{n \in \mathbf{Z}} \frac{1}{n} e^{2inx} \sim \pi - \sum_{n \in \mathbf{N}} \frac{\sin(2nx)}{n}.$$

(c) Let $f(x)$ be the period-π extension of the function $x \cdot \chi_{(-\pi/2,\pi/2)}(x)$. Then

$$c(0) = \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} x\, dx = 0,$$

and for $n \neq 0$,

$$c(n) = \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} x\, e^{-2inx} dx = \frac{(-1)^n i\pi}{2n},$$

so that

$$f(x) \sim \frac{i}{2}\sum_{n \in \mathbf{Z}} \frac{(-1)^n}{n} e^{2inx} = -\sum_{n \in \mathbf{N}} (-1)^n \frac{\sin(2nx)}{n}.$$

(d) Let $f(x)$ be the period-2π extension of the function $|x| \cdot \chi_{-(\pi,\pi)}(x)$. Then

$$c(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |x| dx = \frac{\pi}{2},$$

and for $n \neq 0$,

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |x| \, e^{-2inx/2\pi} \, dx = \frac{1}{\pi} \int_{0}^{\pi} x \, e^{-inx} \, dx = \frac{\pi}{2} e^{-inx}$$

$$f(x) \sim \frac{\pi}{2} - \frac{4}{\pi} \sum_{n \in \mathbf{N}} \frac{1}{(2n+1)^2} \cos((2n+1)x).$$

### 3.3 Fourier Transform

Fourier series is a useful tool for representation and approximation of periodic functions via trigonometric functions. Non-periodic functions can be considered periodic with period $\infty$. Taking the limit as the period $L \to \infty$ in the Fourier series leads to the Fourier integral from which Fourier transform and its inverse are derived. Fourier transform, transforms a function (signal) from its time domain to its frequency domain. The inverse does the opposite.

**Theorem 3.4** [16]**:** If $f$ is continuously differentiable function with $\int_{-\infty}^{\infty} |f(t)| dt < \infty$, then the Fourier transform of $f$ is:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \,. \tag{3.13}$$

Its inverse is

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega \,. \tag{3.14}$$

**Example 3.5:**

The Fourier transform of the rectangular wave

$$f(t)=\begin{cases}1 & if\ -\pi \le t \le \pi \\ 0 & otherwise\end{cases}$$

is

$$\hat{f}(\omega)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$$

Using Euler's formula,

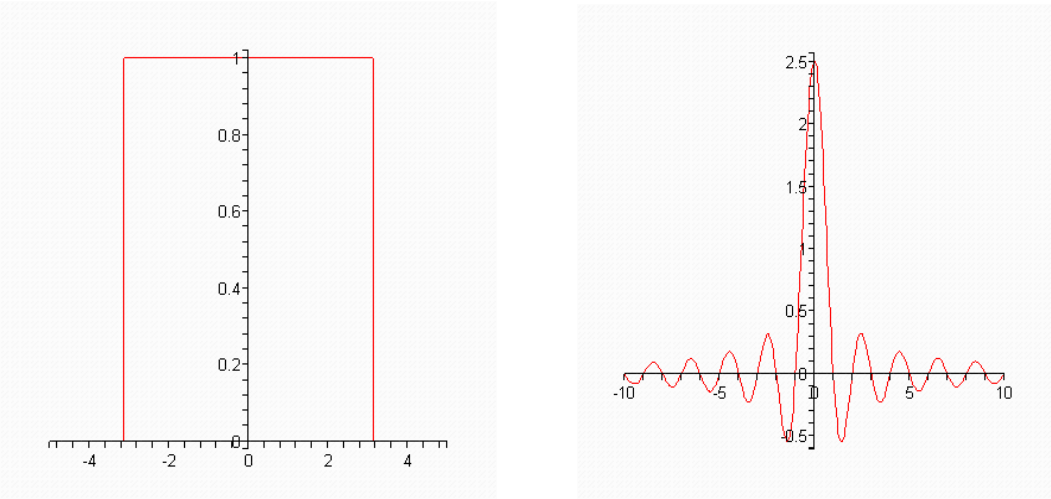
$$f(t)e^{-i\omega t}=f(t)(\cos\omega t - i\sin\omega t)$$

The product $f(t)\sin(\omega t)$ is odd since $f$ is even. Therefore the integral of the imaginary part of the product $f(t)\sin(\omega t)$ is zero and the Fourier transform of $f$ is reduced to

$$\hat{f}(\omega)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)\cos(\omega t)\,dt$$

$$=\frac{1}{\sqrt{2\pi}}\int_{-\pi}^{\pi}\cos(\omega t)dt$$

$$=\frac{\sqrt{2}\sin(\omega\pi)}{\sqrt{\pi}\omega}$$




(a) (b)

**Figure** 3.4

Figure 3.4(a) shows the function $f$ which is well-localized in time domain while figure 3.4(b) shows the Fourier transform $\hat{f}$ is not localized in the frequency domain, as the uncertainty principle asserts.

Fourier transform has a drawback; it shows the frequency content of a function $f$, but it doesn't specify where. i.e. the values of $t$ where the function has low and high frequencies. The reason for this is that Fourier transform uses the sine and cosine as a basis functions and they are infinitely long, so they pick up the different frequencies of $f$ without their location.

Computationally, we work on discrete signals not continuous ones. Therefore, a discrete version of the Fourier transform is needed. Fourier Transform (3.13) can be approximated by discrete Fourier transform (DFT). The function $f$ must be assumed zero outside some finite interval i.e.

$f(x) = 0$ for $|x| > A/2$. The Fourier transform of such a function with limited extent is given by [4]:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\omega x} dx = \int_{-A/2}^{A/2} f(x) e^{-i2\pi\omega x} dx,$$

This is the integral we wish to approximate numerically.

The interval of integration $[-A/2, A/2]$ is divided into to $N$ intervals of length $\Delta x = A/N$ assuming that $N$ is even, so the points are defined as $x_n = -N/2, ..., N/2.$ The set of points are

$$x_{-N/2} = \frac{-A}{2}, ..., x_0 = 0, ..., x_{N/2} = \frac{A}{2},$$

Assuming the function $f$ is known at these points. Letting the integral be

$$g(x) = f(x) e^{-i2\pi\omega x},$$

Applying the trapezoid rule to this integral we obtain

$$\int_{-A/2}^{A/2} g(x) dx = \frac{\Delta x}{2} \left\{ g\left(\frac{-A}{2}\right) + 2 \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}-1} g(x_n) + g\left(\frac{A}{2}\right) \right\}$$

With the assumption that $g$ is even, the trapezoid rule approximation may be written as:

$$\hat{f}(\omega) == \int_{A/2}^{A/2} g(x) dx \approx \Delta x \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} g(x_n) = \frac{A}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} f(x_n) e^{-i2\pi\omega x_n}$$

This approximation can be evaluated for any value of $\omega$.

For convenience, we adopt the notation

$$\omega_N = e^{i2\pi/N} = \cos\left(\frac{2\pi}{N}\right) + i\sin\left(\frac{2\pi}{N}\right),$$

$$\omega_N^{-nk} = e^{-i2\pi nk/N} = \cos\left(\frac{2\pi nk}{N}\right) - i\sin\left(\frac{2\pi nk}{N}\right)$$

and

$$\omega_N^{nk} = e^{i2\pi nk/N} = \cos\left(\frac{2\pi nk}{N}\right) + i\sin\left(\frac{2\pi nk}{N}\right)$$

The following definition uses the above notations.

**Definition 3.6** [4]**: Discrete Fourier Transform (DFT)**

Let $N$ be a positive integer and let $f_n$ be a sequence of $N$ complex numbers where $n = 0,1,...,N-1.$ Then its discrete Fourier transform is another sequence of $N$ complex numbers given by:

$$F_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n \omega_N^{-nk} \quad \text{for} \quad k = 0,1,...,N-1. \tag{3.15}$$

The output of the DFT $F_k$, is a complex-valued sequence. We will examine the real and imaginary parts of the DFT independently. These two sequences are denoted by $u\{F_k\}$ and $v\{F_k\}$ respectively, and they are defined as follows:

$$u\{F_k\} = \frac{1}{N}\sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} u\{f_n\}\cos\left(\frac{2\pi nk}{N}\right) + v\{f_n\}\sin\left(\frac{2\pi nk}{N}\right) \tag{3.16}$$

$$v\{F_k\} = \frac{1}{N}\sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} v\{f_n\}\cos\left(\frac{2\pi nk}{N}\right) - u\{f_n\}\sin\left(\frac{2\pi nk}{N}\right) \tag{3.17}$$

**Definition 3.7** [4]**: Inverse Discrete Fourier Transform (IDFT)**

Let $N$ be an even positive integer and let $F_k$ be a sequence of $N$ complex numbers where $k = \frac{-N}{2}+1,...,\frac{N}{2}$ . Then its inverse discrete Fourier transform is another sequencer of $N$ complex numbers given by

$$f_n = \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} F_k \omega_N^{nk} \qquad \text{for } n = \frac{-N}{2}+1,...,\frac{N}{2}. \tag{3.18}$$

If $N$ is odd positive integer and $F_k$ is a sequence of $N$ complex numbers, where $k = -\frac{N-1}{2},...,\frac{N-1}{2}$, then its inverse discrete Fourier transform is another sequence of $N$ complex numbers given by

$$f_n = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} F_k \omega_N^{nk} \quad \text{for } k = -\frac{N-1}{2},...,\frac{N-1}{2}. \tag{3.19}$$

**Example 3.6: Numerical evaluation of the FT and its inverse**

Consider $6$ points from the complex function

$$f = e^{(\frac{Ii}{3})}$$

The DFT and IDFT of these $6$ points can be computed using any math program such as (maple, matlab,…,etc.). Table 3.1 shows the input sequence $f_n$, the DFT $F_k$ and the IDFT $f_n$, respectively.

**Table** 3.1

| The 6 points | Discrete Fourier transform | Inverse Discrete Fourier transform | Error |
|---|---|---|---|
| 0.9450+0.3272I | 08143+1.904I | 0.945+0.3272I | $1\times10^{-16}$-$6\times10^{-17}$I |
| 0.7859+0.6184I | 0.1172-0.9762I | 0.7859+0.6134I | $1\times10^{-16}$-$2\times10^{-16}$I |
| 0.5403+0.8415I | 0.2672-0.3565I | 0.5403+0.8415I | 0-$1\times10^{-16}$I |
| 0.2352+0.9719I | 0.3203-0.137I | 0.2352+0.9719I | $8\times10^{-17}$I -$1\times10^{-16}$I |
| -0.0957+0.9954I | 0.364+0.0437I | -0.0957+0.9954I | -$6\times10^{-17}$+0I |
| -0.416+0.9093I | 0.4317+0.3235I | -0.416+0.9093I | -$2\times10^{-16}$-$2\times10^{-16}$I |

### 3.4 Fast Fourier Transform (FFT)

The fast Fourier transform is a well-known efficient algorithm that reduces the number of iterations in approximating a function. i.e. it reduces the cost of calculating the coefficients $\{c_n\}_{n=0}^{N}$ when $N$ is large. Also, FFT is a set of methods all designed to compute the various forms of the DFT efficiently. FFT uses many methods but we will focus on splitting methods.

### 3.4.1 Splitting Method [4]

The goal of the FFT is to efficiently compute the DFT of a sequence $f_n$ which has length $N$ and is assumed to have a period of length $N$.

Using definition (3.15) , we get:

$$F_k = \sum_{n=0}^{N-1} f_n \omega_N^{-nk} \qquad \text{for } k = 0,1,\ldots, N-1.$$

where $\omega_N = e^{i2\pi/N}$.

Lets begin with the case $N = 2^M$ where $M$ is a natural number. Now split $x_n$ into its even and odd subsequences by letting $y_n = f_{2n}$ and $z_n = f_{2n+1}$, then formula (3.15) becomes:

$$F_k = \sum_{n=0}^{\frac{N}{2}-1} y_n \omega_N^{-2nk} + z_n \omega_N^{-(2n+1)k} \qquad \text{for } k = 0,\ldots, N-1. \tag{3.20}$$

Also, $e^{-i4\pi nk/N} = e^{-i2\pi nk/(N/2)}$ is equivalent to $\omega_N^{-2nk} = \omega_{N/2}^{-nk}$ .

So formula (3.20) can be written as:

$$F_k = Y_k + \omega_N^{-k} Z_k$$

where

$$Y_k = \sum_{n=0}^{\frac{N}{2}-1} y_n \omega_{N/2}^{-nk} \text{ is the DFT of the sequence } y_n,$$

and

$$Z_k = \sum_{n=0}^{\frac{N}{2}-1} z_n \omega_{N/2}^{-nk} \text{ is the DFT of the sequence } z_n.$$

Each $Y_k$ and $Z_k$ is a DFT of length $N/2$ or half-length DFT with period $N/2$. we conclude:

$$F_{k+\frac{N}{2}} = Y_{k+\frac{N}{2}} + \omega_N^{-\left(k+\frac{N}{2}\right)} Z_{k+\frac{N}{2}} \qquad \text{where} \quad k = 0,\ldots,\frac{N}{2}-1.$$

Notice that $\omega_N^{-N/2} = -1$ and the subsequences $Y_k$ and $Z_k$ have a period of $N/2$, we conclude:

$$F_k = Y_k + \omega_N^{-k} Z_k$$

$$F_{k+\frac{N}{2}} = Y_k - \omega_N^{-k} Z_k \qquad \text{where} \quad k = 0,..., \frac{N}{2}.$$

The idea of splitting method is to express the DFT sequence of length $N$ as a combination of two subsequences both of length $N/2$ and we repeat the splitting method again on each subsequence. A full FFT algorithm results when the splitting idea is applied to the computation of $Y_k$ and $Z_k$. Finally, the problem of computing a DFT of length $N$ has been replaced by computing $N$ DFTs of length 1. At this point there are really no DFTs left to be done, since the DFT of a sequence of length 1 is itself.

### 3.5 Two-Dimensional Fourier Transforms

The discrete Fourier transform (DFT) of an image is a representation of an image as a double sum of complex exponentials of varying magnitudes, frequencies, and phases. The Fourier transform plays a critical role in a broad range of image processing applications, including analysis, denoising, and compression.

There are two important reasons for using DFT:

- The input and the output of DFT are both discrete, which makes it convenient for computer manipulations.

- Fast Fourier Transform FFT is a fast algorithm for computing the DFT.

The DFT is usually defined for a discrete function $f(m,n)$ that is nonzero over the finite region $0 \le m \le M-1$ and $0 \le n \le N-1$. The two-dimensional $M \times N$ Discrete Fourier transform of $f(m,n)$ is defined by:

$$F(\omega_1,\omega_2) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n) e^{-j(2\pi/M)\omega_1 m} e^{-j(2\pi/N)\omega_2 n},$$

where

$$\omega_1 = 0,1,...,M-1,$$
$$\omega_2 = 0,1,...,N-1.$$

The variables $\omega_1$ and $\omega_2$ are frequency variables; their units are radians per sample. $F(\omega_1,\omega_2)$ is often called the frequency-domain representation of $f(m,n)$. $F(\omega_1,\omega_2)$ is a complex-valued function that is periodic both in $\omega_1$ and $\omega_2$, with period $2\pi$. Because of the periodicity, only the range $-\pi \le \omega_1,\omega_2 \le \pi$ is displayed.

The inverse $N \times M$ two-dimensional Fourier transform is given by:

$$f(m,n) = \frac{1}{MN}\sum_{\omega_1=0}^{M-1}\sum_{\omega_2=0}^{N-1} F(\omega_1,\omega_2) e^{j(2\pi/M)\omega_1 m} e^{j(2\pi/N)\omega_2 n},$$

where $m = 0,1,...,M-1$ and $n = 0,1,...,N-1$.

This equation means that $f(m,n)$ can be represented as a sum of an infinite number of complex exponentials (sinusoids) with different

frequencies. The magnitude and phase of the contribution at the frequencies $(\omega_1, \omega_2)$ are given by $F(\omega_1, \omega_2)$.

### 3.6 Fourier Image Compression

Imagine that we want to scan a white and black photograph line by line. We can assume that the image has an infinite resolution. The ideal scan would be an infinite sequence of numbers that can be considered the values of a continuous light intensity function I(t). But, in practice we can store a finite amount of data in memory, so we have to choose a finite number of values I(1) through I(n). This process is known as sampling.

Sampling seems tradeoff between quality and price. The bigger the sample the better the quality of the final image, but requires more memory and higher screen resolution, resulting in higher costs. However, this conclusion is not entirely true. Sampling theory says that we can sample an image and reconstruct it later in memory without loss of quality if we can do the following:

1. Transform the infinity function from the time domain $I(t)$ to frequency domain $G(f)$.

2. Find the maximum frequency $f_m$.

3. Sample $I(t)$ at a rate slightly higher than $2f_m$. For example if

   $f_m =$ 22000 Hz, samples are generated at the rate of 44100 Hz .

4. Store the sampled values in the bitmap. The resulting image would be equal in quality to the original quality on the photograph.

There are two points to consider:

The first point is that $f_m$ could be infinite. The value of $f_m$ should be selected so that the frequencies that are greater than $f_m$ are not contributed i.e. have low amplitude. But, in this case there would be lose in quality.

The second point is that the bitmap (the resolution) may be too small for a generated sample (in step 3). So, we have to choose a smaller sample and still there would be a loss in image quality.

The previous result was proved by Harry Nyquist, and $2f_m$ is called Nyquist rate. Nyquist rate is being used in practical situations. For example the range of human hearing is between 16 Hz and 22000 Hz. So, when the sound is digitized at a high quality, it sampled at the rate of 44000Hz. Any thing lower than that result would be distortion. Note that Nyquist rate is the difference between the maximum and the minimum frequencies, the bandwidth of the signal.

Notice that once we take samples of a signal we may loose the behavior of the signal between samples, and we may miss some important information. But, in practice all analog signals have limited frequency response because they are created by sources like mouth, microphone or speaker whose response speed is limited, so the changing in real signal is so limited. We can say that finite bandwidth of real signals is what makes their digitizing possible. In the next two examples, the software FAWAV [26] is used.

**Example 3.7:**

(a)

(b)

(c)

(d)

(e)

(f )

**Figure** 3.5

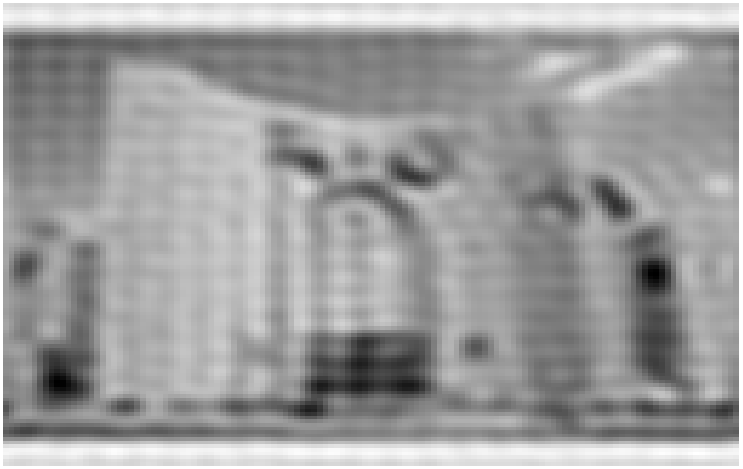
Figure 3.5 (a) represents the original image and figures 3.5 (b,c,d,e,f ) represent the Fourier series compressed images at 10, 20, 30, 40,50 terms

respectively. Its clear that as the number of terms of the Fourier series increases the resolution of the reconstructed image increases.

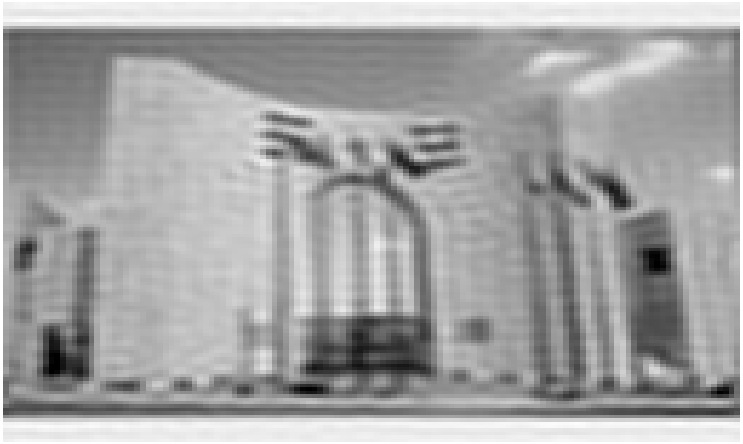
**Exapmle 3.8:**

Figure 3.6 (a) represents $512 \times 512$ image for An-Najah national university, Nablus, Palestine. The image type is a bitmap and its size is 257 KB. Now we will compress this image using Fourier series using different reolutions (different terms).

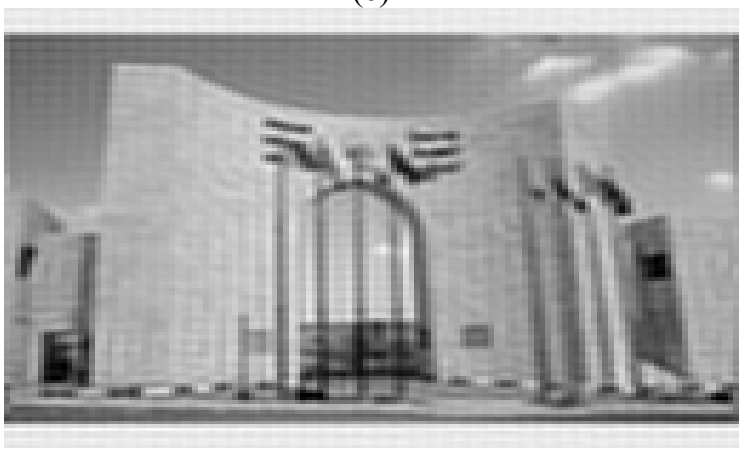
(a)



(b)



(c)



(d)

(e)

**Figure** 3.6

Figures 3.6 (b,c,d,e) represent the Fourier series compressed images at 20, 30, 40,50 terms respectively. Its clear that the resolution is proportional to the number of terms used in the partial sum of Fourier series.

# Chapter Four

# Wavelets

## 4.1 Wavelet and Signal Processing

## 4.2 Wavelet Transform

### 4.2.1 The Continuous Wavelet Transform and Its Inverse

### 4.2.2 Wavelet Series

## 4.3 Haar Transform

### 4.3.1 First-level Haar Transform

### 4.3.2 Conservation and Compaction of Energy

### 4.3.3 Haar Transform Multiple Levels

### 4.3.4 Haar Wavelet Transform Via Scalar Product

### 4.3.5 Haar Wavelet Inverse Transform

### 4.3.6 Multiresolution Analysis, Multiple Levels

# Chapter Four

## Wavelets

In Fourier series, the orthogonal system $\{\sin nx, \cos nx\}_{n=0}^{\infty}$ is used to represent a function. In wavelet series, another orthogonal system is used. This system consists of functions generated, by scaling and dilation, from one function called mother wavelet. Wavelets are functions that satisfy certain mathematical requirements and are used in analyzing data or other functions. Wavelets cut up data into different frequency components, and then study each component with a resolution matched to its scale. They have advantages over traditional Fourier methods in analyzing functions.

We can compare wavelet analysis to viewing an object from a distance, then closer, then through a microscope. When we look from a distance, we see the whole shape of the object, but not the small details. Looking closer, we can see more details. When we look through a microscope, we see small details, but not the whole shape. This is why it is important to analyze in different scales. In short, when we change the scale of the wavelet, we get new information about the function being analyzed.

Wavelet analysis procedure is to adapt a wavelet prototype function, called an analyzing wavelet or mother wavelet. Temporal analysis is performed with a contracted, high frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low-frequency version of the same wavelet. Representing a signal or a function

by a wavelet expansion enables us to analyze the signal using the coefficients in the corresponding series expansion. In this series, the terms with small coefficients contribute less to the sum. Therefore we can neglect terms with coefficients of magnitude less than a specific value called threshold. Storing coefficients of magnitude greater than a given threshold leads to less storage space; this is the main idea of data compression.

Some applied fields that makes use of wavelets include astronomy, nuclear engineering, signal and image processing, neurophysiology, music, speech recognition, image compression, human vision and solving partial differential equations.

## 4.1 Wavelets and Signal Processing

We start this section by an example that illustrates the use of wavelet in signal processing. The example transforms a discrete signal using running averages and differences. The result is a transformed (new) signal consists of averages and differences. Grouping the averages together results in a signal of averages called the trend subsignal whose length is half length of the original signal. Similarly, the signal of differences is called the fluctuation subsignal whose length is half the length of the original signal. The trend subsignal represents (approximates) the original signal while the fluctuation signal represents the details in the signal.

**Example 4.1:**

Let $\{x_i\}_{i=0}^{n}$ for some even non-negative integer. For each $i = 0,1,...\frac{n-1}{2}$, let

$$\hat{x}_{2i} = \frac{x_{2i} + x_{2i+1}}{2} \text{ and } \hat{x}_{2i+1} = x_{2i} - \hat{x}_{2i} = \frac{x_{2i} - x_{2i+1}}{2} \tag{4.1}$$

The first subsignal in (4.1) is the trend (scale) subsignal and it consists of running averages. The second subsignal in (4.1) is the detail (fluctuation) subsignal and it consists of running differences. Equation (4.1) represents a discrete transform. The inverse transform of (4.1) is easily found as:

$$x_{2i} = \hat{x}_{2i} + \hat{x}_{2i+1} \text{ and } x_{2i+1} = \hat{x}_{2i} - \hat{x}_{2i+1} \tag{4.2}$$

Consider the following sequence of numbers which represent a discrete signal:

56 40 8 24 48 48 40 16

Applying the transform in (4.1), we get:

First pair**:** $\frac{56+40}{2} = \frac{96}{2} = 48$ and $56 - 48 = 8$

Second pair: $\frac{8+24}{2} = \frac{32}{2} = 16$ and $8 - 16 = -8$

Third pair: $\frac{48+48}{2} = 48$ and $48 - 48 = 0$

The last pair: $\frac{40+16}{2} = \frac{56}{2} = 28$ and $40 - 28 = 12$

The first row of the following table shows the original signal while the second row shows the transformed signal using running averages and

differences, see equation (4.1). The first 4 entries of the second row show the trend subsignal and the last 4 entries of the same row show the detail subsignal. Denote the trend signal by $\mathbf{a}^1$ and the detail signal by $\mathbf{d}^1$. The following table shows the original signal and its transformation.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Original signal | 56 | 40 | 8 | 24 | 48 | 48 | 40 | 16 |
| $<\mathbf{a}^1 \mid \mathbf{d}^1>$ | 48 | 16 | 48 | 28 | 8 | -8 | 0 | 12 |

The original signal in the first row can be reconstructed from the transformed signal in the second row by the inversion formulas in equation (4.2). For example, $48+8=56$, $48-8=40$, $16+(-8)=8$, $16-(-8)=24$, and so on.

The original signal is approximated by the trend subsignal whose length is half the length of the original subsignal. Taking the transform (4.1) on the trend subsignal will reduce the length of the second trend subsignal by one half which means one fourth of the original signal.

Applying equation (4.1) on the first trend subsignal and leaving the first details subsignal untouched implies.

$$\frac{48+16}{2}=\frac{64}{2}=32 \quad , \quad \frac{48-16}{2}=\frac{32}{2}=16$$

And 48, 28 will be replaced by:

$$\frac{48+28}{2}=\frac{76}{2}=38 \quad , \quad \frac{48-28}{2}=\frac{20}{2}=10$$

Denoting the 2-level trend signal by $\mathbf{a}^2$ and the detail signal by $\mathbf{d}^2$ We get the following table:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Original signal | 56 | 40 | 8 | 24 | 48 | 48 | 40 | 16 |
| $<\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1>$ | 32 | 38 | 16 | 10 | 8 | -8 | 0 | 12 |

These eight numbers also represent the original information, but if we invert the above procedure twice, then we go back to the original sequence. However, we must be careful and keep track of numbers we calculated as averages and the ones that are calculated as differences.

Finally, we repeat the process on 2-level trend subsignal leaving the fluctuation subsignal untouched. The numbers 16, 10 are not changed since they are differences, but we replace the averages 32, 38 by their average and difference, so, we replace 32 by $\frac{32+38}{2}=\frac{70}{2}=35$ and we replace 38 by $\frac{32-38}{2}=\frac{-6}{2}=-3$, so, we obtain the table:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Original signal | 56 | 40 | 8 | 24 | 48 | 48 | 40 | 16 |
| $<\mathbf{a}^3 \mid \mathbf{d}^3 \mid \mathbf{d}^2 \mid \mathbf{d}^1>$ | 35 | -3 | 16 | 10 | 8 | -8 | 0 | 12 |

Notice that transformation (4.1) changes the signal but preserves the information in it. The information can be recovered through the inversion formula (4.2). The new signal obtained after three transformations is of the same length. However, the third trend subsignal has a length of one eighth of the original signal. The approximation of the signal by the third trend is not accurate since it consists of only one value. But when the original signal consists of large number of values, as in sampling audio signals which is

measured by kilo-hertz (Khz), the approximation by the third and higher level trend signals is acceptable. The detail signal usually consists of values that are relatively much smaller than the values of the trend subsignal. By looking at the details, we see that some details can be used to improve the accuracy. This is done by replacing numbers that are numerically smaller than a certain fixed number (threshold) by zeros. For instance, if we remove all numbers which numerically are smaller than 4 from the transformed signal $< \mathbf{a}^3 \mid \mathbf{d}^3 \mid \mathbf{d}^2 \mid \mathbf{d}^1 >$ , we obtain the following table:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 35 | 0 | 16 | 10 | 8 | -8 | 0 | 12 |

To reconstruct the signal, we apply the inverse formula (4.2). Based on these numbers we will perform the inversion formula as follows: first, $< \mathbf{a}^3 \mid \mathbf{d}^3 > \xrightarrow{formula\,4.2} < \mathbf{a}^2 >$, then $< \mathbf{a}^2 \mid \mathbf{d}^2 > \xrightarrow{formula\,4.2} < \mathbf{a}^1 >$, and finally, $< \mathbf{a}^1 \mid \mathbf{d}^1 > \xrightarrow{formula\,4.2}$ original signal

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 35 | 8 | 16 | - 8 | 0 | 0 | 10 | 12 |
| 35 | 8 | 16 | -8 | 35 | 0 | 10 | 12 |
| 51 | 8 | 19 | -8 | 45 | 0 | 25 | 12 |
| 59 | 43 | 11 | 27 | 45 | 45 | 37 | 13 |

Here the four sequences explain the reconstruction after thresholding which gives an approximation of the original signal. The first row of the

following table shows the reconstructed signal from the transformed sequence with threshold 4. The second row shows the original signal.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 59 | 43 | 11 | 27 | 45 | 45 | 37 | 13 |
| 56 | 40 | 8 | 24 | 48 | 48 | 40 | 16 |

Repeating the thresholding and reconstruction with a threshold of 9, we get the following table with the first row for the reconstructed signal and the second row for the original one. The following table show the original sequence and the reconstructed one after thresholding by 9.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 51 | 51 | 19 | 19 | 45 | 45 | 37 | 13 |
| 56 | 40 | 8 | 24 | 48 | 48 | 40 | 16 |

Figure 4.1 shows the plot of the original signal together with the reconstructed ones.

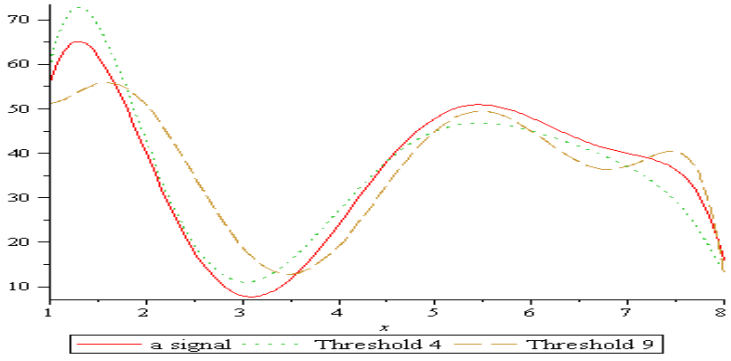



**Figure** 4.1

The energy of the original signal is 11840 while the energy for the reconstructed signal after thresholding by 4 is 11786 or 99.544% of the

energy of the original signal. The energy for the reconstructed signal after thresholding by 9 is 11521 or 97.306% of the energy in the original signal.

The accuracy in the reconstructed sequence depends on the choice of the threshold. After applying a threshold on the transformed version of the given sequence, large part of the signal consists of zeros. For instance, in example 4.1 with thresholding at 9, we store the four numbers 35, 16, 10, 12 instead of the original eight numbers, and still we are able to recover about 97% of the original signal. This is the key of lossy compression.

## 4.2 Wavelet Transform

Just as with the Fourier transform, there are three types of wavelet transforms; The Continuous Wavelet Transform (CWT), the Discrete Wavelet Transform (DWT), and the Wavelet series. The CWT transforms a continuous signal to a continuous signal. The DWT transforms the discrete signal to a discrete signal. The wavelet series transforms a continuous signal to a discrete signal.

### 4.2.1 The Continuous Wavelet Transform and Its Inverse

The continuous wavelet transform of a function $f$ depends on a mother wavelet $\psi$ . The mother wavelet can be any real or complex continuous function that satisfies the following properties:

1. The total area under the curve of the function is zero, i.e. $\int_{-\infty}^{\infty} \psi(t)\, dt = 0$ suggesting a function that oscillates above and bellow the t-axis. Such a function tends to have a wavy appearance.

2. $\psi(t)$ is square integrable which means it has a finite energy.

   This implies that the energy of the function is finite, suggesting that the function is localized in some finite interval and zero or almost zero outside this interval. These properties justify the name "wavelet". There are many known functions that satisfy these conditions, and some of them have been researched and are commonly used for wavelet transforms.

3. The admissibility conditions which are:

   - $\psi$ is smooth, maybe infinitely differentiable.
   - $\psi$ is in computationally convenient form.
   - $\psi$ has compact support, i.e. all functions values are zero outside a certain bounded interval.

**Example 4.2:**

Some of well-known mother wavelets are:

1. The Morlet wavelet, shown in figure 4.2, is defined as:

$$\psi(t) = e^{-t^2} \cos\left(\pi t \sqrt{\frac{2}{\ln 2}}\right) \approx e^{-t^2} \cos(2.885\pi t).$$

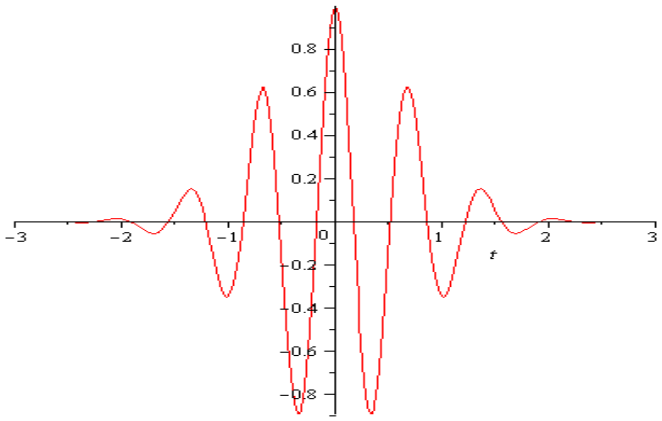



**Figure** 4.2

This cosine curve has oscillations dampened by the exponential (Gaussian) factor. More than 99% of its energy is concentrated in the interval $-2.5 \le t \le 2.5$.

2. The Mexican Hat wavelet, shown in figure 4.3, is defined as:

$$\psi(t) = (1 - 2t^2)e^{-t^2}$$

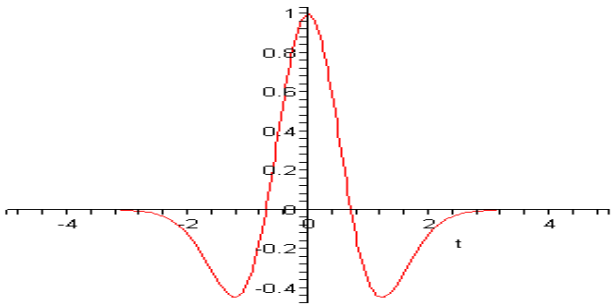



**Figure** 4.3

This is the second derivative of the (negative) Gaussian function $-0.5e^{-t^2}$.

More than 99% of its energy is concentrated in the interval $-2 \le t \le 2$.

**Definition 4.1** [20]**:** Once a wavelet $\psi(t)$ has been chosen, the Continuous Wavelet Transform (CWT) of a square integrable function $f$ is defined as**:**

$$W(a,b) = \int_{-\infty}^{\infty} f(t)\overline{\psi}_{a,b}(t)\, dt \tag{4.3}$$

where $\psi_{a,b} = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$ is the function generated from the mother wavelet $\psi$ by scaling and translation and $\overline{\psi}$ is the complex conjugate of $\psi$.

The quantity $1/\sqrt{|a|}$ is the normalizing factor that ensures that the energy of $\psi(t)$ remains independent of the factors $a$ and $b$, i.e.

$$\int_{-\infty}^{\infty} |\psi_{a,b}(t)|^2\, dt = \int_{-\infty}^{\infty} |\psi(t)|^2\, dt$$

**Example 4.3:**

Consider the Mexican hat wavelet

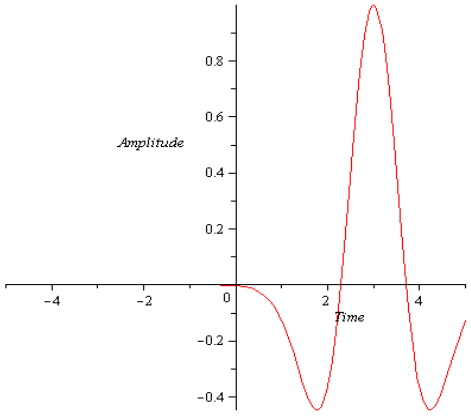
$$\psi(t) = (1-2t^2)e^{-t^2}$$

Also

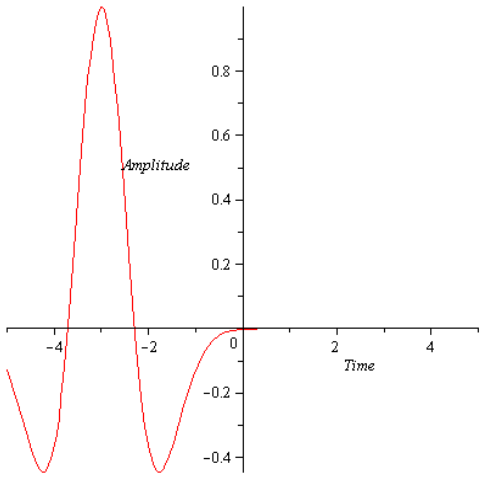
$$\psi_{a,b} = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) = \frac{1}{\sqrt{a}}\left(1-2\left(\frac{t-b}{a}\right)^2\right)e^{-\left(\frac{t-b}{a}\right)^2}$$

Fixing $a=1$ and the translating factor $b$ could be any integer,

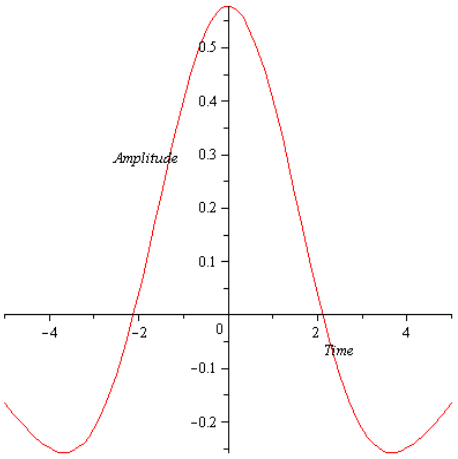
$$\psi_{1,b} = \psi(t-b) = \left(1-2(t-b)^2\right)e^{-(t-b)^2},$$



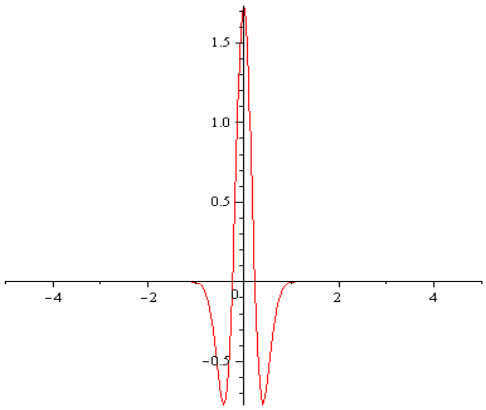

(a) $\psi_{1,3} = \left(1-2(t-3)^2\right)e^{-(t-3)^2}$




(b) $\psi_{1,-3} = \left(1-2(t+3)^2\right)e^{-(t+3)^2}$




(c) $\psi_{3,0} = \frac{1}{\sqrt{3}}\left(1-2\left(\frac{t}{3}\right)^2\right)e^{-\left(\frac{t}{3}\right)^2}$



(d) $\psi_{1/3,0} = \sqrt{3}\left(1-2(3t)^2\right)e^{-(3t)^2}$

**Figure** 4.4

Figure 4.4 (a,b) is the graph of the Mexican hat wavelet where $a=1$ and $b=3$, $a=1$ and $b=\frac{1}{3}$ respectively.

Also, fixing $b=0$ and the scaling factor (or dilation) parameter $a \in \mathbf{R}$ and $a \neq 0$,

$$\psi_{a,0} = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t}{a}\right) = \frac{1}{\sqrt{|a|}}\left(1-2\left(\frac{t}{a}\right)^2\right)e^{-\left(\frac{t}{a}\right)^2}$$

Notice, the scaling factor $a>1$ stretch the wave and $0<a<1$ shrink the wave as we can see in figures 4.4 (c,d) respectively.

**Example 4.4:**

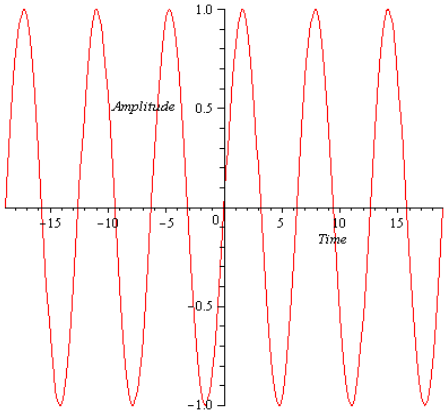
Consider the signal

$$f(t)=\sin(t) \qquad t\in[-2\pi,2\pi]$$

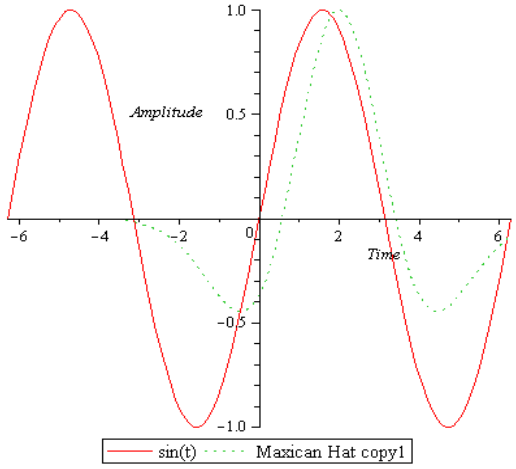
Also, consider two translated copies of the Mexican Hat wavelet which are:

$$\psi_1(t)=\left(1-2\left(\frac{1}{2}t-1\right)^2\right)e^{-\left(\frac{1}{2}t-1\right)^2}$$

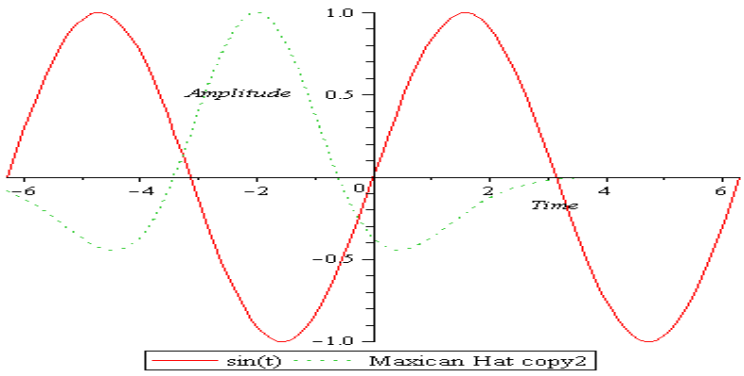
$$\psi_2(t)=\left(1-2\left(\frac{1}{2}t+1\right)^2\right)e^{-\left(\frac{1}{2}t+1\right)^2}$$




(a) $f(t)=\sin(t)$




(b)




(c)

**Figure** 4.5

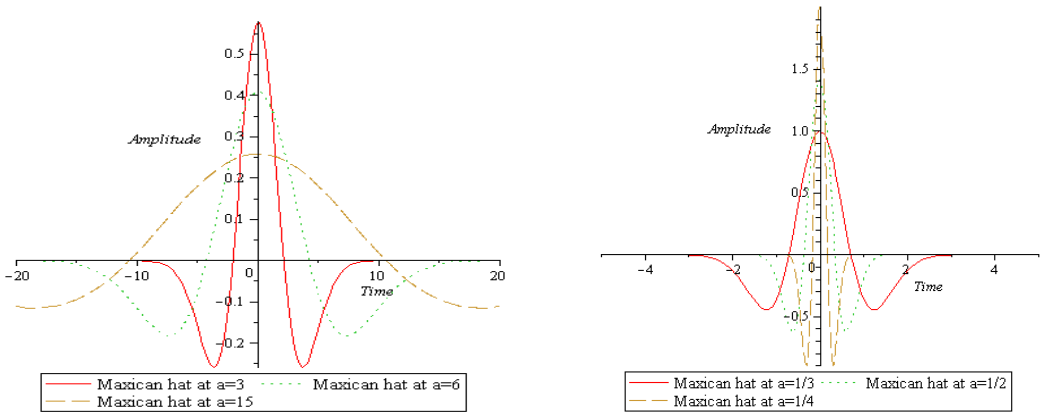
Figure 4.5 (b) is positioned at a point where the sine wave has a maximum. At this point there is a good match between the function being analyzed (the sine), and the wavelet. The wavelet replicated the features of the sine wave. As a result, the inner product of the sine and the wavelet is a large positive number. In contrast, figure 4.5 (c) is positioned where the sine wavelet has minimum. At that point the wave and the wavelet are almost mirror images of each other. Where the sine wave is positive, the wavelet is negative and vice versa. The product of the sine and the wavelet at this point is negative, and the CWT, becomes a large negative number. Between points –1 and 1 the CWT drops from positive to zero to negative.

As the wavelet translated, from left to right, along the sine wave, the CWT alternates between positive and negative values producing small waves.

This shape shows the close match between the function being analyzed (the sine wave) and the analyzing wavelet. We notice that they both have similar shapes and similar frequencies.

**Example 4.5:**

Let's extend the analysis to cover different frequencies. This is done by scaling the wavelet. Figure 4.6 (a) shows what happens when the wavelet stretched




(a) (b)

**Figure** 4.6

such that it covers several periods of the sine wave. Translating the wavelet from left to right doesn't affect the match between the wavelet and the sine wave by much. So, as the result the CWT varies just a little. The wider the wavelet, the closer the CWT to a constant. Notice how the amplitude of the wavelet has been reduced, and so the area also reduced and produced a small constant. Similar thing happens in figure 4.6 (b) where the wavelet

has shrinked and is much narrower than only cycle of the sine wave. Since the wavelet is so thin the inner product of it and the sine wave is always a small number (positive or negative) regardless of the precise position of the wavelet relative to the sine wave. We conclude that translating the wavelet does not affect its match to the sine wave. The result is a CWT that is close to a constant.

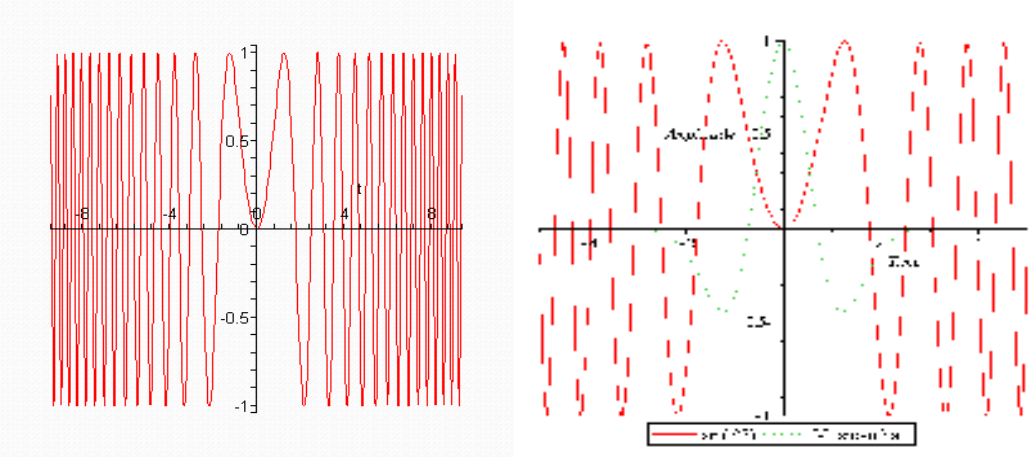
It is clear that the quality of the match depends on the choice of wavelet. If the wavelet is very different from $f(t)$ at any frequencies and any times, the values of the resulting $W(a,b)$ will all be small and will not exhibit much variation. As a result, different wavelets should be selected to compress different image types (*bi*-level, continuous-tone, and discrete-tone), but the precise choice of wavelet is still the subject of much research.

In example 4.5, the CWT of the sine function can be evaluated. However, in general this is impossible, either in practice or in principle, and the calculations have to be done numerically.

The next example is slightly more complex and leads to a better understanding of the CWT.

**Example 4.6:**

Consider analyzing $f(t) = \sin(t^2)$ using the Mexican hat wavelet, whose graph is given in figure 4.7




(a) $f(t) = \sin(t^2)$ (b)

**Figure** 4.7

Wavelets are subject to uncertainty principle. The Haar wavelet is very well localized in the time domain. In contrast, the Mexican hat wavelet and especially the Morlet wavelet is localized in frequency domain, but are spread over time. An important result of the uncertainty principle is that it is impossible to achieve a complete simultaneous mapping of both time and frequency. Wavelets provides near an optimal solution for this problem, and this is one of the features that makes them superior to Fourier analysis.

**Definition 4.2** [20]**: Inverse Wavelet Transform**

Let $\Psi(\omega)$ be the Fourier transform of the $\psi(t)$:

$$\Psi(\omega) = \int_{-\infty}^{\infty} \psi(t) e^{-i\omega t}\, dt\,.$$

If $W(a,b)$ is a CWT of a function $f(t)$ with a wavelet $\psi(t)$, then the inverse CWT is defined by:

$$f(t) = \frac{1}{C}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{|a|^2}W(a,b)\psi_{a,b}(t)\,da\,db,$$

Where the quantity $C$ depends on the Fourier transform of $\Psi(\omega)$ and is defined as

$$C = \int_{-\infty}^{\infty}\frac{|\Psi(\omega)|^2}{|\omega|}\,d\omega.$$

The inverse CWT exists if $C$ is positive and finite.

### 4.2.2 Wavelet Series

The series expansion of a function $f$ in terms of a given set of simple functions $\{f_n\}_{n=0}^{\infty}$ which satisfy certain properties is:

$$f(x) = \sum_{n=0}^{\infty} a_n f_n(x) \tag{4.4}$$

for some coefficients $a_n$. The function $f$ and the set $\{f_n\}_{n=0}^{\infty}$ must satisfy certain conditions that enable us finding the coefficients $a_n$. For example, Taylor series expands a function that is analytic at $x_0$ using the orthogonal system $\{(x-x_0)^n\}_{n=0}^{\infty}$ while Fourier series expands a square integrable function using the orthogonal trigonometric system $\{\sin nx, \cos nx\}_{n=0}^{\infty}$. It is the orthogonality of such systems that enables us to compute the coefficients of the corresponding series.

**Definition 4.3** [7]**:** Let $\psi$ be a function that is defined on $\mathbf{R}$, smooth, and has compact support. For each $j,k \in \mathbf{Z}$, define a function $\psi_{j,k}$ by

$$\psi_{j,k}(x) = 2^{j/2}\,\psi\left(2^{j}x - k\right), \quad x \in \mathbf{R}.$$

The function $\psi_{j,k}$ is a scaling by $j$ units and a translation by $k$ units of $\psi$. The function $\psi$ is called the mother wavelet.

**Example 4.7:** Consider the mother wavelet $\psi(x) = e^{-x^2}$. We use scaling and translations to generate the wavelets

$$\psi_{j,k}(x) = 2^{j/2}\,e^{-(2^{j}x-k)^2}, \quad x \in \mathbf{R},\ \mathrm{j,k} \in \mathbf{Z}.$$

To understand the scaling role of $j$, fix $k = 0$.

Then

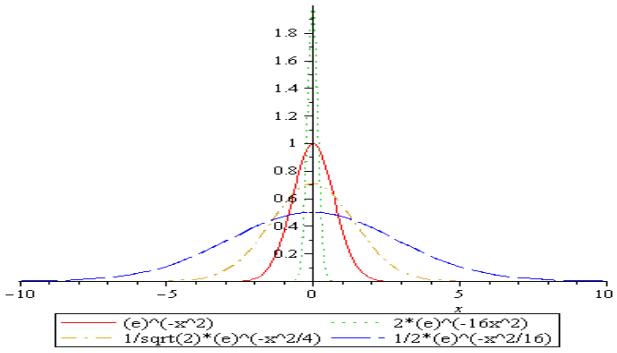
$$\psi_{j,0}(x) = 2^{j/2}\,\psi\left(2^{j}x\right) = 2^{j/2}\,e^{-(2^{j}x)^2} \quad , j \in \mathbf{Z}, \quad x \in \mathbf{R}.$$

If $j$ is positive then the graph of $\psi_{j,0}$ is similar to the graph $\psi$ but compressed, but if $j$ is negative then the graph of $\psi_{j,0}$ similar to the graph of $\psi$ but less localized as in figure 4.8 (a).

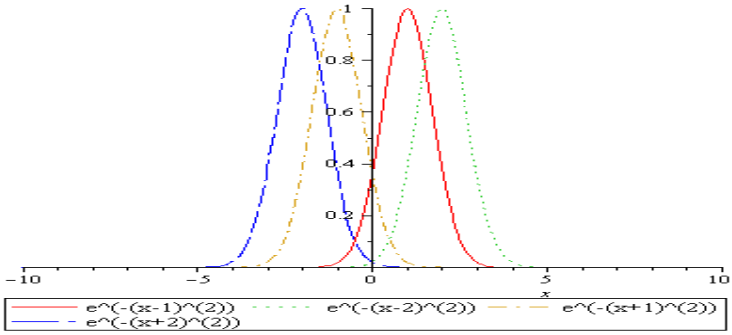
Also, to understand the translation role of $k$, fix $j = 0$

$$\psi_{0,k}(x) = \psi\left(x - k\right) = e^{-(x-k)^2} \quad , k \in \mathbf{Z}, \quad x \in \mathbf{R}.$$

If $k$ is positive then the graph of $\psi_{0,k}$ is similar to the graph $\psi$ but shifted to the right and if $k$ is negative then the graph of $\psi_{0,k}$ is similar to the graph of $\psi$ but shifted to the left as in figure 4.8 (b).




(a)




(b)

**Figure** 4.8

**Theorem 4.1** [7]**:** The system of wavelets $\{\psi_{j,k}\}_{j,k\in Z}$ form an orthonormal system which means

$$\int_{-\infty}^{\infty}\psi_{j,k}(x)\overline{\psi_{\hat{j},\hat{k}}(x)}\,dx=\begin{cases}1 & if \quad j=\hat{j}\,, \quad k=\hat{k}\\ 0 & Otherwise\end{cases} \tag{4.5}$$

**Definition 4.4** [7]**: Wavelet series**

The wavelet series expansion of a given function $f$ is given by:

$$f(x)=\sum_{j\in Z}\sum_{k\in Z}c_{j,k}\,\psi_{j,k}(x) \tag{4.6}$$

where the coefficients $c_{j,k}$ are uniquely defined by:

$$c_{j,k}=\int_{-\infty}^{\infty}f(x)\overline{\psi_{j,k}(x)}\,dx \tag{4.7}$$

The wavelet system $\{\psi_{j,k}\}_{j,k\in Z}$ is fully determined by the function $\psi$, we only need to store information about the signal function $\psi$ to be able to use (4.6). For practical and theoretical reasons, the mother wavelet must satisfy the admissibility conditions.

If we want to use (4.6) to calculate derivatives of the function $f$, so we need at least to assume that $\psi$ is differentiable. Compact support of $\psi$ is very important for computer-based calculations, since the computer can handle only a finite numbers. So, functions without compact support will have to be truncated at some point.

In practice, the mother wavelet $\psi$ need not satisfy all of the above properties. In such cases, the requirements of the mother wavelet can be modified to suite each application individually. For example, the requirement of the mother wavelet to have compact support can be replaced by a function that decays rapidly to zero. In other words, if $\psi$ does not have compact support, it is sufficient that $\exists\, c > 0$ and $\alpha > 0$ such that

$$|\psi(x)| \le c\, e^{-\alpha|x|} \quad , \forall\ x \in \mathbf{R} \tag{4.8}$$

Assuming the convergence of the wavelet series, we may use it to approximate a given function by:

$$f(x) \approx \sum_{j=-n}^{n} \sum_{k=-n}^{n} c_{j,k}\, \psi_{j,k}(x) \tag{4.9}$$

For sufficiently large value of $n \in \mathbf{N}$.

**Example 4.8:**

The Haar mother wavelet has the compact support $[0,1)$ and is given by the piecewise polynomial:

$$\psi(x)=\begin{cases}1 & , \quad x\in\left[0,\frac{1}{2}\right)\\ -1 & , \quad x\in\left[\frac{1}{2},1\right)\\ 0 & , \quad \textit{Otherwise}\end{cases}$$

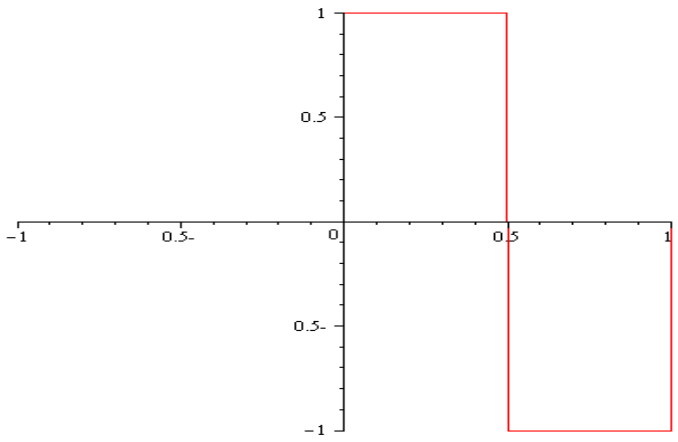



**Figure 4.9:** Haar mother wavelet

The Haar mother wavelet satisfies the requirement that each $f\in L^2(\mathbf{R})$ has an expansion as in equation (4.6), the condition (4.5) satisfies as well. However, $\psi$ is not continuous for $x=0, x=\frac{1}{2},$ and $x=1$. If we want to apply the representation (4.6) for signal transmission on finite machines, an infinite sum can't be evaluated; therefore the series is truncated and becomes:

$$f(x) \approx \sum_{j=-n}^{n} \sum_{k=-n}^{n} \widetilde{c}_{j,k}\, \psi_{j,k}(x) \tag{4.10}$$

If $\psi$ is the Haar wavelet, then even if $f$ is infinitely differentiable function, finite machines will produce a function which is not continuous.

## 4.3 Haar Transform

Haar transform is used to analyze discrete signals. Like all wavelet transform, the Haar transform decomposes a discrete signal into two subsignals each of half its length. The first subsignal, $\mathbf{a}^1$, is a running average or trend; the second subsignal, $\mathbf{d}^1$, is a running difference or fluctuation.

Let $\mathbf{f} = (f_1, f_2, ..., f_N)$ be a sampled signal where $N$ is even.The first trend subsignal is $\mathbf{a}^1 = (a_1, a_2, ..., a_{N/2})$ where $a_1 = (f_1 + f_2)/\sqrt{2}$. Similarly, $a_2 = (f_3 + f_4)/\sqrt{2}$ and so on. Each component of $\mathbf{a}^1$ is produced by taking the average of successive pairs of values of $\mathbf{f}$, and then multiplying these averages by $\sqrt{2}$. In general, the formula for the $m^{th}$ component of $\mathbf{a}^1$ is:

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \qquad m = 1, 2, ..., N/2 \tag{4.11}$$

**Example 4.9:**

Let $\mathbf{f} = (4, 6, 10, 12, 8, 6, 5, 5)$ then:

$$a_1 = \frac{4+6}{\sqrt{2}} = 5\sqrt{2} \quad , \quad a_2 = \frac{10+12}{\sqrt{2}} = 11\sqrt{2}$$

$$a_3 = \frac{8+6}{\sqrt{2}} = 7\sqrt{2} \quad , \quad a_4 = \frac{5+5}{\sqrt{2}} = 5\sqrt{2}$$

$\therefore$ The first trend subsignal is:

$\mathbf{a}^1 = \left(5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2}\right)$.

This result can be calculated as the following diagram:

| | | | | |
|---|---|---|---|---|
| $\mathbf{f}$ : | 4 6 | 10 12 | 8 6 | 5 5 |
| | 5 | 11 | 7 | 5 |
| $\mathbf{a}^1$ : | $5\sqrt{2}$ | $11\sqrt{2}$ | $7\sqrt{2}$ | $5\sqrt{2}$ |

We multiply by $\sqrt{2}$ to ensure that the Haar transform preserves the energy of a signal.

The other subsignal is called the first fluctuation. The first fluctuation of the signal $\mathbf{f}$ is denoted by $\mathbf{d}^1 = (d_1, d_2, ..., d_{N/2})$ and is computed by taking a running difference, $d_1 = (f_1 - f_2)/\sqrt{2}$. Similarly, $d_2 = (f_3 - f_4)/\sqrt{2}$.

In general, the formula for the $m^{th}$ component of $\mathbf{d}^1$ is:

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \qquad m = 1, 2, ..., N/2 \qquad (4.12)$$

**Example 4.10:**

For the signal $\mathbf{f}$ in example 4.9,

$$a_1 = \frac{4-6}{\sqrt{2}} = -\sqrt{2} \quad , \quad a_2 = \frac{10-12}{\sqrt{2}} = -\sqrt{2}$$

$$a_3 = \frac{8-6}{\sqrt{2}} = \sqrt{2} \quad , \quad a_4 = \frac{5-5}{\sqrt{2}} = 0$$

$\therefore$ The first fluctuation subsignal $\mathbf{d}^1$ is:

$\mathbf{d}^1 = \left(-\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0\right)$.

Also, $\mathbf{d}^1$ can be calculated as the following diagram

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\mathbf{f}$ : | 4 | 6 | 10 | 12 | 8 | 6 | 5 5 |
| | $-1$ | | $-1$ | | 1 | | 0 |
| $\mathbf{d}^1$ : | $-\sqrt{2}$ | | $-\sqrt{2}$ | | $\sqrt{2}$ | | 0 |

### 4.3.1 First-level Haar Transform

The Haar transform is performed in several levels. The first level is the mapping $H_1$ defined by:

$$H_1 : \mathbf{f} \to \left(\mathbf{a}^1 \mid \mathbf{d}^1\right) \tag{4.13}$$

Examples 4.9 and 4.10 gives:

$$H_1 : (4,6,10,12,8,6,5,5) \;\to\; \left(5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2} \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0\right)$$

The mapping $H_1$ in (4.13) has an inverse. Its inverse maps the transformed signal $\left(\mathbf{a}^1 \mid \mathbf{d}^1\right)$ back to the original signal $\mathbf{f}$.

For each $i = 1, 2, ..., N/2.$

$$a_i = \frac{f_{2i-1} + f_{2i}}{\sqrt{2}} \qquad \text{and} \qquad d_i = \frac{f_{2i-1} - f_{2i}}{\sqrt{2}}$$

We get,

$$f_{2i-1} = \frac{a_i + d_i}{\sqrt{2}} \qquad \text{and} \qquad f_{2i} = \frac{a_i - d_i}{\sqrt{2}}$$

The 1-level Haar inverse transform is:

$$\mathbf{f} = \left(\frac{a_1 + d_1}{\sqrt{2}}, \frac{a_1 - d_1}{\sqrt{2}}, ..., \frac{a_{N/2} + d_{N/2}}{\sqrt{2}}, \frac{a_{N/2} - d_{N/2}}{\sqrt{2}}\right) \tag{4.14}$$

**Example 4.11:**

To obtain the original signal from the transformed signal,

$\left(5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2} \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0\right)$

we apply formula (4.14)

$$f_1 = \frac{a_1 + d_1}{\sqrt{2}} = \frac{5\sqrt{2} + \left(-\sqrt{2}\right)}{\sqrt{2}} = \frac{4\sqrt{2}}{\sqrt{2}} = 4 \qquad f_2 = \frac{a_1 - d_1}{\sqrt{2}} = \frac{5\sqrt{2} - \left(-\sqrt{2}\right)}{\sqrt{2}} = \frac{6\sqrt{2}}{\sqrt{2}} = 6$$

$$f_3 = \frac{a_2 + d_2}{\sqrt{2}} = \frac{11\sqrt{2} + \left(-\sqrt{2}\right)}{\sqrt{2}} = \frac{10\sqrt{2}}{\sqrt{2}} = 10 \qquad f_4 = \frac{a_2 - d_2}{\sqrt{2}} = \frac{11\sqrt{2} - \left(-\sqrt{2}\right)}{\sqrt{2}} = \frac{12\sqrt{2}}{\sqrt{2}} = 12$$

$$f_5 = \frac{a_3 + d_3}{\sqrt{2}} = \frac{7\sqrt{2} + \sqrt{2}}{\sqrt{2}} = \frac{18\sqrt{2}}{\sqrt{2}} = 8 \qquad f_6 = \frac{a_3 - d_3}{\sqrt{2}} = \frac{7\sqrt{2} - \sqrt{2}}{\sqrt{2}} = \frac{6\sqrt{2}}{\sqrt{2}} = 6$$

$$f_7 = \frac{a_4 + d_4}{\sqrt{2}} = \frac{5\sqrt{2} + 0}{\sqrt{2}} = \frac{5\sqrt{2}}{\sqrt{2}} = 5 \qquad f_8 = \frac{a_4 - d_4}{\sqrt{2}} = \frac{5\sqrt{2} - 0}{\sqrt{2}} = \frac{5\sqrt{2}}{\sqrt{2}} = 5$$

$\therefore \quad \mathbf{f} = (4, 6, 10, 12, 8, 6, 5, 5)$

**Small fluctuation feature**

The magnitudes of the components of the fluctuation signal are often significantly smaller than the magnitude of the components of the original signal.

**Example 4.12:**

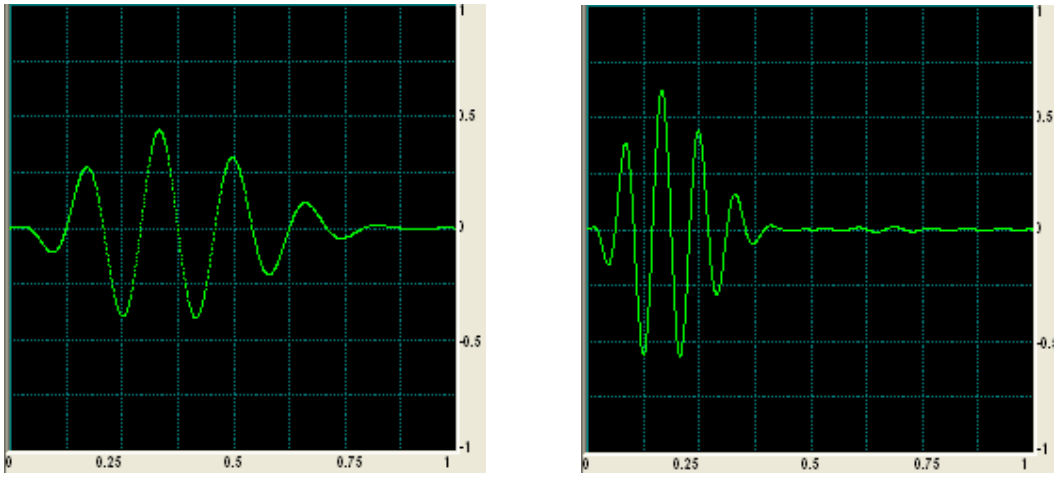
For the signal $\mathbf{f} = (4, 6, 10, 12, 8, 6, 5, 5)$ , the average magnitude of the components is 7. While, for its first fluctuation subsignal $\mathbf{d}^1 = \left(-\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0\right)$, the average magnitude of the components is $0.75\sqrt{2} \approx 1.06066$ . Notice that the magnitudes of the fluctuation's components

are an average of 6.6 times smaller than the magnitudes of the original signal's components.

**Example 4.13:**

Consider the signal

$$g(x) = 20x^2(1-x)^4 \cos 12\pi x \qquad 0 \le x < 1$$




a b

**Figure** 4.10

Figure 4.10(a,b) represent the original signal and its 1–level Haar transform, respectively. Notice, in figure 4.10(b) the trend subsignal is in the left half, over the interval $[0,0.5)$, while the fluctuation subsignal is in the right half over the interval $[0.5,1)$. It is clear that most of fluctuation's values are close to $0$ in magnitude. Also, the trend subsignal looks like the original signal, except it is shrinked by half in the length and expanded vertically by factor of $\sqrt{2}$.

Equation in (2.7) holds with a small time step size $h = t_{k+1} - t_k$ for each $k = 1, 2, 3, ..., N-1$, where $N$ is an even integer. If $h$ is small enough, then the successive values $f_{2m-1} = g(t_{2m-1})$ and $f_{2m} = g(t_{2m})$ of the signal $\mathbf{f}$ will be very

close to each other since $g$ is continuous. So, the fluctuation values satisfy:

$$d_m = \frac{g(t_{2m-1}) - g(t_{2m})}{\sqrt{2}} \approx 0$$

And the trend values satisfy:

$$a_m = \frac{g(t_{2m-1}) + g(t_{2m})}{\sqrt{2}}$$

$$\approx \frac{2\,g(t_{2m})}{\sqrt{2}} = \sqrt{2}\,g(t_{2m})$$

This equation shows that the componenets of $\mathbf{a}^1$ are approximately the same as sample values of $\sqrt{2}\,g(x)$ for $x = t_2, t_4, ..., t_N$. It shows that the graph of the first trend subsignal is similar in appearance to the graph of $g$. This is the reason that small fluctuation feature is important in signal compression. Compressing a signal means that we can save a signal using smaller number of bits. We can save the trend subsignal only and then we perform the inverse Haar transform using zero values for fluctuation subsignal to obtain an approximation of the original signal. Since the length of the trend subsignal is half the length of the original signal, this would achieve 50% compression.

### 4.3.2 Conservation and Compaction of Energy

In this section, we discuss the most two important properties of 1-level Haar transform.

1. Haar transform conserves the energy of signals.

2. Haar transform perform compaction of the energy of signals.

**Conservation of energy**

The energy of a Haar transformed signal $\left(\mathbf{a}^1 \mid \mathbf{d}^1\right)$ is:

$$\varepsilon_{\left(\mathbf{a}^1|\mathbf{d}^1\right)} = a_1{}^2 + a_2{}^2 + a_3{}^2 + \ldots + a^2{}_{N/2} + d_1{}^2 + d_2{}^2 + d_3{}^2 + \ldots + d^2{}_{N/2}$$

$$a_1^2 + d_1^2 = \left[\frac{f_1 + f_2}{\sqrt{2}}\right]^2 + \left[\frac{f_1 - f_2}{\sqrt{2}}\right]^2$$

$$= \frac{f_1^2 + 2f_1f_2 + f_2^2}{2} + \frac{f_1^2 - 2f_1f_2 + f_2^2}{2} = \frac{2f_1^2 + 2f_2^2}{2} = f_1^2 + f_2^2.$$

Similarly, $a_m{}^2 + d_m{}^2 = f^2{}_{2m-1} + f^2{}_{2m}$.

Therefore, $\varepsilon_{\left(\mathbf{a}^1|\mathbf{d}^1\right)} = f_1{}^2 + f_2{}^2 + f_3{}^2 + \ldots + f_N{}^2$.

In other words, $\varepsilon_{\left(\mathbf{a}^1|\mathbf{d}^1\right)} = \varepsilon_{\mathbf{f}}$, which justifies the conservation of energy property.

**Example 4.14:**

The energy for the signal $\mathbf{f} = (4,6,10,12,8,6,5,5)$ is:

$$\varepsilon_{\mathbf{f}} = 4^2 + 6^2 + 10^2 + 12^2 + 8^2 + 6^2 + 5^2 + 5^2 = 446,$$

The 1-level Haar transform of this signal is:

$(\mathbf{a}^1 \mid \mathbf{d}^1) = (5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2} \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)$, and its energy is:

$$\varepsilon_{(\mathbf{a}^1 \mid \mathbf{d}^1)} = 25 \times 2 + 121 \times 2 + 49 \times 2 + 25 \times 2 + 2 + 2 + 2 + 0 = 446.$$

So, the 1-level Haar transform has conserved the energy.

**Compaction of energy**

The energy of the trend subsignal $\mathbf{a}^1$ accounts for a large percentage of energy of the transformed signal $(\mathbf{a}^1 \mid \mathbf{d}^1)$.

Compaction of energy occur whenever the magnitudes of the fluctuation values are significantly smaller then the trend's values.

The next two examples sho how Haar transform redistributes the energy of a signal by compressing most of the energy in the trend subsignal.

**Example 4.15:**

Consider the signal $\mathbf{f} = (4, 6, 10, 12, 8, 6, 5, 5)$ in example 4.14. Its trend subsignal is:

$$\mathbf{a}^1 = (5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2}),$$

with energy

$$\varepsilon_{\mathbf{a}^1} = 25 \times 2 + 121 \times 2 + 49 \times 2 + 25 \times 2 = 440.$$

On the other hand, the fluctuation subsignal is:

$$\mathbf{d}^1 = (-\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0),$$

with energy

$$\varepsilon_{\mathbf{d}^1} = 2+2+2+0 = 6.$$

The energy of the trend subsignal accounts for $440/446 = 98.7\%$ of the total energy of the signal. In other words, the 1-level Haar transform has allocated $98\%$ of the total energy of $\mathbf{f}$ in the 1-level trend subsignal $\mathbf{a}^1$.

**Example 4.16:**

Consider the signal $\mathbf{f}$ generated from 10 equally spaced sampled values of the function $g(x) = 20x^2(1-x)^4 \cos 12\pi x$. Its energy is $127.308$ while the energy of its first trend subsignal $\mathbf{a}^1$ is $127.305$. So, $99.998\%$ of the total energy is compacted into the half-length subsignal $\mathbf{a}^1$. See figure 4.10(b).

### 4.3.3 Haar Transform, Multiple levels

The 1-level Haar transform can be repeated giving multiple levels Haar transform. The second level of the Haar transform is performed by computing a second trend $\mathbf{a}^2$ and a second fluctuation $\mathbf{d}^2$ from the first trend $\mathbf{a}^1$ only. In other words, $H_1 : \mathbf{a}^1 \to (\mathbf{a}^2 \mid \mathbf{d}^2)$. The 2-level Haar wavelet transform $H_2$ is defined by the mapping: $H_2 : \mathbf{f} \to (\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1)$.

The generalization to the k-level is straight forward. Applying the 1-level transform, $H_1 : \mathbf{a}^{k-1} \to (\mathbf{a}^k \mid \mathbf{d}^k)$, then the k-level Haar wavelet transform is given by: $H_k : \mathbf{f} \to (\mathbf{a}^k \mid \mathbf{d}^k \mid \mathbf{d}^{k-1} \mid \ldots \mid \mathbf{d}^1)$.

**Example 4.17:**

Recall the signal in example 4.14. To get the second trend subsignal $\mathbf{a}^2$ we apply formula (4.11) to the 1-level trend subsignal $\mathbf{a}^1$,

$\mathbf{a}^1$: $5\sqrt{2}$ $11\sqrt{2}$ $7\sqrt{2}$ $5\sqrt{2}$

$\mathbf{a}^2$: 16 12

And to get the second fluctuation subsignal $\mathbf{d}^2$ we apply formula (4.12) to $\mathbf{a}^1$,

$\mathbf{a}^1$: $5\sqrt{2}$ $11\sqrt{2}$ $7\sqrt{2}$ $5\sqrt{2}$

$\mathbf{d}^2$: $-6$ 2

So, the 2-level Haar transform of $\mathbf{f}$ is the signal

$$\left(\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1\right) = \left(16,12 \mid -6,2 \mid -\sqrt{2},-\sqrt{2},\sqrt{2},0\right).$$

Furthermore,

$\mathbf{a}^2$: 16 12

$\mathbf{a}^3$: $14\sqrt{2}$

and

$\mathbf{a}^2$: 16 12

$\mathbf{d}^3$: $2\sqrt{2}$

So, the 3-level Haar transform of $\mathbf{f}$ is:

$$\left(\mathbf{a}^3 \mid \mathbf{d}^3 \mid \mathbf{d}^2 \mid \mathbf{d}^1\right) = \left(14\sqrt{2} \mid 2\sqrt{2} \mid -6,2 \mid -\sqrt{2},-\sqrt{2},\sqrt{2},0\right).$$

**Remark:** Further transformation can not be done.

The energy of 2-level Haar transformed signal is:

$$\varepsilon_{(\mathbf{a}^2|\mathbf{d}^2|\mathbf{d}^1)} = (16)^2 + (12)^2 + 6^2 + 2^2 + (-\sqrt{2})^2 + (-\sqrt{2})^2 + (\sqrt{2})^2$$

$$= 256 + 144 + 36 + 4 + 2 + 2 + 2 + 2 = 446$$

Notice that the energy of 2-level Haar transform equals the energy of the orginal signal $\mathbf{f}$, i.e. $\varepsilon_{(\mathbf{a}^2|\mathbf{d}^2|\mathbf{d}^1)} = \varepsilon_{\mathbf{f}}$ since Haar transform conserves the energy.

The energy of the second trend subsignal is $\varepsilon_{\mathbf{a}^2} = (16)^2 + (12)^2 = 256 + 144 = 400$. Notice, the 2-level Haar transformed signal $(\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1)$ has $(400/446)\times 100\% \approx 90\%$ of the total energy of $\mathbf{f}$ contained in the second trend subsignal $\mathbf{a}^2$ which is $1/4$ the length of $\mathbf{f}$. This is further compaction, or localization, of the energy of $\mathbf{f}$. Moreover, $\varepsilon_{\mathbf{a}^3} = (14\sqrt{2})^2 = 196 \times 2 = 392$, so the third trend subsignal has almost $(392/446)\times 100\% \approx 89\%$ of the total energy of $\mathbf{f}$ .

The 3-level Haar transform $(\mathbf{a}^3 \mid \mathbf{d}^3 \mid \mathbf{d}^2 \mid \mathbf{d}^1)$ has almost 88% of the total energy of $\mathbf{f}$ contained in the third trend $\mathbf{a}^3$ which is $1/8$ the length of $\mathbf{f}$.

**Definition 4.5** [25]**:** The cumulative energy profile of a signal $\mathbf{f}$ is defined by:

$$\left( \frac{f_1^2}{\varepsilon_f}, \frac{f_1^2 + f_2^2}{\varepsilon_f}, \frac{f_1^2 + f_2^2 + f_3^2}{\varepsilon_f}, \cdots, 1 \right)$$

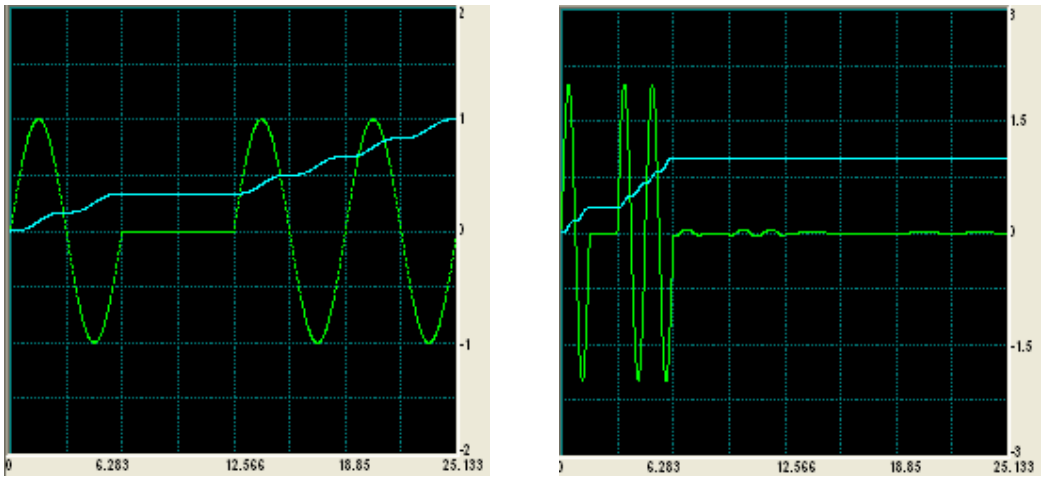
The cumulative energy profile of the signal $\mathbf{f}$ provides a summery of accumulation of energy in signal as time proceeds. The next two examples

show how Haar transform redistributes and localizes the energy in a continuous signal.

**Example 4.18:**

Consider the piecewise function

$$g(x)=\begin{cases}\sin(\mathrm{x}) & 0\le x\le 2\pi\\ 0 & 2\pi<\mathrm{x}<4\pi\\ \sin(x) & 4\pi\le x\le 8\pi\end{cases}$$




a b

**Figure** 4. 11

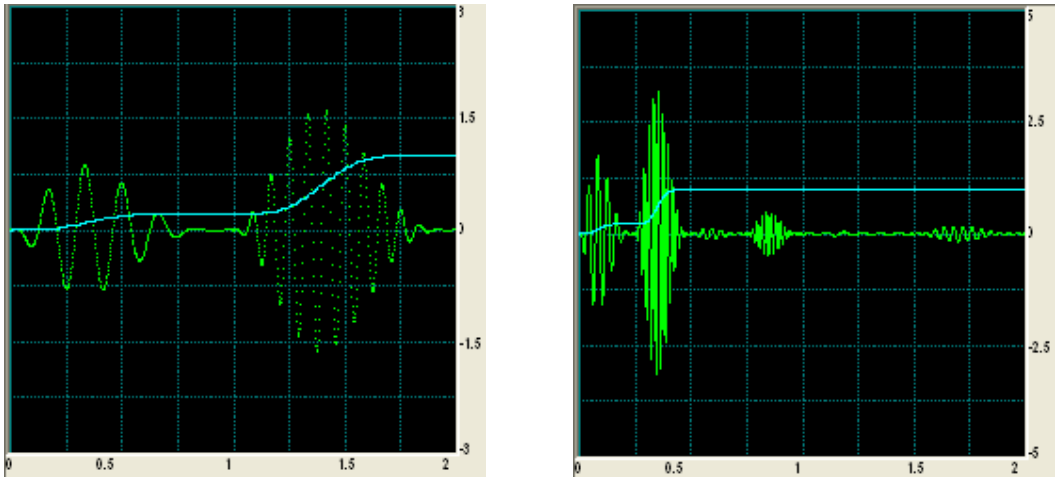
Figure 4.11(a) represents the original signal and its cumulative energy profile, also figure 4.11(b) represents the 2-level Haar transform for the signal $g(x)$ and its cumulative energy profile. It's clear that the 2-level Haar transform has redistributed and localized the energy of original signal.

**Example 4.19:**

Consider the piecewise function defined by:

$$g(x)=\begin{cases}40x^2(1-x)^4\cos(12\pi x) & 0\le x<1\\ 40x^2(1-x)^4\cos(12\pi x) & 1\le x\le 2\end{cases}$$

Its energy redistributed in two different time intervals.




a b

**Figure** 4.12

Figure 4.12(a) represents the original signal and its cumulative energy profile, also figure 4.12(b) represents the 2-level Haar transform for the signal $g(x)$ and its cumulative energy profile. It's clear that the 2-level Haar transform has redistributed and localized the energy of original signal.

### 4.3.4 Haar Wavelet Transform Via Scalar Product

A Haar wavelet is the simplest type of wavelets. It has a compact support, but unfortunately, Haar wavelets are not continuously differentiable which somewhat limits their applications. In discrete form, Haar wavelets are related to a mathematical operation called the Haar Transform. The Haar transform serves as a prototype for all other wavelet transforms. Studying Haar transform in details will help in understanding the more complicated wavelet transform.

The 1-level Haar wavelets are defined as:

$$\begin{aligned} \mathbf{W}_1^1 &= (\beta_1, \beta_2, 0, 0, ..., 0) \\ \mathbf{W}_2^1 &= (0, 0, \beta_1, \beta_2, ..., 0) \\ &\vdots \\ \mathbf{W}_{N/2}^1 &= (0, 0, ..., \beta_1, \beta_2) \end{aligned} \tag{4.15}$$

The numbers $\beta_1$ and $\beta_2$ are called wavelet numbers. The 1-level Haar scaling signals are defined as:

$$\begin{aligned} \mathbf{V}_1^1 &= (\alpha_1, \alpha_2, 0, 0, ..., 0) \\ \mathbf{V}_2^1 &= (0, 0, \alpha_1, \alpha_2, ..., 0) \\ &\vdots \\ \mathbf{V}_{N/2}^1 &= (0, 0, ..., \alpha_1, \alpha_2) \end{aligned} \tag{4.16}$$

As with the wavelet numbers, $\alpha_1$ and $\alpha_2$ are called scaling numbers.

The scaling numbers and the wavelet numbers are chosen so that the system $\left\{\mathbf{V}_m^1, \mathbf{W}_m^1\right\}_{m=1}^{N/2}$ is orthonormal. This means:

$$\mathbf{V}_n^1 \cdot \mathbf{V}_m^1 = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

$$\mathbf{W}_n^1 \cdot \mathbf{W}_m^1 = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

$$\mathbf{V}_n^1 \cdot \mathbf{W}_m^1 = 0 \quad \text{for all } n, m$$

In addition, we require that the fluctuation in a constant signal is zero which means $\mathbf{f}.\mathbf{W}_m^k = 0$.

The orthonormality of the fluctuation signals and the previous statement must satisfy:

$$\begin{aligned} \mathbf{W}_1^1 \cdot \mathbf{W}_1^1 &= \beta_1^2 + \beta_2^2 = 1 \\ \mathbf{P}_0 \cdot \mathbf{W}_1^1 &= \beta_1 + \beta_2 = 0 \end{aligned} \tag{4.17}$$

where $\mathbf{P}_0 = (1,1,1,1,.....,1)$.

Solving for the Haar wavelet numbers gives:

$$\beta_1 = \frac{1}{\sqrt{2}},\ \beta_2 = \frac{-1}{\sqrt{2}} \tag{4.18}$$

Similarly, the Haar scaling signals must satisfy:

$$\mathbf{V}_1^1 \cdot \mathbf{V}_1^1 = \alpha_1^2 + \alpha_2^2 = 1$$

$$\mathbf{V}_1^1 \cdot \mathbf{W}_1^1 = \alpha_1\beta_1 + \alpha_2\beta_2 = 0$$

Using the Haar wavelet numbers (4.18) and solving for the Haar scaling numbers gives:

$$\alpha_1 = \frac{1}{\sqrt{2}}, \alpha_2 = \frac{1}{\sqrt{2}}, \tag{4.19}$$

As can be seen, the scaling numbers are related to the wavelet numbers by the relation: $\alpha_1 = -\beta$ and $\alpha_2 = \beta_1$.

The 1-level Haar wavelets have a number of interesting properties. First, each wavelet has energy of 1. Second, each wavelet consist of a rapid fluctuation between only two non-zero values $\pm\sqrt{2}$ with an average values of 0. Finally, they are similar to each other. Each wavelet is a translation in time by an even number of time-units of the first Haar wavelet $\mathbf{W}_1^1$. The second Haar wavelet $\mathbf{W}_2^1$ is a translation forward in time by 2 units of $\mathbf{W}_1^1$ and $\mathbf{W}_3^1$ is a translation forward in time by four units of $\mathbf{W}_1^1$, and so on.

The 1-level Haar scaling signals are quite similar to the Haar wavelets. They all have energy 1 and have a support consisting of just two consecutive time indices. They are all translated by an even multiple of time-units of the first scaling signal $\mathbf{V}_1^1$. Unlike the Haar wavelets, the average values of the Haar scaling signals are not zero, in fact each scaling signal has an average value of $\frac{\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}}{2} = \frac{2}{\sqrt{2}} \times \frac{1}{2} = \frac{1}{\sqrt{2}}$ .

Using the 1-level Haar wavelet, we can express the values $d_1, d_2, ..., d_{N/2}$ for the first fluctuation subsignal $\mathbf{d}^1$ as a scalar products. For example,

$$d_1 = \frac{f_1 - f_2}{\sqrt{2}} = f_1 \times \frac{1}{\sqrt{2}} + f_2 \times \frac{-1}{\sqrt{2}} = \mathbf{f}.\mathbf{W}_1^1 \ .$$

Similarly,

$$d_2 = \frac{f_3 - f_4}{\sqrt{2}} = \frac{0+0+f_3-f_4}{\sqrt{2}} = 0+0+f_3\times\frac{1}{\sqrt{2}} + f_4\times\frac{-1}{\sqrt{2}} = \mathbf{f}.\mathbf{W}_2^1 \ .$$

And so on, we can rewrite formula (4.12) in terms of scalar products with the 1-level Haar wavelets:

$$d_m = \mathbf{f}\,.\,\mathbf{W}_m^1 \qquad \text{for } m = 1,2,\cdots,N/2 \tag{4.20}$$

Using 1-level Haar scaling signals, the values $a_1, a_2, ..., a_{N/2}$ for the first trend subsignal $\mathbf{a}^1$ are:

$$a_1 = \frac{f_1+f_2}{\sqrt{2}} = f_1\times\frac{1}{\sqrt{2}} + f_2\times\frac{1}{\sqrt{2}} = \mathbf{f}\,.\,\mathbf{V}_1^1$$

$$a_2 = \frac{f_3+f_4}{\sqrt{2}} = 0+0+f_3\times\frac{1}{\sqrt{2}} + f_4\times\frac{1}{\sqrt{2}} = \mathbf{f}\,.\,\mathbf{V}_2^1$$

And so on. We can rewrite formula (4.11) in terms of scalar products with the 1-level scaling signals :

$$a_m = \mathbf{f}\,.\,\mathbf{V}_m^1 \qquad \text{for } m = 1,2,\cdots,N/2 \tag{4.21}$$

Now $H_1 : \mathbf{f} \to \left(\mathbf{a}^1 \mid \mathbf{d}^1\right)$.

The 2-level Haar scaling signals are defined by:

$$\begin{aligned}
\mathbf{V}_1^2 &= \left(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},0,0,...,0\right) \\
\mathbf{V}_2^2 &= \left(0,0,0,0,\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},.0,...,0\right) \\
&\vdots \\
\mathbf{V}_{N/4}^2 &= \left(0,0,...,0,\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)
\end{aligned} \tag{4.22}$$

These scaling signals are all translations by multiple of four time-units of the first scaling signal $\mathbf{V}_1^2$, they all have energy 1 and average value of $\frac{\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}}{4}=\frac{2}{4}=\frac{1}{2}$.

Moreover, the values of the 2-level trend $\mathbf{a}^2$ are:

$$a_2=\frac{a_1+a_2}{\sqrt{2}}=\frac{\mathbf{f}.\mathbf{V}_1^1+\mathbf{f}.\mathbf{V}_2^1}{\sqrt{2}}=\frac{f_1}{\sqrt{2}}+\frac{f_2}{\sqrt{2}}+\frac{f_3}{\sqrt{2}}+\frac{f_3}{\sqrt{2}}$$

$$=f_1\times\frac{1}{\sqrt{2}}+f_2\times\frac{1}{\sqrt{2}}+f_3\times\frac{1}{\sqrt{2}}+f_3\times\frac{1}{\sqrt{2}}=\mathbf{f}.\mathbf{V}_1^2\ .$$

The 2-level trend subsignal is:

$$\mathbf{a}^2=\left(\mathbf{f}.\mathbf{V}_1^2,\mathbf{f}.\mathbf{V}_2^2,\mathbf{f}.\mathbf{V}_3^2,...,\mathbf{f}.\mathbf{V}_{N/4}^2\right) \tag{4.23}$$

Likewise, the two 2-level Haar wavelets are defined by:

$$\begin{aligned}\mathbf{W}_1^2&=\left(\frac{1}{2},\frac{1}{2},\frac{-1}{2},\frac{-1}{2},0,0,...,0\right)\\ \mathbf{W}_2^2&=\left(0,0,0,0,\frac{1}{2},\frac{1}{2},\frac{-1}{2},\frac{-1}{2},.0,...,0\right)\\ &\vdots\\ \mathbf{W}_{N/4}^2&=\left(0,0,...,0,\frac{1}{2},\frac{1}{2},\frac{-1}{2},\frac{-1}{2}\right)\end{aligned} \tag{4.24}$$

Note that these wavelets have support of length 4, since they all translation by multiple of four time-unit of the first wavelet $\mathbf{W}_1^2$ . They also have energy 1 and average value 0. the 2-level fluctuation is:

$$d_2=\frac{d_1-d_2}{\sqrt{2}}=\frac{\mathbf{f}.\mathbf{V}_1^1-\mathbf{f}.\mathbf{V}_2^1}{\sqrt{2}}=\frac{f_1}{\sqrt{2}}+\frac{f_2}{\sqrt{2}}-\frac{f_3}{\sqrt{2}}-\frac{f_3}{\sqrt{2}}$$

$$=f_1\times\frac{1}{\sqrt{2}}+f_2\times\frac{1}{\sqrt{2}}-f_3\times\frac{1}{\sqrt{2}}-f_3\times\frac{1}{\sqrt{2}}=\mathbf{f}.\mathbf{W}_1^2\ .$$

The 2-level fluctuation subsignal is:

$$\mathbf{d}^2 = \left(\mathbf{f}.\mathbf{V}_1^2, \mathbf{f}.\mathbf{W}_2^2, \mathbf{f}.\mathbf{W}_3^2, ..., \mathbf{f}.\mathbf{W}_{N/4}^2\right) \tag{4.25}$$

Now, $H_2 : \mathbf{f} \to \left(\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1\right)$.

### 4.3.5 Haar Wavelet Inverse Transform

We discussed the Haar transform using a scalar product with scaling signals and wavelets. Now, we will discuss how the inverse Haar transform can also be described in term of these elementary signals.

**Multilinear Resolution Analysis (MRA)**

Discrete signals are synthesized by beginning with a very low resolution signal and successively adding on details to create higher resolution versions ending with a complete synthesized of the signal at the finest resolution. To define the first level of Haar MRA, we rewrite the Haar inverse transform in formula (4.14) as:

$$\mathbf{f} = \left(\frac{a_1}{\sqrt{2}}, \frac{a_1}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, ..., \frac{a_{N/2}}{\sqrt{2}}, \frac{a_{N/2}}{\sqrt{2}}\right) + \left(\frac{d_1}{\sqrt{2}}, \frac{-d_1}{\sqrt{2}}, \frac{d_2}{\sqrt{2}}, \frac{-d_2}{\sqrt{2}}, ..., \frac{d_{N/2}}{\sqrt{2}}, \frac{-d_{N/2}}{\sqrt{2}}\right)$$

This formula shows that $\mathbf{f}$ can be expressed as a sum of two signals that we shall call the first averaged signal and the first detail signal.

That is:

$$\mathbf{f} = \mathbf{A}^1 + \mathbf{D}^1 \tag{4.26}$$

Where the signal $\mathbf{A}^1$ is the first averaged signal and is defined by**:**

$$\mathbf{A}^1 = \left(\frac{a_1}{\sqrt{2}}, \frac{a_1}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, ..., \frac{a_{N/2}}{\sqrt{2}}, \frac{a_{N/2}}{\sqrt{2}}\right) \tag{4.27}$$

And the signal $\mathbf{D}^1$ is the first detail signal and is defined by:

$$\mathbf{D}^1 = \left( \frac{d_1}{\sqrt{2}}, \frac{-d_1}{\sqrt{2}}, \frac{d_2}{\sqrt{2}}, \frac{-d_2}{\sqrt{2}}, ..., \frac{d_{N/2}}{\sqrt{2}}, \frac{-d_{N/2}}{\sqrt{2}} \right) \tag{4.28}$$

Notice that

$$\mathbf{A}^1 = \left( \frac{a_1}{\sqrt{2}}, \frac{a_1}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, \frac{a_2}{\sqrt{2}}, ..., \frac{a_{N/2}}{\sqrt{2}}, \frac{a_{N/2}}{\sqrt{2}} \right)$$

$$= a_1 \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, ..., 0 \right) + a_2 \left( 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, ..., 0 \right) + ... + \left( 0, 0, ..., \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)$$

$$\mathbf{A}^1 = a_1 \mathbf{V}_1^1 + a_2 \mathbf{V}_2^1 + ... + a_{N/2} \mathbf{V}_{N/2}^1 \tag{4.29a}$$

So, the averaged signal $\mathbf{A}^1$ is expressed in terms of Haar scaling signals.

In the same way we can express the detail signal $\mathbf{D}_1$ as:

$$\mathbf{D}^1 = d_1 \mathbf{W}_1^1 + d_2 \mathbf{W}_2^1 + ... + d_{N/2}\, \mathbf{W}_{N/2}^1 \tag{4.29b}$$

Applying the scalar product formulas for the coefficients in equation (4.20) and (4.21), we can rewrite equations (4.29 a) and (4.29 b) respectively as follows:

$$\mathbf{A}^1 = \left( f \,.\, \mathbf{V}_1^1 \right) \mathbf{V}_1^1 + \left( f \,.\, \mathbf{V}_2^1 \right) \mathbf{V}_2^1 + ... + \left( f \,.\, \mathbf{V}_{N/2}^1 \right) \mathbf{V}_{N/2}^1$$

$$\mathbf{D}^1 = \left( f \,.\, \mathbf{W}_1^1 \right) \mathbf{W}_1^1 + \left( f \,.\, \mathbf{W}_2^1 \right) \mathbf{W}_2^1 + ... + \left( f \,.\, \mathbf{W}_{N/2}^1 \right) \mathbf{W}_{N/2}^1$$

The first formula shows that the averaged signal is a combination of Haar scaling signals with the values of the first trend subsignal as its coefficients, and the second formula shows that the detail signal is a

combination of Haar wavelets with the values of the first detail signal as its coefficients.

**Example 4.20:**

In examples (4.9) and (4.10), the 1-level Haar transform of the signal

$$\mathbf{f} = (4,6,10,12,8,6,5,5)$$

is

$$(\mathbf{a}^1 \mid \mathbf{d}^1) = (5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2} \mid -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)$$

Applying formula (4.27), the averaged signal is:

$$\mathbf{A}^1 = (5,5,11,11,7,7,5,5)$$

Using formula (4.29 a) the first averaged signal can also be expressed in terms of Haar scaling signals as:

$$\mathbf{A}^1 = (5\sqrt{2}\,\mathbf{V}_1^1, 11\sqrt{2}\,\mathbf{V}_2^1, 7\sqrt{2}\,\mathbf{V}_3^1, 5\sqrt{2}\,\mathbf{V}_4^1).$$

Comparing these last two equations we can see that the position of the repeated averages corresponds precisely with the supports of the scaling signals.

Also, we found that the first fluctuation signal for $\mathbf{f}$ is $\mathbf{d}^1 = (-\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0)$. Using formula (4.28) we obtain: $\mathbf{D}^1 = (-1,1,-1,1,1,-1,0,0)$, and using formula (4.29 b), the first detail signal can also be expressed in terms of Haar wavelets as:

$$\mathbf{D}^1 = (-\sqrt{2}\,\mathbf{W}_1^1, -\sqrt{2}\,\mathbf{W}_2^1, \sqrt{2}\,\mathbf{W}_3^1, 0\,\mathbf{W}_4^1)$$

This formula shows that the values of $\mathbf{D}^1$ occur in successive pairs of rapidly fluctuating values positioned at the supports of the Haar wavelets. Using the result of $\mathbf{A}^1$ computed above, we have

$$\mathbf{f} = \mathbf{A}^1 + \mathbf{D}^1$$

$$\mathbf{f} = (5,5,11,11,7,7,5,5) + (-1,1,-1,1,1,-1,0,0)$$

$$= (4,6,10,12,8,6,5,5).$$

This equation illustrates the basic idea of MRA. The signal $\mathbf{f}$ can be expressed as a sum of a lower resolution signal $\mathbf{A}^1$ and a fluctuation signal $\mathbf{D}^1$. The lower resolution signal $\mathbf{A}^1$ serves as an approximation to the original signal. Increasing the resolution can be obtained by adding the detail signal $\mathbf{D}^1$ to $\mathbf{A}^1$.

### 4.3.6 Multiresolution Analysis, Multiple levels

The idea of the first level of the Haar MRA of a signal in the previous section can be extended to further to as many levels as the number of times the length of the signal can be divided by 2.

The second level of MRA of a signal $\mathbf{f}$ involves expressing $\mathbf{f}$ as:

$$\mathbf{f} = \mathbf{A}^2 + \mathbf{D}^2 + \mathbf{D}^1 \tag{4.30}$$

where $\mathbf{A}^2$ is the second averaged signal and $\mathbf{D}^2$ is the second detail signal.

Formula (4.26) express $\mathbf{f}$ as $\mathbf{f} = \mathbf{A}^1 + \mathbf{D}^1$ while formula (4.30) express $\mathbf{f}$ as $\mathbf{f} = \mathbf{A}^2 + \mathbf{D}^2 + \mathbf{D}^1$, which implies that:

$$\mathbf{A}^1 = \mathbf{A}^2 + \mathbf{D}^2 \tag{4.31}$$

This formula says that computing the second average signal $\mathbf{A}^2$ and the second detail signal $\mathbf{D}^2$ consist of performing a first level MRA of the signal $\mathbf{A}^2$. So, the second level averaged signal $\mathbf{A}^2$ satisfies

$$\mathbf{A}^2 = \left(\mathbf{f}.\mathbf{V}_1^2\right)\mathbf{V}_1^2 + \left(\mathbf{f}.\mathbf{V}_2^2\right)\mathbf{V}_2^2 + \ldots + \left(\mathbf{f}.\mathbf{V}_{N/4}^2\right)\mathbf{V}_{N/4}^2$$

and the second level detail signal $\mathbf{D}^2$ satisfies:

$$\mathbf{D}^2 = \left(\mathbf{f}.\mathbf{W}_1^2\right)\mathbf{W}_1^2 + \left(\mathbf{f}.\mathbf{W}_2^2\right)\mathbf{W}_2^2 + \ldots + \left(\mathbf{f}.\mathbf{W}_{N/4}^2\right)\mathbf{W}_{N/4}^2$$

**Example 4.21:**

If $\mathbf{f} = (4,6,10,12,8,6,5,5)$ we computed $\mathbf{a}^2$ and $\mathbf{a}^2 = (16,12)$. Therefore,

$$\mathbf{A}^2 = 16\,\mathbf{V}_1^2 + 12\,\mathbf{V}_2^2$$

$$= 16\left(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},0,0,0,0\right) + 12\left(0,0,0,0,\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$$

So, $\mathbf{A}^2 = (8,8,8,8,6,6,6,6)$

By comparing the results from examples 4.20 and 4.21 we notice that the second averaged signal $\mathbf{A}^2$ has values created from averages that involve twice as many values as the averages that create $\mathbf{A}^1$. But, the second averaged signal reflects more long term trends than those reflected in the first averaged signal $\mathbf{A}^1$. Consequently, these averages are repeated for twice as many time units. The signal has the second level fluctuation $\mathbf{d}^2 = (-6,2)$. Consequently,

$$\mathbf{D}^2 = -6\,\mathbf{W}_1^2 + 2\,\mathbf{W}_2^2$$

$$= -6\left(\frac{1}{2},\frac{1}{2},\frac{-1}{2},\frac{-1}{2},0,0,0,0\right) + 2\left(0,0,0,0,\frac{1}{2},\frac{1}{2},\frac{-1}{2},\frac{-1}{2}\right)$$

$$= (-3,-3,3,3,1,1,-1,-1)$$

Using $\mathbf{D}^1$ from example 4.20, we get

$$\mathbf{f} = \mathbf{A}^2 + \mathbf{D}^2 + \mathbf{D}^1$$

$$= (8,8,8,8,6,6,6,6) + (-3,-3,3,3,1,1,-1,-1) + (-1,1,-1,1,1,-1,0,0)$$

This formula illustrates the idea of MRA. The full resolution signal $\mathbf{f}$ is produced from a very low resolution, averaged signal $\mathbf{A}^2$ consisting of repetitions of the two averaged values, 8 and 6, to which are added two detail signals. The first addition for this averaged signal with enough details to produce the next higher resolution averaged signal $(5,5,11,11,7,7,5,5)$ the second addition provides enough details to produce the full resolution signal $\mathbf{f}$. In general, if the number $N$ of signal values is divisible k time by 2, then the k-level MRA is:

$$\mathbf{f} = \mathbf{A}^k + \mathbf{D}^k + \ldots + \mathbf{D}^2 + \mathbf{D}^1$$

**Methods of Wavelet Transform Compression**

We will illustrate the general principles of wavelet compression of signals.

Step 1: Perform a wavelet transform of the signal.

Step 2: Set equal to 0 all values of the wavelet transform which are insignificant, i.e. which lies below some threshold values specified by the user.

Step 3: Save only the significant, non-zero values of the transform obtained from step 2 and a significant map showing their indices. This should be a much smaller data set than the original signal.

Step 4: To reconstruct the signal, perform the inverse wavelet transform of the signal saved in step 3, assigning zero values to the insignificant values which were not transmitted. This decompression step produces an approximation of the original signal.

**Summery**

- The basic method of wavelet transform compression consisted of setting equal to zero all transformed values whose magnitudes lie below a threshold value.
- The compression version of the signal consists of the significant, non-zero, values of the transform which survived the thresholding, along with a significance map indicating their indices.
- Reconstruction (decompression) consists of using the significance map and the significant transform values to reconstruct the transform, and then performing an inverse wavelet transform to produce an approximation of the original signal.

# Chapter Five

# Daubechies Wavelet Transform

## 5.1 The Daub4 Wavelet Transform

## 5.2 Daub4 Inverse Transform

## 5.3 Conservation and Compaction of Energy

## 5.4 Other Daubechies Wavelets

## 5.5 Compression of Audio Signals

## 5.6 Two Dimensional Wavelet Transforms

### 5.6.1 Discrete Images

### 5.6.2 2D Wavelet Transform

### 5.6.3 Wavelet and Scaling Images

## 5.7 Applications

### 5.7.1 Fingerprints

### 5.7.2 Compression of Images

## 5.8 Conclusion

# Chapter 5

# Daubechies Wavelet Transform

In this chapter we consider the popular Daubechies wavelet transforms. Daubechies wavelet transforms are defined in the same way as the Haar wavelet transforms; i.e. by computing running averages and differences via scalar product with scaling signals and wavelets. The only difference between them consists in how these scaling signals and wavelets are defined. For the Daubechies wavelet transforms, the scaling signals and wavelets have longer support, i.e. they produce averages and differences using just a few more values from the signal. This difference makes the new transform more efficient in many applications especially in data compression.

## 5.1 The Daub4 Wavelet Transform

There is a variety of Daubechies transforms, but they are very similar. The simplest one is the Daub4 wavelet transform which is defined as follows:

Let $\mathbf{f}$ be a sampled signal with an even number $N$ of values. The 1-level Daub4 transform is the mapping $D_1 : \mathbf{f} \to (\mathbf{a}^1 \mid \mathbf{d}^1)$ where $\mathbf{a}^1 = (a_1, a_2, ..., a_{N/2})$ is the first trend subsignal and $\mathbf{d}^1 = (d_1, d_2, ..., d_{N/2})$ is the first fluctuation subsignal. Each component $a_m$ of the trend subsignal $\mathbf{a}^1$ is the dot product of $\mathbf{f}$ by a 1-level scaling signal $\mathbf{V}_m^1$:

$$a_m = \mathbf{f} \cdot \mathbf{V}_m^1 \tag{5.1}$$

Likewise, each component $d_m$ of $\mathbf{d}^1$ is the dot product of $\mathbf{f}$ by a 1-level wavelet signal $\mathbf{W}_m^1$.

$$d_m = \mathbf{f} \cdot \mathbf{W}_m^1 \tag{5.2}$$

Higher level Daub4 transforms can be extended to multiple levels as many times as the signal's length can be divided by 2. The extension to higher levels is similar to the way the Haar transform is extended, i.e. by applying 1-level Daub4 transform $D_1 : \mathbf{a}^1 \to (\mathbf{a}^2 \mid \mathbf{d}^2)$ to the first trend subsignal $\mathbf{a}^1$. The 2-level Daub4 transform $D_2$ is then defined by the mapping: $D_2 : \mathbf{f} \to (\mathbf{a}^2 \mid \mathbf{d}^2 \mid \mathbf{d}^1)$.

The generalization to the k-level is straight forward. Apply the 1-level transform: $D_1 : \mathbf{a}^{k-1} \to (\mathbf{a}^k \mid \mathbf{d}^k)$, then the k-level Daub4 transform is given by: $D_k : \mathbf{f} \to (\mathbf{a}^k \mid \mathbf{d}^k \mid \mathbf{d}^{k-1} \mid ... \mid \mathbf{d}^1)$.

The 1-level Daub4 scaling signals are defined by:

$$\begin{aligned}
&\mathbf{V}_1^1 = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, ..., 0) \\
&\mathbf{V}_2^1 = (0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0, 0, ..., 0) \\
&\vdots \\
&\mathbf{V}_{(N/2)-1}^1 = (0, 0, 0, 0, ..., 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) \\
&\mathbf{V}_{N/2}^1 = (\alpha_3, \alpha_4, 0, 0, ..., 0, \alpha_1, \alpha_2)
\end{aligned}, \tag{5.3}$$

The numbers $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are called scaling numbers. The 1-level Daub4 wavelet signals are defined by:

$$\begin{aligned}
&\mathbf{W}_1^1 = (\beta_1, \beta_2, \beta_3, \beta_4, 0, 0, ..., 0) \\
&\mathbf{W}_2^1 = (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, 0, 0, ..., 0) \\
&\vdots \\
&\mathbf{W}_{(N/2)-1}^1 = (0, 0, 0, 0, ..., 0, \beta_1, \beta_2, \beta_3, \beta_4) \\
&\mathbf{W}_{N/2}^1 = (\beta_3, \beta_4, 0, 0, 0, ..., 0, \beta_1, \beta_2)
\end{aligned} \tag{5.4}$$

Now, the numbers $\beta_1, \beta_2, \beta_3, \beta_4$ are called wavelet numbers.

The scaling numbers and wavelet numbers are chosen so that the system $\{\mathbf{V}_m^1, \mathbf{W}_m^1\}_{m=1}^{N/2}$ is orthonormal which means:

$$\mathbf{V}_n^1 \cdot \mathbf{V}_m^1 = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

$$\mathbf{W}_n^1 \cdot \mathbf{W}_m^1 = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases}$$

$$\mathbf{V}_n^1 \cdot \mathbf{W}_m^1 = 0 \quad \text{for all } n, m .$$

In addition, the following property is required for computing the fluctuation numbers.

**Property I** [25]**:** if $\mathbf{f}$ is (approximately) linear over the support of a k-level Daub4 wavelet $\mathbf{W}_m^k$, then the k-level fluctuation value $\mathbf{f} . \mathbf{W}_m^k$ is (approximately) zero.

The orthonormality of the fluctuation signals and property I imply that the Daub4 wavelets (fluctuations) must satisfy:

$$\begin{aligned}
\mathbf{W}_1^1 \cdot \mathbf{W}_1^1 &= \beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 = 1 \\
\mathbf{P}_0 \cdot \mathbf{W}_1^1 &= \beta_1 + \beta_2 + \beta_3 + \beta_4 = 0 \\
\mathbf{P}_1 \cdot \mathbf{W}_1^1 &= 0\beta_1 + 1\beta_2 + 2\beta_3 + 3\beta_4 = 0 \\
\mathbf{W}_1^1 \cdot \mathbf{W}_2^1 &= \beta_1\beta_3 + \beta_2\beta_4 = 0
\end{aligned} \tag{5.5}$$

Where $\mathbf{P}_0 = (1,1,1,1,.....,1)$ and $\mathbf{P}_1 = (0,1,2,3,4,....N)$.

Solving for the Daub4 wavelet numbers gives:

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}},\ \beta_2 = \frac{-3+\sqrt{3}}{4\sqrt{2}}, \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}} \tag{5.6}$$

Similarly, the Daub4 scaling signals must satisfy:

$$\begin{aligned}
\mathbf{V}_1^1 \cdot \mathbf{V}_1^1 &= \alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 = 1 \\
\mathbf{V}_1^1 \cdot \mathbf{W}_1^1 &= \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_3\beta_3 + \alpha_4\beta_4 = 0 \\
\mathbf{V}_1^1 \cdot \mathbf{W}_2^1 &= \alpha_1\beta_3 + \alpha_2\beta_4 = 0 \\
\mathbf{V}_2^1 \cdot \mathbf{W}_1^1 &= \alpha_3\beta_1 + \alpha_4\beta_2 = 0
\end{aligned}$$

Using the Daub4 wavelet numbers (5.6) and solving for the Daub4 scaling numbers gives:

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}},\ \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ \ \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}},\ \ \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{5.7}$$

As can be seen, the scaling numbers are related to the wavelet numbers by the relations: $\alpha_1 = -\beta_4, \alpha_2 = \beta_3$, $\alpha_3 = -\beta_2$, and $\alpha_4 = \beta_1$. In addition, the scaling numbers satisfy the relation: $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \sqrt{2}$. This relations

means that each component of the 1-level trend subsignal $\mathbf{a}^1$; $a_m = \mathbf{f}.\mathbf{V}_m^1$ is the weighted average of four consecutive values of $\mathbf{f}$ multiplied by $\sqrt{2}$.

Each Daub4 scaling signal has a support of just four time-units. The second scaling signal $\mathbf{V}_2^1$ is just a translation by two time-units of the first scaling signal $\mathbf{V}_1^1$ and each subsequence scaling signal is a translation by a multiple of two time-units of $\mathbf{V}_1^1$. For $\mathbf{V}_{N/2}^1$, we have to translate $\mathbf{V}_1^1$ by $N/2-1$ time-units which would send $\alpha_3$ and $\alpha_4$ beyond the length $N$ of the signal $\mathbf{f}$. This is avoided by wraping-around to the start giving $\mathbf{V}_{N/2}^1 = (\alpha_3, \alpha_4, 0, 0, ..., 0, \alpha_1, \alpha_2)$. The same is true for the Daub4 wavelet signals $\mathbf{W}_m^1$.

## 5.2 Daub4 Inverse Transform

Each level of Daub4 transform has an inverse. The inverse for 1-level Daub4 transformed signal $(\mathbf{a}^1 \mid \mathbf{d}^1)$ is given by:

$$\mathbf{f} = \mathbf{A}^1 + \mathbf{D}^1 \tag{5.8a}$$

where $\mathbf{A}^1$ is the first averaged signal:

$$\mathbf{A}^1 = a_1\mathbf{V}_1^1 + a_2\mathbf{V}_2^1 + ... + a_{N/2}\,\mathbf{V}_{N/2}^1 \tag{5.8b}$$

And $\mathbf{D}^1$ is the first detail (fluctuation) signal,

$$\mathbf{D}^1 = d_1\mathbf{W}_1^1 + d_2\mathbf{W}_2^1 + ... + d_{N/2}\,\mathbf{W}_{N/2}^1 \tag{5.8c}$$

To prove this, expand the signal $\mathbf{f}$ in terms of the orthonormal basis $\{\mathbf{V}_m^1, \mathbf{W}_m^1\}_{m=1}^{N/2}$:

$$\mathbf{f} = \sum_{m=1}^{N/2} r_m \mathbf{V}_m^1 + \sum_{m=1}^{N/2} s_m \mathbf{W}_m^1$$

Taking the inner product with $\mathbf{V}_m^1$, and using the orthonormality of the basis, we get: $r_m = \mathbf{f} \cdot \mathbf{V}_m^1$. Similarly, we get $s_m = \mathbf{f} \cdot \mathbf{W}_m^1$. Therefore

$$\begin{aligned}\mathbf{f} &= \sum_{m=1}^{N/2} (\mathbf{f} \cdot \mathbf{V}_m^1)\mathbf{V}_m^1 + \sum_{m=1}^{N/2} (\mathbf{f}.\mathbf{W}_m^1)\mathbf{W}_m^1 \\ &= \sum_{m=1}^{N/2} a_m \mathbf{V}_m^1 + \sum_{m=1}^{N/2} d_m \mathbf{W}_m^1 \\ &= \mathbf{A}^1 + \mathbf{D}^1\end{aligned}$$

Using natural basis $\{\mathbf{V}_m^0, \mathbf{W}_m^0\}_{m=1}^{N/2}$, where each element (vector) consists of zeros except for a one in the $m^{th}$ component, we can write:

$$\mathbf{V}_m^1 = \alpha_1 \mathbf{V}_{2m-1}^0 + \alpha_2 \mathbf{V}_{2m}^0 + \alpha_3 \mathbf{V}_{2m+1}^0 + \alpha_4 \mathbf{V}_{2m+2}^0 \tag{5.9a}$$

with a wrap-around defined by $\mathbf{V}_{m+N}^0 = \mathbf{V}_m^0$. Similarly the second level Daub4 scaling signals satisfy

$$\mathbf{V}_m^2 = \alpha_1 \mathbf{V}_{2m-1}^1 + \alpha_2 \mathbf{V}_{2m}^1 + \alpha_3 \mathbf{V}_{2m+1}^1 + \alpha_4 \mathbf{V}_{2m+2}^1 \tag{5.9b}$$

with a wrap-around defined by $\mathbf{V}_{m+N/2}^1 = \mathbf{V}_m^1$.

Notice that each second-level Duab4 scaling signal $\mathbf{V}_m^2$ lives for just 10 time-units, and translate by $4m$ time-units of $\mathbf{V}_1^2$ (if we include wrap-around).

The second-level trend values are $\{\mathbf{f}.\mathbf{V}_m^2\}$ and they measure trends over 10 successive values of $\mathbf{f}$ located in the same time positions as the non-zero values of $\mathbf{V}_m^2$. Hence, these trends are measured over short time intervals that are shifted by multiples of 4 time-unit of the interval consisting of the

first 10 time-units. These 10-unit trends are slightly more than twice as long as the trends measured by the first level scaling subsignals.

The k-level Daub4 scaling signals are defined by formulas similar to (5.9 a) and (5.9 b). The 1-level Daub4 wavelets satisfy:

$$\mathbf{W}_m^1 = \beta_1 \mathbf{W}_{2m-1}^0 + \beta_2 \mathbf{W}_{2m}^0 + \beta_3 \mathbf{W}_{2m+1}^0 + \beta_4 \mathbf{W}_{2m+2}^0$$

Similarly, the 2-level Daub4 wavelets are defined by:

$$\mathbf{W}_m^2 = \beta_1 \mathbf{W}_{2m-1}^1 + \beta_2 \mathbf{W}_{2m}^1 + \beta_3 \mathbf{W}_{2m+1}^1 + \beta_4 \mathbf{W}_{2m+2}^1$$

The inverse of the 2-level Daub4 transform is described by the formula

$$\mathbf{f} = \mathbf{A}^2 + \mathbf{D}^2 + \mathbf{D}^1 \tag{5.10a}$$

where

$$\mathbf{A}^2 = \left(\mathbf{f}.\mathbf{V}_1^2\right)\mathbf{V}_1^2 + \left(\mathbf{f}.\mathbf{V}_2^2\right)\mathbf{V}_2^2 + \ldots + \left(\mathbf{f}.\mathbf{V}_{N/4}^2\right)\mathbf{V}_{N/4}^2$$

$$\mathbf{D}^2 = \left(\mathbf{f}.\mathbf{W}_1^2\right)\mathbf{W}_1^2 + \left(\mathbf{f}.\mathbf{W}_2^2\right)\mathbf{W}_2^2 + \ldots + \left(\mathbf{f}.\mathbf{W}_{N/4}^2\right)\mathbf{W}_{N/4}^2 \tag{5.10b}$$

The product $\mathbf{f}.\mathbf{V}_m^2$ is the $m^{th}$ component of the 2-level trend subsignal of $\mathbf{a}^2$ and $\mathbf{f}.\mathbf{W}_m^2$ is the $m^{th}$ component of the 2-level fluctuation subsignal $\mathbf{d}^2$. The signal $\mathbf{D}^1$ is the first detail signal which is defined in (5.8 c). Since $\mathbf{A}^1 = \mathbf{A}^2 + \mathbf{D}^2$, the second detail subsignal $\mathbf{D}^2$ provides the details needed to produce the first averaged signal from the second averaged signal.

The k-level Daub4 transform has an inverse that produce the following Multi-resolution Analysis (MRA) for the signal $\mathbf{f}$ :

$$\mathbf{f} = \mathbf{A}^k + \mathbf{D}^k + ... + \mathbf{D}^2 + \mathbf{D}^1$$

Let $N_k = N/2^k$

Formulas for $\mathbf{A}^k$ and $\mathbf{D}^k$ are:

$$\mathbf{A}^k = \left(\mathbf{f}.\mathbf{V}_1^k\right)\mathbf{V}_1^k + \left(\mathbf{f}.\mathbf{V}_2^k\right)\mathbf{V}_2^k + ... + \left(\mathbf{f}.\mathbf{V}_{N_k}^k\right)\mathbf{V}_{N_k}^k$$

$$\mathbf{D}^k = \left(\mathbf{f}.\mathbf{W}_1^k\right)\mathbf{W}_1^k + \left(\mathbf{f}.\mathbf{W}_2^k\right)\mathbf{W}_2^k + ... + \left(\mathbf{f}.\mathbf{W}_{N_k}^k\right)\mathbf{W}_{N_k}^k$$

Agin, the product $\mathbf{f}.\mathbf{V}_m^k$ is the $m^{th}$ component of the k-level trend subsignal of $\mathbf{a}^k$ and $\mathbf{f}.\mathbf{W}_m^k$ is the $m^{th}$ component of the k-level fluctuation subsignal $\mathbf{d}^k$.

## 5.3 Conservation and Compaction of Energy

In this section we will show that the first level Daub4 transform preserves the energy of the signal. Higher level can be treated in the same way.

Define the matrix $\mathbf{D}_N$ as the matrix whose rows are given by:

$$\text{Row}_{2m\text{-}1} = \mathbf{V}_m^1, \quad \text{Row}_{2m} = \mathbf{W}_m^1, \quad \text{where} \quad m = 1,2,...,N/2.$$

It follows that:

$$\mathbf{D}_N = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ \alpha_3 & \alpha_4 & 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_1 & \alpha_2 \\ \beta_3 & \beta_4 & 0 & 0 & 0 & 0 & 0 & \cdots & \beta_1 & \beta_2 \end{pmatrix} \quad (5.11)$$

The rows of $\mathbf{D}_N$ are orthonormal, therefore $\mathbf{D}_N$ is an orthogonal matrix, and this means;

$$\mathbf{D}_N^T \mathbf{D}_N = \mathbf{I}_N \quad (5.12)$$

where $\mathbf{D}_N^T$ is the transpose of the matrix $\mathbf{D}_N$ and $\mathbf{I}_N$ is the $N \times N$ identity matrix.

Now,

$$\mathbf{D}_N \mathbf{f}^T = \left(a_1, d_1, a_2, d_2, \cdots, a_{N/2}, d_{N/2}\right)^T$$

Therefore, the energy of the 1-level Daub4 transform is:

$$\varepsilon_{\left(\mathbf{a}^1 | \mathbf{d}^1\right)} = {a_1}^2 + {d_1}^2 + {a_2}^2 + {d_2}^2 + \cdots + {a_{N/2}}^2 + {d_{N/2}}^2$$

So,

$$\varepsilon_{\left(\mathbf{a}^1 | \mathbf{d}^1\right)} = \left(\mathbf{D}_N \mathbf{f}^T\right)^T \left(\mathbf{D}_N \mathbf{f}^T\right)$$

$$= \left(\mathbf{f}^T\right)^T \mathbf{D}_N^T \mathbf{D}_N \mathbf{f}^T$$

$$= \mathbf{f}\, \mathbf{I}_N \mathbf{f}^T$$

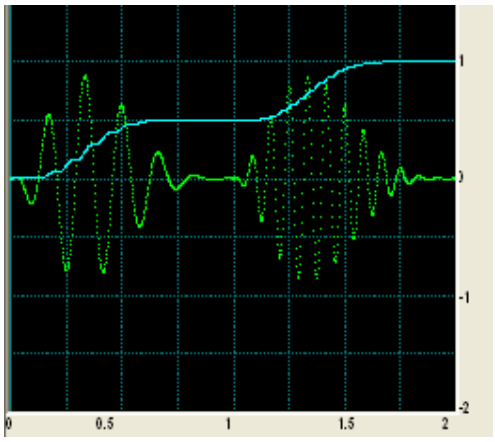
$$= \mathbf{f}\, \mathbf{f}^T = \varepsilon_{\mathbf{f}}$$

The above results show that the 1-level Daub4 transform has the conservation energy property. This result can be generalized to every level Daub4.

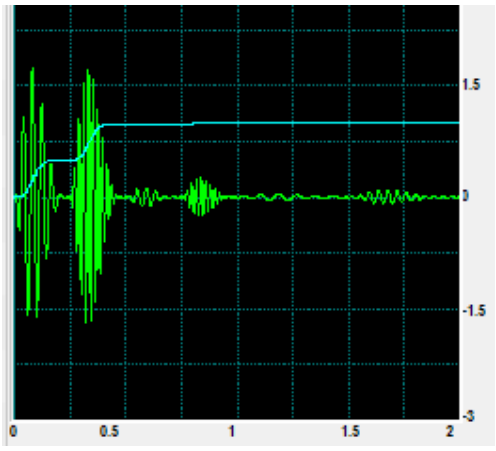
**Example 5.1:**

Consider the signal

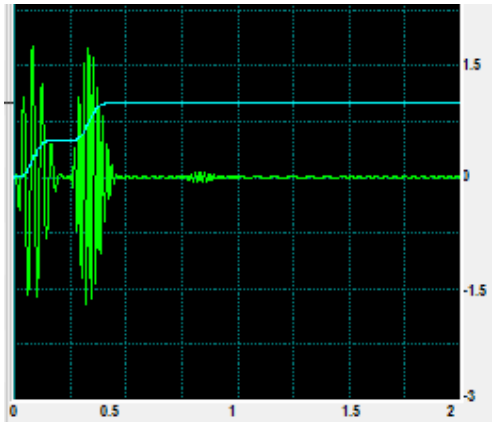
$$g(x)=\begin{cases}40x^2(1-x)^4\cos(12\pi x) & 0\le x<1\\ 40x^2(x-1)^2(2-x)^4\cos(24\pi x) & 1\le x\le 2\end{cases}$$




(a)




(b)




(c)

**Figure** 5.1

Figures 5.1 (b,c) show the 2-level Haar transform and the 2-level Daub4 transform respectively, but as can be seen, the Daub4 transform also redistribute the energy of the signal into a more compact form, also, the cumulative energy profile shows that the 2-level Daub4 transform has redistributed almost all of the energy of the signal into the second trend subsignal, it is obvious that the Daub4 transform achieves a more compact redistribution of the energy of the signal.

## 5.4 Other Daubechies Wavelets

There is a variety of Daub transforms, all are constructed in a way similar to Daub4. The idea is to increase the supports of scaling and wavelet signals to get more accurate results. As an example, we consider the Daub6 transform:

The 1-level Daub6 scaling signals are defined by:

$$\begin{aligned}
&\mathbf{V}_1^1 = (\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,0,0,...,0)\\
&\mathbf{V}_2^1 = (0,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,\ 0,0,...,0)\\
&\mathbf{V}_3^1 = (0,0,0,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6,0,0,...,0)\\
&\vdots\\
&\mathbf{V}_{(N/2)-1}^1 = (\alpha_5,\alpha_6,0,0,0,0,...,0,\alpha_1,\alpha_2,\alpha_3,\alpha_4)\\
&\mathbf{V}_{N/2}^1 = (\alpha_3,\alpha_4,\alpha_5,\alpha_6,0,0,...,0,\alpha_1,\alpha_2)
\end{aligned} \tag{5.13}$$

The numbers $\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5,\alpha_6$ are scaling numbers. A wrap-around occurrs in $\mathbf{V}_{(N/2)-1}^1$ and $\mathbf{V}_{N/2}^1$ . The 1-level Daub4 wavelet signals are defined by:

$$\begin{aligned}
\mathbf{W}_1^1 &= (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0) \\
\mathbf{W}_2^1 &= (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0) \\
\mathbf{W}_3^1 &= (0, 0, 0, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0) \\
&\vdots \\
\mathbf{W}_{(N/2)-1}^1 &= (\beta_5, \beta_6, 0, 0, 0, 0, ..., 0, \beta_1, \beta_2, \beta_3, \beta_4) \\
\mathbf{W}_{N/2}^1 &= (\beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0, \beta_1, \beta_2)
\end{aligned} \tag{5.14}$$

The numbers $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$ are wavelet numbers. A wrap-around occurrs in $\mathbf{W}_{(N/2)-1}^1$ and $\mathbf{W}_{N/2}^1$ . The scaling numbers satisfy the following properties:

$$\alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 + \alpha_5^2 + \alpha_6^2 = 1 \tag{5.15a}$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = \sqrt{2} \tag{5.15b}$$

Equation (5.15 a) means that the energy of each scaling signal $\mathbf{V}_m^1$ has an energy 1, and, equation (5.15 b) means that the trend values $\mathbf{f}.\mathbf{V}_m^1$ are average of six successive values of $\mathbf{f}$ multiplied by $\sqrt{2}$. The wavelet numbers satisfy:

$$\beta_1{}^2 + \beta_2{}^2 + \beta_3{}^2 + \beta_4{}^2 + \beta_5{}^2 + \beta_6{}^2 = 1$$

The above equation says that each $\mathbf{W}_m^1$ has energy of 1.

**Property II** [25]**:** if $\mathbf{f}$ is (approximately) quadratic over the support of a k-level Daub6 wavelet $\mathbf{W}_m^k$, then the k-level fluctuation value $\mathbf{f}.\mathbf{W}_m^k$ is (approximately) zero.

The orthonormality of the fluctuation signals and property II imply that the Daub6 wavelets (fluctuations) must satisfy:

$$\begin{aligned}
&\mathbf{W}_1^1.\mathbf{W}_1^1 = \beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 + \beta_5^2 + \beta_6^2 = 1 \\
&\mathbf{P}_0.\mathbf{W}_1^1 = \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 = 0 \\
&\mathbf{P}_1.\mathbf{W}_1^1 = 0\beta_1 + 1\beta_2 + 2\beta_3 + 3\beta_4 + 4\beta_5 + 5\beta_6 = 0 \\
&\mathbf{P}_2.\mathbf{W}_1^1 = 0^2\beta_1 + 1^2\beta_2 + 2^2\beta_3 + 3^2\beta_4 + 4^2\beta_5 + 5^2\beta_6 = 0 \\
&\mathbf{W}_1^1.\mathbf{W}_2^1 = \beta_1\beta_3 + \beta_2\beta_4 + \beta_3\beta_5 + \beta_4\beta_6 = 0 \\
&\mathbf{W}_1^1.\mathbf{W}_3^1 = \beta_1\beta_5 + \beta_2\beta_6 = 0
\end{aligned} \tag{5.16}$$

where $\mathbf{P}_0 = (1,1,1,1,.....,1)$, $\mathbf{P}_1 = (0,1,2,3,4,....N)$ and $\mathbf{P}_2 = (0^2,1^2,2^2,3^2,4^2,....N)$.

Solving for the Daub6 wavelet numbers gives:

$$\beta_1 = 0.0352262918857095, \qquad \beta_2 = 0.0854412738820267,$$

$$\beta_3 = -0.135011020010255, \qquad \beta_4 = -0.459877502118491,$$

$$\beta_5 = 0.806891509311092, \qquad \beta_6 = -0.332670552950083.$$

Similarly, the Daub6 scaling signals must satisfy:

$$\begin{aligned}
&\mathbf{V}_1^1.\mathbf{V}_1^1 = \alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 + \alpha_5^2 + \alpha_6^2 = 1 \\
&\mathbf{P}_0.\mathbf{W}_1^1 = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = 0 \\
&\mathbf{P}_1.\mathbf{W}_1^1 = 0\alpha_1 + 1\alpha_2 + 2\alpha_3 + 3\alpha_4 + 4\alpha_5 + 5\alpha_6 = 0 \\
&\mathbf{P}_2.\mathbf{W}_1^1 = 0^2\alpha_1 + 1^2\alpha_2 + 2^2\alpha_3 + 3^2\alpha_4 + 4^2\alpha_5 + 5^2\alpha_6 = 0 \\
&\mathbf{V}_1^1.\mathbf{V}_2^1 = \alpha_1\alpha_3 + \alpha_2\alpha_4 + \alpha_3\alpha_5 + \alpha_4\alpha_6 = 0 \\
&\mathbf{V}_1^1.\mathbf{W}_2^1 = \alpha_3\beta_1 + \alpha_4\beta_2 + \alpha_5\beta_3 + \alpha_6\beta_4 = 0
\end{aligned} \tag{5.17}$$

Where $\mathbf{P}_0 = (1,1,1,1,.....,1)$, $\mathbf{P}_1 = (0,1,2,3,4,....N)$ and $\mathbf{P}_2 = (0^2,1^2,2^2,3^2,4^2,....N)$.

Using the Daub6 wavelet numbers and solving for the Daub6 scaling numbers gives:

$$\alpha_1 = 0.332670552950083, \qquad \alpha_2 = 0.806891509311092,$$

$$\alpha_3 = 0.459877502118491, \qquad \alpha_4 = -0.135011020010255,$$

$$\alpha_5 = -0.0854412738820267, \qquad \alpha_6 = 0.0352262918857095.$$

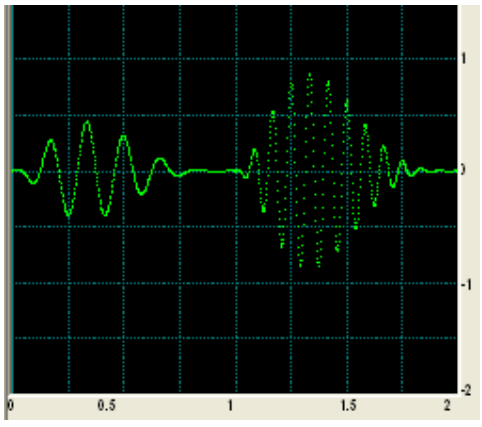
As can be seen, the scaling numbers are related to the wavelet numbers by the relations: $\alpha_1 = -\beta_6, \alpha_2 = \beta_5, \alpha_3 = -\beta_4, \alpha_4 = \beta_3, \alpha_5 = -\beta_2, \alpha_6 = \beta_1.$

**Example 5.2:**

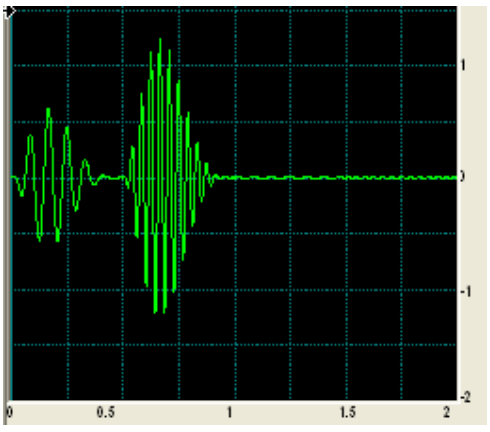
Consider the signal

$$g(x) = \begin{cases} 20x^2(1-x)^4 \cos(12\pi x) & 0 \le x < 1 \\ 40(x-1)^2(2-x)^4 \cos(24\pi x) & 1 \le x \le 2 \end{cases}$$

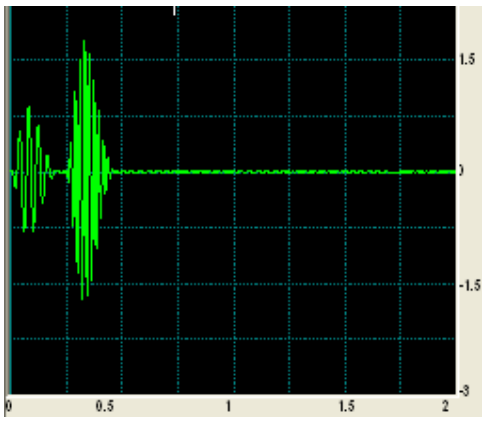
Using the FAWAV software [26], we get:




(a)



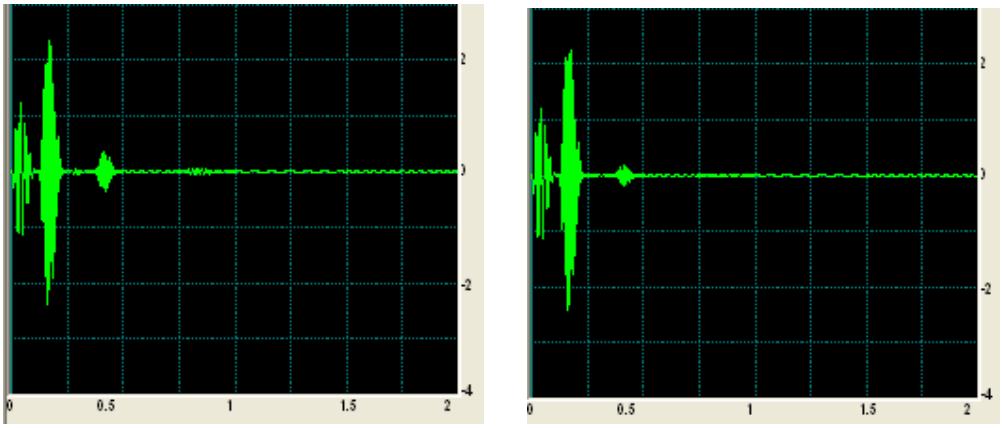

(b)




(c)

**Figure** 5.2

Figure 5.2 (a,b and c) represent the original signal, the 1-level Daub4 transform and the 1-level Daub6 transform, respectively. It is clear that the Daub4 fluctuation values are significantly larger in magnitude than the Daub6 fluctuation values. Also, Daub6 transform redistributes the energy of the signal into a more compact form comparing with Daub4 transdorm. Figure 5.3 (a,b) represent the 5-level Daub4 and 5-level Duab6 transforms for the signal in figure 5.2 (a).




(a) (b)

**Figure** 5.3

The following property is used in computing higher order Daub transforms.

**Property III** [25]**:** if $\mathbf{f}$ is (approximately) equal to polynomial of degree less than $J/2$ over the support of a k-level Daub$J$ wavelet $\mathbf{W}_m^k$, then the k-level fluctuation value $\mathbf{f}.\mathbf{W}_m{}^K$ is (approximately) zero.

The Daub $J$ transforms for $J = 8,10,\cdots,20$, are defined in the same way as with Daub4 and Daub6; using property III and orthonormality In general, Daub $J$ wavelet with a larger value for $J$ improves the MRA for smoother signals (signals sampled from analog signals having more differentiability).

However, such wavelets are not always the best as in the following example.

**Example 5.3:**

The signal:

$$g(x) = 20x^2(1-x)^4\cos(12\pi x) \qquad 0 \le x < 1$$

is sampled at $1024$ sample points then compressed using Haar, Daub4 and Daub20 wavelets. Table 5.1 shows the minimum number of transform values needed to capture $99.99\%$ of the energy of the signal.

**Table** 5.1

| Wavelet transform | Values for $99.99\%$ energy | Compression percentage |
|---|---|---|
| Haar | 453 | 56% |
| Daub4 | 194 | 81% |
| Daub20 | 193 | 81% |

It is clear from the previous table that the Daub4 and Daub20 transforms do the best job, while Haar transform is worst.

**Coiflet wavelets**

Now we will talk about another class of wavelets, the Coif*I* wavelets. These wavelets are designed to maintain a close match between the trend values and the original signal values. These wavelets were first constructed by Daubechies, who called them "Coiflets". All of Coif*I,* $I = 6,12,....$ wavelets are defined in similar way; so we will concentrate on the simplest case of Coif6 wavelets. Coif6 wavelet and scaling numbers are found using the orthonormality property and property III in the same way in Daub6. The difference is in the way the wavelet and scaling signals are defined.

The first-level Coif6 wavelets defined as follows:

$$\begin{aligned}
\mathbf{W}_1^1 &= (\beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0, \beta_1, \beta_2) \\
\mathbf{W}_2^1 &= (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0) \\
\mathbf{W}_3^1 &= (0, 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, 0, 0, ..., 0) \\
&\vdots \\
\mathbf{W}_{(N/2)-1}^1 &= (, 0, 0, 0, 0, ..., 0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6) \\
\mathbf{W}_{N/2}^1 &= (\beta_5, \beta_6, 0, 0, ..., 0, \beta_1, \beta_2, \beta_3, \beta_4)
\end{aligned} \tag{5.18}$$

Notice there are wrap-arounds for the first and the last wavelets $\mathbf{W}_{(N/2)-1}^1$ and $\mathbf{W}_{N/2}^1$.

The first-level Coif6 scaling signals are defined as:

$$\begin{aligned}
\mathbf{V}_1^1 &= (\alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, ..., 0, \alpha_1, \alpha_2) \\
\mathbf{V}_2^1 &= (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, ..., 0) \\
\mathbf{V}_3^1 &= (0, 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, 0, 0, ..., 0) \\
&\vdots \\
\mathbf{V}_{(N/2)-1}^1 &= (0, 0, 0, 0, ..., 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \\
\mathbf{V}_{N/2}^1 &= (\alpha_5, \alpha_6, 0, 0, 0, 0, ..., 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)
\end{aligned} \tag{5.19}$$

Again, there is wrap-around for $\mathbf{V}_{(N/2)-1}^1$ and $\mathbf{V}_{N/2}^1$.

Using the orthonormaility and property III, the wavelet numbers for the Coif6 wavelet signals can be found as:

$$\beta_1 = \frac{-3+\sqrt{7}}{16\sqrt{2}}, \qquad \beta_2 = \frac{-1+\sqrt{7}}{16\sqrt{2}}, \qquad \beta_3 = \frac{14-2\sqrt{7}}{16\sqrt{2}}$$

$$\beta_4 = \frac{-14-2\sqrt{7}}{16\sqrt{2}}, \qquad \beta_5 = \frac{5+\sqrt{7}}{16\sqrt{2}}, \qquad \beta_6 = \frac{-1+\sqrt{7}}{16\sqrt{2}}$$

The scaling numbers are related to the wavelet numbers by:

$$\alpha_1 = -\beta_6, \quad \alpha_2 = \beta_5, \quad \alpha_3 = -\beta_4, \quad \alpha_4 = \beta_3, \quad \alpha_5 = -\beta_2, \quad \alpha_6 = \beta_1 \tag{5.20}$$

The Coif6 scaling numbers satisfy the following identity

$$\alpha_1^{\,2} + \alpha_2^{\,2} + \alpha_3^{\,2} + \alpha_4^{\,2} + \alpha_5^{\,2} + \alpha_6^{\,2} = 1 \tag{5.21}$$

Identity (5.21) implies that each Coif6 scaling signal has energy 1. Moreover, the wavelet numbers satisfy

$$\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 = 0 \tag{5.22a}$$

$$0\beta_1 + 1\beta_2 + 2\beta_3 + 3\beta_4 + 4\beta_5 + 5\beta_6 = 0 \tag{5.22b}$$

Equations (5.22 a,b) show that Coif6 wavelet is similar to Daub4 wavelet, so it will produce a zero fluctuation value if the signal is linear over its support. However, the difference between Coif6 and Daub4 wavelet is the properties of the scaling numbers. The coif6 scaling numbers satisfy

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = \sqrt{2} \tag{5.23a}$$

$$-2\alpha_1 - 1\alpha_2 + 0\alpha_3 + 1\alpha_4 + 2\alpha_5 + 3\,\alpha_6 = 0 \tag{5.23b}$$

$$(-2)^2\alpha_1 + (-1)^2\,\alpha_2 + 0^2\alpha_3 + 1^2\alpha_4 + 2^2\alpha_5 + 3^2\,\alpha_6 = 0 \tag{5.23c}$$

Equation (5.23 a) implies that Coif6 trend values are averages of successive values of a signal $\mathbf{f}$ (with wrap-around when $\mathbf{f}.\mathbf{V}_1^1$ and $\mathbf{f}.\mathbf{V}_{N/2}^1$ are computed). Notice that no Daub$J$ scaling numbers satisfy any of the equations (5.23 b,c). However, these three equations have an important consequence. When a signal consists of sample values (equally spaced values) of an analog signal, then a Coif6 transform produces a much closer match between trend subsignals and the original signal values than can be obtained with any of the Daub$J$ transforms. Since there is a close match between trends and subsignal values, this means that the following approximations hold to a high degree of accuracy;

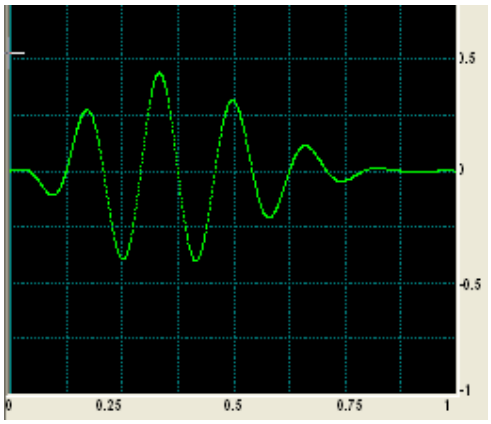
$$\begin{aligned} \mathbf{a}_m^1 &\approx \sqrt{2}\, g\left(t_{2m}\right) \\ \mathbf{a}_m^2 &\approx 2\, g\left(t_{4m}\right) \end{aligned} \tag{5.24}$$

This approximation holds for higher levels, but, in general, the accuracy decreases as the numbers of levels increases.

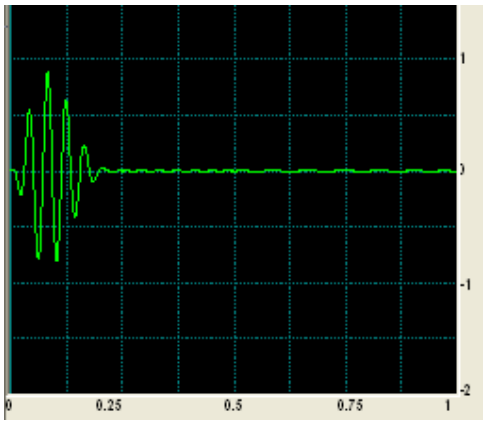
**Example 5.4:**

Consider the signal

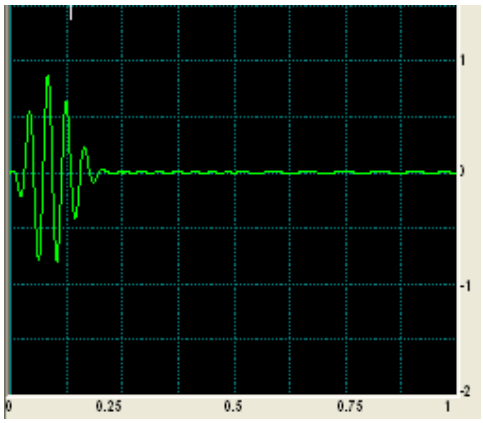
$$g(x) = 20x^2(1-x)^4 \cos(12\pi x),\ x \in [0,1)$$




(a)




(b)




(c)

**Figure** 5.4

Figure 5.4 (a) represents the signal $g$ which is obtained from $2^{14} = 16384$ sample values of the signal. Figure 5.4 (b,c) shows the 2-level Daub4 transform and the 2-level Coif6 transform for $g$, respectively. While these two figures seem very similar to each other, the maximum error between the 2-level Daub4 trend values and samples of $2g(4x)$ over the interval $[0, 0.25)$ is $3.76 \times 10^{-3}$ and the maximum error in the Coif6 case is $4.84 \times 10^{-7}$, which is much smaller. For the 1-level transforms we find that the maximum error between the first trend and samples of $\sqrt{2}\, g(t_{2m})$ is $8.87 \times 10^{-4}$ in the Daub4 case, and $8.59 \times 10^{-8}$ in the Coif6 case. This property of trends providing close approximations of the analog signal, which is shared by all the Coiflet transforms providing a useful means of interpolating the trend subsignals.

## 5.5 Compression of Audio Signals

A wavelet transform can be used to localize the energy of a signal into a shorter signal. Compressing an audio signal means converting the signal into a new format with smaller size to store or to transmit. In practice, most audio signals are discrete samples of a sound signal as in a computer audio file, or a compact disc.

**Example 5.5:**

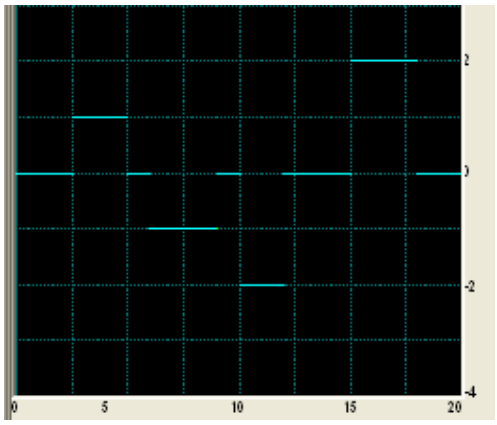
To illustrate the general principle of compression of signals in one dimension, we will use Haar wavelet transform to compress the following signal whose graph is given in figure 5.5 (a).

$$g(x)=\begin{cases}0, & 0\le x<2.5\\ 1, & 2.5\le x<5\\ 0, & 5\le x<6\\ -1. & 6\le x<9\\ 0, & 9\le x<10\\ -2, & 10\le x<12\\ 0, & 12\le x<15\\ 2, & 15\le x<18\\ 0, & 18\le x\le 20\end{cases}$$

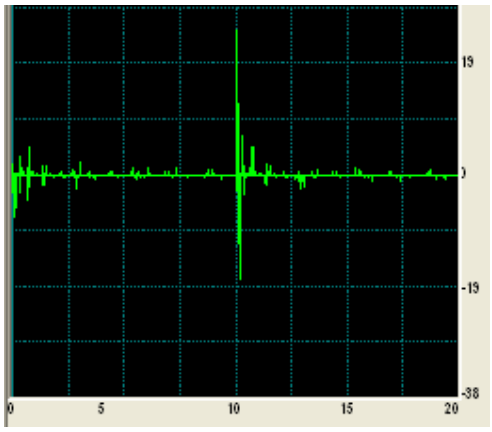
This signal can be compressed effectively by Haar transform since it is piecewise constant.This signal has $1024$ values over the time interval $[0,20]$. Figures 5.5 (b,c) shows that large portion of the 1-level and the 2-level Haar transform values are zeros or very close to zero. Table 5.2 shows the number of coefficients used to compress the signal with $99.99\%$ of its energy preserved using different levels.

**Table** 5.2

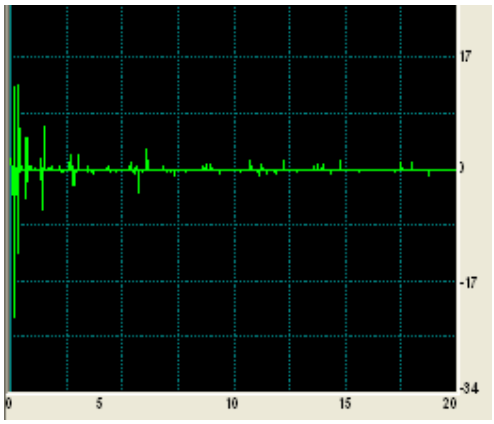
| Haar levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 272 | 3.8:1 |
| 4-level | 48 | 21.3:1 |
| 10-level | 32 | 32:1 |




(a)




(b)




(c)

**Figure** 5.5

n table 5.3, Daub4 is used. Comparison between the two tables shows that Haar transform is more efficient for this piecewise constant function.

**Table** 5.3

| Daub4 levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 279 | 3.7:1 |
| 4-level | 89 | 11.5:1 |
| 10-level | 96 | 10.7:1 |

The results obtained by Coif6 transform are close to those obtained using Daub4 as given in table 5.4.

**Table** 5.4

| Coif6 levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 281 | 3.6:1 |
| 4-level | 95 | 11:1 |
| 10-level | 100 | 10.4:1 |

According to tables 5.2, 5.3, and 5.4 we can see that Haar transform is the best method to compress the piecewise signal given in example 5.5.

However, the signal in the next example will not compressed nearly so well; this signal requires more sophisticated wavelet transform.

**Example 5.6:**

Consider the signal

$$g(x) = x^2 \sin(x) \quad \text{where} \quad x \in [0, 2\pi]$$

whose graph is given in figure 5.6.

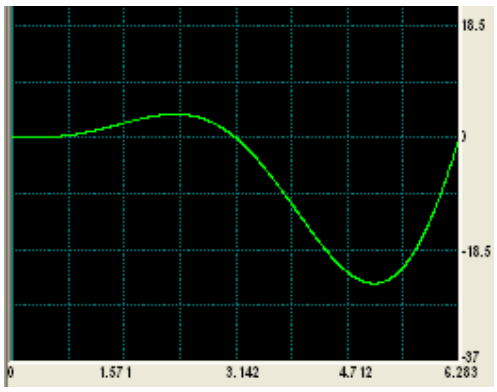



**Figure** 5.6

This signal can be compressed effectively at energy 99.99% with Coif6 transform, see table 5.5.

**Table** 5.5

| Coif6 levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 425 | 2.4:1 |
| 4-level | 54 | 18.9:1 |
| 10-level | 23 | 44.5:1 |

Moreover, Daub4 transform will compress the signal $g(x)$ in a effectively, see table 5.6.

**Table** 5.6

| Daub4 levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 425 | 2.4:1 |
| 4-level | 54 | 18.9:1 |
| 10-level | 25 | 41:1 |

Table 5.7 shows that Harr transform for this signal is much less efficitive than either of Daub4 or Coif6 transforms.

**Table** 5.7

| Haar levels | Number of coefficients | Ratio of compression |
|---|---|---|
| 1-level | 427 | 2.4:1 |
| 4-level | 125 | 8.2:1 |
| 10-level | 123 | 8.3:1 |

## 5.6 Two Dimensional Wavelet Transforms

Many basic ideas in 2D wavelet analysis are similar to the 1D case; so we will not repeat the similar formulas but we will focus on the ideas that are needed for 2D case, and we will describe some applications of 2D wavelet analysis.

### 5.6.1 Discrete Images

A digital image consists of number of pixels (picture elements) arranged in rows and columns. The 2D data that we will work with are discrete images. A discrete image $\mathbf{f}$ is an array of $M$ rows and $N$ columns of real numbers:

$$\mathbf{f} = \begin{pmatrix} f_{1,M} & f_{2,M} & \cdots & f_{N,M} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1,3} & f_{2,3} & \cdots & f_{N,3} \\ f_{1,2} & f_{2,2} & \cdots & f_{N,2} \\ f_{1,1} & f_{2,1} & \cdots & f_{N,1} \end{pmatrix} \tag{5.25}$$

Notice that the way the values of $\mathbf{f}$ are displayed in the above array is not the most commonly used one. But, we will use it because it corresponds with the case where $\mathbf{f}$ is an array of sampled values.

$$\mathbf{f} = \begin{pmatrix} g_{(x_1, y_M)} & g_{(x_2, y_M)} & \cdots & g_{(x_N, y_M)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{(x_1, y_2)} & g_{(x_2, y_2)} & \cdots & g_{(x_N, y_2)} \\ g_{(x_1, y_2)} & g_{(x_2, y_2)} & \cdots & g_{(x_N, y_2)} \\ g_{(x_1, y_1)} & g_{(x_2, y_1)} & \cdots & g_{(x_N, y_1)} \end{pmatrix} \tag{5.26}$$

The above matrix shows $\mathbf{f}$ as a two dimensional array of sample values of a function of two variables $g(x, y)$ at the sample points $(x_j, y_k)$ in the Cartesian coordinates plane.

We can view a discrete image in one of two ways. First, as a single column consists of $M$ 1D signals having length $N$,

$$\mathbf{f} = \begin{pmatrix} \mathbf{f}_M \\ \vdots \\ \mathbf{f}_2 \\ \mathbf{f}_1 \end{pmatrix} \tag{5.27}$$

where the signals are:

$$\mathbf{f}_M = (f_{1,M}, f_{2,M}, \cdots, f_{N,M})$$
$$\vdots$$
$$\mathbf{f}_2 = (f_{1,2}, f_{2,2}, \cdots, f_{N,2})$$
$$\mathbf{f}_1 = (f_{1,1}, f_{2,1}, \cdots, f_{N,1})$$

Second, as a single row consists of $N$ 1D signals of length $M$

$$\mathbf{f} = (\mathbf{f}^1, \mathbf{f}^2, \cdots, \mathbf{f}^N) \tag{5.28}$$

with the columns being the signals

$$\mathbf{f}^1 = \begin{pmatrix} f_{1,M} \\ \vdots \\ f_{1,2} \\ f_{1,1} \end{pmatrix} \qquad \mathbf{f}^2 = \begin{pmatrix} f_{2,M} \\ \vdots \\ f_{2,2} \\ f_{2,1} \end{pmatrix} \qquad \cdots \qquad \mathbf{f}^N = \begin{pmatrix} f_{N,M} \\ \vdots \\ f_{N,2} \\ f_{N,1} \end{pmatrix}$$

Notice that the row index for each column increases from bottom to the top instead from the top to the bottom which is more common notation in image processing.

The energy $\varepsilon_{\mathbf{f}}$ of a discrete image $\mathbf{f}$ is defined to be the sum of the squares of all of its values. It follows that $\varepsilon_{\mathbf{f}}$ is the sum energies of the row signals in (5.27);

$$\begin{aligned} \varepsilon_{\mathbf{f}} &= \varepsilon_{\mathbf{f}_1} + \varepsilon_{\mathbf{f}_2} + \cdots + \varepsilon_{\mathbf{f}_M} \\ &= \sum_{j=1}^{M} \sum_{i=1}^{N} f_{ij}^2 \end{aligned} \quad .$$

Also, the energy $\varepsilon_{\mathbf{f}}$ is the sum of the energies of all of columns signals;

$$\begin{aligned} \varepsilon_{\mathbf{f}} &= \varepsilon_{\mathbf{f}^1} + \varepsilon_{\mathbf{f}^2} + \cdots + \varepsilon_{\mathbf{f}^N} \\ &= \sum_{j=1}^{N} \sum_{i=1}^{M} f_{ij}^2 \end{aligned}$$

As a result from the last two identities the 2D wavelet transform below conserves the energy of discrete images.

### 5.6.2 Wavelet Transform of Discrete Images

A digital image can be represented by a matrix whose entries represent the light intensity. The 1-level wavelet transform of a 2D image with even number of rows and columns can be defined as follows:

Step 1: Perform 1-level, 1D wavelet transform, on each row of $\mathbf{f}$ , this will produce a new image.

Step 2: Perform the same 1D wavelet transform on each column on the new image obtained from step1.

Now, A 1-level wavelet transform of an image $\mathbf{f}$ can be symbolized as follows:

$$\mathbf{f} \to \left( \begin{array}{c|c} \mathbf{h}^1 & \mathbf{d}^1 \\ \hline \mathbf{a}^1 & \mathbf{v}^1 \end{array} \right) \tag{5.29}$$

where the subimages $\mathbf{a}^1, \mathbf{h}^1, \mathbf{d}^1,$ and $\mathbf{v}^1$ each have $M/2$ rows and $N/2$ columns.

The subimage $\mathbf{a}^1$ is created by computing trends along rows of $\mathbf{f}$ followed by computing trends along columns; so it is averaged, lower resolution version of the image $\mathbf{f}$. Since a 1D trend computation is $\sqrt{2}$ times an average of successive values in a signal, and the 2D trend subimage $\mathbf{a}^1$ was computed from trends along both rows and columns, it follows that each values of $\mathbf{a}^1$ is equal to 2 times an average of a small square containing adjacent values from the image $\mathbf{f}$. We will express $\mathbf{a}^1$ as a scalar products of the image $f$ with scaling signals, as we did in 1D case.

The $\mathbf{h}^1$ subimage is created by computing trends along rows of the image followed by computing fluctuations along columns. Consequently, whenever there are horizontal edges in an image, the fluctuations along columns are able to detect these edges. We shall refer to this subimage as the first horizontal fluctuation. The subimage $\mathbf{v}^1$ is similar to $\mathbf{h}^1$. We shall refer to this subimage as the first vertical fluctuation. The subimage $\mathbf{d}^1$ is created by computing the fluctuations along both rows and columns. We shall refer to this subimage as the first diagonal fluctuation, since it tends to emphasize diagonal features.

Notice that, the basic principle for 1D wavelet analysis still apply here in 2D setting. For example, the fact that the fluctuation values are generally much smaller than trend values is still true. So, the fluctuation subimages $\mathbf{h}^1, \mathbf{d}^1,$ and $\mathbf{v}^1$ have significantly smaller values than the values in the trend subimage $\mathbf{a}^1$. In fact, to make the values for $\mathbf{h}^1, \mathbf{d}^1,$ and $\mathbf{v}^1$ visible, they are displayed on a logarithmic intensity scale, while the values for the trend subimage $\mathbf{a}^1$ are displayed using an ordinary linear scale. Also, 2D wavelet transforms conserves energy. The energy of an image is the sum of the energies of each of its rows or each of its columns. Since, 1D wavelet transforms of the rows preserve energies, the image obtained in step1 will have the same energy as the original image. Also, since the 1D wavelet transforms of the columns preserve their energies, it follows that the transform obtained in step2 has the same energy is the image from step1. So, the 1-level wavelet transform has the same energy as the image in step1.

The multiple levels of 2D wavelet transforms are defined by repeating the 1-level transform of the previous trend. For example, a 2-level wavelet transform is performed by computing a 1-level transform of the trend subimage $\mathbf{a}^1$ as follows:

$$\mathbf{a}^1 \rightarrow \left( \begin{array}{c|c} \mathbf{h}^2 & \mathbf{d}^2 \\ \hline \mathbf{a}^2 & \mathbf{v}^2 \end{array} \right)$$

The 1-level fluctuations $\mathbf{h}^1, \mathbf{d}^1,$ and $\mathbf{v}^1$ remain unchanged. In general, a k-level transform is defined by performing a 1-level transform on the previous trend subimage $\mathbf{a}^{k-1}$. Notice that the 2D wavelet transform perform compaction and conservation of the energy. Most of the energy of the image is successively localized into smaller and smaller trend subimages.

**Example 5.7:**

Consider the following discrete image:

$$\mathbf{f} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix}$$

We apply the 1-level Haar Transform in 2D as follows:

Computing the trend and the fluctuation subimages along rows as in (4.18) and (4.19) respectively, we get:

$$\begin{pmatrix} 8/\sqrt{2} & 6/\sqrt{2} & -4/\sqrt{2} & 2/\sqrt{2} \\ 14/\sqrt{2} & 14/\sqrt{2} & -6/\sqrt{2} & 2/\sqrt{2} \\ 22/\sqrt{2} & 26/\sqrt{2} & -2/\sqrt{2} & 2/\sqrt{2} \\ 14/\sqrt{2} & 14/\sqrt{2} & -2/\sqrt{2} & 6/\sqrt{2} \end{pmatrix}$$

Repeat the same on columns to obtain the transformed image

$$\hat{\mathbf{f}} = \left(\begin{array}{cc|cc} 3 & 4 & -1 & 0 \\ -4 & -6 & 0 & 2 \\ \hline 11 & 10 & -5 & 2 \\ 18 & 20 & -2 & 4 \end{array}\right)$$

The energy for the original image is:

$$\varepsilon_{\mathbf{f}} = 6^2 + 8^2 + 10^2 + 4^2 + 10^2 + 12^2 + 14^2 + 12^2 + 4^2 + 10^2 + 8^2 + 6^2 + 2^2 + 6^2 + 4^2 + 2^2 = 1076$$

And the energy for the transformed image is:

$$\varepsilon_{\hat{\mathbf{f}}} = 18^2 + 20^2 + (-2)^2 + 4^2 + 11^2 + 10^2 + (-5)^2 + 2^2 + (-4)^2 + (-6)^2 + 2^2 + 3^2 + 4^2 + (-1)^2 = 1076$$

The 1-level Haar transform preserved the energy, while the energy of its first trend subsignal is:

$$\varepsilon_{\mathbf{a}^1} = 18^2 + 20^2 + 11^2 + 10^2 = 945$$

So, $87.83\%$ of the total energy is compacted into the left bottom corner subimage $\mathbf{a}^1$.

Moreover, 2-level Haar transform in 2D performed by computing a 1-level transform of the trend subimage $\mathbf{a}^1$ (repeat the previous procedure)

$$\left(\begin{array}{c|c} 21/\sqrt{2} & 1/\sqrt{2} \\ \hline 38/\sqrt{2} & -2/\sqrt{2} \end{array}\right) \rightarrow \left(\begin{array}{c|c} 17/2 & -3/2 \\ \hline 59/2 & -1/2 \end{array}\right)$$

Its energy is $29.5^2 + (-.5)^2 + 8.5^2 + (-1.5)^2 = 945$ which is the same as the energy of the 1-level trend subimage $\mathbf{a}^1$. The energy for the subimage $\mathbf{a}^2$ is $29.5^2 = 870.25$. So, $92.1\%$ of the total energy is compacted into the left bottom corner subimage $\mathbf{a}^2$.

### 5.6.3 Wavelet and Scaling Images

In 1D case, the various levels of wavelet transform can be computed by scalar products of the image $\mathbf{f}$ with elementary images called scaling images and wavelets.

**Definition 5.1** [25]**:** A scalar product of two discrete images $\mathbf{f}$ and $\mathbf{g}$, both having $M$ rows and $N$ columns, is defined by:

$$\mathbf{f}.\mathbf{g} = f_{1,1}g_{1,1} + f_{1,2}g_{1,2} + \cdots + f_{N,M}g_{N,M} \tag{5.30}$$

In other words, $\mathbf{f}.\mathbf{g}$ is the sum of all products of the same indexed values of $\mathbf{f}$ and $\mathbf{g}$.

**Example 5.8:**

Consider the two images $\mathbf{f} = \begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix}$ and $\mathbf{g} = \begin{bmatrix} 3 & 6 \\ 1 & 2 \end{bmatrix}$.

$$\mathbf{f}.\mathbf{g} = 2\times 3 + 3\times 6 + 5\times 1 + 4\times 2 = 37$$

The 1-level wavelet transform of a 2D image consists of the 4 components; $\mathbf{a}^1$, $\mathbf{h}^1, \mathbf{d}^1,$ and $\mathbf{v}^1$ as in (5.29). These components can be defined in terms of elementary images, but, first we need the following definitions:

**Definition 5.2** [25]**: Tensor Product**

The tenser product of two vectors is defined as :

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{pmatrix} \otimes \begin{pmatrix} b_1 & b_2 & b_3 & \cdots & b_M \end{pmatrix} = \begin{pmatrix} a_1b_1 & a_1b_2 & a_1b_3 & \cdots & a_1b_M \\ a_2b_1 & a_2b_2 & a_2b_3 & \cdots & a_2b_M \\ a_3b_1 & a_3b_2 & a_3b_3 & \cdots & a_3b_M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_Nb_1 & a_Nb_2 & a_Nb_3 & \cdots & a_Nb_M \end{pmatrix}$$

**Definition 5.3** [25]**:**

We define the following elementary images as:

a. Elementary wavelet: $\mathbf{W}^1_{m,n} = \mathbf{W}^1_m \otimes \mathbf{W}^1_n$.

b. Scaling image: $\mathbf{V}^1_{m,n} = \mathbf{V}^1_m \otimes \mathbf{V}^1_n$.

c. Horizontal image: $\mathbf{H}^1_{m,n} = \mathbf{W}^1_m \otimes \mathbf{V}^1_n$.

d. Vertical image: $\mathbf{D}^1_{m,n} = \mathbf{V}^1_m \otimes \mathbf{W}^1_n$.

Using the Tensor product we will compute the 1-level elementary wavelet $\mathbf{W}^1_1$, 1-level scaling image $\mathbf{V}^1_1$, 1-level horizontal image $\mathbf{H}^1_1$ and 1-level vertical image $\mathbf{D}^1_1$, respectively.

$$\mathbf{W}^1_{1,1} = \mathbf{W}^1_1 \otimes \mathbf{W}^1_1$$

$$\mathbf{W}^1_1 \otimes \mathbf{W}^1_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, -1/\sqrt{2}, 0, 0, \cdots, 0\right) = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ -1/2 & 1/2 & 0 & 0 & \cdots & 0 \\ 1/2 & -1/2 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Similarly, we compute:

$$\mathbf{V}_{1,1}^1 = \mathbf{V}_1^1 \otimes \mathbf{V}_1^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 1/2 & 1/2 & 0 & 0 & \cdots & 0 \\ 1/2 & 1/2 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$\mathbf{H}_{1,1}^1 = \mathbf{W}_1^1 \otimes \mathbf{V}_1^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 1/2 & -1/2 & 0 & 0 & \cdots & 0 \\ 1/2 & -1/2 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$\mathbf{D}_{1,1}^1 = \mathbf{V}_1^1 \otimes \mathbf{W}_1^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ -1/2 & -1/2 & 0 & 0 & \cdots & 0 \\ 1/2 & 1/2 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Notice that the Haar wavelet $\mathbf{W}_{1,1}^1, \mathbf{H}_{1,1}^1$ and $\mathbf{D}_{1,1}^1$ all have energy 1 and an average value of 0, like all the other Haar wavelets $\mathbf{W}_{m,n}^1, \mathbf{H}_{m,n}^1$, and $\mathbf{D}_{m,n}^1$ . Moreover, notice that the support of the Haar wavelet $\mathbf{W}_{1,1}^1, \mathbf{H}_{1,1}^1$ and $\mathbf{D}_{1,1}^1$ is a 2 by 2 square, and so the support of each Haar wavelet $\mathbf{W}_{m,n}^1, \mathbf{H}_{m,n}^1$ and $\mathbf{D}_{m,n}^1$ is a 2 by 2 square as well. But, for the Daubechies wavelets, the supports of the wavelets $\mathbf{W}_{1,1}^1, \mathbf{H}_{1,1}^1$ and $\mathbf{D}_{1,1}^1$ are not $2 \times 2$ ones.

Like all the other Haar scaling images $\mathbf{V}_{m,n}^1$.the scaling signal $\mathbf{V}_{1,1}^1$ has energy 1 and average value $1/2$, Moreover, notice that the support of the Haar scaling image $\mathbf{V}_{1,1}^1$ is a 2 by 2 square, and so the support of each Haar scaling image $\mathbf{V}_{m,n}^1$ is a 2 by 2 square as well. But, the supports of the scaling image $\mathbf{V}_{m,n}^1$ for the Daubechies wavelets are small squares, but not $2 \times 2$ ones.

**Definition 5.4:**

a. The entries of the 1-level scaling image $\mathbf{a}^1$ are defined as:

$$a^1_{m,n} = \mathbf{f}.\mathbf{V}^1_{m,n}$$

b. The entries of the 1-level horizontal fluctuation $\mathbf{h}^1$ are defined as:

$$h^1_{m,n} = \mathbf{f}.\ \mathbf{H}^1_{m,n}$$

c. The entries of the 1-level diagonal fluctuation $\mathbf{d}^1$ are defined as:

$$d^1_{m\,n} = \mathbf{f}.\mathbf{W}^1_{m,n}$$

d. The entries of the 1-level scaling image $\mathbf{v}^1$ are defined as:

$$v^1_{m,n} = \mathbf{f}.\ \mathbf{D}^1_{m,n}$$

**Example 5.9:**

We use 1-level Haar transform to transform the image

$$f = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix}$$

Using definition (5.2, 5.3) and (5.4) compute the 1-level scaling image $\mathbf{a}^1$, the 1-level horizontal fluctuation $\mathbf{h}^1$, the 1-level diagonal fluctuation $\mathbf{d}^1$, and the 1-level scaling image $\mathbf{v}^1$,

First, compute the four entries in the subimage $\mathbf{a}^1$:

$$\mathbf{V}_{1,1}^{1} = \mathbf{V}_{1}^{1} \otimes \mathbf{V}_{1}^{1} = \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left( 1/\sqrt{2}, 1/\sqrt{2}, 0, 0 \right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

$$a_{1,1}^{1} = \mathbf{f}.\mathbf{V}_{1,1}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix} = 18$$

Similarly;

$$\mathbf{V}_{1,2}^{1} = \mathbf{V}_{1}^{1} \otimes \mathbf{V}_{2}^{1} = \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left( 0, 0, 1/\sqrt{2}, 1/\sqrt{2} \right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

$$a_{1,2}^{1} = \mathbf{f}.\mathbf{V}_{1,2}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix} = 20$$

$$\mathbf{V}_{2,1}^{1} = \mathbf{V}_{2}^{1} \otimes \mathbf{V}_{1}^{1} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left( 1/\sqrt{2}, 1/\sqrt{2}, 0, 0 \right) = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$a_{2,1}^{1} = \mathbf{f}.\mathbf{V}_{2,1}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 11$$

$$\mathbf{V}_{2,2}^1 = \mathbf{V}_2^1 \otimes \mathbf{V}_2^1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left(0, 0, 1/\sqrt{2}, 1/\sqrt{2}\right) = \begin{pmatrix} 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$a_{2,2}^1 = \mathbf{f}.\mathbf{V}_{2,2}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 10$$

Second, compute the four entries in the subsignal $\mathbf{v}^1$:

$$\mathbf{D}_{1,1}^1 = \mathbf{V}_1^1 \otimes \mathbf{W}_1^1 = \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, -1/\sqrt{2}, 0, 0\right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \end{pmatrix}$$

$$v_{1,1}^1 = \mathbf{f}.\mathbf{D}_{1,1}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \end{pmatrix} = -2$$

$$\mathbf{D}_{1,2}^1 = \mathbf{V}_1^1 \otimes \mathbf{W}_2^1 = \begin{pmatrix} 0 \\ 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(0, 0, 1/\sqrt{2}, -1/\sqrt{2}\right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 1/2 & -1/2 \end{pmatrix}$$

$$v_{1,2}^1 = \mathbf{f}.\mathbf{D}_{1,2}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 1/2 & -1/2 \end{pmatrix} = 4$$

$$\mathbf{D}_{2,1}^1 = \mathbf{V}_2^1 \otimes \mathbf{W}_1^1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left(1/\sqrt{2}, -1/\sqrt{2}, 0, 0\right) = \begin{pmatrix} 1/2 & -1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$v_{2,1}^1 = \mathbf{f}.\mathbf{D}_{2,1}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & -1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = -5$$

$$\mathbf{D}_{2,2}^1 = \mathbf{V}_2^1 \otimes \mathbf{W}_2^1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left(0, 0, 1/\sqrt{2}, -1/\sqrt{2}\right) = \begin{pmatrix} 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$v_{2,2}^1 = \mathbf{f}.\mathbf{D}_{2,2}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 2$$

Third, compute the four entries of the subsignal $\mathbf{h}^1$:

$$\mathbf{H}_{1,1}^1 = \mathbf{W}_1^1 \otimes \mathbf{V}_1^1 = \begin{pmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, 1/\sqrt{2}, 0, 0\right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2 & -1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

$$h_{1,1}^1 = \mathbf{f}.\mathbf{H}_{1,1}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2 & -1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix} = -4$$

$$\mathbf{H}_{1,2}^1 = \mathbf{W}_1^1 \otimes \mathbf{V}_2^1 = \begin{pmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(0, 0, 1/\sqrt{2}, 1/\sqrt{2}\right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/2 & -1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

$$h_{1,2}^1 = \mathbf{f}.\mathbf{H}_{1,2}^1 = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/2 & -1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix} = -6$$

$$\mathbf{H}_{2,1}^{1} = \mathbf{W}_{2}^{1} \otimes \mathbf{V}_{1}^{1} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left( 1/\sqrt{2}, 1/\sqrt{2}, 0, 0 \right) = \begin{pmatrix} -1/2 & -1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$h_{2,1}^{1} = \mathbf{f}.\mathbf{H}_{2,1}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} -1/2 & -1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 3$$

$$\mathbf{H}_{2,2}^{1} = \mathbf{W}_{2}^{1} \otimes \mathbf{V}_{2}^{1} = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \\ 0 \end{pmatrix} \otimes \left( 0, 0, 1/\sqrt{2}, 1/\sqrt{2} \right) = \begin{pmatrix} 0 & 0 & -1/2 & -1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$h_{2,2}^{1} = \mathbf{f}.\mathbf{H}_{2,2}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & -1/2 & -1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = 4$$

Fourth, compute the four entries of the subsignal $\mathbf{d}^{1}$:

$$\mathbf{W}_{1,1}^{1} = \mathbf{W}_{1}^{1} \otimes \mathbf{W}_{1}^{1}$$

$$= \begin{pmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left( 1/\sqrt{2}, -1/\sqrt{2}, 0, 0 \right) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \end{pmatrix}$$

$$d_{1,1}^{1} = \mathbf{f}.\mathbf{W}_{1,1}^{1} = \begin{pmatrix} 2 & 6 & 4 & 2 \\ 4 & 10 & 8 & 6 \\ 10 & 12 & 14 & 12 \\ 6 & 8 & 10 & 4 \end{pmatrix} . \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 \end{pmatrix} = 0$$

$$\mathbf{W}_{1,2}^{1}=\mathbf{W}_{1}^{1}\otimes\mathbf{W}_{2}^{1}=\begin{pmatrix}0\\0\\-1/\sqrt{2}\\1/\sqrt{2}\end{pmatrix}\otimes\left(0,0,1/\sqrt{2},-1/\sqrt{2}\right)=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&-1/2&1/2\\0&0&1/2&-1/2\end{pmatrix}$$

$$d_{1,2}^{1}=\mathbf{f}.\mathbf{W}_{1,2}^{1}=\begin{pmatrix}2&6&4&2\\4&10&8&6\\10&12&14&12\\6&8&10&4\end{pmatrix}.\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&-1/2&1/2\\0&0&1/2&-1/2\end{pmatrix}=2$$

$$\mathbf{W}_{2,1}^{1}=\mathbf{W}_{2}^{1}\otimes\mathbf{W}_{1}^{1}==\begin{pmatrix}-1/\sqrt{2}\\1/\sqrt{2}\\0\\0\end{pmatrix}\otimes\left(1/\sqrt{2},-1/\sqrt{2},0,0\right)=\ =\begin{pmatrix}-1/2&1/2&0&0\\1/2&-1/2&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}$$

$$d_{2,1}^{1}=\mathbf{f}.\mathbf{W}_{2,1}^{1}=\begin{pmatrix}2&6&4&2\\4&10&8&6\\10&12&14&12\\6&8&10&4\end{pmatrix}.\begin{pmatrix}-1/2&1/2&0&0\\1/2&-1/2&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}=-1$$

$$\mathbf{W}_{2,2}^{1}=\mathbf{W}_{2}^{1}\otimes\mathbf{W}_{2}^{1}=\begin{pmatrix}-1/\sqrt{2}\\1/\sqrt{2}\\0\\0\end{pmatrix}\otimes\left(0,0,1/\sqrt{2},-1/\sqrt{2}\right)=\begin{pmatrix}0&0&-1/2&1/2\\0&0&1/2&-1/2\\0&0&0&0\\0&0&0&0\end{pmatrix}$$

$$d_{2,2}^{1}=\mathbf{f}.\mathbf{W}_{2,2}^{1}=\begin{pmatrix}2&6&4&2\\4&10&8&6\\10&12&14&12\\6&8&10&4\end{pmatrix}.\begin{pmatrix}0&0&-1/2&1/2\\0&0&1/2&-1/2\\0&0&0&0\\0&0&0&0\end{pmatrix}=0$$

The transformed signal will be on the form:

$$\left(\begin{array}{cc|cc}h_{2,1}&h_{2,2}&d_{2,1}&d_{2,2}\\h_{1,1}&h_{2,1}&d_{1,1}&d_{1,2}\\\hline a_{2,1}&a_{2,2}&v_{2,1}&v_{2,2}\\a_{1,1}&a_{1,2}&v_{1,1}&v_{1,2}\end{array}\right)$$

The transformed signal is:

$$\left(\begin{array}{cc|cc} 3 & 4 & -1 & 0 \\ -4 & -6 & 0 & 2 \\ \hline 11 & 10 & -5 & 2 \\ 18 & 20 & -2 & 4 \end{array}\right)$$

The energy for the original signal $\mathbf{f}$ and the transformed signal using 1-level Haar transform in 2D is the same 1076. To obtain 2-level Haar transform we will repeat the previous procedure on trend subsignal $\mathbf{a}^1$ which contains most of the energy.

First, compute the only entry of the subsignal $\mathbf{a}^2$:

$$\mathbf{V}_{1,1}^2 = \mathbf{V}_1^1 \otimes \mathbf{V}_1^1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, 1/\sqrt{2}\right) = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

$$a_{1,1}^2 = \mathbf{a}^1 . \mathbf{V}_{1,1}^2 = \begin{pmatrix} 11 & 10 \\ 18 & 20 \end{pmatrix} . \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} = 59/2$$

Second, compute the only entry of the subsignal $\mathbf{v}^2$:

$$\mathbf{D}_{1,1}^2 = \mathbf{V}_1^1 \otimes \mathbf{W}_1^1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, -1/\sqrt{2}\right) = \begin{pmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{pmatrix}$$

$$v_{1,1}^2 = \mathbf{a}^1 . \mathbf{D}_{1,1}^2 = \begin{pmatrix} 11 & 10 \\ 18 & 20 \end{pmatrix} . \begin{pmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{pmatrix} = -1/2$$

Third, compute the only entry of the subsignal $\mathbf{h}^2$:

$$\mathbf{H}_{1,1}^2 = \mathbf{W}_1^1 \otimes \mathbf{V}_1^1 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left(1/\sqrt{2}, 1/\sqrt{2}\right) = \begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

$$h_{1,1}^2 = \mathbf{a}^1 . \mathbf{H}_{1,1}^1 = \begin{pmatrix} 11 & 10 \\ 18 & 20 \end{pmatrix} . \begin{pmatrix} -1/2 & -1/2 \\ 1/2 & 1/2 \end{pmatrix} = 17/2$$

Fourth, compute the only entry of the subsignal $\mathbf{d}^1$:

$$\mathbf{W}_{1,1}^2 = \mathbf{W}_1^1 \otimes \mathbf{W}_1^1 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \otimes \left( 1/\sqrt{2}, -1/\sqrt{2} \right) = \begin{pmatrix} -1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix}$$

$$d_{1,1}^2 = \mathbf{a}^1 . \mathbf{W}_{1,1}^2 = \begin{pmatrix} 11 & 10 \\ 18 & 20 \end{pmatrix} . \begin{pmatrix} -1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix} = -3/2$$

So, the 2-level Haar transform of the original signal $\mathbf{f}$ is:

$$\left( \frac{11}{18} \middle| \frac{10}{20} \right) \rightarrow \left( \frac{17/2}{59/2} \middle| \frac{-3/2}{-1/2} \right)$$

What is true for the first level remains true for all subsequent levels. The values of each subimage $\mathbf{a}^k, \mathbf{h}^k, \mathbf{d}^k$ and $\mathbf{v}^k$ are computed by scalar products with the scaling images $\mathbf{W}_{m,n}^k$, and the wavelets $\mathbf{D}_{m,n}^k$, $\mathbf{H}_{m,n}^k$, and $\mathbf{V}_{m,n}^k$, respectively.

## 5.7 Applications

In these sections, we discuss some major image compression applications of wavelets.

### 5.7.1 Finger prints

The Federal Bureau of Investigation (FBI) in the USA launched a project to store and process tens of millions of sets of finger prints electronically. The FBI project has boosted the research on wavelets.

The FBI has been collecting fingerprints since 1924 [5]. By the year 1994, the total number of fingerprints reached over 200 million. Over 30

million set of fingerprints are searched each time they investigate the records of a suspect. The list is updated at a rate of about 30000 fingerprint per day. The processing is manual and time consuming. However, storing over 200 million fingerprint set on a computer, and searching over 30 million fingerprint set require large storage space since each fingerprint set occupy about 10 Mb and this require long time when transmitting data between different offices of the FBI. Still, processing and comparison is done manually. The FBI started searching for methods to reduce the size of the storage space and to automate the process of fingerprints. The FBI adopted wavelet transform for the compression and processing of fingerprints. While Fourier transform can be used in compressing and processing such huge number of fingerprint sets, wavelet transform proved to be more efficient.

Another use of wavelets is in using fingerprints or the unique type of hemoglobin in blood to allow or refuse entries into high security buildings. In Singapore, a new security system based on wavelets transforms was introduced in Hitachi Tower in 2003. Later, 1500 employees got access to the building by scanning their fingers. The scanner uses infrared rays to trace the hemoglobin in blood in order to capture the vein patterns in the finger, and then compare them to a pre-stored image of vein patterns of the person. These patterns determine the person uniquely. The whole process need to be fast to allow or deny entry of a given person.

### 5.7.2 Compression of Images

One of the most important applications of wavelets is image compression. Since an image is 2D signal, the same techniques for 2D signals are used. To illustrate, we consider the following example.

**Example 5.10:**

The image in figure 5.7 (a) is an example used very often in wavelet analysis. It shows a $512 \times 512$ image of Lena. This image is gray– scale with light intensity values from 0 to 255 (0 indicates pure black and 255 indicates pure white, and other values indicates shades of gray, between these two extremes).To compress Lena image in figure 5.7 (a) we shall use its 4-level Coif12 transform;


(a) (b)

**Figure** 5.7

Figure 5.7 (b) represents the 4-level Coif12 transform for Lena image, notice that the transform is very small compared to the original image since in the transform decreases the length and the width of the image according to the number of the levels, so the image will be decreased.

**Example 5.11:**

For the same image from example 5.10 we will compress it using 2-level Haar wavelet series

(a)

(b)

(c)

(d)

(e)

**Figure** 5.8

Figure 5.8 (a,b,c,d,e) represent the original image and the compressed images using 2-level Haar wavelet series, where the thresholds $1/2$, $1/2^3$, $1/2^4$, and $1/2^8$ respectively.

Figure 5.8 (a) represents the original image where 263169 samples are used and its energy is 4634258609.

Figure 5.8 (b) represents the reconstructed image after the 2-level Haar wavelet series, where the threshold is $1/2$, its energy is 4631351017.68097, 3.7% of the coefficients used, 0.07 bits per pixel and the compression ratio is 0.9993725877.

Figure 5.8 (c) represents the reconstructed image after the 2-level Haar wavelet series, where the threshold is $1/2^3$, its energy is 4641370350.85884, 6.4% of the coefficients used, 0.25 bits per pixel and the compression ratio is 1.001534602.

Figure 5.8 (d) represents the reconstructed image after the 2-level Haar wavelet series, where the threshold is $1/2^4$, its energy is 4618888060.62109, 7.% of the coefficients used, 0.35 bits per pixel and the compression ratio is 0.9966832779.

Figure 5.8 (e) represents the reconstructed image after the 2-level Haar wavelet series, where the threshold is $1/2^8$, its energy is 4633945785.05686, 40. % of the coefficients used 3.6 bits per pixel and the compression ratio is 0.9999324975.

**Example 5.12:**

For the same image from example 5.10 we will compress it using 2-level Daub 4 wavelet series

(a)

(b)

(c)

(d)

(e)

**Figure** 5.9

Figure 5.9 (a,b,c,d,e) represent the original image and the compressed images using 2-level Daub4 wavelet series, with thresholds $1/2$, $1/2^3$, $1/2^4$, and $1/2^8$ respectively.

Figure 5.9 (a) represents the original image where 263169 sample used and its energy is 4634258609.

Figure 5.9 (b) represents the reconstructed image after the 2-level Daub4 wavelet series, where the threshold is $1/2$, its energy is 4768905656.29883, 3.6% of the coefficients used 0.07 bits per pixel and the compression ratio is 1.029054712.

Figure 5.9 (c) represents the reconstructed image after the 2-level Daub4 wavelet series, where the threshold is $1/2^3$, its energy is 4649671220.0748, 6.3% of the coefficients used 0.25 bits per pixel and the compression ratio is 1.003325799.

Figure 5.9 (d) represents the reconstructed image after the 2-level Daub4 wavelet series, where the threshold is $1/2^4$, its energy is 4622571014.11522, 6.7% of the coefficients used 0.34 bits per pixel and the compression ratio is 0.9974780011.

Figure 5.9 (e) represents the reconstructed image after the 2-level Daub4 wavelet series, where the threshold is $1/2^8$, its energy is 4634029848.74203, 35.2% of the coefficients used 3.17 bits per pixel and the ratio compression is 0.9999506372.

From examples 5.10 and 5.11 we notice that the lower thresholding (the more the ratio of bit rate increases) the less compression of image we obtain lower average speed of transferring data, but better quality of the reconstructed image we get.

**Example 5.13:**

For the same image from example 5.10 we will compress it using 3-level Haar and Daub4 and Coif6 wavelet series at a fixed thresholding value

(a)

(b)

(b')

(b")

(c)

(c')

(c")

**Figure** 5.10

Figure 5.10 (a) represents the original image and figure 5.10 (b,b',b") represent the reconstructed images using 3-level Haar, Daub 4 and Coif 6 wavelet series at a fixed thresholding value which is $\frac{1}{2}$ respectively, figure 5.10 (b) its energy is 4603786336, 1.% of the coefficients used, 0.02 bits per pixel and the compression ratio is 0.9934245632. While figure 5.10 (b') its energy is 4794384050.90519, 0.9% of the coefficients used, 0.02 bits per pixel and the compression ratio is 1.034552548. Also, figure 5.16 (b") its

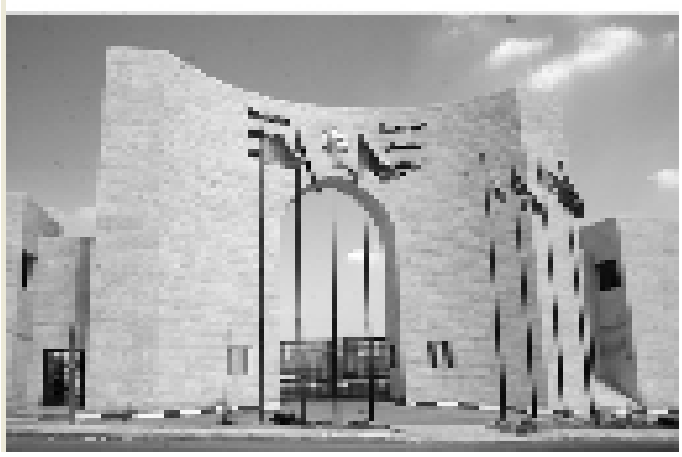
energy is 4671914069.34679 , 0.9% of the coefficients used, 0.02 bits per pixel and the compression ratio is 1.008125455.

Figure 5.10 (c,c',c") represent the reconstructed images using 3-level Haar, Daub 4 and Coif 6 wavelet series at a fixed thresholding value which is $1/2^3$. Figure 5.10 (c) its energy is 4585718315.63846, 1.6% of the coefficients used, 0.07 bits per pixel and the compression ratio is 0.9895257695. While figure 5.10 (c') its energy is 4622181129.64237, 1.6% of the coefficients used, 0.06 bits per pixel and the compression ratio is 0.091100036. Also, figure 5.10 (c") its energy is 4620789559.59031 , 1.6% of the coefficients used, 0.06 bits per pixel and the compression ratio is 0.997093591 .

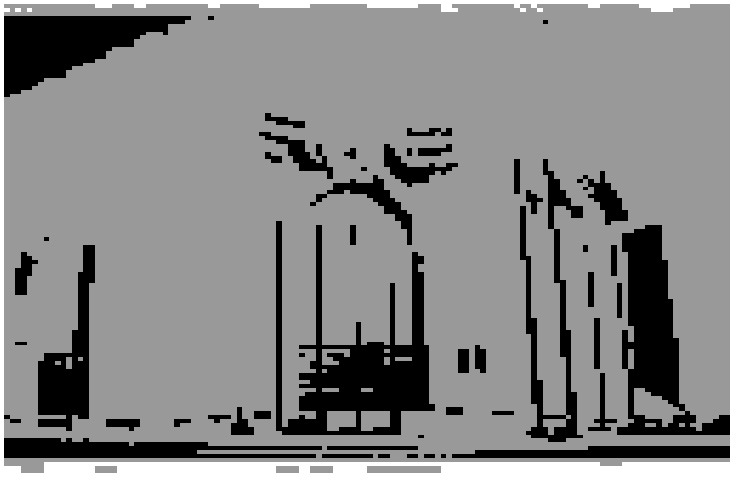
Notice that the reconstructed images of 3-level Daub 4 and Coif 6 wavelet series transform is clearer than the one prepared by Haar wavelet series at the same level and same threshold even both have the same bits per pixel and percentage of coefficients being used.

**Example 5.14:**

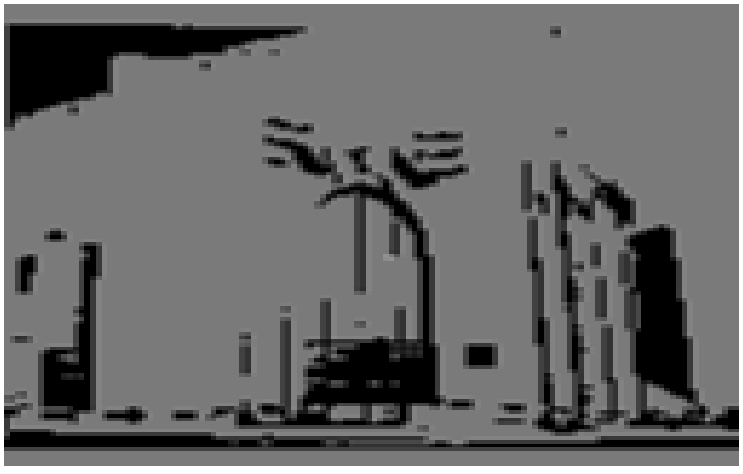
For the same image from example 3.9 we will compress it using 2-level Haar and Daub 4 and Coif 6 wavelet series at a fixed thresholding value



(a)

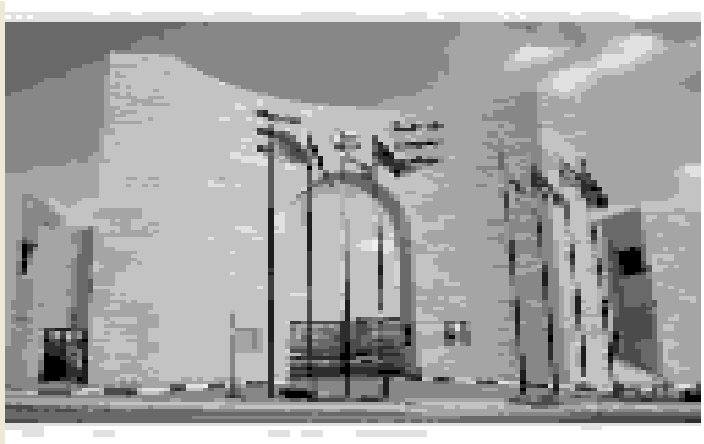


(b)

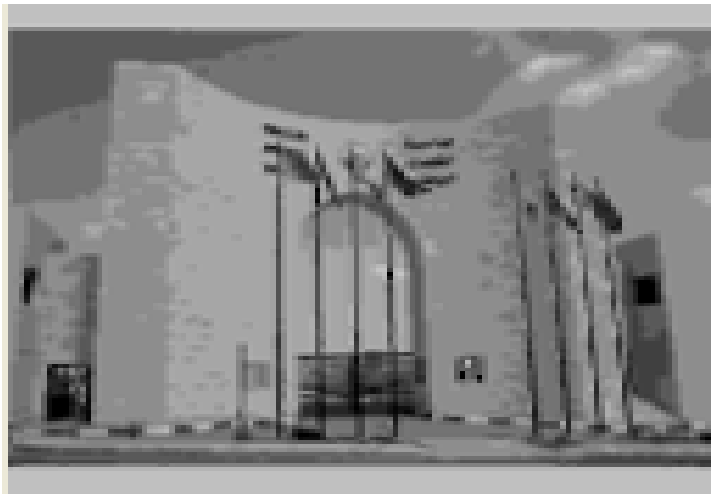


(b')

(b")

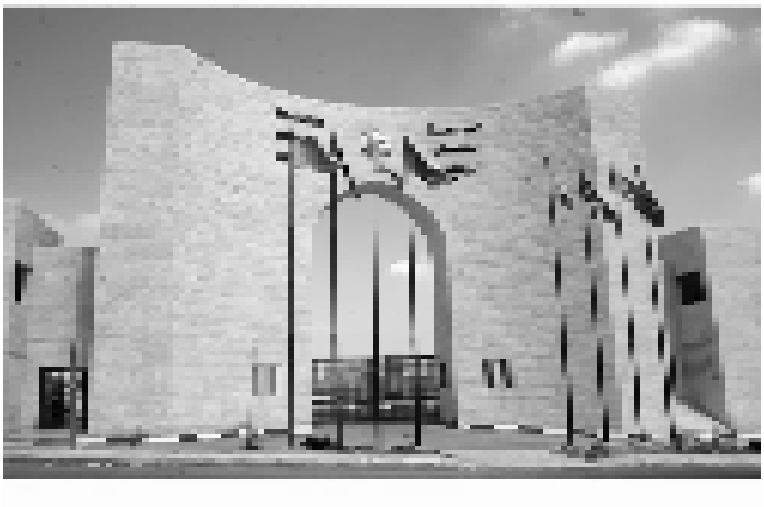


(c)

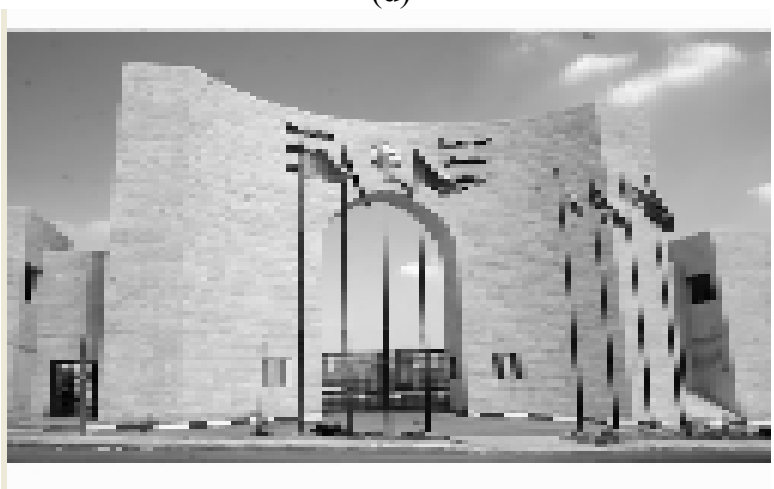


(c')

(c")



(d)



(d')

(d")

**Figure** 5.11

Figure 5.11 (a) represents the original image where 263169 sample used and its energy is 8596962998.

Figure 5.11 (b,b',b") represent the reconstructed images using 2-level Haar, Daub 4 and Coif 6 wavelet series at a fixed thresholding value which is $1/2$ respectively, figure 5.11 (b) its energy is 9394226775, 5.3% of the coefficients used, 0.11 bits per pixel and the compression ratio is 1.092737840. While figure 5.11 (b') its energy is 8665213903.66226, 5.1% of the coefficients used, 0.1 bits per pixel and the compression ratio is 1.007938955. Also, figure 5.11 (b") its energy is 8379409629.48208, 5.1% of the coefficients used, 0.1 bits per pixel and the compression ratio is 0.9746941601.

Figure 5.11 (c,c',c") represent the reconstructed images using 2-level Haar, Daub 4 and Coif 6 wavelet series at a fixed thresholding value which is $1/2^3$, respectively, figure 5.11 (c) its energy is 8673786643.17264, 6.5%

of the coefficients used, 0.26 bit per pixel and the compression ratio is 1.008936138. While figure 5.11 (c') its energy is 8538627768.59786, 6.5% of the coefficients used, 0.26 bit per pixel and the compression ratio is 0.9932144376. Also, figure 5.11 (c'') its energy is 8435345747.42825, 6.5% of the coefficients used, 0.26 bit per pixel and the compression ratio is 0.9812006576.

Figure 5.11 (d,d',d'') represent the reconstructed images using 2-level Haar, Daub 4 and Coif 6 wavelet series at a fixed thresholding value which is $1/2^8$. Figure 5.11 (d) its energy is 8601794659.94574, 32.3% of the coefficients used, 2.9 bit per pixel and the compression ratio is 1.000562020. While figure 5.11 (d') its energy is 8598852073.02674, 28.9% of the coefficients used, 2.6 bit per pixel and the compression ratio is 1.000219737. Also, figure 5.11 (d'') it energy is 8596796018.56208, 28.7% of the coefficients used, 2.59 bit per pixel and the compression ratio is 0.9999805770.

Notice that the reconstructed images of 2-level Daub 4 and Coif 6 wavelet series transform is clearer than the one prepared by Haar wavelet series at the same level and same threshold even both have the same bits per pixel and percentage of coefficients being used.

## 5.8 Compression by Fourier Transform and the Wavelet Transform

Data compression is very active with new approaches, ideas, and techniques being developed and implemented all the time. JPEG is a

Fourier-based compression format widely used for color image compression but is not perfect. This is why in 1995 the JPEG committee has decided to develop a new wavelet-based standard for t compression of still images, to be known as JPEG2000. The final draft was approved by the JPEG committee in 2000 and this new standard is expected to improve the following areas:

- High compression efficiency.
- The ability to handle large images, up to $2^{32} \times 2^{32}$ while the original JPEG can handle images of up to $2^{16} \times 2^{16}$.
- Progressive image transmission.
- Easy, fast access to various points in the compressed stream.
- The decoder can pan, zoom, rotate and crop the image while decompressing only parts of it.
- Error resilience. Error-correcting codes can be included in the compressed stream, to improve transmission reliability in noisy environments.

The new important approaches introduced by JPEG2000 are:

1. The decoder can decompress the entire image or parts of it in lower quality and lower resolution.

2. The decoder can extract parts of the compressed stream and assemble them to create a new compressed stream without having to do any decompression.

The advantages of JPEG2000 are:

1. Saves time and space.

2. Prevents the building of image noise.

In the next two examples, we use the softwarte waveanalyzer [24] for comparison between between Fourier transform and the wavelet transform using the JPEG and JPEG2000 compression formats.

**Example 5.15:**

Figure 5.12 shows a 24-bit colored image of Archimedes. Its dimension is $1920 \times 2560$ and the uncompressed size is 14400 Kb. Now we will compress this image using JPEG (Fourier-based compression) and JPEG2000 (wavelet-based compression) methods at different compression ratios.

Figures 5.13, 5.15, and 5.17 show reconstructed images from compressed ones with different compression ratios using Fourier transform.

Figure 5.13 is obtained from the compressed image of size 83 Kb with compression ratio 0.6% and 0.1386 bits per pixel. Figure 5.15 obtained from the compressed image of size 319 Kb with compression ratio 2.2% and 0.5320 bits per pixel. Also, figure 5.17 is obtained from a compressed

image of size 3080 Kb with compression ratio 21.4% and 5.1340 bits per pixel.

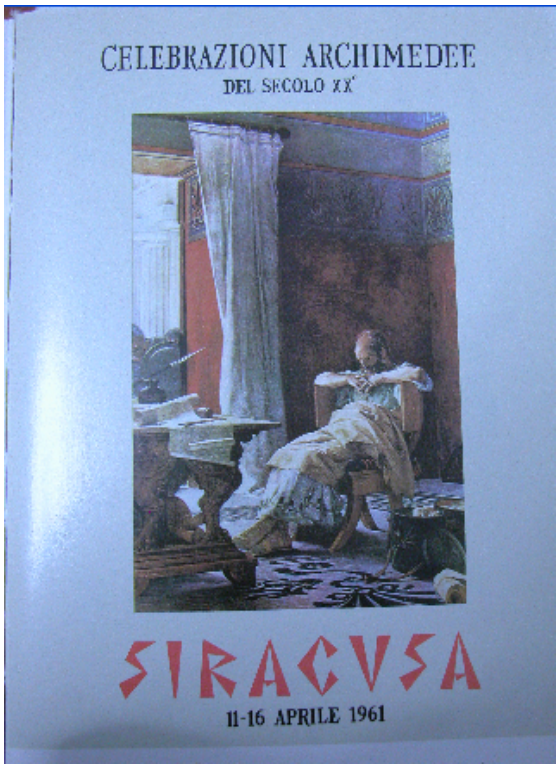



**Figure** 5.12

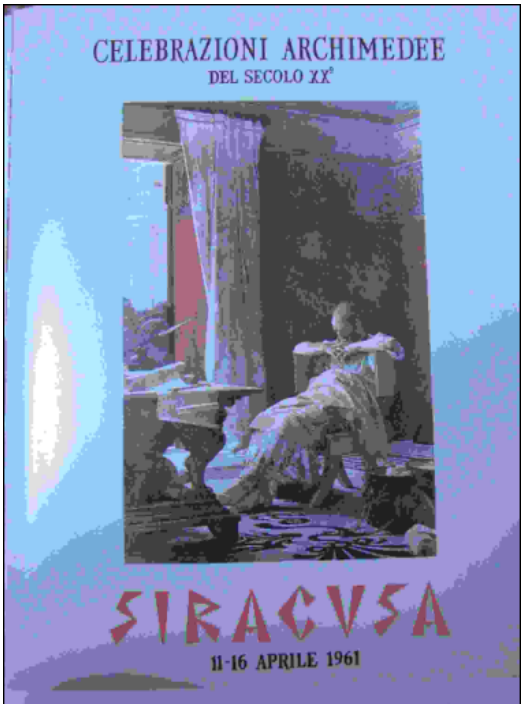



**Figure** 5.13

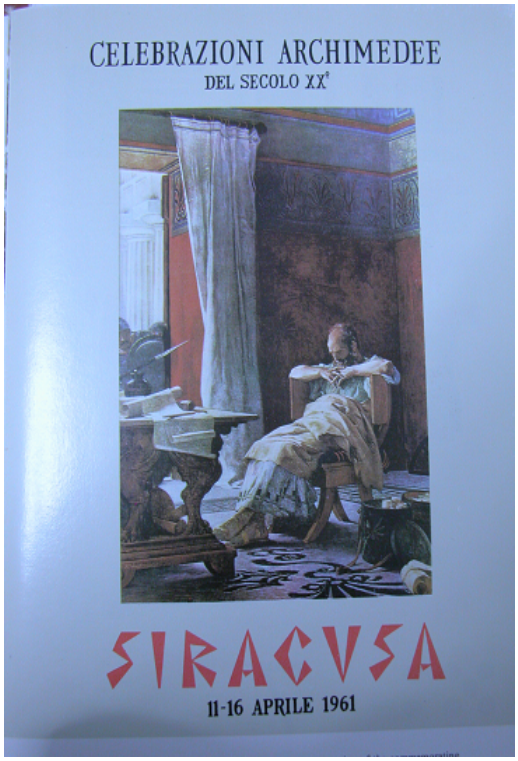



**Figure** 5.14

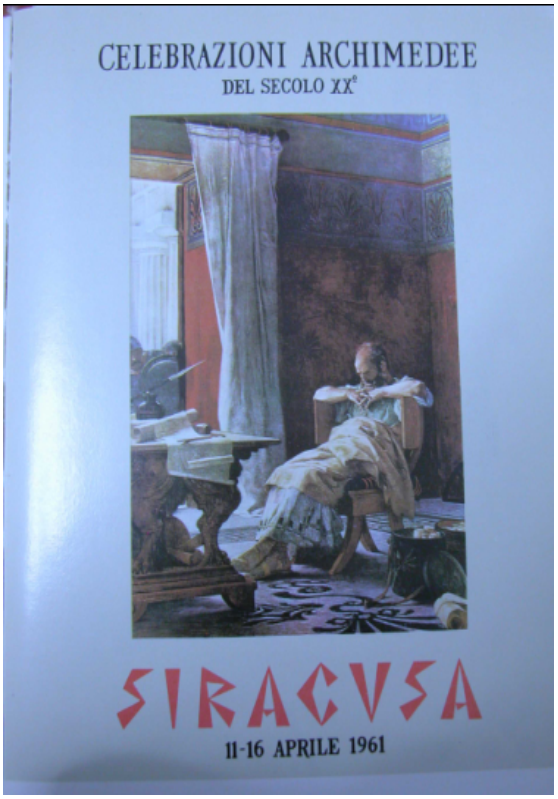



**Figure** 5.15

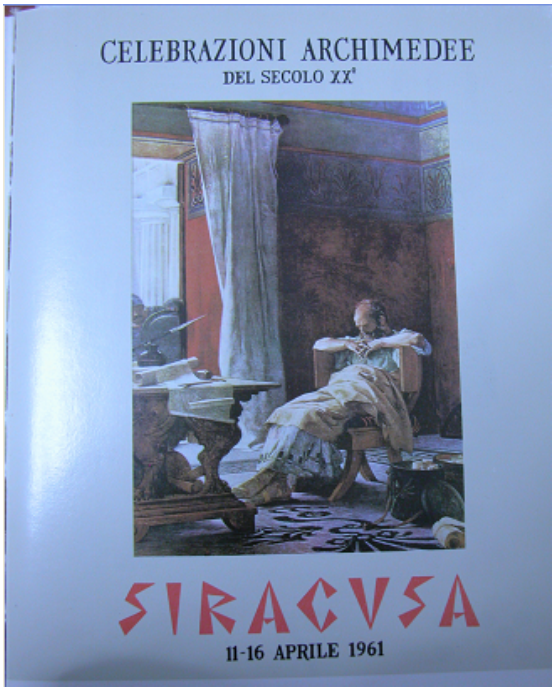



**Figure** 5.16

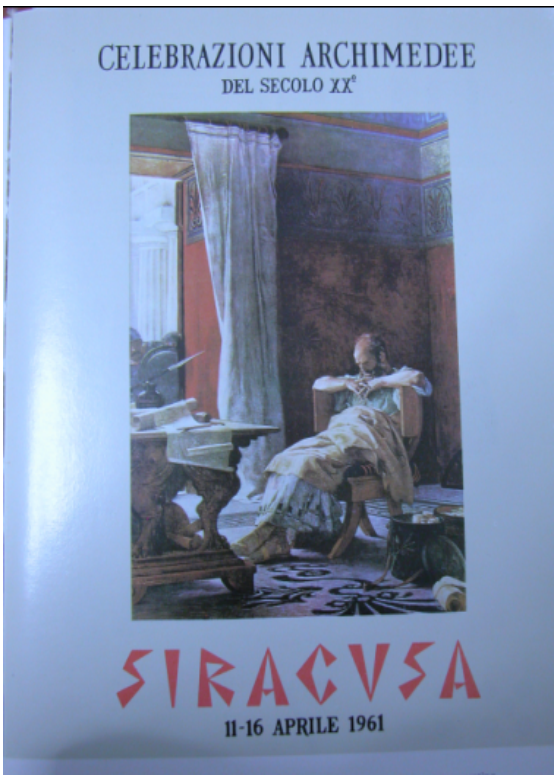



**Figure** 5.17

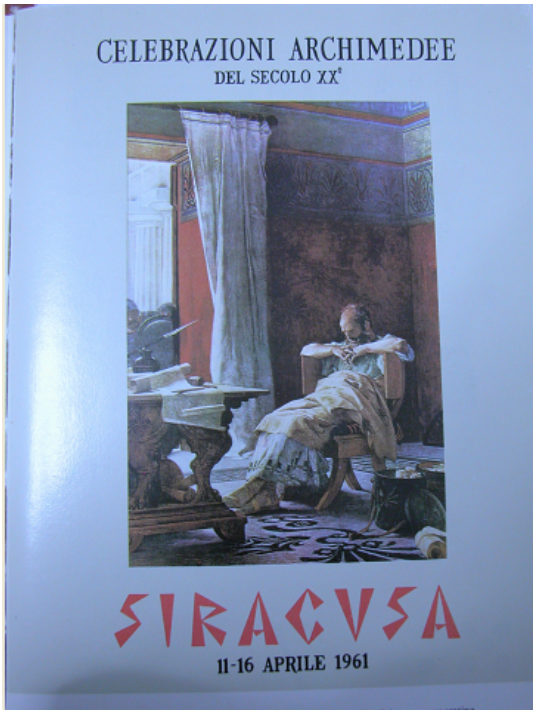



**Figure** 5.18

Figures 5.14, 5.16, and 5.18 show reconstructed images from compressed ones with different compression ratios using wavelet transform. Figure 5.14 is obtained from the compressed image of size 72 Kb with compression ratio 0.5% and 0.1198 bits per pixel. Figure 5.16 is obtained from the compressed image of size is 14 Kb with compression ratio 0.1% and 0.0239 bits per pixel. Also, figure 5.18 is obtained from the cmpressed image of size is 3 Kb with compression ratio is 0.02% and 0.0048 bits per pixel.

This example shows that compressing the image using JPEG 2000 (wavelet based compression) is more efficient than compressing it using JPEG (Fourier based compression) since we could compress the image from 14400 Kb to 3 Kb using JPEG 2000 with reasonable resolution, while the highest compression ratio obtained by JPEG for the same image is 0.6% and the compressed file size or 83 Kb.

**Table** 5.8

| **JPEG (using Fourier transform)** | | | **JPEG2000 (using wavelet transform)** | | |
|---|---|---|---|---|---|
| figure | Compression Ratio | Size | figure | Compression Ratio | Size |
| 5.13 | 0.6% | 83KB | 5.14 | 0.5% | 72KB |
| 5.15 | 2.2% | 319KB | 5.16 | 0.1% | 14KB |
| 5.17 | 21.4% | 3080KB | 5.18 | 0.02% | 3KB |

**Example 5.16:**

Figure 5.19 represents an image for An-Najah University, its dimension is $640 \times 479$, its 24-bit colored image and its size is 898Kb. Now we will compress this image using JPEG and JPEG2000 methods at different compression ratios.

**Figure** 5.19

**Figure** 5.20

**Figure** 5.21

**Figure** 5.22

**Figure** 5.23

Figures 5.20 and 5.22 show reconstructed images from compressed ones with different compression ratios using Fourier transform. Figure 5.20 is obtained from the compressed image of size 7 Kb with compression ratio 0.8% and 0.1979 bits per pixel. Figure 5.22 is obtained from the compressed image of size is 167 Kb with compression ratio 8.69% and 4.4759 bits per pixel.

Figures 5.21 and 5.23 show reconstructed images from compressed ones with different compression ratios using wavelet transform. Figure 5.21 is obtained from the compressed image of size 4 Kb with compression ratio 0.5% and 0.1198 bits per pixel. Figure 5.23 is obtained from the compressed image of size is 2 Kb with compression ratio 0.2% and 0.0471 bits per pixel.

**Table** 5.9

| **JPEG (using Fourier transform)** | | | **JPEG2000 (using wavelet transform)** | | |
|---|---|---|---|---|---|
| figure | Compression Ratio | Size | figure | Compression Ratio | Size |
| 5.20 | 0.8% | 7KB | 5.21 | 0.5% | 4KB |
| 5.22 | 8.69% | 167KB | 5.23 | 0.2 | 2KB |

This example shows that wavelet transform is more efficient than Fourier transform in compressing colored images.

## 5.9 Conclusion

There are two categories of data compression; lossless and lossy. In lossless compression all details are reserved which means the reconstructed data is identical to the uncompressed ones. This type of compression does not achieve high compression ratios and is not considered in this thesis. The other type is the lossy compression where some details is lost in the process of compression which means the reconstructed signal is approximately the same, but not identical, to the original one. The accuracy in the reconstructed signal depends on a threshold value determined by the user.

Smaller threshold values leads to more accuare reconstructed images and higher compression ratio.

Multiresolution analysis (MRA) divides the signal into a trend subsignal and a finite sequence of detailed subsignals. Reconstruction is done by adding a detail subsignal at a time to get the desired resolution.

In this thesis, a computational study of two major transforms used extensively in digital signal processing in general and in data compression in particular is done. The two transforms are the classical Fourier transform and the relatively new wavelet transform. A computational comparison between the two transforms has been made showing that the wavelet transform is more efficient than Fourier transform in data compression. As a result, several data compression software, using wavelet transforms, were introduced in the last two decades. Examples of these are the image compressor JPEG2000 and the text compressor DJVU.

There are several well-known wavelet transforms. In this thesis, we considered Haar, Daub*J* for *J*= 4, 6, and 20, and the Coiflet wavelet transforms. Haar wavelet transform is the simplest one and is used as a prototype for other wavelet transforms. A comparison between these transforms shows efficiency of each one compared to the others depends on the signal. For instance, the Haar transform is considered efficient for piecewise constant signals. For continuous signals, Daub*J* and Coiflet transforms are more efficient.

Results from section 5.7 shows that wavelet transform is more efficient than Fourier transform for image compression. Tables 5.8 and 5.9 clearly show the efficiency of the wavelet transform when compared with Fourier transform for colored images.

## References


1. Allen, R.L., Mills, D.W.: **Signal Analysis Time, Frequency, Scale and Structure**, WILEY-INTERSCIENCE.

2. Amara Graps, http://www.amara.com/ftpstuff/IEEEwavelet.pdf.

3. Boggess, A., Narcowich, F.J.: **First Course in Wavelets with Fourier Analysis**, Prentice Hall, 2001.

4. Brigges, W.L., Henson, and V.E.: **The DFT, An Owner's Manual for the Discrete Fourier Transform**, Siam,1995.

5. Brislawn, Christopher M., http://www.c3.lanl.gov/~brislawn/FBI/FBI.html.

6. Burden, R.L., Faires, J.D.: **Numerical Analysis,** Brooks Cole, 2004.

7. Christensen, O., Christensen, K.L.: **Approximation Theory from Taylor Polynomial to Wavelets**, Birkhäuser, Boston, Basel, 2004.

8. Chui, C.K.: **An Introduction to Wavelets, Volume1 (Wavelet Analysis and Its Applications)** ,Academic Press,1992.

9. Daubechies, I.: **Ten Lectures on Wavelets (CBMS-NSF Regional Conference Series in Applied Mathematics)**, 1992.

10. Debnath, L.: **Wavelet Transform & Their Application**, Birkhäuser, Boston, Basel, 2002.

11. Frazier, M.W.: **An Introduction to wavelets Through Linear Algebra (Undergraduate Texts in Mathematics)**, Springer, 2001.

12. Grattan-Guinness I., J. R. Ravets J.R., **Joseph Fourier, 1768-1830, A survey of his Life and Work,** The MIT press, 1972.

13. Hernandez, E., Weiss, G.: **First Course on wavelets**, CRC, 1996.

14. Jaffard, S., Meyer, Y., Ryan, and R.D.: **Wavelets: Tools For Science & Technology**, Society for Industrial Mathematics, 2001.

15. Kaiser, G.: **A Friendly Guide to Wavelets**, Birkhäuser Boston, 1994.

16. Kreyszig, E.: **Advanced Engineering Mathematics, Student Solution Manual and Study Guide,** Wiley, 2006.

17. Misiti, M., Oppenheim, G. and Misiti, Y.: **Wavelets and Their Applications (Digital Signal and Image Processing Series)**, ISTE Publishing Company, 2007.

18. Mix, F., Olejniczak J.: **Wavelets**, Wiley- Inter Science, 2003.

19. Pensky, M. A.: **Introduction to Fourier Analysis Wavelets (Brooks / Cole Series in Advanced Mathematics)**, Brooks Cole, 2001.

20. Salamon, D.: **Data Compression,** 3rd. Springer. 2004.

21. Sidney, C. B., Ramesh, R. and Gopinath, A.: **Introduction to Wavelets and Wavelets Transforms**, Prentice Hall, 1997.

22. Stark, H.G.: **Wavelet and Signal Processing**, Springer-Verlag, Berlin, Heidelberg, Netherland, 2005.

23. Tang, Y.Y., Wickerhauser, V., Pong, C.Y. and Li C.H.: **Wavelet Analysis and Its Applications : Second International Conference, WAA 2001 Hong Kong, China, December 18-20-2001.Procedings (Lecture Notes in Computer Science)**, World Scientific Pub Co Inc,2003.

24. Vinther, Michael, **Image Analyzer (software), version 1.28**, http://meesoft.logicnet.dk/

25. Walker, J.S.: **A Primer on Wavelets and their Scientific Application**, Chapman & Hall/CRC,Boca Rota, London, New York, Washington D.C.,1999.

26. Walker, J.S.: **FAWAVE, A Fourier\Wavelet Analyzer**, **(software) version 2,** http://www.uwec.edu/walkerjs

27. Wallnut, D.F.: **An Introduction to Wavelets Analysis**, Birkhäuser, Boston, Basel, 2002.

**جامعة النجاح الوطنية**

**كلية الدراسات العليا**

# ضغط البيانات بواسطة المويجات

**إعداد**

**رنا بسام داود اسميرات**

**إشراف**

**د. أنور صالح**

**قدمت هذه الأطروحة استكمالا لمتطلبات درجة الماجستير في الرياضيات المحوسبة بكلية الدراسات العليا في جامعة النجاح الوطنية في نابلس، فلسطين**

**2009**

**ضغط البيانات بواسطة المويجات**

**إعداد**

**رنا بسام داود اسميرات**

**إشراف**

**د. أنور صالح**

## الملخص

هناك نوعان من ضغط البيانات: الأول بدون خسارة (مطابق) والثاني بخسارة (تقريبي). في النوع الأول، التفاصيل محفوظة لكن دون الحصول على نسبة ضغط كافية وهذا النوع من ضغط البيانات لم يناقش في هذه الأطروحة. اما في النوع الثاني، فتكون هناك خسارة لبعض التفاصيل وحجم هذه التفاصيل يتناسب طرديا مع نسبة الضغط المرجوة ويتحكم المستخدم بذلك. باستخدام هذا النوع من الضغط، يمكن الحصول على نسبة ضغط عالية جدا مع دقة كافية في البيانات المسترجعة من البيانات المضغوطة.

في هذا البحث، تمت دراسة تحويلات فوريه التقليدية وتحويلات المويجات الحديثة نسبيا، وهما من أكثر التحويلات استخداما في ضغط البيانات. إضافة لذلك، قمنا بعمل مقارنة محوسبة بين التحويلين الأساسيين وتبين أن تحويلات المويجات كانت أكثر فعالية من تحويلات فوريه التقليدية. إن نسب الضغط العالية التي يمكن الحصول عليها بواسطة تحويلات المويجات كانت حافزا لظهور العديد من برمجيات ضغط البيانات بواسطة المويجات في العقدين الأخيرين. وكأمثلة على ذلك: برمجيات JPEG 2000 لضغط الصور وبرمجيات DJVU لضغط النصوص.