

Efficient Signal Transmission and Wavelet-based Compression

Susan E. Kelly*

Department of Mathematics
University of Wisconsin–La Crosse

James S. Walker†

Department of Mathematics
University of Wisconsin–Eau Claire

Abstract

In this paper we describe how digital signals can be transmitted by analog carrier signals. This is the method of *pulse code modulation*. Transmission by pulse code modulation is widely used thanks to its abilities to withstand corruption by noise and to transmit multiple signals simultaneously. Digital signals can also be used to compress analog signals, thus allowing for more rapid transmission, by employing *wavelet series*. We shall describe how wavelet series are employed and compare them with Fourier series, showing that in many instances wavelet series provide a better technique.

Introduction

We now live in a digital world. Information is transmitted digitally in ever increasing amounts. One common method of transmitting digital signals is called *pulse code modulation* (PCM). A PCM signal is an analog signal that encodes the bits of the digital signal via a *carrier wave*. The frequency of this carrier wave is modified in such a way that the bits can be read off at the receiving end. This PCM method has a couple of beautiful features. First, it can withstand large amounts of noise in the environment which typically corrupt the signal by the time it reaches the receiver. Second, it allows for *simultaneous* transmission of several different messages using just one PCM signal.

An important benefit of digital signals is their ability to compress analog signals. Instead of transmitting the entire analog signal, it is more efficient to transmit numbers—digitally encoded—which describe how to reconstruct the signal at the receiving end. A classical method for producing these numbers is to compute coefficients in a Fourier series. Recently, a whole new approach, called *wavelet series*, has been developed. Coefficients of a wavelet series are computed, digitally encoded, then transmitted in digital form. At the receiving end, these coefficients are used to produce a wavelet series that provides an excellent approximation to the original signal. Wavelet series are often more efficient—less coefficients are needed for a good approximation—than Fourier series.

*kelly.susa@uwlax.edu

†walkerjs@uwec.edu

In the next section we shall describe the theory behind PCM signals and illustrate their application to digital signal transmission. We then discuss Fourier series and wavelet series and their application to compression of analog signals. The paper concludes with a brief survey of references where further details and extensions of these ideas can be found.

The numerical data and graphs in this paper were produced with the free software FAWAV, which can be downloaded from the website [1].

1 PCM and Digital Signal Transmission

In this section we shall explain how an analog carrier signal can be modified to encode a sequence of bits. Suppose, for example, that we want to transmit the following sequence of seven bits:

$$1000110. \quad (1)$$

In order to represent any of the bits having value 1, we use a signal of the form:

$$g(t) = e^{-\gamma(t-k)^2} \cos 2\pi\nu t. \quad (2)$$

In Fig. 1(a) we show a graph of this signal, using $k = 6$, $\gamma = 2$, and $\nu = 6$. While in Fig. 1(b) we show a different graph, using $k = 3$, $\gamma = 25$, and $\nu = 12$. Examining these figures, it is not hard

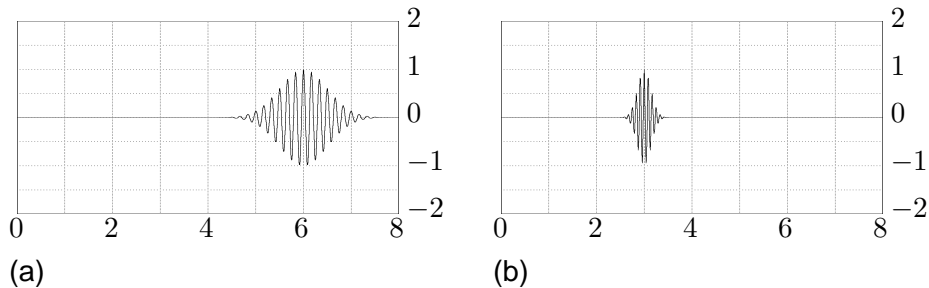


Figure 1: (a) Pulse centered at position $t = 6$. (b) Pulse centered at position $t = 3$.

to see that the parameter k controls where the center of the amplitude factor $e^{-\gamma(t-k)^2}$ lies, and that the parameter γ controls how quickly this factor damps down toward 0 as t moves away from k . In (b), the factor γ equals 25, which is larger than the value of $\gamma = 2$ for (a). Hence the amplitude of the pulse in (b) damps down more quickly to 0.

The parameter ν in Eq. (2) determines the *frequency* of the factor $\cos 2\pi\nu t$. Hence the cosine factors for the pulses shown in Fig. 1 have frequencies of $\nu = 6$ and $\nu = 12$. As you can see from the figure, the oscillations of the pulse in (b) are faster (twice faster, in fact) than the oscillations of the pulse in (a).

It is a standard task in electrical engineering to design a circuit having maximum response centered on a desired frequency ([2], [8]). See Fig. 2. For example, if we want to send the bit 1 using the PCM signal shown in Fig. 1(b), then we should design a receiver having an electric circuit element with maximum response at $\nu = 12$. The response curve shown in Fig. 2(b) would provide an excellent detector for such a transmission of the bit 1. Likewise, if we wanted to use the PCM signal in Fig. 1(a), then the maximum response for the receiver should be centered at $\nu = 6$.

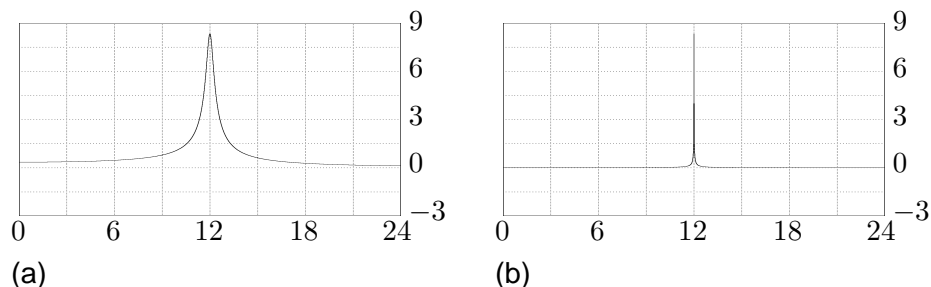


Figure 2: Response curves for electric circuits with maximum response at $\nu = 12$. (a) A broad response curve, unsuitable for precise detection. (b) A narrow, precisely tuned response curve, suitable for accurate detection.

By combining both such circuit elements, we could even have a receiver that could detect both bits, transmitted simultaneously. We shall return to this point later.

Now that we have discussed how to transmit a single bit having value 1, we can turn to the question of how to transmit a sequence of bits having different values, as in (1).

1.1 Transmitting a bit sequence

The method of transmitting a bit sequence, such as the one shown in (1), is to simply add together signals of the type shown in Eq. (2). One frequency, ν_1 , is used for transmitting bits of value 1, and a second frequency $\nu_0 \neq \nu_1$ is used for transmitting bits of value 0.

For example, in order to transmit the sequence of bits 1 0 0 0 1 1 0, we use the following PCM signal:

$$f(t) = \{e^{-\gamma(t-2)^2} + e^{-\gamma(t-3)^2} + e^{-\gamma(t-4)^2} + e^{-\gamma(t-7)^2}\} \cos 2\pi\nu_0 t + \{e^{-\gamma(t-1)^2} + e^{-\gamma(t-5)^2} + e^{-\gamma(t-6)^2}\} \cos 2\pi\nu_1 t \quad (3)$$

with $\gamma = 150$, $\nu_0 = 232$, and $\nu_1 = 280$. See Fig. 3(a). In Figures 3 and 4 we illustrate the scheme for decoding the seven bits from the PCM signal at the receiving end.

In order to isolate an individual pulse at position $t = k$, the PCM signal $f(t)$ is multiplied by a *filter* (or *window*) function $w_k(t)$. This function $w_k(t)$ is equal to 1 in an interval $(k - \epsilon, k + \epsilon)$ centered at k , and decreases smoothly down to 0 for $t < k - 1 + \delta$ and $t > k + 1 - \delta$. See Fig. 3(b), where the filters w_1 and w_4 are shown. Multiplying the PCM signal f by the filter w_k , produces a single pulse $f(t)w_k(t)$ which is non-zero only within the interval $[k - 1 + \delta, k + 1 - \delta]$ centered at $t = k$. In Figures 3(c) and (d), we show the pulses produced by multiplying the PCM signal f by w_1 and w_4 . This produces single pulses centered at $t = 1$ and $t = 4$.

In Fig. 4(b), we show the whole set of filters $\{w_k\}_{k=0}^8$ for detection of the pulses at positions $t = 0, 1, 2, \dots, 8$. These filters satisfy

$$\sum_{k=0}^8 w_k(t) = 1 \quad (4)$$

for $0 \leq t \leq 8$. Equation (4) implies that the PCM signal $f(t)$ is equal to the sum of the filter functions $f(t)w_k(t)$ over the interval $[0, 8]$. For decoding longer PCM signals, Eq. (4) is extended

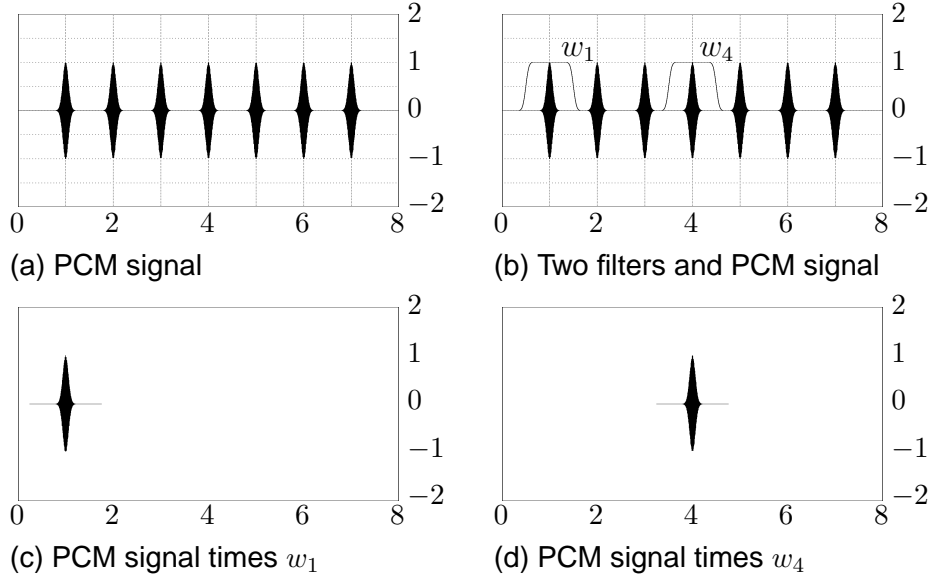


Figure 3: Filtering of PCM signal to process individual pulses. The filters w_1 and w_4 in (b) are used to cut out the individual pulses at positions $t = 1$ and $t = 4$. The cutout pulses are shown in (c) and (d).

via a larger set of filters, say $\{w_k\}_{k=0}^N$, satisfying

$$\sum_{k=0}^N w_k(t) = 1 \quad (5)$$

for some large positive integer N .

While (5) is a handy property, it is not absolutely necessary for the decoding scheme to work. In fact, for decoding to work well it is sufficient that the following double inequality holds:

$$A \leq \sum_{k=0}^N w_k(t) \leq B \quad (6)$$

where A and B are positive constants. Often a value of A that is significantly larger than 1 is used, producing an amplification of the PCM signal. This is often needed because PCM signals are significantly reduced in amplitude after transmission over long distances.

Notice that in Figures 3(c) and (d) the individual pulses are shown as having restricted domains. Domains of $[\delta, 2 - \delta]$ and $[3 + \delta, 5 - \delta]$ for w_1 and w_4 , respectively. The domain of each pulse $f(t)w_k(t)$ is restricted in order to facilitate processing by the frequency detector. Since $f(t)w_k(t) = 0$ outside of the interval $[k - 1 + \delta, k + 1 - \delta]$, there is no need for the detector to process this pulse outside of this interval.

In Fig. 4(c), we show the result of processing each isolated pulse by a frequency detector. The PCM signal is shown at the bottom of the figure. The top of the figure shows the magnitude of the response of a frequency detector—a frequency detector which sweeps through a set of *tuning frequencies* $\{2, 4, 6, \dots, 1024\}$ just like a dial on a radio. This set includes the frequencies $\nu_0 = 232$ and $\nu_1 = 280$ used for transmitting bits of value 0 and 1, respectively. (The reason for displaying

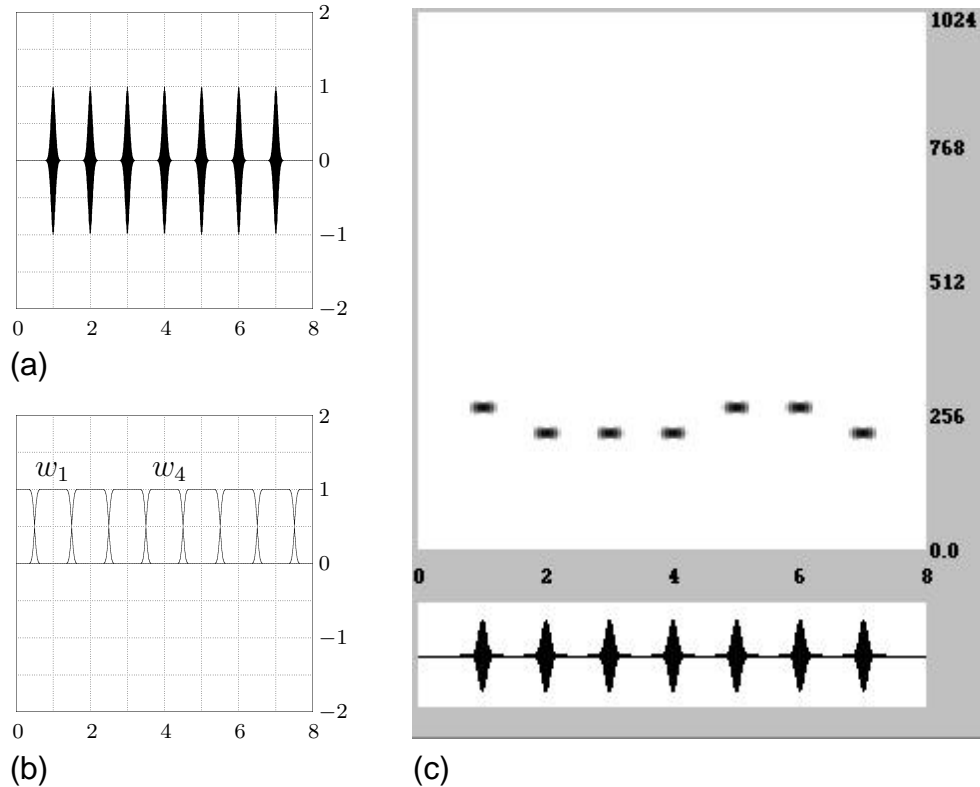


Figure 4: PCM transmission of bit sequence 1000110. (a) PCM signal. (b) Filter bank used for detection. (c) Detection: [bottom] PCM signal, [top] Magnitude of responses to filter bank output. Darker pixels represent more intense response.

responses for a large range of frequencies will be clearer after we discuss simultaneous transmission.) As the reader can see, the bit sequence 1000110 is easily read off from the display of the detector response.

1.2 Simultaneous messages and denoising

The beauty of PCM signal transmission consists in its wealth of capabilities. In particular, it provides a simple way of transmitting several messages simultaneously using just one PCM signal. Furthermore, it allows for accurate recovering of signals even after the addition of large amounts of noise. Sometimes the noise is so severe that it will not be apparent from the graph of the signal that any transmission has occurred at all. Nothing but noise will be visible. Yet, upon decoding, the transmitted bits will stand out clearly from the noise. We now provide a couple of examples illustrating these ideas.

First, we show how the following three bit sequences: 1001100, 1100111, and 0101001 can be transmitted simultaneously. The method consists of adding together three PCM signals using three different values of ν_0 and ν_1 . For the three sequences above, we use the following values of ν_0 and ν_1 : $\nu_0 = 768 - 32, \nu_1 = 768 + 32, \nu_0 = 512 - 32, \nu_1 = 512 + 32$, and $\nu_0 = 256 - 32, \nu_1 = 256 + 32$, respectively. The resulting signal is shown at the bottom of Fig. 5(a), and its decoding is shown at the top of that same figure. It is clear from this figure that

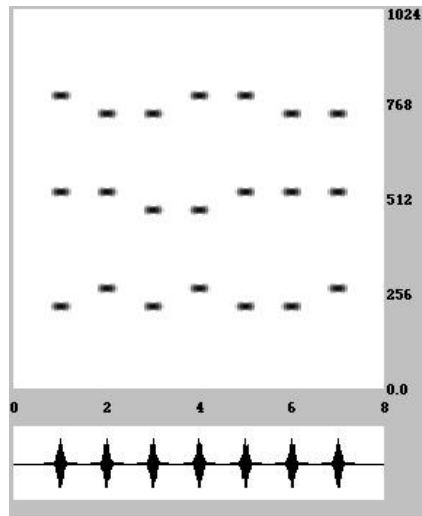


Figure 5: Decoding three simultaneous messages.

the three separate messages have been decoded from one PCM signal. With present technology, several thousand separate messages can be transmitted with just one PCM signal [5].

Second, we simulate the recovery of a bit sequence from a PCM signal that has been severely corrupted by additive noise. In Fig. 6(a) we show an example of a PCM signal for transmitting the bit sequence 1 1 0 0 1 0 1. The frequencies of $\nu_0 = 480$ and $\nu_1 = 544$ were used for transmitting the bits. For the type of noise to be added to this signal, we used a simulation of Gaussian random noise with a standard deviation of 1. Gaussian noise is frequently used for modeling real-world noise—because the Central Limit Theorem implies that sums of noise from many independent sources, over the course of transmission, will yield a total noise that is approximately Gaussian. A discrete realization of Gaussian noise with a standard deviation of 1 is shown in Fig. 6(b), and a histogram of its values is shown in Fig. 6(c). This histogram illustrates the Gaussian distribution underlying the random noise. The addition of the noise to the PCM signal is shown in Fig. 6(d). Notice that it is impossible to tell from this noisy signal that any message has been transmitted at all.

The decoding of the noisy PCM signal is shown in Fig. 7. In Fig. 7(a) we can see the responses to the bits in the PCM signal as prominent black smudges, which stand out from a light grey background resulting from the noise. These smudges are fairly easy for our eyes to decode as the bits 1 1 0 0 1 0 1 encoded by the original PCM signal. For electronic recognition, however, it is better if the frequency responses having magnitude below a threshold are set to zero. In Fig. 7(b) we show such a thresholded response. The responses to the bits are easily detectable electronically.

2 Compressing signals

In Section 1 we saw how we could transmit a signal if we could represent it with a sequence of bits. In this section we shall consider three different ways to obtain such a sequence, using power series, Fourier series, and wavelet series. We shall see why wavelet series often provide the best

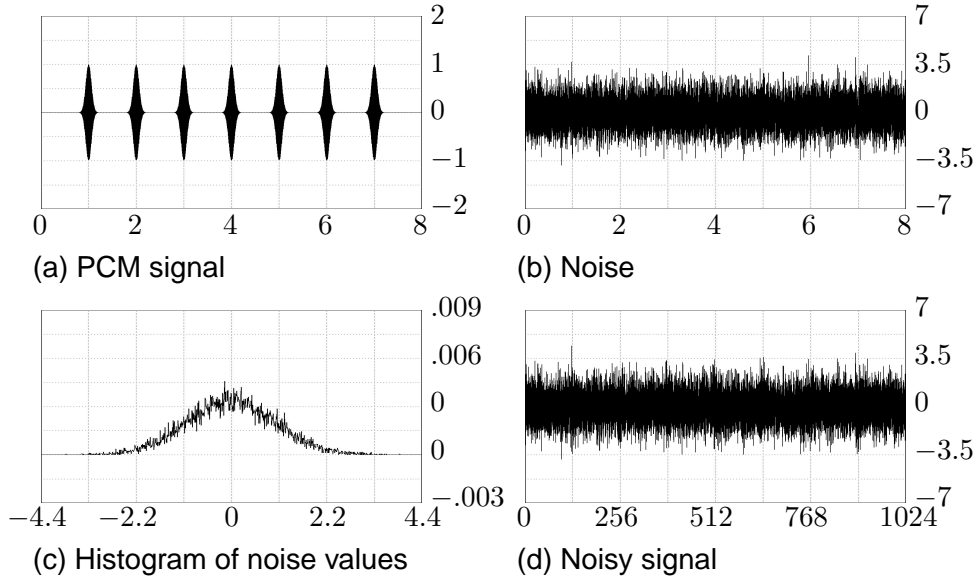


Figure 6: Simulation of additive noise corrupting a PCM signal.

method.

2.1 Power Series

In calculus we see how a function can be represented by a power series. The *Taylor series* of a function $f(t)$ at $t = a$ is given by

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (t - a)^n.$$

If $\lim_{M \rightarrow \infty} |f(t) - \sum_{n=0}^M \frac{f^{(n)}(a)}{n!} (t - a)^n| = 0$ for $|t - a| < R$, we say that f is equal to the sum of its Taylor series on the interval described by $|t - a| < R$. For example, the Taylor series for $f(t) = \sin t + 2 \cos 2t$ centered about the origin, is

$$\sin t + 2 \cos 2t = \sum_{n=0}^{\infty} (c_n t^{2n+1} + d_n t^{2n}) \quad (7)$$

where

$$c_n = \frac{(-1)^n}{(2n+1)!} \quad \text{and} \quad d_n = \frac{(-1)^n 2^{2n+1}}{(2n)!}$$

for all real t . Thus to transmit the function $f(t)$ one need only know the coefficients, c_n and d_n , of this power series. These coefficients can be written as 16-bit binary codes. These codes can then be strung together to give us the bit sequences examined in Section 1.

Of course in practice only a finite number of coefficients can be used, and when we truncate the series by using only the first N terms we introduce error in our representation. For each t -value, this error would be $|\sum_{n=N}^{\infty} \frac{f^{(n)}(a)}{n!} (t - a)^n|$. In Fig. 8 we have the graph of the function $f(t) = \sin t + 2 \cos 2t$. The Taylor series for this function at $t = 0$ converges for all values of t , but when we only use a finite number of terms in the series we obtain errors. The series works best near $t = 0$ where the series is being expanded about. As we add more terms, the reconstructed

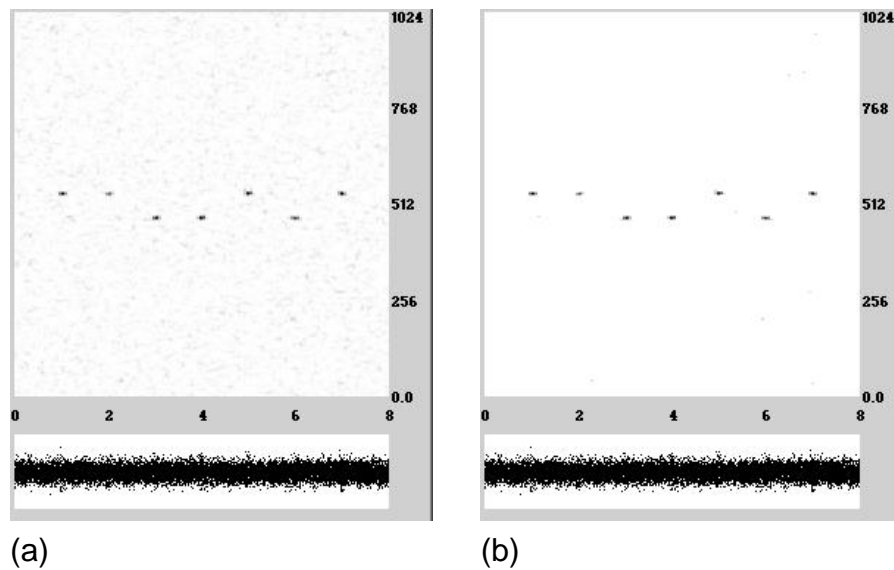


Figure 7: (a) Decoding noisy message. (b) Thresholded decoding.

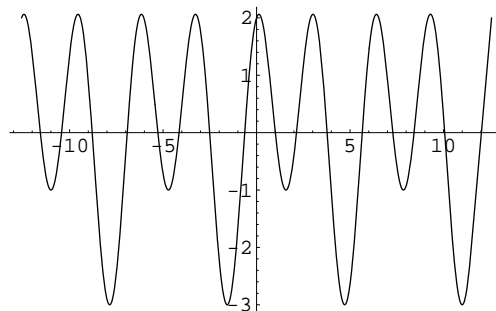


Figure 8: $f(t) = \sin t + 2 \cos 2t$

series resembles the function for a larger domain, but since we are adding terms of the form t^n , at some point the reconstructed function will increase or decrease without bound. These ideas are illustrated in Fig. 9.

For a Taylor series about $y = a$, we can represent a function in terms of powers of $t - a$. As illustrated with the above example, even when the power series converges everywhere, a finite sum may not represent the function well for values a distance away from $t = a$.

Another problem with Taylor series in many applications is that computing the derivatives needed for the coefficients is not always practical or possible. What we need in practical applications is a series with more easily computed coefficients and which requires fewer terms to give a good reconstruction of our original signal. In Section 2.2, we shall see how Fourier series improve our work in both of these respects. In Section 2.3, we shall see how to compress data still further using wavelets.

2.2 Fourier Series

Another way to represent many functions was introduced by Jean B.J. Fourier in the early 19th century. A Fourier series represents a periodic function in terms of sine and cosine functions. For

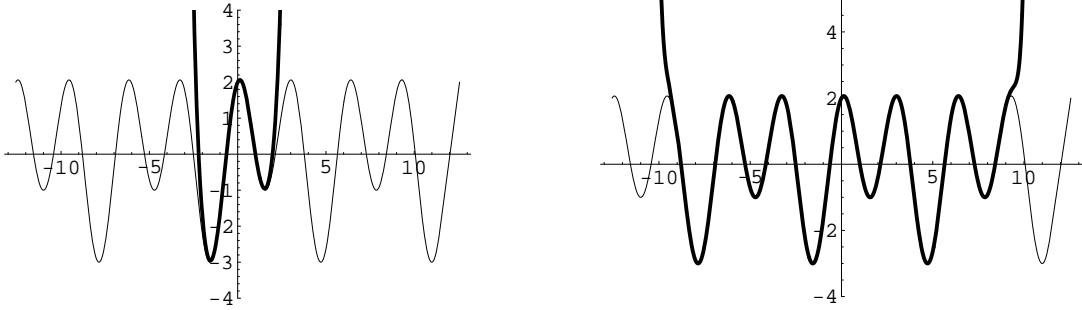


Figure 9: Taylor series at $t = 0$ for $\sin(t) + 2 \cos(2t)$. Left: 10 terms. Right: 50 terms. The original function is shown in lighter print.

a $2W$ -periodic function $f(t)$, its *Fourier series* is the series in the following correspondence:

$$f(t) \sim \frac{1}{2}a_0 + \sum_{n=1}^{\infty} [a_n \cos(\frac{n\pi t}{W}) + b_n \sin(\frac{n\pi t}{W})], \quad (8)$$

where the *Fourier coefficients* are defined to be

$$a_n = \frac{1}{W} \int_{-W}^W f(t) \cos(\frac{n\pi t}{W}) dt \quad \text{for } n = 0, 1, 2, \dots$$

$$b_n = \frac{1}{W} \int_{-W}^W f(t) \sin(\frac{n\pi t}{W}) dt \quad \text{for } n = 1, 2, \dots$$

Note that the 2π -periodic signal, $\sin t + 2 \cos 2t$, from the previous section can be captured exactly with only two nonzero Fourier coefficients!

A Fourier series can also be written in the complex form

$$f(t) \sim \sum_{n=-\infty}^{\infty} c_n e^{in\pi t/W}, \quad (9)$$

where the Fourier coefficients are

$$c_n = \frac{1}{2W} \int_{-W}^W f(t) e^{-in\pi t/W} dt.$$

One can view a Fourier series as a sum of terms having single frequencies, the amplitudes of those terms are the Fourier coefficients for the function. Fourier series are expansions of functions using the basis functions $\{1, \sin(\frac{nt}{W}), \cos(\frac{nt}{W})\}_{n=1}^{\infty}$ or $\{e^{i\frac{n\pi t}{W}}\}_{n \in \mathbf{Z}}$, which are orthogonal bases of periodic functions on $[-W, W]$.

Most signals are not periodic, however, and so we need to broaden the notion of Fourier series. For piecewise smooth functions in $L^1(\mathbf{R}) = \{f : \int_{-\infty}^{\infty} |f(t)| dt < \infty\}$, this can be done via the *Fourier transform*

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi\xi t} dt \quad (10)$$

and the *Fourier inversion formula*

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{i2\pi\xi t} d\xi.$$

The discrete form of Fourier series for analog signals is a consequence of the *Sampling Theorem*.

Theorem 2.1 (Sampling Theorem) *If a continuous function $f(t) \in L^1(\mathbf{R})$ is bandlimited, that is, $\hat{f}(\xi)$ is only non-zero on some interval $[-W, W]$, then $f(t)$ is completely determined by a sequence of points spaced $\frac{1}{2W}$ units apart. Specifically,*

$$f(t) = \sum_{n=-\infty}^{\infty} f\left(\frac{n}{2W}\right) \text{sinc}(2Wt - n),$$

where $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$.

Discussions of Theorem 2.1 can be found in [10], [11], and [17].

Theorem 2.1 allows us to use sample values of our function, $\{f(\frac{n}{2W})\}$, to create a bit sequence as examined in Section 1. In our work we are not only interested in creating a bit sequence, but we are also interested in compressing our data by creating as small a bit sequence as possible while still retaining the ability to generate a good approximation to the original function. To explore this issue further, we shall investigate the discrete Fourier transform.

In practice, our function $f(t)$ will exist only for a finite duration of time. Therefore, we can scale the function so that $f(t) = 0$ for $t \notin [0, 1]$. We can then apply Theorem 2.1 to $\hat{f}(\xi)$ in place of $f(t)$. Hence, $\hat{f}(\xi)$ can be recovered from the set $\{\hat{f}(k)\}_{k \in \mathbf{Z}}$.

Using the definition of the Fourier transform in Eq. (10) and the fact that $f(t)$ is zero off of $[0, 1]$, we obtain

$$\begin{aligned} \hat{f}(k) &= \int_0^1 f(t) e^{-i2\pi kt} dt \\ &= \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} \left[f\left(\frac{n}{N}\right) e^{-i2\pi k \frac{n}{N}} \right] \frac{1}{N}. \end{aligned}$$

This last sum comes from the definition of a Riemann integral. Thus, for N large,

$$\hat{f}(k) \approx F_k, \tag{11}$$

where

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f\left(\frac{n}{N}\right) e^{-i2\pi nk/N} \quad \text{for all } k \in \mathbf{Z} \tag{12}$$

is the N -point *discrete Fourier transform* (DFT) of the sequence $\{f_n = f(\frac{n}{N})\}_{n=0}^{N-1}$.

Formula (11) shows how the DFT values are determined by our function's sample values. Now let us investigate how to recover our sample values, and hence our function, from a finite number of DFT values. We will begin with the finite sum

$$\begin{aligned} \sum_{k=0}^{N-1} F_k e^{i2\pi nk/N} &= \sum_{k=0}^{N-1} \left[\frac{1}{N} \sum_{j=0}^{N-1} f\left(\frac{j}{N}\right) e^{-i2\pi jk/N} \right] e^{i2\pi nk/N} \\ &= \frac{1}{N} \sum_{j=0}^{N-1} \left[\sum_{k=0}^{N-1} (e^{-i2\pi/N})^{(j-n)k} \right] f\left(\frac{j}{N}\right), \end{aligned}$$

for $n = 0, 1, 2, \dots, N - 1$. Note that $(e^{-i2\pi/N})^{(j-n)}$ is a nontrivial root of $x^N - 1 = (x - 1)(x^{N-1} + x^{N-2} + \dots + x + 1)$ for $j - n \neq 0$, and thus $\sum_{k=0}^{N-1} [(e^{-i2\pi/N})^{(j-n)}]^k = 0$ for $j \neq n$. From this we obtain

$$\sum_{k=0}^{N-1} F_k e^{i2\pi nk/N} = f\left(\frac{n}{N}\right). \quad (13)$$

Thus to obtain all needed information about our function, we need to store either the N samples $\{f_n = f(\frac{n}{N})\}_{n=0}^{N-1}$ or the N DFT values $\{F_k\}_{k=0}^{N-1}$.

With the desire to compress our data we hope to use only a fraction of the DFT values—those having the largest magnitudes—and still get a good reconstructed function. Unfortunately, the magnitudes of the DFT values often damp down very slowly and hence we often need most, if not all, of these coefficients to obtain a good reconstruction of our original signal. Examples will be provided in the next subsection, when we compare this DFT-based method of compression with wavelet-based compression.

2.3 Wavelets

In an age when we seek faster and sharper scanners for medical diagnosis, more efficient computer storage techniques for larger amounts of computer data, and higher fidelity audio and visual signals with less information used, a more efficient way to convert analog signals to digital ones is needed. Many people working in Harmonic Analysis helped to provide the groundwork for the wavelet bases which were introduced in the 1980's. Wavelets are beneficial in the areas just mentioned and in many other applications because of the highly efficient ways a wavelet series can represent a function. As we shall illustrate below, they often allow for a better-reproduced analog signal using less data than with Fourier techniques.

Let us begin by defining a wavelet basis. We will work with wavelet bases for the set of functions $L^2(\mathbf{R}) = \{f : \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty\}$. A *wavelet basis* of $L^2(\mathbf{R})$ is a basis of functions generated using dilations by 2^{-m} and translations by $n2^{-m}$ of a single function. Here m and n are both integers. That is, if $\psi(t)$ is such a generating function, then

$$\{\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n)\}_{m,n \in \mathbf{Z}} \quad (14)$$

is a wavelet basis of $L^2(\mathbf{R})$. By a basis we mean that, for each $f(t) \in L^2(\mathbf{R})$, the following holds:

$$f(t) = \sum_{m \in \mathbf{Z}} \sum_{n \in \mathbf{Z}} \beta_n^m \psi_{m,n}(t) \quad (15)$$

where

$$\beta_n^m = \int_{-\infty}^{\infty} f(t) \psi_{m,n}(t) dt \quad (16)$$

are the wavelet coefficients. We shall refer to ψ as the *generating wavelet* for this wavelet basis.

A related way to represent our function $f(t)$ is with *scaling functions* generated by a single function ϕ . For any integer M , the set $\{\psi_{m,n}\}_{m < M, n \in \mathbf{Z}}$ is a basis of a subset $V_M \subset L^2(\mathbf{R})$. We can use $\phi(t)$ to generate another basis for V_M , the set $\{\phi_{M,n}(t) = 2^{M/2}\phi(2^M t - n)\}_{n \in \mathbf{Z}}$. Since $\lim_{M \rightarrow \infty} V_M = L^2(\mathbf{R})$, it follows that for M sufficiently large

$$f(t) \approx \sum_{n \in \mathbf{Z}} \alpha_n^M \phi_{M,n}(t),$$

where

$$\alpha_n^M = \int_{-\infty}^{\infty} f(t)\phi_{M,n}(t)dt. \quad (17)$$

We shall refer to ϕ as the *generating scaling function* for these bases $\{\phi_{M,n}(t)\}_{n \in \mathbf{Z}}$. We shall see below how both bases are used when we examine the fast wavelet transform.

There are an infinite number of wavelet bases for $L^2(\mathbf{R})$. The Haar wavelet system is undoubtedly the simplest one. For the Haar system, $\phi(t)$ is the characteristic function $\chi_{[0,1)}(t)$. The generator ψ for the Haar wavelets is defined by $\psi(t) = \chi_{[0,1/2)}(t) - \chi_{[1/2,1)}(t)$. The space V_M is the set of all $L^2(\mathbf{R})$ functions which are constant on each interval $[k/2^M, (k+1)/2^M)$, for all integers k .

The generators ϕ and ψ for the Haar system satisfy the following identities:

$$\begin{aligned} \phi(t) &= \phi(2t) + \phi(2t-1) \\ \psi(t) &= \phi(2t) - \phi(2t-1). \end{aligned}$$

Substituting $2^{m-1}t$ into these identities, and multiplying by $2^{\frac{m-1}{2}}$, yields

$$\begin{aligned} 2^{\frac{m-1}{2}}\phi(2^{m-1}t) &= \frac{1}{\sqrt{2}} \left[2^{\frac{m}{2}}\phi(2^m t) + 2^{\frac{m}{2}}\phi(2^m t - 1) \right] \\ 2^{\frac{m-1}{2}}\psi(2^{m-1}t) &= \frac{1}{\sqrt{2}} \left[2^{\frac{m}{2}}\phi(2^m t) - 2^{\frac{m}{2}}\phi(2^m t - 1) \right]. \end{aligned}$$

From these last two identities we obtain the following relations between Haar scaling coefficients and wavelet coefficients:

$$\begin{aligned} \alpha_k^{m-1} &= \frac{1}{\sqrt{2}}\alpha_{2k}^m + \frac{1}{\sqrt{2}}\alpha_{2k+1}^m \\ \beta_k^{m-1} &= \frac{1}{\sqrt{2}}\alpha_{2k}^m - \frac{1}{\sqrt{2}}\alpha_{2k+1}^m. \end{aligned} \quad (18)$$

These identities show that the $(m-1)$ st level coefficients $\{\alpha_k^{m-1}\}, \{\beta_k^{m-1}\}$ are obtained from the m th level scaling coefficients $\{\alpha_k^m\}$.

If we assume that N is a power of two, say $N = 2^R$, then the fact that ϕ is supported on $[0, 1)$ implies that when $M = R$ the support of each function $\phi_{M,n}(t)$ will only cover one sample point n/N . Hence in view of (17) we are justified in assuming that $\alpha_n^R = 2^{-R/2}f_n$. Thus, the scaling coefficients at the R th level can be set equal to a constant times f_n . In practice, the constant factor $2^{-R/2}$ is dropped and we assume that $\alpha_n^R = f_n$ for each n . The reason for dropping the constant factor $2^{-R/2}$ is that then (18), for $m = R$, defines an orthogonal matrix transformation on each pair of adjacent sample values f_{2k} and f_{2k+1} .

We have now shown that, given the scaling coefficients $\{\alpha_n^R = f_n\}$ at the R th level, we can use (18) repeatedly to derive all scaling coefficients $\{\alpha_n^m\}$ and wavelet coefficients $\{\beta_n^m\}$ for all levels $m < R$. Given the initial data $(\alpha_0^R, \alpha_1^R, \dots, \alpha_{N-1}^R) = (f_0, f_1, \dots, f_{N-1})$, we apply (18) to get the data $(\alpha_0^{R-1}, \dots, \alpha_{N/2-1}^{R-1}, \beta_0^{R-1}, \dots, \beta_{N/2-1}^{R-1})$. For example, if $(f_0, f_1, \dots, f_7) = (2, 2, 2, 2, 4, 4, 4, 12)$, then $R = 3$, and using (18) yields

$$(\alpha_0^2, \alpha_1^2, \alpha_2^2, \alpha_3^2, \beta_0^2, \beta_1^2, \beta_2^2, \beta_3^2) = (2\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 0, 0, 0, -4\sqrt{2}).$$

We may then repeat the application of (18) on the scaling coefficients $(\alpha_0^{R-1}, \dots, \alpha_{N/2-1}^{R-1})$ from this last data to obtain $(\alpha_0^{R-2}, \dots, \alpha_{N/4-1}^{R-2}, \beta_0^{R-2}, \dots, \beta_{N/4-1}^{R-2}, \beta_0^{R-1}, \dots, \beta_{N/2-1}^{R-1})$. In the example above, this second application of (18) yields

$$(\alpha_0^1, \alpha_1^1, \beta_0^1, \beta_1^1, \beta_2^1, \beta_3^1, \beta_2^2, \beta_3^2) = (4, 12, 0, -4, 0, 0, 0, -4\sqrt{2}).$$

Continuing in this way, we obtain at the R^{th} step, the data $(\alpha_0^0, \beta_0^0, \beta_0^1, \beta_1^1, \dots, \beta_{N/2-1}^{R-1})$. For the example, we can perform one more application of (18), thereby obtaining

$$(\alpha_0^0, \beta_0^0, \beta_0^1, \beta_1^1, \beta_2^1, \beta_3^1, \beta_2^2, \beta_3^2) = (8\sqrt{2}, -4\sqrt{2}, 0, -4, 0, 0, 0, -4\sqrt{2}).$$

Since (18) involves only six arithmetic operations on two adjacent coefficients α_{2k}^m and α_{2k+1}^m , it follows that there is a very fast computer algorithm, a *fast wavelet transform* W_H , which performs the transformation:

$$(f_0, f_1, \dots, f_{N-1}) \xrightarrow{W_H} (\alpha_0^0, \beta_0^0, \beta_0^1, \beta_1^1, \dots, \beta_{N/2-1}^{R-1}). \quad (19)$$

Note also that (18) is invertible:

$$\alpha_{2k}^m = \frac{1}{\sqrt{2}}\alpha_k^{m-1} + \frac{1}{\sqrt{2}}\beta_k^{m-1} \quad (20)$$

$$\alpha_{2k+1}^m = \frac{1}{\sqrt{2}}\alpha_k^{m-1} - \frac{1}{\sqrt{2}}\beta_k^{m-1}.$$

Hence, the fast wavelet transform in (19) is invertible, by another very fast computer algorithm, which we denote by W_H^{-1} :

$$(\alpha_0^0, \beta_0^0, \beta_0^1, \beta_1^1, \dots, \beta_{N/2-1}^{R-1}) \xrightarrow{W_H^{-1}} (f_0, f_1, \dots, f_{N-1}). \quad (21)$$

Since $\int_{-\infty}^{\infty} \psi(t) dt = 0$ and ψ is supported on the interval $[0, 1)$, it follows that when $f(t)$ is constant on the interval $[2^{-m}n, 2^{-m}(n+1))$, then β_n^m will be zero. The transformation

$$(2, 2, 2, 2, 4, 4, 4, 12) \xrightarrow{W_H} (8\sqrt{2}, -4\sqrt{2}, 0, -4, 0, 0, 0, -4\sqrt{2})$$

illustrates this phenomenon in an elementary way. A more complex illustration is shown in Fig. 10. In Fig. 10(a), we show 1024 samples of the signal

$$f(t) = \lfloor 5(\sin 8\pi t)(\cos 4\pi t) \rfloor / 5, \quad 0 \leq t < 1,$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer function. The signal $f(t)$ is a step function, constant over several subintervals of $[0, 1)$. The fast Haar transform of the sample data (f_n) will produce a large number of zeros. Consequently, a good approximation of the original signal can be obtained with just a fraction of the non-zero Haar coefficients. In Fig. 10(b) we show the graph of W_H^{-1} applied to the 195 highest magnitude Haar coefficients [all other coefficients set equal to zero]. This reconstructed signal is a good approximation of the original.

One way to quantify the accuracy of the approximation is to use relative R.M.S. error. Given a discrete signal $\{f_j\}_{j=0}^{N-1}$ and its reconstruction $\{g_j\}_{j=0}^{N-1}$, the *relative R.M.S. error* is defined by

$\mathcal{D}(f, g) = \sqrt{\sum_{j=0}^{N-1} |f_j - g_j|^2} / \sqrt{\sum_{j=0}^{N-1} |f_j|^2}$. For convenience, we shall refer to $\mathcal{D}(f, g)$ as *rel. error*. As a standard of good approximation, we shall require that a reconstructed signal approximate the original with a rel. error of at most 10^{-2} . The reconstructed Haar signal in Fig. 10(b) approximates the original signal with the required rel. error of 10^{-2} . By comparison, we show a discrete Fourier approximation in Fig. 10(c), using the top 195 highest magnitude DFT values. The rel. error is then 2.6×10^{-2} . This increase in rel. error is reflected in the ragged appearance of the Fourier approximation. In order to obtain a rel. error of 10^{-2} , the Fourier approximation required 448 coefficients. For this step function signal, the Haar series clearly out-performs the Fourier method in all respects.

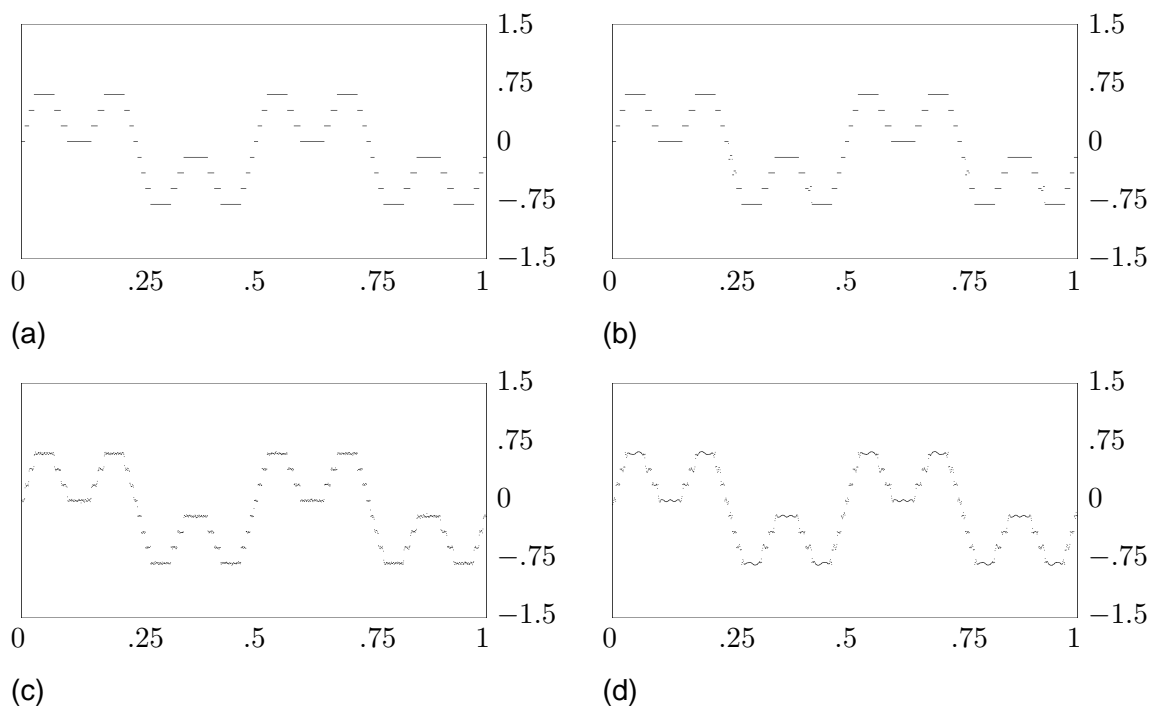


Figure 10: (a) Step function data, 1024 points. (b) Haar series, 195 highest magnitude coefficients: rel. error 10^{-2} . (c) Fourier approximation, 195 highest mag. coefficients: rel. error 2.6×10^{-2} . (d) Coif18 series, 195 highest mag. coefficients: rel. error 3.2×10^{-2} .

The simple nature of the functions in the Haar system makes it easy to study the structure of Haar wavelets. Their lack of smoothness, however, makes them a poor basis for many applications. While they performed well in the step function example just considered, this same example illustrates that Haar series produce step function approximations for all types of signals. If our samples are taken from a smoothly varying function, then the step function approximations provided by Haar series will be unsuitable. In our work, we shall instead use Coiflets, which are wavelet bases of I. Daubechies and R. Coifman [7], [4].

The Coiflets are an infinite class of wavelets for $L^2(\mathbf{R})$. Similar to the Haar wavelets, these wavelets have compact support: for each such system the generating scaling function $\phi(t)$ and generating wavelet $\psi(t)$ are zero off of intervals of finite length. Also, just as with the Haar system, we have

$$\int_{-\infty}^{\infty} \phi(t) dt = 1 \quad \text{and} \quad \int_{-\infty}^{\infty} \psi(t) dt = 0.$$

These last two integral conditions are true for all wavelets on $L^2(\mathbf{R})$. The added benefits of Coiflets are their smoothness and the *vanishing moment conditions*:

$$\int_{-\infty}^{\infty} t^k \phi(t) dt = 0 \quad \text{and} \quad \int_{-\infty}^{\infty} t^k \psi(t) dt = 0$$

for some positive integers k .

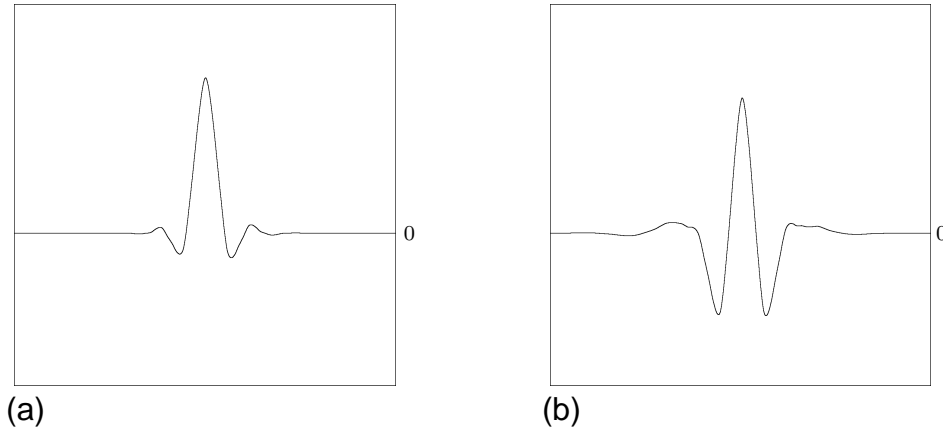


Figure 11: (a) Coif18 generating scaling signal. (b) Coif18 generating wavelet.

For the remainder of our work, we shall use the Coiflets known as the *Coif18 system*. For the Coif18 system, both the generating scaling function $\phi(t)$ and the generating wavelet $\psi(t)$ are supported on the interval $[-6, 11]$. In Fig. 11, we show graphs of ϕ and ψ . These Coif18 generating functions are twice continuously differentiable. That implies that any partial sum for a Coif18 wavelet series will be twice continuously differentiable as well. Such smoothness for Coif18 partial sums will provide better approximations for smoothly varying signals than the step function partial sums provided by Haar series. Besides smoothness, the Coif18 generating wavelet $\psi(t)$ has six vanishing moments

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0, \quad 0 \leq k \leq 5,$$

and the Coif18 generating scaling function $\phi(t)$ also has six vanishing moments

$$\int_{-\infty}^{\infty} t^k \phi(t) dt = 0, \quad 1 \leq k \leq 6.$$

The benefits of vanishing moments can be seen when computing the wavelet coefficients. For instance, since $\int_{-\infty}^{\infty} \psi(t) dt = 0$, if $f(t)$ is constant on the interval $[2^{-m}(n-6), 2^{-m}(n+11))$, then β_n^m will be zero. Similarly, all of the vanishing moments for ψ imply that if $f(t)$ is a polynomial of degree less than 6 on this same interval, then β_n^m will also be zero. Put another way, if f can be closely approximated by a polynomial of degree less than 6 on an interval containing the support of $\psi_{m,n}$, then β_n^m will be approximately zero. We shall see that these facts are useful for compressing signals. The vanishing of the moments of the generating scaling function ϕ are important for making the approximation $\alpha_n^R \approx 2^{-R/2} f_n$ which we discussed above. This approximation is much

better justified for Coif18 scaling functions than for Haar scaling functions (see [15] for more details), and it justifies replacing (α_n^R) by the signal samples (f_n) .

With this background for the Coif18 system, we can now describe how they allow us to compress signals. As with the Haar wavelet system, the Coif18 system allows for the computation of $\{\alpha_n^{m-1}\}$ and $\{\beta_n^{m-1}\}$ by applying an orthogonal transformation to a small number (18 to be precise) of adjacent values of $\{\alpha_n^m\}$. (The small number of adjacent values needed is another advantage of using Coif18 wavelets.) Iterating this process R times yields a fast wavelet transform W_C , which converts the initial data $(\alpha_0^R, \dots, \alpha_{N-1}^R) = (f_0, \dots, f_{N-1})$ as follows:

$$(f_0, f_1, \dots, f_{N-1}) \xrightarrow{W_C} (\alpha_0^0, \beta_0^0, \beta_0^1, \beta_1^1, \dots, \beta_{N/2-1}^{R-1}). \quad (22)$$

This transformation W_C has a fast inverse W_C^{-1} , as well. (See [19] and [20] for more details.) Since the Coif18 wavelet has six vanishing moments, many of the wavelet coefficients β_n^m are often zero or very small. This will give us our desired compression. We retain only coefficients which have significant size, all other coefficients are set to zero and are not transmitted. These high-magnitude coefficients will give us our binary codes for transmission. Binary encoding only a small set of high-magnitude coefficients will often produce a set of data much smaller in size than the N original signal values. At the receiving end, we use only these high-magnitude coefficients, along with zeros inserted for the coefficients we have dropped, and apply W_C^{-1} to reconstruct a wavelet series approximation to our signal.

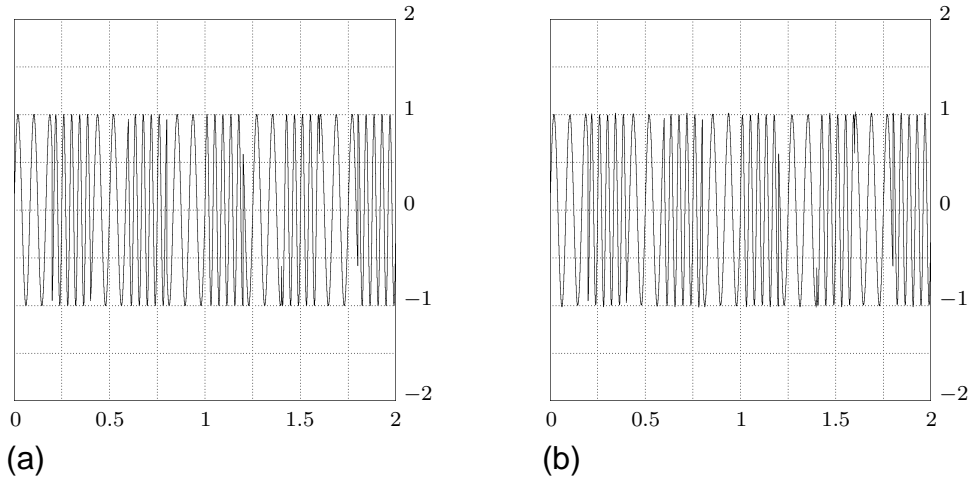


Figure 12: (a) Original signal. (b) Coif18 wavelet series with 310 terms.

We can illustrate such compression with the following alternating frequency sine function

$$\sum_{k=1}^4 \{ \chi_{[2kc, (2k+1)c]}(t) \sin(24\pi t) + \chi_{[(2k+1)c, (2k+2)c]}(t) \sin(48\pi t) \}, \quad (23)$$

with $c = 0.2$, over the interval $[0, 2)$ using 16,384 sample points. Note, for convenience, our function is once again discontinuous, but there also exists continuous functions with sharp transitions that would pass through this same sampled data. To obtain a rel. error of 10^{-2} , we find that the Coif18 wavelet requires only 301 coefficients compared to the 2,674 coefficients needed

for the Fourier approximation. Comparing the Fourier approximations shown in Fig. 13 with our Coiflet example in Fig. 12, we can also see that the errors in the Fourier case are not seen with our wavelet reconstruction. In Fig. 13(a), we see that using the same number of terms for the Fourier approximation as for the wavelet series produces a much poorer reconstruction. Its rel. error is 8.5×10^{-2} which is 8.5 times larger than the rel. error for the 301-term Coif18 wavelet series. Considering that the total number of signal values is 16,384, the 301 non-zero coefficients for the Coif18 reconstruction represents a significant compression of the signal.

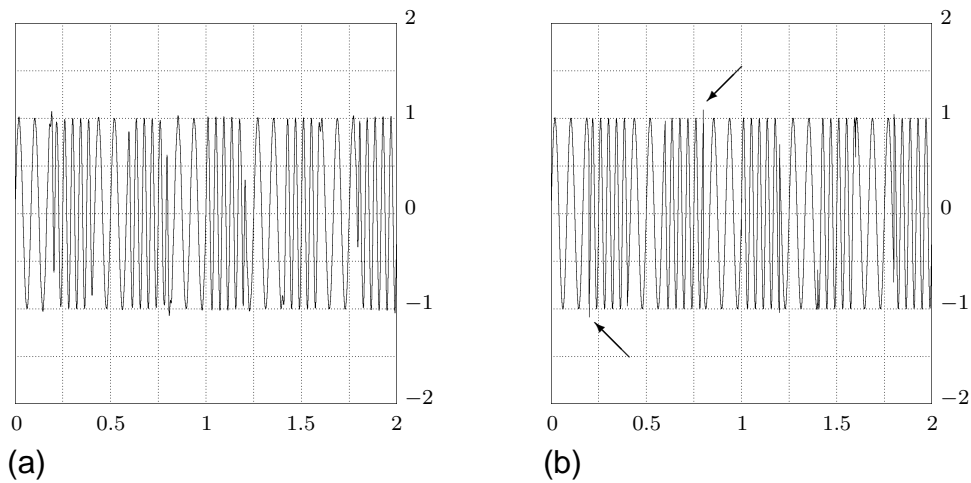


Figure 13: (a) Fourier approximation with 301 terms for signal in Fig. 12(a). (b) Fourier approximation, 2,674 terms. The arrows indicate some prominent defects in the Fourier approximation.

The two compression examples in this section illustrate some of the variety of signal data which can be compressed using wavelets. For step function data, which are locally constant, the Haar wavelets performed well—better than a Fourier approximation, and also better than Coif18 wavelets [compare Figures 10(b) and (d)]. This is due to the step nature of the Haar system. For more smoothly varying data, as in samples from speech and music, the Coif18 wavelets give excellent compression, clearly superior to Fourier approximations. This was illustrated with a discretely sampled signal with sharp transitions.

We have seen how wavelets can often allow us to compress our data quite well. Let us carefully reexamine the properties of wavelets that allowed us to do this. By the definition of a wavelet, the basis functions $\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n)$ can zoom in on particular areas of a signal. As m increases, the supports of these basis functions decrease in size, and varying n allows us to move along to particular areas to examine the signal. This allows the wavelet coefficients $\beta_n^m = \int_{-\infty}^{\infty} f(t)\psi_{m,n}(t) dt$ to pick up information about local behaviors of the function $f(t)$. The vanishing moment conditions help us to compress the data from smoothly varying signals because many of the wavelet coefficients β_n^m are either zero or very small. For different types of signals, different wavelet systems can be employed. In particular, for step functions, the Haar wavelet system performs very well. While for smoother functions, a smoother wavelet system like Coif18 works much better. The Coif18 system combines a large number of vanishing moments for both wavelets and scaling functions (hence good compression) with a relatively small support (thus allowing for rapid computation). The Coif18 system is just one of many wavelet systems that can be

employed for compression. Some compression schemes make use of multiple systems, choosing the best system for a given signal (and transmitting a small amount of overhead to the receiver to identify which system was used). Besides wavelet series, localized Fourier series—Fourier expansions restricted to smooth time-windowed portions of the signal—have been found to be especially effective for compressing audio signals [14]. More details on data compression can be found in [13], [16], and [20].

3 Conclusion

In this paper we have seen how a PCM signal can carry the bits of a digital signal by carrying the 1's of a binary code with a different frequency than the 0's. By using more frequencies, this PCM signal can carry large numbers of digital signals simultaneously. PCM signals are also highly resistant to corruption by additive noise.

We next examined three ways to represent a function with such digital codes. The codes came from the coefficients for a discrete Fourier transform, a discrete Haar wavelet series, and a discrete Coif18 wavelet series. We saw how wavelets allowed us to better compress our signals so that we needed less data to represent our function.

In the references listed, several would be appropriate reading for undergraduate students. Students may find [3] interesting if they wish to read how several radio signals can be transmitted simultaneously. References [17] and [11] will give the reader more of the theory and applications of Fourier series. Finally, the reader may wish to learn more about wavelets and their uses by consulting [1], [9], [20], and [19].

References

- [1] <http://www.uwec.edu/academic/curric/walkerjs/>
- [2] R.L. Boylstad. 1994. *Introductory Circuit Analysis*, 9th edition. Prentice-Hall, Upper Saddle River, NJ.
- [3] C. Benson. 1993. How to tune a radio. In “Applications of Calculus,” Philip Straffin, Ed., *Resources for Calculus*, 3:126-134.
- [4] C.S. Burrus, R.H. Gopinath, H. Guo. 1998. *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice-Hall, Englewood Cliffs, NJ.
- [5] P. Catala. (1997). Pulse Code Modulation. Chap. 23 of *The Communications Handbook*, CRC Press, Boca Raton, FL.
- [6] I. Daubechies. 1988. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 51: 909-996.
- [7] I. Daubechies. 1992. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA.
- [8] T.L. Floyd. 2001. *Electronics Fundamentals*, 5th edition. Prentice-Hall, Upper Saddle River, NJ.

- [9] M. Frazier. 1999. *An Introduction to Wavelets through Linear Algebra*. Springer-Verlag, New York, NY.
- [10] J.R. Higgins. 1985. Five short stories about the cardinal series. *Bull. Am. Math. Soc., New Ser.*, 12: 45–89.
- [11] S. Kelly. 2000. Using the Shannon sampling theorem to design compact discs. *The UMAP Journal of Undergraduate Mathematics and Its Applications*, 21(2): 157-166.
- [12] S. Mallat. 1988. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbf{R})$. *Transactions of the American Mathematical Society*. 315(1):69-87.
- [13] S. Mallat. 1998. *A Wavelet Tour of Signal Processing*. Academic Press, New York, NY.
- [14] H.S. Malvar. 1992. *Signal Processing with Lapped Transforms*. Artech House, Norwood.
- [15] H.L. Resnikoff, R.O. Wells. 1998. *Wavelet Analysis. The Scalable Structure of Information*. Springer-Verlag, New York, NY.
- [16] M. Vetterli, J. Kovačević. 1995. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ.
- [17] J. Walker. 1988. *Fourier Analysis*. Oxford University Press, New York, NY.
- [18] J. Walker. 1996. *Fast Fourier Transforms*. CRC Press, Boca Raton, FL.
- [19] J. Walker. 1997. Fourier analysis and wavelet analysis. *Notices of the American Mathematical Society*, 44(6): 658-670.
- [20] J. Walker. 1999. *A Primer on Wavelets and their Scientific Applications*. CRC Press, Boca Raton, FL.