

3.4.5 IDFT in MUSIMAT

The code for the IDFT in the MUSIMAT programming language is as follows:

```
Complex I = Complex( 0.0, 1.0 ); // define imaginary number i
Complex E = Complex( 2.718281, 0.0 ); // define e

ComplexList IDFT( ComplexList X ) {
    Integer N = Length( X ); // get length of X
    ComplexList x; // place to store result
    Complex I2pidN = I * 2.0 * Pi / N; // calculate constants once

    For ( Integer n = 0; n < N; n++) {
        x[n] = 0.0; // prepare to accumulate
        For ( Integer k = 0; k < N; k++) {
            x[n] = x[n] + X[k] * Exp( E, I2pidN * n * k );
        }
    }
    Return( x );
}
```

3.5 Analyzing Real-World Signals

So far, we've restricted the DFT and IDFT to periodic waveforms containing only harmonics (integer multiples) of the fundamental analysis frequency. This is fine as far as it goes, but most actual musical signals are not so well-behaved. What happens if we try to analyze a signal whose spectrum isn't aligned with the fundamental analysis frequency? Consider the following waveform:

$$x(n) = \sin\left(f2\pi\frac{n}{N}\right), \quad \text{Nonintegral Test Signal (3.31)}$$

where $N = 16$ and frequency $f = 3/4$.

If the fundamental analysis period of the DFT is also $N = 16$, then $f = 3/4$ is clearly not an integer multiple of the fundamental analysis frequency. The DFT would receive input as shown in figure 3.14. What the DFT will do with this signal—what it does with every input signal—is interpret it as *one period of an infinitely repeating periodic function* at the fundamental analysis frequency. That is, the DFT operates on the example input function $x(n)$ as though it were like the one in figure 3.15. This is the *periodic extension* of the Fourier transform.

Notice the discontinuity in the waveform in figure 3.15. Periodic discontinuities in a waveform produce a spectrum with many high-frequency harmonics. The DFT “hears” a click in $x(n)$ when the waveform has such a discontinuity. This is revealed by looking at the magnitude spectrum that the DFT produces from this signal (figure 3.16). We see that there is some energy at all analyzed

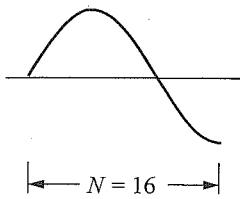


Figure 3.14
Nonharmonic test signal.

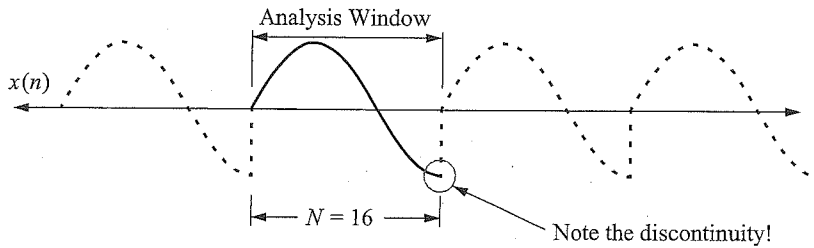


Figure 3.15
How the DFT interprets a nonperiodic input signal.

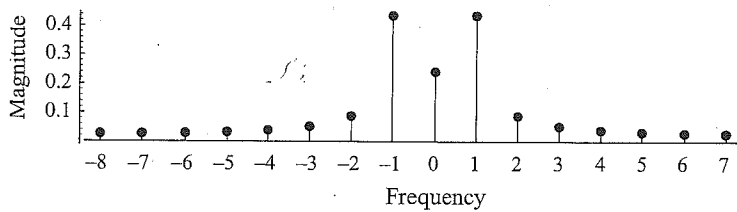


Figure 3.16
Magnitude spectrum of a nonintegral test signal.

frequencies. This is a problem, because we know from equation (3.31) that there is only one frequency component in the test signal at $f = 3/4$ Hz.

There are actually two problems here:

- *Picket Fence Effect.* We're trying to represent a frequency that is not an integer multiple of the fundamental analysis frequency f_N , so the results don't fit properly as harmonics of f_N . We are unable to view the underlying continuous spectrum because the DFT limits us to integer multiples of the fundamental analysis frequency f_N . This is analogous to trying to observe a row of evenly spaced trees through a picket fence.
- *Leakage.* Discontinuities at the edge of the analysis window spray noise throughout the rest of the spectrum. This phenomenon is called *leakage* because energy that should be in one spectral harmonic spreads away (leaks) into adjacent harmonics.

We must solve these problems because they arise whenever we try to analyze a signal with frequencies that are not locked to the rate of the analyzer.

3.5.1 Solving the Picket Fence Problem

Because of the picket fence phenomenon, we don't know whether two adjacent harmonics represent distinct partials or a single partial whose frequency is not a harmonic of the fundamental analysis frequency. There are a couple of things we can do to disambiguate these two cases.

If possible, resample the data, increasing the sampling rate R and/or the fundamental analysis frequency N until there are enough data points in the spectrum to disambiguate the two interpretations.

Alternatively, pad the signal to be analyzed with M additional zero-valued samples. This does not change its spectrum but increases its length. If we then increase the size of the analysis window to include the zero-valued padding samples in the DFT, we decrease the fundamental analysis frequency f_N , increasing the frequency resolution of the spectrum.⁵ For example, if to a signal of length N samples we add $M = N$ additional zero-valued samples (doubling the signal's length) and take the DFT of $M + N$ samples, we increase the spectral frequency resolution by a factor of $(M + N)/N$. This is like adjusting the distance between the pickets in the fence until they line up with the evenly spaced trees.

3.5.2 Solving the Leakage Problem

The leakage problem is more serious. The best we can do is devise a work-around. We create a function exactly as long as the analysis window that gradually fades in and fades out at the edges. If we multiply the signal to be analyzed by this function, we decrease the effect of any discontinuities at the edges of the analysis window because the discontinuities are heavily attenuated at the analysis window edges. This helps reduce the impact of the discontinuities at the analysis window edges on the resulting spectrum.

But there's no free lunch because any alteration of the input signal will have some effect on the resulting spectrum. This is because the function that fades in and fades out is itself a signal, and it too has a spectrum. Later I describe several of these fade-in/fade-out functions and show their spectra.

To summarize, when analyzing waveforms that are not strictly harmonics of the fundamental analysis frequency:

- The spectrum of partials that are inharmonic to the fundamental analysis frequency can still be interpreted correctly because the DFT splits fractional frequencies proportionately into adjacent harmonics. We can interpolate between them to recover the fractional frequency components.
- The signal can be faded in and out at the edges of the analysis window to reduce the broadband spectral influence of discontinuities that would otherwise occur. The general term for this process is called *windowing*. The resulting spectrum will be a better approximation of the actual underlying signal, but we must account for the effects that windowing has on the resulting spectrum.

3.6 Windowing

To focus the DFT on part of the input signal, we can extract part of the signal or we can window it. So far, we've set the DFT summation limits to extract the desired portion of the signal (see equation (3.8)), covering the range of $n = 0$ to $n = N - 1$. Alternatively, we can multiply the function by 0 everywhere except for the N samples we want to select, which we multiply by 1. This is windowing with a rectangular function, shown in figure 3.17 as the function $w(n, N)$, defined as

$$w(n, N) = \begin{cases} 1 & 0 \leq n < N, \\ 0 & \text{otherwise,} \end{cases} \quad \text{Rectangular Window Function (3.32)}$$

where N is the number of samples in the window and n indexes the window function.

3.6.1 Windowed DFT

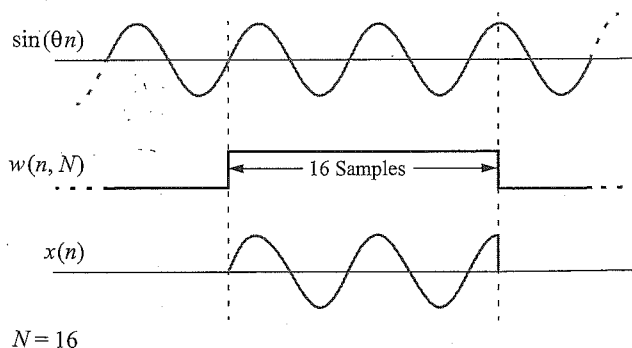
We can rewrite the DFT to express windowing explicitly as follows:

$$X(k) = \sum_{n=-\infty}^{\infty} x(n)w(n, N)e^{-i2\pi kn/N}, \quad \text{Windowed DFT (3.33)}$$

where $w(n, N)$ is the windowing function given in equation (3.32). Note that the sample index n now traverses all of time, but because of the windowing function, the result is the same as equation (3.8).

3.6.2 Tapering Functions

Windowing can introduce discontinuities at the edges of the analysis window if the underlying waveform is not a harmonic of the fundamental analysis frequency. The signal $x(n)$ selected by the window in figure 3.17 has such a discontinuity at its right edge. As shown in the figure, the underlying



$N = 16$

Figure 3.17
Windowing with a rectangular function.

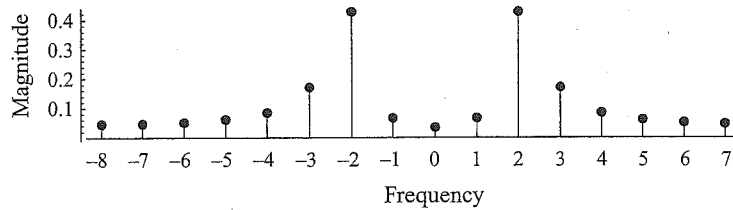


Figure 3.18
Spectrum of a rectangular function.

function $x(n)$ is 16 samples long, and by inspection we see that it contains 2.25 periods of a sine wave; hence the underlying function can be written as a periodic function:

$$\sin \frac{2\pi fn}{N},$$

where $N = 16$ and $f = 2.25$.

The magnitude spectrum of this signal (figure 3.18) shows that the DFT has added many spurious high-frequency harmonics, introduced by the discontinuity in the analyzed signal.

Analysis of nonharmonic signals via either sample extraction or windowing with a rectangular function can result in spurious energy estimates such as this unless we take steps to prevent it. We can diminish the impact of discontinuities at the edge of the analysis window by replacing the rectangular windowing function with a function that tapers to zero at its edges. Such functions are called *apodization functions*, or *tapering functions*. Tapering cannot be done with plain sample extraction; therefore windowing is necessary.

Triangular Window The simplest tapering function is the *triangular window*. Its shape is shown in figure 3.19 as $w(n, N)$. It just consists of a complementary pair of slopes that make a tent function. Its equation is

$$w(n, N) = \begin{cases} 1 - \left| \frac{n - (N/2)}{N/2} \right|, & 0 \leq n \leq N, \\ 0 & \text{otherwise,} \end{cases} \quad \text{Triangular Window (3.34)}$$

where N is the length of the window and n is the current sample. The operator $||$ takes the absolute value of the expression it contains. The triangular window function w behaves differently depending upon whether the value of n is inside or outside the range of 0 to N . If it is outside, then w simply returns 0; otherwise it returns the appropriate point of the triangular function.

Figure 3.19 shows the result of windowing a sinusoid with a triangular window function. The resulting spectrum is shown in figure 3.20. The magnitude spectrum in figure 3.20 is much less noisy. Only frequencies near those that actually contain energy show significant strength. Thus, we have effectively enabled the DFT to be used with realistic signals because now we can remove much of the spurious noise that is introduced by discontinuities at the analysis window edges.

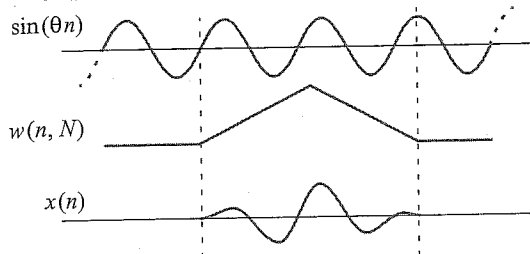


Figure 3.19
Windowing with a triangular function.

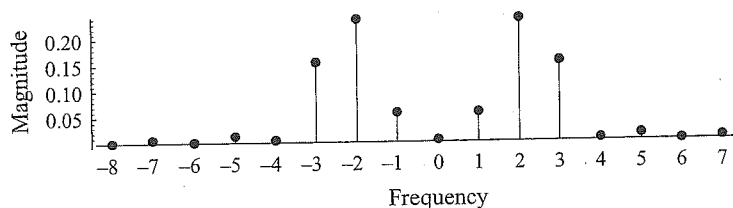


Figure 3.20
Spectrum of triangular windowing.

Whereas the highest peak in the rectangular-windowed DFT is about 0.43, the highest peak for the triangular-windowed DFT is only about 0.24. It is lower because applying the triangular function attenuates the signal wherever the triangular window function is less than 1.0 (everywhere but in the middle).

Other Window Functions The triangular window is but one of many windowing functions. There seems to be a rather bewildering variety of them—the Bartlett window, Welch window, Parzen (triangular) window, Hann or hanning window, Hamming window, Blackman window, Lanczos window, Kaiser window, Gaussian window, and so on.

While each of these windows has a particular advantage in certain situations, any windowing function that reduces the discontinuities at the edges of the analysis window is a big improvement over the rectangular function. Here's a look at some of the standard window functions.

Hann (Hanning) Window The equation for the *Hann window* (named after Julius von Hann and often referred to as the *hanning window*)⁶ is

$$H(x, n, N) = \begin{cases} (1 - a) \cos\left(2\pi\frac{n}{N} + \pi\right) + a, & 0 \leq n < N, \\ 0 & \text{otherwise,} \end{cases} \quad (3.35)$$

where $a = 1/2$. It is shown in figure 3.21a.

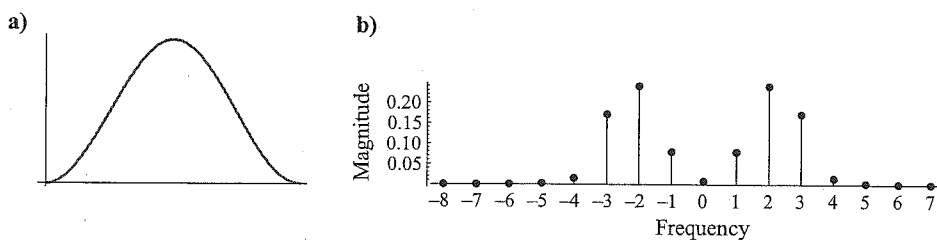


Figure 3.21
Hann window and resulting spectrum.

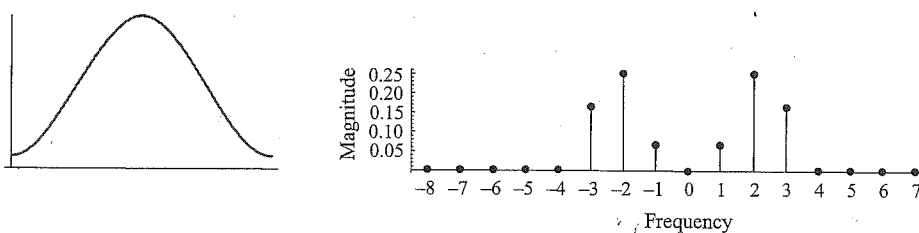


Figure 3.22
Hamming window and resulting spectrum.

Because it is just an inverted cosine wave scaled by 0.5 and offset by 0.5, this window is sometimes called the raised inverted cosine window. Its advantage is that, unlike the triangular window, it has no sharp edges (no sudden change in derivative) at all, so it is more effective at eliminating spurious artifacts from the analysis. Figure 3.21b shows the magnitude spectrum of the test signal, equation (3.31), windowed with the Hann window. Note that the peak amplitude is around 0.24, so there is some energy loss due to the overall attenuation of the signal, similar to what happened with the triangular window.

Hamming Window The equation for the *Hamming window* (named after Richard W. Hamming) is the same as equation (3.35) except that $a = 0.54$ (see figure 3.22). The elevated value for a causes the Hamming window to let in more energy overall and, in particular, let in a little energy at the edges of the analysis window. The amplitude peak is a little higher than with the Hann window, but it also lets some of the broadband energy from the window edge back in.

Figure 3.23 shows the triangular, Hann, and Hamming windows superimposed. Note that both the Hann and Hamming windows emphasize the middle part of the input signal more than the triangular window does.

Bear in mind that windowing a signal always modifies its spectrum. If we perform an inverse transform on this spectrum, we will get back the *windowed version* of the signal. So, for example, performing the IDFT on the spectrum in figure 3.20 would reproduce the waveform $x(n)$, shown at the bottom of figure 3.19, not the original signal $\sin \theta n$ shown at the top of that figure.

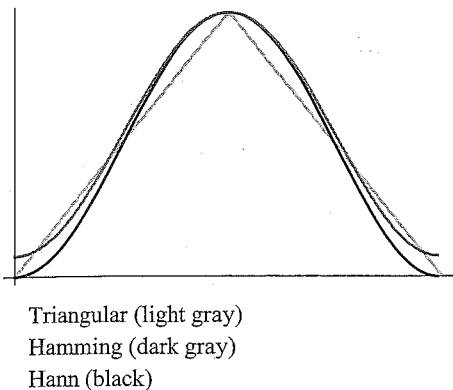


Figure 3.23
 Triangular, Hann, and Hamming windows.

If we must window a signal $x(n)$ but need to get the original back after DFT/IDFT analysis/synthesis, we can divide the reproduced $x(n)$ by the windowing function to un-compensate $x(n)$. Any part of the signal windowed to zero is irretrievably lost (and very attenuated values might not be properly restored because of limited computer arithmetic precision). This means there is a problem with the triangular and Hann windows because they go to zero at their extremes, so we'd end up dividing by zero at those points, which is not meaningful. However, we could do this with the Hamming window because it does not reach zero at its extremities (which may have been one of the motivations for its development).

3.7 Fast Fourier Transform

With the addition of the technique of windowing, the DFT is able to handle real-world signals—almost. The next hurdle to its practical use is the sheer amount of computation required to analyze realistic-sized signals. If we want to perform a DFT of length N , then at each stage we must perform a complex multiplication of $w(n)$ against all N phasor frequencies. Since we must repeat this operation for all N input samples, the total number of stages is on the order of N^2 . For signals beyond a certain size, we will quickly run out of computing power, patience, or both. The fast Fourier transform (FFT) reduces the number of computations from N^2 to $N \log_2 N$. For small values, the FFT does not have a big advantage, but as N grows, the FFT outperforms the DFT by enough to make a substantial difference, and it is widely used.

To take a practical example, say we wanted to perform the Fourier transform on just 1 second of a stereo audio signal, recorded at the conventional rate of $R = 44,100$ samples per second per channel. Since it's stereo, there are two channels, and we must calculate two DFTs, one for each of the two channels. Each channel has $N = 44,100$, for a combined total of $2N = 88,200$ samples. Further, say we perform the calculations on a computer that can perform 1,000,000 stages of the

Fourier transform per second (a fast computer by today's standards). For the FFT it would take about one and a half seconds to do the calculation, whereas for the DFT it would take a little over two hours.

	N	Order	Stages	Time
FFT	88,200	$N \log_2 N$	$= 1.45 \times 10^6$	$= 1.45$ s
DFT	88,200	N^2	$= 7.78 \times 10^9$	$= 7779.24$ s, or 2.16 hours

The FFT requires that the input signal length be a power of 2. But this requirement can be worked around easily: for length N choose the nearest power of 2 greater than N , and set the extra samples to zero. (All this does is slightly compress the resulting spectrum, exactly the same as padding the DFT with zeros.)

The FFT achieves its efficiency by reducing the number of stages that must be performed. The first step in reducing the computation, according to the Danielson-Lanczos (1942) lemma, is to rewrite the DFT into the sum of two smaller DFTs, each of length $N/2$. One DFT operates on just the $N/2$ even-numbered samples, and the other operates on just the $N/2$ odd-numbered samples. Here is a derivation. Let $W_N = e^{-i\omega/N}$ so that the DFT can be written as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad 0 \leq k \leq N-1,$$

(ignoring the normalization of the sum by $1/N$ for simplicity).

The order in which we sum the terms doesn't matter (that is, addition is commutative). We can add two summations, one indexing only the even terms, the other only the odd terms. Stepping through just the even terms is equivalent to indexing by $2n$, whereas for the odd terms it's equivalent to indexing by $2n+1$. Each DFT must perform only $(N/2) - 1$ summations:

$$X(k) = \sum_{n=0}^{(N/2)-1} x(2n) W_N^{k(2n)} + \sum_{n=0}^{(N/2)-1} x(2n+1) W_N^{k(2n+1)}, \quad (3.36)$$

where N is even.

Let's take a closer look at the probe phasor in the odd DFT:

$$W_N^{k(2n+1)} = e^{-i\omega k(2n+1)/N}.$$

Remembering that $x^{a+b} = x^a x^b$, we can expand this into

$$e^{-i\omega k(2n+1)/N} = e^{i\omega k(2n)/N} \cdot e^{-i\omega k/N}.$$

Notice that the term $e^{-i\omega k/N}$ does not depend upon n and that it appears in every term of the odd DFT. That means we can factor it out of the entire odd DFT summation. Since $e^{-i\omega k/N} = W_N^k$ we

can rewrite equation (3.36) to read

$$\begin{aligned} X(k) &= \sum_{n=0}^{(N/2)-1} x(2n) W_N^{k(2n)} + W_N^k \sum_{n=0}^{(N/2)-1} x(2n-1) W_N^{k(2n)} \\ &= X_E(k) + W_N^k \cdot X_O(k), \end{aligned}$$

where $X_E(k)$ and $X_O(k)$ are the even and odd DFTs. Notice that the probe phasors for the even and odd DFTs are now the same. The final summation of the odd DFT is additionally multiplied by W_N^k . This demonstrates that the DFT can be rewritten as a sum of two half-length DFTs.

This lemma can be recursively applied (so long as N is even), so we can progressively divide these two DFTs into 4, 8, . . . DFTs of length $N/4$, $N/8$, . . . until we have subdivided all the way down to DFTs of length 1. What is a DFT of length 1? If we set $N = 1$ in equation (3.8), the DFT equation reduces to $X(0) = x(0)$. So the DFT of a signal of length 1 is just the value of the sample.

The recursive application of the Danielson-Lanczos lemma leads directly to the so-called radix 2 Cooley-Tukey (1965) fast Fourier transform. (A radix 4 FFT would partition the DFT into four subtransforms of length $N/4$.)

In outline, the FFT algorithm is as follows. Before the main FFT algorithm begins, the data are rearranged (by a technique called bit reversal) into a form that can be more efficiently accessed by the algorithm, and the values of the probe phasor are precomputed. The FFT algorithm itself consists of $\log_2 N$ stages in which successively longer subtransforms are computed from the previous stages. This process is repeated N times for a total of $N \log_2 N$ times.

Since the FFT implements exactly the same transform as the DFT, only more efficiently, I don't pursue the implementation of the FFT further, but this information should hopefully allow readers to make sense of other treatments of the subject (see Bracewell 1999; Smith 2003).

3.8 Properties of the Discrete Fourier Transform

Operations such as addition, multiplication, and shifting in the time domain have corresponding operations in the frequency domain, and vice versa. This section sets out some of the Fourier transform's most important properties.

3.8.1 Linearity of the Fourier Transform

The Fourier transform establishes a mathematical relation between periodic signals and their associated spectra. How the Fourier transform relates to the properties of superposition and proportionality determines whether it is a linear operation or not.

Superposition As shown in figure 3.24, when two signals are added together, their spectra are added also. The figure shows the addition of the signals $f(t)$ and $g(t)$ and of their spectra $F(k)$ and