# FFT.[http://alwayslearn.com]

8 maja 2017 13:05

http://alwayslearn.com/DFT%20and%20FFT%20Tutorial/DFTandFFT_TheDFT.html
http://alwayslearn.com/DFT%20and%20FFT%20Tutorial/DFTandFFT_BasicIdea.html

## 1. The basic Idea.

**A Fourier Transform converts a wave in the time domain to the frequency domain.**

Note, for a full discussion of the Fourier Series and Fourier Transform that are the foundation of the DFT and FFT, see the Superposition Principle, Fourier Series, Fourier Transform Tutorial.

Every wave has one or more frequencies and amplitudes in it. An example is a sound wave. If someone speaks, whistles, plays an instrument, etc., to generate a sound wave, then any sample of that sound wave has a set of frequencies with amplitudes that describe that wave.

According to the mathematician **Joseph Fourier**, you can take a set of **sine waves** of different amplitudes and frequencies and sum them together to equal any waveform. These component sine waves each have a frequency and amplitude. A plot of frequency versus magnitude (amplitude) on an x-y graph of these sine wave components is a frequency spectrum, or frequency domain, plot. See Diagram 1, below.

**An inverse Fourier Transform converts the frequency domain components back into the original time wave.**

You can reassemble the time wave from the frequency components using the **Inverse Fourier Transform**. The inverse Fourier won't be discussed here, but after learning the Fourier the Inverse is very easy to learn, because the math is almost identical. Using the Fourier and Inverse Fourier together, not only can you reassemble the original wave, you can also change the time wave by altering its frequency components. You can add them, remove them, or tweak their values. This is a powerful method by which to change the character of the time wave.

A **DFT** is a **"Discrete Fourier Transform"**. An **FFT** is a **"Fast Fourier Transform"**. The **IDFT** below is **"Inverse DFT"** and **IFFT** is **"Inverse FFT"**. A DFT is a Fourier that transforms a discrete number of samples of a time wave and converts them into a frequency spectrum. However, calculating a DFT is sometimes too slow, because of the number of multiplies required. An FFT is an algorithm that speeds up the calculation of a DFT. In essence, an FFT is a DFT for speed. The entire purpose of an FFT is to speed up the calculations.

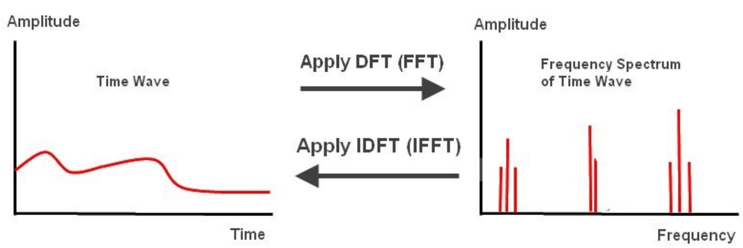



**Diagram 1**

The equation for the Discrete Fourier Transform is:

DFT:

$$F(n) = \sum_{k=0}^{N-1} x(k) e^{-jk2\pi \frac{n}{N}}$$

for n = 0 ... N - 1

**Equation 1**

Where F(n) is the amplitude at the frequency, n, and N is the number of discrete samples taken.

## 2. DFT

The Discrete Fourier Transform converts discrete data from a time wave into a frequency spectrum. Using the DFT implies that the finite segment that is analyzed is one period of an infinitely extended periodic signal.

The DFT equation:

$$F(n) = \sum_{k=0}^{N-1} x(k) e^{\frac{-j2\pi kn}{N}}$$

Equation 1

x(k) is the time wave that is converted to a frequency spectrum by the DFT.

Here are key concepts required to understand a DFT:

1. The "**sampling rate**", **sr**. The sampling rate is the number of samples taken over a time period. For simplicity we will make the time interval between samples equal. This is the "**sample interval**", **si**.
2. The **fundamental period**, **T**, is the **period** of all the samples taken. This is also called the "**window**".
3. The "**fundamental frequency**" is **f0**, which is 1/T. f0 is the first harmonic, the second harmonic is 2*f0, the third is 3*f0, etc.
4. The number of samples is **N**.
5. The "**Nyquist Frequency**", **fc**, is half the sampling rate. The Nyquist frequency is the maximum frequency that can be detected for a given sampling rate. This is because in order to measure a wave you need at least two sample points to identify it (trough and peak).
6. "**Euler's formula**" :
   $e^{jx} = \cos(x) + j\sin(x)$
7. The sampled part of the time wave, x(t), should be "typical" of how the wave behaves over all time that it exists.

8. $$W_N = e^{\frac{-j2\pi}{N}}$$

This notation makes handling the exponential easier. This is sometimes called the "**twiddle factor**."

For simplicity, we will sample a sine wave with a small number of points, N, and perform a DFT on it, then we will employ each of the concepts above. Note, the sine wave is a **time wave**, and could be any wave in nature, for example a sound wave. The horizontal axis is time. The vertical axis is amplitude.

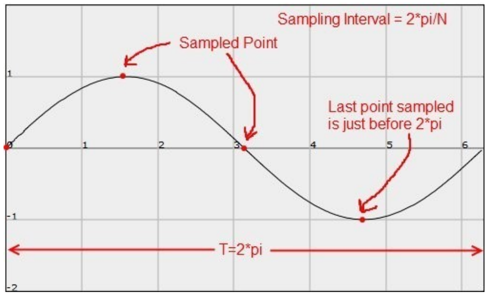



Diagram 1

Notice how in the diagram above we are sampling four points. The fundamental period, T, of the wave sampled is set to 2*pi. This applies to any wave we want to sample. The interval between samples is 2*pi/N, so in this case it is 2*pi/4. Thus, the interval between samples is pi/2 in this case.

The time wave is thus, x(k) = sin(pi/2*k) for k = 0 to N -1. The **last point sampled** is always the point just before 2*pi, because the wave is considered to be a repeating pattern and wraps around back to the value at k = 0, so you aren't missing any information.

Z tego wynika, że **time wave** jest wyrożone:

x(k) = sin(si * k);

gdzie:

x(k) jest wartością amplitudy fali dla danego czasu (co było wspomniane wyżej) w zakresie okresu przebiegu fali,

si jest okresem/czasem (dla tego przykładu stałym) pomiędzy próbkami,

k jest zwyczajnie liczbą porządkową kolejnej próbki.

(teraz bedzie korelacja powyższego do czasu [t])

We also need to know the time taken to sample the wave, so that we can tie it to a frequency. In our example, the time taken for the fundamental period, T, is 0.1 seconds (this value is measured when the wave is captured). That means the sine wave is a 10 Hz wave. Hertz = cycles per second. Also, the **sampling interval**, si, is the **fundamental period** time divided by the number of samples. So, si = T/N = (0.1)/4 seconds, or 0.025 seconds. The **sampling rate,** or frequency**,** sr, = 1/si = 40 Hz, or 40 samples per second.

For the sine wave, the value at each of the four points sampled is:

$$\begin{aligned}
&for\ x(k) = sin(\tfrac{k\pi}{2}) \\
&at\ k = 0, sin(0) = 0 \\
&at\ k = 1, sin(\tfrac{\pi}{2}) = 1 \\
&at\ k = 2, sin(\pi) = 0 \\
&at\ k = 3, sin(\tfrac{3\pi}{2}) = -1
\end{aligned}$$

And, before we plug into the DFT, some more on Wn, the **twiddle factor**, referenced above:

$$W_N^{kn}\ is\ the\ same\ as\ e^{\frac{-j2\pi kn}{N}}$$

$$W_4^{kn}\ for\ N = 4$$

The DFT formula, then, for a four point sample and with the twiddle factor is:

$$F(n) = \sum_{k=0}^{4-1} x(k) e^{\frac{-j2\pi kn}{4}} = \sum_{k=0}^{4-1} x(k) W_4^{kn}$$

Now, **Euler's Formula** for N=4:

$$W_4^{kn} = \cos(\frac{-2\pi kn}{4}) + j\sin(\frac{-2\pi kn}{4})$$

Equation 2

For the equation above, where k*n = 0 to N - 1, i.e. 0 to 3, here are the results:

$$\begin{aligned}
&W_4^{kn}\ values\ are: \\
&W_4^0 = 1 \\
&W_4^1 = -j \\
&W_4^2 = -1 \\
&W_4^3 = j
\end{aligned}$$

Notice that any additional integer values of kn will cycle back around. For example, kn = 4 cycles back to kn=0, so the value is 1. kn = 5 cycles back around to kn = 1, so the value is -j. The equation "**kn modulus 4**" determines which value of W is selected. Also, note that for larger samples the cycle is bigger. So for N=8 the equation would be "kn modulus 8". This is probably why W is called the "twiddle factor".

Now put this together for the DFT:

This is the general equation:
$$F(n) = x(0)W_4^{(0)(n)} + x(1)W_4^{(1)(n)} + x(2)W_4^{(2)(n)} + x(3)W_4^{(3)(n)}$$

Here is the DFT worked out for all four points and for four frequencies:

$$F(0) = x(0)W_4^{(0)(0)} + x(1)W_4^{(1)(0)} + x(2)W_4^{(2)(0)} + x(3)W_4^{(3)(0)}$$
$$F(0) = (0)(1) + (1)(1) + (0)(1) + (-1)(1) = 0$$

$$F(1) = x(0)W_4^{(0)(1)} + x(1)W_4^{(1)(1)} + x(2)W_4^{(2)(1)} + x(3)W_4^{(3)(1)}$$
$$F(1) = (0)(1) + (1)(-j) + (0)(-1) + (-1)(j) = -2j$$

$$F(2) = x(0)W_4^{(0)(2)} + x(1)W_4^{(1)(2)} + x(2)W_4^{(2)(2)} + x(3)W_4^{(3)(2)}$$

$$F(2) = (0)(1) + (1)(-1) + (0)(1) + (-1)(-1) = 0$$

$$F(3) = x(0)W_4^{(0)(3)} + x(1)W_4^{(1)(3)} + x(2)W_4^{(2)(3)} + x(3)W_4^{(3)(3)}$$
$$F(3) = (0)(1) + (1)(j) + (0)(-1) + (-1)(-j) = 2j$$

Evaluating the output data. Each F(n) value outputs a phase at a particular frequency. The frequency of the point is determined by the fundamental frequency multiplied by n, i.e. **f = f0*n**, where f0=1/T = 10Hz. The output values are the phase of the frequencies, which are represented by a real part and an imaginary part thus: real + j*imaginary. The fundamental frequency, first harmonic, is 10 Hz as calculated above. The **magnitude at a frequency** is Calculated thus **sqrt(real*real + imaginary*imaginary)**.

Below is a frequency spectrum plot for the sine wave determined from the DFT we just worked through:

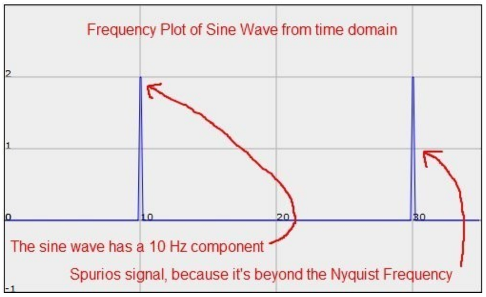



Diagram 2

The frequency plot is in the "**frequency domain**". The magnitudes are plotted in Diagram 2. The spike at 10 Hz shows that the DFT pulled out one of the frequencies that is in the sine wave. In fact, the sine wave is a 10 Hz sine wave, so that makes sense. However, the spike at 30 Hz should not be there, because there is no 30 Hz wave in the sine wave. So what accounts for that spike? Well, this is where the **Nyquist Frequency, fc,** mentioned above comes in. The Nyquist frequency is the cut off point above which the data from the DFT is no longer valid. The sampling rate is 40 Hz, and fc is half the sampling frequency, which means that any frequency above 20 Hz will not be valid in this case. So, the 30 Hz frequency is a spurious signal.

That completes analysis of a very simple wave.

Most waves will have many more frequencies in them, and thus many more spikes of various magnitudes along the frequency spectrum. For example, Diagram 3, below, is a plot of a triangle wave in time and its corresponding frequency spectrum:

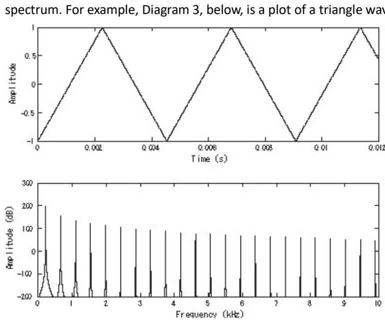



Diagram 3

The Next section is on the **FFT**. The FFT builds on the knowledge of the DFT described above, so it should be understood before moving on to the FFT.

## 3. FFT

This is a "decimation in time" FFT, because the input value to the FFT is the time wave, x(t). The time wave could be a sound wave, for example.

### Speed!

The only reason to learn the FFT is for speed. An FFT is a very efficient DFT calculating algorithm.

**How fast is an FFT versus a "straight" DFT?**

For a DFT $O(N^2)$ operations are required.
For an FFT only $O(NLog_2(N))$ operations are required.
For 1024 samples a straight DFT requires

$$1024^2 = 1048576\ operations$$

For the same number of samples an FFT requires only

$$1024 * Log_2(1024) = 1024 * 10 = 10240\ operations$$

Equation 1

This means that a 1024 sample FFT is 102.4 times faster than the "straight" DFT. For larger numbers of samples the speed advantage improves. For example, for 4096 samples the FFT is over 340 times faster.

x(k) is the time wave that is converted to a frequency spectrum by the DFT.

Learning the FFT is a bit of a challenge, but I'm hoping this tutorial will make it relatively easy to learn. Here is a basic outline of the tutorial:

1. First, you'll need to learn the **"Danielson-Lanczos Lemma"** (D-L Lemma). This will require long equation writing, but it's a vital component of the FFT. I'll give several examples.
2. You'll need to understand the "**twiddle factor**" --
3. 
4. $W_N^n$

. This was discussed a little during the DFT tutorial. It along with the "D-L Lemma" are essential to understanding how an FFT works.

1. Then the **"Butterfly Diagram"** will be explained. This builds on the first two concepts above. The Butterfly diagram is a diagrammatic representation of an FFT algorithm.
2. You'll also learn about the "reverse bit pattern" for data input and the reason for it.

### The Danielson-Lanczos Lemma

<< Previous Next >>

This is the first key to understanding the FFT. It takes quite a few steps, but I've broken the tutorial down into small digestible steps to make this as smooth as possible.

Here is the Danielson-Lanczos Lemma:

Danielson-Lanczos Lemma

$$F(n) = \sum_{k=0}^{N-1} x(k) e^{\frac{-j2\pi kn}{N}}$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x(2k) e^{\frac{-j2\pi kn}{(\frac{N}{2})}} + W_N^n \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

E = Even Term O = Odd Term

Equation 2

Note it is a DFT broken up into two summations of half the size of the original. The first summation is the "even terms", E, and the second is the "odd terms", O. W is the "**twiddle factor**", and understanding it is another key to understanding the FFT. Here is the "twiddle factor".

Twiddle Factor

$$W_N^n = e^{\frac{-j2\pi n}{N}}$$

Equation 3

## How to Expand the DFT

To expand the DFT into even and odd terms as in the lemma above, you do the following. For the even term you substitute 2k into k, then you create a summation of half the size of the original. For the odd term you substitute 2k + 1 into k, then create a summation of half the size of the original.

Here is a the summation halved:

$$\sum_{k=0}^{N-1} --> \ reduce\ by\ half\ --> \sum_{k=0}^{\frac{N}{2}-1}$$

The example below shows the process required for a first level expansion.

$$F(n) = \sum_{k=0}^{N-1} x(k) e^{\frac{-j2\pi kn}{N}} = E + O$$

Substitute 2k into k for the even term, and half the summation length:

$$E = \sum_{k=0}^{\frac{N}{2}-1} x(2k) e^{\frac{-j2\pi(2k)n}{N}} = \sum_{k=0}^{\frac{N}{2}-1} x(2k) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

Substitute 2k+1 into k for the odd term, and half the summation length:

$$O = \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi(2k+1)n}{N}}$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{N}{2})}} e^{\frac{-j2\pi n}{N}}$$

$$= W_N^n \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

Equation 4

Note the "**twiddle factor**" above and where it comes from.

And putting the even and odd terms together from above, we get the Danielson Lanczos first level expansion:

$$F(n) = E + O$$

$$= \sum_{k=0}^{\frac{N}{2}-1} x(2k) e^{\frac{-j2\pi kn}{(\frac{N}{2})}} + W_N^n \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

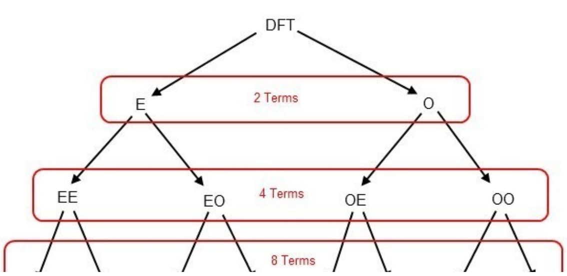
Equation 5

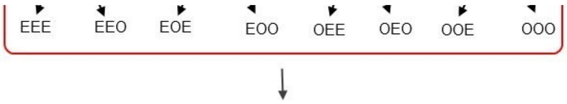
The above is a first level break down. You can continue to break each term down into even and odd terms until you run out of samples and only have one value in the summation. Like this:

$$\sum_{k=0}^{0} = 1$$

Equation 6

This happens because you keep halving the number of values summed on each expansion of the equation. **For the FFT we want all summations to be expanded down to 1 term**. Here is the pattern of expansion for the Danielson-Lanczos Lemma:






This shows the binary nature of the Danielson-Lanczos Lemma. Each term can be broken again and again into even and odd terms until samples are exhausted. This is why an FFT requires a power of base 2 samples, e.g. 2, 4, 8, 16, etc.

As shown in the diagram above, the D-L Lemma breaks down in a binary manner. That is, the number of terms expands as follows 1, 2, 4, 8, 16, 32, etc. In order to get all of the summation to unity, 1, therefore, we must have a power of base 2 number of samples, or N=2^r samples. So, **an FFT requires N = 2^r samples.**

For a sample size of N=2, a first level expansion will be enough to get the summations to unity. The first level expansion will look like this after plugging into *equation 5* above.

$$F(n) = \sum_{k=0}^{0} x(2k) e^{\frac{-j2\pi kn}{(\frac{2}{2})}} + W_2^n \sum_{k=0}^{0} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{2}{2})}}$$

$$= x(0)e^0 + W_2^n x(1) e^0 = x(0) + W_2^n x(1)$$

Equation 7

The summations are reduced to unity, and all that remains is a twiddle factor and input values, x(0) and x(1). This is the general form used for the **Butterfly diagram**, shown later in this tutorial.

Next example will be expansion of the D-L Lemma to 4 terms.

Expansion of the Danielson-Lanczos to Four Terms

<< Previous Next >>

For N = 4 samples, the equation must be expanded again to four terms. Below is the expansion to four terms. As with the first level expansion, substitute 2k into k and reduce the summation by half for the even terms and substitute 2k+1 into k and reduce the summation by half for the odd terms. The E and O below refer to equation 5.

Expand the equation to another level, i.e. to 4 terms starting with E and O from above:

$$E = \sum_{k=0}^{\frac{N}{2}-1} x(2k) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

$$O = W_N^n \sum_{k=0}^{\frac{N}{2}-1} x(2k+1) e^{\frac{-j2\pi kn}{(\frac{N}{2})}}$$

For both E and O, 2k is substituted into k for the even terms, and 2k+1 into k for the odd terms, and the size of the summation is halved.

$$E = EE + EO$$
$$O = OE + OO$$

Here is the even value expanded from E:

$$EE = \sum_{k=0}^{\frac{N}{2(2)}-1} x(2(2k)) e^{\frac{-j2\pi(2k)n}{(\frac{N}{2})}} = \sum_{k=0}^{\frac{N}{4}-1} x(4k)) e^{\frac{-j2\pi kn}{(\frac{N}{4})}}$$

Here is the odd value expanded from E:

$$EO = \sum_{k=0}^{\frac{N}{2(2)}-1} x(2(2k+1)) e^{\frac{-j2\pi(2k+1)n}{(\frac{N}{2})}}$$
$$= \sum_{k=0}^{\frac{N}{4}-1} x(4k+2)) e^{\frac{-j2\pi kn}{(\frac{N}{4})}} e^{\frac{-j2\pi n}{(\frac{N}{2})}}$$
$$= W_{\frac{N}{2}}^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+2)) e^{\frac{-j2\pi kn}{(\frac{N}{4})}}$$

Here is the even value expanded form O:

$$OE = W_N^n \sum_{k=0}^{\frac{N}{2(2)}-1} x(2(2k)+1) e^{\frac{-j2\pi(2k)n}{(\frac{N}{2})}}$$
$$= W_N^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+1) e^{\frac{-j2\pi kn}{(\frac{N}{4})}}$$

Here is the odd value expanded from O:

.

$$OO = W_N^n \sum_{k=0}^{\frac{N}{2(2)}-1} x(2(2k+1)+1) e^{\frac{-j2\pi(2k+1)n}{(\frac{N}{2})}}$$
$$OO = W_N^n \sum^{\frac{N}{4}-1} x(4k+3) e^{\frac{-j2\pi kn}{(\frac{N}{4})}} e^{\frac{-j2\pi n}{(\frac{N}{2})}}$$

$$k=0$$

$$OO = W_N^n W_{\frac{N}{2}}^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+3)e^{\frac{-j2\pi kn}{(\frac{N}{4})}}$$

And finally:

Putting it all together:

$$F(n) = E + O = EE + EO + OE + OO$$

$$= \sum_{k=0}^{\frac{N}{4}-1} x(4k))e^{\frac{-j2\pi kn}{(\frac{N}{4})}} + W_{\frac{N}{2}}^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+2))e^{\frac{-j2\pi kn}{(\frac{N}{4})}} +$$

$$W_N^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+1)e^{\frac{-j2\pi kn}{(\frac{N}{4})}} + W_N^n W_{\frac{N}{2}}^n \sum_{k=0}^{\frac{N}{4}-1} x(4k+3)e^{\frac{-j2\pi kn}{(\frac{N}{4})}}$$

Equation 8

Now, N = 4 samples, and using the same procedure as was used for two samples, equation 8 becomes:

For N=4 and K=0

$$F(n) = x(0) + W_2^n x(2) + W_4^n x(1) + W_4^n W_2^n x(3)$$

Equation 9

Once again, as with N=2, the summations have been reduced to unity, and all you have remaining are "twiddle factors" and the input values, x(0), x(1), x(2), and x(3).

The next example will be an 8 term expansion, shown but not worked through.

The Danielson-Lanczos Lemma Expanded to 8 Terms

<< Previous Next >>

Expansion of the Danielson-Lanczos Lemma to 8 terms:

$$F(n) = \sum_{k=0}^{N/8-1} x(8k)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} + W_{\frac{N}{4}}^n \sum_{k=0}^{N/8-1} x(8k+4)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} +$$

$$W_{\frac{N}{2}}^n \sum_{k=0}^{N/8-1} x(8k+2)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} + W_{\frac{N}{2}}^n W_{\frac{N}{4}}^n \sum_{k=0}^{N/8-1} x(8k+6)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} +$$

$$W_N^n \sum_{k=0}^{N/8-1} x(8k+1)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} + W_N^n W_{\frac{N}{4}}^n \sum_{k=0}^{N/8-1} x(8k+5)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} +$$

$$W_N^n W_{\frac{N}{2}}^n \sum_{k=0}^{N/8-1} x(8k+3)e^{\frac{-j2\pi kn}{(\frac{N}{8})}} + W_N^n W_{\frac{N}{2}}^n W_{\frac{N}{4}}^n \sum_{k=0}^{N/8-1} x(8k+7)e^{\frac{-j2\pi kn}{(\frac{N}{8})}}$$

Equation 10

After the expansion above you can plug in for N = 8 samples, and k=0, since all summation would be unity when N=8.

The Danielson-Lanczos for 8 input values

$$F(n) = x(0) + W_2^n x(4) + W_4^n x(2) + W_4^n W_2^n x(6) + W_8^n x(1) +$$
$$W_8^n W_2^n x(5) + W_8^n W_4^n x(3) + W_8^n W_4^n W_2^n x(7)$$

Equation 11

## Danielson-Lanczos Lemma Observations

<< Previous Next >>

Note two things about the equations 7, 9 and 11, repeated below. First, the order of input values, x(n), is "**reverse binary**".For example, left to right the order for the 4 term equation is x(0), x(2), x(1) and x(3). The order for the 8 term equation is x(0), x(4), x(2), x(6), x(1), x(5), x(3), and x(7). This naturally happens when the D-L Lemma is expanded. The **Butterfly Diagram** makes use of this fact. The second thing to note is that the "**twiddled factors**",W, build up with each new expansion, so that you multiply more together. The Butterfly diagram also deals with this by the adding of "stages", which you will see later in this tutorial.

$$F(n) = \sum_{k=0}^{0} x(2k)e^{\frac{-j2\pi kn}{(\frac{2}{2})}} + W_2^n \sum_{k=0}^{0} x(2k+1)e^{\frac{-j2\pi kn}{(\frac{2}{2})}}$$

$$= x(0)e^0 + W_2^n x(1)e^0 = x(0) + W_2^n x(1)$$

Equation 7

For N=4 and K=0

$$F(n) = x(0) + W_2^n x(2) + W_4^n x(1) + W_4^n W_2^n x(3)$$

Equation 9

The Danielson-Lanczos for 8 input values

$$F(n) = x(0)+W_2^n x(4)+W_4^n x(2)+W_4^n W_2^n x(6)+W_8^n x(1)+ W_8^n W_2^n x(5) + W_8^n W_4^n x(3) + W_8^n W_4^n W_2^n x(7)$$

Equation 11

## More on the "Reverse Binary" pattern.

Here are two examples of the "reverse binary" example:

**For 4 inputs:**

Count from 0 to 3 in binary 00, 01, 10, 11. Now, reverse the bits of each number and you get 00, 10, 01, 11. In decimal this is 0, 2, 1, and 3.

So, the values in the D-L equation would be x(0), x(2), x(1), and x(3). This is what you see in equation 9 above.

**For 8 inputs:**

Count from 0 to 7 in binary 000, 001, 010, 011, 100, 101, 110, 111. Now, reverse the bits of each number and you get 000, 100, 010, 110, 001, 101, 011, 111. In decimal this sequence is 0, 4, 2, 6, 1, 5, 3, and 7.

So, the values in the D-L equation for 8 samples would be x(0), x(4), x(2), x(6), x(1), x(5), x(3), and x(7). This is what you see in equation 11 above.

The same pattern holds for all expansions of the D-L Lemma, and is made use of by the Butterfly Diagram.

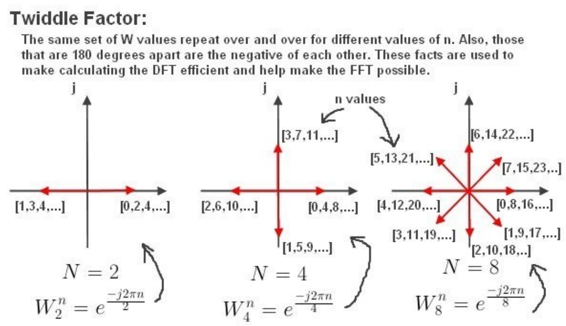
Next I'll discuss the "twiddle factor" and then put it together with the Danielson-Lanczos Lemma to create the Butterfly diagram, which is the FFT in diagram form.

## The "Twiddle Factor"

<< Previous Next >>

The twiddle factor, W, describes a "rotating vector", which rotates in increments according to the number of samples, N. Here are graphs where N = 2, 4 and 8 samples.

c




## The Redundancy and Symmetry of the "Twiddle Factor"

As shown in the diagram above, the twiddle factor has redundancy in values as the vector rotates around. For example W for N=2, is the same for n = 0, 2, 4, 6, etc. And W for N=8 is the same for n = 3, 11, 19, 27, etc.

Also, the symmetry is the fact that values that are 180 degrees out of phase are the negative of each other. So for example, W for N =4 samples, where n = 0,4,8, etc, are the negative of n = 2,6,10, etc.

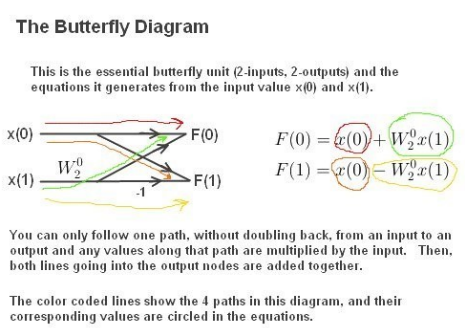
The Butterfly diagram takes advantage of this redundancy and symmetry, which is part of what makes the FFT possible.

## The Butterfly Diagram

<< Previous Next >>

The Butterfly Diagram builds on the Danielson-Lanczos Lemma and the twiddle factor to create an efficient algorithm. The **Butterfly Diagram is the FFT** algorithm represented as a diagram.

First, here is the **simplest butterfly**. It's the basic unit, consisting of just two inputs and two outputs.




That diagram is the fundamental building block of a butterfly. It has two input values, or N=2 samples, x(0) and x(1), and results in two output values F(0) and F(1). The diagram comes form the D-L Lemma for two inputs.

This can be shown by taking *equation 7* above and plugging in for values n=0 and n=1, thus:

The 2 input DL-Lemma:

$$F(n) = x(0) + W_2^n x(1)$$

Plug in n=0 and n = 1:

$$F(0) = x(0) + W_2^0 x(1)$$
$$F(1) = x(0) + W_2^1 x(1)$$

and using the relationship:

$$W_2^1 = -W_2^0$$
$$F(1) = x(0) - W_2^0 x(1)$$

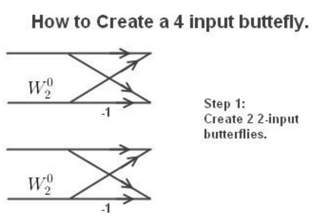
So, the Butterfly comes from the Danielson-Lanczos Lemma, but it also uses the twiddle factor to take advantage of redundancies and symmetry in the D-L Lemma.

To get a full understanding of the Butterfly, a four input Butterfly will be required. That is described next.

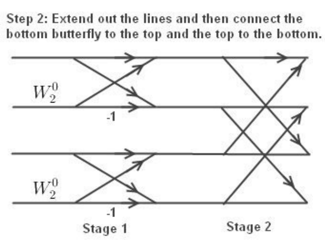
Constructing A 4 Input Butterfly Diagram

<< Previous    Next >>

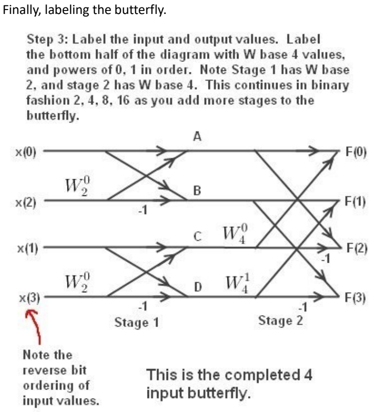
Here I will show you step-by-step how to construct a 4 input Butterfly Diagram.




Next extend lines and connect upper and lower butterflies.




Finally, labeling the butterfly.




Note the order of input values is "reverse bit" order. The Butterfly uses the natural expansion order of the Danielson-Lanczos Lemma, which is why the input is ordered that way. This was described earlier.

The four output equations for the butterfly are derived below.

The equations derived from the 4 input butterfly

Stage 1:

$$A = x(0) + W_2^0 x(2)$$
$$B = x(0) - W_2^0 x(2)$$
$$C = x(1) + W_2^0 x(3)$$
$$D = x(1) - W_2^0 x(3)$$

Stage 2:

$$F(0) = A + W_4^0 C$$
$$F(1) = B + W_4^1 D$$
$$F(2) = A - W_4^0 C$$
$$F(3) = B - W_4^1 D$$

Substituting back in for A, B, C and D:

$$F(0) = x(0) + W_2^0 x(2) + W_4^0(x(1) + W_2^0 x(3))$$
$$F(1) = x(0) - W_2^0 x(2) + W_4^1(x(1) - W_2^0 x(3))$$
$$F(2) = x(0) + W_2^0 x(2) - W_4^0(x(1) + W_2^0 x(3))$$
$$F(3) = x(0) - W_2^0 x(2) - W_4^1(x(1) - W_2^0 x(3))$$

Equation 12

## The N Log N savings

Remember, for a straight DFT you needed N*N multiplies. The N Log N savings comes from the fact that there are two multiplies per Butterfly. In the 4 input diagram above, there are 4 butterflies. so, there are a total of 4*2 = 8 multiplies. 4 Log(4) = 8. **This is how you get the computational savings in the FFT!** The log is base 2, as described earlier. See *equation 1*.

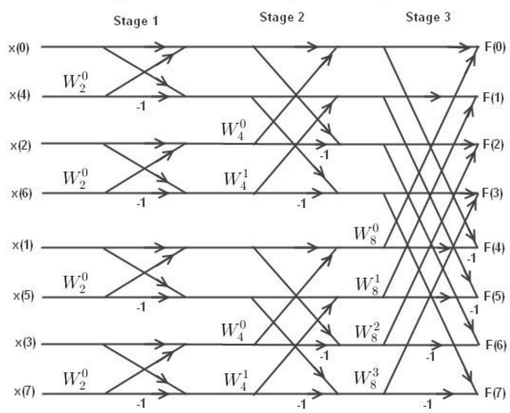
In the next part I provide an 8 input butterfly example for completeness.

An 8 Input Butterfly

<< Previous Next >>

Here is an example of an 8 input butterfly:

An 8 Input Butterfly. Note, you double a 4 input butterfly, extend output lines, then connect the upper and lower butterflies together with diagonal lines.




An The 8 input butterfly diagram has 12 2-input butterflies and thus 12*2 = 24 multiplies.

N Log N = 8 Log (8) = 24. A straight DFT has N*N multiplies, or 8*8 = 64 multiplies. That's a pretty good savings for a small sample. The savings are over 100 times for N = 1024, and this increases as the number of samples increases.

You can keep expanding the butterfly by the same procedure.