

# The Fourier Analysis Tool in Microsoft Excel

Douglas A. Kerr

Issue 1  
March 4, 2009

## ABSTRACT AND INTRODUCTION

The spreadsheet application *Microsoft Excel* includes a tool that will calculate the discrete Fourier transform (DFT) or its inverse for a set of data. Users not familiar with digital signal processing may find it difficult to understand the scales used for the input and output data for this process. In this article we review the concept of the discrete Fourier transform, summarize how the Excel tool is used, and explain the meaning of the scales for the data.

## THE FOURIER TRANSFORM

Imagine that we have a function of time that represents a segment of an audio waveform. If we calculate the Fourier Transform of this source function, we get a new function that describes the *amplitude frequency spectrum* of the waveform. This portrays how the power in the waveform is distributed over frequency.<sup>1</sup> The waveform is said to be described in the *time domain*, while the spectrum is described in the *frequency domain*.

We can have the same concept when the source function occurs “in space” rather than in time, such as a plot of the height variations in a surface as we travel along a line across it, or the variation in the density of a photographic print as we travel along a line across it. In this case, the source function is in the *space domain*, and the Fourier transform is in the *spatial frequency domain*.

Then, if we have the amplitude spectrum of a waveform we can calculate its *inverse Fourier transform*, which will be the waveform itself.

Calculating a Fourier transform (or inverse Fourier transform) is very tedious. It involves the integration of the products of continuous functions—in theory, an infinite number of them. That’s because the transform result is itself a continuous function, meaning that it can

---

<sup>1</sup> Actually, this distribution is really indicated by a *power spectral density function*, or *power spectrum*. The *amplitude spectrum*, with which we deal here, is the square root of this function, and the word amplitude (contrasted with power) is a reminder of this.

have a distinct value for an infinite number of points along its "domain".

But in another context, the job becomes more practical.

## THE DISCRETE FOURIER TRANSFORM

The Nyquist-Shannon sampling theorem tells us that if we have a waveform that contains only components below a certain frequency, then if we record the instantaneous value of the waveform at repetitive instants occurring at a rate of twice that frequency, we will have a complete description of the waveform. This means, for example, that from this suite of data alone we could precisely reconstruct the entire waveform, or conduct any possible kind of analysis of the waveform itself. This suite of data is said to be a "sampled" representation of the waveform.

If we have this suite of sampled data for a waveform segment, we can readily calculate a sampled description of the Fourier transform of the waveform itself. This is said to be the *discrete Fourier transform* (DFT) of the waveform. "Discrete" implies that both the source waveform and the resulting function are described with "discrete" values (samples)

Just as the suite of sampled data is a complete, precise description of the waveform itself, so is the suite of sampled data of the DFT a complete, precise description of the actual Fourier transform of the waveform.

And there is the *inverse discrete Fourier transform* (IDFT), which will take the sampled description of, for example, the amplitude frequency spectrum of a waveform and give us the sampled representation of the waveform itself.

The discrete Fourier transform is often, incorrectly, called the *fast Fourier transform* (FFT). This is not a particular kind of transform. Rather, it is a highly-efficient procedure for calculating the discrete Fourier transform. Especially during the earlier days of computing, when computational resources were at a premium, the only practical way to compute a DFT was by way of the FFT procedure. Thus people came to think that "FFT" was the name of the transform it practiced and the result it produced.

If in fact we use the FFT procedure to compute the DFT of a function (and there are other ways to do it), it is perfectly accurate to say, "we ran the FFT on the input data", but in fact the result is the DFT of the input function.

### THE EXCEL FOURIER ANALYSIS TOOL

The spreadsheet application *Microsoft Excel* will take a suite of data and calculate its discrete Fourier transform (DFT) (or the inverse discrete Fourier transfer). It uses the FFT procedure to compute the DFT.

#### The two data suites

The suite of input data must have a size that is an integral power of 2 (such as 16 points or 1024 points). This is a requirement of the FFT procedure used to calculate the DFT.

The suite of output data will have the same size as the suite of input data.

#### Operation

To do this operation, we open the Tools>Data Analysis>Fourier Analysis dialog (Data Analysis functionality must be installed in Excel), indicate whether it is the forward or inverse transform we want to perform, and identify the suite of source data as a range. We then indicate where we want the output data deposited. We are apparently invited to specify a range for that, but the output data will always be deposited into an area the same size as the input data area (and running in the same “direction”), so only the starting cell we cite makes any difference.

Note that the menu does not identify this as **discrete** Fourier analysis.

#### The scales of the data

But now there comes the matter of understanding the frequency and time scales of the two sets of data points.

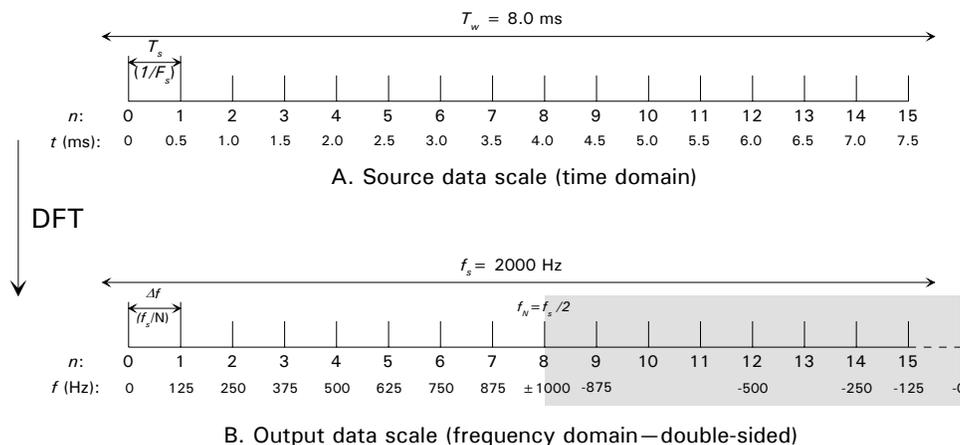


Figure 1. Data scales with the DFT

Let's imagine for the moment that we have a set of 16 values describing 8 ms of waveform, and we wish to take its DFT. The 16 values are entered into a range on an Excel spreadsheet.

Figure 1 will help us to understand this. We must first be clear on the time scale for the source data (describing the waveform), which is a creature of the time domain. Panel A shows the scale applicable to the source data.

We execute the Excel DFT tool and receive a set of 16 output values. We know that these represent, in sampled form, a "curve" that is the amplitude frequency spectrum of the source waveform. This spectrum is a creature of the frequency domain. But what is the frequency scale for it?

We see the 16 points (numbered 0-15) assigned to our 8-ms waveform segment. We designate the number of points as  $N$ . The time interval between the points is 0.5 ms; we call this the sampling interval,  $T_s$ . We also see that these 16 points embrace our total of 8 ms, a period that we call the overall sampling window time,  $T_w$ . The sampling frequency here,  $T_s$ , is 2000 Hz:  $1/(0.5 \text{ ms})$ .

Note the peculiar way I draw the 8-ms span. Since we have 16 points, the distance between the first and last point is 15 interpoint times (only 7.5 ms altogether). So we have to think of each point in the center of a time slice 0.5 ms long, and our whole sampling window embraces 16 of these.

Now in panel B we see the output data points. This will be on a scale of frequency. But what scale?

It turns out that the entire span of the frequency scale<sup>2</sup> always corresponds to the sampling frequency,  $f_s$ . Thus, if we have  $N$  points, the distance between points corresponds to a frequency increment of  $f_s/N$  (in this case, 125 Hz:  $2000/16$ ).

Now we are ready to deal with another peculiarity.

### **The double-sided frequency scale**

Firstly, note that if our signal meets the Nyquist criterion for eligibility to be represented by samples taken at a rate of  $T_s$ , it will have no frequencies at or above the "Nyquist frequency",  $T_N$ , which is  $T_s/2$  (1000 Hz in our example). Thus we will have no need for results at

---

<sup>2</sup> Again, we have to draw it with the little peculiarity I just mentioned.

points 8-15 in our chart (the area shaded in gray), and it might seem that this portion of the output is redundant.

But, in formal mathematical work, any signal we think of as having frequency  $f$  must be thought of as comprising two components, one with frequency  $f$  and one with frequency  $-f$ . Thus, in such work, we often have a “double-sided” frequency scale, extending both above and below zero.

We have that same situation here, but with a wrinkle. The “negative frequency” part of the scale, rather than being to the left of zero, has been scooted to the right, just beyond the “positive frequency” part of the scale.

And as we might expect (if we ignore a little subtlety we’ll get to a little later), the result data in that portion is just a mirror image of that in the “positive frequency” part of the scale. Thus (so far as we can see at this point), the value at point 14 is the same as the value at point 2, and the value at point 10 is the same as the value at point 6.

You may wonder why the program even outputs that part of the list of results. We’ll see the answer in the next section.

### **Result values as complex numbers**

Another peculiarity we may notice is that the output values will sometimes have this appearance:  $0.00021 + 0.00013i$ . What’s with that?

Each of the output values describes a component of the original waveform at a certain frequency. This can be thought of as a “sine wave” of that frequency with a certain amplitude **and a certain time phase**.

There are several ways that can be represented. First, note that although we call these recurrent functions “sine waves”, in mathematical work, we ordinarily represent them with a **cosine** function. This has the same shape as a sine wave, but has its time reference in a different place. The reason the cosine representation is used is just that it makes some of the math more tidy (in particular the matter of the plus or minus signs of certain quantities we encounter in our calculations).

Accordingly, we could describe a component of the waveform at a certain frequency by stating the amplitude of a cosine function at that frequency plus its time phase (stated as an angle).

Alternatively, we can consider the component to be composed of the combination of a cosine function and a sine function with individual amplitudes.

It turns out that this latter form can be stated mathematically by giving the amplitude of a cosine wave as a complex number; that is, a number which is the sum of a regular “real” number plus an “imaginary” number, which is a multiple of the square root of -1 (which we represent by the symbol  $i$ ). (Don’t try to visualize this!) The real part is the amplitude of the **cosine** sub-component; the imaginary part is the amplitude of the **sine** sub-component.

In mathematics, we write it this way:  $25.2+67.3i$ . 25.2 is the amplitude of the cosine sub-component, and 67.3 is the amplitude of the sine sub-component.<sup>3</sup>

Now, in many cases, we don’t care about the time phase of the components. We just want the “absolute amplitude frequency spectrum”, which would consist of the *magnitudes* of the component amplitudes, with the phase angles just not mentioned. The values returned by the Excel DFT tool can easily be converted to that form with this function:

=IMABS(<output data point>)

### Back to the “redundant” part of the output table

Now that we’re aware of the phase angle implications of the components, we must note that when we think of a component as comprising two subcomponents, one at frequency  $f$  and one at frequency  $-f$ , although they have the same physical phase angle (they are after all just two manifestations of the same component), in the “complex” representation, the sign of the imaginary part of the amplitude is reversed in the “negative frequency” aspect.

Accordingly, the result of the DFT for point 10 is not quite identical to that for point 6—the imaginary part of the amplitude has a reversed sign. Thus it is really required that the seemingly-redundant portion of the output table be delivered.

---

<sup>3</sup> In electrical engineering, a different convention is used. There, the “imaginary unit” is represented by  $j$ , not  $i$ , to avoid confusion with the use of the symbol  $i$  for current. And this format is used:  $25.2+j67.3$ . It is not as “algebraically rigorous” as  $25.2+67.3j$ , but is perhaps easier to recognize when we encounter it.

### **"Over-Nyquist" components**

Suppose in fact that our input waveform did contain a component at, say, 1175 Hz (beyond the Nyquist limit for the sampling rate we are using). The amplitude of that component would show up in our result at point 9. But we have at point 9 the amplitude of the component at -875 Hz (the negative frequency alter ego of the component at 875 Hz). So in fact the numerical value we find at point 9 of the output data suite is the sum of the amplitude for the component at 1175 Hz and the "reversed phase" amplitude of the component at 875 Hz.

Clearly this means that some data has been "scrambled", and in fact, this frequency response does not unambiguously represent our input waveform.

This is an example of "aliasing", an anomaly that occurs in the sampled representation of a waveform with a sampling rate that is not greater than twice the frequency of the highest-frequency component of the waveform.

### **THE INVERSE TRANSFORM**

Next we will look at the application of the inverse discrete Fourier transform, again with emphasis on understanding the scales of the two data suites.

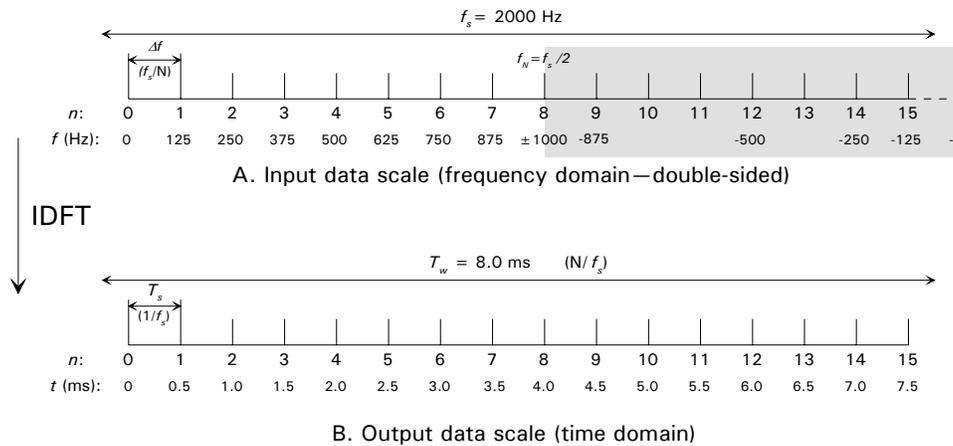
There are two classical situations in which we may wish to employ the inverse discrete Fourier transform:

- We have the amplitude spectrum of a waveform and would like to determine the waveform itself.
- We have the amplitude frequency response of a system, and would like to know its impulse response: the waveform that will come out of the system when we send into it an "impulse", which is a single pulse with zero time width (but finite energy)..

#### **An example of the first case**

We will first use an example that can be visualized in terms of the first case. For ease of comparison, we will choose numbers such that the two scales are the same as in our earlier example. Figure 2 shows this situation.

Assume that we have data describing the amplitude frequency spectrum of an unknown waveform. The data points are spaced at every 125 Hz along the frequency axis (starting with 0), and the waveform has no non-zero value at 1000 Hz or above.



**Figure 2. Data scales with the inverse DFT**

Thus we can assume that the Nyquist frequency of the underlying waveform is 1000 Hz. Accordingly, since the Nyquist frequency falls at the center of our frequency domain scale (point  $N/2$ ), we will use a scale of 16 points. Since the distance between points is 125 Hz, the entire “span” of the frequency domain scale is 2000 Hz (with the now-familiar “wrinkle”). As we saw before, that must be the sampling frequency of the original signal (in the time domain),  $f_s$ . (When we get the waveform itself, it will be described in terms of samples at that rate.)

The amplitude values are all “real” (not complex); that is, they all imply zero phase angle.

This time, when we enter the input data into our spreadsheet, we must do some special housekeeping. We saw before that the result of the Fourier transform will always have each actual component at two places in the scale, one for  $+f$  and one for  $-f$ . So we must enter each value (except for the “zero-frequency” value, which goes only into point 0) into two places: points 1 and 15, 2 and 14, and so forth.<sup>4</sup> (We can in fact fix up our spreadsheet to do this for us so we really only have to enter the “positive frequency” values.)

Now back to the matter of our scale. We have seen that in this case it is appropriate to use a 16-point representation of the input spectrum, with a total span of 2000 Hz (twice our assumed Nyquist frequency).

---

<sup>4</sup> If in fact our data points were “complex” (implying a phase angle), then we would place the “complex conjugate” of the value in the “negative frequency” position. That is, if point 3 had the value  $0.06+0.07i$ , we would put into point 13 the value  $0.06-0.07i$ .

This total span is in fact the sampling frequency of the actual waveform ( $f_s$ ).

Thus, on the output data scale (in the time domain), the increment between points is just the sampling interval, which is  $1/f_s$ . In our example, where we have determined that the sampling rate for the waveform must be 2000 Hz, this increment between points must represent 0.5 ms.

With that knowledge, we can interpret the result of the IDFT as a waveform on a specific time scale. But there is a wrinkle, which we'll actually explore in our discussion of the second case.

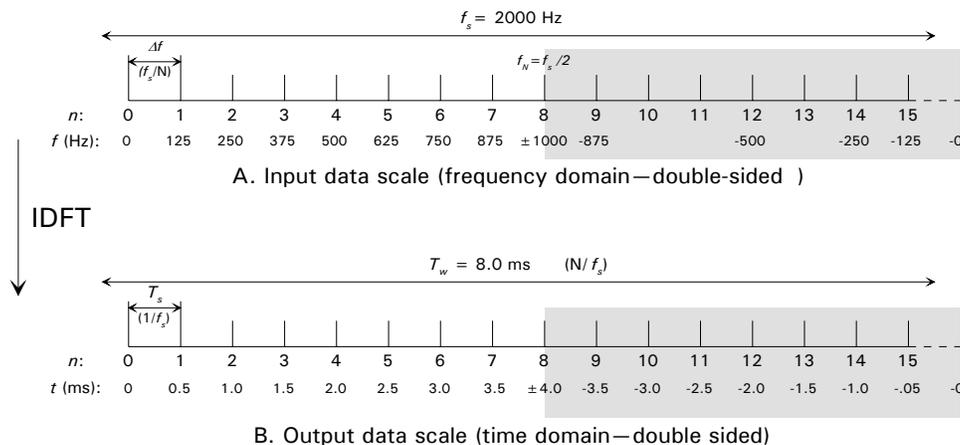
**An example of the second case**

Now, let's use an example in which we have the frequency response of a system and wish to determine the impulse response: the waveform that will come out of the system when we input a zero-width pulse (an *impulse*).

Here again the inverse discrete Fourier transform will do the trick, but there is a wrinkle.

Although of course this could never happen in real life, when we consider certain system frequency responses that can only exist on paper, the output waveform must begin before the time of the input pulse (such a system is described as *prophetic*).

Thus, our output scale must extend both below and above zero. In Excel, the time scale is "folded" to become double-sided, just as we saw for the frequency scale. We see this interpretation of the time-domain scale in figure 3B.



**Figure 3. Double-sided time domain scale**

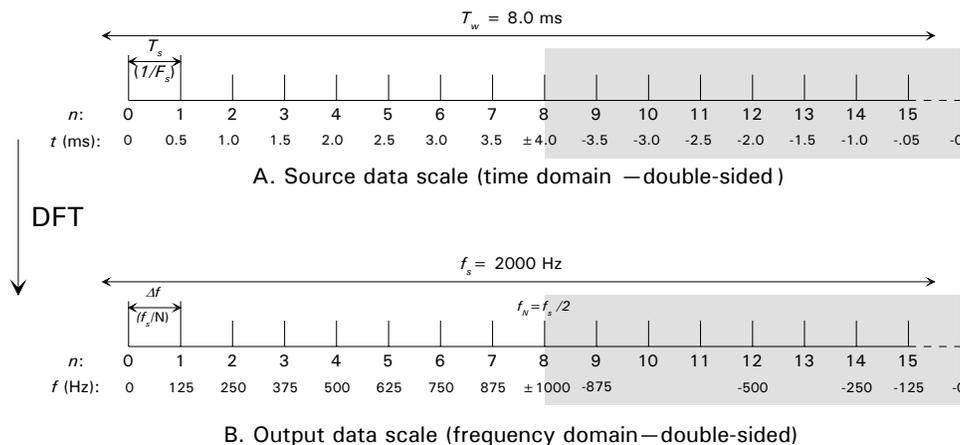
Here, a value presented at point 11 represents the value of our waveform at time -2.5 ms (2.5 ms before we introduced the input pulse). Again, this could not happen in reality, but it is a characteristic of various convenient fictional systems.

Now, how would the value of the waveform at +5.5 ms (5.5 ms after the input pulse) be shown?

Well, the range of our output does not extend that far—it would not include a point at +5.5 ms.

Now, let’s go back to the (forward) DFT. We suggested (in figure 1) that the time scale there was single-sided. But in figure 3, for the IDFT, we saw that the time scale is actually double sided. Figure 4 shows that same outlook fin the case of the DFT.

In fact, we can look at the source data time scale either way in DFT operation. The reason is that, owing to the “cyclic” nature of these scales, in a context where we only observe the waveform over a finite window, point 12 could represent the value of our waveform at  $t = +6.0$  sec (as suggested on figure 1A), or it could represent a point on our waveform at  $t = -2.0$  sec (as suggested on figure 4A). Either would have the identical effect on the DFT output (the amplitude frequency spectrum of the waveform).



**Figure 4. DFT—both scales shown double-sided**

**SUMMARY OF SCALE FACTORS**

The chart in Appendix A shows, for any known scale factor in a “DFT setup”, all the other scale factors.

### TWO PI IN THE SKY

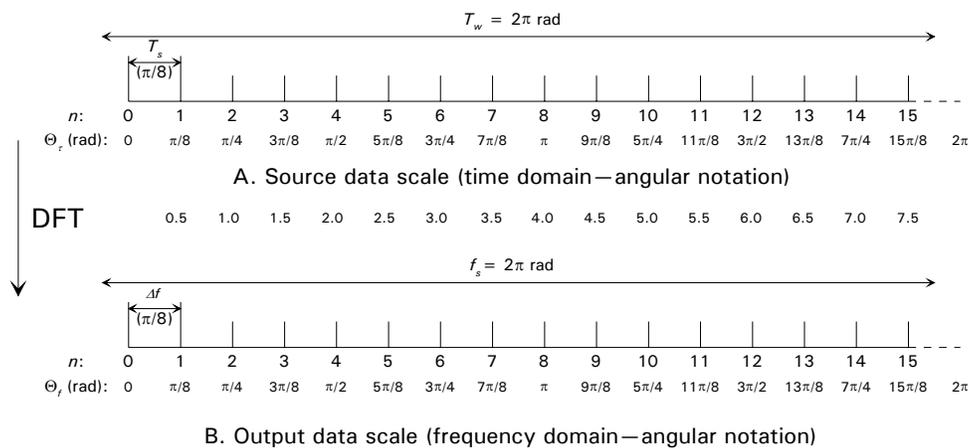
There is another scheme of reckoning our two scales that is often encountered in engineering work with the Fourier transform. It is based on the concept that both our scales are actually cyclic.

By that we mean that, in our example with an 8-ms long time scale, a value at point 3 can not only represent a value of the waveform at  $t = 1.5$  ms but also at  $t = 9.5$  ms, 17.5 ms, 25.5 ms, and so forth.

Thus, we can imagine our scale as not being a line segment but rather a circle. Now, without concern for the actual numerical scale involved, we can speak of locations along the scale in terms of angle “around the circle”. Of course, since this is a mathematical concept, we express the angle in radians. One complete cycle around the circle (the span of our scale, as seen in the ordinary way) corresponds to an angle of  $2\pi$  radians.

Having done this, we can define the DFT itself, or describe the DFT of certain common waveform shapes, in terms of angle  $\theta_t$  (lower-case Greek *theta*) on the time axis and angle  $\theta_f$  on the frequency axis, in a completely general way, independent of the actual time (or frequency) scale involved.

Figure 5 shows our two scales labeled this way. Note that, now we think of them as cyclic, there is no longer any reason to uniquely label the portion that represented “negative” values. An instant at “negative time” can just be looked at as an instant “at positive time the previous time around”.



**Figure 5. Scales in angular notation**

#

**APPENDIX A**  
**Scale Factors**

This chart shows, for any known scale factor in a system of a time scale and frequency scale related by the discrete Fourier transform (or its inverse), all the other scale factors.

In each case, the number of points in the source and result data suites is denoted  $N$ .

Use the row labeled at its left by the scale factor that is known. All the corresponding scale factors for the time-domain and frequency-domain scales are then listed.

The shaded cell is the one whose value is known.

	Time domain scale			Frequency domain scale	
What is known	Total time span $T_W$	Sampling interval $T_s$	Sampling frequency $f_s$	Scale width $f_w$	Frequency increment $\Delta f$
<b>Time scale</b> Total span $T_W$	$T_W$	$\frac{T_W}{N}$	$\frac{N}{T_W}$	$\frac{N}{T_W}$	$\frac{1}{T_W}$
<b>Time scale</b> Sampling interval $T_s$	$NT_s$	$T_s$	$\frac{1}{T_s}$	$\frac{1}{T_s}$	$\frac{1}{NT_s}$
<b>Time scale</b> Sampling frequency $f_s$	$\frac{N}{f_s}$	$\frac{1}{f_s}$	$f_s$	$f_s$	$\frac{f_s}{N}$
<b>Frequency scale</b> Total width $f_w$	$\frac{N}{f_w}$	$\frac{1}{f_w}$	$f_w$	$f_w$	$\frac{f_w}{N}$
<b>Frequency scale</b> Increment $\Delta f$	$\frac{1}{\Delta f}$	$\frac{1}{N\Delta f}$	$N\Delta f$	$N\Delta f$	$\Delta f$

Note that in every case, when using “angular” notation, the total time span and total width of the frequency scale are both  $2\pi$ .