An Extensive Guide to the

Discrete Fourier Transform

researched and written by Francis Miguel J. Sanguyo

supplementary material for Linear Algebra

written from July 18, 2021 to October 2021

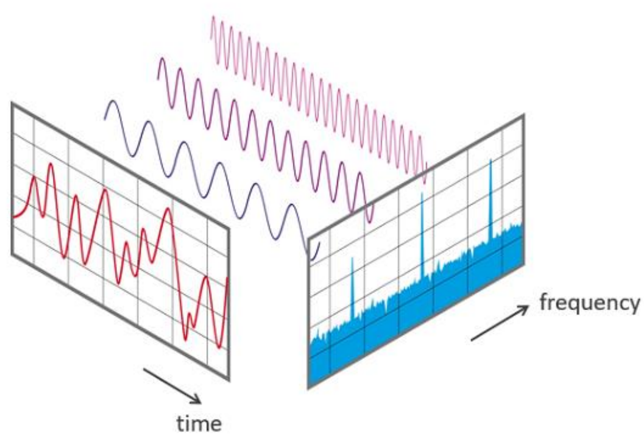
Table of Contents

Section 1 - Introduction to the Fourier Transform

In applied mathematics, it is common to first transform data values or mathematical expressions into a more suitable form for solving certain kinds of problems more simply. For instance, measured values that seem to follow an exponential trend may be transformed by taking logarithms so that the values follow a linear trend, allowing them to be tested for correlation or to be estimated by finding the equation for the regression line of the transformed values, which in turn estimates the original data values. In linear algebra, expressing vectors in terms of special bases such as orthonormal bases and eigenbases for the corresponding vector space greatly simplifies calculations because these bases have certain provable properties, such as orthogonality or diagonalizability, that imply simply-written mathematical relationships. In integral calculus, techniques such as trigonometric substitution and u-substitution allow the exact expressions for the indefinite integrals of many common functions to be derived.

$$\begin{tikzcd}
\text{(problem)} \arrow[r, "\text{Fourier transform}"] \arrow[d, dotted, "\text{solve}"', "\text{(hard)}"] & \text{(transformed problem)} \arrow[d, "\text{solve}"', "\text{(easy)}"] \\
\text{(solution)} & \text{(transformed solution)} \arrow[l, "\text{inverse Fourier transform}"']
\end{tikzcd}$$

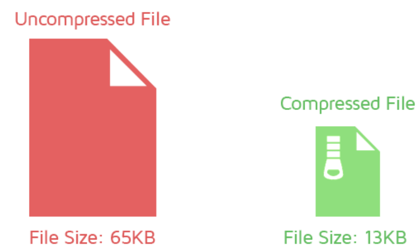
This technique of looking at the input object from another point-of-view obtained by applying a certain transformation in to solving a certain problem is ubiquitous in mathematics, and in fact, goes beyond just statistical analysis and linear algebra. While matrix transformations are appropriate for transforming vectors in R^n to other vectors in R^n, there are also ***function transforms*** that are applicable to the set of functions $F(a, b)$ defined on an arbitrary interval $[a, b]$. One of the most well-known of these function transformations is the ***Fourier Transform*** (FT), named after the French mathematician *Joseph Fourier*, which views a function $y = Y(x)$ in x, usually representing a physical measurement or signal, as a combination of sinusoidal waves or more generally "rotating" waves in the complex plane, each with differing frequencies f.



The Fourier Transform of a function $Y(x)$, represented in red in the above diagram, is mathematically denoted by $U(f) = \mathcal{F}\{Y(x)\}$ and all values of $U(f)$ are expressed as complex numbers in the form $a + bi$, where $i = \sqrt{-1}$. Unlike $Y(x)$, $U(f)$ is instead a function of the wave frequency f that may compose $Y(x)$, in a sense that the Fourier Transform looks from the perspective of the sine waves or rotating waves making up the graph of $Y(x)$, rather than the variable x itself. The magnitude and phase of each value of $U(f)$ represents how much does a wave of a certain frequency f and phase contribute to the function $Y(x)$. If such frequency is prevalent in $Y(x)$, then the magnitude of $U(f)$ is large and the graph of U shows a peak at that frequency f. In this way, the Fourier Transform can be viewed as a “detector” for seeing whether certain frequencies make up the graph of $Y(x)$. More generally, the Fourier Transform $U(f)$ describes the distribution of frequencies composing a function $Y(x)$.

We can also express the original function as $Y(x) = \mathcal{F}^{-1}\{U(f)\} = \mathcal{F}^{-1}\{\mathcal{F}\{Y(x)\}\}$ by applying what is called the ***Inverse Fourier Transform*** (IFT), which “reconstructs” $Y(x)$ from its component frequencies whose distribution is dictated by $U(f)$. When we refer to the functions $Y(x)$ and $U(f)$ in terms of a measurement variable and its frequency, respectively, we call the domain of $Y(x)$ as the ***time domain***, although x need not represent time itself, and the domain of $U(f)$ as the ***frequency domain***. Together, the Fourier Transform and Inverse Fourier Transform allow us to translate between the time domain and frequency domain pertaining to a mathematical value describing the object.

To some, it may seem odd to interpret a function as a combination of waves as discussed above, however, this simple act of viewing its function values in terms of its component frequencies actually paves the way for many applications in information technology, the arts, the natural sciences, and mathematics itself, be it for deciphering the chemical composition of a sample, quickly compressing the images and videos we watch on our screens each day, analyzing the musical qualities of a sound sample either by observation or machine learning, generating visual spectra we often see in music videos, approximating certain kinds of functions for the ease of mathematical analysis, or for analytically solving differential equations describing more complex systems.



Although function transforms involving the operation of integration are often expensive to compute directly, the Fourier Transform is special in the fact that there are certain techniques to compute it much more efficiently, allowing larger sets of data to be handled in a shorter time. This method is called the ***Fast Fourier Transform*** (FFT), which computes the Fourier transform values of a finite set of data values / ordinates, otherwise referred to as a ***Discrete Fourier Transform*** (DFT), corresponding to a given numerical range for a variable such as time or length. The major breakthrough the FFT achieves is that it computes DFT values in an algorithmic complexity of $O(n \log_2 n)$, which is pronounced as *“O of n log base 2 of n”*. This means that doubling the number of data values by k times multiplies the number of steps to compute the output value by a factor of $2^k \log_2(2^k)$ or $k(2^k)$ times. In comparison, directly computing the DFT has a complexity of $O(n^2)$, meaning that doubling the number of data values requires $(2^k)^2 = (2^k)(2^k)$ times more steps to compute the FT values. The difference in the times by which these algorithms compute these values becomes stark in much larger data sets. For instance, while it may take a few minutes for the FFT to process millions of input values, direct computation can take several hours just to compute the same result. The algorithm for translating from the frequency domain back to the time domain, or in other words, for reconstructing the original function from its component frequencies, has a very similar implementation to the FFT. The method for this computation is called the ***Inverse Fast Fourier Transform*** (IFFT), and the finite set of output values is referred to as the ***Inverse Discrete Fourier Transform*** (IDFT).

Due to the computational advantages of the FFT and IFFT, it and algorithms built upon it are not uncommon in the realm of digital technology, and it may be applied in a variety of techniques such as noise filtering, the encoding and decoding of images and audio files, and modulation of signals in modern communication devices. It has been described by the American mathematician Gilbert Strang as *“the most important numerical algorithm of our lifetime”*.

Section 2 - Center-of-Mass Interpretation of the Fourier Transform

Reference: But what is the Fourier Transform? A visual introduction. - YouTube

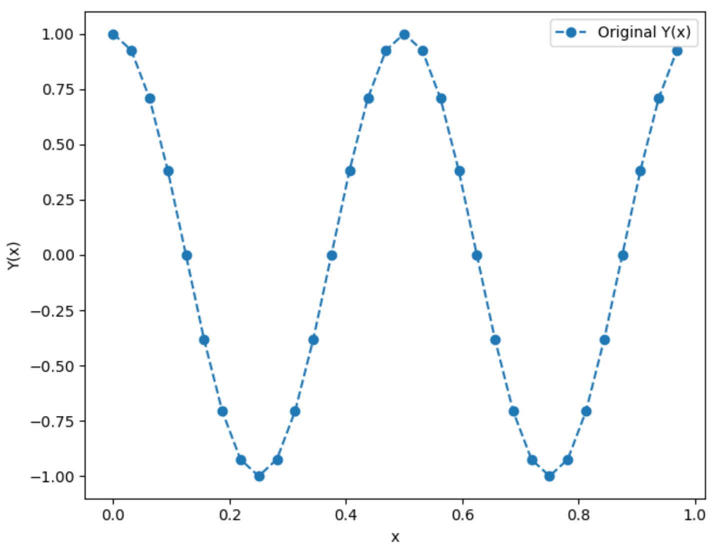
But what do we mathematically mean by a "Fourier Transform", and if the Fourier Transform is the function between frequency and its complex value, how is this relationship shaped exactly by a function in the time domain?

Say that we have a real function $y = Y(x)$, say $Y(x) = \cos(4\pi x)$ in the time domain, where x is a real variable associated with a measurement. This often represents the behavior of a physical phenomenon, such as voltage, sound pressure, image color, light intensity, or displacement, usually through either time, locations, or through some measurement of distance. We select the closed interval $[0, 1]$ as the time domain wherein $Y(x)$ is defined for all points $x \in [0, 1]$ from which we will translate into the frequency domain. From this domain, we select $n = 32$ values of x, which are:

$$x_0 = 0, \quad x_1 = \frac{1}{32}, \quad x_2 = \frac{2}{32}, \quad x_3 = \frac{3}{32}, \ldots, \quad x_{31} = \frac{31}{32}$$

Later, we will allow the case where the time domain is not $[0, 1]$.

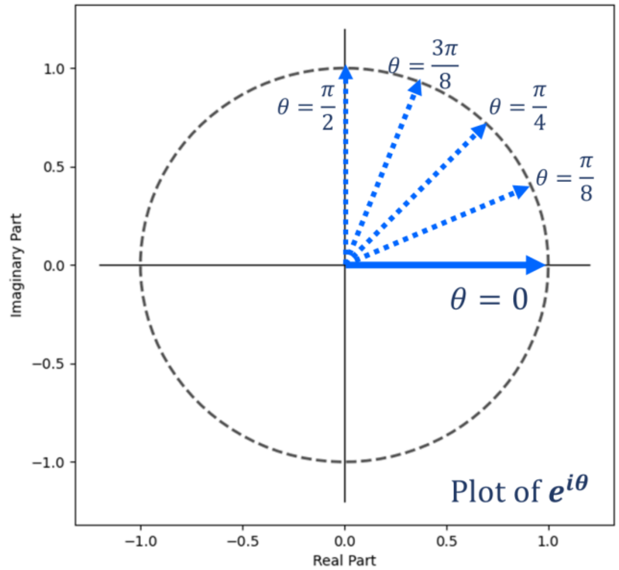
Below is the plot of the graph of $Y(x)$.



Now recall that a complex exponential $e^{i\theta}$, where $i = \sqrt{-1}$, has a value given by Euler's Formula:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

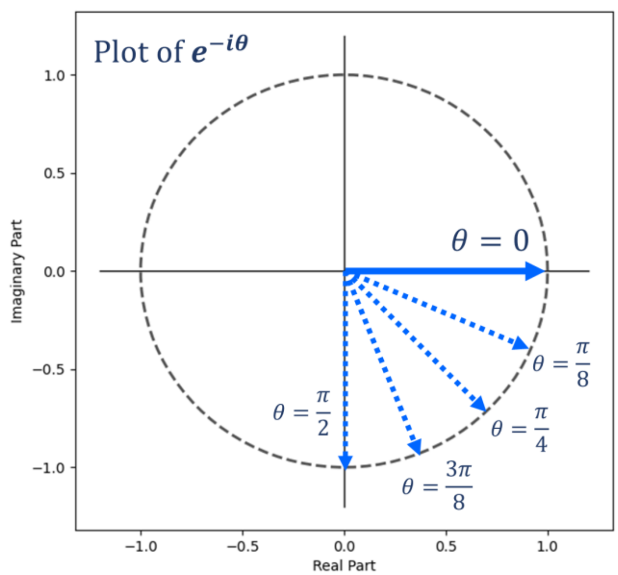
This expression can be associated with the complex plane as a radius vector in R^2 starting from the origin to the point $(\cos\theta, \sin\theta)$. Therefore, as we increase θ, the point the vector makes goes around the origin the counter-clockwise direction and traces out a unit circle once θ hits 2π, and increasing it further causes it to revolve around the origin repeatedly. Since one full revolution is simply 2π radians, we can say that θ in $e^{i\theta}$ represents the angle from the positive real axis to the radius vector in the counter-clockwise direction.



If we instead make the exponent negative, then Euler's Formula becomes:

$$e^{-i\theta} = \cos\theta - i\sin\theta$$

Thus, the vector now has its tip at $(\cos\theta, -\sin\theta)$ and θ in $e^{-i\theta}$ represents the angle from the positive real axis to the radius vector in the *clockwise* direction. Increasing θ results in the radius vector rotating in the clockwise direction. For deriving the Fourier Transform and Discrete Fourier Transform, we shall use the convention of negative complex exponents instead of positive exponents.

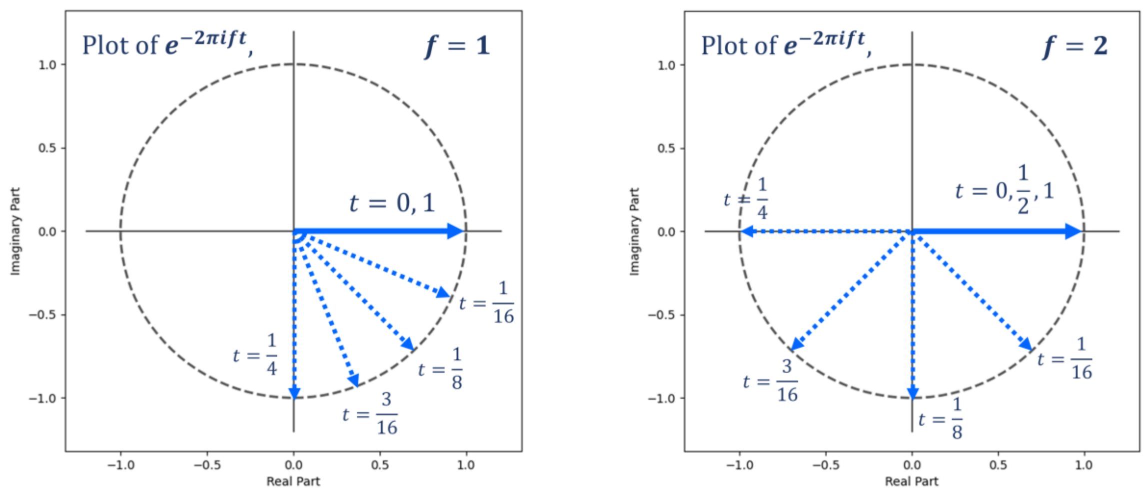


If we say that θ is directly proportional to the time t of rotation, we can associate the rotation of the radius vector with each of the component waves making up $Y(x)$, with higher frequencies corresponding to a faster rotation and negative frequencies resulting in a counter-clockwise rotation.

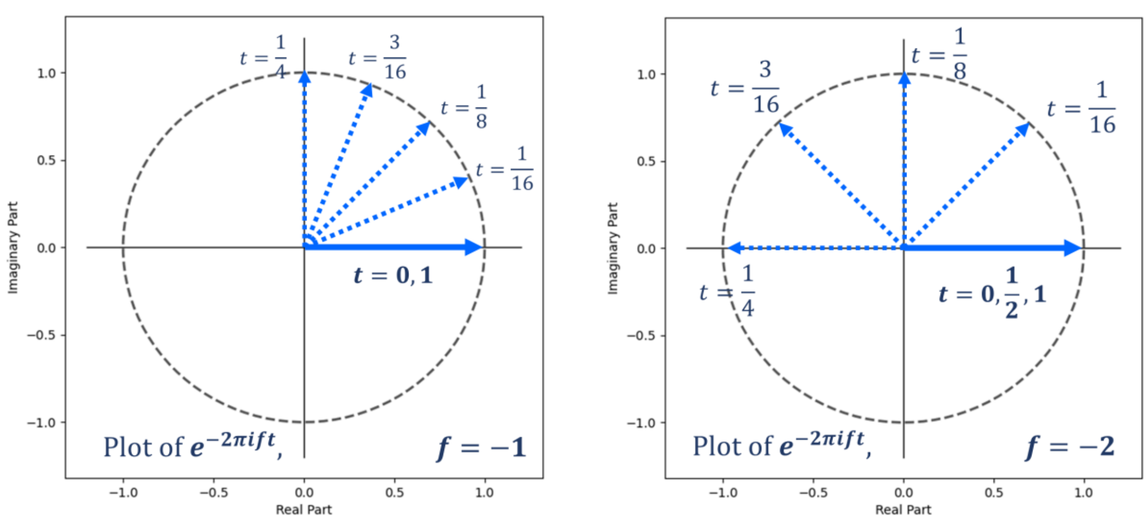
For this reason, we wish to we express $e^{-i\theta}$ in terms of the frequency f and the time parameter t of a component wave. Recall that frequency $f \in (-\infty, \infty)$ is defined as the reciprocal of the period $t = T$, that is, $f = 1/T$. If we make one full revolution, then $\theta = 2\pi$ or $\theta = 2\pi * 1$, but since this also makes one period for t, we must have $t = T$ and therefore $f = 1/t$ or $ft = 1$, meaning that we can write:

$$\theta = 2\pi f t$$

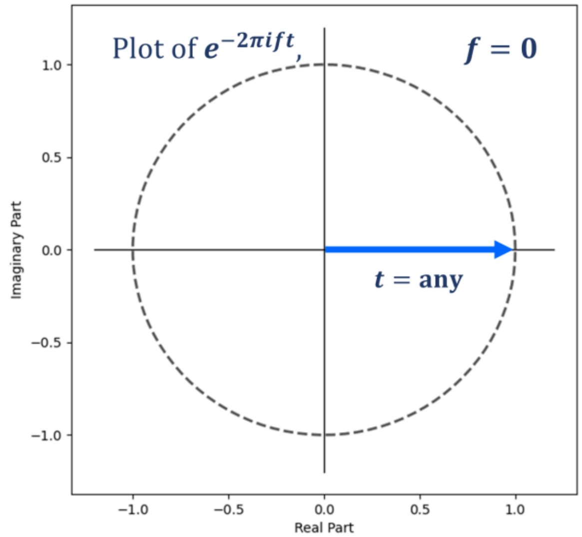
Thus, if we wish to trace out the unit circle in the complex plane with a frequency of f (of which is in Hertz if t is in Seconds) as we vary t, then we simply plot the expression $e^{-2\pi i f t}$ on the complex plane at all values of t in an interval we choose. If $f > 0$, then as we increase t, the resulting vector clockwise about the origin:



If $f < 0$, then the vector rotates in the counter-clockwise direction:



If $f = 0$, then $e^{-2\pi i f t} = 1$ and the tip of the vector is stuck on the point $(1, 0)$ regardless the value of t:

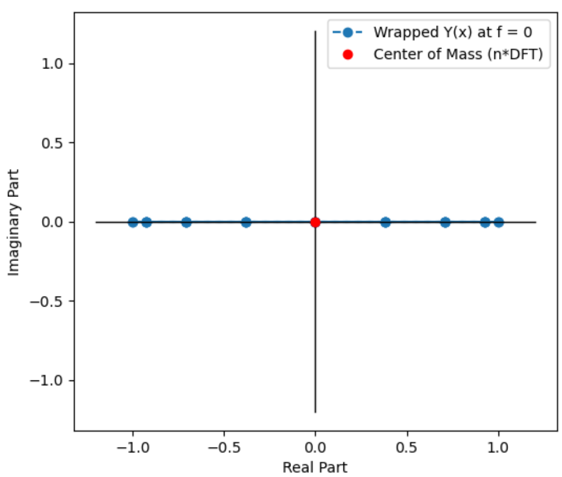


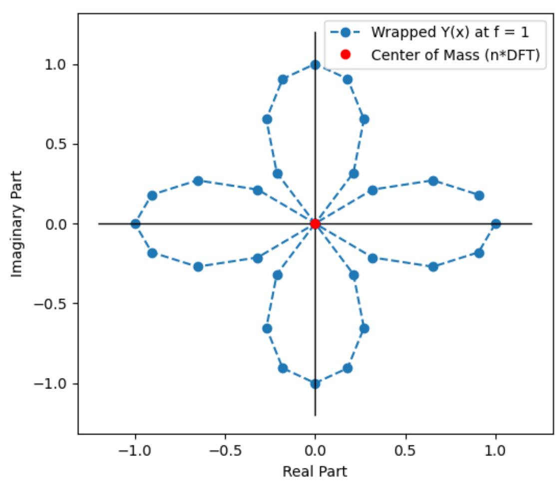
Having known how complex exponentials are related to rotation on the complex plane, we now "wrap" the graph of our function $Y(x)$ in the time domain into this unit circle, in such a way that the $n = 32$ points on the graph of $Y(x)$ get translated to points on the complex plane at different angles with respect to the positive real axis. We will set $t = x$, since we wish to distribute the points of $Y(x)$ radially as the radius vector rotates in the complex plane. We will call this "wrapped" function $R(x, f)$:

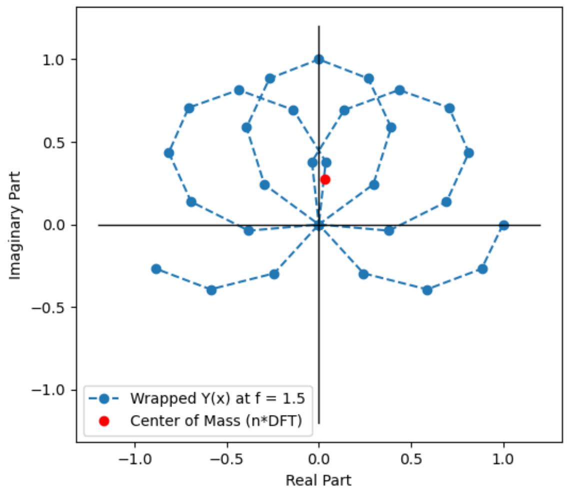
$$R(x, f) = Y(x)\, e^{-2\pi i f x}$$

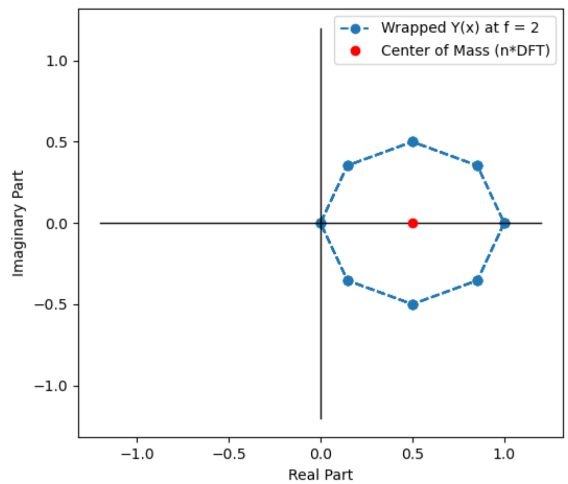
Now we select a value of the frequency f by which we wrap the function and keep it still. As we vary x from $x = 0$ to $x = 1$ the point representing the value of $R(x, f)$ traces out a curved figure that goes around the origin. The radius vector to each point has length $|Y(x)|$ and an angle of either $2\pi f x$ radians if $Y(x) > 0$ or $-2\pi f x$ radians if $Y(x) < 0$ from the positive real axis in the clockwise direction. In other words, negative function values flip the radius vector about the origin.

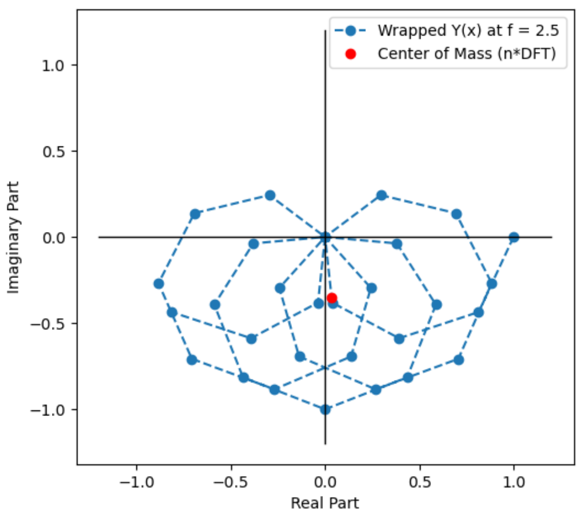
Below are the plots of $R(x, f)$ at $f = 0,\ 1,\ 1.5,\ 2,\ 2.5,$ and 3 in the complex plane:

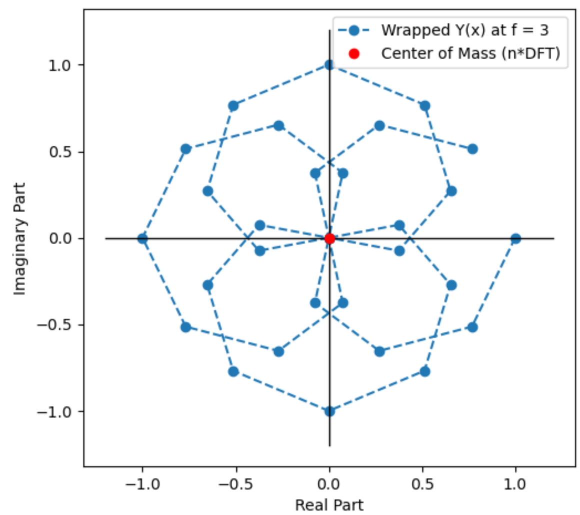








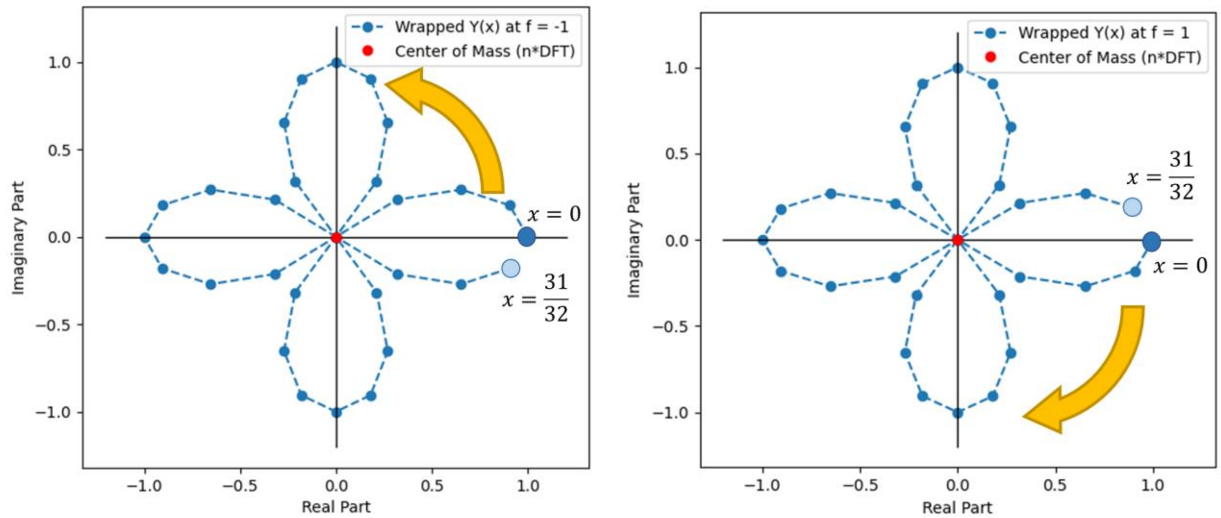




Notice that every time we set f to a new value, the graph of $R(x, f)$ traces out a new shape around the origin. Since $x \in [0, 1]$, a frequency of 1 unit causes the graph of $R(x, f)$ to complete one full revolution in the clockwise direction as x varies from 0 to 1. This is because $-2\pi i f x = -2\pi i(1)(1) = -2\pi i$, which corresponds to a rotation of 2π radians or one revolution.

If $f = 2$, then the graph would make two revolutions going clockwise around the origin and thus its function values are separated by twice the angle. This is because the greater the $|f|$, the faster the vector rotates and the larger the separation angle between consecutive function values of $Y(x)$, resulting in different graphs.

On the other hand, if $f = -1$, then $R(x, f)$ goes around the origin once but in the counter-clockwise direction, which has the net effect of flipping the figure about the real axis, as compared below, with $f = -1$ on the left and $f = 1$ on the right:



If $f = -2$, then it makes two full revolutions in the counter-clockwise direction. Other than the negative sign in front, increasing $|f|$ still causes $Y(x)$ to be distributed more sparsely.

If we think of the graph of $R(x, f)$ at a certain frequency f as a system of point masses (or a wire if the whole interval $[0, 1]$ was translated), then the figure would have a center of mass, which is represented by a point somewhere in the complex plane. The center of mass is indicated by the red dot in the above figures.

The x and y coordinates of the center of mass are simply the arithmetic means of the x-coordinates and y-coordinates, respectively, of these n points.

$$x_c = \frac{1}{n}\sum_{k=1}^{n} x_k, \qquad y_c = \frac{1}{n}\sum_{k=1}^{n} x_k$$

However, the x-coordinate of each point is simply the real component of $R(n, f)$, and the y-coordinate is the imaginary component of $R(n, f)$ in the complex plane. Furthermore, the center of mass can also be represented in a single complex number c whose real and imaginary components are its x and y coordinates, respectively, on the complex plane. Therefore we can write:

$$x_c = \frac{1}{n}\sum_{k=1}^{n} \mathrm{Re}\big(R(x_k, f)\big), \qquad y_c = \frac{1}{n}\sum_{k=1}^{n} \mathrm{Im}\big(R(x_k, f)\big)$$

and we also have:

$$c = x_c + iy_c$$

$$c = \frac{1}{n}\sum_{k=1}^{n} \mathrm{Re}\big(R(x_k, f)\big) + \frac{i}{n}\sum_{k=1}^{n} \mathrm{Im}\big(R(x_k, f)\big)$$

$$c = \frac{1}{n}\sum_{k=1}^{n} \Big(\mathrm{Re}\big(R(x_k, f)\big) + i\,\mathrm{Im}\big(R(x_k, f)\big)\Big)$$

$$c = \frac{1}{n}\sum_{k=1}^{n} R(x_k, f)$$

$$c = \frac{1}{n}\sum_{k=1}^{n} Y(x)e^{-2\pi i f(x_k)}$$

For clarity, we can write the exponential alternatively as:

$$c = \frac{1}{n}\sum_{k=1}^{n} Y(x)\exp(-2\pi i f x_k)$$

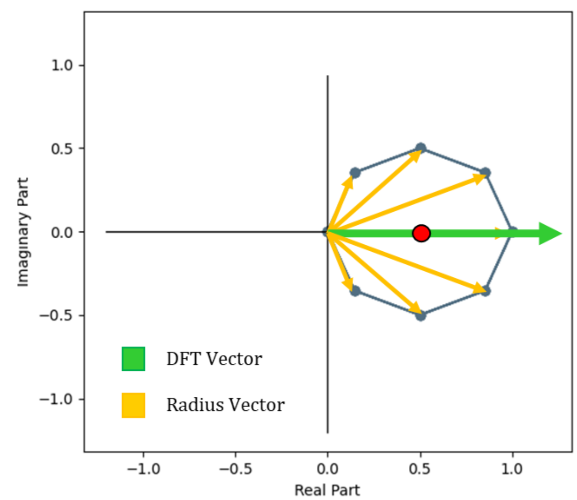
Multiplying both sides by n yields:

$$nc = \sum_{k=1}^{n} Y(x)\exp(-2\pi i f x_k)$$

If we change the indices to 0 to $n-1$ to follow conventions in computer programming, the summation on the right is called the ***Discrete Fourier Transform*** (DFT) of $Y(x)$ and it is defined on a frequency f_j as:

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k)\exp(-2\pi i f_j x_k)$$

It can be interpreted as the sum of the radius vectors that result by distributing the function values of $Y(x)$ around the unit circle at a frequency of f revolutions per unit of x.



From the left side, the DFT can also be viewed as the nth scalar multiple of the vector to the center of mass. This implies that the value of the DFT has a magnitude $|U(f)|$ *proportional* to the distance of the center of mass from the origin and a phase angle that is *equal to* the angle from the positive real axis to the center of mass.

Next, we note that we will use two separate indices for writing the definition of the DFT: k runs through the values x_k selected from the time domain, and j runs through the values f_j corresponding to the frequency domain. Both have values range from 0 to $n-1$, that is:

$$k = 0, 1, 2, \ldots, n-1 \qquad \text{and} \qquad j = 0, 1, 2, \ldots, n-1$$

The input of the DFT is assumed to be the vector $\mathbf{y} = (Y(x_0), Y(x_1), Y(x_2), \ldots, Y(x_{n-1}))$ in R^n which contains n values of $Y(x)$ on $[0, 1]$. The output of the DFT is also a vector $\mathbf{u} = \left(U(f_0), U(f_1), U(f_2), \ldots, U(f_{n-1})\right)$ in R^n containing the n complex DFT values for the frequency domain.

Although we can theoretically pick any n unique values of x for the DFT as long as its range is roughly the time domain, this is not the case for the n frequencies allowed in the DFT. In fact, the following facts show ***assumptions for the frequency in the DFT***:

a. *The DFT is defined only on n discrete values of x and n discrete values of f.*

b. *Each input value for the DFT is associated with the value of $Y(x)$ in the time domain and each output value is associated with the value of the DFT $U(f)$ in the frequency domain.*

c. *At the first nonzero discrete frequency, f_1, wrapping the function in the complex plane results in a graph that covers exactly 1 revolution around the origin.*

d. *The jth frequency f_j is the jth scalar multiple of the first frequency f_1, that is:*

$$f_0 = 0,\, f_1 = f_1,\;\, f_2 = 2f_1,\;\; f_3 = 3f_1, \ldots, f_{n-1} = (n-1)f_1.$$

Fact (a) ensures that the DFT transforms vectors from R^n to R^n. Fact (c) requires that $f_1 = 1$ for a time domain of $[0, 1]$ since the wrapping action must cover exactly one revolution from $x = 0$ to $x = 1$, that is:

$$\exp\left(-2\pi i f_j x_k\right) \text{ traces out one revolution from } x_0 = 0$$

$$-2\pi f_1(1) = -2\pi$$

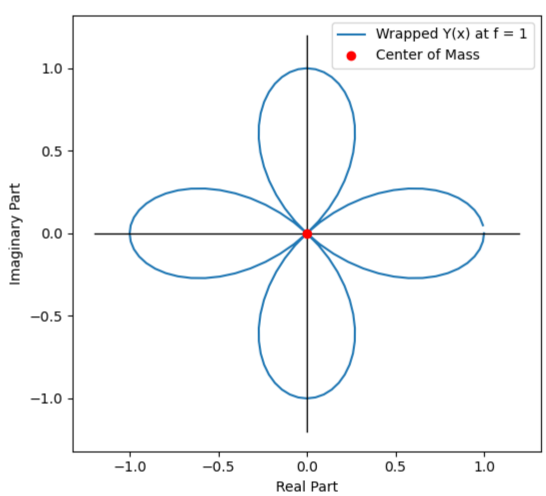
$$f_1 = 1$$

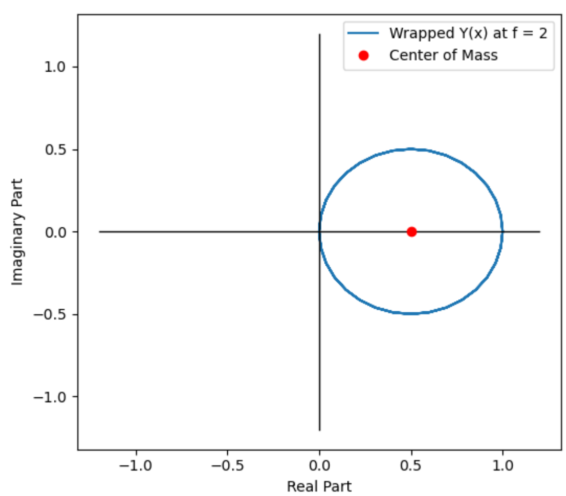
Finally, fact (d) requires that $f_0 = 0, \quad f_1 = 1, \quad f_2 = 2, \quad f_3 = 3, \ldots., f_{n-1} = n - 1$, or more simply:

$$f_j = j, \qquad j = 0, 1, 2, 3, \ldots, n-1$$

While these assumptions apply for any time domain chosen, the relation $f_j = j$ applies when the time domain is $[0, 1]$. We will assume this until we discuss expanding the time domain beyond $[0, 1]$.

As we take the limit $n \to \infty$, that is, we take more and more samples for $Y(x)$, the set of points becomes a continuous curve representing the graph of $R(x, f)$:





Then, the center of mass c of the wrapped graph converges to the following value:

$$c = \int_0^1 Y(x) \exp(-2\pi i f x)\; dx$$

Now if we take the limit as the time domain expands from $[0, 1]$ to all real numbers, or the interval $(-\infty, \infty)$, the integral on the right side becomes the following and it can be formally defined as the ***Continuous Fourier Transform*** (CFT) of $Y(x)$:

$$U(f) = \mathcal{F}\{Y(x)\} = \int_{-\infty}^{\infty} Y(x) \exp(-2\pi i f x)\ dx$$

As before, the Continuous Fourier Transform at f can be interpreted as a scalar multiple of the center of mass c of the figure formed by the graph of wrapping $Y(x)$ from all values of x in the complex plane.

Thus, we can write the DFT and CFT as:

$$\text{DFT:} \qquad U(f_j) = \sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi i f_j x_k)$$

$$\text{CFT:} \qquad U(f) = \mathcal{F}\{Y(x)\} = \int_{-\infty}^{\infty} Y(x) \exp(-2\pi i f x)\ dx$$

Note that there is another way to write the formula for the DFT using the indices j and k given that the values of x sampled are equally spaced in the time domain. We will discuss this in Section 5.

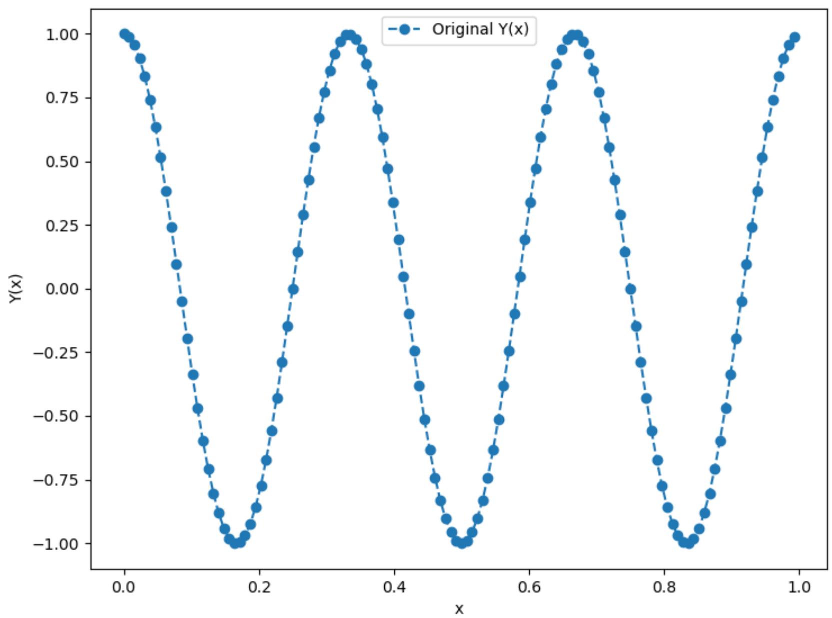
Section 3 - Discrete Fourier Transform as a Frequency Detector

Now that we know about that the fact that the Fourier Transform is a scalar multiple of the center of mass for the "wrapped" graph of $Y(x)$, how does this relate to the Fourier Transform as a "frequency detector"?

Suppose $Y(x) = \cos(6\pi x)$, which itself is a cosine wave of frequency of 3 Hz, and we choose $n = 128$ values of x in $[0, 1]$:

$$x = 0,\ \frac{1}{128}, \frac{2}{128}, \frac{3}{128}, \dots, \frac{127}{128}$$

The plot of $Y(x)$ is shown below:

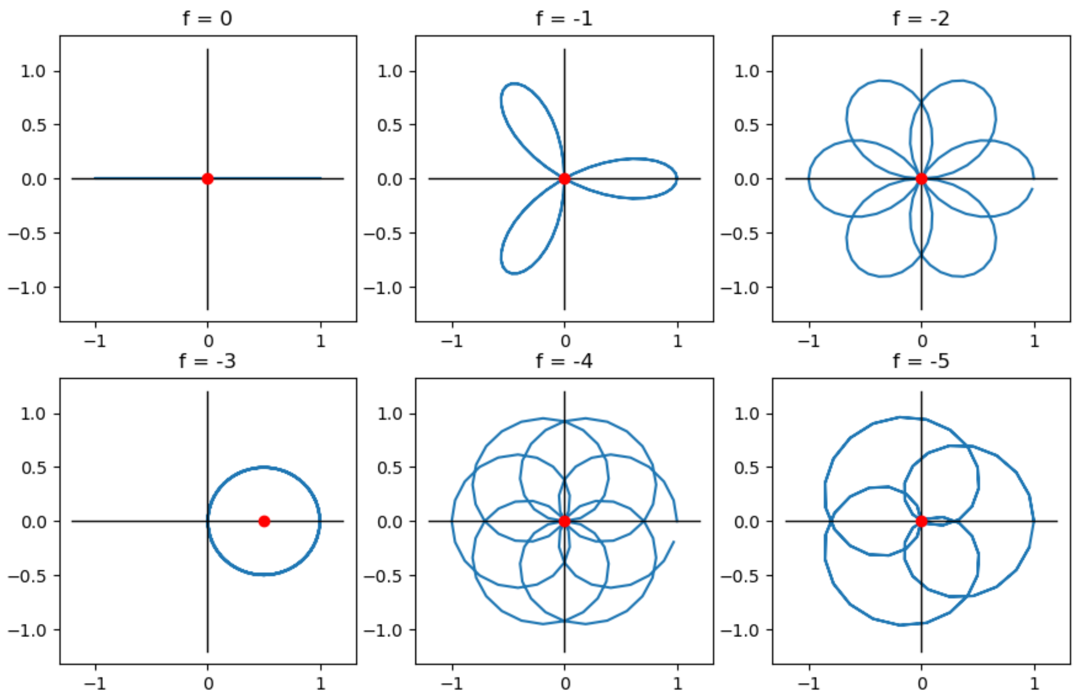


We wrap the graph of $Y(x)$ around the unit circle in the complex plane, forming $R(x, f) = \cos(6\pi x) \exp(-2\pi i f x)$, which traces out a figure on the complex plane going around the origin for each value of the frequency f. Below shows the plots of the wrapped curves and its center of mass (represented by the red point) for frequencies $f = 0, 1, 2, 3, 4,$ and 5:

For most values of f, the plot would look like symmetric flower with points distributed quite evenly around the origin. In these cases, the center of mass might wobble slightly but it would simply be near the origin $0 + 0i$, and since $b - a$ is constant, the magnitude $|U(f)|$of the DFT at all these frequencies is very small.

However, when $f = 3$, the plot becomes asymmetrical and the points are distributed at different distances from the origin. This means that the center of mass would also be far from the origin, and therefore $|U(f)|$ becomes large, which corresponds to the fact that the frequency $f = 3$ had been "detected" by the DFT. This is expected since $Y(x)$ is a sinusoid of 3 Hz.

Below shows the plots for the wrapped curves for the negative frequencies $f = 0, -1, -2, -3, -4,$ and -5:



Notice that the wrapped curves for the negative frequencies are simply the curves for positive frequencies but flipped about the real axis. This also means that the center of mass will also be flipped about the real axis. Moreover, the center of mass is far from the origin at $f = -3$, since $f = 3$ had a center of mass that was off from the origin. The center of masses for all other frequencies have negligible distances to the origin. Thus, the magnitude of the DFT also peaks at $f = -3$.

This occurrence between negative and positive frequencies is due to the following theorem for negative frequencies:

Theorem 1 – DFT of Negative Frequencies

For any frequency $f = m$, the Fourier Transform or Discrete Fourier Transform of a real-valued function $Y(x)$ at the negative frequency $f = -m$ is the complex conjugate of the FT or DFT at $f = m$.

In equation form, this may be written as:

$$U(-m) = \overline{U(m)}, \qquad \text{provided } Y(x) \text{ is real} - \text{valued}$$

This also implies that $U(-m)$ has the same magnitude but negative of the phase of $U(m)$ and that $\mathrm{Re}\big(U(-m)\big) = \mathrm{Re}\big(U(m)\big)$ and $\mathrm{Im}\big(U(-m)\big) = -\mathrm{Im}\big(U(m)\big)$. This applies to $U(-m)$ similarly.

The proof for Theorem 1 involves the fact that $Y(x) = \overline{Y(x)}$ for any real-valued function $Y(x)$.

Proof of Theorem 1 for DFT:

$$U(-m) = \sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi i(-m)x_k)$$

$$U(-m) = \sum_{k=0}^{n-1} Y(x_k) \exp(2\pi i m x_k)$$

$$U(-m) = \sum_{k=0}^{n-1} Y(x_k)(\cos(2\pi m x_k) + i \sin(2\pi m x_k))$$

$$U(-m) = \sum_{k=0}^{n-1} \overline{Y(x_k)}\overline{(\cos(-2\pi m x_k) + i \sin(-2\pi m x_k))}$$

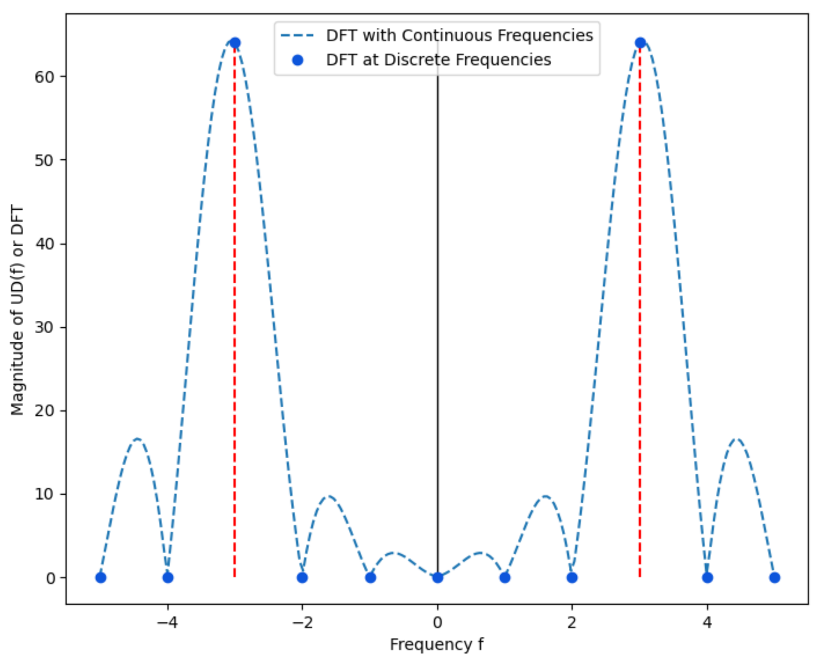
$$U(-m) = \sum_{k=0}^{n-1} \overline{Y(x_k) \exp(-2\pi i m x_k)}$$

$$U(-m) = \overline{\sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi i m x_k)}$$

$$U(-m) = \overline{U(m)}$$

The corresponding proof for CFT is similar.

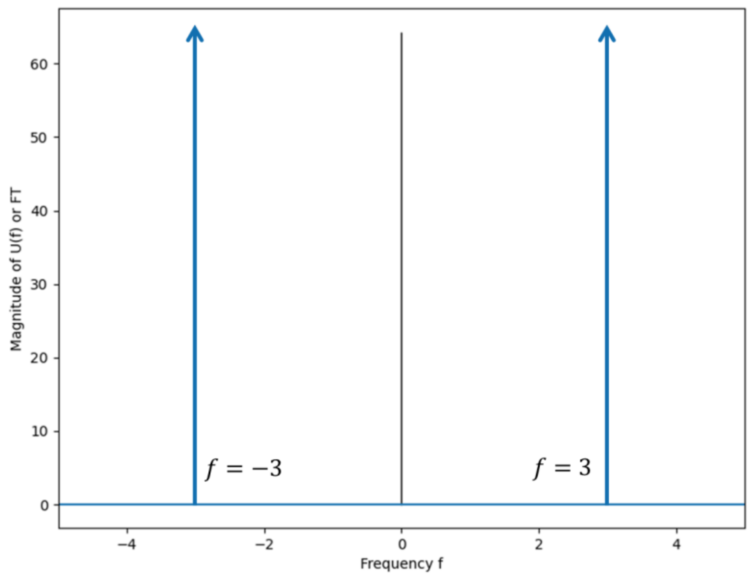
If we plot the magnitudes $|U(f)|$ of the DFT, or the distance of the center of mass from the origin, against all frequencies f within the interval $[-5, 5]$, we get the following graph:



It can be observed that the plot has spikes the highest at $f = 3$ and $f = -3$, as if the DFT has "detected" that $Y(x)$ is composed of a component wave of frequency $f = 3$ Hz.

Note that the actual DFT itself does not allow a continuous frequency domain. In fact, the graph for the magnitude of the actual DFT of $Y(x)$ would consist only of the blue points in the above figure. The dashed lines indicate the magnitude of the DFT or n times the distance of the center of mass outside of the discrete frequencies.

The values we have computed using the DFT only serve as approximations to the CFT, however, since we limited x to a bounded interval. As we expand the time domain and let it approach $(-\infty, \infty)$, the CFT values on any frequency that is not 3 or -3 approach zero or $(0 + 0i)$, while the CFT values at $f = 3$ and $f = -3$ become infinitely large in magnitude. How the DFT approximates the CFT is discussed in Section 21. Taking the limit gives the CFT of $Y(x) = \cos(6\pi x)$, whose magnitude has the following graph, with infinite spikes at $f = 3$ and $f = -3$:



One way to interpret the above graph is that the Fourier Transform detected exactly the frequency $f = 3$ Hz (and $f = -3$ Hz) and no other frequency.

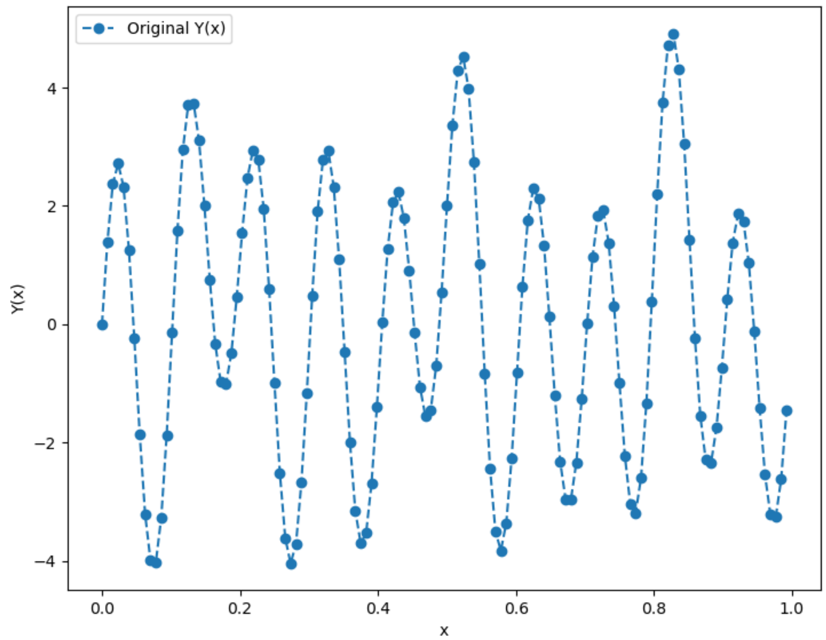
A similar scenario occurs for any function $Y(x)$ that is a linear combination of sines and/or cosines. If these sine and cosine functions have frequencies $f_1, f_2, \ldots, f_n$, then at those frequencies, the wrapped figure becomes asymmetrical and its center of mass would be far from the origin. Thus, the FT would "detect" these n frequencies and its magnitude would infinitely spike at frequencies $f_1, f_2, \ldots, f_n$ and $-f_1, -f_2, \ldots, -f_n$, while the FT value at any other frequency is 0.

The DFT also detects these frequencies and its magnitude spikes at the positive frequencies. As the sample size n approaches infinity, these spikes become infinitely tall and infinitely narrow. At every other frequency, the DFT value would have a negligible magnitude.

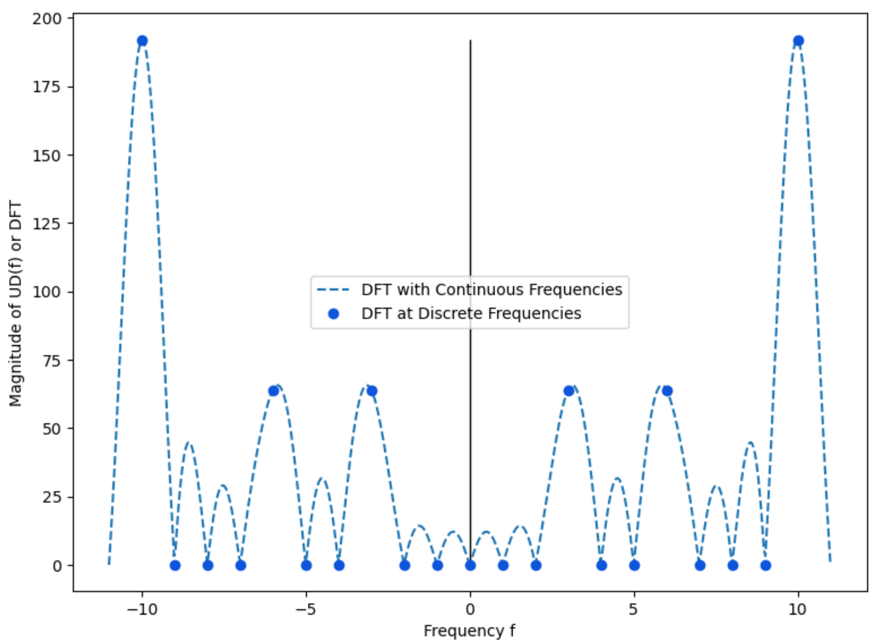
For example, say that $Y(x) = 3\sin(20\pi x) - \cos(6\pi x) + \sin\left(12\pi x + \frac{\pi}{2}\right)$ is a linear combination of three sinusoids at 3 Hz, 6 Hz, and 10 Hz, respectively, and we take $n = 128$ values of x from $[0, 1]$:

$$x = 0, \ \frac{1}{128}, \frac{2}{128}, \frac{3}{128}, \ \ldots, \ \frac{127}{128}$$

The function $Y(x)$ has the following graph:

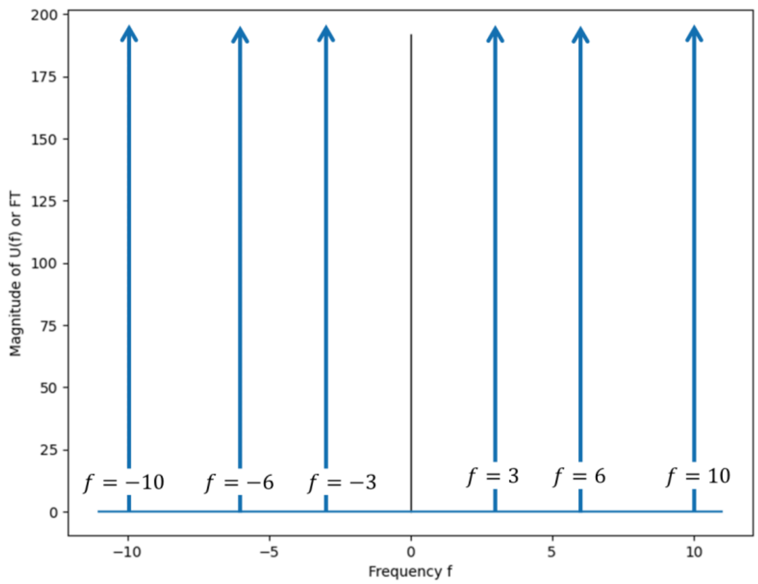


The plot below shows the magnitude of the DFT as f varies from $f = -7$ to $f = 7$:



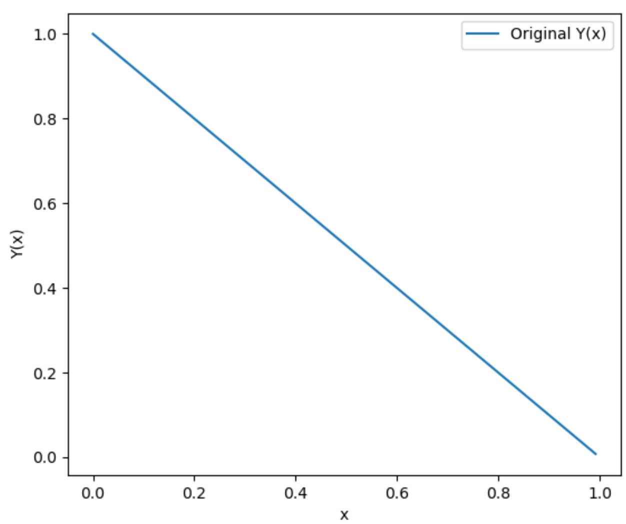
As expected, there are spikes at $f = 3, 6,$ and 10 and their negative frequencies in the above graph and thus the DFT detects component waves of frequencies $f = 3, 6,$ and 10. There are also smaller hills around the larger spikes which simply tells that the center of mass wobbles around the center as the frequency for wrapping the function changes.

The CFT of $Y(x)$ is plotted below:

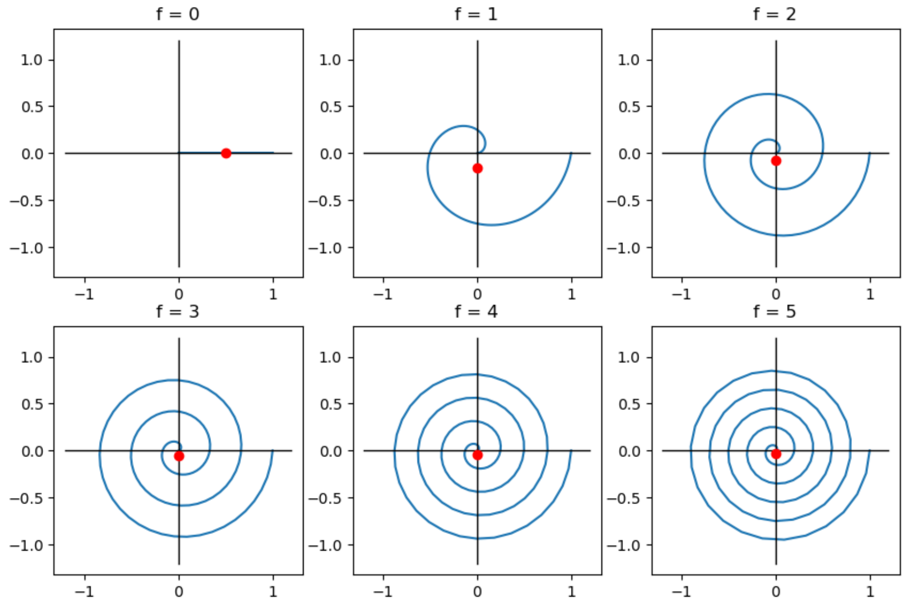


This means that as the number of values of x increases to infinity, the spikes at $f = 3, 6,$ and 10 and their respective negative frequencies become infinitely large and infinitely narrow, while the magnitudes at all other frequencies approach zero.

The function $Y(x)$ that is plugged into the DFT need not be a sinusoid or a linear combination of sinusoids. For instance, say that $n = 32$ and $Y(x) = 1 - x$, whose graph is the following:



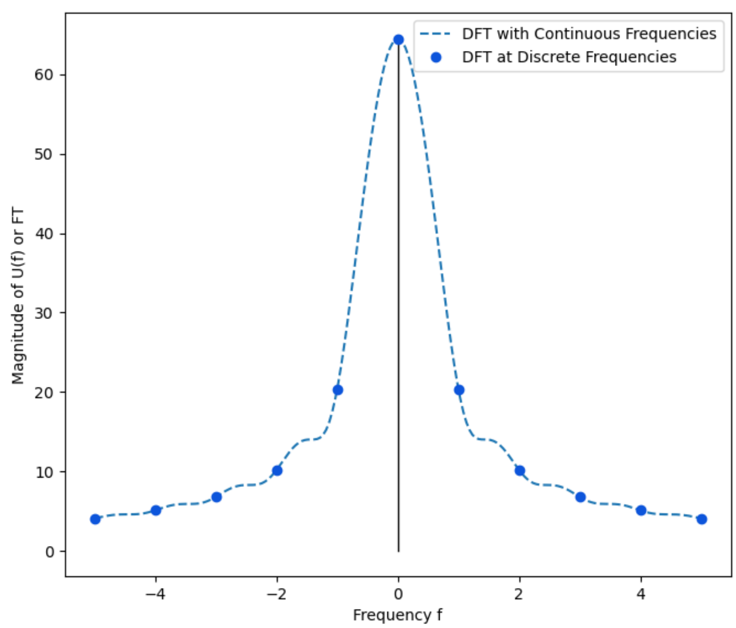
Wrapping the graph of $Y(x)$ at frequencies $f = 0, 1, 2, 3, 4,$ and 5 results in the following curves and center of masses:



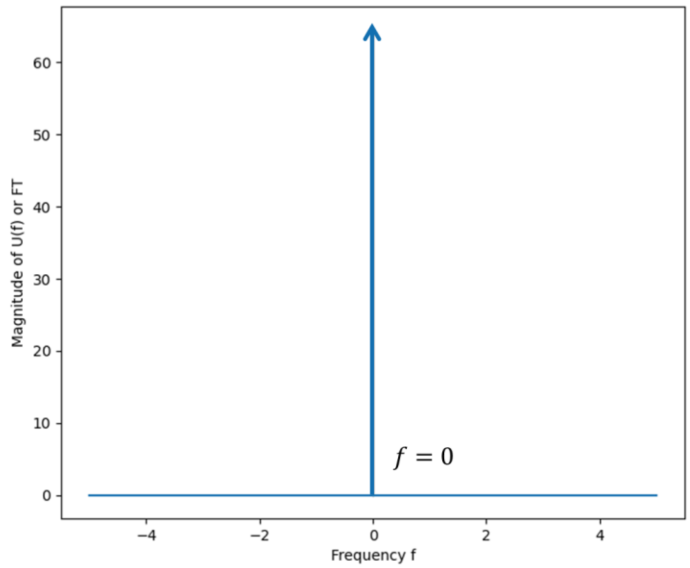
Notice that the center of mass for $f = 0, 1, 2,$ and 3 are located quite off from the origin, meaning that the magnitude of the DFT will be non-negligible at a range of frequencies around $f = 0$.

In fact, the figure formed in all of the above six plots is called an ***Archimedean Spiral***, where the distance of a point on the graph from the origin increases at a constant rate as the point goes around the spiral.

Below shows the plot of the magnitude of the DFT versus frequency:



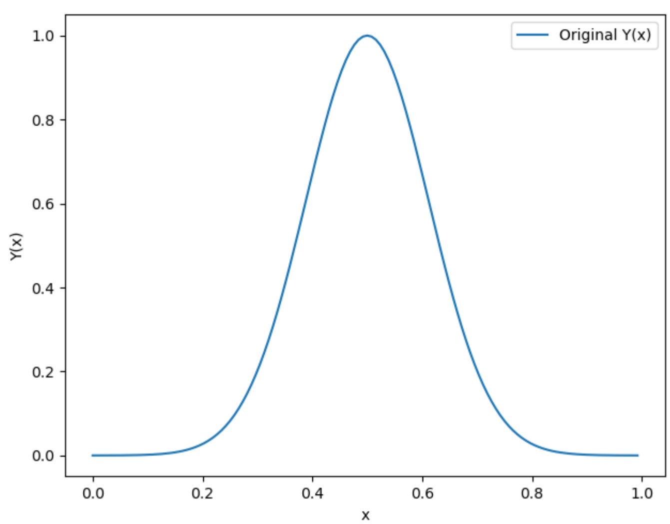
Thus, the DFT detects a range of frequencies composing $Y(x)$. However, the CFT of $Y(x)$, which is plotted below, still results in a spike, but at $f = 0$, despite $Y(x)$ not being a linear combination of sinusoidal functions:



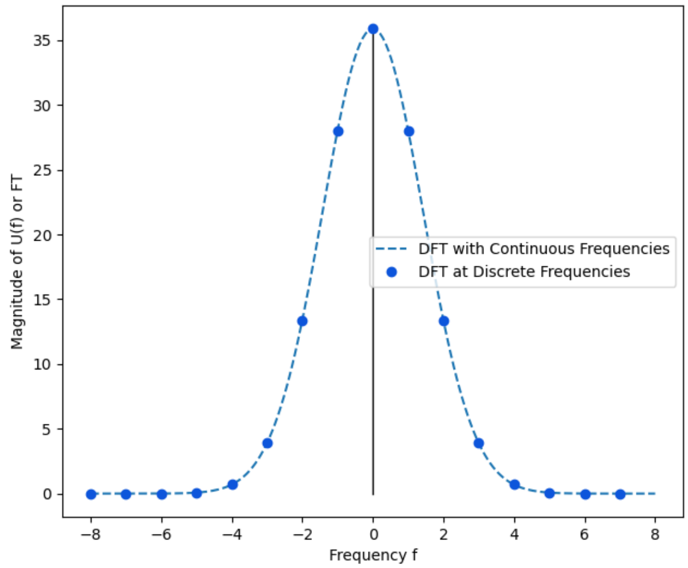
Next, let's say that $n = 128$ and $Y(x)$ represents the following function, whose graph is a Gaussian curve or a bell curve:

$$Y(x) = \exp(-40(x - 0.5)^2)$$

The graph of $Y(x)$ is shown below:



Then the graph for the magnitude of the DFT of $Y(x)$, as shown below, also forms a bell curve:



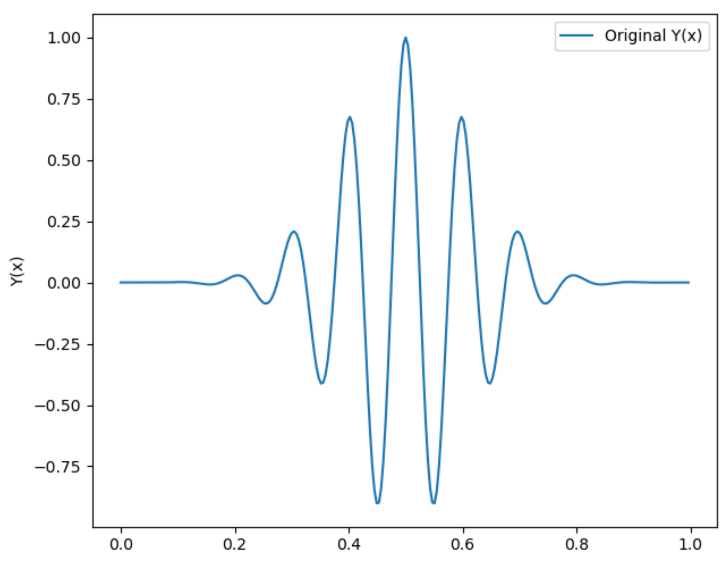
Once again, this means that the DFT has detected a range of frequencies around zero to be present in $Y(x)$. Unlike the previous example, however, it can be proved that the graph of the magnitude of the CFT of $Y(x)$ does not form an infinite spike at $f = 0$ but another bell curve with nonzero width. According to the FT, this means that $Y(x)$ is composed of an infinite number of frequencies within a range around $f = 0$.

Click the Link to more about the FT of a Gaussian: [Fourier Transform--Gaussian -- from Wolfram MathWorld]

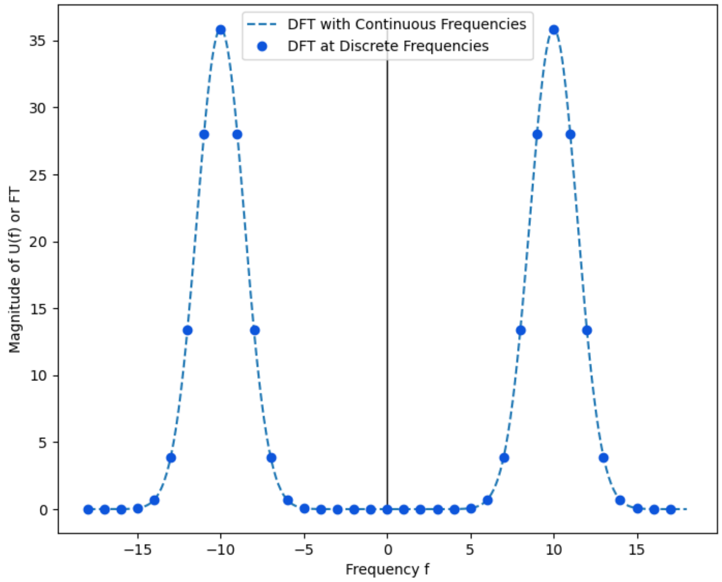
We can also set $n = 256$ and set $Y(x)$ to be the following function, which represents a ***Gaussian wave packet*** combining the characteristics of a Gaussian curve and a sinusoid:

$$Y(x) = \exp(-40(x - 0.5)^2) * \cos(20\pi x)$$

The graph of $Y(x)$ is shown below:



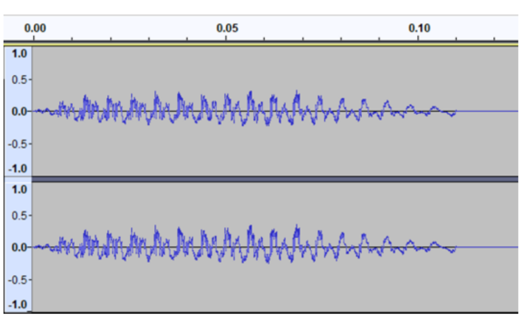
Then the graph of the magnitude for the DFT of $Y(x)$ forms a Gaussian curve or bell curve around $f = 10$, which is the frequency of the sinusoid, and consequently at $f = -10$:



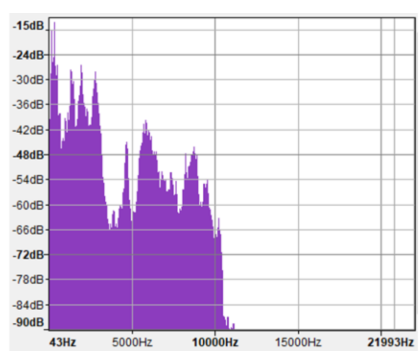
According to the DFT, this means that $Y(x)$ is composed of many frequencies near $f = 10$, including $f = 7$ through 13. As before, the graph for the magnitude of the CFT of $Y(x)$ will also be a bell curve around $f = 10$ (and $f = -10$) with a finite width.

In many applications outside of pure mathematics, $Y(x)$ represents a signal describing a physical measurement or variable, such as sound pressure, displacement, voltage, or energy, obtained at discrete moments in time and often the graph of $Y(x)$ does not follow a pattern or looks much more complex than the functions we studied in this section (below is an example of such signal). Since there are a discrete number of data points to define $Y(x)$, the continuous Fourier Transform cannot be used to analyze the frequencies making up $Y(x)$ and thus computing the discrete Fourier Transform is necessary of analysis. As we will see later, there is a well-known method, which is the Fast Fourier Transform (FFT), to compute for the DFT much more efficiently than using the center-of-mass formula we developed earlier.

Furthermore, negative frequencies and values of f outside of selected discrete frequencies are excluded when analyzing the frequency components of a signal via the DFT. It is also common to base the analysis on the magnitudes of the DFT values, rather than the wrapped curves or the complex DFT values themselves, as their value indicates how much of a frequency makes up the signal. Unless stated, we will exclude negative frequencies from the frequency domains in the succeeding examples.



$Y(x)$ as an audio signal



$|U_D(f)|$ or Magnitude of the DFT of $Y(x)$

Section 4 - Inner Product Interpretation of the Fourier Transform

Reference: Circles Sines and Signals - Discrete Fourier Transform Example (jackschaedler.github.io)

Recall that the Dot Product or Euclidean Inner Product for two vectors $\mathbf{u} = (u_1, u_2, \dots, u_n)$ and $\mathbf{v} = (v_1, v_2, \dots, v_n)$ in $\mathbb{R}^n$ is the sum of the products of the corresponding entries:

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + \cdots + u_nv_n = \|\mathbf{u}\|\|\mathbf{v}\| \cos\theta$$

Note that the expression $\mathbf{u} \cdot \mathbf{v}$ refers specifically to a dot product, and thus for inner products, we will introduce a new notation, $\langle \mathbf{u}, \mathbf{v} \rangle$, which will refer to the more general ***inner product*** between two vectors in a vector space where this inner product is defined (we will investigate this further in the next chapter).

As with vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$, we can interpret the dot product as a measure of *similarity* or *correlation* between the directions of two vectors. If $\mathbf{u} \cdot \mathbf{v} > 0$, then $\mathbf{u}$ and $\mathbf{v}$ are generally pointing in similar directions. If $\mathbf{u} \cdot \mathbf{v} = 0$, then the vectors are *orthogonal* (perpendicular) to one another and their directions are not similar. And if $\mathbf{u} \cdot \mathbf{v} < 0$, then the vectors are generally pointing in opposite directions.

We can in fact, extend this notion to the vector space of real functions $C(a, b)$ defined on the interval $[a, b]$ by applying the same principle to their function values. Suppose that we have two real functions $\boldsymbol{f} = f(x)$ and $\mathbf{g} = g(x)$ in $C(a, b)$, and we wish to find the inner product $\langle \mathbf{f}, \mathbf{g} \rangle$. Now pick n equally spaced values $x = x_1, x_2, \dots, x_n$ in $[a, b]$ where:

$$\Delta x = x_{k+1} - x_k \text{ for } k = 1, 2, \dots, n-1$$

and evaluate both $\mathbf{f}$ and $\mathbf{g}$ at each selected value of x. Then we can define $\langle \mathbf{f}, \mathbf{g} \rangle$ as the dot product of the vectors $\big(f(x_1),\ f(x_2), \dots, f(x_n)\big)$ and $\big(g(x_1),\ g(x_2), \dots, g(x_n)\big)$:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \big(f(x_1), f(x_2), \dots, f(x_n)\big) \cdot \big(g(x_1), g(x_2), \dots, g(x_n)\big)$$

$$\langle \mathbf{f}, \mathbf{g} \rangle = f(x_1)g(x_1) + f(x_2)g(x_2) + \cdots + f(x_n)g(x_n)$$

$$\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{k=1}^{n} f(x_k)g(x_k)$$

This referred to as the ***discrete inner product on functions***.

Now suppose we multiply the inner product by Δx:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \sum_{k=1}^{n} f(x_k)g(x_k)\,\Delta x$$

As $n \to \infty$, $\Delta x \to 0$ and the value of the inner product approaches a certain value, which is represented by taking the definite integral of the product $f(x)g(x)$ from $x = a$ to $x = b$. Thus, we define the ***continuous inner product on functions defined on*** $[\boldsymbol{a}, \boldsymbol{b}]$ as:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_a^b f(x)g(x)\,dx$$

It can be proved that this inner product has the same properties of the dot product in Theorem 3.2.2; that is, it satisfies the following properties for any two real functions $\mathbf{f} = f(x)$, $\mathbf{g} = g(x)$, and $\mathbf{h} = h(x)$ and any scalar k:

(a) Symmetry Property: $\langle \mathbf{f}, \mathbf{g} \rangle = \langle \mathbf{g}, \mathbf{f} \rangle$

(b) Distributive Property: $\langle \mathbf{f}, (\mathbf{g} + \mathbf{h}) \rangle = \langle \mathbf{f}, \mathbf{g} \rangle + \langle \mathbf{f}, \mathbf{h} \rangle$

(c) Homogeneity Property: $k\langle \mathbf{f}, \mathbf{g} \rangle = \langle k\mathbf{f}, \mathbf{g} \rangle$

(d) Positivity Property: $\langle \mathbf{f}, \mathbf{f} \rangle \geq 0$ and $\langle \mathbf{f}, \mathbf{f} \rangle = 0$ if and only if $\mathbf{f} = 0$

As with the dot product, the inner product on functions can also be interpreted as a measure of similarity between two functions.

For a more interactive approach to this topic, check the following links:

Circles Sines and Signals - Signal Correlation (jackschaedler.github.io)

Circles Sines and Signals - Correlation With Sine and Cosine (jackschaedler.github.io)

Let's say that we measure the inner product on two sinusoidal functions $\mathbf{f} = \cos(x)$ and $\mathbf{g} = \cos(x + \phi)$ of the same frequency but differing phase shifts, where ϕ is the phase shift of $\mathbf{g}$ in radians relative to $\mathbf{f}$. Then the inner product $\langle \mathbf{f}, \mathbf{g} \rangle$ on the interval $[-10, 10]$ can be written as:

$$\langle \mathbf{f}, \mathbf{g} \rangle = \int_{-10}^{10} \cos x \cos(x + \phi)\, dx$$

As we vary ϕ, $\langle \mathbf{f}, \mathbf{g} \rangle$ alternates in value as tabulated below:

$\boldsymbol{\phi}$	**0**	$\boldsymbol{\pi/4}$	$\boldsymbol{\pi/2}$	$\boldsymbol{3\pi/4}$	$\boldsymbol{\pi}$	$\boldsymbol{5\pi/4}$	$\boldsymbol{3\pi/2}$	$\boldsymbol{7\pi/4}$
$\langle \mathbf{f}, \mathbf{g} \rangle$	10.46	7.39	0	−7.39	−10.46	−7.39	0	7.39

Notice that when $\phi = \frac{\pi}{2}$ and $\phi = \frac{3\pi}{2}$, $\langle \mathbf{f}, \mathbf{g} \rangle = 0$ and the functions are orthogonal, and $\langle \mathbf{f}, \mathbf{g} \rangle > 0$ whenever $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$ (note that $\frac{3\pi}{2}$ radians around the circle is the same as $-\frac{\pi}{2}$), not unlike the dot product of two vectors in $\mathbb{R}^n$ that are spaced by an angle of ϕ radians.

From this, we also know that $\langle \cos x, \sin x \rangle = 0$ since $\sin x = \cos\left(\frac{\pi}{2} - x\right) = \cos\left(x - \frac{\pi}{2}\right)$.

Thus, an inner product of any sinusoidal signal $\mathbf{f}$ with a cosine or a sine can tell us the phase difference between the two functions given that they have the same frequency.

We can also use the inner product to detect whether a sinusoidal function has the same frequency with another. Say that we have functions $\mathbf{f} = \cos\left(\frac{1}{2}\pi x\right)$, $\mathbf{g} = \cos(2\pi x)$, and $\mathbf{h} = \cos(6\pi x)$ with frequencies of $\frac{1}{4}$, 1, and 3 hertz, respectively, and we take the inner product of each possible pairs of functions on the interval $[-10, 10]$. Then we get the following values:

	$\mathbf{f}$ (1/4 Hz)	$\mathbf{g}$ (1 Hz)	$\mathbf{h}$ (3 Hz)
$\mathbf{f}$ (1/4 Hz)	$\langle \mathbf{f}, \mathbf{f} \rangle = 10$	$\langle \mathbf{f}, \mathbf{g} \rangle = 0$	$\langle \mathbf{f}, \mathbf{h} \rangle = 0$
$\mathbf{g}$ (1 Hz)	$\langle \mathbf{g}, \mathbf{f} \rangle = 0$	$\langle \mathbf{g}, \mathbf{g} \rangle = 10$	$\langle \mathbf{g}, \mathbf{h} \rangle = 0$
$\mathbf{h}$ (3 Hz)	$\langle \mathbf{h}, \mathbf{f} \rangle = 0$	$\langle \mathbf{h}, \mathbf{g} \rangle = 0$	$\langle \mathbf{h}, \mathbf{h} \rangle = 10$

Notice that every inner product between cosine waves of differing frequencies is zero and that the only inner products in the table that are positive are those between the same cosine waves. If we were to change the interval of integration a bit, then the inner products between cosine waves of differing frequencies would not be exactly zero, but it would be close to zero, indicating that waves of different frequencies are orthogonal and therefore dissimilar to one another.

This fact holds true even if any of the waves were phase-shifted or multiplied by scalar. Thus, the inner product of any sinusoidal function with a cosine or a sine of a certain frequency can tell us whether the function has that same frequency, which forms the idea of "detecting" frequencies using the Fourier Transform, as we will discuss shortly after.

So far, we have dealt with taking the inner product between a sinusoidal function and another sine or a cosine. What about for functions with more complicated graphs? If we were to express such a function $\mathbf{f}$ as a sum of sines and cosines and $\mathbf{g}$ is a sinusoidal function whose frequency f is free to vary, then the value of $\langle \mathbf{f}, \mathbf{g} \rangle$ can tell us whether that frequency f is present in the function $\mathbf{f}$; if $\langle \mathbf{f}, \mathbf{g} \rangle$ is close to zero, then a wave of frequency f does not contribute to the overall graph of $\mathbf{f}$, but if the magnitude of $\langle \mathbf{f}, \mathbf{g} \rangle$ were large, then that frequency f is significant and $\mathbf{f}$ "contains" that frequency.

In fact, we can view the Continuous Fourier Transform as a complex number whose components are inner products, as shown below (ignoring technicalities regarding improper integrals):

$$U(f) = \int_{-\infty}^{\infty} Y(x) \exp(-2\pi i f x) \; dx$$

$$U(f) = \int_{-\infty}^{\infty} Y(x)(\cos(-2\pi f x) + i \sin(-2\pi f x)) \, dx$$

$$U(f) = \left(\int_{-\infty}^{\infty} Y(x) \cos(-2\pi f x) \; dx \right) + i \left(\int_{-\infty}^{\infty} Y(x) \sin(-2\pi f x) \, dx \right)$$

If we let $\mathbf{Y} = Y(x)$, $\mathbf{c} = \cos(-2\pi f x)$, and $\mathbf{s} = \sin(-2\pi f x)$, then the Continuous Fourier Transform is expressible as:

$$U(f) = \langle \mathbf{Y}, \mathbf{c} \rangle + i \langle \mathbf{Y}, \mathbf{s} \rangle$$

In this way, we can see that the Continuous Fourier Transform detects not just whether the frequency f is in $Y(x)$, but it also detects the phase shift of the wave contributing to $Y(x)$.

If the frequency f is not in $Y(x)$, then both $\langle \mathbf{Y}, \mathbf{c} \rangle$ and $\langle \mathbf{Y}, \mathbf{s} \rangle$ are close to zero and the magnitude of $U(f)$ is small. If that particular frequency were in $Y(x)$, however, then the magnitude of $U(f)$ is large, but this does not tell us directly how large are the magnitudes of $\langle \mathbf{Y}, \mathbf{c} \rangle$ and $\langle \mathbf{Y}, \mathbf{s} \rangle$, but their relative magnitudes tell us the phase of $U(f)$.

If the wave needs to have a phase shift close to either 0 or 180 degrees for it to "align" with $Y(x)$, then $|\langle \mathbf{Y}, \mathbf{c} \rangle|$ would be large and $|\langle \mathbf{Y}, \mathbf{s} \rangle|$ would be close to zero. On the other hand, if it needs to have a phase shift close to either 90 or -90 degrees, then $|\langle \mathbf{Y}, \mathbf{s} \rangle|$ would be large and $|\langle \mathbf{Y}, \mathbf{c} \rangle|$ would be close to zero. If the phase shift were any in between, then both $\langle \mathbf{Y}, \mathbf{c} \rangle$ and $\langle \mathbf{Y}, \mathbf{s} \rangle$ have moderately large values.

This is one way of explaining why we first considered the Fourier Transform as a "frequency detector".

The Discrete Fourier Transform can also be expressed similarly as a complex number of inner products, specifically the Euclidean Inner Product, whose values also detect the frequency and phase of a wave from the function $Y(x)$:

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi f_j i x_k)$$

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k)(\cos(-2\pi f_j x_k) + i \sin(-2\pi f_j x_k))$$

$$U(f_j) = \left(\sum_{k=0}^{n-1} Y(x_k) \cos(-2\pi f_j x_k) \right) + i \left(\sum_{k=0}^{n-1} Y(x_k) \sin(-2\pi f_j x_k) \right)$$

$$U(f_j) = (\mathbf{y} \cdot \mathbf{c}) + i(\mathbf{y} \cdot \mathbf{s})$$

where $\mathbf{y}$, $\mathbf{c}$, and $\mathbf{s}$ are vectors in $\mathbb{R}^n$ whose mth entries are the values of $Y(x)$, $\cos(-2\pi f_j x)$, and $\sin(-2\pi f_j x)$, respectively, evaluated at $x = x_m$.

NOTE: *There are two other interpretations, that being the matrix interpretation and polynomial interpretation, for the DFT (but not the continuous FT). The former is discussed in Section 5 and the latter in Section 6.*

Section 5 – Sampling for DFT and Matrix Interpretation of the DFT

While the formal definition for the Continuous Fourier Transform is sound mathematically, it is simply next to impossible to compute for its exact value given an arbitrary function $Y(x)$ without analytical methods as it does not make sense to compute an *infinite* number of values, and in fact, it would take forever to actually get to that point.

Instead, we can only select a finite number of discrete values x_k in the time domain through a process called ***sampling***. From these sampled values, we compute the value of the function $Y(x_k)$ at each of these points. This inevitably results in a loss of information on some details of the function $Y(x)$, but the more points we sample, the more accurate our data values will be to $Y(x)$. Then we compute the Discrete Fourier Transform using these sampled values.

How exactly are we going to sample both the physical and frequency domains and what values are we going to sample then?

Before, we assumed that all the n x_k's are arbitrarily picked in the time domain $[a,b]$. In practical applications, however, the points that are sampled are equally spaced in $[a,b]$ and includes the starting boundary a while excluding the ending boundary b. For example, if we sampled five values from the interval [0, 1], we would get the values 0, 0.2, 0.4, 0.6, and 0.8. If we assume sampling equally-spaced points on the time domain [0, 1], we can rewrite x_k as:

$$x_k = \frac{k}{n}, \qquad k = 0, 1, 2, \dots, n-1$$

In this case, we can refer to the value of n as not only the ***number of samples***, but also as the ***sampling rate*** $s = n$ in the unit of frequency or Hz, that is, the number of samples taken per unit of the time domain, or when written mathematically:

$$s = \frac{n}{L}, \qquad L = \text{width of time domain}$$

Note that $s \neq n$ if the width of the time domain is other than 1.

As discussed before, we can express each of the discrete frequencies f_j being referred to in the DFT as:

$$f_j = j, \qquad j = 0, 1, 2, \dots, n-1$$

Thus, the maximum frequency being referred to in the values of the DFT is $f_{n-1} = n - 1$.

As we will prove later, however, only the DFT values corresponding to the first half of the n frequencies, from $f_0 = 0$ to $f_{\frac{n}{2}-1} = \frac{n}{2} - 1$ actually convey useful information about the original data and that the DFT values for the second half are only complex conjugates of values in the first half. It is as if the second half of the DFT values is a "reflection" of those in the first half about the middle frequency or ***Nyquist frequency*** $f_N = \frac{s}{2} = \frac{n}{2}$ which is half of the sampling rate, as we will discuss later.

This phenomenon is described by the following theorem:

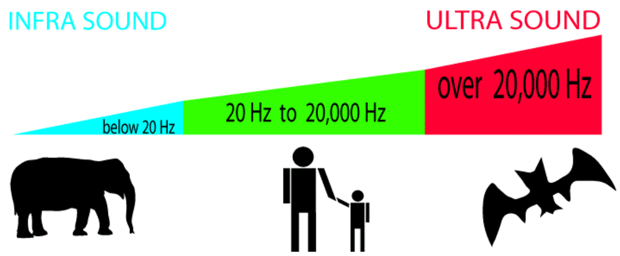
Theorem 2 – Nyquist-Shannon Sampling Theorem

A periodic signal must be sampled at more than twice the highest frequency component of the signal which is to be measured.

This principle was first discovered by the British mathematician Sir Edmund Taylor Whittaker in 1915, but its name honors Harry Nyquist and Claude Shannon, both of which who contributed to the description and proof of the theorem.

For example, if a signal has 500 Hz as its highest frequency component, then it must be sampled at a sampling rate of 1000 Hz or greater. This allows the Nyquist frequency to be 500 Hz or greater and thus the DFT captures all the frequency information of the signal. On the other hand, if the sampling rate is fixed at 500 Hz, then the DFT will be accurate only for signals whose highest frequency component is 250 Hz or less.

This is principle also explains why the sampling rate of most digital audio samples is at 44100 Hz. According to the sampling theorem, the highest frequency that may be accurately captured by the audio sample is 22050 Hz. However, since the hearing range of humans extends only up to 20000 Hz, this is sufficient to capture all the frequency information in the sound that may be perceived by the listener.



To see a visualization of the Nyquist-Shannon Sampling Theorem, click the link below:

[Circles Sines and Signals - Nyquist Frequency (jackschaedler.github.io)]

The DFT values on the time domain $[0, 1]$ can be calculated using the earlier formula but rewritten slightly assuming equally-spaced sampling:

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi i f_j x_k)$$

$$U(j) = \sum_{k=0}^{n-1} Y(k) \exp\left(-2\pi i \left(\frac{jk}{n}\right)\right)$$

Let ω be the complex number such that ω^n is 1:

$$\omega = \exp\left(\frac{-2\pi i}{n}\right)$$

We can interpret ω as the "unit" step of rotating clockwise around the unit circle in the complex plane at a step angle of $\frac{2\pi}{n}$ radians. Thus, ω^n corresponds to n steps around the unit circle and therefore represents one revolution.

Then the DFT can be rewritten more simply as:

$$U(f_j) = \sum_{k=0}^{n-1} Y(k)\, \omega^{jk}$$

Assuming the time domain $[0, 1]$ for x, we know that $f_j = j$. Then we can interpret $U(f_j) = U(j)$ as "distributing" the n function values $Y(x_k)$ around the unit circle in the complex plane for every j steps, each step of which represents $\frac{2\pi}{n}$ radians in the clockwise direction. Since there are n function values, this process would make a total of $j(n-1)$ steps, which corresponds to j revolutions. Thus $U(f_1)$ distributes the function values per 1 step in 1 revolution, $U(f_2)$ distributes each value per two steps (so twice the angle) in 2 revolutions, $U(f_{n-1})$ distributes per $(n-1)$ steps (so $n-1$ times the angle) in $(n-1)$ revolutions, and so on.

Since there are n discrete values for x_k, $Y(x_k)$, f_j, and $U(f_j)$, we can rewrite the DFT by expanding it into the following system of equations:

$$U(f_0) = Y(x_0)\omega^{(0)(0)} + Y(x_1)\omega^{(0)(1)} + Y(x_2)\omega^{(0)(2)} + \cdots + Y(x_{n-1})\omega^{(0)(n-1)}$$

$$U(f_1) = Y(x_0)\omega^{(1)(0)} + Y(x_1)\omega^{(1)(1)} + Y(x_2)\omega^{(1)(2)} + \cdots + Y(x_{n-1})\omega^{(1)(n-1)}$$

$$U(f_2) = Y(x_0)\omega^{(2)(0)} + Y(x_1)\omega^{(2)(1)} + Y(x_2)\omega^{(2)(2)} + \cdots + Y(x_{n-1})\omega^{(2)(n-1)}$$

$$\vdots$$

$$U(f_{n-1}) = Y(x_0)\omega^{(n-1)(0)} + Y(x_1)\omega^{(n-1)(1)} + Y(x_2)\omega^{(n-1)(2)} + \cdots + Y(x_{n-1})\omega^{(n-1)(n-1)}$$

This can be rewritten as a matrix transformation on the vector $\mathbf{y}$ in R^n containing the function values $Y(x_k)$:

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} \omega^{(0)(0)} & \omega^{(0)(1)} & \omega^{(0)(2)} & \cdots & \omega^{(0)(n-1)} \\ \omega^{(1)(0)} & \omega^{(1)(1)} & \omega^{(1)(2)} & \cdots & \omega^{(1)(n-1)} \\ \omega^{(2)(0)} & \omega^{(2)(1)} & \omega^{(2)(2)} & \cdots & \omega^{(2)(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^{(n-1)(0)} & \omega^{(n-1)(1)} & \omega^{(n-1)(2)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}$$

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}$$

$$\mathbf{u} = F_n\mathbf{y}$$

where F_n is the nth order Fourier Matrix whose entries $[F_n]_{ij}$ are given by:

$$[F_n]_{ij} = \omega^{ij}, \qquad i, j = 0, 1, 2, \ldots n-1$$

The above equation forms the ***matrix interpretation of the Discrete Fourier Transform***, which states that the DFT is a ***matrix transformation*** from $\mathbb{R}^n$ to $\mathbb{R}^n$, and thus also a ***matrix operator*** on $\mathbb{R}^n$, which maps function values in the time domain to DFT values in the frequency domain.

Computing the DFT directly might seem plausible, but since the number of coefficients in F_n grow by the square of n, the algorithm of multiplying the input vector by the Fourier Matrix directly would be inefficient as it has an algorithmic complexity of $O(n^2)$. However, as we will see in Sections 6 through 9, the Fast Fourier Transform allows us to compute the DFT much more efficiently at a complexity of $O(n \log_2 n)$.

Section 6 - Radix-2 Fast Fourier Transform (R2-FFT) and Polynomial Evaluation

More Info: [The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever? - YouTube]

Recall that if

$$\omega = \exp\left(\frac{-2\pi i}{n}\right)$$

then the values of the Discrete Fourier Transform on the time domain $[0, 1]$ for x at frequencies $f_0 = 0, \;\; f_1 = 1, \dots, f_{n-1} = n - 1$ can be computed by multiplying the function values of $Y(x)$ at $x = x_0, x_1, \dots, x_{n-1}$ by the nth order Fourier Matrix:

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}$$

$$\mathbf{u} = F_n \mathbf{y}$$

Unfortunately, this algorithm would have a complexity of $O(n^2)$, which disallows the method for computing the DFT's of large data sets. However, upon rewriting the system in the following form:

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & (\omega^2)^2 & \cdots & (\omega^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & (\omega^{n-1})^2 & \cdots & (\omega^{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}$$

And multiplying through, we can see that the system simply evaluates the polynomial

$$P(z) = Y(x_0) + z\,Y(x_1) + z^2\,Y(x_2) + \cdots + z^{n-1}\,Y(x_{n-1})$$

at $z = 1, \omega, \omega^2, \dots \omega^{n-1}$, of which are n complex numbers on the unit circle whose phases are spaced out equally around it. These are collectively known as the ***nth roots of unity***, since the nth power of each value is 1, as we will prove later.

The above equation, which relates the DFT to evaluating a polynomial $P(z)$, is referred to as the ***polynomial interpretation of the discrete Fourier transform***.

For clarity, let us represent $Y(x_k)$ as y_k and $U(f_k)$ as u_k for $k = 0, 1, 2, \dots, (n-1)$. Then we can rewrite $P(z)$ as:

$$P(z) = y_0 + y_1 z + y_2 z^2 + \cdots + y_{n-1} z^{n-1}$$

However, even simply evaluating $P(z)$ at n distinct points by direct substitution would still have complexity $O(n^2)$ since it is equivalent to multiplying the vector $\mathbf{y}$ of polynomial coefficients by the $n \times n$ Fourier matrix. It is noted that the Fourier matrix also has the form of the ***Vandermonde Matrix***:

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & (\omega^2)^2 & \cdots & (\omega^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & (\omega^{n-1})^2 & \cdots & (\omega^{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

If we can express the polynomial in such a way that the value of $P(z)$ at one point implies the value of $P(z)$ at another point, then such evaluation would be made much more efficient. To find such a method, we will need to consider the problem of evaluating a polynomial $P(z)$ at n points efficiently.

Section 7 - Polynomial Evaluation using Recursion

Let us take a simpler example to demonstrate that this is in fact possible if we had the pair $z_0, -z_0$; that is, we pick a point on the domain of $P(z)$ and we also pick its negative. Suppose $P(z) = 2z^3 - 2z^2 + 3z + 3$, and we wish to evaluate it at $z = 1$ and $z = -1$. If we were to directly substitute these values into $P(z)$, we would get:

$$P(1) = 2(1)^3 - 2(1)^2 + 3(1) + 3 = 6$$

$$P(-1) = 2(-1)^3 - 2(-1)^2 + 3(-1) + 3 = -4$$

If we count a kth power as $k - 1$ multiplications and count subtraction as adding by a negative, this process would all-in-all require 12 multiplications and 6 additions, for a total of 18 operations. Can we find a relationship between $P(z)$ and $P(-z)$ so as to calculate these same values more efficiently?

There does not seem to be an obvious relationship between the final computed values, however, negating z causes the signs of only the odd-power terms (terms with powers of $1, 3, 5, \dots$) to flip:

$$P(-1) = 2(-1)^3 - 2(-1)^2 + 3(-1) + 3$$

$$P(-1) = -2(1)^3 - 2(1)^2 - 3(1) + 3$$

which is comparable to the computation for $P(1)$:

$$P(1) = 2(1)^3 - 2(1)^2 + 3(1) + 3$$

In fact, evaluating an even power with a negative does not change its value while evaluating an odd power with a negative only flips its sign. This also holds true if we were to plug in complex numbers into $P(z)$ since they share similar algebraic properties as real numbers.

This also implies that if a polynomial $P(z)$ only has even-power terms (with or without a constant term), then $P(-z) = P(z)$ for all z.

Motivated by these facts, we rewrite $P(z)$ by separating the terms with even powers from those with odd powers and factoring:

$$P(z) = 2z^3 - 2z^2 + 3z + 3$$

$$P(z) = (-2z^2 + 3) + (2z^3 + 3z)$$

$$P(z) = (-2z^2 + 3) + z(2z^2 + 3)$$

Then, by applying the fact we developed earlier:

$$P(-z) = (-2(-z)^2 + 3) + (-z)(2(-z)^2 + 3)$$

$$P(-z) = (-2z^2 + 3) - z(2z^2 + 3)$$

Notice that both the polynomials $-2z^2 + 3$ and $2z^2 + 3$ only have even-power terms. This will always be the case whenever we rewrite the polynomial in terms of odd and even coefficients since any two consecutive odd numbers or two consecutive even numbers have a difference of 2.

Now if we let:

$$Q_1(z) = -2z + 3, \qquad Q_2(z) = 2z + 3$$

Then we can rewrite $P(z)$ and $P(-z)$ as:

$$P(z) = Q_1(z^2) + zQ_2(z^2)$$

$$P(-z) = Q_1(z^2) - zQ_2(z^2)$$

By making a variable substitution of $v = z^2$, we can rewrite a polynomial of degree n into two polynomials whose degrees are less than half of n!

Thus, if we wish to find $P(1)$ and $P(-1)$ using this method, we set z to be the unsigned value $z = 1$ and since $z^2 = 1$, we have:

$$P(1) = Q_1(1) + Q_2(1)$$

$$P(-1) = Q_1(1) - Q_2(1)$$

Evaluating Q_1 and Q_2 at $z^2 = 1$ gives:

$$Q_1(1) = -2(1) + 3 = 1$$

$$Q_2(1) = 2(1) + 3 = 5$$

then get $P(1)$ and $P(-1)$ from the derived expression above:

$$P(1) = Q_1(1^2) + (1)\,Q_2(1^2) = (1) + (1)(5) = 6$$

$$P(-1) = Q_1(1^2) - (1)\,Q_2(1^2) = (1) - (1)(5) = -4$$

which agrees with the previous results. Notice that this process took only 6 multiplications and 4 additions adding to a total of just 10 operations performed, compared to 18 from direct evaluation.

Notice that each iteration required only 2 values to be computed. Below is the table showing the values computed:

Level 1	**Level 2**
$\boldsymbol{P(1) = 6}$	$Q_1(1) = 1$
$\boldsymbol{P(1) = -4}$	$Q_2(1) = 5$

So far, we've only been dealing with 1 pair of numbers, however the DFT evaluates the $(n-1)$th degree polynomial $P(z)$ (ignore the example) at n points $z = 1, \omega, \omega^2, \dots, \omega^{n-1}$. In our case, if $P(z)$ for the DFT were to be the cubic polynomial $2z^3 - 2z^2 + 3z + 3$ used in the above example, then we will need to evaluate it at 4 points or 2 pairs.

What if we had a higher degree polynomial? Say that $P(z) = 2z^7 + 3z^6 - 3z^4 + 2z^3 - 5z + 8$, and we wish to evaluate it on the 8 values $z = 1, -1, i, -i,\ \sqrt{2}, -\sqrt{2}, \sqrt{2}i,$ and $-\sqrt{2}i$. Then the polynomial can be rewritten as:

$$P(z) = 2z^7 + 3z^6 - 3z^4 + 2z^3 - 5z + 8$$

$$P(z) = (3z^6 - 3z^4 + 8) + (2z^7 + 2z^3 - 5z)$$

$$P(z) = (3z^6 - 3z^4 + 8) + z(2z^6 + 2z^2 - 5)$$

$$P(z) = (3(z^2)^3 - 3(z^2)^2 + 8) + z(2(z^2)^3 + 2(z^2) - 5)$$

If we let

$$Q_1(z) = 3z^3 - 3z^2 + 8, \qquad Q_2(z) = 2z^3 + 2z - 5$$

then $P(z)$ and $P(-z)$ can be expressed as:

$$P(z) = Q_1(z^2) + zQ_2(z^2)$$

$$P(-z) = Q_1(z^2) - zQ_2(z^2)$$

Since we have 4 pairs of input values with opposite signs, we would have to get $P(z)$ AND $P(-z)$ by plugging in the unique unsigned values $z = 1, i, \sqrt{2}$, and $\sqrt{2}i$. However, each pair $P(z)$ AND $P(-z)$ require the values of $Q_1(z^2)$ AND $Q_2(z^2)$. We could directly evaluate these smaller polynomials, but we want to make our calculation as efficient as possible.

Instead, what we must do is to evaluate both $Q_1(z^2)$ and $Q_2(z^2)$ at the 4 unique squared values $z^2 = 1^2, i^2, \sqrt{2}^2$, and $(\sqrt{2}i)^2$ or $z^2 = 1, -1, 2, -2$, which forms 2 pairs of values with opposite signs.

We will apply the same technique as before, but on the cubic polynomials $Q_1(z)$ and $Q_2(z)$.

For $Q_1(z)$:

$$Q_1(z) = 3z^3 - 3z^2 + 8$$

$$Q_1(z) = (-3z^2 + 8) + (3z^3)$$

$$Q_1(z) = (-3z^2 + 8) + z(3z^2)$$

For $Q_2(z)$:

$$Q_2(z) = 2z^3 + 2z - 5$$

$$Q_2(z) = (-5) + (2z^3 + 2z)$$

$$Q_2(z) = (-5) + z(2z^2 + 2)$$

If we let:

$$R_1(z) = -3z + 8, \qquad R_2(z) = 3z$$

$$S_1(z) = -5, \qquad S_2(z) = 2z + 2$$

Then we can write $Q_1(z)$ and $Q_1(-z)$ as:

$$Q_1(z) = R_1(z^2) + zR_2(z^2)$$

$$Q_1(-z) = R_1(z^2) - zR_2(z^2)$$

and we can write $Q_2(z)$ and $Q_2(-z)$ as:

$$Q_2(z) = S_1(z^2) + zS_2(z^2)$$

$$Q_2(-z) = S_1(z^2) - zS_2(z^2)$$

For $Q_1(z)$ and $Q_2(z)$, we have to substitute in the unsigned values $z = 1$ and 2. However, they require the values of $R_1(z^2), R_2(z^2)$, $S_1(z^2)$, and $S_2(z^2)$, meaning that we need to evaluate R_1, R_2, S_1, and S_2 at the squared values $z^2 = 1^2, 2^2 = 1, 4$, which yields the following values:

$$R_1(1) = 5, \qquad R_2(1) = 3$$

$$R_1(4) = -4, \qquad R_2(4) = 12$$

$$S_1(1) = -5, \qquad S_2(1) = 4$$

$$S_1(4) = -5, \qquad S_2(4) = 10$$

Then we have the following values for Q_1:

$$Q_1(1) = (5) + (1)(3) = 8$$

$$Q_1(-1) = (5) - (1)(3) = 2$$

$$Q_1(2) = (-4) + (2)(12) = 20$$

$$Q_1(-2) = (-4) - (2)(12) = -28$$

And for Q_2:

$$Q_2(1) = (-5) + (1)(4) = -1$$

$$Q_2(-1) = (-5) - (1)(4) = -9$$

$$Q_2(2) = (-5) + (2)(10) = 15$$

$$Q_2(-2) = (-5) - (2)(10) = -25$$

By plugging in $z = 1, i, \sqrt{2}$, and $\sqrt{2}i$ into the formulas

$$P(z) = Q_1(z^2) + zQ_2(z^2)$$

$$P(-z) = Q_1(z^2) - zQ_2(z^2)$$

We have:

$$P(1) = (8) + (1)(-1) = 7$$

$$P(-1) = (8) - (1)(-1) = 9$$

$$P(i) = (2) + (i)(-9) = 2 - 9i$$

$$P(-i) = (2) - (i)(-9) = 2 + 9i$$

$$P(\sqrt{2}) = (20) + (\sqrt{2})(15) = 20 + 15\sqrt{2}$$

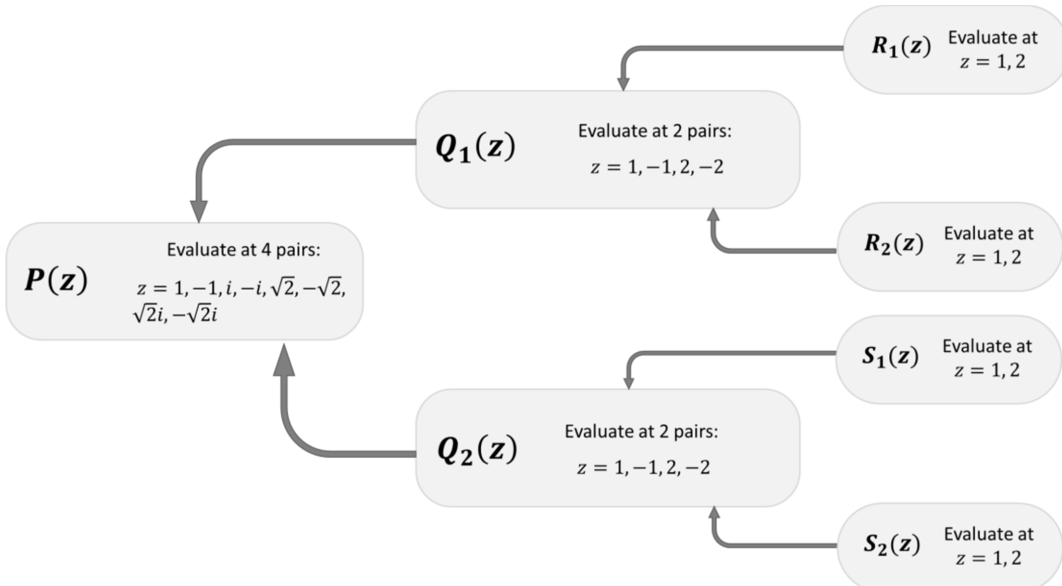
$$P(-\sqrt{2}) = (20) - (\sqrt{2})(15) = 20 - 15\sqrt{2}$$

$$P(\sqrt{2}i) = (-28) + (\sqrt{2}i)(-25) = -28 - 25\sqrt{2}i$$

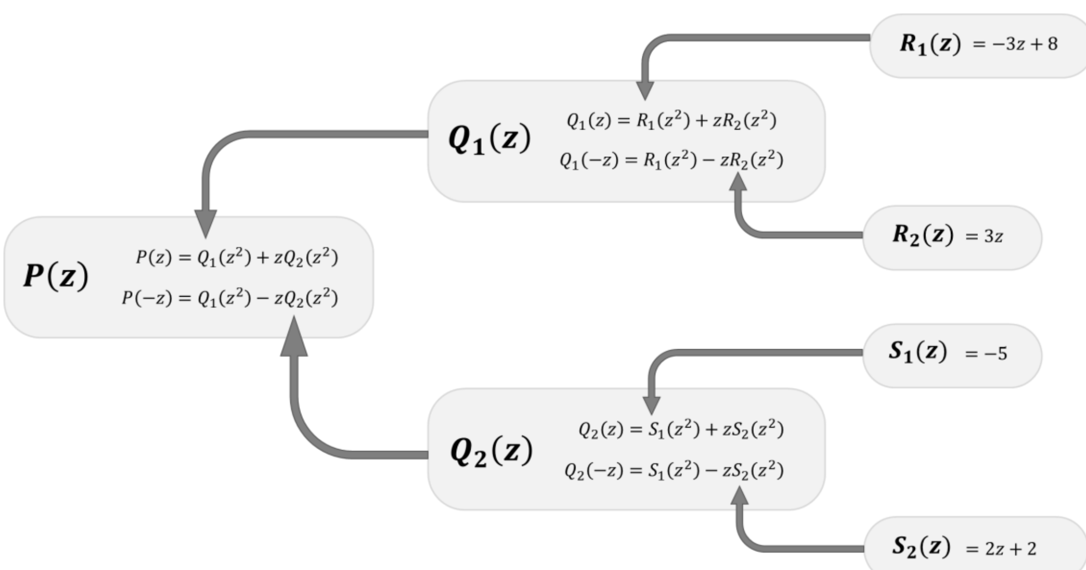
$$P(-\sqrt{2}i) = (-28) - (\sqrt{2}i)(-25) = -28 + 25\sqrt{2}i$$

Notice that at every iteration, we have $n = 8$ values that are to be computed.

Below is a diagram showing the steps to compute $P(z)$ at 8 points:



The equations used in each step is shown below:



Below is a table marking all the values computed in the example:

Level 1	Level 2	Level 3
$\boldsymbol{P(1) = 7}$	$Q_1(1) = 8$	$R_1(1) = 5$
$\boldsymbol{P(-1) = 9}$	$Q_2(1) = -1$	$R_2(1) = 3$
$\boldsymbol{P(\sqrt{2}) = 20 + 15\sqrt{2}}$	$Q_1(2) = 20$	$R_1(4) = -4$
$\boldsymbol{P(-\sqrt{2}) = 20 - 15\sqrt{2}}$	$Q_2(2) = 15$	$R_2(4) = 12$
$\boldsymbol{P(i) = 2 - 9i}$	$Q_1(-1) = 2$	$S_1(1) = -5$
$\boldsymbol{P(-i) = 2 + 9i}$	$Q_2(-1) = -9$	$S_2(1) = 4$
$\boldsymbol{P(\sqrt{2}i) = -28 - 25\sqrt{2}i}$	$Q_1(-2) = -28$	$S_1(4) = -5$
$\boldsymbol{P(\sqrt{2}i) = -28 + 25\sqrt{2}i}$	$Q_2(-2) = -25$	$S_2(4) = 10$

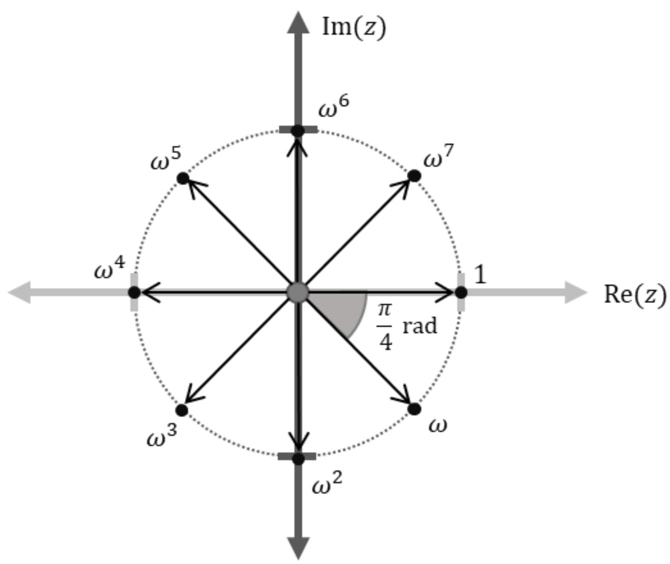
What we did above to get these values is to break a problem into smaller subproblems, which may be subdivided into more subproblems in the same way, which we keep doing until we hit a solvable problem, which is called a ***base case***. For instance, to compute $P(z)$ for the 4 pairs of numbers, we had to compute both $Q_1(z)$ and $Q_2(z)$ for 2 pairs (the squares of the previous numbers), which required us to compute $R_1(z), R_2(z), S_1(z)$, and $S_2(z)$ for 1 pair (which are also squares of the previous values). The last four functions became our base case since we found them directly by evaluating their expressions, while $P(z), Q_1(z)$ and $Q_2(z)$ are the ***recursive cases*** since they can be split each into smaller subproblems. This technique of splitting a problem into subproblems by using a rule is called ***recursion***, and this is key to the efficiency of the FFT compared to direct evaluation.

The procedure above required $6 + 8 + 8 = 22$ multiplications and $4 + 8 + 8 = 20$ additions for a total of 42 operations executed. In contrast, if we were to evaluate $P(z)$ directly assuming that raising z to the kth power ($k \geq 2$) counts as $(k - 1)$ multiplications, then each function value is obtained in 21 multiplications and 5 additions, meaning that evaluating at 8 points requires 168 multiplications and 40 additions for a total of 208 operations. The difference between the two methods is very significant; recursion required over 7 times less multiplication operations, half the number of addition operations, and roughly 5 times less the total number of operations compared to direct evaluation. As the number of values to evaluate increases in size, this difference becomes even more paramount.

Section 8 - Radix-2 FFT is Polynomial Evaluation on the Roots of Unity

Note that in Section 7, we were able to split the subproblems in the same way simply because the values we got by squaring the previous set of values also formed pairs of numbers such that one is the negative of the other. However, most values we select from the domain of $P(z)$ do not follow this behavior.

Luckily, the nth roots of unity $z = 1, \omega, \omega^2, \dots, \omega^{n-1}$ in the DFT have this property provided that n is a power of 2, since if n weren't a power of 2, then at least one number in the recursion would not be in a pair. To demonstrate this, fact consider a DFT with $n = 8$, $P(z)$ of degree 7, and $z = 1, \omega, \omega^2, \omega^3, \dots, \omega^7$. As shown in the figure below, they can be plotted on the complex plane as 8 equally-spaced points on the unit circle and the step angle between any two consecutive points is $\frac{2\pi}{8} = \frac{\pi}{4}$ radians.



Notice that these 8 points form pairs wherein one number is the negative of the other as there are 4 pairs of vectors pointing in opposite directions on the unit circle. In this case, we have $1 = -\omega^4$, $\omega = -\omega^5, \omega^2 = -\omega^6$, and $\omega^3 = -\omega^7$. Algebraically, this is a consequence of the following theorem for opposite vectors:

Theorem 3 – Symmetry Fact A

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$. *Then for all integers* n *and* k*:*

$$\omega^k = -\omega^{\frac{n}{2}+k}$$

or if expressed in terms of complex exponentials:

$$\exp\left(-\frac{2\pi ik}{n}\right) = -\exp\left(-\frac{2\pi i}{n}\left(\frac{n}{2} + k\right)\right)$$

$$\exp\left(-\frac{2\pi ik}{n}\right) = -\exp\left(-\frac{2\pi ik}{n} - \pi i\right)$$

$$\exp\left(-\frac{2\pi ik}{n}\right) = -\exp\left(-\left(\frac{2\pi k}{n} + \pi\right)i\right)$$

Proof for Theorem 3:

$$\exp\left(-\left(\frac{2\pi k}{n} + \pi\right)i\right) = \cos\left(-\left(\frac{2\pi ik}{n} + \pi\right)\right) + i\sin\left(-\left(\frac{2\pi ik}{n} + \pi\right)\right)$$

$$\exp\left(-\left(\frac{2\pi k}{n} + \pi\right)i\right) = \cos\left(-\frac{2\pi ik}{n}\right) - i\sin\left(-\frac{2\pi ik}{n}\right)$$

$$\exp\left(-\left(\frac{2\pi k}{n} + \pi\right)i\right) = -\exp\left(-\frac{2\pi ik}{n}\right)$$

$$\exp\left(-\frac{2\pi ik}{n}\right) = -\exp\left(-\left(\frac{2\pi k}{n} + \pi\right)i\right)$$

Upon applying the recursive algorithm which we developed to evaluate the 7th degree polynomial $P(z)$ at these complex number pairs, we see that we would also need to evaluate two 3rd degree polynomials at the squares $1, \omega^2, \omega^4, \omega^6$ of each complex number from each pair. The other numbers $\omega^8, \omega^{10}, \omega^{12}$ and ω^{14} have squares equal to $1, \omega^2, \omega^4$, and ω^6, respectively as a consequence of the following theorem for coterminal vectors:

Theorem 4 – Symmetry Fact B

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$*. Then for all integers m, n and k:*

$$\omega^k = \omega^{k-mn}$$

Proof for Theorem 4:

$$\omega^{k-mn} = \exp\left(-\frac{2\pi i(k-mn)}{n}\right)$$

$$\omega^{k-mn} = \exp\left(-\frac{2\pi ik}{n} + 2m\pi i\right)$$

$$\omega^{k-mn} = \exp\left(-\frac{2\pi ik}{n}\right)\exp(2m\pi i)$$

$$\omega^{k-mn} = \exp\left(-\frac{2\pi ik}{n}\right)(1)$$

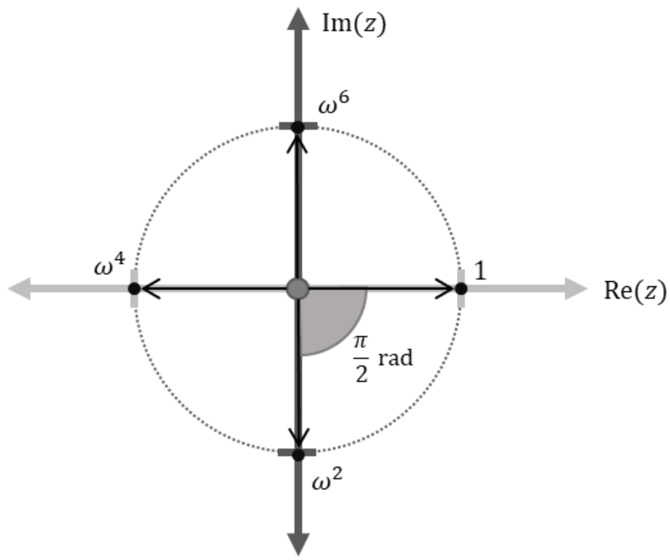
$$\omega^{k-mn} = \exp\left(-\frac{2\pi ik}{n}\right) = \omega^k$$

Theorem 4 also explains why $z = 1, \omega, \omega^2, \dots, \omega^{n-1}$ are the nth roots of unity. If we let $s = 0, 1, 2, \dots, n-1$, then $z = \omega^s$ and:

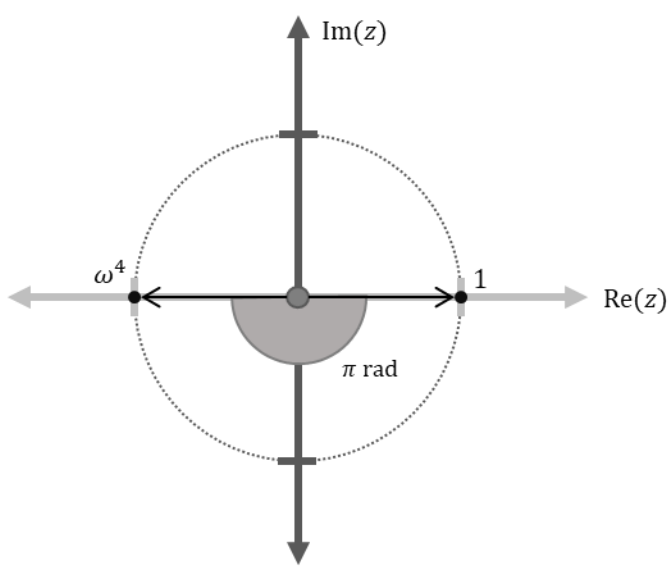
$$z^s = \omega^{sn}$$

Since s is an integer, Theorem 4 implies that $z^s = \omega^{sn-sn} = \omega^0 = 1$, hence z is an nth root of 1 or unity.

We can apply the algorithm again only if these squares also form pairs of a complex number and its negative, which these squares satisfy since two such pairs can be formed: $1 = -\omega^4$, and $\omega^2 = -\omega^6$:

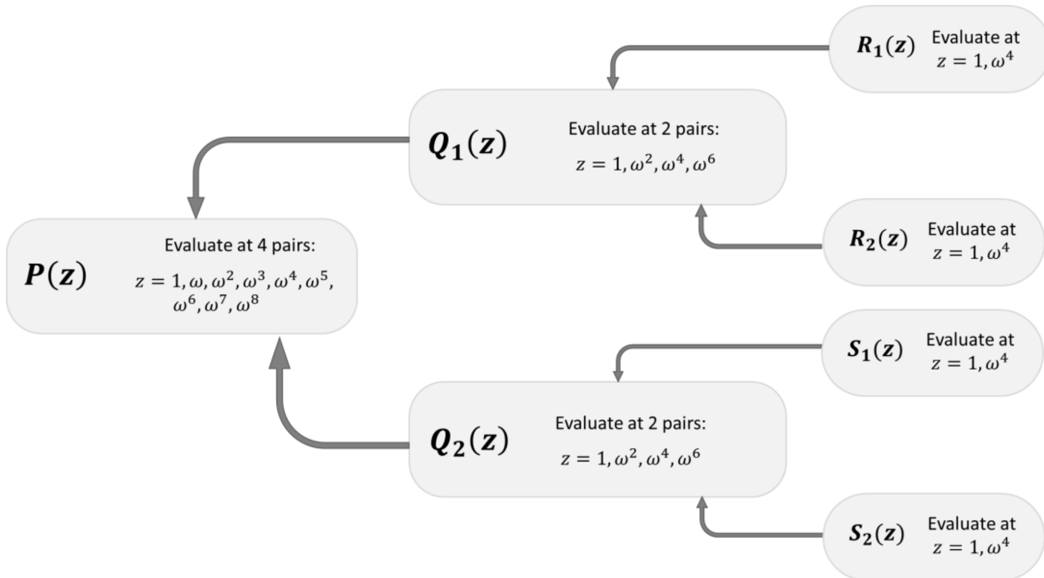


Notice that we did not get new numbers unlike in the previous example, since we simply took every second number in the initial 8 values we wish to evaluate $P(z)$ on. Upon applying the algorithm on the two 3rd degree polynomials, it turns out that we would need to evaluate four 1st degree polynomials on the squares 1 and ω^4, which clearly form a pair since $1 = -\omega^4$ as shown below:



We can directly evaluate these 4 linear polynomials at $z = 1$ and $z = \omega^4 = -1$, or we can apply the recursion again to get 8 0th degree polynomials or 8 constants to get the same result. Either way is no more efficient than the other. Then we can use these values to evaluate the 2 cubic polynomials at $z = 1, \omega^2, \omega^4$, and ω^6, whose values can be substituted into $P(z)$ itself to get its values at $z = 1, \omega, \omega^2, \omega^3, \dots, \omega^7$, which corresponds to the values of the DFT at frequencies $f = 0, f_1, f_2, \dots, f_7$ or if the interval for x is $[0, 1]$, $f = 0, 1, 2, \dots, 7$, respectively.

The diagram below shows at what values are these polynomials computed:



Notice that in the first iteration, we count every $2^{1-1} = 1$ step, then the second iteration counts every $2^{2-1} = 2$ steps, and the third iteration counts every $2^{3-1} = 4$ steps.

We call this algorithm of computing the DFT when n is a power of 2 using recursion the ***Radix-2 Fast Fourier Transform*** (Radix-2 FFT), and it is one of the simplest of the FFT algorithms. Together with other FFT algorithms, it is widely used in many applications to quickly and efficiently compute the DFT for large sets of data values. The reason why it runs much faster than direct evaluation is that recursion significantly reduces the number of mathematical calculations required and the number of operations required does not grow as fast as the former method as larger and larger data sets are introduced. Recursion also explains its algorithmic complexity of $O(n \log_2 n)$, where the factor n corresponds to the fact that each set of polynomials with the same degree needs to be evaluated n times and $\log_2 n$ corresponds to the number of subproblems we would need to get a linear polynomial from splitting a polynomial of degree $n - 1$.

Section 9 - Radix-2 Fast Fourier Transform Procedure

Let:

$$\omega = \exp\left(\frac{-2\pi i}{n}\right)$$

and a value $u_j = U(f_j)$ of the Discrete Fourier Transform at frequency $f = f_j$ be:

$$u_j = \sum_{k=0}^{n-1} y_k \, \omega^{jk}$$

Then the Discrete Fourier Transform $\boldsymbol{u} = (u_0, u_1, u_2, \dots, u_{n-1})$ of a vector $\mathbf{y} = (y_0, y_1, y_2, \dots, y_{n-1})$ is equivalent to multiplying $\mathbf{y}$ by the nth order Fourier Matrix F_n:

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & (\omega^2)^2 & \cdots & (\omega^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & (\omega^{n-1})^2 & \cdots & (\omega^{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\mathbf{u} = F_n \mathbf{y}$$

wherein each term u_j of $\mathbf{u}$ can be interpreted as evaluating the polynomial

$$P(z) = y_0 + y_1 z + y_2 z^2 + \cdots + y_{n-1} z^{n-1}$$

at $z = \omega^j$. Thus $\mathbf{u}$ contains the function values $u_0, u_1, u_2, u_3, \dots, u_{n-1}$ of $P(z)$ at $z = 1, \omega, \omega^2, \omega^3, \dots, \omega^{n-1}$, respectively.

If n is a power of 2, we can use the following recursive algorithm, called the ***Radix-2 Fast Fourier Transform***:

Suppose we wish to evaluate $P(z)$ at paired points $z_0, -z_0, z_1, -z_1, z_2, -z_2, \dots, z_m, -z_m$, where $m = \frac{n}{2} - 1$:

1. Let $P(z)$ be the following $(n-1)$th degree polynomial:

$$P(z) = y_0 + y_1 z + y_2 z^2 + \cdots + y_{n-1} z^{n-1}$$

2. If $n = 1$, then return $P(z)$, which is simply the constant y_0 (***Base Case***).

3. Define the following mth degree polynomials $Q_1(z)$ and $Q_2(z)$ whose coefficients correspond to the even-power and odd-power terms of $P(z)$, respectively:

$$Q_1(z) = y_0 + y_2 z + y_4 z^2 + \cdots + y_{n-2}\, z^m$$

$$Q_2(z) = y_1 + y_3 z + y_5 z^2 + \cdots + y_{n-1}\, z^m$$

4. Evaluate both $Q_1(s)$ and $Q_2(s)$ at $s = (z_0)^2, (z_1)^2, (z_2)^2, \dots, (z_m)^2$. If the values for s form a set of paired values, then perform the algorithm again for both polynomials from step 1, with the new z referring to the old s. Otherwise, evaluate the polynomials $Q_1(z)$ and $Q_2(z)$ directly (***Recursive Case***).

5. Then for all z_j where $j = 0, 1, 2, \dots, m$, compute:

$$P(z_j) = Q_1\left(\left(z_j\right)^2\right) + z_j\, Q_2\left(\left(z_j\right)^2\right)$$

$$P(-z_j) = Q_1\left((z_j)^2\right) - z_j\, Q_2\left((z_j)^2\right)$$

6. Return the function values $P(z_j)$ and $P(-z_j)$ for all z_j.

In computing the DFT using the FFT, we initially set $z_0 = 1,\ \ z_1 = \omega,\ \ \ z_2 = \omega^2,\ \ \dots,\ z_{n-1} = \omega^{n-1}$ for the main problem. Subproblems created from the recursion step will take some of these initial values every second, fourth, eighth, … values beginning with $z_0 = 1$.

Section 10 - Deriving Algebraically the Recursive Property of Radix-2 FFT

Recall the following relationship from Theorem 4:

$$\omega^k = \omega^{k-mn}, \quad m \in Z$$

where:

$$\omega = \exp\left(-\frac{2\pi i}{n}\right)$$

Then, assuming n is a power of 2 and therefore is even, we can divide the computation for the DFT value u_j into two parts, one on odd-numbered values and the other on even-numbered values:

$$u_j = \sum_{k=0}^{n-1} y_k \, \omega^{jk}$$

$$u_j = y_0\omega^{0j} + y_1\omega^{1j} + y_2\omega^{2j} + \cdots + y_{n-1}\omega^{(n-1)j}$$

$$u_j = \left(y_0\omega^{0j} + y_2\omega^{2j} + \cdots + y_{n-2}\omega^{(n-2)j}\right) + \left(y_1\omega^{1j} + y_3\omega^{3j} + \cdots + y_{n-1}\omega^{(n-1)j}\right)$$

$$u_j = \sum_{k=0}^{\frac{n}{2}-1} y_{2k} \, \omega^{2kj} + \sum_{k=0}^{\frac{n}{2}-1} y_{2k+1} \, \omega^{(2k+1)j}$$

$$u_j = \sum_{k=0}^{\frac{n}{2}-1} y_{2k} \, \omega^{2kj} + \omega^j \sum_{k=0}^{\frac{n}{2}-1} y_{2k+1} \, \omega^{2kj}$$

$$u_j = \sum_{k=0}^{\frac{n}{2}-1} y_{2k} \, (\omega^2)^{kj} + \omega^j \sum_{k=0}^{\frac{n}{2}-1} y_{2k+1} \, (\omega^2)^{kj}$$

Let ϕ be worth two unit steps or $\omega * \omega = \omega^2$:

$$\phi = \omega^2 = \exp\left(-\frac{2\pi i}{\left(\frac{n}{2}\right)}\right)$$

Then we can rewrite u_j as:

$$u_j = \sum_{k=0}^{\frac{n}{2}-1} y_{2k} \, \phi^{kj} + \omega^j \sum_{k=0}^{\frac{n}{2}-1} y_{2k+1} \, \phi^{kj}$$

Notice that each of the two sums is the value of the DFT for the even and odd values, respectively, but with half the data values as the original DFT.

This means that we can split a single DFT sum into more numerous but easier-to-compute sums, as written in the following pattern / pseudo-formula:

$$\mathrm{DFT}_j(\text{all terms}) = \mathrm{DFT}_j(\text{even terms}) + \omega^j \, \mathrm{DFT}_j(\text{odd terms})$$

As before, this ***divide-and-conquer approach*** describes how the FFT works to compute the DFT and explains why it has a time complexity of $O(n \log n)$ in contrast to the $O(n^2)$ of directly computing the DFT.

Section 11 - Inverse Discrete Fourier Transform (IDFT) and Inverse Fast Fourier Transform (IFFT)

Just as there is a way to transform a function from the time domain to the frequency domain, there is also a notion for the ***Inverse Fourier Transform*** (IFT) for transforming a function in the frequency domain back to the time domain. Using the IFT, we can reconstruct sets of information that have been processed in the frequency domain, and this technique is utilized in solving differential equations, noise reduction, and image compression.

The ***Inverse Continuous Fourier Transform*** (ICFT) is written very similarly to the CFT, but with the frequency f as the variable being integrated from and the exponential having a positive power instead:

$$\text{IFT:} \qquad Y(x) = \mathcal{F}^{-1}\{U(f)\} = \int_{-\infty}^{\infty} U(f)\exp(2\pi i f x)\ df$$

We can also compute the ***Inverse Discrete Fourier Transform*** (IDFT) in a similar way to the DFT. Recall that the IDFT can be expressed as the following linear system:

$$\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\mathbf{u} = F_n \mathbf{y}$$

which may be interpreted as evaluating the polynomial

$$P(z) = y_0 + y_1 z + y_2 z^2 + \cdots + y_{n-1} z^{n-1}$$

at $z = 1, \omega, \omega^2, \omega^3, \ldots, \omega^{n-1}$, of which are complex numbers spaced around the unit circle.

Since F_n is a square matrix, we can write the following expression for the IDFT if we can show that F_n is invertible:

$$\mathbf{y} = (F_n)^{-1}\mathbf{u}$$

How can we show that F_n is invertible, and what is its inverse given that n is a power of 2? Let's take F_4 as an example:

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix}$$

First, note that since F_4 is a complex matrix, we will have to use the Complex Euclidean Inner Product or Hermitian Inner Product to test for orthogonality. This expression is defined as follows for any two vectors $\mathbf{u}$ and $\mathbf{v}$ in the space $\mathbb{C}^n$ of complex-valued vectors:

$$\mathbf{u} \cdot \mathbf{v} = \bar{\mathbf{v}}\mathbf{u} = u_1\overline{v_1} + u_2\overline{v_2} + \cdots + u_n\overline{v_n}$$

where $\overline{v_j}$ is the complex conjugate of v_j. Recall that the complex conjugate of a complex number plotted on the complex plane is simply its reflection about the real axis. This fact is also geometrically evident as conjugation negates only the imaginary component. Thus, the complex conjugate of ω^k can be written according to the following theorem:

Theorem 5 – Conjugate of a Root of Unity

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$*. Then for all integers* n *and* k*:*

$$\overline{\omega^k} = \omega^{n-k} = \omega^{-k}$$

Proof for Theorem 5:

$$\overline{\omega^k} = \overline{\exp\left(-\frac{2\pi\imath k}{n}\right)}$$

$$\overline{\omega^k} = \overline{\cos\left(-\frac{2\pi k}{n}\right) + \imath\sin\left(-\frac{2\pi k}{n}\right)}$$

$$\overline{\omega^k} = \cos\left(-\frac{2\pi k}{n}\right) - i\sin\left(-\frac{2\pi k}{n}\right)$$

$$\overline{\omega^k} = \cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right)$$

$$\overline{\omega^k} = \exp\left(\frac{2\pi ik}{n}\right)$$

$$\overline{\omega^k} = \omega^{-k} = \omega^{n-k}$$

In the last step, we used the fact that $\omega^k = \omega^{n+k}$ from Theorem 4.

Note that while the dot product and the integral inner product satisfy the symmetry property, that is, $\langle\mathbf{u},\mathbf{v}\rangle = \langle\mathbf{v},\mathbf{u}\rangle$, the Hermitian Inner Product does not. In fact, it can be shown that switching the order of the vectors negates the value of the Hermitian inner product:

$$\mathbf{u}\cdot\mathbf{v} = -(\mathbf{v}\cdot\mathbf{u})$$

Now we will show that every column in F_4 is orthogonal to any other column in F_4 using the Hermitian Inner Product. In other words, we are trying to show that F_4 is an ***orthogonal matrix***. Let $\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}$, and $\mathbf{c_3}$ be the columns in F_4. We can show that the inner products between $\mathbf{c_0}$ and any column in F_4 are:

$$\mathbf{c_0}\cdot\mathbf{c_0} = 1+1+1+1 = 4$$

$$\mathbf{c_0}\cdot\mathbf{c_1} = 1+\omega^{-1}+\omega^{-2}+\omega^{-3} = 1+\omega^3+\omega^2+\omega = 1+(-\omega)+(-1)+\omega = 0$$

$$\mathbf{c_0}\cdot\mathbf{c_2} = 1+\omega^{-2}+\omega^{-4}+\omega^{-6} = 1+\omega^2+1+\omega^2 = 1+(-1)+1+(-1) = 0$$

$$\mathbf{c_0}\cdot\mathbf{c_3} = 1+\omega^{-3}+\omega^{-6}+\omega^{-9} = 1+\omega+\omega^2+\omega^3 = 1+\omega+(-1)+(-\omega) = 0$$

We can also show the inner products between $\mathbf{c_1}$ and any column in F_4 as:

$$\mathbf{c_1}\cdot\mathbf{c_0} = 1+\omega+\omega^2+\omega^3 = 1+\omega+(-1)+(-\omega) = 0$$

$$\mathbf{c_1}\cdot\mathbf{c_1} = (1)(1)+(\omega)(\omega^{-1})+(\omega^2)(\omega^{-2})+(\omega^3)(\omega^{-3}) = 1+1+1+1 = 4$$

$$\mathbf{c_1}\cdot\mathbf{c_2} = (1)(1)+(\omega)(\omega^{-2})+(\omega^2)(\omega^{-4})+(\omega^3)(\omega^{-6}) = 1+\omega^{-1}+\omega^{-2}+\omega^{-3} = 1+\omega^3+\omega^2+\omega$$
$$= 1+(-\omega)+(-1)+\omega = 0$$

$$\mathbf{c_1}\cdot\mathbf{c_3} = (1)(1)+(\omega)(\omega^{-3})+(\omega^2)(\omega^{-6})+(\omega^3)(\omega^{-9}) = 1+\omega^{-2}+\omega^{-4}+\omega^{-6} = 1+\omega^2+1+\omega^2$$
$$= 1+(-1)+1+(-1) = 0$$

The proofs for $\mathbf{c_2}$ and $\mathbf{c_3}$ are similar.

From these calculations, we can infer that if $i, j = 0, 1, 2, 3$:

$$\mathbf{c}_i \cdot \mathbf{c}_j = \begin{cases} 4, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

We can generalize this to any Fourier Matrix F_n. If $i, j = 0, 1, 2, \ldots, (n-1)$ and $\mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c}_{n-1}$ are the columns of F_n:

$$\mathbf{c}_i \cdot \mathbf{c}_j = \begin{cases} n, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

We call a matrix whose columns are orthogonal to every other column in the matrix an ***orthogonal matrix***, and F_n is one important instance of an orthogonal matrix.

Since F_n is also symmetric, the jth column vector $\mathbf{c}_j$ has the same values as the jth row vector $\mathbf{r}_j$. Therefore:

$$\mathbf{c}_i \cdot \mathbf{c}_j = \overline{\mathbf{r}_j}\mathbf{c}_i = \begin{cases} n, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

or if we switch the indices:

$$\overline{\mathbf{r}_i}\mathbf{c}_j = \begin{cases} n, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$$

Since matrix multiplication involves multiplying a row from one matrix by a column in another matrix, $\overline{\mathbf{r}_i}\mathbf{c}_j$ can be thought of as the (i, j)th entry of the matrix product $\overline{F_n}F_n$, where $\overline{F_n}$ is the matrix whose entries are complex conjugates of the corresponding entries in F_n. From the above expression, we can infer that:

$$\overline{F_n}F_n = \begin{bmatrix} n & 0 & \cdots & 0 \\ 0 & n & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & n \end{bmatrix}$$

$$\overline{F_n}F_n = nI$$

$$\left(\frac{1}{n}\overline{F_n}\right)F_n = I$$

Therefore, F_n is invertible and

$$F_n^{-1} = \frac{1}{n}\overline{F_n}$$

$$F_n^{-1} = \frac{1}{n}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix}$$

Thus, the equation for the IDFT can be rewritten as:

$$\mathbf{y} = \left(\frac{1}{n}\overline{F_n}\right)\mathbf{u}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \frac{1}{n}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix}\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{n-1} \end{bmatrix}$$

which we refer to as the ***matrix interpretation of the Inverse DFT***.

The above linear system may be interpreted as finding the coefficients $y_0, y_1, y_2, \ldots, y_{n-1}$ of $P(z)$ such that the values of $P(z)$ at $z = 1, \omega, \omega^2, \ldots, \omega^{n-1}$ are $u_0, u_1, u_2, \ldots, u_{n-1}$, respectively. In other words, we are finding the coefficients of a polynomial which interpolates the n points $(1, u_0)$, (ω, u_1), (ω^2, u_2), … , (ω^{n-1}, u_{n-1}). This is the ***polynomial interpretation of the Inverse DFT***.

We can also write the IDFT as the following sum:

$$y_k = \frac{1}{n}\sum_{j=0}^{n-1} u_j \omega^{-jk}$$

More formally, the IDFT can be written and defined as:

$$Y(x_k) = \frac{1}{n}\left(\sum_{j=0}^{n-1} U(f_j)\omega^{-jk}\right), \qquad \omega = \exp\left(-\frac{2\pi i}{n}\right)$$

Or in terms of the sampled frequencies $f_j = j$ and sampled values $x_k = \frac{k}{n}$:

$$Y(x_k) = \frac{1}{n}\left(\sum_{j=0}^{n-1} U(f_j)\exp\left(-2\pi i * \frac{-jk}{n}\right)\right)$$

IDFT: $$Y(x_k) = \frac{1}{n}\left(\sum_{j=0}^{n-1} U(f_j)\exp(2\pi i f_j x_k)\right)$$

The algorithm for computing the IDFT using the FFT is referred to as the ***Inverse Fast Fourier Transform*** (IFFT). The procedure for the ***Radix-2 IFFT*** is almost identical to the Radix-2 FFT but with two key differences:

a. We define ω in code with a positive exponent instead, while using the FFT algorithm

b. After the FFT and its recursions are performed, we divide all its output values by n.

We also refer to $U(f)$ as the set of inputs and $Y(x)$ as the set of outputs for the IFFT, though this change does not affect the values of the calculation either way.

Below is a Python Implementation of the Radix-2 IFFT (credits to Reducible):

```
def IFFT_calc(U: list):
    """Calculates the inverse discrete fourier transform of U recursively, save for the last step of dividing each output by the number of outputs,
      using a slightly modified version of the Fast Fourier Transform (FFT) algorithm."""


    n = len(U)

    if n == 1:
        return U
    w = rect(1, 2*pi/n)
    Ueven, Uodd = U[::2], U[1::2]
    Yeven, Yodd = IFFT_calc(Ueven), IFFT_calc(Uodd)
    Y = [0]*n

    # loop through each of the n//2 pairs of n complex inputs around the unit circle
    for k in range(n//2):
        evenPart = Yeven[k]
        oddPart = w**k * Yodd[k]

        # A property of positive-negative pairs allows us to get two function values by simply changing one operation.
        Y[k] = evenPart + oddPart
        Y[k + n//2] = evenPart - oddPart

        # Here mathematically, Y[k + n//2] is equivalent to Y[-k], since negating a complex number on the unit circle corresponds to rotating it
        # by pi radians or 180 degrees about the circle

    return Y



def IFFT(U: list):
    """Computes the discrete inverse fourier transform at each value of U given that U has a length "n" equivalent to a power of 2 (so n = 1, 2, 4,
    8, 16, 32, and so on)."""

    # check if the length of U is a power of 2
    n = len(U)
    isPowerof2 = True
    quotient = n
    while quotient > 1:
        quotient, remainder = quotient // 2, quotient % 2
        if remainder != 0:
```

Here is a link to a Python Implementation of the IFFT:

[the-pythons-nest/func_IFFT.py at main · NotAMadTheorist/the-pythons-nest (github.com)]()

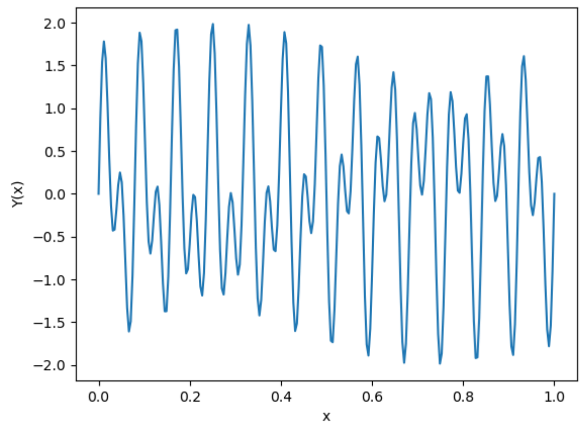
Section 12 - Back-and-Forth Example Using the FFT and IFFT

By applying the FFT and IFFT, signal data corresponding to the time domain may be reconstructed from the values of the frequency domain.

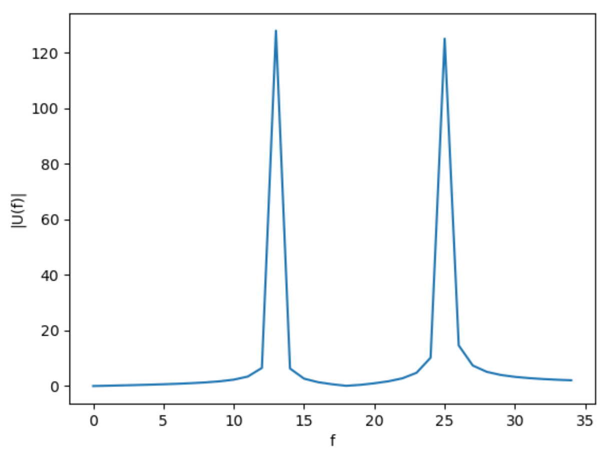
Say that we have a signal composed of two sinusoidal waves: one at 13 Hz and another at 25 Hz. The signal's value within the time domain $[0, 1]$ is given by:

$$Y(x) = \sin(26\pi x) + \sin(50\pi x)$$

Shown below is the graph of $Y(x)$ on $[0, 1]$:

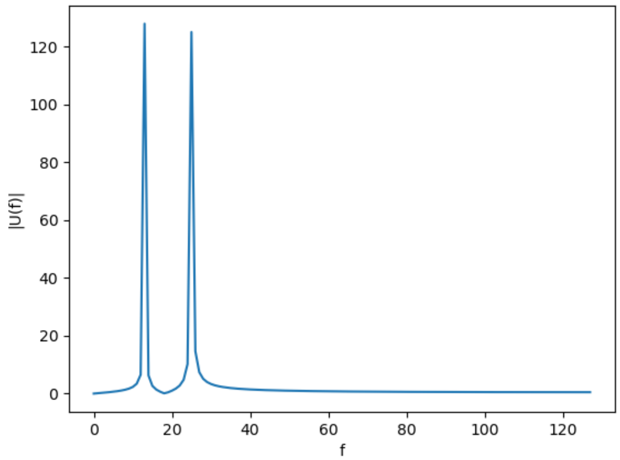


Suppose we take $n = 2^8 = 256$ values from [0,1]. Then we can compute the discrete fourier transform $U(f) = \mathcal{F}(Y(x))$ using the FFT algorithm. However, since $U(f)$ has complex values, we cannot plot $U(f)$ in two dimensions, although its magnitude $|U(f)|$ can be plotted instead, resulting in what is called a ***power spectrum***:

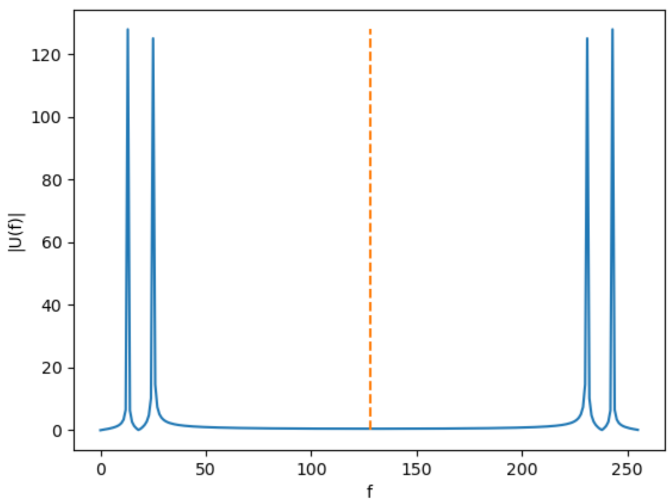


Notice that the power spectrum above considers frequencies that are below 35 Hz. This is done for clarity, as we expect "spikes" in the DFT value at the detected frequencies $f = 13$ Hz and $f = 25$ Hz, which is as observed.

Although there are $n = 256$ DFT values, only the first half of DFT values corresponding to frequencies from 0 Hz to just below the ***Nyquist frequency*** $f_N = \frac{s}{2}$ which is always half of the sampling rate, convey usable information about the signal. Since the time domain is $[0, 1]$, $s = n = 256$ and we have $f_N = 128$, thus the power spectrum may be enlarged to a frequency domain of $[0, 128)$:

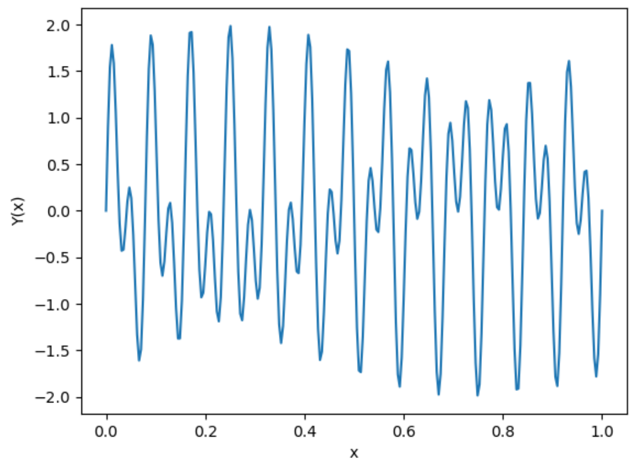


If we were to plot the magnitudes of all 256 DFT values, then the second half of the plot is simply a reflection of the first half about the vertical dashed line corresponding to the Nyquist frequency:



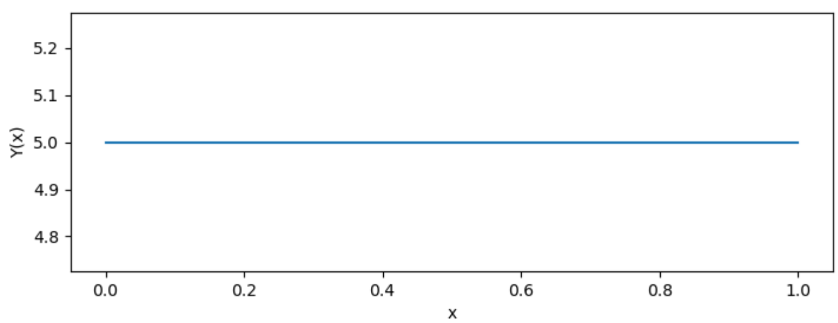
Clearly, the second half of the DFT values do not convey information about the signal's actual frequency components, as expected by the Nyquist-Shannon Sampling Theorem. However, they are still DFT values and thus the frequency domain is still $[0, 256)$.

We can also reconstruct the original signal $Y(x)$ from its component frequencies by applying the IFFT on all 256 complex DFT values. As shown below, the plot of the reconstructed signal is identical to the original signal:

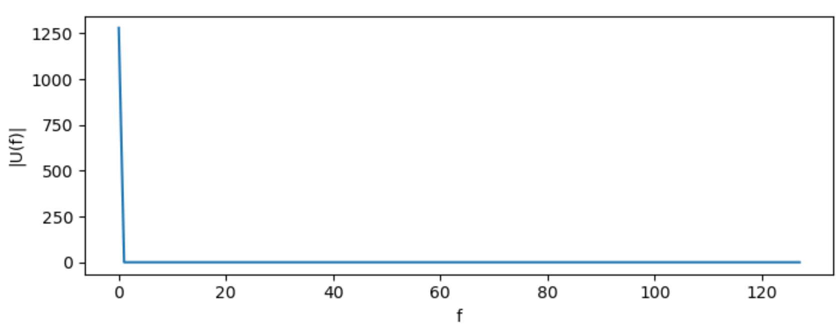


Note that both the first half and second half of DFT values are needed in order to reconstruct the original signal on the time domain $[0, 1]$, even though the peaks in the second half do not correspond to the actual frequency components of the signal. If we were to take only the first half, then we would only have 128 sample points instead of the original 256 sample points for the reconstructed signal.

To find what the DFT is for a constant function, suppose that $Y(x) = 5$, whose plot is shown below:



Then its power spectrum would consist of only one spike with height of $|U(f)| = 1280 = 5n$ at $f = 0$ Hz and is zero everywhere else:



We can confirm this statement mathematically by substituting in $Y(x_j) = 5$, where $j = 0, 1, 2, \dots, 255$, to the linear system for computing the DFT:

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{255}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{255} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{510} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{255} & \omega^{510} & \cdots & \omega^{65025} \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 5 \\ \vdots \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} U(0) \\ U(1) \\ U(2) \\ \vdots \\ U(255) \end{bmatrix} = \begin{bmatrix} 5 * 256 \\ 5(1 + \omega + \omega^2 + \cdots + \omega^{255}) \\ 5(1 + \omega^2 + \omega^4 + \cdots + \omega^{510}) \\ \vdots \\ 5(1 + \omega^{255} + \omega^{510} + \cdots + \omega^{65025}) \end{bmatrix}$$

$$\begin{bmatrix} U(0) \\ U(1) \\ U(2) \\ \vdots \\ U(255) \end{bmatrix} = \begin{bmatrix} 1280 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which confirms that $U(0) = 1280 = 5n$ and $U(k) = 0,\ \ k = 1, 2, 3, \dots, 255$.

In fact, this may be generalized and formally stated as the following theorem:

Theorem 6 – DFT of a Constant Function

If $Y(x) = c$, where c is a constant, and $U(f)$ is the Discrete Fourier Transform of $Y(x)$ at n samples, then:

$$U(0) = cn, \quad U(k) = 0, \quad \text{where } k = 1, 2, 3, \dots, n-1.$$

Section 13 - Sampling above the Nyquist Frequency

More Info: [Circles Sines and Signals - Sine Wave Aliasing (jackschaedler.github.io)]

[Circles Sines and Signals - Nyquist Frequency (jackschaedler.github.io)]

As observed from the previous example, the magnitudes of the DFT at frequencies above the Nyquist frequency are just reflections of those below the Nyquist frequency.

To answer why this phenomenon occurs, we will need to recall from Section 4 that the DFT can be interpreted as a complex number whose components are dot products of the original values and sinusoidal bases:

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k) \exp(-2\pi i f_j x_k)$$

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k)\left(\cos(-2\pi f_j x_k) + i \sin(-2\pi f_j x_k)\right)$$

$$U(f_j) = \left(\sum_{k=0}^{n-1} Y(x_k) \cos(-2\pi f_j x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k) \sin(-2\pi f_j x_k)\right)$$

$$U(f_j) = (\mathbf{y} \cdot \mathbf{c}) + i(\mathbf{y} \cdot \mathbf{s})$$

where $\mathbf{y}$, $\mathbf{c}$, and $\mathbf{s}$ are vectors in $\mathbb{R}^n$ whose kth entries are the values of $Y(x)$, $\cos(-2\pi f_j x)$, and $\sin(-2\pi f_j x)$, respectively, evaluated at $x = x_k = \frac{k}{n}$, where $k = 0, 1, 2, \ldots, n-1$.

Thus, the DFT at any frequency also samples $\cos(-2\pi f_j x)$ and $\sin(-2\pi f_j x)$ at n values $x_0, x_1, x_2, \ldots, x_{n-1}$. Notice that the values of the cosine and sine at these n values only form n spaced points and thus do not completely identify the graph of the original cosine or sine wave of frequency f_j. This suggests that there are infinitely many cosine or sine waves that can interpolate through these n points but with frequencies than f_j, and this is in fact true.

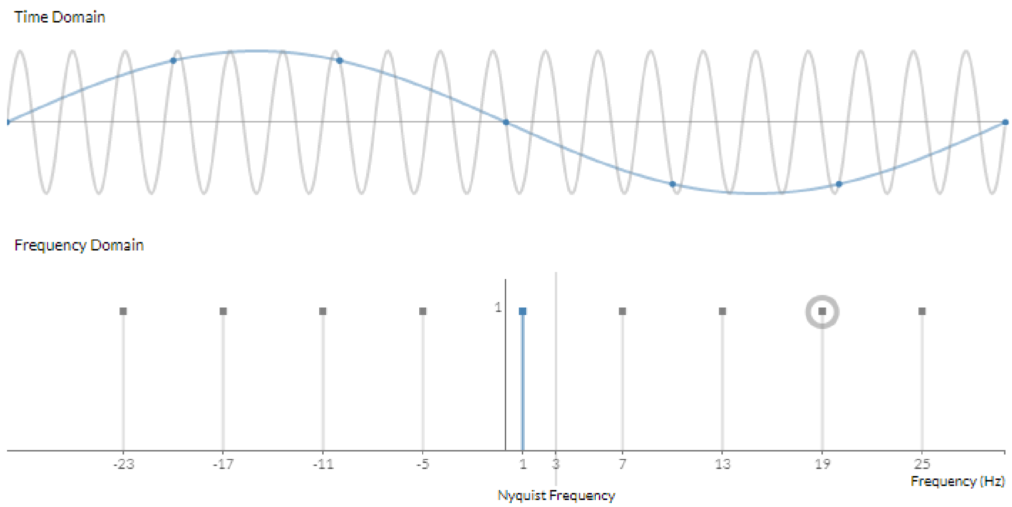
We state this more formally in the following theorem:

Theorem 7 – Sampled Sine Wave Theorem

Given a sampling rate of s Hz and any integer m, a sine wave at a frequency f Hz is indistinguishable from a sine wave of frequency $f + ms$ Hz after sampling.

For example, sampling a 1 Hz sine wave at a fixed sampling rate of 6 Hz gives the same sampled points as if the sine wave had frequencies 7 Hz, 13 Hz, 19 Hz, … or frequencies of -5 Hz, -11 Hz, -17 Hz and so on. Note that since $\sin(-\theta) = -\sin(\theta)$, a sine wave of a negative frequency $-f$ is the same as a sine wave of frequency f but with each output value flipped in sign.

The diagram below shows the sample points of a 1 Hz sine wave (in blue) sampled at 6 Hz are also samples of a 19 Hz sine wave (in gray):



Proof of Theorem 7:

Suppose $Y(x) = \sin(-2\pi f_0 x)$ at $x = x_k = \frac{k}{s}, \quad k = 0, 1, 2, \dots, s$, where s is the sampling rate and the number of samples taken. Let $Y_m(x) = \sin(-2\pi f_m x)$, where m is any integer and $f_m = f_0 + ms$. Then:

$$Y_m(x_k) = \sin(-2\pi(f_0 + ms)x_k)$$

$$Y_m(x_k) = \sin(-2\pi f_0 x_k - 2\pi m s x_k)$$

$$Y_m(x_k) = \sin(-2\pi f_0 x_k - 2\pi m k)$$

However, since m and k are both integers, so is mk. Therefore, the term $-2\pi mk$ will always be an integer multiple of 2π and since sine has a period of 2π, we must have:

$$Y_m(x_k) = \sin(-2\pi f_0 x_k)$$

$$Y_m(x_k) = Y(x_k)$$

For cosine waves, the above theorem still holds and can be stated as the following:

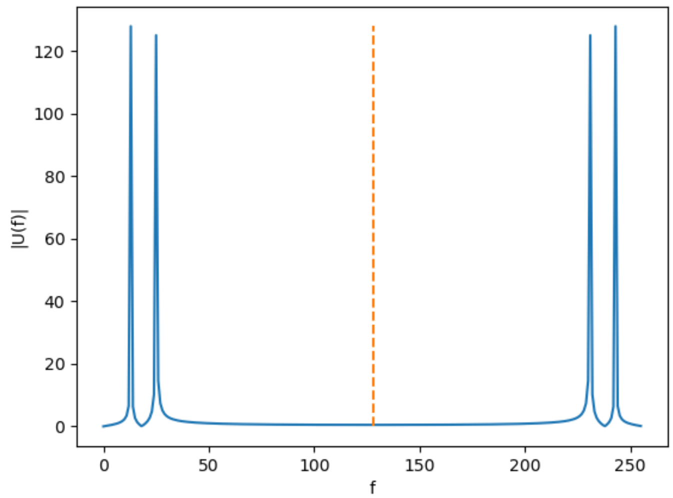
Theorem 8 – Sampled Cosine Wave Theorem

Given a sampling rate of s Hz and any integer m, a cosine wave at a frequency f Hz is indistinguishable from a cosine wave of frequency $f + ms$ Hz after sampling.

The proof for Theorem 8 is similar to that of Theorem 7.

An important difference to mark between cosine waves and sine waves is that negative frequencies do not affect the value of a cosine wave since $\cos(-\theta) = \cos\theta$. Thus, a cosine wave of any frequency f is identical to a cosine wave of frequency $-f$.

Now let us recall that the whole power spectrum from the previous example for the FFT and IFFT was symmetric about the Nyquist frequency, as marked by the dashed line at $f = f_N = 128$ Hz:



Suppose we evaluate $U(25)$ using the dot product formula for the DFT instead of the FFT. Then we can write $U(25)$ as:

$$U(25) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-50x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-50x_k)\right)$$

Although evaluating the above expression results in the same complex value and magnitude ($|U(25)| \approx 130$) as obtained by the FFT, we would have to sample $\cos(-50\pi x)$ and $\sin(-50\pi x)$, which both represent sine waves of frequency 25 Hz, at a sampling rate of $s = n = 256$ Hz.

According to Theorems 7 and 8, a sine wave of frequency 25 Hz is indistinguishable from a sine wave of frequency $25 - 256 = -231$ Hz, and a cosine wave at 25 Hz is indistinguishable from a -231 Hz after sampling. Moreover, a sine wave of frequency 231 Hz is simply the -231 Hz sine wave but with each value flipped in sign, while a cosine wave of frequency 231 Hz is identical to that of -231 Hz.

Therefore, the sampled values for the sine basis at $f = 231$ Hz are simply negatives of the sampled values for $f = 25$ Hz and the values for the cosine bases are the same at both frequencies. Written mathematically, we have:

$$\cos(-50\pi x_k) = \cos(-462\pi x_k)$$

$$\sin(-50\pi x_k) = -\sin(-462\pi x_k)$$

where $x_k = \frac{k}{n}$ and $k = 0, 1, 2, 3 \ldots, n-1$.

Thus, $U(231)$ can be expressed as:

$$U(231) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-462x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-462x_k)\right)$$

$$U(231) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-50x_k)\right) - i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-50x_k)\right)$$

$$U(231) = \overline{U(25)}$$

$$|U(231)| = |U(25)|$$

This explains why a peak occurs at $f = 231$ Hz that is identical to the peak at $f = 25$ Hz. Notice that both peaks are at the same distance, at 103 units, from the dashed line marking the Nyquist frequency $f_N = 128$ Hz.

This result also means that the complex value $U(231)$ is simply the complex conjugate of that of $U(25)$.

We can generalize the above result to any DFT at any frequency through the following theorem:

Theorem 9 – DFT of Frequencies above the Nyquist Frequency

Suppose $U(f)$ is the discrete Fourier transform of $Y(x)$ on the time domain $[0,1]$ with a sampling rate of n. Let $f_N = n/2$ be the Nyquist frequency. Then at frequencies $f = 0, 1, 2, \ldots, f_N - 1$ Hz below f_N:

$$U(f) = \overline{U(n-f)}$$

and:

$$|U(f)| = \overline{U(n-f)}$$

Proof for Theorem 9:

Suppose $f_j = 0, 1, 2, \ldots, f_N - 1$, where $f_N = n/2$, $j = 0, 1, 2, \ldots, f_N - 1$, and n is the sampling rate. Then according to the inner product interpretation of the Discrete Fourier Transform:

$$U(f_j) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-2\pi f_j x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-2\pi f_j x_k)\right)$$

The expression in each sum is evaluated at the sample points are given by $x = x_k = \frac{k}{n}$, where $k = 0, 1, 2, \ldots, n-1$.

According to Theorems 7 and 8:

$$\cos(-2\pi f_j x_k) = \cos(2\pi(f_j - n)x_k)$$

$$\sin(-2\pi f_j x_k) = \sin(2\pi(f_j - n)x_k)$$

From the trig identities $\sin(-\theta) = -\sin\theta$ and $\cos(-\theta) = \cos\theta$, we also have:

$$\cos(2\pi(f_j - n)x_k) = \cos(2\pi(n - f_j)x_k)$$

$$\sin(2\pi(f_j - n)x_k) = -\sin(2\pi(n - f_j)x_k)$$

Combining these results, we have:

$$\cos(-2\pi f_j x_k) = \cos(2\pi(n - f_j)x_k)$$

$$\sin(-2\pi f_j x_k) = -\sin(2\pi(n - f_j)x_k)$$

Therefore, the DFT at $f = n - f_j$ can be written as:

$$U(n - f_j) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-2\pi(n - f_j)x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-2\pi(n - f_j)x_k)\right)$$

$$U(n-f_j) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-2\pi f_j x_k)\right) - i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-2\pi f_j x_k)\right)$$

$$U(n-f_j) = \overline{U(f_j)}$$

which leads to the following results:

$$U(f_j) = \overline{U(n-f_j)}$$

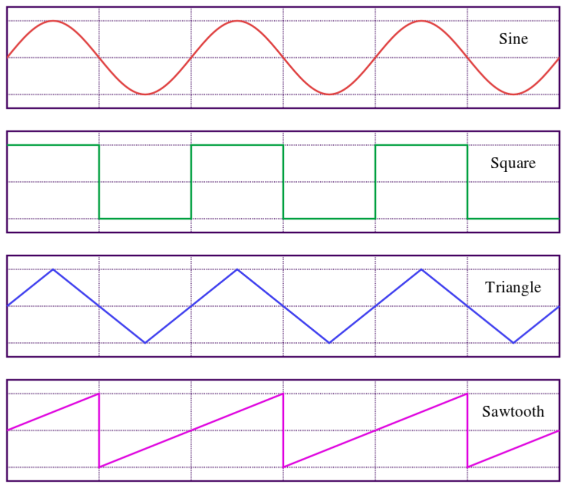
$$|U(f_j)| = |U(n-f_j)|$$

Theorem 7 explains why the DFT at frequencies above the Nyquist frequency do not convey information about the signal's actual frequency content.

Section 14 - DFT of Square Waves, Triangle Waves, and Sawtooth Waves

So far, we have been dealing with the DFTs of sines, cosines, and their sums. Pure sine and cosine signals produce a wavy shape, or a ***sinusoidal waveform*** when plotted in the time domain. However, there are also other pure signals encountered in practical applications, particularly electronics and electronic music, which do not have such shapes.

Three of the most common ***non-sinusoidal waveforms*** are the square wave, triangle wave, and sawtooth wave, whose waveforms are shown below:



A ***square wave*** with a frequency of f and an amplitude of A has the following signal value $Y(x) = \mathrm{sq}_{f,A}(x)$ at any value x in the time domain:

$$Y(x) = \mathrm{sq}_{f,A}(x) = \begin{cases} A, & \text{if } p < 1/2 \\ 0, & \text{if } p = 1/2 \\ -A, & \text{if } p > 1/2 \end{cases}, \qquad \text{where } p = fx \bmod 1$$

A ***triangle wave*** with a frequency of f and an amplitude of A has the following signal value $Y(x) = \mathrm{tr}_{f,A}(x)$ at any value x in the time domain:

$$Y(x) = \mathrm{tr}_{f,A}(x) = \begin{cases} 0, & \text{if } p = 0, 2 \\ pA, & \text{if } 0 < p < 1 \\ (2-p)A, & \text{if } 1 < p < 3 \\ (p-4)A, & \text{if } 3 < p < 4 \end{cases}, \qquad \text{where } p = 4(fx \bmod 1)$$

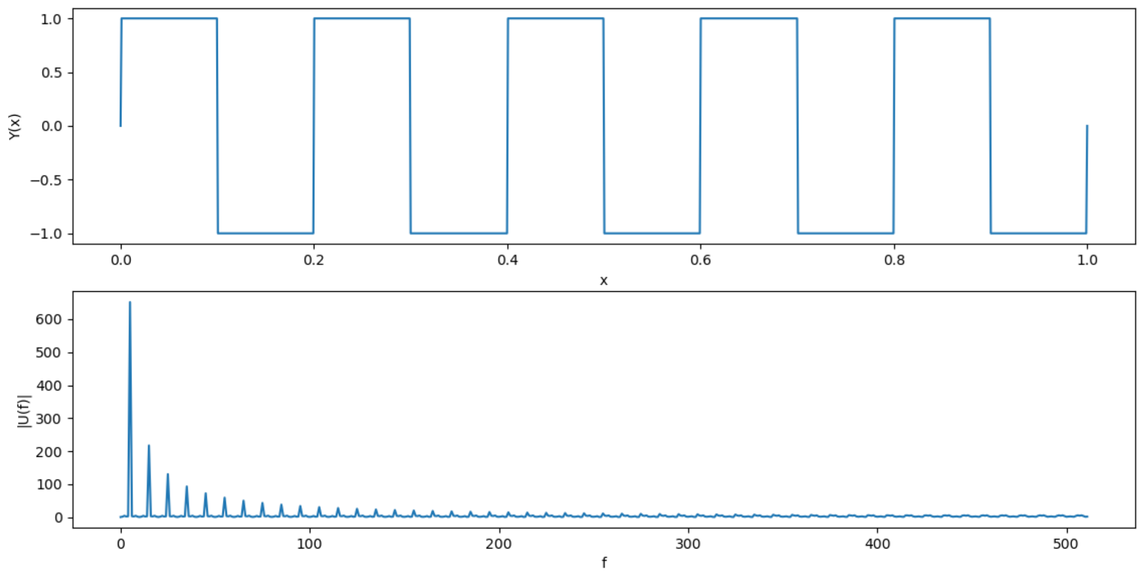
A ***rising sawtooth*** wave with a frequency f and an amplitude of A has the following signal value $Y(x) = \mathrm{swr}_{f,A}(x)$:

$$Y(x) = \mathrm{swr}_{f,A}(x) = \begin{cases} 0, & \text{if } p = 0 \\ (p-1)A, & \text{otherwise} \end{cases}, \qquad \text{where } p = 2(fx \bmod 1)$$

A ***falling sawtooth*** wave with a frequency f and an amplitude of A has the following signal value $Y(x) = \mathrm{swf}_{f,A}(x)$:

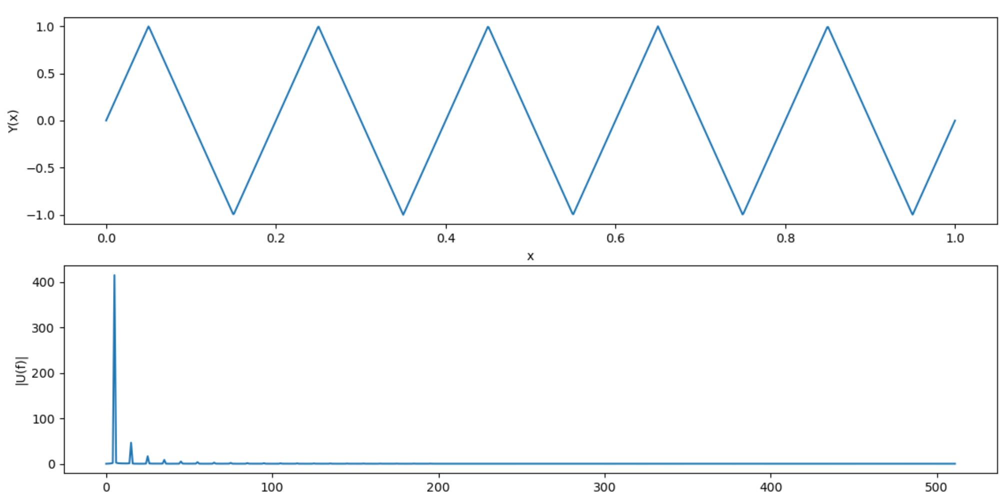
$$Y(x) = \mathrm{swf}_{f,A}(x) = \begin{cases} 0, & \text{if } p = 0 \\ (1-p)A, & \text{otherwise} \end{cases}, \qquad \text{where } p = 2(fx \bmod 1)$$

Suppose we have a square wave signal $Y(x) = sq_{5,1}(x)$ with frequency $f = 5$ Hz and amplitude of 1. If we take its DFT with a sample rate of $s = n = 512$ Hz using the FFT, then the power spectrum of the signal shows successive spikes with falling magnitudes:



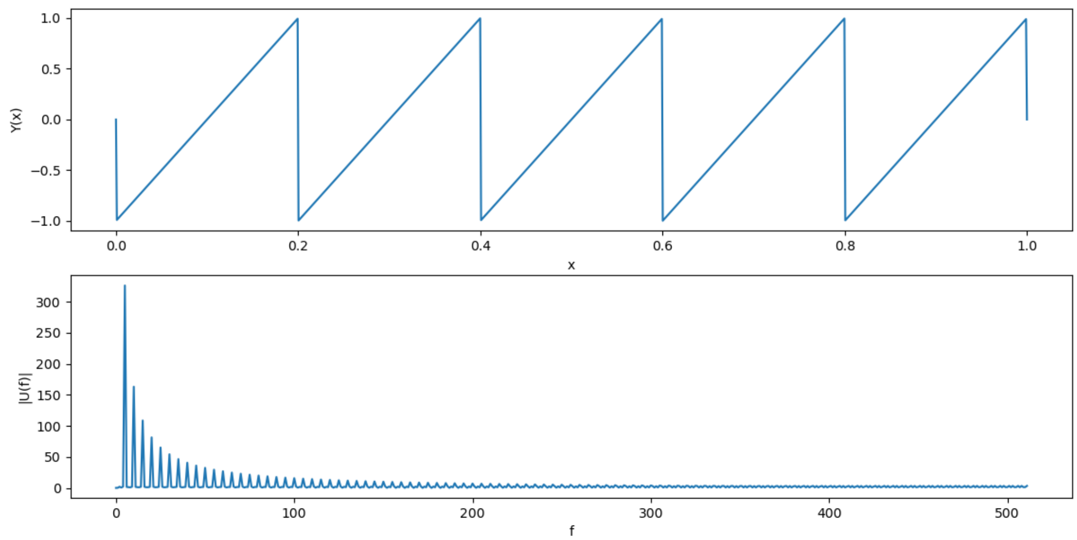
The spikes occur at $f = 5, 15, 25, 35, 45, \ldots 5(2k-1), \ldots$ Hz, where $k = 1, 2, 3, \ldots$, and the magnitudes of the spikes relative to the magnitude of the maximum spike are $M = 1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}, \ldots \frac{1}{1+2k}, \ldots$. Thus, all spike frequencies are odd-numbered multiples of the ***fundamental frequency*** $f = 5$ Hz. Both these patterns apply to all square waves and this can be mathematically proven by using the Fourier series, as we will see in the next chapter.

Now suppose our signal is a triangle wave $Y(x) = \mathrm{tr}_{5,1}(x)$ with frequency $f = 5$ Hz and amplitude of 1. Its power spectrum when $n = 256$ is shown below:



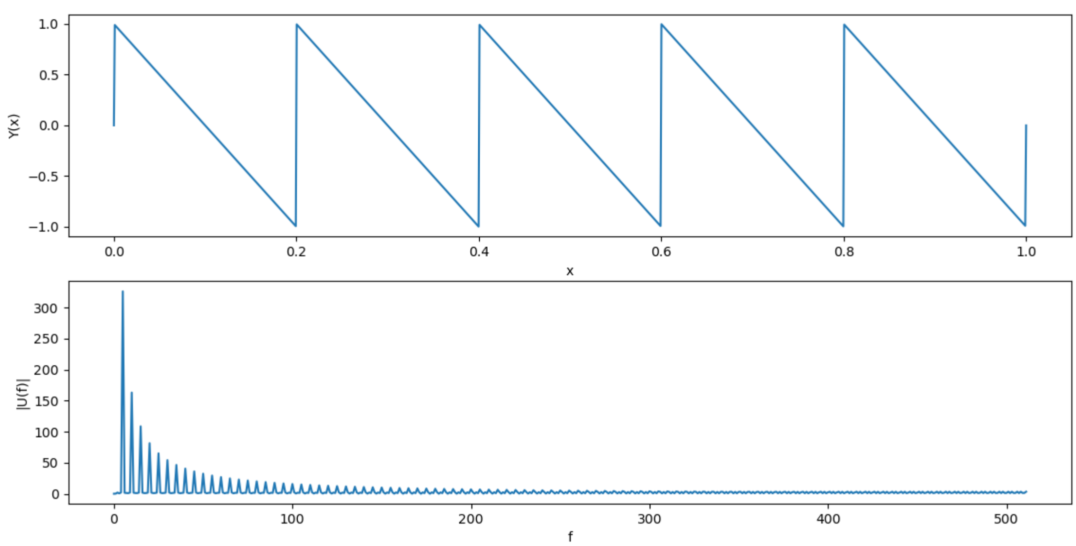
The spikes occur at frequencies $f = 5, 15, 25, \ldots, 5(2k-1), \ldots$ Hz, where $k = 1, 2, 3 \ldots$, and the relative magnitudes of the spikes are $M = 1, \frac{1}{9}, \frac{1}{25}, \ldots, \frac{1}{(2k-1)^2}, \ldots$. Like in the square wave, the spike frequencies are odd-numbered multiples of the fundamental frequency $f = 5$ Hz, but the magnitudes of the spikes fall much more quickly at higher frequencies, thus triangle waves have much less high frequency content.

The power spectrum for a rising sawtooth wave $Y(x) = \mathrm{swr}_{5,1}(x)$ with frequency $f = 5$ Hz and amplitude 1 consists of spikes that are much closer than that of the square and triangle waves:

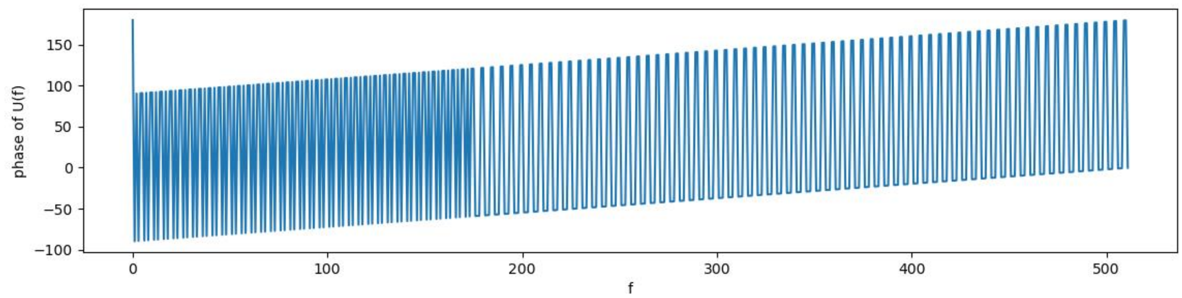


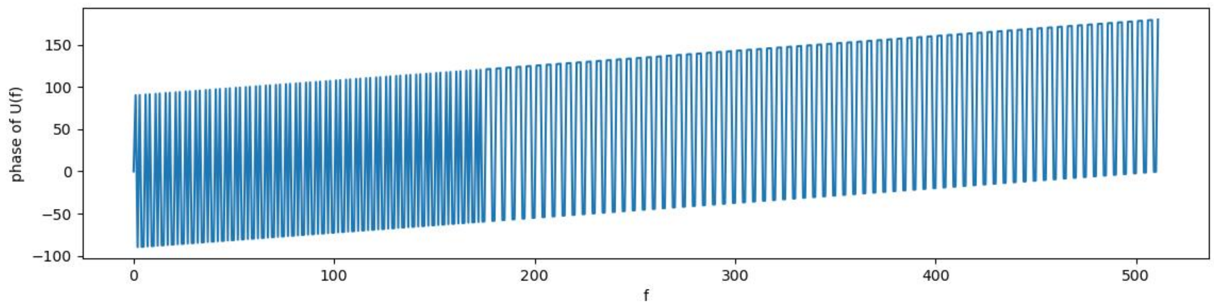
The spikes occur at the frequencies $f = 5, 10, 15, 20, \ldots, 5k, \ldots$ Hz, where $k = 1, 2, 3, 4, \ldots$, and the relative magnitudes of the spikes are $M = 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{k}, \ldots$. In this case, the spike frequencies are integer multiples of the fundamental frequency $f = 5$ Hz.

The power spectrum of a falling sawtooth wave at the same frequency and amplitude is identical to that of the rising sawtooth wave, despite being a different waveform:



The reason for the difference in waveforms is the difference in the phases of the complex DFT values between rising sawtooth waves and falling sawtooth waves. The diagram below shows the phases (in degrees) of the DFT values for the rising sawtooth wave (top) and the falling sawtooth wave (bottom):





Notice that the phases form a similar pattern, but the high and low phases alternate differently between the waveforms, resulting in the rising and falling patterns in the waveforms themselves.

In general, the frequency content of square waves, triangle waves, and sawtooth waves is composed of an infinite number of discrete frequency values at different magnitudes, which corresponds to the "detected" frequencies or spikes in their respective power spectra. Below summarizes the spike frequencies and relative magnitudes for each of the waveforms:

Waveform	Spike Frequencies	Relative Magnitudes
Square	$f = f_1,\ \ 3f_1,\ \ 5f_1, \dots,\ \ (2k+1)f_1, \dots$	$M = 1, \frac{1}{3}, \frac{1}{5}, \dots, \frac{1}{2k+1}, \dots$
Triangle	$f = f_1,\ \ 3f_1,\ \ 5f_1, \dots,\ \ (2k+1)f_1, \dots$	$M = 1, \frac{1}{9}, \frac{1}{25}, \dots, \frac{1}{(2k+1)^2}, \dots$
Rising Sawtooth	$f = f_1, 2f_1, 3f_1, \dots, kf_1, \dots$	$M = 1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{k}, \dots$
Falling Sawtooth	$f = f_1, 2f_1, 3f_1, \dots, kf_1, \dots$	$M = 1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{k}, \dots$

Section 15 - DFT of Periodic Functions

Notice that the last section concluded that square waves, triangle waves, and sawtooth waves, all of which are periodic functions, with integer frequencies had power spectra that consisted only of spikes at integer frequencies, including the frequency of the wave. This was due to the fact that the function repeated itself a sufficient number of times (> 8 times) within the time domain $[0, 1]$. If the function did not repeat, the power spectrum wouldn't have these visible spikes and it would be distributed across the frequency domain.

Thus, we state the following conjecture:

Conjecture 10 - DFT of Periodic Functions:

All periodic functions which repeat at sufficiently large integer frequencies have power spectra that consist only of spikes at multiples of a certain frequency, called the ***fundamental frequency****, of the original function.*

To demonstrate this, suppose we have the following periodic functions in the time domain, where $s = n = 1024$, the fundamental frequency is $f = 15$ Hz and the period is $T = \frac{1}{f}$:

$$Y_1(x) = -p^2 + 2p + 4, \qquad \text{where } \; p = 5\left(\frac{x \bmod T}{T}\right)$$

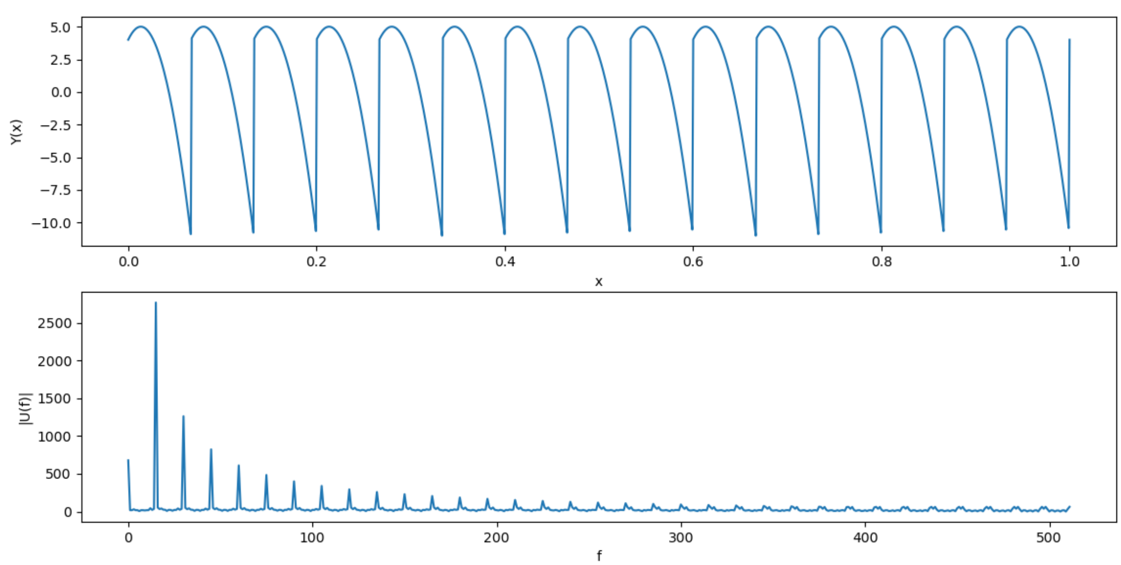
$$Y_2(x) = p^3 - 6p^2 + 9p - 2, \qquad \text{where } \; p = 4\left(\frac{x \bmod T}{T}\right)$$

$$Y_3(x) = 5e^{-p} * \sin^2 p \qquad \text{where } \; p = \pi\left(\frac{x \bmod T}{T}\right)$$

$$Y_4(x) = \mathrm{sq}_{1,1}(p) + \mathrm{swr}_{3,1}(p), \qquad \text{where } \; p = \frac{x \bmod T}{T}$$

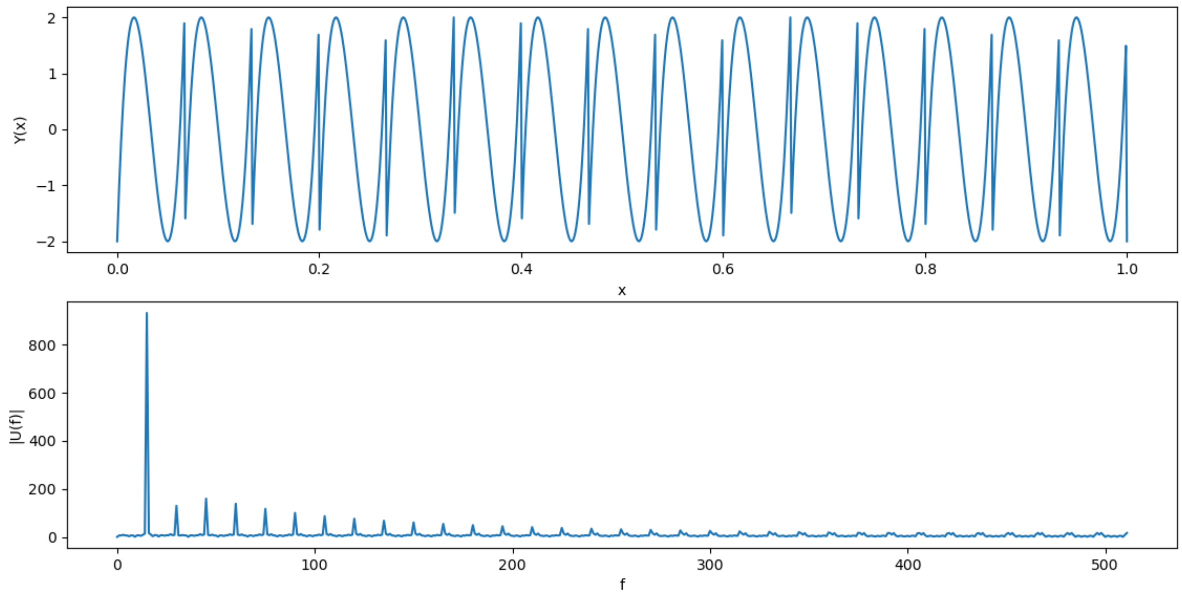
The functions $Y_1(x), Y_2(x), Y_3(x)$, and $Y_4(x)$ represent a periodic parabola, cubic curve, smooth hill curve, and a curve with sharp corners, respectively.

Suppose $k = 0, 1, 2, 3, \dots$ and $\bar{k} = 1, 2, 3, 4, \dots$ .Then the power spectrum of $Y_1(x)$ is shown below:



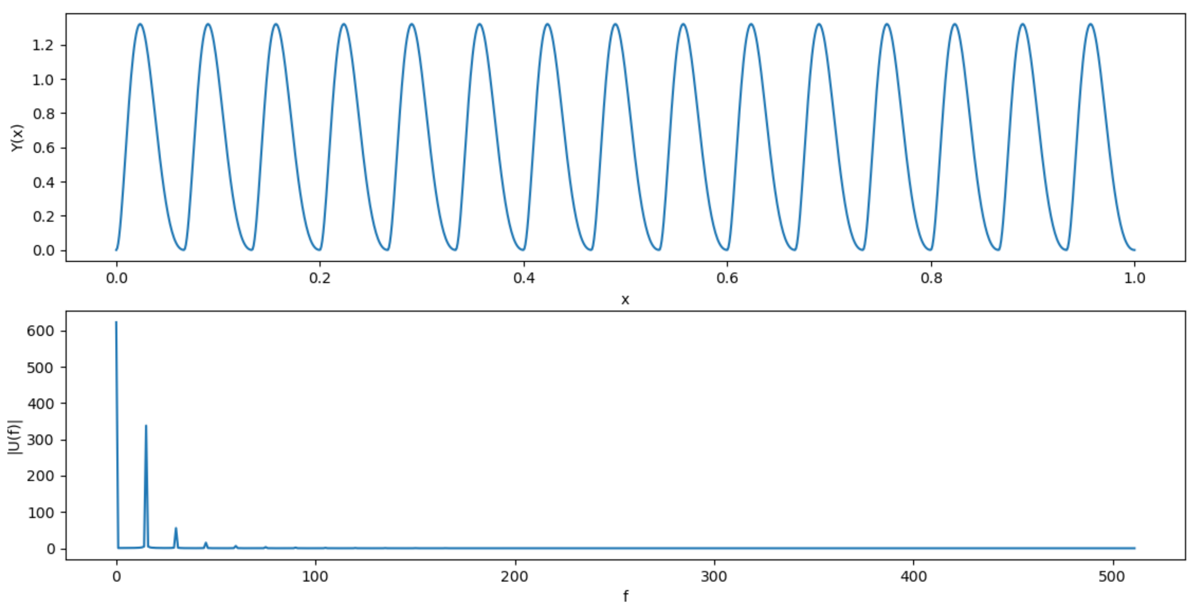
Its power spectrum has spikes at $f = 0,\ 15,\ 30,\ 45, \dots, 15k$ Hz and has relative spike heights of $0.245,\ 1.0,\ 0.456,\ 0.297, \dots$, thereby confirming the conjecture for this case.

The power spectrum for $Y_2(x)$ is shown below:



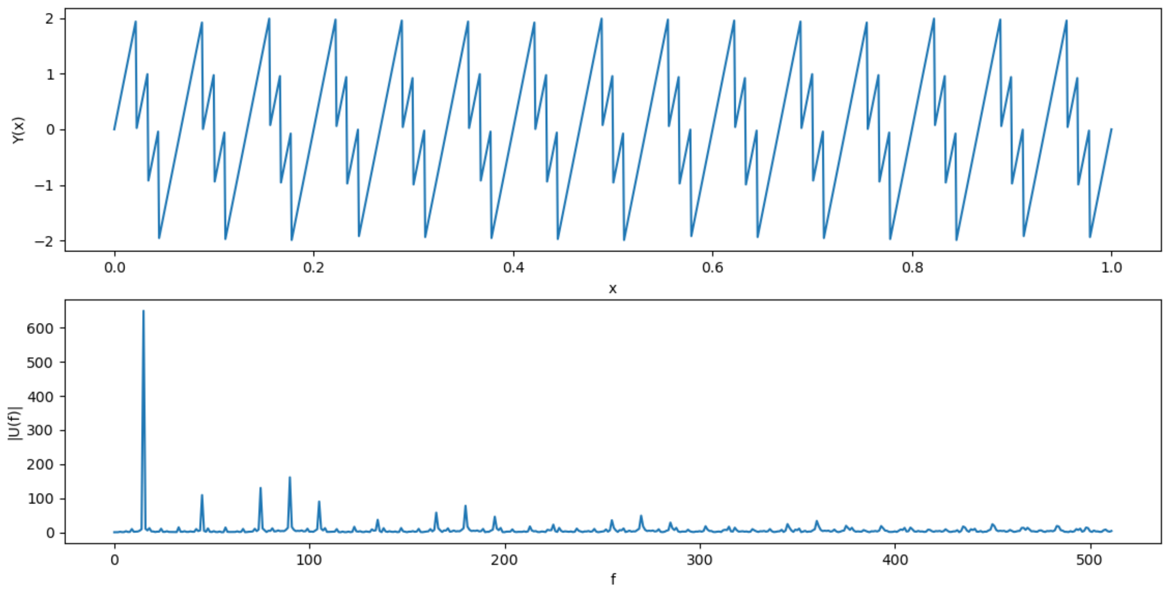
Its power spectrum has spikes at $f = 15,\ 30,\ 45,\ 60, \dots, 15\bar{k}$ Hz, with relative spike heights of $1,\ 0.139,\ 0.171,\ 0.148, \dots$, which also confirms the conjecture. Notice that this spectrum has no spike at $f = 0$ Hz.

The power spectrum for $Y_3(x)$ is shown below:



Its power spectrum has spikes at $f = 0,\ 15,\ 30$ and 45 Hz, and has relative spike heights of $1.0,\ 0.543,\ 0.089$, and 0.089. Since the spike frequencies are all multiples of 15 Hz, this result in line with the above conjecture. Notice that the original function is continuous, unlike $Y_1(x)$ and $Y_2(x)$, and that the height of the spikes drops off rapidly, meaning that $Y_3(x)$ has mostly low-frequency content.

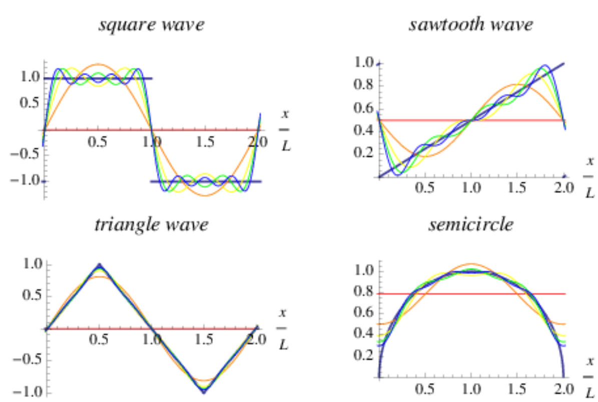
Finally, the power spectrum for $Y_4(x)$ is shown below:



Its power spectrum has spikes at $f = 15,\ \ 45,\ \ 75,\ \ 90,\ \ 105,\ \ 135,\ \ 165$ and 180 Hz. Notice that the pattern of spikes is not necessarily decreasing, later spikes occur in groups of three, and no spike occurs at $f = 0$ Hz. Since all the spike frequencies are multiples of 15 Hz, this result also supports the above conjecture.

Recall that taking the IDFT of the DFT allows us to reconstruct the original signal from its component frequencies. According to the aforementioned conjecture, many, if not all, periodic functions have their frequency content concentrated on multiples of their fundamental frequencies, which may or may not include $f = 0$ Hz. Thus, we can reconstruct the periodic function by just using an infinite number of cosines and sines whose frequencies are multiples of the fundamental frequency, instead of being any frequency.

The infinite sum that allows us to express a periodic function $Y(x)$ as a sum of cosines and sines is called the ***Fourier Series***, and the values by which we adjust the frequency content (or adjusting the heights of the spikes) are called ***Fourier Coefficients***. As the number of terms in the sum approaches infinity, the value of the Fourier Series approaches $Y(x)$ at all x, as seen in the graphs below:



Section 16 - Linearity and Scaling Properties of the Fourier Transform

The Continuous Fourier Transform has several properties with regards to linearity, scaling, differentiation, integration, and multiplication of transforms. Some of these properties, including those we will discuss in this section, are also exhibited by the Discrete Fourier Transform

First, the Fourier Transform has the ***linearity property***, which the DFT also always satisfies.

Theorem 11 – Linearity Property of Fourier Transform

If $Y(x) = aY_1(x) + bY_2(x)$ represents a signal in the time domain and a and b are complex scalars, then its Continuous Fourier Transform $U(f)$ is given by:

$$U(f) = aU_1(f) + bY_2(f)$$

where $U(f) = \mathcal{F}(Y(x))$, $\quad U_1(f) = \mathcal{F}(Y_1(x))$, $\quad$ and $\quad U_2(f) = \mathcal{F}(Y_2(x))$.

Note that $U(f)$, $U_1(f)$, and $U_2(f)$ refer to the complex CFT values, not the magnitudes of the corresponding CFTs.

To explain why this also holds for the DFT, consider the matrix equation for computing the DFT directly:

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}, \qquad \omega = \exp\left(\frac{-2\pi i}{n}\right)$$

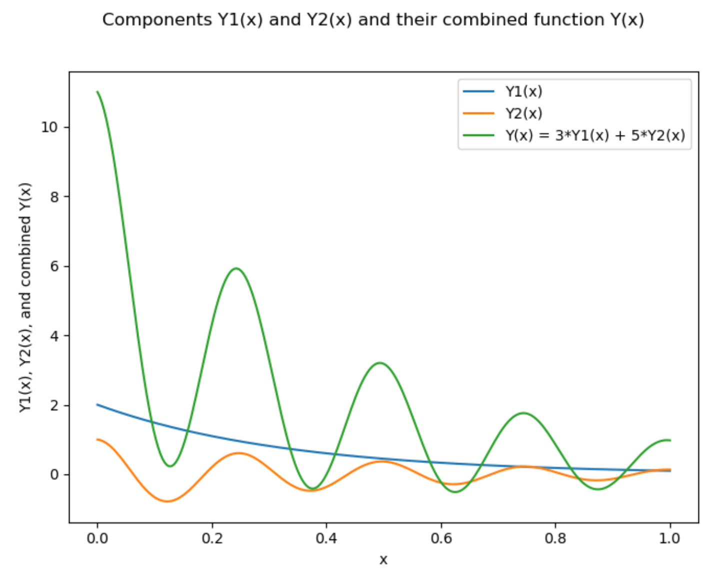
or more simply expressed as:

$$\mathbf{u} = F_n \mathbf{y}$$

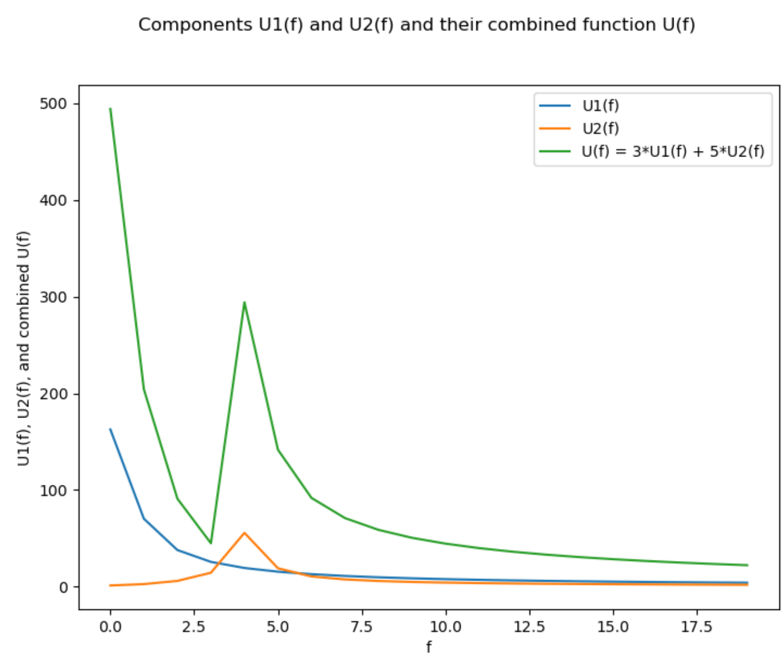
This means that the DFT is a ***matrix transformation*** on the space of complex vectors $\mathbb{C}^n$, since the output vector $\mathbf{u}$ is the matrix product of F_n and $\mathbf{y}$. However, Theorem 1.8.3 also implies that the DFT is also a ***linear transformation***, hence the linearity property must hold for vectors in $\mathbb{C}^n$ and complex scalars.

It is also true that the Fourier Transform is a linear transformation on the set of functions $F(a, b)$ defined on the interval $[a, b]$, where a and b are real numbers.

For instance, say that we have two functions $Y_1(x) = 2e^{-3x}$ and $Y_2(x) = \cos(8\pi x)\, e^{-2x}$ in addition to their linear combination given by $Y(x) = 3Y_1(x) + 5Y_2(x)$ defined on the time domain $[0, 1]$. Suppose we sample $n = 2^8 = 128$ points from the domain. The plot for $Y_1(x), Y_2(x)$ and $Y(x)$ is shown in the figure below:



Then if $U_1(f)$, $U_2(f)$ and $U(f)$ are the DFTs of $Y_1(x), Y_2(x)$, and $Y(x)$, respectively, computed using the FFT, we must have $U(f) = 3U_1(f) + 5U_2(f)$ due to the linearity property. The graph below shows the power spectra of the three functions, where the green curve is the magnitude of the DFT when the two component DFTs are scaled and combined:



Another important property of the Fourier Transform is the time-scaling property:

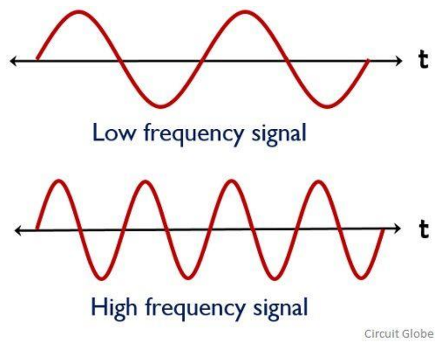
Theorem 12 – Time-Scaling Property of the Fourier Transform

If $U_1(f) = \mathcal{F}\big(Y_1(x)\big)$ *and* $Y_2 = Y_1(ax)$ *where a is a complex scalar, then:*

$$U_2(f) = \mathcal{F}(Y_2(x)) = \frac{1}{|a|} U_1\left(\frac{f}{a}\right)$$

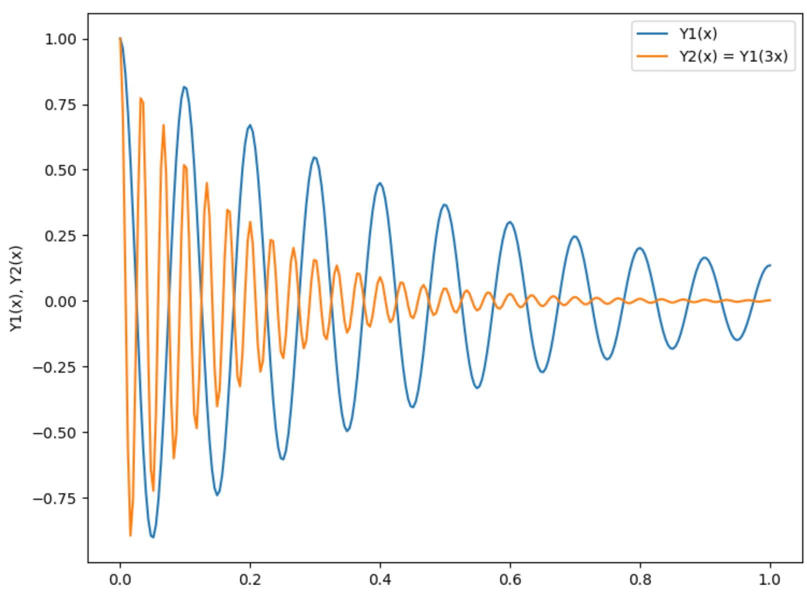
This means that horizontally stretching the graph of a function in the time domain to twice its size causes the function's CFT to have twice the magnitude and to have its graph compressed horizontally to half its size. On the other hand, compressing the graph for the time domain causes the FT to have a smaller magnitude and to have its frequency domain expanded horizontally.

This makes sense, since stretching the graph of a sinusoid causes its period to increase and its frequency to decrease, leading to its frequency domain to compress, and compressing a sinusoid horizontally causes its period to decrease and its frequency to increase, leading to its frequency domain to stretch.

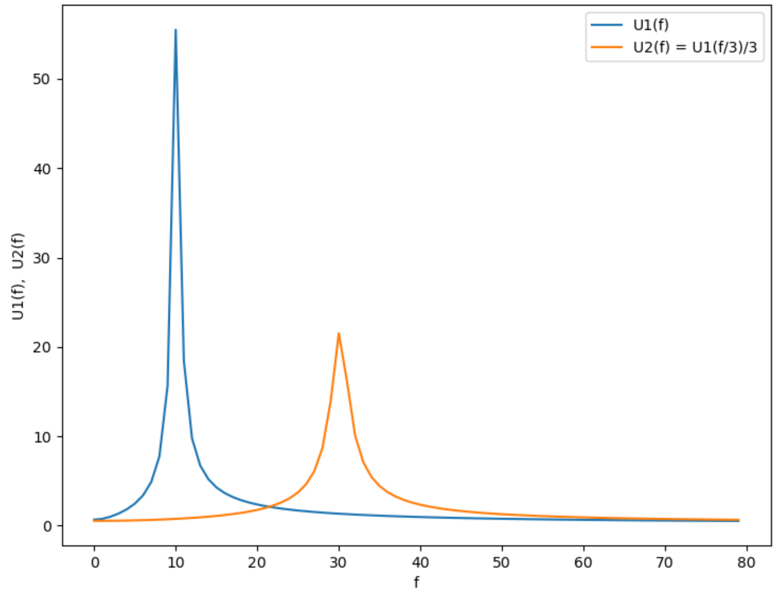


However, this property may be difficult to perfectly observe with a DFT due to the gaps between associated frequency values, as we illustrate in the following example.

Suppose we have $Y_1(x) = \cos(20\pi x)\, e^{-2x},\ Y_2(x) = Y_1(3x)$, and $n = 128$ samples on $[0, 1]$. The graphs for $Y_1(x)$ and $Y_2(x)$ are shown below:



If $U_1(f)$ and $U_2(f)$ are the DFTs of $Y_1(x)$ and $Y_2(x)$, then we should expect from the above property that $U_2(f)$ is stretched thrice horizontally and has one-third the height compared to $U_1(f)$. The graph below shows the power spectra for the two functions:

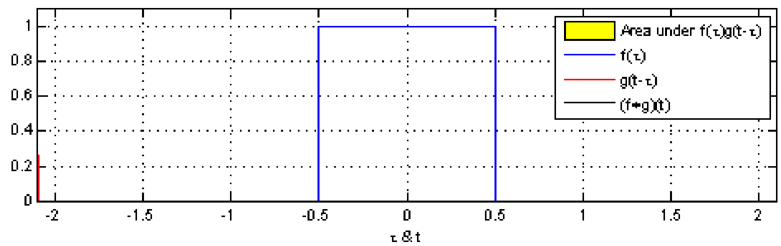


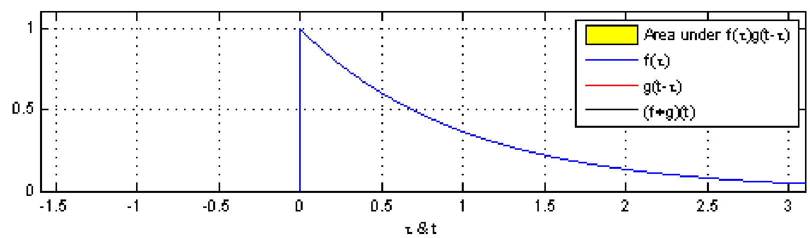
It can be observed from the above plot that the peaks for $U_1(f)$ and $U_2(f)$ occur at $f = 10$ Hz and $f = 30$ Hz, respectively, which is as expected. However, the ratio of the height of the blue peak to the orange peak above is only about 2.56, not exactly 3. A good reason for this is that the DFT only serves as an approximation to the CFT since it is defined only at discrete frequencies, resulting in the discrepancy in peak heights. We can make the DFT more accurate by decreasing the gaps between frequency values, as we will see in Section 18.

Section 17 - Computing Discrete Convolutions using the FFT and IFFT

The last property we will deal with in this section involves a certain mathematical operation between two functions, which is called ***convolution***. If $f(x)$ and $g(x)$ are functions defined on $(-\infty, \infty)$, then their ***convolution function***, written as $(f \oplus g)(x)$, is another function which expresses how the shape or graph of one function is modified by the shape of the other. In other words, the net effect of convolution is that it "blends" the two functions together. It is important to note that convolution is commutative, that is, $(f \oplus g)(x) = (g \oplus f)(x)$.

The graph of the convolution function may be generated by "sliding" the graph of first function over the second function, resulting in a displacement τ, and its height at every x (or t in the example) is the area of the intersection of the two shapes, as is shown in the following animated images:





Computing the convolution directly requires calculations involving integrals, however, the convolution theorem as stated below, will allow us to compute for the convolution of two functions using the FFT and IFFT:

Theorem 13 – Convolution Theorem for the Fourier Transform

Suppose $U_1(f) = \mathcal{F}(Y_1(x)),\ U_2(f) = \mathcal{F}(Y_2(x)),$ *and* $U(f) = \mathcal{F}(Y(x))$.

Then $U(f) = U_1(f) * U_2(f)$ *for all* f *if and only if* $Y(x) = (Y_1 \oplus Y_2)(x)$ *for all* x.

In other words, multiplication in the frequency domain always corresponds to convolution in the time domain.

This suggests the following procedure to compute the ***discrete convolution*** of two functions defined at discrete points:

1. Compute the DFTs of both functions using the FFT.
2. Get the DFT of the discrete convolution by multiplying the corresponding DFT values from Step 1.
3. Compute the discrete convolution by applying the IFFT on the DFT values from Step 2.

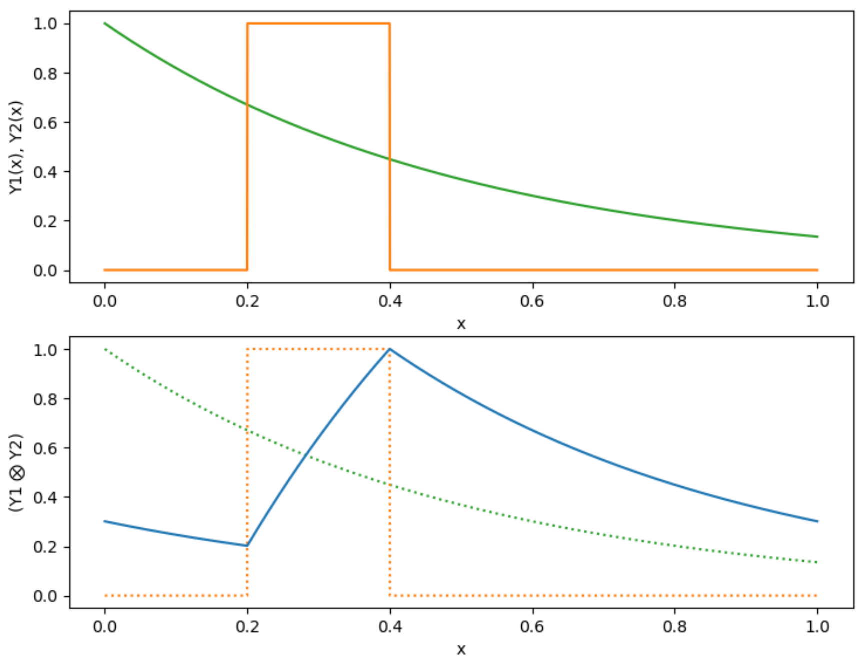
For the examples in this section, we will also "normalize" the discrete convolution by dividing each value by the maximum value in the convolution. This ensures that the discrete convolution will always have values from 0 to 1 and that there are no overly large values relative to the values of the original functions.

For the following examples, the two functions $Y_1(x)$ and $Y_2(x)$ that are to be convolved have a range of $[0, 1]$. We will also introduce the boxcar and triangle pulse functions, of which represent short rectangular and triangular pulses or signals, respectively:

$$\mathrm{box}(x, x_0, w, A) = \begin{cases} A, & \text{if } 0 < x - x_0 < w \\ 0, & \text{otherwise} \end{cases}$$

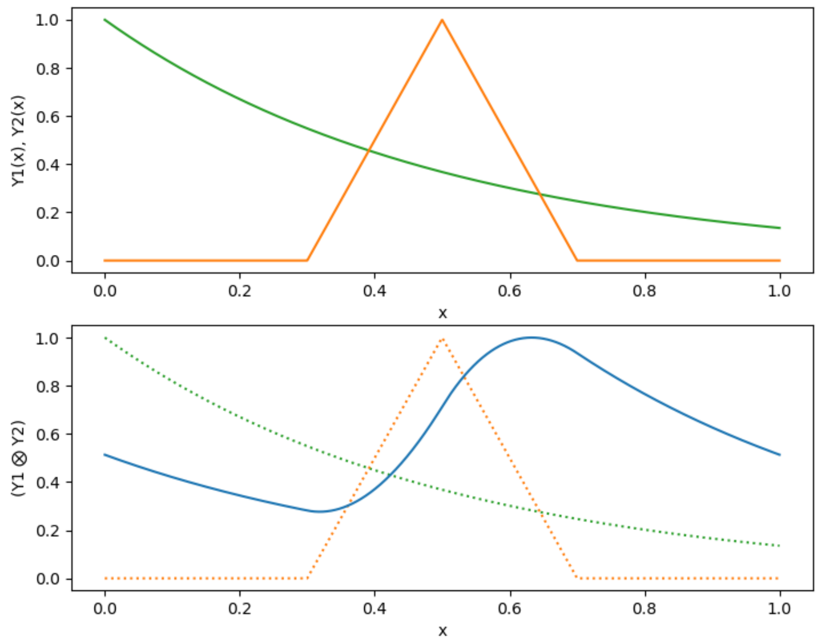
$$\mathrm{trp}(x, x_0, w, A) = \begin{cases} 0, & \text{if } x < x_0 \text{ or } x > x_0 + w \\ pA, & \text{if } 0 < p \leq 1 \\ (2 - p)A, & \text{if } 1 < p < 2 \end{cases} \quad , \text{ where } p = 2\left(\frac{x - x_0}{w}\right)$$

First, suppose we had a decaying exponential $Y_1(x) = e^{-2x}$ and we convolved it with a box car $Y_2(x) = \mathrm{box}(x, 0.2, 0.2, 1)$. These functions (in green and orange, respectively) together with their convolution $(Y_1 \oplus Y_2)(x)$ (in blue) are plotted below:



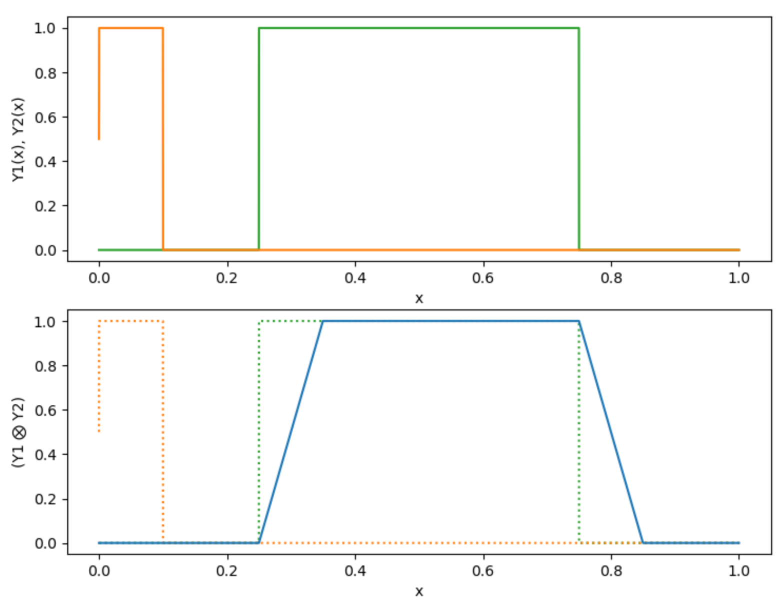
Notice that the graph of the convolution resembles a decaying exponential outside the interval $[0.2, 0.4]$, but rises sharply within $[0.2, 0.4]$ due to the boxcar value being 1. This means that the convolution had "blended" the properties of the two original curves.

Now suppose we convolve the same decaying exponential $Y_1(x) = e^{-2x}$ but with a triangle pulse $Y_2(x) = \mathrm{trp}(x, 0.3, 0.4, 1)$ instead. This results in the following plot:



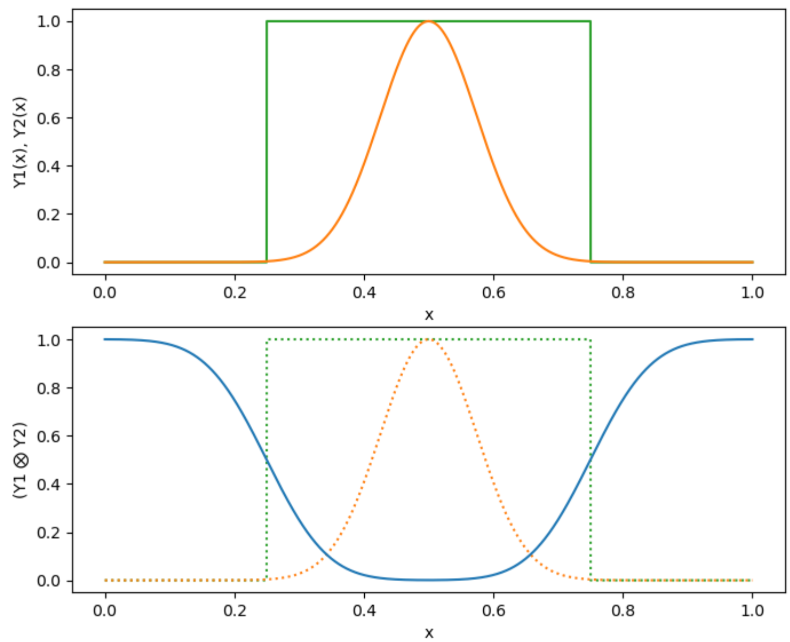
Notice that the graph of the convolution decays exponentially outside $[0.3, 0.7]$ but rises smoothly within $[0.3, 0.7]$ at a rate proportional to the height of the triangular pulse at any value of x.

If we convolved two boxcar functions, $Y_1(x) = \text{box}(x,\ 0.25,\ 0.5, 1)$ and $Y_2(x) = \text{box}(x,\ 0,\ 0.1,\ 1)$, with the former being longer than the latter, then the following plot results:



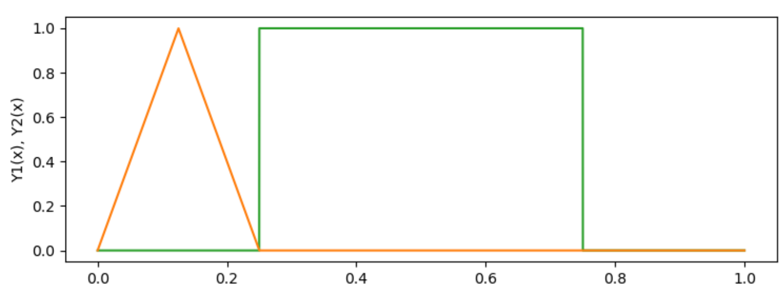
Notice that the graph of the convolution is a trapezoid marking the intersecting area when the smaller orange pulse or $Y_2(x)$ is translated across the large green pulse or $Y_1(x)$.

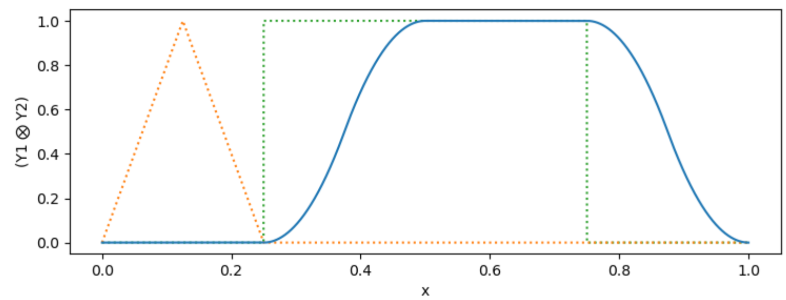
If we convolve the same green boxcar function $Y_1(x) = \text{box}(x,\ 0.25,\ 0.5, 1)$ with a bell curve function $Y_2(x) = e^{-90(x-0.5)^2}$, then the resulting plot is as follows:



Unlike in the previous example, the value of the convolution function decreases when the area of intersection between the orange bell curve (which is being moved across) and the green boxcar pulse increases. However, it still blends the two functions together, as it is curved toward the edges of the boxcar and nearly flat everywhere else.

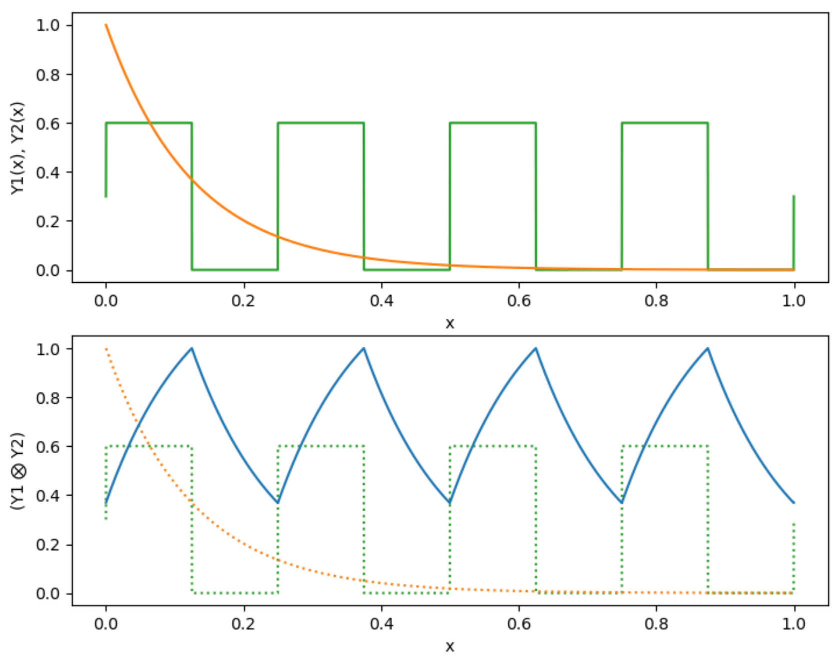
If we convolve the same green boxcar function $Y_1(x) = \text{box}(x,\ 0.25,\ 0.5,\ 1)$ with a triangle pulse $Y_2(x) = \text{trp}(x,\ 0,\ 0.25,\ 1)$, the plot is as shown below:





Notice that as we slide the orange triangular pulse across the green pulse, the value of the convolution function changes in proportion to the area of intersection between the pulses, and that the convolution function is curved despite being generated from pulses whose graphs are composed only of line segments.

Finally, if we convolve a square wave $Y_1(x) = \mathrm{sq}(x,\ 4,\ 0.3,\ 1) + 0.3$ with a decaying exponential $Y_2(x) = e^{-8x}$, the convolution, as plotted below, still "blends" the characteristics of the two functions together:



Notice that the value of the convolution function rises on values of x where $Y_1(x) = 0.6$ and decays whenever $Y_1(x) = 0$ and appears to be composed of multiple exponential decays and growths. This example of convolution comes up in the study of electric circuits, where the square wave $Y_1(x)$ may represent digital signals that keep a circuit running and $Y_2(x)$ represents the voltage decaying if these digital signals were absent.

Section 18 - Expanding the Time Domain and Increasing Frequency Resolution

So far, we have been concerned with the computation and properties of the DFT when the time domain is in $[0,1]$ and the frequency domain is $[0,s]$, where $\left[0,\frac{s}{2}\right]$ conveys the frequency content of $Y(x)$, and the sampling rate s is equal to n.

To extend the usage of the DFT to the time domain $[0,L]$, where L is a real number, we will need to define what the new frequency domain is in terms of the sampling rate $s=n/L$. Note that this does not in any way affect how the DFT values are calculated and thus the FFT is still applicable even to this case.

Recall from Section 2 that the DFT is a linear transformation from R^n to R^n which translates from the time domain to the frequency domain and has the following four assumptions:

a. *The DFT is defined only on* n *discrete values of* x *and* n *discrete values of* f.
b. *Each input value for the DFT is associated with the value of* $Y(x)$ *in the time domain and each output value is associated with the value of the DFT* $U(f)$ *in the frequency domain.*
c. *At the first nonzero discrete frequency,* f_1, *wrapping the function results in a graph that covers exactly 1 revolution around the origin.*
d. *The* jth *frequency* f_j *is the* jth *scalar multiple of the first frequency* f_1, *that is:*

$$f_0=0,\ f_1=f_1,\ f_2=2f_1,\ \ f_3=3f_1,\ \ldots,\ f_{n-1}=(n-1)f_1.$$

Moreover, recall that the components of the DFT can be interpreted as the coordinates of a center of mass resulting from wrapping the graph of $Y(x)$ along the unit circle. This is mathematically expressed as the following general formula:

$$U(f_j)=\sum_{k=0}^{n-1}Y(x_k)\exp(-2\pi i f_j x_k)$$

Assumption (c) requires that wrapping the graph of $Y(x)$ on $[0,L]$ will correspond to a wound-up graph which makes exactly one revolution clockwise around the origin when $f=f_1$, that is:

$$-2\pi f_1 L=-2\pi$$

$$f_1=\frac{1}{L}$$

Thus, the first frequency or the ***gap between frequencies*** will simply be the reciprocal of the length of the time domain.

Furthermore, assumption (d) implies that the maximum frequency in the frequency domain is:

$$f_{n-1}=(n-1)f_1=\frac{n-1}{L}$$

If we define:

$$R=\frac{n}{L}=s$$

Then the new frequency domain is within $[0,R]$, where the frequency range R is equivalent to the sampling rate s, and the portion of the domain which conveys the frequency content of $Y(x)$ is $\left[0,\frac{R}{2}\right]$, with $f_N=\frac{R}{2}$ being the Nyquist frequency, according to the Nyquist-Shannon Sampling Theorem.

Time Domain		**Frequency Domain**
$x \in [0, L]$	$\longleftrightarrow$	$f \in [0, R]$

The above discussion implies that increasing the sampling rate while preserving the time domain causes the frequency domain to expand while the gaps between discrete frequencies stays constant, and vice versa. This makes sense, since the Nyquist-Shannon Sampling Theorem requires the highest frequency component to be analyzed, and so is the frequency range, to be directly proportional to the sampling rate.

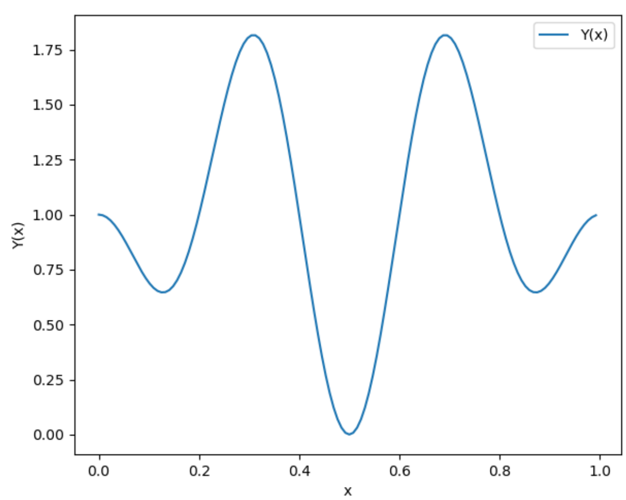
On the other hand, expanding the time domain while keeping the sampling rate constant causes the gaps between discrete frequencies to shrink while the frequency domain stays the same, and vice versa.

Time Domain	**Frequency Domain**
Length L	*Length* $R = s$
Sampling Rate s	*Sampling Rate L*

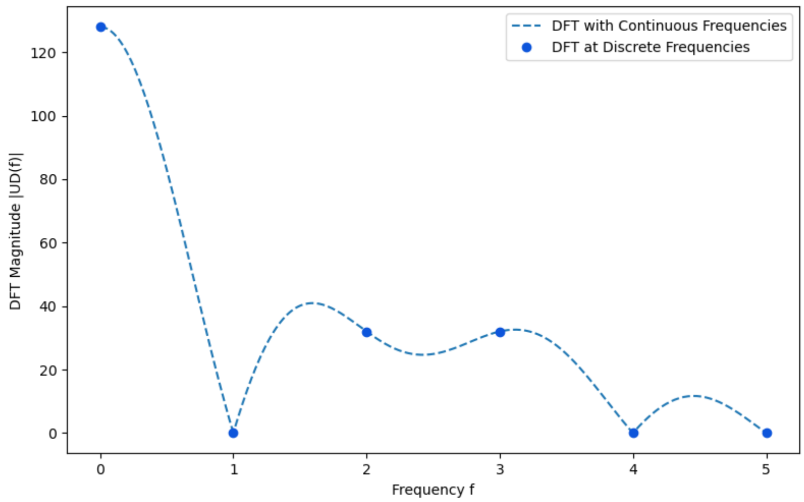
Both methods are ways of making the DFT more accurate by increasing the number of samples n, however, the same methods of calculation, that being either the Fast Fourier Transform or direct calculation of the DFT, will still apply.

For example, say that $Y(x) = \sin^2(2\pi x) + \cos^2(3\pi x)$ and we take its DFT on the interval $[0, L]$ with a sampling rate of $s = 128$. The component sinusoids $\sin(2\pi x)$ and $\cos(3\pi x)$ have frequencies of 1 Hz and 1.5 Hz, respectively, but since squaring a sinusoid seems to double its frequency, we expect the power spectrum of $Y(x)$ to have spikes at $f = 2$ and 3 regardless the value of L.

At $L = 1$, $Y(x)$ has the following plot:

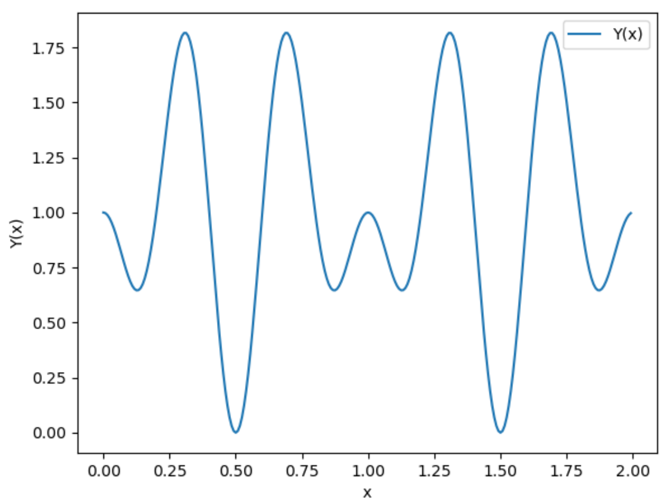


The power spectrum or the graph of $|U_D(f)|$ of $Y(x)$ is plotted below, where the blue points represent the values of the DFT that can be computed via the FFT and the dashed curve represents values of the DFT outside the allowable discrete frequencies:

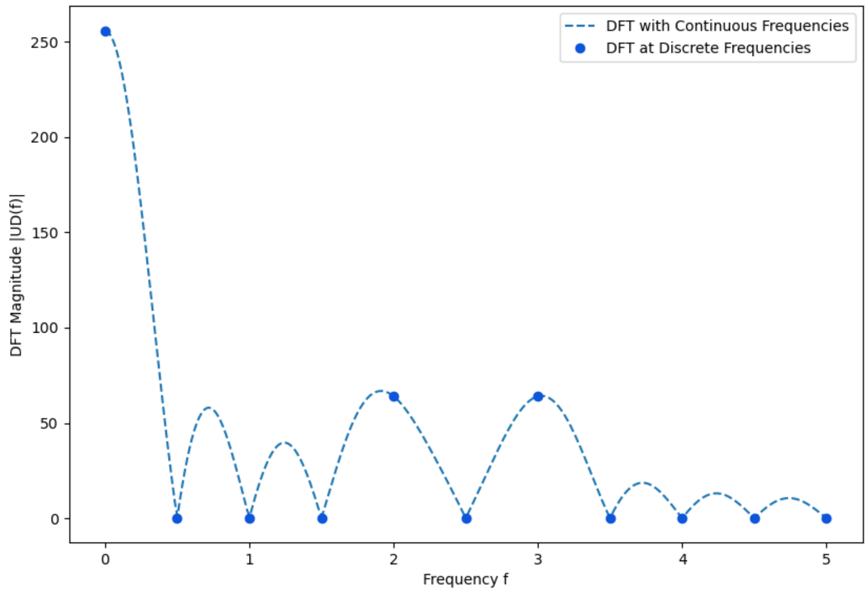


It can be seen from the above plot that there is no spike at $f = 2$ and 3 and thus the DFT does not distinguish well between the 2 Hz and 3 Hz frequency components.

Now if the length of the time domain is doubled by setting $L = 2$, $Y(x)$ will now repeat twice:

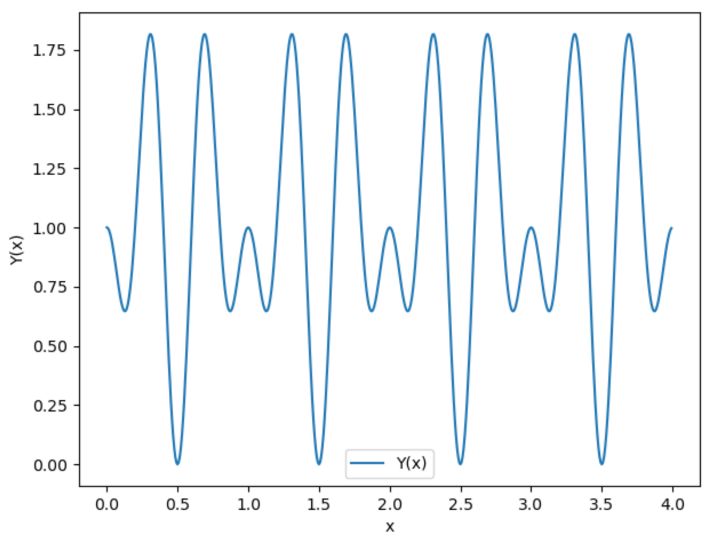


This causes the gap between discrete frequencies to be halved $\left(f_1 = \frac{1}{2}\right)$, and thus the DFT is better able to distinguish the two frequencies $f = 2$ and $f = 3$, as evidenced in the new power spectrum plotted below:

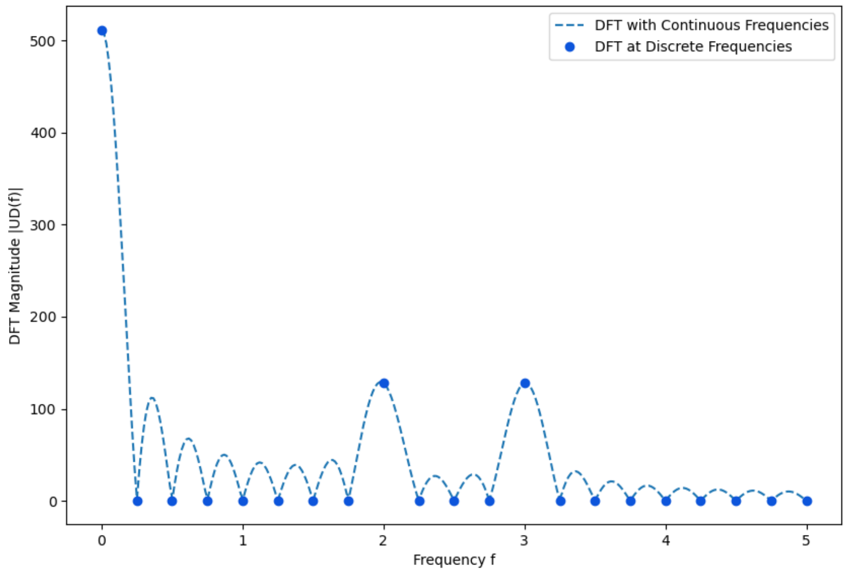


However, the large hills on $f = 2$ and 3 do not significantly stand out from the other smaller hills in the above plot.

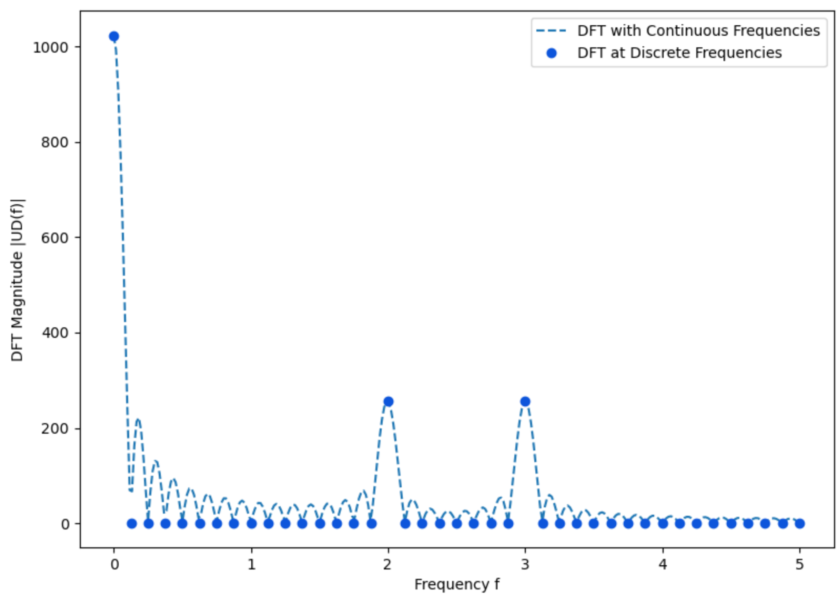
Doubling the length once more by setting $L = 4$ now leads to $Y(x)$ repeating four times:



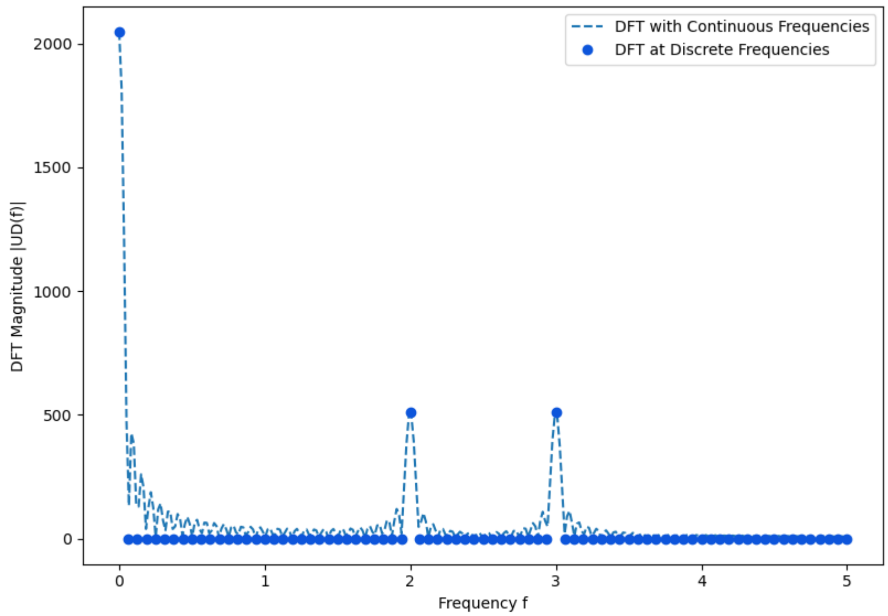
The gap between frequencies halves again $\left(f_1 = \frac{1}{4}\right)$, and the DFT begins to distinguishes between $f = 2$ and $f = 3$ well, as seen in the power spectrum below:



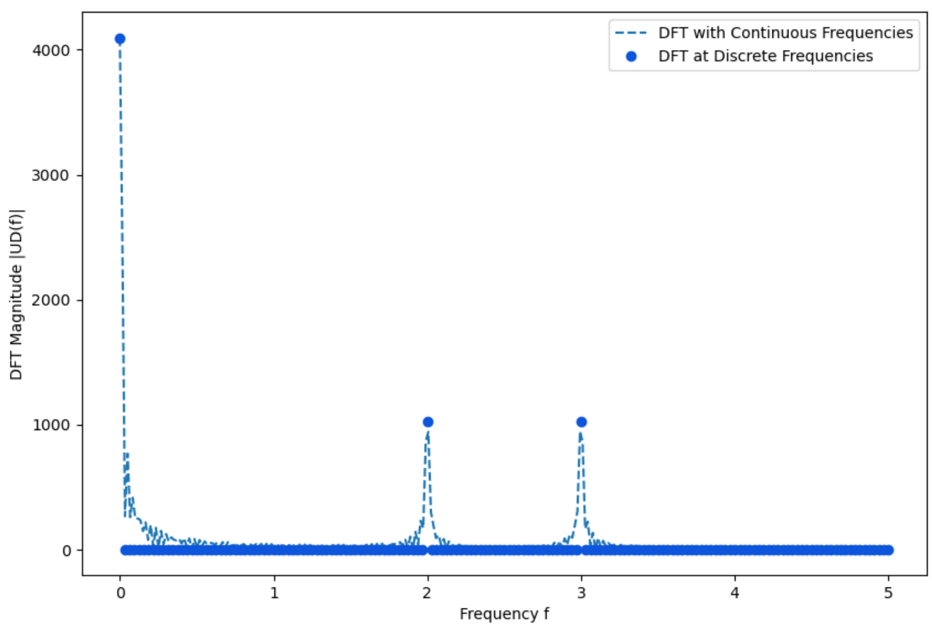
When $L = 8$, and therefore $f_1 = \frac{1}{8}$, the spikes at $f = 2$ and 3 are more significantly visible compared to the smaller hills in the power spectrum:



This improves as L is increased to 16:



... and as L is increased to 32:



Notice that even though the magnitude of the DFT increases exponentially as L is increased, the relative heights of the peaks in the power spectrum do not change. In fact, the spikes at $f = 2$ and 3 have 1/4 the magnitude or height compared to the spike at $f = 0$ in all six cases.

The formation of spikes as L increases is also expected by the conjecture on repeating functions discussed earlier. As L increases, $Y(x)$ repeats itself more and more on the time domain and its power spectrum eventually forms spikes when L is sufficiently large ($L \geq 8$).

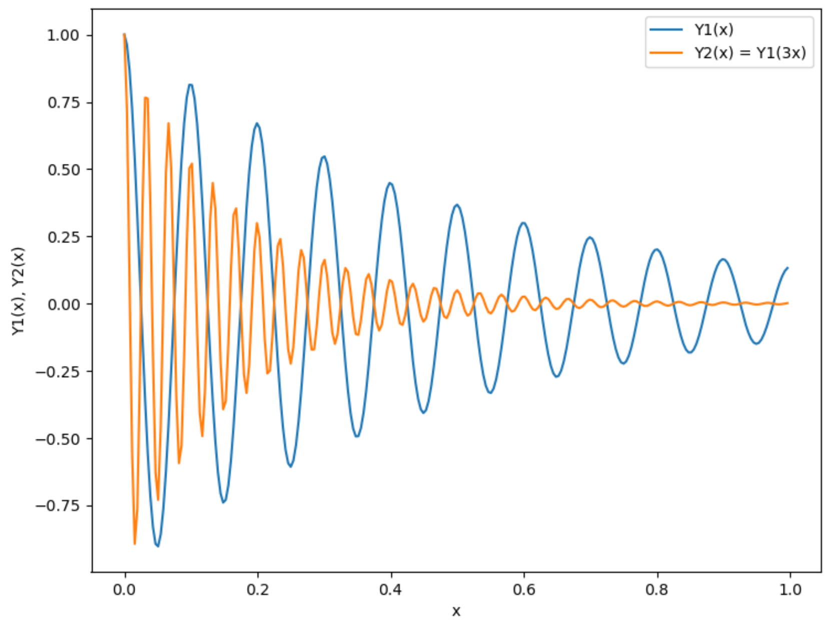
The ability for the DFT to distinguish between different frequencies is called its ***frequency resolution***. As we kept doubling L, the frequency resolution of the DFT became better and better, especially when $L \geq 8$, and this is also evidence that the DFT became more and more accurate as the sample size n increased.

To generalize, we say that expanding the time domain for a function $Y(x)$ increases the frequency resolution of its DFT $U_D(f)$ because the gaps between discrete frequency values decreases while the frequency range stays the same.

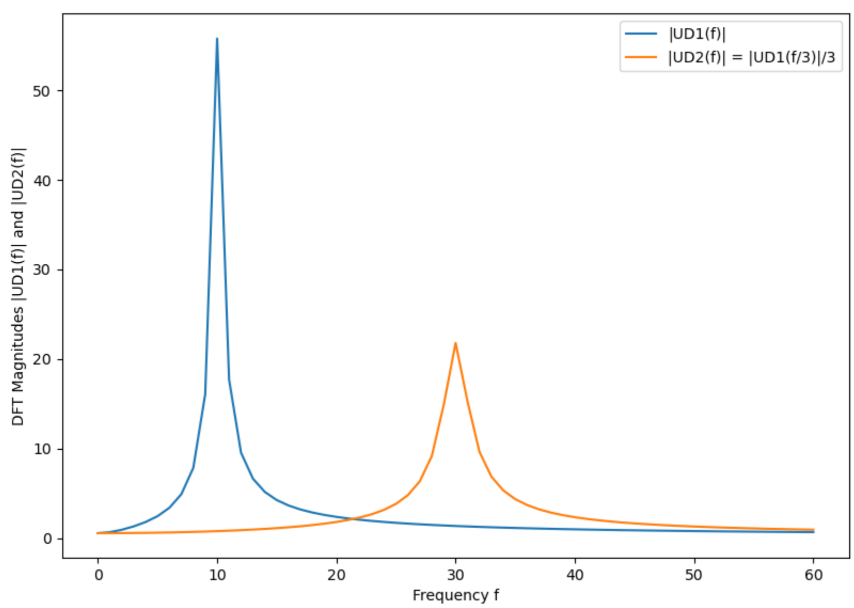
For the second example, suppose we have the following functions once more at a sampling rate of $s = 256$ and defined on the time domain $[0, L]$:

$$Y_1(x) = \exp(-2x) * \cos(20\pi x), \qquad Y_2 = Y_1(3x)$$

Plots for $Y_1(x)$ and $Y_2(x)$ are given below:



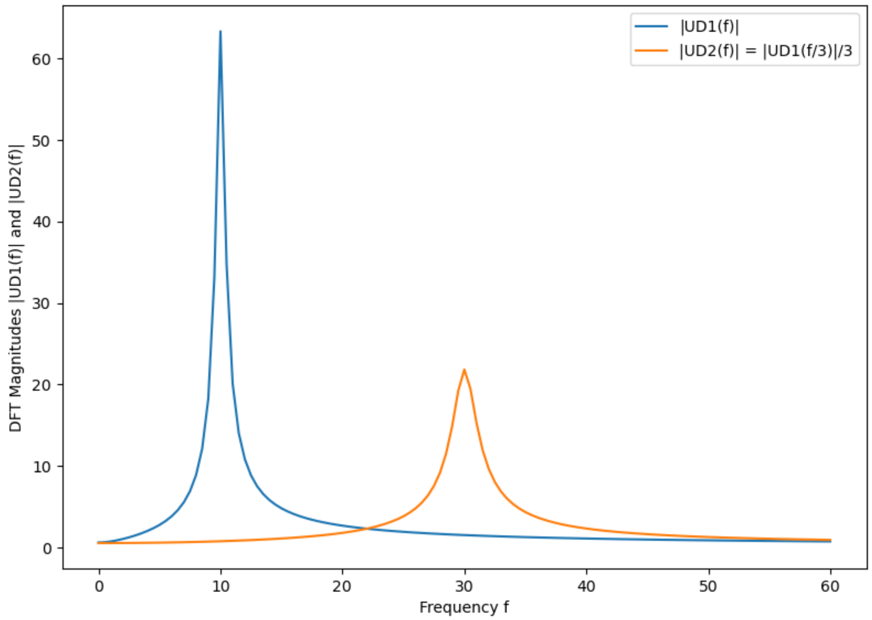
Before, we found that their power spectra had the following graphs when the time domain was restricted to $[0, 1]$ or $L = 1$:



The locations of the peaks, that being $f = 10$ and 30, correspond to the fact that $Y_1(x)$ and $Y_2(x)$ are decaying sinusoids at frequencies 10 Hz and 30 Hz, respectively.

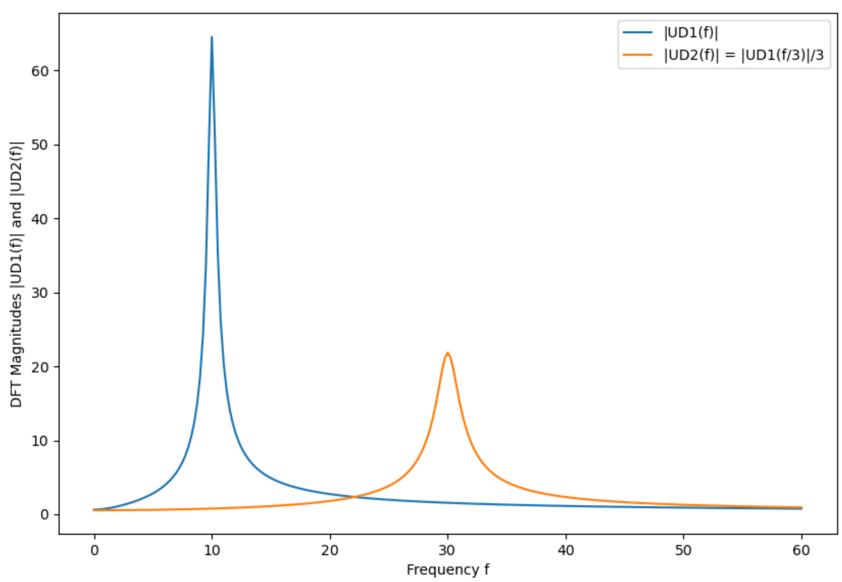
The time scaling property of the FT requires that the ratio of the height of the blue peak to the orange peak is exactly 3. However, the heights of the peaks in the above plot have a ratio of 2.56, which is a discrepancy caused by the fact that the DFT only approximates the FT.

If we set $L = 2$, the power spectrum has its overall shape preserved while its magnitudes increase slightly:



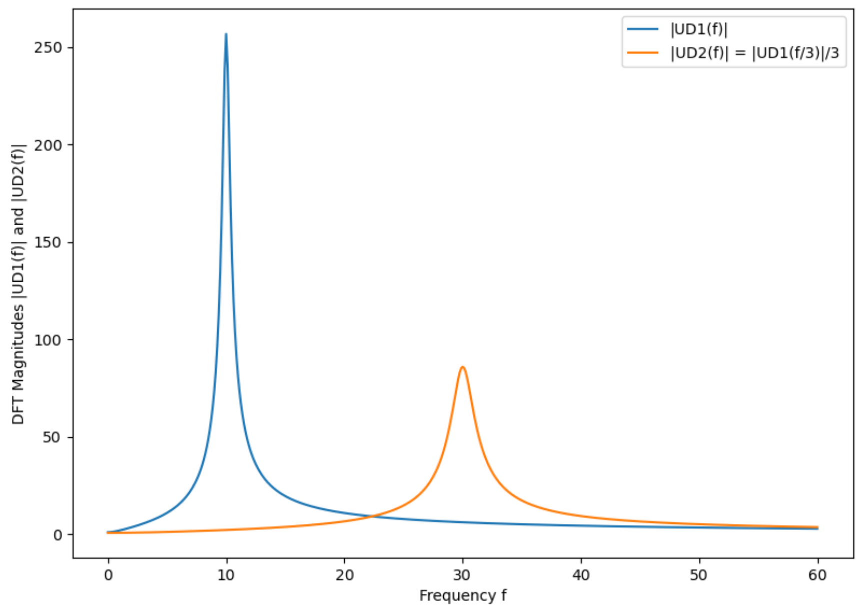
The ratio between the heights of the peaks becomes 2.90, which is closer to 3.

When $L = 4$, the power spectrum barely changes in height and shape:



However, the ratio now stabilizes at 2.95 on this point on. In fact, for higher values of L, the ratio stays at 2.95 while the shape and magnitude of the power spectrum stay virtually identical to the spectrum for $L = 4$. This is unlike the previous example, where the magnitude of the power spectrum increased exponentially and the shape of the graph changes significantly while the height ratios between spikes did not change as L increased.

We can also make the DFT more accurate and thus bring the height ratio closer to 3 by increasing the sampling rate s. For instance, when $s = 1024$, the height ratios when $L = 1, 2, 4,$ and 8 are 2.59, 2.93, 2.99, and 2.99, respectively, and the magnitudes of the DFT values have increased as shown in the plot below:



From both examples, we can infer that increasing the sample size $n = sL$ either by expanding the time domain beyond $[0, 1]$ or by increasing the sampling rate s can make the DFT more accurate in relation to the FT. However, the former often has a more profound effect on the accuracy of the DFT.

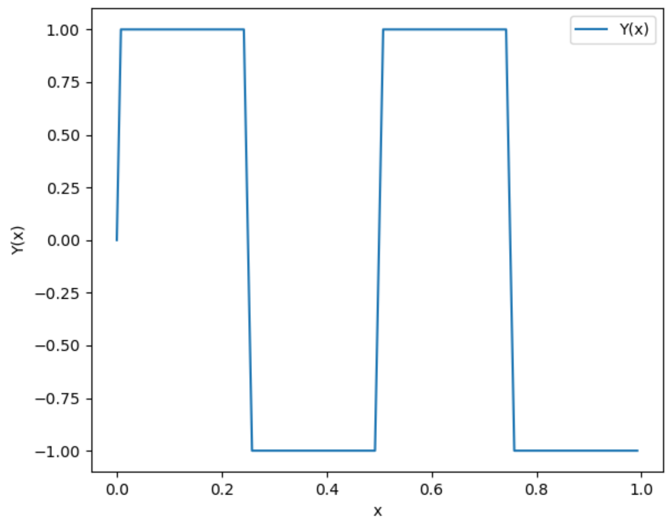
There is a fundamental limitation to this, however, as we have to take in more samples n in order to compute the DFT of $Y(x)$. Fortunately, using the FFT significantly shortens computational time compared to if the DFT were evaluated directly at higher values of n. Although the FFT works only if n is a power of 2, its reliability and speed even at large n supports it as the preferred choice for computing the DFT for frequency analysis.

Section 19 - Zero Padding and Polynomial Multiplication

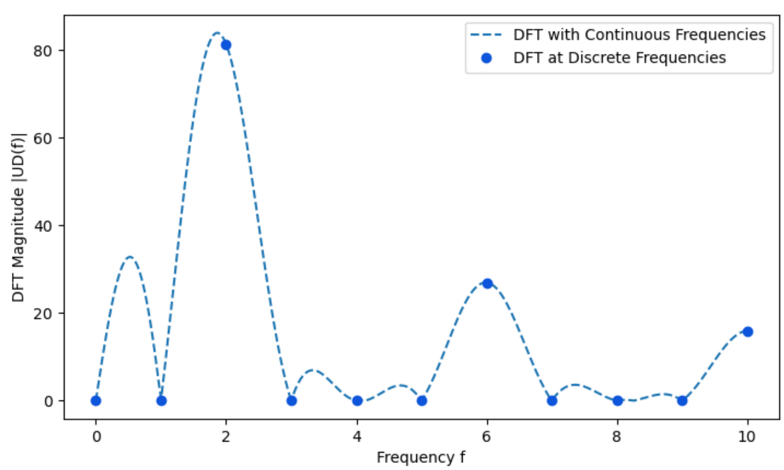
There are two choices to give new values of $Y(x)$ when expanding the time domain without adding new data: (a) repeat the function $Y(x)$ again, and (b) set $Y(x)$ to zero outside a selected interval. The latter technique of adding zeroes after the original data values is called ***zero-padding***, and although it decreases the gap between discrete frequencies while preserving the frequency range, it does NOT actually increase the frequency resolution of the DFT, unlike repetition.

Suppose that the sampling rate is $s = 128$, the time domain is $[0, L]$, and $Y(x) = \mathrm{sq}(x, 2, 1)$ represents a 2 Hz square wave with amplitude 1 on the interval $[0, 1]$ and $Y(x) = 0$ when $x > 1$.

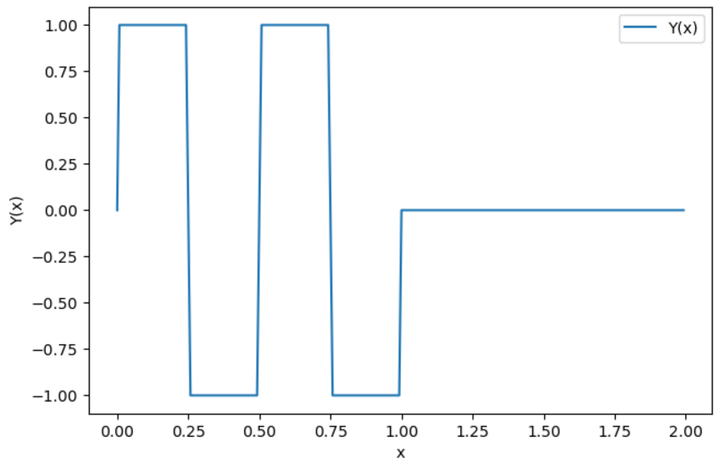
The graph of $Y(x)$ when $L = 1$ is plotted below:



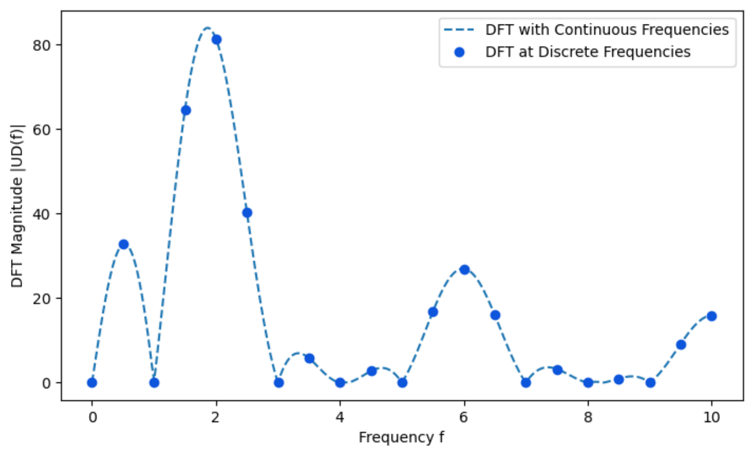
The power spectrum of $Y(x)$ when $L = 1$ is shown below:



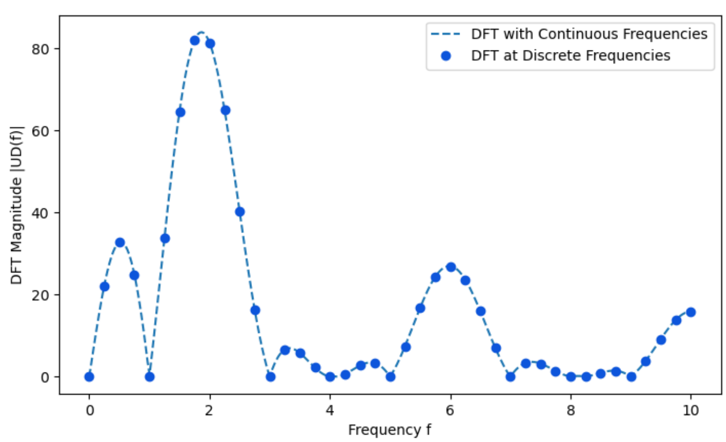
When L is doubled to $L = 2$, the graph of zero-padded $Y(x)$ becomes:



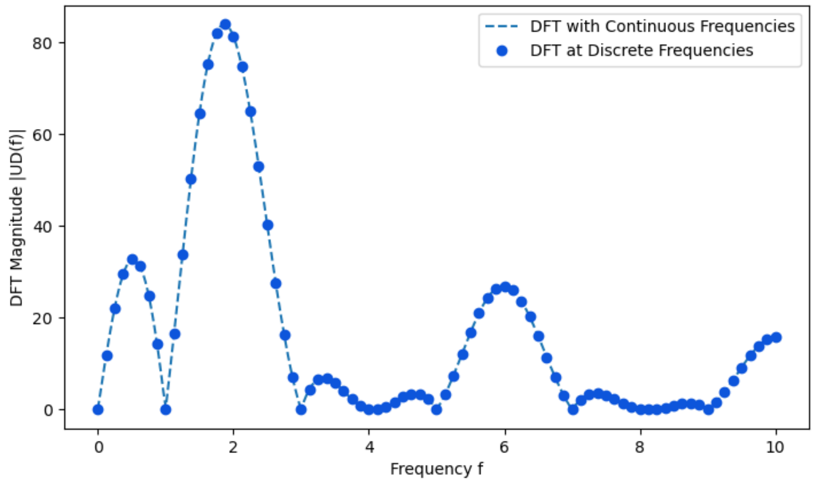
However, the shape and magnitude values of the power spectrum do not change and succeeding points only interpolate the dashed curve, which represents the magnitude of the computed DFT on frequencies outside the allowable discrete frequencies:



A similar behavior occurs as L is increased to $L = 4$:



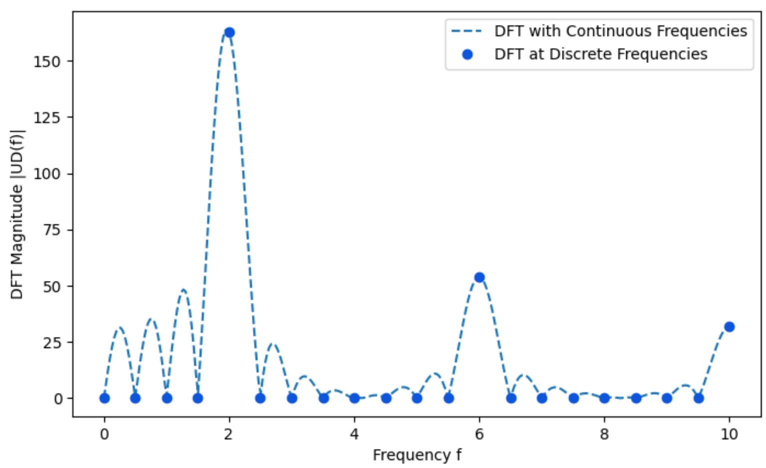
and when L is increased to $L = 8$:



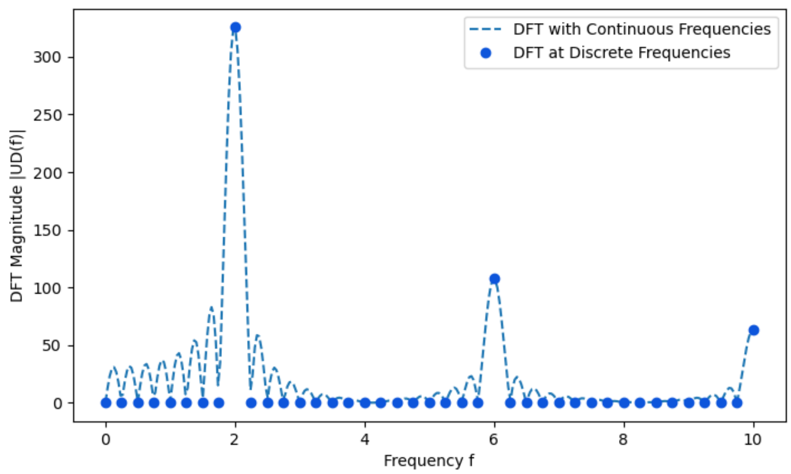
This means that zero-padding does not add any new frequency information and it only interpolates the frequency content of $Y(x)$ within $[0, 1]$.

In the center-of-mass interpretation for the DFT, zero-padding can be viewed as adding points on the origin to the figure, which does not affect the shape of the wrapped graph for $Y(x)$ but brings its center of mass closer to the origin since n increases. This however, does NOT affect the sum of all the complex values $Y(x_k) * \exp(-2\pi i x_k f_j)$ corresponding to points in the wrapped graph as adding by zero does not change its value. Since this sum is the value for the DFT, the values for the DFT do not change and therefore its power spectrum stays the same as before.

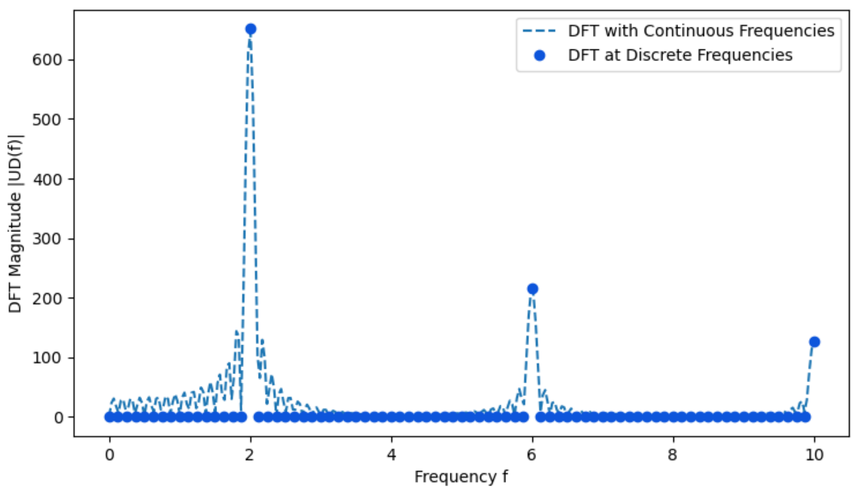
If we were to repeat the function instead, then the power spectrum of $Y(x)$ changes significantly as seen in the plot below for its power spectrum when $L = 2$:



... when $L = 4$:



... and when $L = 8$:



In contrast to zero-padding, repeating the function adds new frequency information to the DFT, therefore making it more accurate relative to the FT of the function.

The technique of zero-padding is not useful for increasing the accuracy of the DFT, but it may be used to bring the number of data values to a power of 2 and therefore allowing the FFT to be used. For example, if there are $n = 53$ recorded values for $Y(x)$, then 11 zeroes may be added after those 53 values to bring the total number to $n = 64 = 2^6$. Although the frequency domain will still have 64 values, only 53 values in the time domain are required.

Another instance in which zero-padding may be applied is in efficient ***polynomial multiplication*** using the FFT and IFFT.

Suppose we wish to get the product $P_3(x)$ of the following two polynomials:

$$P_1(x) = 3 - x + 3x^2 - 7x^3 + 0x^4 + 6x^5 + 7x^6 + 7x^7$$

$$P_2(x) = 5 - 3x - 3x^2 + 7x^3 - 5x^4 + x^5 + 0x^6 - x^7$$

$$P_3(x) = P_1(x) * P_2(x) = ?$$

Although $P_3(x)$ can be obtained by applying the distributive property on $P_1(x)$ and $P_2(x)$ repeatedly or by using the lattice method, it is more efficient to compute $P_3(x)$ by applying the FFT and IFFT on the original polynomial coefficients.

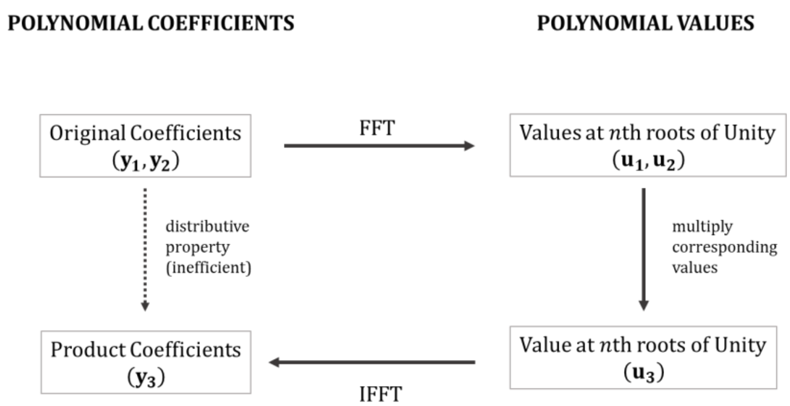
According to the polynomial interpretation for the DFT, we can interpret the coefficients of $P_1(x)$ and $P_2(x)$ in the same way as data values for two functions $Y_1(x)$ and $Y_2(x)$ in the time domain. Since the product $P_3(x)$ has degree $7 + 7 = 14$, which corresponds to 15 coefficients, and 16 is the lowest power of 2 that is at least 15, we have to write their coefficients in the vectors $\mathbf{y_1}$ and $\mathbf{y_2}$ in R^{16} from lowest power to highest power by adding 8 zeroes after the coefficients of the leading terms:

$$\mathbf{y_1} = (3, -1, 3, -7, 0, 6, 7, 7, \quad 0, 0, 0, 0, 0, 0, 0, 0)$$

$$\mathbf{y_2} = (5, -3, -3, 7, -5, 1, 0, -1, \quad 0, 0, 0, 0, 0, 0, 0, 0)$$

Furthermore, applying the FFT on the polynomial coefficients of $P_1(x)$ and $P_2(x)$ results in the values $\mathbf{u_1}$ and $\mathbf{u_2}$ of both polynomials on the 16th roots of unity in the complex plane. To get the values of the product on the 16th roots of unity, we multiply the corresponding values together, or in other words, multiply the kth entry of $\mathbf{u_1}$ to the kth entry of $\mathbf{u_2}$, resulting in the vector $\mathbf{u_3}$. This vector contains the values of the product polynomial on the 16th roots of unity, and therefore the coefficients $\mathbf{y_3}$ of $P_3(x)$ may be obtained by taking the IFFT of $\mathbf{u_3}$.

The above process, which is summarized on the diagram below, is identical to discrete convolution, but it translates back and forth between polynomial coefficients and polynomial values on the nth roots of unity instead of the time domain and frequency domain.



Doing the above calculations results in the following coefficient vector in R^{16} for the 14th degree product $P_3(x)$:

$$\mathbf{y_3} = (15, -14, 9, -20, -10, 80, -48, 31, -6, -5, 27, -28, 1, -7, -7, 0)$$

Therefore, we can write:

$$P_3(x) = P_1(x) * P_2(x)$$

$$P_3(x) = 15 - 14x + 9x^2 - 20x^3 - 10x^4 + 80x^5 - 48x^6 + 31x^7 - 6x^8 - 5x^9 + 27x^{10} - 28x^{11} + x^{12} - 7x^{13} - 7x^{14}$$

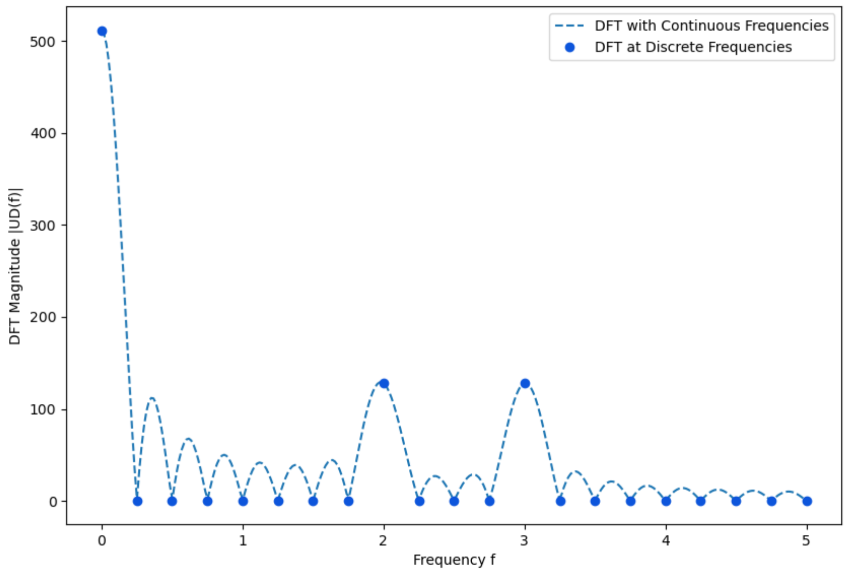
The zero at the end of $\mathbf{y_3}$ is excluded since terms with zero coefficients do not count as leading terms.

The above discussion has shown that polynomial multiplication can be efficiently performed by applying the FFT and IFFT provided zero-padding is done to bring the number of values to a power of 2. Unlike lattice multiplication or multiplication using the distributive property, this algorithm as a time complexity of $O(n \log n)$ instead of $O(n^2)$ when multiplying two $(n-1)$th degree polynomials and thus it can efficiently compute the product polynomial, particularly when the degrees of the polynomials are large.

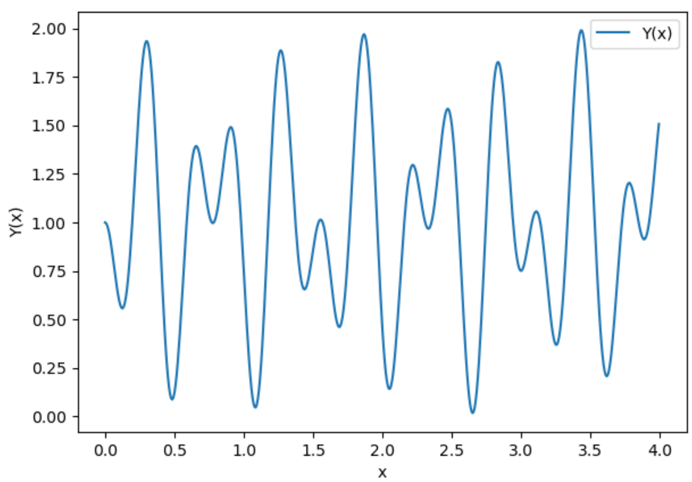
Note that the above method can be extended to multiply three or more polynomials together by taking the FFT of each coefficient vector and multiplying their corresponding values into one vector before applying the IFFT.

Section 20 - Spectral Leakage and Windowing

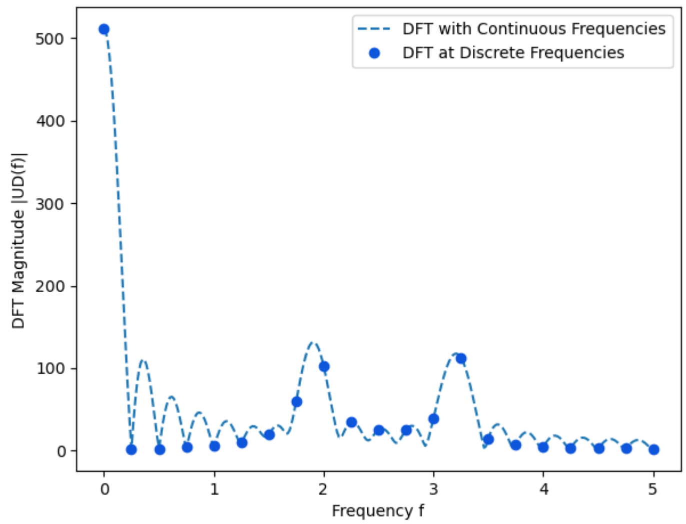
So far, the frequencies of the sinusoids used have integer frequencies. However, notice that in the example where $Y(x) = \sin^2(2\pi x) + \cos^2(3\pi x)$ and $L = 4$, the graph for the DFT at frequencies outside of the discrete frequencies forms abrupt hills in contrast to the spike produced by the points of the DFT at the discrete frequencies alone:



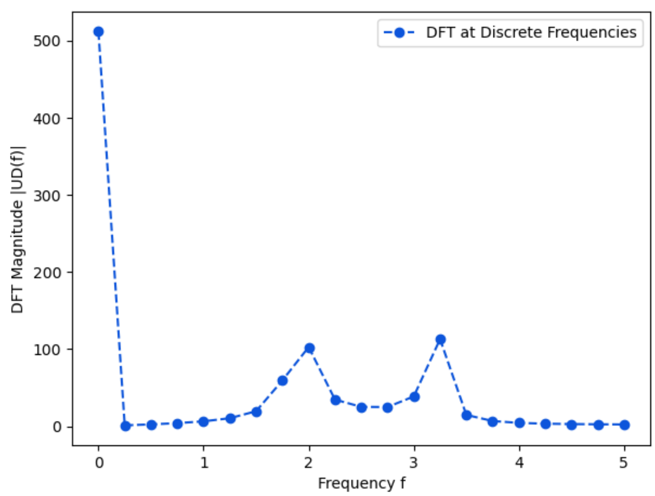
This means that if we were to shift the frequency of either sinusoid, $\sin^2(2\pi x)$ or $\cos^2(3\pi x)$, from the integer frequencies 2 and 3 Hz, respectively, into non-integer values, the values of the DFT at discrete frequencies will no longer highlight a sharp spike and thus describes the power spectrum of $Y(x)$ much less accurately. This is indeed the case when we have $Y(x) = \sin^2(1.9\pi x) + \cos^2(3.2\pi x)$, which has frequencies 1.9 and 3.2 Hz, and whose plot when its time domain is expanded to $[0, L]$, where $L = 4$, is shown below:



... and whose power spectrum at $L = 4$ is shown below:

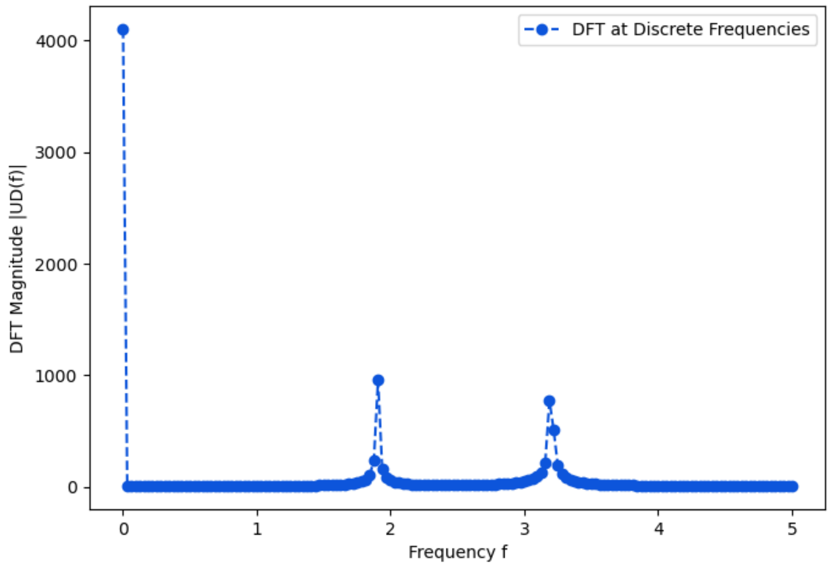


If we exclude the values of the DFT at continuous frequencies, we see that the plot of the DFT no longer shows a sharp spike:

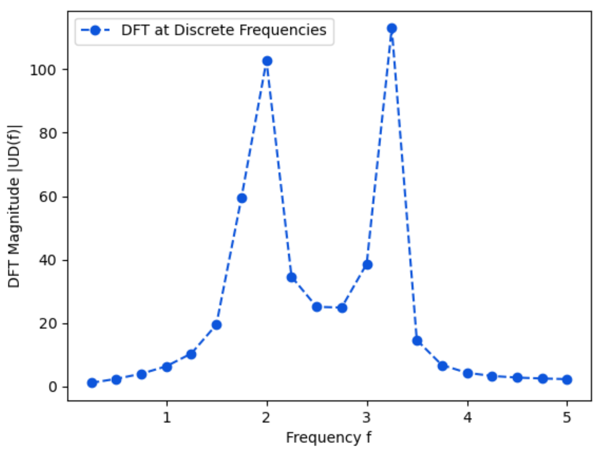


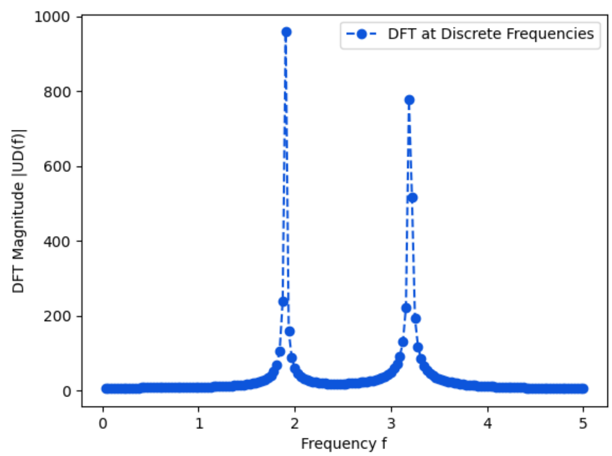
This phenomenon is called ***spectral leakage*** and it occurs when a frequency predominantly composing $Y(x)$ is not exactly one of the discrete frequencies allowed in the DFT. In this case, while the discrete frequencies $f = 2.0$, and $f = 3.25$ are close to the frequencies $f = 1.9$ and $f = 3.2$, respectively, there is still a discrepancy between both pairs of frequencies and therefore the energy at these two frequencies spreads or leaks into the surrounding frequencies.

While the effect of leakage can be reduced by increasing our frequency resolution, as the plot for the power spectrum of $Y(x)$ at $L = 32$ shows below, the spread of energy caused by leakage is still visible, in contrast to the example in the last section where the spectrum shows a sharp spike with negligible tails on its left and right:

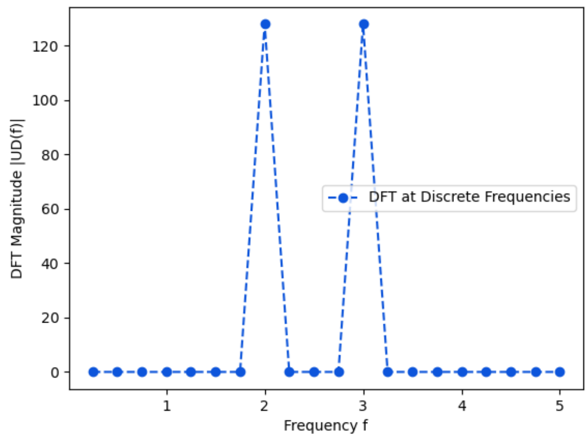


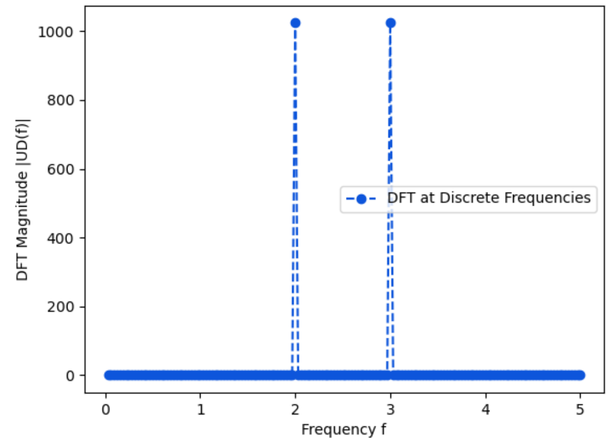
To see the spikes more clearly, let us remove $U(0)$ from both plot spectra when $L = 4$ (left) and $L = 32$ (right), and exclude continuous DFT values:





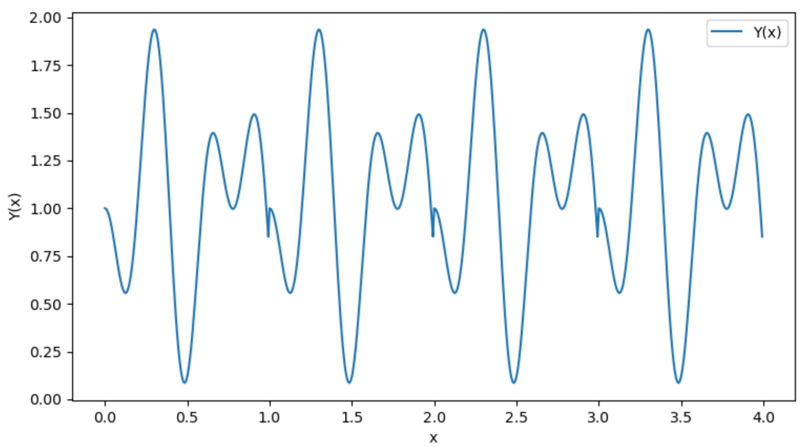
The spread of energy from the spikes is unlike the power spectrum of $Y(x) = \sin^2(2\pi x) + \cos^2(3\pi x)$ at $L = 4$ (left) and $L = 32$ (right) as shown below:



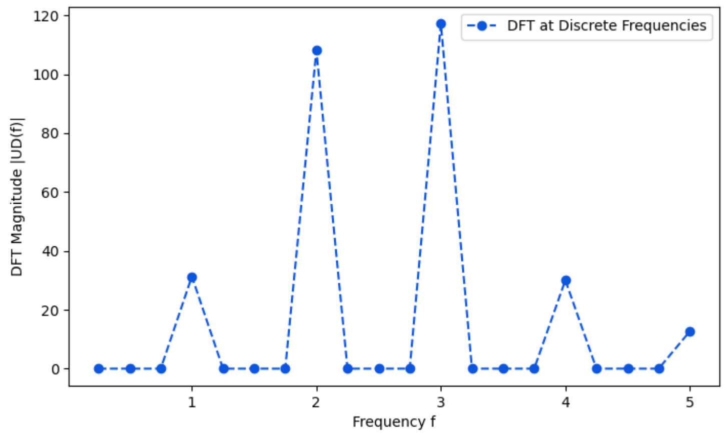


Furthermore, repeating the $Y(x)$ strictly eliminates the issue of spectral leakage since the spikes in a strictly repeating function converge sharply, but it may introduce other frequencies into the power spectrum that do not describe $Y(x)$ before it was repeated since the value of $Y(x)$ may abruptly change at certain values of x.

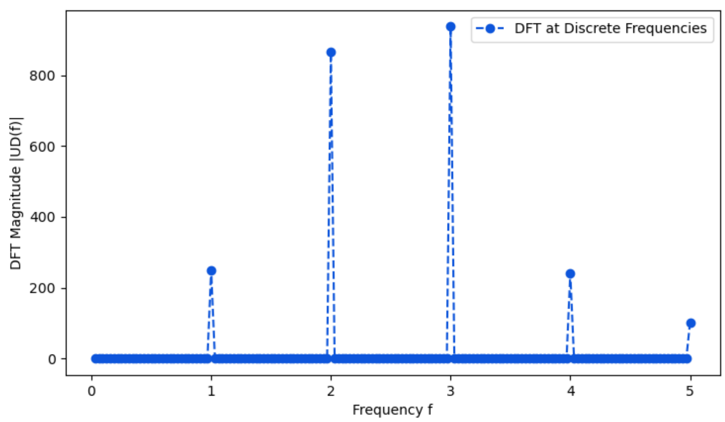
For example, if we take the previous function $Y(x) = \sin^2(1.9\pi x) + \cos^2(3.2\pi x)$ and let it strictly repeat its value on $[0, 1]$ when $L = 4$, the graph of $Y(x)$ has abrupt changes at $x = 1, 2, 3$, and 4:



and its power spectrum, excluding $U(0)$, has additional peaks at $f = 1, 4$ and 5 in addition to the main peaks at $f = 2$ and 3:



Setting $L = 32$ does not remove the introduced peaks as observed below:



A more common way to reduce leakage is by ***windowing,*** or multiplying $Y(x)$ in the time domain by a ***window function***. A window function always has a range of $[0, 1]$ and it scales the values of $Y(x)$ to zero at the boundaries of the time domain before repetition is applied while keeping values in the middle. This also ensures that $Y(x)$ repeats smoothly when it is repeated beyond the original time domain. As long as the features of the signal are repeated sufficiently in the middle of the domain, this will not affect the most dominant frequency components of $Y(x)$. However, windowing does affect the shape of the power spectrum and often helps highlight the main peaks of the spectrum while lessening the tails of each peak.

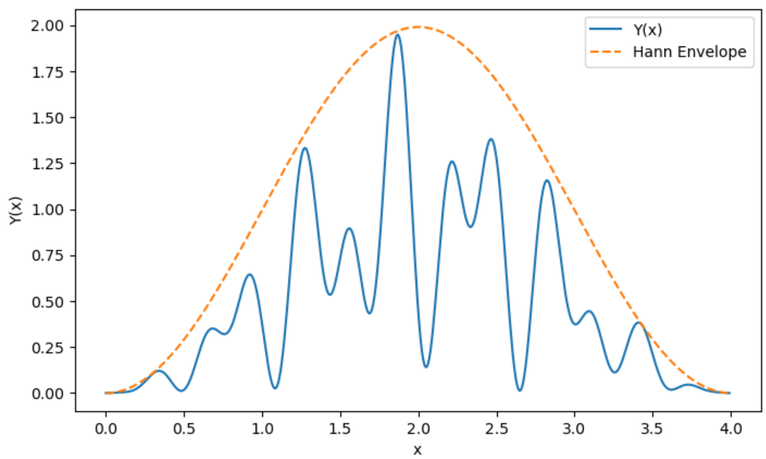
There are several windowing functions, such as the Triangular, Blackman, and Flat-top Windows, that may be used to lessen leakage, but one of the most commonly used windowing functions is the ***Hann Window***, whose value is defined as follows and whose graph follows the shape of each peak in the function $\sin^2 x$:

$$H(x, L) = \sin^2\left(\frac{\pi x}{L}\right)$$

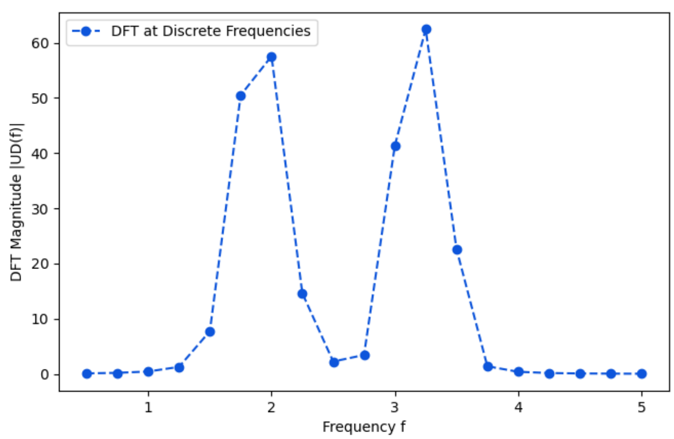
If we apply the Hann Window to the previous example without strict repetition, we have:

$$Y(x) = H(x, L) * (\sin^2(1.9\pi x) + \cos^2(3.2\pi x))$$

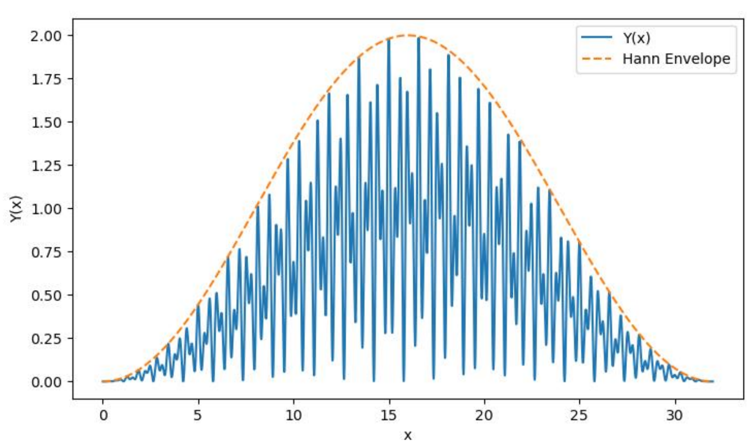
whose graph at $L = 4$ is the original function enveloped by the window:



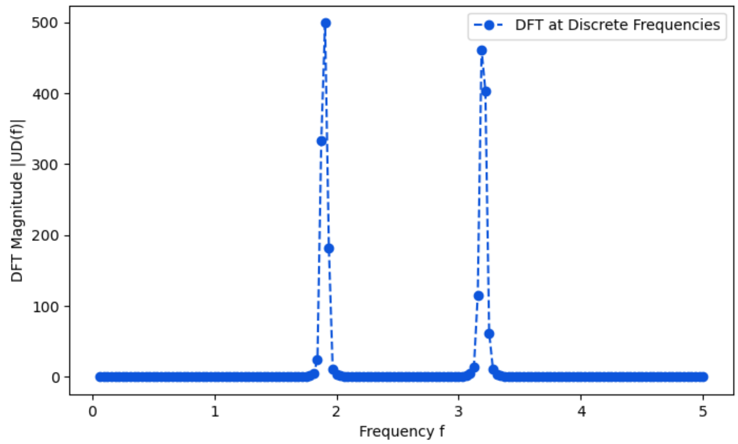
… and whose power spectrum, excluding the first two $U(f)$ values, is as follows:



When $L = 32$, the plot of $Y(x)$ becomes the following:

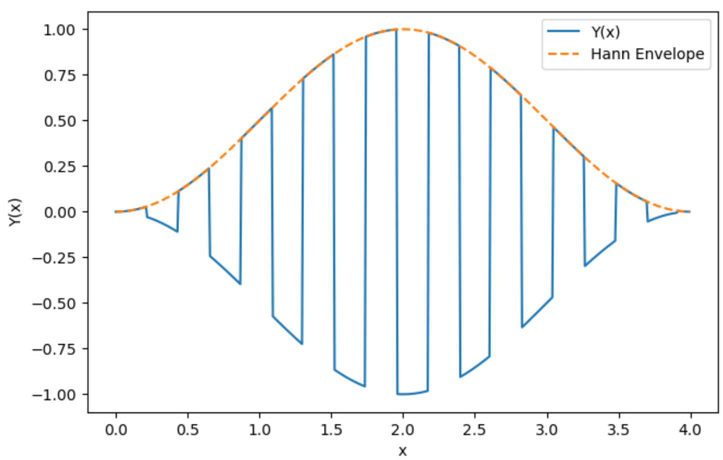


However, the shape of the power spectrum shown below differs from the case when windowing is not applied, particularly at the tails of the spikes:

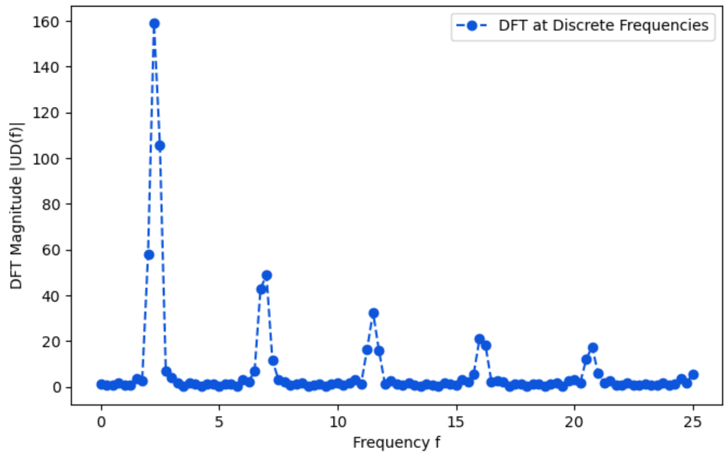


As before, the frequencies corresponding to the highest peaks in the graph are very close to $f = 1.9$ and 3.2, respectively. However, it can be observed that tails around the spikes on $f = 1.9$ and 3.2 are smaller, indicating that the effect of leakage has been reduced due to windowing.

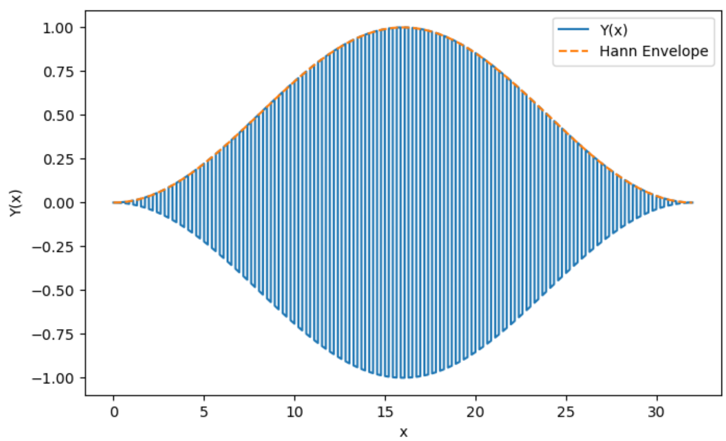
Now suppose that $Y(x) = \text{sq}(x, 2.3, 1)$ is a square wave of frequency 2.3 Hz on the time domain $[0, L]$ and we apply the Hann window as before. Then we have $Y(x) = H(x, L) * \text{sq}(x, 2.3, 1)$. When $L = 4$, then the plot of $Y(x)$ is as shown:



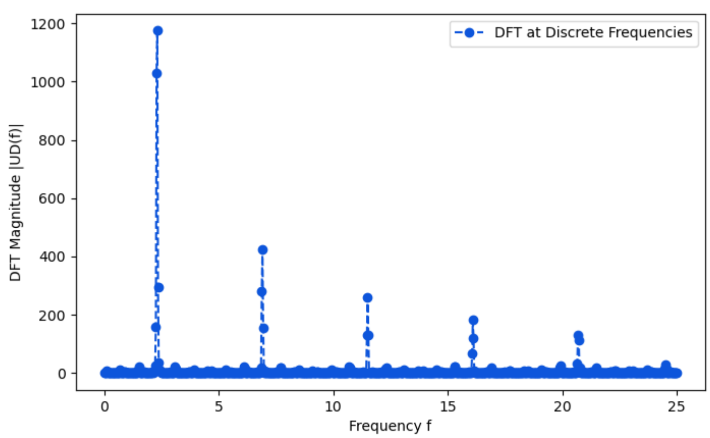
...and its power spectrum is:



When $L = 32$, the plot of $Y(x)$ becomes the following:



...and its power spectrum becomes:



In both spectra, the frequency spikes at $f = 2.3,\ 6.9,\ 11.5, \ldots$ look sharp and have negligible or no tails around them as an effect of windowing. It is also noted that the heights of the spikes approach a ratio of $1 : \frac{1}{3} : \frac{1}{5} : \frac{1}{7} : \ldots$ as the frequency resolution is increased.

Section 21 - Approximating the Continuous Fourier Transform (CFT) using the DFT for Real Functions

Recall from Section 2 that the formula for the Continuous Fourier Transform (CFT) is given by the following improper integral:

$$U(f) = \int_{-\infty}^{\infty} Y(x) \exp(-2\pi i f x) \; dx$$

If we rewrite the above expression as a Riemann Sum and treat Δx as the constant gap between discrete values in the time domain, we can see that the CFT can be approximated as a DFT multiplied by Δx:

$$U(f) = \sum_{k=0}^{\infty} Y(x_k) \exp(-2\pi i f x) \, \Delta x$$

$$U(f) = \left(\sum_{k=0}^{\infty} Y(x_k) \exp(-2\pi i f x) \right) \Delta x$$

$$\text{CFT} \approx \text{DFT} * \Delta x$$

However, since $\Delta x = \frac{1}{s}$, or in other words the gap between values in the time domain is always the reciprocal of the sampling rate, the CFT can be approximated as the DFT divided by s:

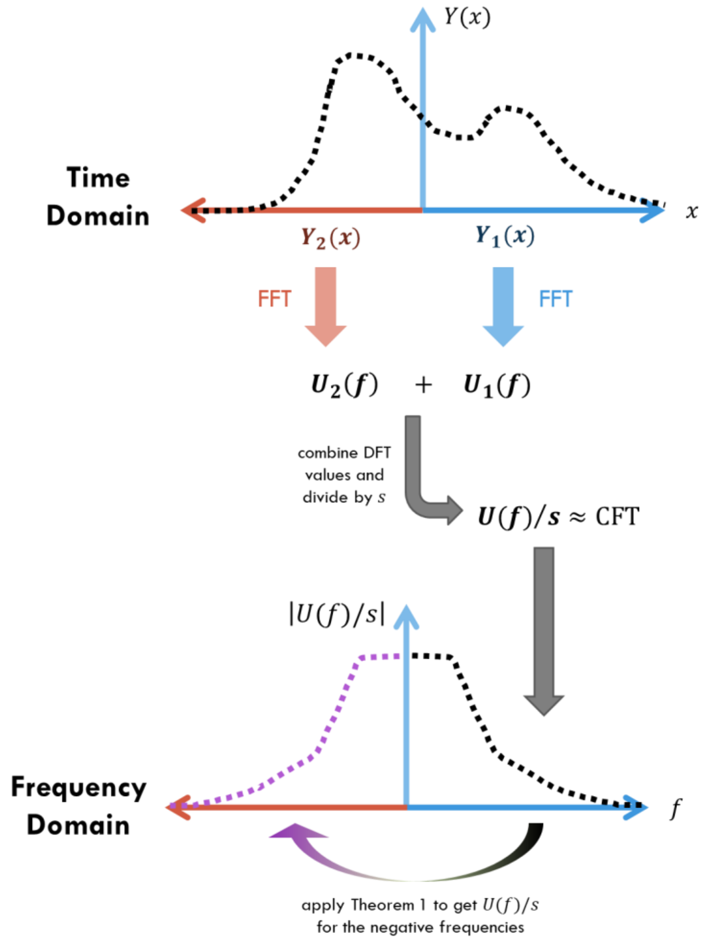
$$\text{CFT} \approx \frac{\text{DFT}}{s}$$

It is important to note, however, that the time domain of a CFT is always all real numbers or $(-\infty, \infty)$. Fortunately, the DFT may be expanded onto an interval of $[-L, L]$ by expanding the time domain beginning from the origin $x = 0$ and into the positive side $[0, L]$ and negative side $[-L, 0]$ of $[-L, L]$.

This suggests the following procedure to approximate the CFT of a function $Y(x)$ in the time domain by using the ***scaled DFT function*** $U(f)/s$:

1. Select a time domain $[-L, L]$ for $Y(x)$ with x-values sampled at a sampling rate of s.
2. Evaluate $Y(x)$ on the positive side $[0, L]$ and take the DFT $U_1(f)$ of these values.
3. Evaluate $Y(x)$ on the negative side $[-L, 0]$ and take the DFT $U_2(f)$ of these values.
4. Sum of the corresponding DFT values of $U_1(f)$ and $U_2(f)$ to get the overall DFT value $U(f) = U_1(f) + U_2(f)$ of $Y(x)$.
5. Multiply $U(f)$ by $1/s$ to get an approximation for the CFT of $Y(x)$ on the frequency domain $[0, s]$.

The diagram below visualizes the above procedure:



We shall demonstrate that the above process produces an approximation to the true continuous Fourier Transform by taking certain functions $Y(x)$ and comparing their DFT values computed from the above procedure at sampling rate $s = 256$ with the values of their Fourier Transform Pairs. Six of such common pairs are written in the following table:

FUNCTION NAME	TIME DOMAIN - $Y(x)$	FREQUENCY DOMAIN - $U(f)$
Symmetric Decaying Exponential	$Y(x) = \exp(-a\lvert x\rvert)$	$U(f) = \dfrac{2a}{a^2 + \omega(f)^2}$
Right-sided Decaying Exponential	$Y(x) = \exp(-ax)\, u(x)$	$U(f) = \dfrac{1}{a + i\omega(f)}$

Bell Curve / Gaussian Curve	$Y(x) = \exp\left(\frac{-x^2}{2a^2}\right)$	$U(f) = a\sqrt{2\pi}\ \exp\left(-\frac{1}{2}\ a^2\omega(f)^2\right)$
Right-sided Cosine Function	$Y(x) = \cos(ax)\,u(x)$	$U(f) = \frac{\pi}{2}\left(\delta\left(f - \frac{a}{2\pi}\right) + \delta\left(f + \frac{a}{2\pi}\right)\right) + \frac{i\omega(f)}{a^2 - \omega(f)^2}$
Box Function around $\boldsymbol{x = 0}$	$Y(x) = \mathrm{box}(x, -0.5, 1, 1)$	$U(f) = \mathrm{sinc}\left(\frac{\omega(f)}{2}\right)$
Right-sided Exponentially Decaying Cosine Function	$Y(x) = \cos(ax)\exp(-bx)\ u(x)$	$U(f) = \frac{b + i\omega(f)}{a^2 + \left(b + i\omega(f)\right)^2}$

where i represents the imaginary number:

$$i = \sqrt{-1}$$

$u(x)$ represents the Heaviside Step Function:

$$u(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{otherwise} \end{cases}$$

$\omega(f)$ represents the angular frequency ω (in radians per second) for a given frequency value f (in cycles per second) in the frequency domain:

$$\omega(f) = 2\pi f$$

$\delta(f - c)$ represents the Dirac-Delta Function, which is simply an infinite spike at $f = c$ and corresponds to the DFT or CFT detecting a frequency of c Hz,

and $\mathrm{sinc}(f)$ is the sinc function:

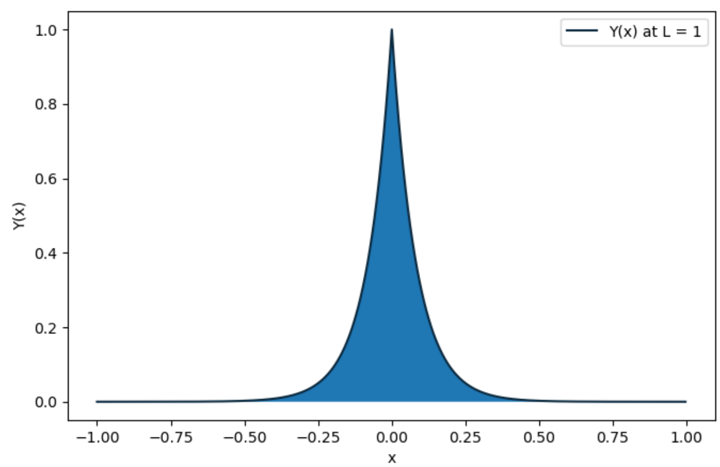
$$\mathrm{sinc}(f) = \frac{\sin f}{f}$$

Examples are given in Section 22.

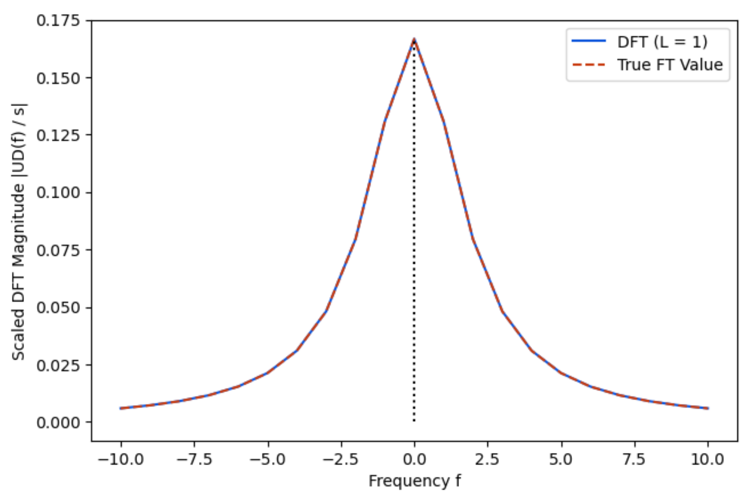
Section 22 - Examples on Approximating the CFT using DFT

Example 1 – Symmetric Decaying Exponential

Suppose $Y(x) = \exp(-a|x|)$, where $a = 12$. The graph for $Y(x)$ for a time domain of $[-L, L]$, where $L = 1$, is shown below:



With negative frequencies included, the power spectrum of $Y(x)$ at $L = 1$ is shown below, with the blue smooth curve representing the magnitude of the scaled DFT values ($\text{DFT} * \Delta x$ or DFT/s) obtained from applying the above procedure, and the red dashed curve representing the magnitude of the CFT values evaluated from the Fourier Transform pair of $Y(x) = \exp(-a|x|)$:



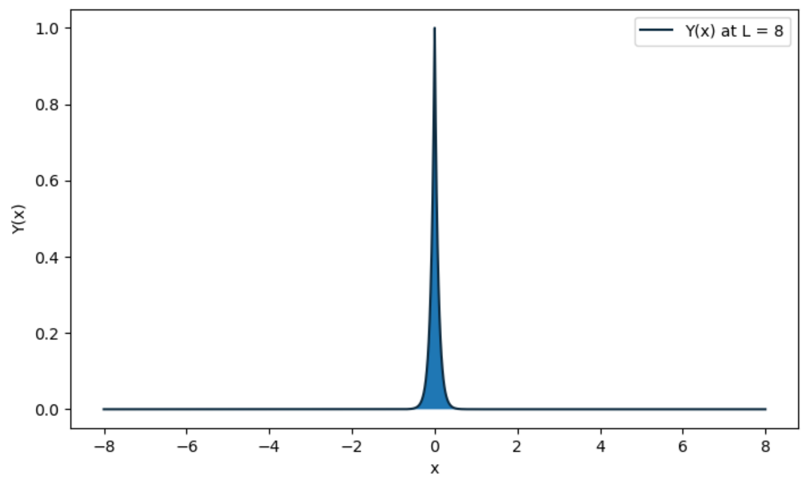
As the red and blue curves above overlap, it is observed that the DFT values are exactly the same with the CFT values for this specific case.

Moreover, notice that the power spectrum is symmetric about $f = 0$. This is implied by Theorem 1, which is restated below, and this will always be the case for the power spectra of all real-valued functions since a complex number and its conjugate always have the same magnitude:

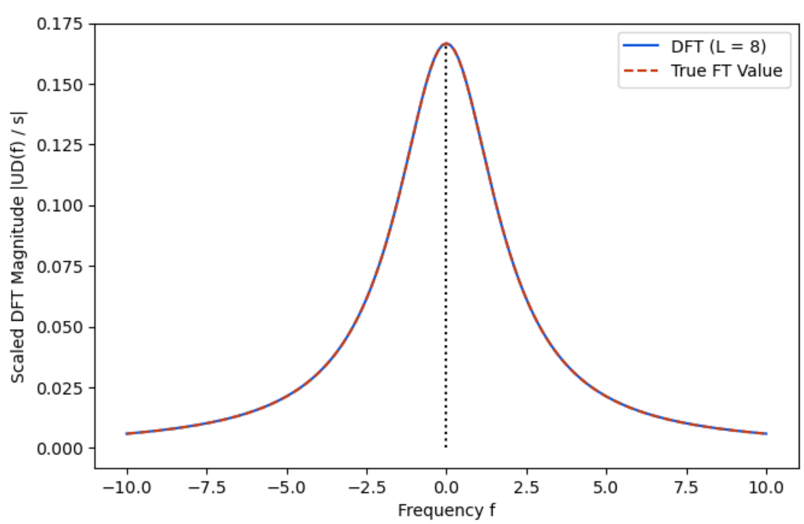
> *For any frequency $f = m$, the Fourier Transform or Discrete Fourier Transform of a real-valued function $Y(x)$ at the negative frequency $f = -m$ is the complex conjugate of the FT or DFT at $f = m$.*

The above procedure alone computes the Fourier Transform only for non-negative frequencies or for the frequency domain $[0, s]$. However, Theorem 1 allows the Fourier Transform to be known for negative frequencies given its values for positive frequencies.

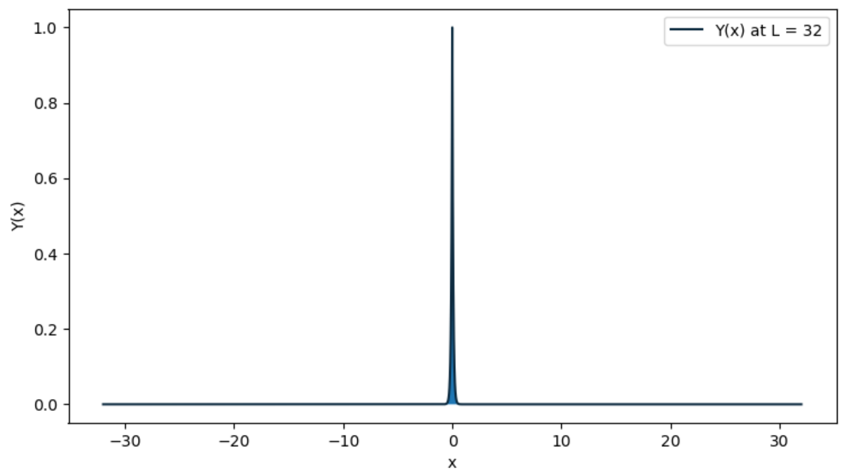
Next, we set $L = 8$, with $Y(x)$, whose plot is shown below, now having a time domain of $[-8, 8]$:



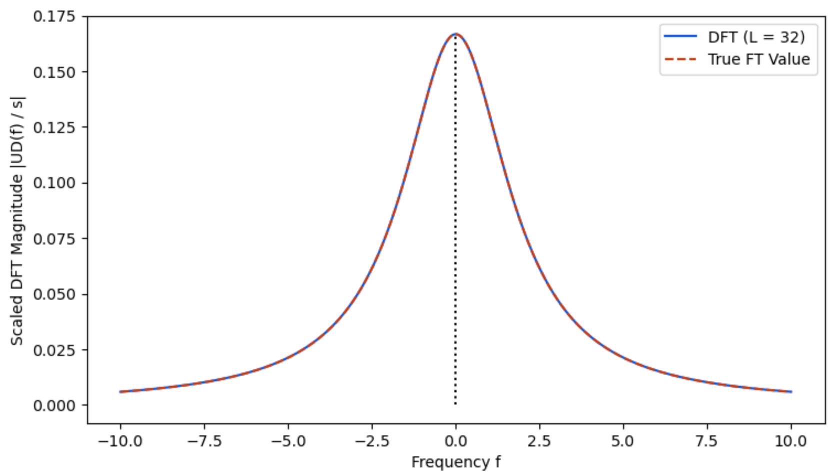
Once again, the power spectrum below shows that the DFT magnitudes are equal to the CFT magnitudes:



Finally, we set $L = 32$, or a time domain of $[-32, 32]$, giving the following plot for $Y(x)$:



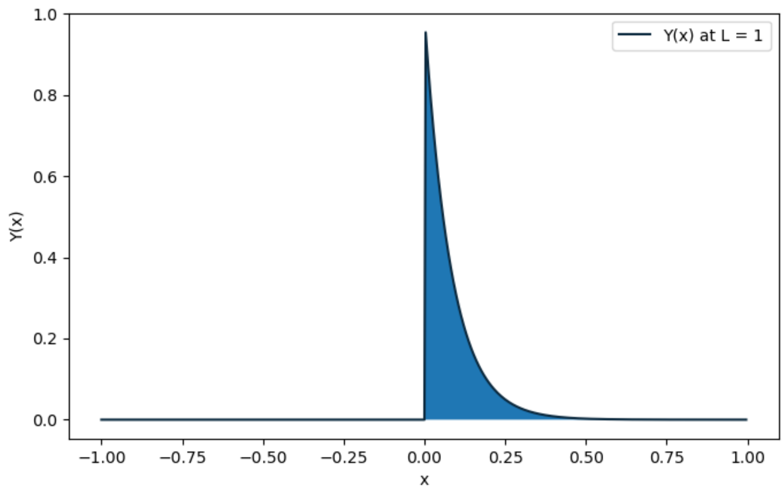
and the power spectrum for $Y(x)$ is virtually unchanged:



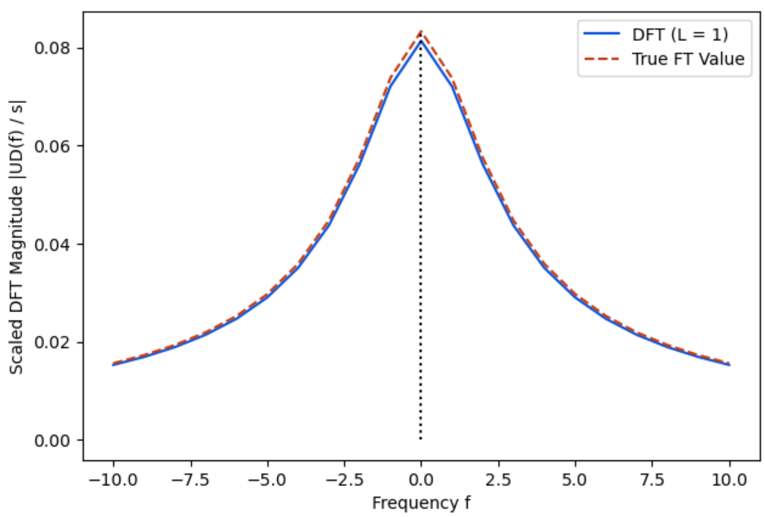
As we will see in later examples, when L approaches infinity, or as the time domain $[-L, L]$ approaches the set of real numbers $(-\infty, \infty)$, the scaled DFT values approach the true CFT values and therefore the DFT approximates the CFT better.

Example 2 – Right-sided Decaying Exponential

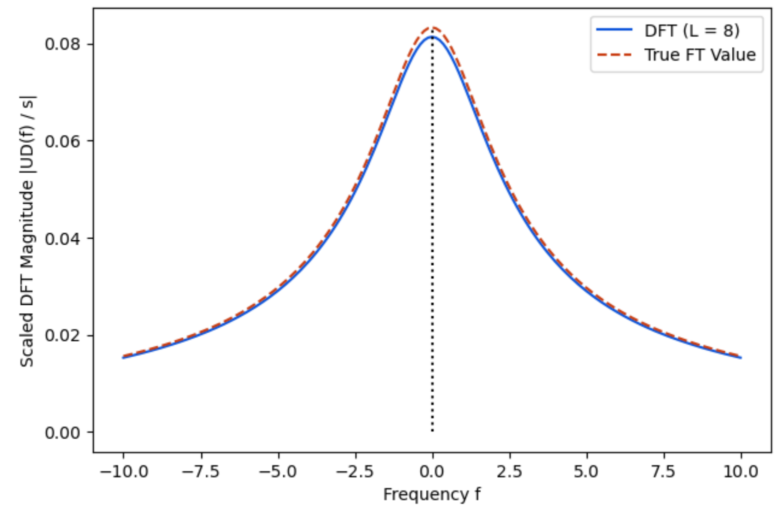
Suppose $Y(x) = \exp(-ax)\,u(x)$, where $a = 12$. The graph for $Y(x)$ for a time domain of $[-L, L]$, where $L = 1$, is shown below, with the step function $u(x)$ scaling $Y(x)$ in such a way that $Y(x) = 0$ whenever $x \leq 0$:



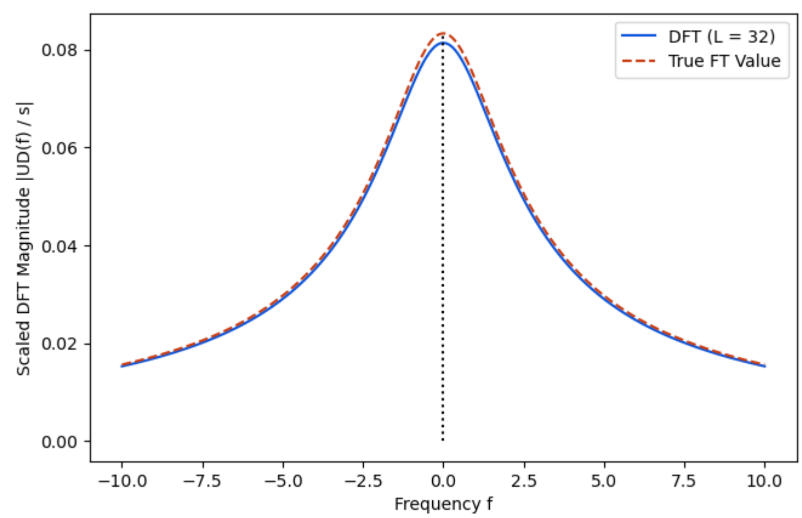
When $L = 1$, the power spectrum of $Y(x)$ shows that while the two sets of values are close, there are small discrepancies, especially at frequencies near $f = 0$, between the scaled DFT values (in blue) and the true CFT values (in red), unlike in the previous example:



When $L = 8$ or when the time domain is expanded to $[-8, 8]$, the discrepancy between values is still visible:

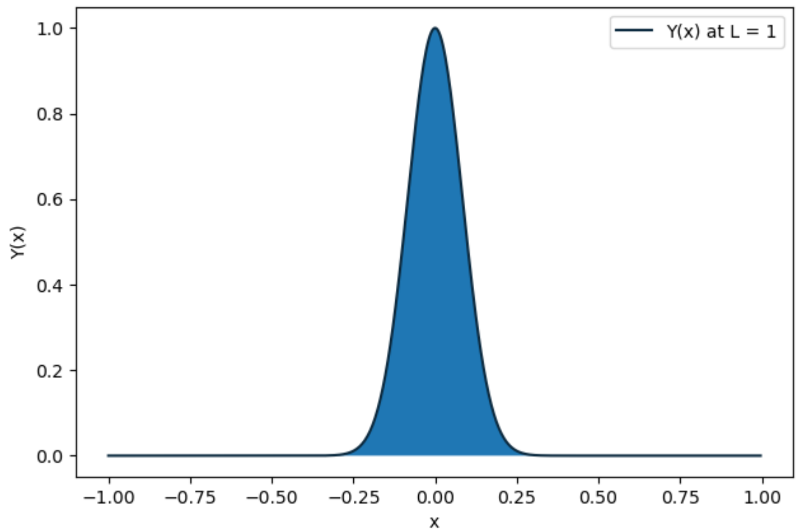


The same scenario occurs when $L = 32$ or when the time domain is further expanded to $[-32, 32]$:

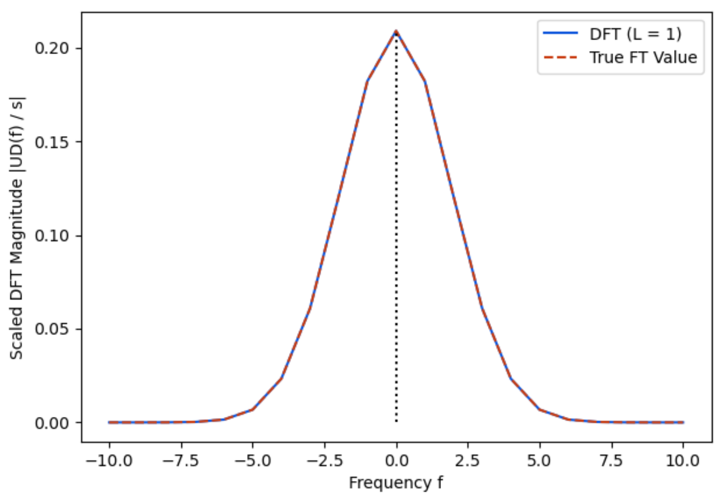


<u>Example 3 – Bell Curve / Gaussian Curve</u>

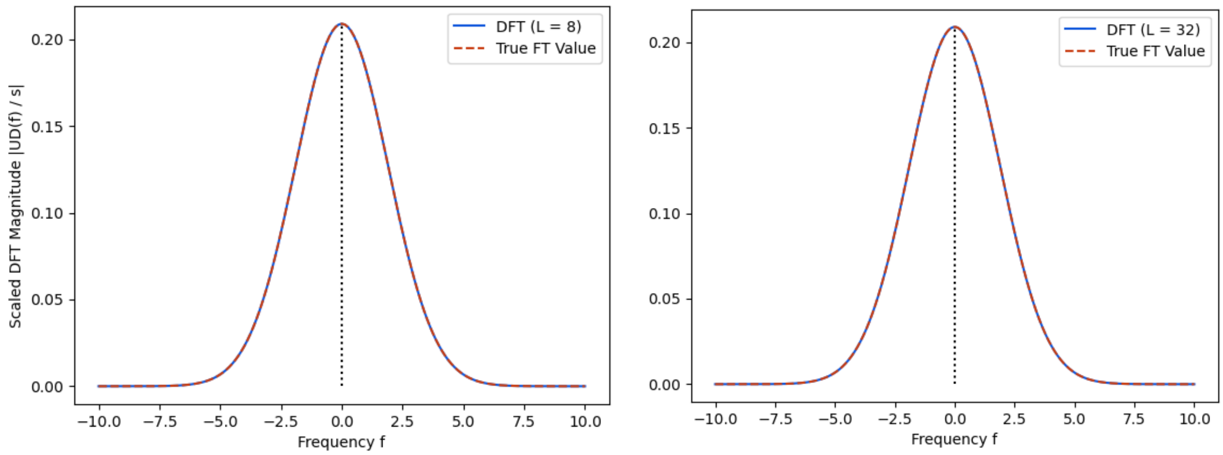
Suppose $Y(x) = \exp\left(-\frac{x^2}{2a^2}\right)$, where $a = \frac{1}{12}$. Below is the plot for $Y(x)$ on $[-1, 1]$ or when $L = 1$:



Similar to the first example, the scaled DFT values seem to agree exactly with the true CFT values according to the power spectrum at $L = 1$:



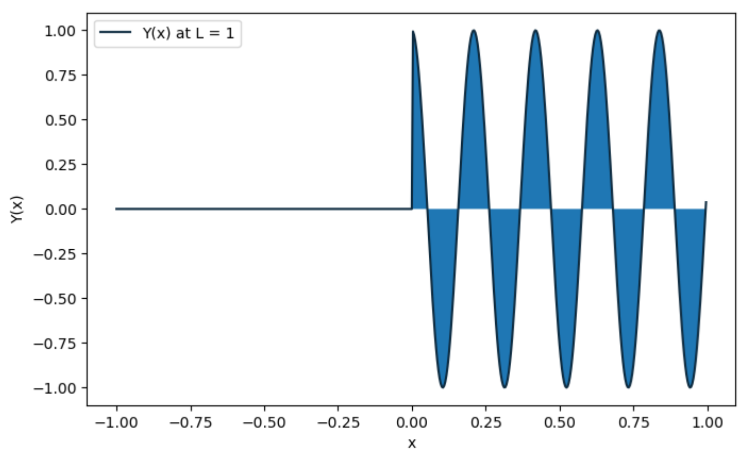
A similar case occurs when $L = 8$ and $L = 32$:



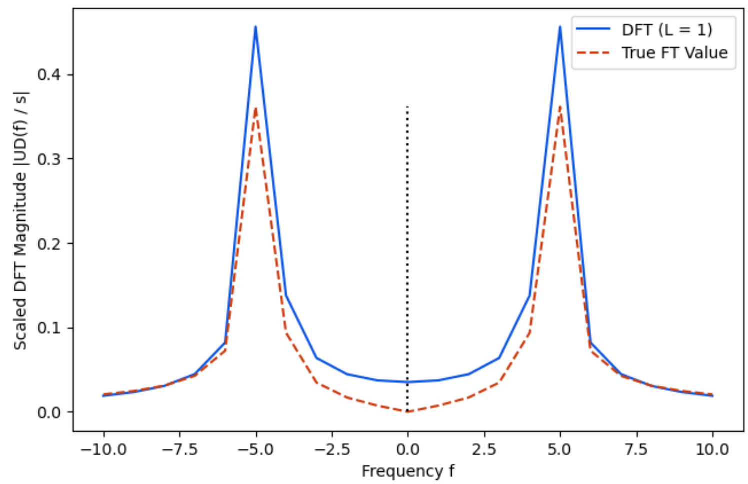
Notice from the shape of the power spectra that the Fourier Transform of a bell curve in the time domain is also another bell curve but in the frequency domain. This is a consequence of the fact that the expressions for both $Y(x)$ and its Fourier Transform pair have the form $A\exp(-Bx^2)$ or $A\exp(-Bf^2)$, where $A, B > 0$ are constants.

Example 4 – Right-sided Cosine Function

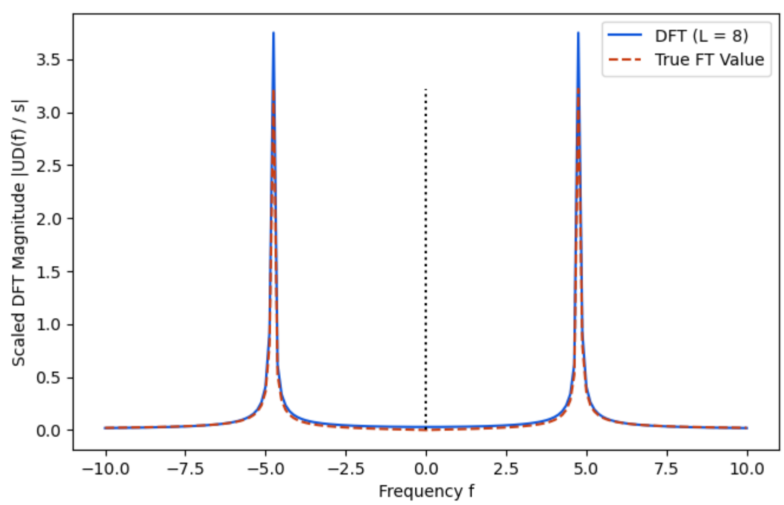
Suppose that $Y(x) = \cos(ax)\ u(x)$, where $a = 30$, and $Y(x)$ represents half of a cosine function with frequency $f = \frac{30}{2\pi} \approx 4.77$ Hz. The plot for $Y(x)$ on the time domain $[-1, 1]$ or when $L = 1$ is shown below:



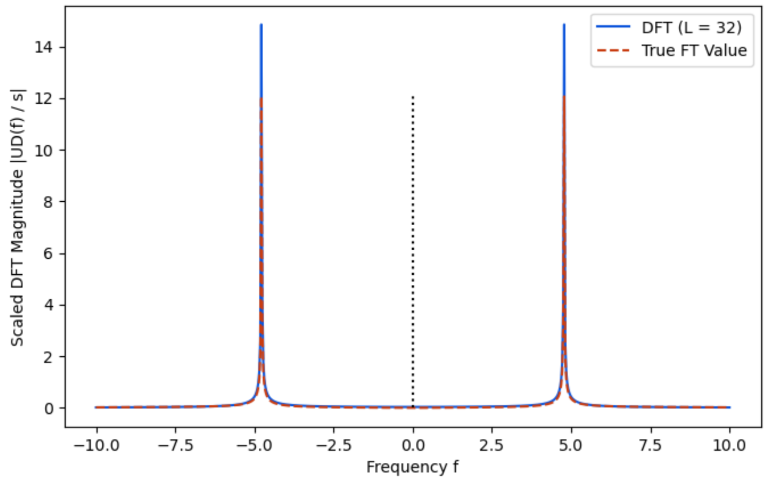
Since $Y(x)$ has a 4.77 Hz wave, we expect its power spectrum to spike at the detected frequencies $f = 4.77$ Hz and $f = -4.77$ Hz. As the spectrum below shows, this is indeed the case, but since $Y(x)$ is not completely a cosine wave (it is zero when $x \leq 0$), the spikes also extend slightly to the surrounding frequencies:



Moreover, there are large discrepancies between the scaled DFT values and the true CFT value despite both curves having similar shapes. This problem is quickly reduced as the time domain is expanded to $[-8,8]$ or L is set to 8, however, as shown in the following power spectrum:



Discrepancies are further reduced when $L = 32$:



Notice that unlike in the previous three examples, the heights of the spikes keeps on increasing as the time domain is expanded further. This is due to the fact that the Fourier Transform has detected frequencies $f = \pm 4.77$ Hz in $Y(x)$ and thus the true CFT has "infinite spikes" at $f = \pm 4.77$ Hz.

In the following expression for the FT pair of $Y(x)$, the infinite spikes at $f = \frac{a}{2\pi}$ and $f = -\frac{a}{2\pi}$, respectively, are symbolized by the Dirac Delta Functions $\delta(f - \frac{a}{2\pi})$ and $\delta\left(f + \frac{a}{2\pi}\right)$, while the remaining term on the right corresponds to the slight tails around the infinite spikes:

$$U(f) = \frac{\pi}{2}\left(\delta\left(f - \frac{a}{2\pi}\right) + \delta\left(f + \frac{a}{2\pi}\right)\right) + \frac{i\omega(f)}{a^2 - \omega(f)^2}, \qquad a = 30$$

If $Y(x) = \cos(ax)$ instead and $Y(x)$ were a pure cosine function, then its power spectrum would consist only of infinite spikes at $f = \frac{a}{2\pi}$ and $f = -\frac{a}{2\pi}$ with no tails around the spikes, and its Fourier Transform pair would only be:

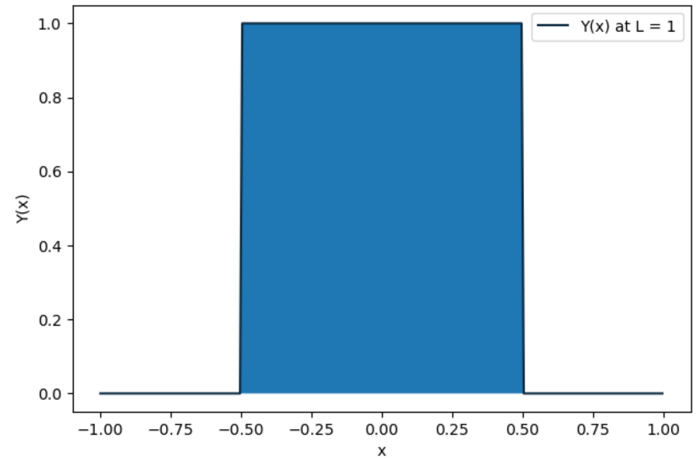
$$U(f) = \pi\left(\delta\left(f - \frac{a}{2\pi}\right) + \delta\left(f + \frac{a}{2\pi}\right)\right), \qquad a = \frac{30}{2\pi} \approx 4.77$$

Example 5 – Boxcar Function around $x = 0$

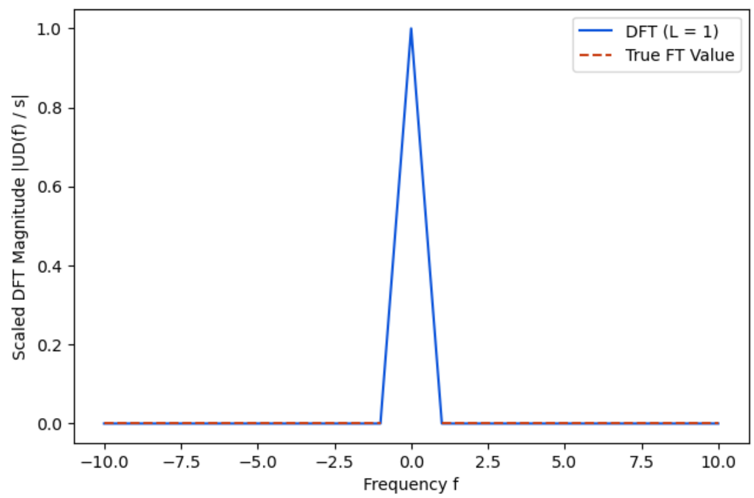
Suppose that $Y(x) = \text{box}(x, -0.5, 1, 1)$ represents a boxcar function centered around $x = 0$, or in other words:

$$Y(x) = \begin{cases} 1, & \text{if } -0.5 < x < 0.5 \\ 0, & \text{otherwise} \end{cases}$$

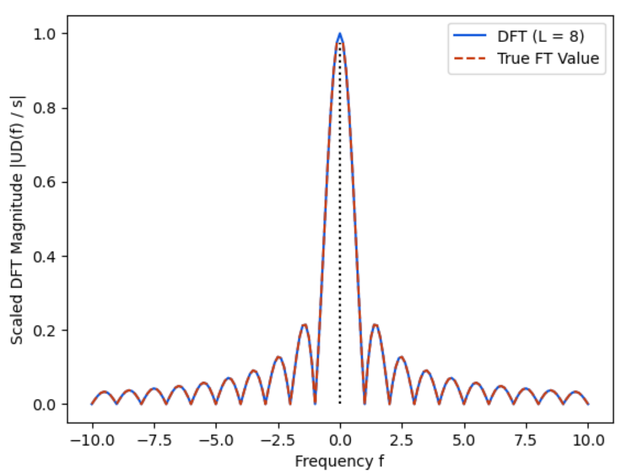
The plot for $Y(x)$ on the time domain $[-1, 1]$ or when $L = 1$ is shown below:

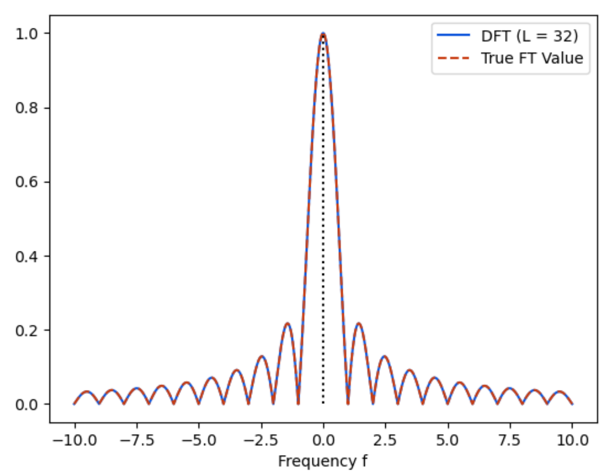


When $L = 1$, the power spectrum of $Y(x)$ shows only a spike on $f = 0$ instead of the shape of a sinc function in f:



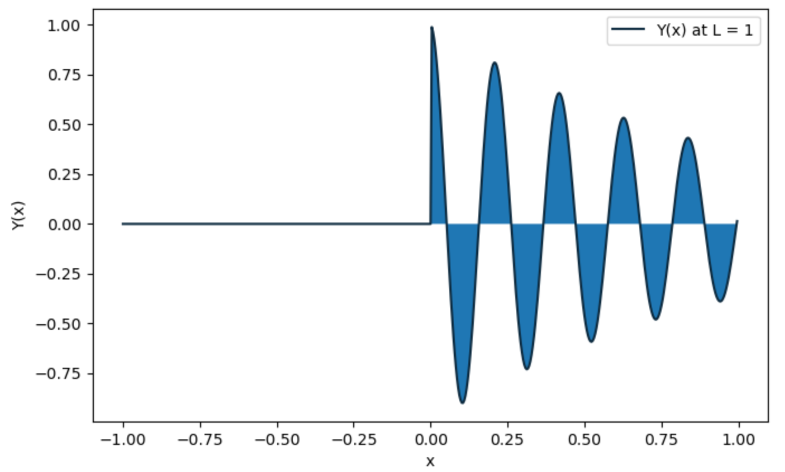
However, when $L = 8$ and $L = 32$, the power spectrum shows the shape of a sinc function, with the scaled DFT values agreeing exactly to the computed CFT values, as evidenced in both spectra:



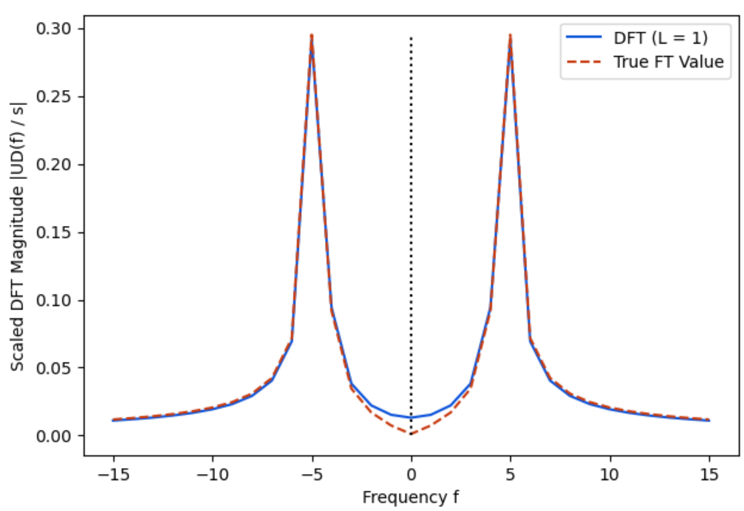


Example 6 – Right-sided Exponentially Decaying Cosine Function

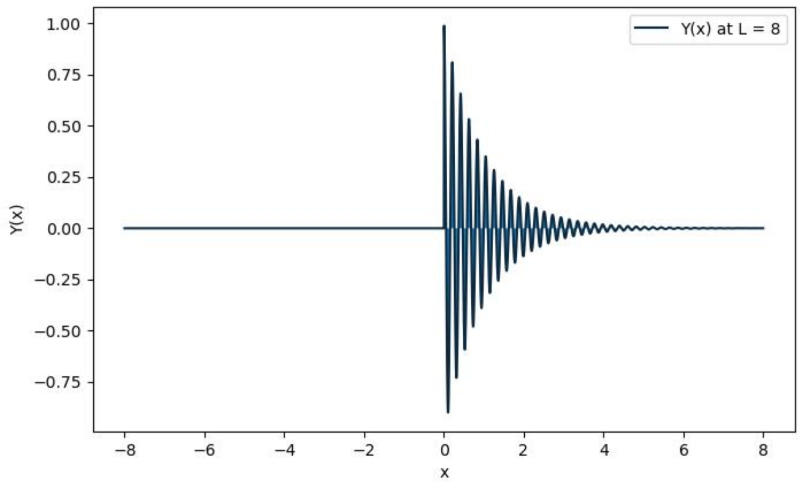
Suppose that $Y(x) = \cos(ax)\exp(-bx)\,u(x)$, where $a = 1$ and $b = 30$. This represents an exponentially decaying cosine function with frequency $f = \frac{30}{2\pi} \approx 4.77$ Hz. The plot for $Y(x)$ on a time domain of $[-1, 1]$ or $L = 1$ is shown below:



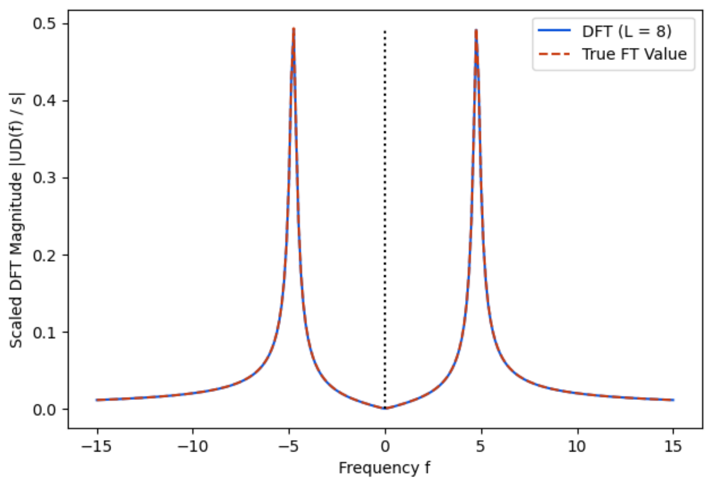
The power spectrum for $Y(x)$ below shows that the DFT detects frequencies $f = \pm 4.77$ Hz, as indicated by the peaks at these frequencies. However, there is also some discrepancy between the scaled DFT values and the true CFT values especially at $f = 0$, as with the fourth example:



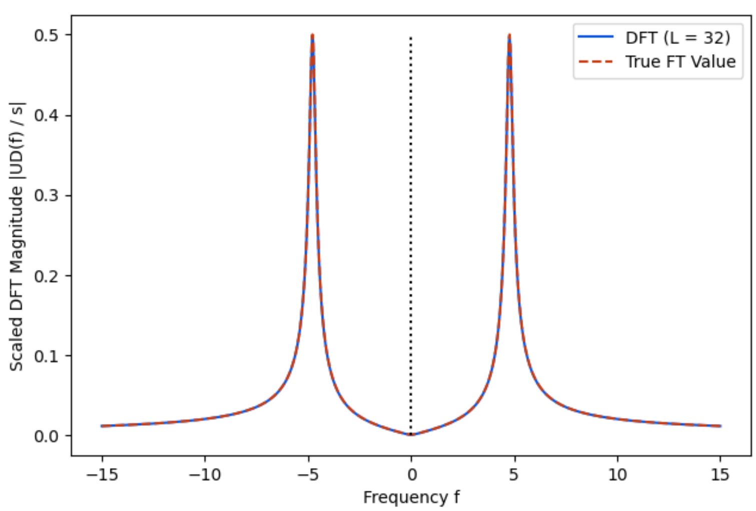
When the time domain is expanded to $[-8, 8]$ or when $L = 8$, the plot for $Y(x)$ now becomes:



and the scaled DFT values begin to agree well with the true CFT values of $Y(x)$, as evidenced in the following power spectrum:



A similar case occurs in the power spectrum when the time domain is further expanded to $[-32, 32]$ or when $L = 32$:



It is worth noting that the peaks, corresponding to the detected frequencies $f = \pm 4.77$ Hz in the power spectrum of $Y(x) = \cos(ax)\exp(-bx)\ u(x)$ approach a finite height of around 0.5 units as the time domain expanded away from $x = 0$. This is unlike the power spectrum of $Y(x) = \cos(ax)\,u(x)$ shown in the fourth example, where the spikes at $f = \pm 4.77$ Hz approach an infinite height as the time domain expands. In both cases, however, the spikes have light tails around them as a consequence of the step function $u(x)$.

Appendix 1 - Compilation of Formulas and Procedures

Euler's Formula

$$e^{i\theta} = \cos\theta + i\sin\theta$$

$$e^{-i\theta} = \cos\theta - i\sin\theta$$

Discrete and Continuous Fourier Transforms using Indexed Values

$$\text{DFT:} \qquad U(f_j) = \sum_{k=0}^{n-1} Y(x_k)\exp(-2\pi i f_j x_k)$$

$$\text{CFT:} \qquad U(f) = \mathcal{F}\{Y(x)\} = \int_{-\infty}^{\infty} Y(x)\exp(-2\pi i f x)\ dx$$

Assumptions regarding DFT

a. *The DFT is defined only on n discrete values of x and n discrete values of f.*

b. *Each input value for the DFT is associated with the value of $Y(x)$ in the time domain and each output value is associated with the value of the DFT $U(f)$ in the frequency domain.*

c. *At the first nonzero discrete frequency, f_1, wrapping the function in the complex plane results in a graph that covers exactly 1 revolution around the origin.*

d. *The jth frequency f_j is the jth scalar multiple of the first frequency f_1, that is:*

Inner Product on Functions

$$\text{Discrete Inner Product:} \qquad \langle \mathbf{f}, \mathbf{g} \rangle = \sum_{k=1}^{n} f(x_k)g(x_k)$$

$$\text{Continuous Inner Product:} \qquad \langle \mathbf{f}, \mathbf{g} \rangle = \int_{a}^{b} f(x)g(x)\ dx$$

Inner Product Interpretation of DFT and CFT

$$\text{CFT:} \quad U(f) = \langle \mathbf{Y}, \mathbf{c} \rangle + i\langle \mathbf{Y}, \mathbf{s} \rangle = \left(\int_{-\infty}^{\infty} Y(x)\cos(-2\pi f x)\ dx\right) + i\left(\int_{-\infty}^{\infty} Y(x)\sin(-2\pi f x)\, dx\right)$$

$$\text{DFT:} \quad U(f_j) = (\mathbf{y}\cdot\mathbf{c}) + i(\mathbf{y}\cdot\mathbf{s}) = \left(\sum_{k=0}^{n-1} Y(x_k)\cos(-2\pi f_j x_k)\right) + i\left(\sum_{k=0}^{n-1} Y(x_k)\sin(-2\pi f_j x_k)\right)$$

Unit Step ω

$$\omega = \exp\left(\frac{-2\pi i}{n}\right)$$

Discrete Fourier Transform on time domain $[0, 1]$ using Indices j and k:

$$U(j) = \sum_{k=0}^{n-1} Y(k) \exp\left(-\frac{2\pi ijk}{n}\right) = \sum_{k=0}^{n-1} Y(k)\, \omega^{jk}$$

Matrix Interpretation of the DFT

The DFT of values $\mathbf{y}$ *in the time domain is given by values* $\mathbf{u}$ *frequency domain and is given by the following linear system:*

$$\begin{bmatrix} U(f_0) \\ U(f_1) \\ U(f_2) \\ \vdots \\ U(f_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} Y(x_0) \\ Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_{n-1}) \end{bmatrix}$$

$$\mathbf{u} = F_n \mathbf{y}$$

where F_n *is the* n*th order Fourier Matrix whose entries* $[F_n]_{ij}$ *are given by:*

$$[F_n]_{ij} = \omega^{ij}, \qquad i, j = 0, 1, 2, \ldots n-1$$

Polynomial Interpretation of the DFT

Based on the matrix interpretation of the DFT, we find that the DFT values are simply the values of the polynomial $P(z)$ *on the* n *roots of unity* $z = 1, \omega, \omega^2, \omega^3, \ldots \omega^{n-1}$.

$$P(z) = Y(x_0) + z\, Y(x_1) + z^2\, Y(x_2) + \cdots + z^{n-1}\, Y(x_{n-1})$$

Radix-2 Fast Fourier Transform Recursive Relation

$$U(f_j) = \sum_{k=0}^{n-1} Y(x_k)\, \omega^{jk}$$

$$U(f_j) = \sum_{k=0}^{\frac{n}{2}-1} Y(x_{2k})\, (\omega^2)^{kj} + \omega^j \sum_{k=0}^{\frac{n}{2}-1} Y(x_{2k+1})\, (\omega^2)^{kj}$$

Appendix 2 - Compilation of Theorems

Theorem 1 – DFT of Negative Frequencies

For any frequency $f = m$, the Fourier Transform or Discrete Fourier Transform of a real-valued function $Y(x)$ at the negative frequency $f = -m$ is the complex conjugate of the FT or DFT at $f = m$.

$$U(-m) = \overline{U(m)}, \qquad \text{provided } Y(x) \text{ is real} - \text{valued}$$

Theorem 2 – Nyquist-Shannon Sampling Theorem

A periodic signal must be sampled at more than twice the highest frequency component of the signal which is to be measured.

Theorem 3 – Symmetry Fact A

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$. Then for all integers n and k:

$$\omega^k = -\omega^{\frac{n}{2}+k}$$

Theorem 4 – Symmetry Fact B

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$. Then for all integers m, n and k:

$$\omega^k = \omega^{k-mn}$$

Theorem 5 – Conjugate of a Root of Unity

Suppose that $\omega = \exp\left(-\frac{2\pi i}{n}\right)$. Then for all integers n and k:

$$\overline{\omega^k} = \omega^{n-k} = \omega^{-k}$$

Theorem 6 – DFT of a Constant Function

If $Y(x) = c$, where c is a constant, and $U(f)$ is the Discrete Fourier Transform of $Y(x)$ at n samples, then:

$$U(0) = cn, \quad U(k) = 0, \quad \text{where } k = 1, 2, 3, \dots, n-1.$$

Theorem 7 – Sampled Sine Wave Theorem

Given a sampling rate of s Hz and any integer m, a sine wave at a frequency f Hz is indistinguishable from a sine wave of frequency $f + ms$ Hz after sampling.

Theorem 8 – Sampled Cosine Wave Theorem

Given a sampling rate of s Hz and any integer m, a cosine wave at a frequency f Hz is indistinguishable from a cosine wave of frequency $f + ms$ Hz after sampling.

Theorem 9 – DFT of Frequencies above the Nyquist Frequency

Suppose $U(f)$ is the discrete Fourier transform of $Y(x)$ on the time domain $[0, 1]$ with a sampling rate of n. Let $f_N = n/2$ be the Nyquist frequency. Then at frequencies $f = 0, 1, 2, \dots, f_N - 1$ Hz below f_N:

$$U(f) = \overline{U(n-f)}$$

and:

$$|U(f)| = \overline{U(n-f)}$$

Conjecture 10 - DFT of Periodic Functions:

All periodic functions which repeat at sufficiently large integer frequencies have power spectra that consist only of spikes at multiples of a certain frequency, called the ***fundamental frequency****, of the original function.*

Theorem 11 – Linearity Property of Fourier Transform

If $Y(x) = aY_1(x) + bY_2(x)$ represents a signal in the time domain and a and b are complex scalars, then its Continuous Fourier Transform $U(f)$ is given by:

$$U(f) = aU_1(f) + bY_2(f)$$

where $U(f) = \mathcal{F}(Y(x))$, $\quad U_1(f) = \mathcal{F}(Y_1(x))$, and $U_2(f) = \mathcal{F}(Y_2(x))$.

Theorem 12 – Time-Scaling Property of the Fourier Transform

If $U_1(f) = \mathcal{F}(Y_1(x))$ and $Y_2 = Y_1(ax)$ where a is a complex scalar, then:

$$U_2(f) = \mathcal{F}(Y_2(x)) = \frac{1}{|a|} U_1\left(\frac{f}{a}\right)$$

Theorem 13 – Convolution Theorem for the Fourier Transform

Suppose $U_1(f) = \mathcal{F}(Y_1(x))$, $U_2(f) = \mathcal{F}(Y_2(x))$, and $U(f) = \mathcal{F}(Y(x))$.

Then $U(f) = U_1(f) * U_2(f)$ for all f if and only if $Y(x) = (Y_1 \oplus Y_2)(x)$ for all x.

Links to References Used

26. Complex Matrices; Fast Fourier Transform - YouTube

But what is the Fourier Transform? A visual introduction. - YouTube

The intuition behind Fourier and Laplace transforms I was never taught in school - YouTube

The Fast Fourier Transform (FFT): Most Ingenious Algorithm Ever? - YouTube

FFT Example: Unraveling the Recursion - YouTube

The Fast Fourier Transform (FFT) - YouTube

The Discrete Fourier Transform (DFT) - YouTube

Nyquist Theorem - an overview | ScienceDirect Topics

Circles Sines and Signals - Nyquist Frequency (jackschaedler.github.io)

Circles Sines and Signals - Signal Correlation (jackschaedler.github.io)

Circles Sines and Signals - Correlation With Sine and Cosine (jackschaedler.github.io)

Circles Sines and Signals - Discrete Fourier Transform Example (jackschaedler.github.io)

Fourier Transforms Properties - Tutorialspoint

algorithms - Concrete FFT polynomial multiplication example - Mathematics Stack Exchange

Convolution - Wikipedia

Convolution theorem - Wikipedia

11- Preprocessing audio data for Deep Learning - YouTube

Librosa Audio and Music Signal Analysis in Python | SciPy 2015 | Brian McFee - YouTube

Basic Sound Processing in Python | SciPy 2015 | Allen Downey - YouTube

Sampling: What Nyquist Didn't Say, and What to Do About It (wescottdesign.com)

(2) Denoising Data with FFT [Python] - YouTube

(2) Image Compression and the FFT - YouTube

Ocean waves simulation with Fast Fourier transform - YouTube

3 Applications of the (Fast) Fourier Transform (ft. Michael Kapralov) - YouTube

Fourier transforms in image processing (Maths Relevance) - YouTube

Convolutions in image processing | Week 1 | MIT 18.S191 Fall 2020 | Grant Sanderson - YouTube

Understanding the FFT Algorithm | Pythonic Perambulations (jakevdp.github.io)

A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science

Convolutional Neural Networks (CNNs) explained - YouTube

WRCPPB0503.qxd (msu.edu)

Window function - Wikipedia

Table of Fourier Transform Pairs (ethz.ch)