The DFT is defined as

\[ X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi kn}{N}}, k = 0, 1, \ldots, N - 1 \]

with inverse

\[ x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi kn}{N}}, n = 0, 1, \ldots, N - 1 \]

Properties of the DFT, continued.

8. Zero Insertion (Related to zero-padding in the frequency domain. It is also possible to use zero-padding in the frequency domain to obtain more densely spaced time samples. However, this must be done carefully to preserve important signal properties, such as recovering a real-valued signal.)

Suppose we have a length-\(N\) signal, \(x(n)\), with \(N\)-point DFT \(X(k)\). A new signal, \(y(n)\), is formed by inserting \(L - 1\) zeros in between every sample of \(x(n)\). This process is known as up-sampling (or interpolation) by \(L\), with the mathematical description \(y(n) = \begin{cases} x \left( \frac{n}{L} \right), & n = 0, L, 2L, \ldots \\ 0, & \text{otherwise.} \end{cases} \)
Plainly, there is no additional information in the up-sampled signal. Plainly, also, the data rate has increased by a factor of $L$.

Now, determine the $LN$-point DFT of $y(n)$.

$$Y(k) = \sum_{n=0}^{LN-1} y(n)e^{-j2\pi nk/LN}$$

$$= \sum_{n=rL, r=0,...,N-1}^{LN-1} x(n/L)e^{-j2\pi nk/LN}$$

$$= \sum_{r=0}^{N-1} x(r)e^{-j2\pi rk/N} = X(k), k = 0, \cdots, LN - 1$$

Recall that for $k$ outside the range $\{0, ..., N - 1\}$, the $X(k)$ just repeat, that is, $Y(k) = X(k \mod N), k = 0, \cdots, LN - 1$. Hence, $Y(k)$ is $L$ repetitions of $X(k)$, e.g., $Y(k) = X(0), \cdots, X(N - 1), X(0), \cdots, X(N - 1), \cdots$.

So, zero insertion in time yields a repetition of the DFT values in the frequency domain.
Example. Let \( x(n) = (3, 2, 1, 2) \) and find \( X(k) \) as well as \( Y(k) \) for \( L = 2 \) and \( L = 4 \). \( X(k) = (8, 2, 0, 2) \)

Dual: Suppose we do zero insertion in the DFT domain, e.g., let \( L = 2 \) and insert one zero after every sample of \( X(k) \), of assumed length \( N \).

\[
Y(k) = \begin{cases} 
X \left( \frac{k}{2} \right), & k \text{ even;} \\
0, & k \text{ odd.}
\end{cases}
\]

What is the IDFT of \( Y(k) \)?

Answer: \( y(n) = \frac{1}{2} x(n), n = 0, \ldots, 2N - 1 \), which is a periodic repetition of \( x(n) \).

\[
y(n) = \frac{1}{2} (x(0), \ldots, x(N - 1), x(0), \ldots, x(N - 1))
\]
Example. The figure below show zero insertion in the DFT domain for $L = 2$ and $L = 4$.

$$x(n) = (3, 2, 1, 2) \xrightarrow{DFT} X(k) = (8, 2, 0, 2)$$

Note that with zero insertion in the DFT domain, upon taking the inverse DFT there is a scaling by $\frac{1}{L}$. 
Zero-padding in the DFT domain and time signal interpolation.

It is possible to use zero-padding in the frequency domain and achieve interpolation of samples in the time domain. The basic idea is as follows.

Suppose we have an analog signal, $x_a(t)$, assumed to be band-limited to $1/2T$ Hz, and sample it at rate $1/T$ Hz, yielding the $N$ samples $x(n) = x_a(nT)$, $n = 0, 1, \ldots, N - 1$. From the Poisson Sum Formula, the discrete-time Fourier transform (DTFT) is then a sum of translates of $X_a(f)$, as

$$X(e^{j2\pi Tf}) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X_a\left(f - \frac{m}{T}\right).$$

This is sketched in the figure below and example shape $X_a(\omega)$.

![Figure. Example $X_a(f)$ and DTFT $X(e^{j2\pi Tf})$ (no aliasing).](image)

Now, instead of sampling at the rate $1/T$ Hz, suppose instead that the signal $x_a(t)$ was sampled at the rate $2/T$ Hz,
yielding the sampled signal $y(n) = x_a(nT/2)$, $n = 0, 1, \ldots, 2N - 1$. Clearly, every other sample of $y(n)$ is a sample of $x(n)$, i.e., $x(n) = y(2n)$, $n = 0, 1, \ldots, 2N - 1$. The DTFT of $y(n)$ is then

$$Y(e^{j2\pi fT/2}) = \frac{2}{T} \sum_{m=-\infty}^{\infty} X_a \left(f - \frac{m}{T/2}\right).$$

This is sketched in the figure below, along with $X(e^{j2\pi fT})$.

![Diagram](image)

Figure. Example $X_a(f)$ and DTFT $X(e^{j2\pi fT})$, with DTFT $Y(e^{j2\pi fT/2})$ corresponding to sampling at rate $2/T$.

Now we can construct a spectrum like $Y(e^{j2\pi fT/2})$ from $X(e^{j2\pi fT})$ if we can remove the spectral lobes of $X(e^{j2\pi fT})$ centered at the odd multiples of $1/T$ Hz. This can be done in
the DFT domain by properly using zero-padding in the frequency domain.

Derivation using DFT:
Suppose we begin with signal $x(n)$, and use zero-insertion (with $L = 2$) to form $y(n) = \begin{cases} x\left(\frac{n}{2}\right), & n = 0, 2, 4, \ldots \\ 0, & \text{otherwise.} \end{cases}$

The $2N$-point DFT of $y(n)$ is then $Y(k) = X(k)$, $k = 0, \dots, N - 1, N, \dots, 2N - 1$. That is, since $L = 2$, $Y(k)$ is just $X(k)$ followed by a repetition of $X(k)$. Example DFT spectra are shown below.

Figure. $N$-point DFT of $x(n)$ and $2N$-point DFT of $y(n)$. Observe that we can construct $Y(k)$ above simply by first computing $X(k)$, and then just repeating the $X(k)$ values. Further, we can generate a spectrum analogous to
\( Y(e^{j2\pi fT/2}) \) above by carefully using zero-padding in the frequency domain. Specifically, define

\[
X_1(k) = \begin{cases} 
X(k), & k = 0, \ldots, \frac{N}{2} - 1 \\
\frac{X(N)}{2}, & k = \frac{N}{2}, \frac{3N}{2} \\
X(k - N), & k = \frac{3N}{2}, \ldots, 2N - 1 \\
0, & \text{otherwise}
\end{cases}
\]

Written as a data sequence, this is

\[
X_1(k) = \left( X(0), \ldots, X\left(\frac{N}{2} - 1\right), 0.5X\left(\frac{N}{2}\right), 0, \ldots, 0, 0.5X\left(\frac{N}{2}\right), X\left(\frac{N}{2} + 1\right), \ldots, X(N - 1) \right)
\]

In the middle of the sequence are \( N - 1 \) zeros. The reason for this construction is to split the spectral energy at \( X\left(\frac{N}{2}\right) \) (if there is any), with half at the frequency \( 1/2T \) Hz and half at \( 3/2T \) Hz. This, combined with the use of \( N - 1 \) zeros, yields a \( X_1(k) \) that will satisfy the condition

\[
X_1(k) = X_1^*(N - k), \quad k = 0, 1, \ldots, 2N - 1,
\]

which must hold if the inverse DFT of \( X_1(k) \) is to provide a real-valued signal \( x_1(n) \). Noting that the 2N-point IDFT uses a scaling of \( 1/2N \), then just applying the IDFT to \( x_1(n) \) would yield amplitudes half the size of \( x(n) \).
The use of zero-padding in the frequency domain to interpolate by 2 in the time domain is summarized as follows.

a) Given $x(n)$ of length $N$, to interpolate by an integer factor $L > 1$, first compute the $N$-point DFT, $X(k)$.

b) Form the length $LN$ sequence (in the DFT domain)

$$X_1(k) = \left(X(0), \ldots, X\left(\frac{N}{2} - 1\right), 0.5X\left(\frac{N}{2}\right), 0, \ldots, 0, 0.5X\left(\frac{N}{2}\right), X\left(\frac{N}{2} + 1\right), \ldots, X(N - 1)\right)$$

where there are $(L - 1)N - 1$ zeros in the middle (the zero-padding).

c) Compute the $LN$-point IDFT and then scale by $L$ to form $x_1(n)$, $n = 0, 1, \ldots, LN - 1$. This yields

$$x_1(n) = \begin{cases} 
    x\left(\frac{n}{L}\right), & n = 0, L, 2L, \ldots \\
    \text{interpolated value,} & \text{otherwise.}
\end{cases}$$
Example. In the DFT figure above, the time signal is \( x(n) = (1, 2, 1, -1, 1, -1, 1, 2) \) with 8-point DFT \( X(k) = (6, 4.2426, 0, -4.2426, 2, -4.2426, 0, 4.2426) \). Then, setting \( L = 2 \), the \( X_1(k) \) signal is formed and the IDFT calculated. The signals are shown in the figure below.

Figure. Zero-padding in DFT domain to yield interpolation by two in the time domain.
9. Circular Convolution

Recall a property of the Z-transform:

\[ y(n) = x(n) * h(n) \quad \rightarrow \quad Y(z) = X(z)H(z) \]

And similarly for the DTFT

\[ y(n) = x(n) * h(n) \quad \rightarrow \quad Y(e^{j2\pi fT}) = X(e^{j2\pi fT})H(e^{j2\pi fT}) \]

However, this property does NOT hold for the DFT. Instead, there is the circular convolution property.

Let \( x(n) \) and \( h(n) \) be length-\( N \) sequences, and let \( X(k) \) and \( H(k) \) be their respective \( N \)-point DFTs. Then the \( N \)-point IDFT of \( \tilde{Y}(k) = X(k)H(k) \) is \( \tilde{y}(n) \), related to \( x(n) \) and \( h(n) \) as their circular (or periodic) convolution

\[ \tilde{y}(n) = x(n) \ast h(n). \]

Circular convolution is defined as follows. Given length-\( N \) sequences \( x(n) \) and \( h(n) \), define \( \tilde{x}(n) \) and \( \tilde{h}(n) \) as their periodic-\( N \) extensions. E.g., \( \tilde{x}(n) = \ldots, x(N-1), x(0), x(1), \ldots, x(N-1), x(0), x(1), \ldots \)

Then their circular convolution is

\[ \tilde{y}(n) = x(n) \ast h(n) = \sum_{r=0}^{N-1} \tilde{x}(r)\tilde{h}(n-r) \]

Proof.
Example. Let \( h(n) = (1, 2, -1) \), \( x(n) = (2, 2, 1) \), and find
\[ \tilde{y}(n) = x(n) \ast h(n). \] Compare to \( y(n) = x(n) \ast h(n) \).

How is circular convolution related to linear convolution? Let
\[ y(n) = x(n) \ast h(n) \] be the regular (linear) convolution and let \( \tilde{y}(n) = x(n) \odot h(n) \) be the circular convolution. Then
\[ \tilde{y}(n) = \sum_{m=-\infty}^{\infty} y(n + mN), \]
That is, \( \tilde{y}(n) \) is a sum of translates of the linear convolution result.
Proof.
Now, how can we use the DFT to do linear convolution?
Suppose that \( x(n) \) and \( h(n) \) are finite length signals, with \( x(n) \) of length \( N_x \) and \( h(n) \) of length \( N_h \). We know that the linear convolution result, \( y(n) = x(n) \ast h(n) \), must have length \( N_y = N_x + N_h - 1 \). So, choose any length \( N \geq N_x + N_h - 1 \), and append \( N - N_x \) zeros to \( x(n) \), and \( N - N_h \) zeros to \( h(n) \). Then compute the \( N \)-point DFT of each and form \( \tilde{Y}(k) = X(k)H(k) \). The \( N \)-point IDFT of \( \tilde{Y}(k) \) is then the linear convolution result. That is, in the expression

\[
\tilde{y}(n) = \sum_{m=-\infty}^{\infty} y(n + mN),
\]

there is no overlap in the translates, so \( y(n) = \tilde{y}(n) \), for \( n = 0, 1, \ldots, N_x + N_h - 2 \).

Figure. Use of DFT to perform linear convolution. Require \( N \geq N_x + N_h - 1 \) so circular convolution is equivalent to linear convolution.
Example. Let $h(n) = (1, 2, -1)$, $x(n) = (2, 2, 1)$, and find $	ilde{y}(n) = x(n) \ast h(n)$ by direct calculation, then using the DFT with DFT lengths 3, 4, 5, and 6. Compare to the linear convolution result.

For $N = 3$, $\tilde{y}(n) = (2, 5, 3)$

For $N = 4$, $\tilde{y}(n) = (1, 6, 3, 0)$

For $N = 5$, $\tilde{y}(n) = (2, 6, 3, 0, -1)$

For $N = 6$, $\tilde{y}(n) = (2, 6, 3, 0, -1, 0)$
Complexity
For certain data lengths, \( N \), the DFT has efficient algorithms for computation, generally referred to as fast Fourier transforms (FFT). A detailed complexity analysis is deferred until the FFT is discussed later. For now, the complexity of the FFT (for \( N \) a power of 2) is simply approximated as \( N \log_2 N \) additions and multiplications per \( N \)-point DFT computation. Note that a brute-force DFT computation, as

\[
X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}}, k = 0,1,\ldots, N - 1
\]

would require \( N \) multiplications and \( N - 1 \) additions per index \( k \), for a total of approximately \( N^2 \) multiplications and additions per \( N \)-point DFT.

“Fast” convolution can be performed using the method outlined in the previous pages, by using an FFT. However, this is only fast if both signals being convolved are relatively long. Analyzing the figure below (and assuming the IDFT complexity is roughly the same as the FFT complexity), determine the number of additions and multiplications for convolving two signals of length \( K \). Compare to the brute force convolution complexity, for the values \( K = 16, 64, 128, 1024, 2^{16}, 2^{20} \).
Brute force convolution (both signals of length $K$).

$$y(n) = \sum_{k=0}^{K-1} h(k)x(n - k)$$

Worst case analysis: the length of $y(n)$ is $2K - 1$. For each value of $y(n)$ the formula requires $K$ multiplies and $K - 1$ additions, for a total of $K(2K - 1)$ multiplies and $(K - 1)(2K - 1)$ additions. (Do a more detailed analysis and derive a better estimate of the complexity.)

For the DFT method, assume that $K$ zeros are appended to each of $h(n)$ and $x(n)$, and the DFT length is $N = 2K$. There are three DFT computations, $N$ multiplies to get the products $\tilde{Y}(k) = X(k)H(k)$, and $N$ multiplication with the IDFT (final scaling by $1/N$). The complexity is then estimated as $3 \times N \log_2 N$ add/multiplies, plus $2N$ multiplies. Using $N = 2K$ this yields a complexity $3 \times 2K \log_2 2K$ add/multiplies, plus $4K$ multiplies. The complexities are compared in the following table.
Complexity comparison of fast convolution vs. brute force convolution computation. $K = \text{length of } h(n) \text{ and } x(n)$.

<table>
<thead>
<tr>
<th>K</th>
<th>$K(2K - 1)$</th>
<th>$(K - 1)(2K - 1)$</th>
<th>$6K\log_2 2K + 4K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>496</td>
<td>465</td>
<td>544</td>
</tr>
<tr>
<td>32</td>
<td>2116</td>
<td>1953</td>
<td>1280</td>
</tr>
<tr>
<td>64</td>
<td>8128</td>
<td>8001</td>
<td>2944</td>
</tr>
<tr>
<td>128</td>
<td>32,640</td>
<td>32,385</td>
<td>6,656</td>
</tr>
<tr>
<td>1024</td>
<td>$2.096 \times 10^6$</td>
<td>$2.096 \times 10^6$</td>
<td>71,680</td>
</tr>
<tr>
<td>$2^{16}$</td>
<td>$8.6 \times 10^9$</td>
<td>$8.6 \times 10^9$</td>
<td>$6.95 \times 10^6$</td>
</tr>
<tr>
<td>$2^{20}$</td>
<td>$2.2 \times 10^{12}$</td>
<td>$2.2 \times 10^{12}$</td>
<td>$1.36 \times 10^8$</td>
</tr>
</tbody>
</table>

Plainly, the $N \log N$ advantage of the FFT dominates the complexity for large value of $K$.
Fast filtering of an on-going signal.
The use of the DFT for fast convolution works fine for fixed-length signals, but what if $h(n)$ is finite length, but $x(n)$ is on-going (of indefinite length)? In this case, we can partition $x(n)$ into non-overlapping blocks of length $K$, filter each block, and then combine the results. This is called the “overlap and add” method.

Below is the general block diagram for fast convolution using the DFT. Assume that $h(n)$ has length no larger than $K$, and partition $x(n)$ into segments of length $K$. Denote the $i$th segment (or block) of the $x(n)$ signal as $x_i(n)$, and given by $x_i(n) = (x(iK), x(iK + 1), \ldots, x(iK + K - 1))$

Noting that the response to block $x_i(n)$ overlaps with the response to block $x_{i+1}(n)$, the overlapping portion is saved and added in to form the on-going filter output.
DFT Resolution Theorem
Suppose analog signal $x_a(t)$ has Fourier transform $X_a(f)$. Choose any sampling rate $1/T$ Hz and integer $N$. Then the sequences $\hat{x}(n), n = 0, 1, \ldots, N - 1$ and $\hat{X}(k), k = 0, 1, \ldots, N - 1$ are an exact DFT pair, where
a. $\hat{x}(n)$ is the sample at $t = nT$ of

$$\hat{x}(t) = \sum_{r=-\infty}^{\infty} x_a(t + rNT), \quad n = 0, 1, \ldots, N - 1;$$

b. $\hat{X}(k)$ is $1/T$ times the sample at $f = \frac{k}{NT}$ of

$$\hat{X}(f) = \sum_{m=-\infty}^{\infty} X_a \left( f + \frac{m}{T} \right), \quad k = 0, 1, \ldots, N - 1.$$

Proof.
Idealized case. Suppose signal $x_a(t)$ were both

i. Bandlimited to $\frac{1}{2T}$ Hz for some value of $T$, and

ii. Duration-limited to $NT$ seconds for some $N$.

We know this can never be true, although it can be approximately correct.

Now,

$$\hat{x}(t) = \sum_{r=\infty}^{\infty} x_a(t + rNT)$$

This is a periodic extension of $x_a(t)$, since $x_a(t)$ is duration limited to $NT$ sec. There is no overlap in the sum of time translates. And

$$\hat{X}(f) = \sum_{m=\infty}^{\infty} X_a(f + \frac{m}{T})$$

This is a periodic extension of $X_a(f)$, since $X_a(f)$ is band limited. There is no overlap in the frequency translates.

Hence, the sequences $\hat{x}(n)$ and $\hat{X}(k)$ are an exact DFT pair, with $\hat{x}(n)$ and $\hat{X}(k)$ exact samples, respectively, of $x_a(t)$ at $t = nT$, and of $X_a(f)$ at $f = \frac{k}{NT}$. That is, the DFT gives exactly (scaled) samples of the analog transform, $X_a(f)$. 
In practice, there must always be some aliasing since $x_a(t)$ cannot be both duration-limited and bandlimited. Instead, one chooses $T$ small enough and $N$ large enough (or recognizes the aliasing that is present). As a rough rule of thumb, $N$ should be at least as large as the duration-bandwidth product (where, for example, the rms duration and rms bandwidth can be used).
Example 1. To illustrate some of the potential pitfalls, let \( x_a(t) = \cos(2\pi f_0 t) \). To avoid aliasing, we must select \( \frac{1}{T} > 2f_0 \). The Fourier transform is

\[
X(f) = \frac{1}{2} \left( \delta(f - f_0) + \delta(f + f_0) \right)
\]

As a specific example, select \( f_0 = 40 \text{ Hz}, \frac{1}{T} = 256 \text{ Hz}, \) and \( N = 64 \). The time samples and DFT are shown in the figure below.

Note that this plot is pretty much what we want and expect – isolated spectral peak at frequencies 40 Hz and 216 Hz (216 = 256 – 40, corresponding to the periodic extension by 256 samples of the peak at frequency -40 Hz). Unfortunately, this obvious interpretation is pretty much wrong.
To clarify what is happening, let’s repeat the process, but using a cosine of 30 Hz, so we select $f_0 = 30$ Hz, $\frac{1}{T} = 256$ Hz, and $N = 64$. The time samples and DFT are shown in the figure below.

Now the nice spectral peak has been smeared out to one roughly centered at 30 Hz. What is going on?
The DFT resolution theorem says samples at $t = nT$ of

$$\hat{x}(t) = \sum_{r=-\infty}^{\infty} x_a(t + rNT), \quad n = 0, 1, \ldots, N - 1; \quad (1)$$

are an exact DFT pair with samples at $\omega = \frac{2\pi k}{NT}$ of

$$\hat{X}(f) = \sum_{m=-\infty}^{\infty} X_a(f + \frac{m}{T}), \quad k = 0, 1, \ldots, N - 1. \quad (2)$$

Now, the sum of translates in the frequency domain is a familiar result of the time sampling process. But what about the sum of translates of the time signal? Rarely would one explicitly form this prior to acquiring the time data samples. Instead, this sum of translates is implicit, and the signal $x_a(t)$ typically represents the result of truncating an analog signal, say $x_a(t)$, to the duration $[0, NT)$ seconds. This is done implicitly using a rectangular window,

$$w(t) = \begin{cases} 1, & 0 \leq t < NT \\ 0, & \text{otherwise} \end{cases}$$

so that the time signal in the theorem above is

$$x_a(t) = x(t)w(t).$$

Since this signal is duration-limited to the range $[0, NT)$ seconds, there is no overlap in the sum of translates in (1) above.

Now, what is the Fourier transform $X_a(f)$?

We have $x_a(t) = x(t)w(t) \xrightarrow{F} X(f) \ast W(f)$. For the signal above, $x(t) = \cos(2\pi f_0 t)$, so

$$X(\omega) = \frac{1}{2} \left( \delta(f - f_0) + \delta(f + f_0) \right)$$
The rectangular window has transform
\[ W(f) = e^{-j2\pi NT/2NT} \operatorname{sinc}(f NT), \]
so
\[ X_a(f) = \frac{1}{2} (W(f - f_0) + W(f + f_0)) \]
Note that the zero-crossings of the \( \operatorname{sinc}(\ ) \) are every \( \frac{1}{NT} \) Hz.

The DFT corresponds to samples of
\[ \hat{X}(f) = \sum_{m=-\infty}^{\infty} X_a \left( f + \frac{m}{T} \right) \]
at the frequencies \( f = \frac{k}{NT} \).

Now, specialize to the case above where \( \frac{1}{T} = 256 \text{Hz}, N = 64, \) and \( f_0 = 40 \text{ Hz} \). Then \( W(f - f_0) \) has peak at \( f = f_0 = 40 \text{ Hz} \), which is an integer multiple of \( \Delta f = \frac{1}{NT} = 4 \text{ Hz} \).

Additionally, the zero-crossings of the \( \operatorname{sinc} \) pulse are exactly every 4 Hz. The figure below shows the DFT magnitudes shown previously (red dots), together with the waveform \( \frac{1}{T} W(f - f_0) \).

![DFT and \( W(f_0-2\pi f_0) \)](image-url)
Conclusion: The DFT gives (scaled) samples of the (sum of translates of the) **windowed** data. Since multiplication in time is equivalent to convolution in frequency, the best that can be hoped for is obtaining samples without aliasing of the true data spectrum, $X(f)$, **convolved** with the window spectrum, $W(f)$, i.e., of $X(f) \ast W(f)$.

Example 2. Modifying the previous example, let $x(t) = 100 \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$. Consider this as an “unknown” signal, and the goal of the DFT analysis is to try to identify the sinusoids in the signal. Note the very different amplitudes (a 40 dB difference). As before, let $\frac{1}{T} = 256$ Hz, and $N = 64$, and begin with $f_1 = 40$ Hz and $f_2 = 80$ Hz. The frequency resolution is $\Delta f = \frac{1}{NT} = 4$ Hz, and both sinusoidal frequencies are integer multiples of $\Delta f$. The DFT of $x(nT)$ is shown below. The sinusoid frequencies are apparent, but this is misleading.
Next, modify the frequencies to $f_1 = 42$ Hz and $f_2 = 82$ Hz. The DFT is plotted below. Note that in this case the low-amplitude cosine is pretty much obscured by the smeared energy of the high-amplitude cosine.

Next, we try zero-padding (length 512 DFT), with the DFT plotted below. (The plot is continuous for convenience.)

Still cannot really see the low-amplitude sinusoid.
Next, let’s modify the data by selecting a non-rectangular window, called the Hanning window, that smoothly tapers the signal to zero at the interval boundaries. This influences $X(f) \ast W(f)$ by modifying $W(f)$. Using this window, the DFT is computed and plotted below.

Comparing the rectangular and Hanning window results (using the decibel amplitude scale) shows the difference.

Rectangular window (top) and Hanning window (bottom).

There are dozens of windows that have been proposed for use in spectral analysis using the DFT. Listed below are a few.

1. Rectangular window: \( w(n) = u(n) - u(n - N) \).
2. Bartlett (triangular) window: 
   \[
   w(n) = \begin{cases} 
   \frac{2n}{N}, & 0 \leq n \leq \frac{N}{2}; \\
   2 - \frac{2n}{N}, & \frac{N}{2} + 1 \leq n \leq N - 1.
   \end{cases}
   \]
3. Hanning (von Hann) window: \( w(n) = \frac{1}{2} - \frac{1}{2} \cos \left( \frac{2\pi n}{N} \right) \).
4. Hamming window: \( w(n) = 0.54 - 0.46 \cos \left( \frac{2\pi n}{N} \right) \).
5. Blackman window:
   \[
   w(n) = 0.42 - 0.5 \cos \left( \frac{2\pi n}{N} \right) + 0.08 \cos \left( \frac{4\pi n}{N} \right).
   \]

Moving down the list above, the windows progressively taper the data to zero more smoothly as the index, \( n \), approaches 0 or \( N \).
The windows listed above, and their DFT magnitudes, are shown in the two figures below. (The DFT plots are in decibels, and truncated to -40 dB.)

Note the width of the window main spectral lobe and the relative magnitude of the first sidelobe. The main central lobe reflects the amount of local spreading due to the convolution effect in the frequency domain of the windowing in the time domain. The magnitude of the side lobes determine the amount of spreading to distant frequencies due to the convolution.
Note that the Hamming window has relatively flat side lobe peaks, with the first side lobe down about 42 dB from the main lobe peak. The Hanning window has similar main lobe width as the Hamming window, but its first side lobe is down only about 32 dB from the main lobe peak. The Blackman
window has the largest main lobe width, but the smallest side lobes.

When used for spectral analysis there is a tradeoff in choice of window. The rectangular window has the narrowest main lobe, but the largest side lobes. As the side lobe amplitude is reduced, the main lobe width gradually increases.