Section 9.3).

Now for the second part of the proof. If the input signal, f, doesn't happen to be a complex exponential, we can express it as a sum of complex exponentials by computing its DFT, F. For each value of k from 0 to $N-1$, $F[k]$ is the complex magnitude of the component with frequency k.

Each input component is a complex exponential with magnitude $F[k]$, so each output component is a complex exponential with magnitude $F[k]G[k]$, based on the first part of the proof.

Because the system is linear, the output is just the sum of the output components:

$$(f * g)[n] = \sum_k F[k]G[k]e_k[n]$$

Plugging in the definition of e_k yields

$$(f * g)[n] = \sum_k F[k]G[k]\exp(2\pi i n k/N)$$

The right hand side is the inverse DFT of the product FG. Thus:

$$(f * g) = \mathrm{IDFT}(FG)$$

Substituting $F = \mathrm{DFT}(f)$ and $G = \mathrm{DFT}(g)$:

$$(f * g) = \mathrm{IDFT}(\mathrm{DFT}(f)\mathrm{DFT}(g))$$

Finally, taking the DFT of both sides yields the Convolution Theorem:

$$\mathrm{DFT}(f * g) = \mathrm{DFT}(f)\mathrm{DFT}(g)$$

QED

10.6 Exercises

Solutions to these exercises are in `chap10soln.ipynb`.

Exercise 10.1 In Section 10.4 I describe convolution as the sum of shifted, scaled copies of a signal. Strictly speaking, this operation is *linear* convolution, which does not assume that the signal is periodic.

But in Section 10.3, when we multiply the DFT of the signal by the transfer function, that operation corresponds to *circular* convolution, which assumes that the signal is periodic. As a result, you might notice that the output contains an extra note at the beginning, which wraps around from the end.

Fortunately, there is a standard solution to this problem. If you add enough zeros to the end of the signal before computing the DFT, you can avoid wrap-around and compute a linear convolution.

Modify the example in `chap10.ipynb` and confirm that zero-padding eliminates the extra note at the beginning of the output.

Exercise 10.2 The Open AIR library provides a "centralized... on-line resource for anyone interested in auralization and acoustical impulse response data" (`http://www.openairlib.net`). Browse their collection of impulse response data and download one that sounds interesting. Find a short recording that has the same sample rate as the impulse response you downloaded.

Simulate the sound of your recording in the space where the impulse response was measured, computed two way: by convolving the recording with the impulse response and by computing the filter that corresponds to the impulse response and multiplying by the DFT of the recording.