

# Unit 6: Discrete Fourier Transforms

## Contents

- Understanding the Discrete Fourier Transform and the FFT
- Scope and Background Reading
- Contents
- References

As discussed in [Unit 4: Fourier Transform](#), for continuous-time (CT) signals, the Fourier transform is used to analyse the behaviour of signals and systems in the frequency domain  $j\omega$ .

For discrete-time (DT) signals and systems we derive and use a discrete Fourier transform (DTFT) to analysis the behaviour of sampled-data systems in frequency domain.

The discrete Fourier transform (DFT) converts a finite sequence of equally-spaced samples of a CT signal into a same-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency. The interval at which the DTFT is sampled is the reciprocal of the duration of the input sequence. An inverse DFT (IDFT) is a Fourier series, using the DTFT samples as coefficients of complex sinusoids at the corresponding DTFT frequencies. It has the same sample-values as the original input sequence. The DFT is therefore said to be a frequency domain representation of the original input sequence. If the original sequence spans all the non-zero values of a function, its DTFT is continuous (and periodic), and the DFT provides discrete samples of one cycle. If the original sequence is one cycle of a periodic function, the DFT provides all the non-zero values of one DTFT cycle.

The DFT is arguably one of the most important discrete transforms, and is used to perform Fourier analysis in many practical applications. In digital signal processing,

interval (often defined by a window function). In image processing, the samples can be the values of pixels along a row or column of a raster image. The DFT is also used to efficiently solve partial differential equations, and to perform other operations such as convolutions or multiplying large integers.

Since it deals with a finite amount of data, it can be implemented in computers by numerical algorithms or even dedicated hardware. These implementations usually employ efficient fast Fourier transform (FFT) algorithms; so much so that the terms "FFT" and "DFT" are often used interchangeably.

– Wikipedia [Discrete Fourier transform](#)

In this unit we will define the DFT and then show how the FFT is derived and implemented.

## Understanding the Discrete Fourier Transform and the FFT

This video, published on YouTube on 29th November 2023 by Brian Douglas ([[Douglas, n.d.](#)]) for the MathWorks as a MATLAB Tech Talk, is a nice gentle introduction to the DFT. It would be useful to watch this video in advance of the first class on this topic

### Understanding the Discrete Fourier Transform and the FFT



# Scope and Background Reading

The primary source for the content of this unit is Chapter 10 of [Karris, 2012] but there are many other places online where you can find out more about the DFT and FFT. A good starting point is the Wikipedia article [Discrete Fourier Transform](#) quoted above. We also recommend the video from Brian Douglas, **Understanding the Discrete Fourier Transform and the FFT** embedded above. The MathWorks article *Discrete Fourier Transform* ([MathWorks, n.d.]) is a useful introduction to the `fft` function used in MATLAB for FFT analysis.

## Contents

- unit6.1
- [Unit 6.2: The Fast Fourier Transform](#)

## References

See [Bibliography](#).

Previous  
< [Unit 5.4: Models of Discrete-Time Systems](#)

Next  
[Unit 6.1: The Discrete Fourier Transform](#) >