**What is Fourier Transform and its applications?**

Fourier transform convert complex curves into sum of a series of cosine waves and an additive term. Fourier analysis used as time series analysis proved its application in Quantum mechanics; Signal processing, Image Processing and filters, representation, Data Processing and Analysis and many more.

Fast Fourier transform

A **fast Fourier transform** (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.

**FFTs** are **used** to sharpen edges and create effects in static images and are widely **used** to turn a number series into sine waves and graphs. The **FFT** quickly performs a discrete Fourier transform (DFT), which is the practical application of Fourier transforms.

Fast Fourier transforms are widely used for applications in engineering, music, science, and mathematics. The basic ideas were popularized in 1965, but some algorithms had been derived as early as 1805. In 1994, Gilbert Strang described the FFT as "the most important numerical algorithm of our lifetime", and it was included in Top 10 Algorithms of 20th Century by the IEEE magazine Computing in Science & Engineering.

Applications

The FFT is used in digital recording, sampling, [additive synthesis](https://en.wikipedia.org/wiki/Additive_synthesis) and [pitch correction](https://en.wikipedia.org/wiki/Pitch_correction) software.

The FFT's importance derives from the fact that it has made working in the frequency domain equally computationally feasible as working in the temporal or spatial domain. Some of the important applications of the FFT include.

* Fast large-integer and polynomial multiplication
* Efficient matrix-vector multiplication for [Toeplitz](https://en.wikipedia.org/wiki/Toeplitz_matrix" \o "Toeplitz matrix), [circulant](https://en.wikipedia.org/wiki/Circulant_matrix" \o "Circulant matrix) and other structured matrices
* Filtering algorithms (see [overlap-add](https://en.wikipedia.org/wiki/Overlap-add_method) and [overlap-save](https://en.wikipedia.org/wiki/Overlap-save_method) methods)
* Fast algorithms for [discrete cosine](https://en.wikipedia.org/wiki/Discrete_cosine_transform) or [sine transforms](https://en.wikipedia.org/wiki/Discrete_sine_transform) (e.g. [fast DCT](https://en.wikipedia.org/wiki/Discrete_cosine_transform) used for [JPEG](https://en.wikipedia.org/wiki/JPEG) and [MPEG](https://en.wikipedia.org/wiki/MPEG)/[MP3](https://en.wikipedia.org/wiki/MP3) encoding and decoding)
* Fast [Chebyshev approximation](https://en.wikipedia.org/wiki/Chebyshev_approximation" \o "Chebyshev approximation)
* Solving [difference equations](https://en.wikipedia.org/wiki/Recurrence_relation)
* Computation of [isotopic distributions](https://en.wikipedia.org/wiki/Mass_spectrometry).[[47]](https://en.wikipedia.org/wiki/Fast_Fourier_transform#cite_note-Fernandez-de-Cossio_2012-47)

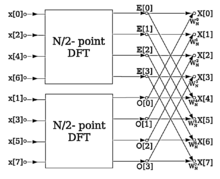


Fig-1: An example FFT algorithm structure, using a decomposition into half-size FFTs

**Why do we use Fourier transformation?**

First and foremost, a Fourier transform of a signal tells you what frequencies are present in your signal and in what proportions. ... For discrete signals, with the development of efficient FFT algorithms, almost always, it is faster to implement a convolution operation in the frequency domain than in the time domain.

**What is difference between DFT and FFT?**

Discrete Fourier Transform, or simply referred to as DFT, is the algorithm that transforms the time domain signals to the frequency domain components. ... Fast Fourier Transform, or FFT, is a computational algorithm that reduces the computing time and complexity of large transforms.

The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors.

As a result, it manages to reduce the complexity of computing the DFT. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In the presence of round-off error, many FFT algorithms are much more accurate than evaluating the DFT definition directly or indirectly. There are many different FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory.

**What makes FFT fast?**

FFT is based on divide and conquer algorithm where you divide the signal into two smaller signals, compute the DFT of the two smaller signals and join them to get the DFT of the larger signal. The order of complexity of DFT is O(n^2) while that of FFT is O(n. logn) hence, FFT is faster than DFT

**What does FFT do in Matlab?**

Y = fft( X ) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm. If X is a vector, then fft(X) returns the Fourier transform of the vector. If X is a matrix, then fft(X) treats the columns of X as vectors and returns the Fourier transform of each column.

**What is the main advantage of FFT?**

**FFT** helps in converting the time domain in frequency domain which makes the calculations easier as we always deal with various frequency bands in communication system another very big **advantage** is that it can convert the discrete data into a contionousdata type available at various frequencies.

**Why FFT is used in signal processing?**

Igor uses the Fast Fourier Transform (FFT) algorithm to compute a Discrete Fourier Transform (DFT). The FFT can be used to simply characterize the magnitude and phase of a signal, or it can be used in combination with other operations to perform more involved computations such as convolution or correlation.