Discrete Fourier transform
Fast Fourier transform
& their application in
Signal Processing

A presentation by
Sujoy ketan Saha

Discrete Fourier transform (DFT) :

- the **discrete Fourier transform (DFT)**, occasionally called the **finite Fourier transform**, is a transform for Fourier analysis of **finite-domain discrete-time signals**
- It is widely employed in **signal processing** and related fields to analyze the frequencies contained in a **sampled signal**, to solve **partial differential equations**, and to perform other operations such as **convolutions**.
- The DFT can be computed efficiently in practice using a **fast Fourier transform** (FFT) algorithm.

- **DIFFERENCE** :**between DFT & FFT :** though FFT algorithms are so commonly employed to compute the DFT, there is a difference:

"DFT" refers to a mathematical transformation, regardless of how it is computed, while "FFT" refers to any one of several efficient algorithms for the DFT.

DFT…

- **Definition: The sequence of *N* complex numbers $x_0, \ldots, x_{N-1}$ is transformed into the sequence of *N* complex numbers $X_0, \ldots, X_{N-1}$ by the DFT according to the formula**

[1]: $$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

- The **inverse discrete Fourier transform (IDFT)** is given by

[2]: $$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

NB: 1.the normalization factor multiplying the DFT and IDFT (here 1 and $1/N$) and the signs of the exponents are merely conventions.

2.A normalization of $1/\sqrt{N}$ for both the DFT and IDFT makes the transforms unitary, which has some theoretical advantages.

3.The convention of a negative sign in the exponent is often convenient because it means that X_k is the amplitude of a "positive frequency" $2\pi k / N$. Equivalently, the DFT is often thought of as a matched filter: when looking for a frequency of +1, one correlates the incoming signal with a frequency of −1.

DFT: **Properties:**

1. Completeness :

The discrete Fourier transform is an invertible, linear transformation

$$\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$$

With $\mathbb{C}$ denoting the set of complex numbers. In other words, for any *N* > 0, an *n*-dimensional complex vector has a DFT and an IDFT which are in turn *n*-dimensional complex vectors.

2. Orthogonality :

The vectors $e^{\frac{2\pi i}{N} kn}$ form an orthogonal basis over the set of *N*-dimensional complex vectors:

$$\sum_{n=0}^{N-1} \left(e^{\frac{2\pi i}{N} kn}\right) \left(e^{-\frac{2\pi i}{N} k' n}\right) = N\ \delta_{kk'}$$

where $\delta_{kk'}$ is the Kronecker delta. This orthogonality condition can be used to derive the formula for the IDFT from the definition of the DFT.

DFT: **Properties:**

3. Periodicity :

If the expression that defines the DFT is evaluated for all integers *k* instead of just for $k = 0, \dots, N-1$ then the resulting infinite sequence is a periodic extension of the DFT, periodic with period *N*. The periodicity can be shown directly from the definition:

$$X_{k+N} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}(k+N)n} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} e^{-2\pi i n} = X_k,$$

where we have used the fact that $e^{-2\pi i} = 1$. In the same way it can be shown that the IDFT formula leads to a periodic extension.

DFT: **Properties:**

4. The shift theorem :

Multiplying x_n by a *linear phase* $e^{\frac{2\pi i}{N}nm}$ **for some integer *m* corresponds to a *circular shift* of the output X_k: X_k is replaced by X_{k-m}, where the subscript is interpreted modulo *N* (i.e. periodically). Similarly, a circular shift of the input x_n corresponds to multiplying the output X_k by a linear phase. Mathematically, if $\{x_n\}$ represents the vector x then**

If $\mathcal{F}(\{x_n\})_k = X_k$

then $\mathcal{F}(\{x_n e^{\frac{2\pi i}{N}nm}\})_k = X_{k-m}$

And $\mathcal{F}(\{x_{n-m}\})_k = X_k e^{-\frac{2\pi i}{N}km}$

Where, the transform [DFT] is denoted by the symbol $\mathcal{F}$ **, as in** $\mathbf{X} = \mathcal{F}(\mathbf{x})$

DFT: **Properties:**

5. Circular convolution theorem and cross-correlation theorem : The cyclic or circular convolution x*y of the two vectors

x = x_k and y = y_n is the vector x*y with components

$$(\mathbf{x} * \mathbf{y})_n = \sum_{m=0}^{N-1} x_m y_{n-m} \qquad n = 0, \ldots, N-1$$

where we continue y cyclically so that

$$y_{-m} = y_{N-m} \qquad m = 0, \ldots, N-1$$

The DFT turns cyclic convolutions into component-wise multiplication.

That is, if $z_n = (\mathbf{x} * \mathbf{y})_n$, then $Z_k = X_k Y_k \qquad k = 0, \ldots, N-1$

where capital letters (*X*, *Y*, *Z*) represent the DFTs of sequences represented by small letters (*x*, *y*, *z*).

DFT: Properties:

Circular convolution theorem and cross-correlation theorem (contd.)... :

- **NB:** if a different normalization convention is adopted for the DFT (e.g., the unitary normalization), then there will in general be a constant factor multiplying the above relation
- The direct evaluation of the convolution summation, above, would require $O(N^2)$ operations, but the DFT (via an FFT) provides an $O(N\log N)$ method to compute the same thing.

- It can be shown that if z_n is the cross-correlation of x_n and y_n:

$$z_n = (\mathbf{x} * \mathbf{y})_n = \sum_{m=0}^{N-1} x_m^* \, y_{m+n}$$

where the sum is again cyclic in m, then the discrete Fourier transform of z_n is: $Z_k = X_k^* \, Y_k$

where capital letters are again used to signify the discrete Fourier transform.

DFT: **Properties:**

6.The unitary DFT :

Another way of looking at the DFT is to note that in the above discussion, the DFT can be expressed as a Vandermonde matrix:

$$\mathbf{F} = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \ldots & \omega_N^{0\cdot (N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \ldots & \omega_N^{1\cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \ldots & \omega_N^{(N-1)\cdot (N-1)} \end{bmatrix}$$

Where: $\omega_N = e^{-2\pi i/N}$ is a primitive Nth root of unity.

The inverse transform is then given by the inverse of the above matrix: $\mathbf{F}^{-1} = \frac{1}{N}\mathbf{F}^*$

With unitary normalization constants $1/\sqrt{N}$ the DFT becomes a unitary transformation, defined by a unitary matrix: $\mathbf{U} = \mathbf{F}/\sqrt{N}$;

$\mathbf{U}^{-1} = \mathbf{U}^*$; $|\det(\mathbf{U})| = 1$.

DFT: **Properties:**

7. Expressing the inverse DFT in terms of the DFT :

Can be easily done via several well-known "tricks”.

1st: we can compute the inverse DFT by reversing the inputs:

$$\mathcal{F}^{-1}(\{x_n\}) = \mathcal{F}(\{x_{N-n}\})/N$$

2nd: one can also conjugate the inputs and outputs: $\mathcal{F}^{-1}(\mathbf{x}) = \mathcal{F}(\mathbf{x}^*)^*/N$

3rd: **a variant of this conjugation trick, which is sometimes preferable because it requires no modification of the data values, involves swapping real and imaginary parts (which can be done on a computer simply by modifying pointers). Define swap(x_n) as x_n with its real and imaginary parts swapped—that is, if $x_n = a + bi$ then swap(x_n) is $b + ai$. Equivalently, swap(x_n) equals.**

then : $\mathcal{F}^{-1}(\mathbf{x}) = \mathrm{swap}(\mathcal{F}(\mathrm{swap}(\mathbf{x})))/N$

i.e. the inverse transform is the same as the forward transform with the real and imaginary parts swapped for both input and output, up to a normalization

DFT: **Properties:**

8. Eigenvalues and eigenvectors:

The eigenvalues of the DFT matrix are simple and well-known, whereas the eigenvectors are complicated, not unique, and are the subject of ongoing research.

Consider the unitary form $\mathbf{U}$ defined above for the DFT of length *N*, where $\mathbf{U}_{m,n} = \omega_N^{mn}/\sqrt{N} = \exp(-2\pi imn/N)/\sqrt{N}$

This matrix satisfies the equation: $\mathbf{U}^4 = \mathbf{I}$.

operating $\mathbf{U}$ twice gives the original data in reverse order, so operating $\mathbf{U}$ four times gives back the original data and is thus the identity matrix. This means that the eigenvalues λ satisfy a characteristic equation: $\lambda^4 = 1$.

SO,the eigenvalues of are the fourth roots of unity:

λ is +1, −1, +*i*, or −*i*.

DFT: **Properties:**
Eigenvalues and eigenvectors (Contd...):

- Since there are only four distinct eigenvalues for this $N \times N$ matrix, they have some multiplicity. The multiplicity gives the number of linearly independent eigenvectors corresponding to each eigenvalues.
- The multiplicity depends on the value of *N* modulo 4

DFT: **Properties:**

- **The real-input DFT:**

If $x_0, \ldots, x_{N-1}$ are real numbers, as they often are in practical applications, then the DFT obeys the symmetry: $X_k = X_{N-k}^*,$ where the star denotes complex conjugation and the subscripts are interpreted modulo *N*.

Therefore, the DFT output for real inputs is half redundant, and one obtains the complete information by only looking at roughly half of the outputs . In this case, the "DC" element X_0 is purely real, and for even *N* the "Nyquist" element $X_{N/2}$ is also real, so there are exactly *N* non-redundant real numbers in the first half + Nyquist element of the complex output *X*.

Using Euler's formula, the interpolating trigonometric polynomial can then be interpreted as a sum of sine and cosine functions.

Generalized/shifted DFT:

- It is possible to shift the transform sampling in time and/or frequency domain by some real shifts *a* and *b*, respectively.

This is known as **generalized DFT** (or **GDFT**), also called the **shifted DFT** or **offset DFT** , and has analogous properties to the ordinary DFT:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}(k+b)(n+a)} \qquad k = 0, \dots, N-1$$

Generalized/shifted DFT:

- Most often, shifts of 1 / 2 (half a sample) are used. While the ordinary DFT corresponds to a periodic signal in both time and frequency domains, $a = 1/2$ produces a signal that is anti-periodic in frequency domain ($X_{k+N} = -X_k$) and vice-versa for $b = 1/2$. Thus, the specific case of $a = b = 1/2$ is known as an *odd-time odd-frequency* discrete Fourier transform (or O^2 DFT).
- **Such shifted transforms are most often used for symmetric data, to represent different boundary symmetries, and for real-symmetric data they correspond to different forms of the discrete cosine and sine transforms.**
- Another interesting choice is $a = b = -(N - 1)/2$, which is called the **centered DFT** (or **CDFT**). The centered DFT has the useful property that, when N is a multiple of four, all four of its eigenvalues have equal multiplicities

Applications of DFT:

The DFT has seen wide usage across a large number of fields :

- **Spectral analysis,**
- **Data compression,**
- **Partial differential equations,**
- **Multiplication of large integers,**
- **Outline of DFT polynomial multiplication algorithm.**

FFT:

- A **Fast Fourier Transform** (**FFT**) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse.

- Let $x_0, \ldots., x_{N-1}$ be complex numbers. The DFT is defined by the formula :

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} \qquad k = 0, \ldots, N-1.$$

Evaluating these sums directly would take O(N^2) arithmetical operations.

An FFT is an algorithm to compute the same result in only O(N log N) operations. In general, such algorithms depend upon the factorization of N, but (contrary to popular misconception) there are O(N log N) FFTs for all N, even prime N.

FFT:

- Many FFT algorithms only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is a primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms.

- Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a $1/N$ factor, any FFT algorithm can easily be adapted for it as well.

The Cooley-Tukey algorithm:

- By far the most common FFT is the Cooley-Tukey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1N_2$ into many smaller DFTs of sizes N_1 and N_2, along with $O(N)$ multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966).
- This method (and the general idea of an FFT) was popularized by a publication of J. W. Cooley and J. W. Tukey in 1965, but it was later discovered that those two authors had independently re-invented an algorithm known to Carl Friedrich Gauss around 1805 (and subsequently rediscovered several times in limited forms).
- The most well-known use of the Cooley-Tukey algorithm is to divide the transform into two pieces of size $N / 2$ at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general (as was known to both Gauss and Cooley/Tukey). These are called the **radix-2** and **mixed-radix** cases, respectively (and other variants such as the split-radix FFT have their own names as well). Although the basic idea is recursive, most traditional implementations rearrange the algorithm to avoid explicit recursion.

Other FFT algorithms:

- *Prime-factor FFT algorithm* ,
- *Bruun's FFT algorithm* ,
- *Rader's FFT algorithm* ,
- *Bluestein's FFT algorithm*.

Digital signal processing:

- **Digital signal processing** (**DSP**) is the study of signals in a digital representation and the processing methods of these signals. DSP inculdes subfields like: audio signal processing, control engineering, digital image processing and speech processing. RADAR Signal processing and communications signal processing are two other important subfields of DSP.
- Since the goal of DSP is usually to measure or filter continuous real-world analog signals, the first step is usually to convert the signal from an analog to a digital form, by using an analog to digital converter. Often, the required output signal is another analog output signal, which requires a digital to analog converter.

Digital signal processing:

- The algorithms required for DSP are sometimes performed using specialized computers, which make use of specialized microprocessors called digital signal processors (also abbreviated *DSP*). These process signals in real time and are generally purpose-designed application-specific integrated circuits (ASICs). When flexibility and rapid development are more important than unit costs at high volume, DSP algorithms may also be implemented using field-programmable gate arrays (FPGAs).

DSP domains

In DSP, engineers usually study digital signals in one of the following domains:

- time domain (one-dimensional signals),
- spatial domain (multidimensional signals),
- frequency domain,
- autocorrelation domain, and
- wavelet domains.

DSP domains

- A sequence of samples from a measuring device produces a time or spatial domain representation,
- whereas a discrete Fourier transform produces the frequency domain information, that is the frequency spectrum.
- Autocorrelation is defined as the cross-correlation of the signal with itself over varying intervals of time or space.