WIKIPEDIA

Discrete Fourier transform

In mathematics, the **discrete Fourier transform** (**DFT**) converts a finite sequence of equally-spaced samples of a function 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 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 the most important discrete transform, used to perform Fourier analysis in many practical applications.[1] In digital signal processing, the function is any quantity or signal that varies over time, such as the pressure of a sound wave, a radio signal, or daily temperature readings, sampled over a finite time interval (often defined by a window function[2]). 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;[3] so much so that the terms "FFT" and "DFT" are often used interchangeably. Prior to its current usage, the "FFT" initialism may have also been used for the ambiguous term "finite Fourier transform".

Relationship between the (continuous) Fourier transform and the discrete Fourier transform. Left column: A continuous function (top) and its Fourier transform (bottom). Center-left column: Periodic summation of the original function (top). Fourier transform (bottom) is zero except at discrete points. The inverse transform is a sum of sinusoids called Fourier series. Center-right column: Original function is discretized (multiplied by a Dirac comb) (top). Its Fourier transform (bottom) is a periodic summation (DTFT) of the original transform. Right column: The DFT (bottom) computes discrete samples of the continuous DTFT. The inverse DFT (top) is a periodic summation of the original samples. The FFT algorithm computes one cycle of the DFT and its inverse is one cycle of the DFT inverse.
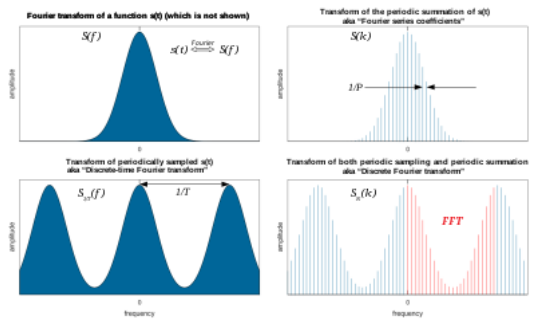
Depiction of a Fourier transform (upper left) and its periodic summation (DTFT) in the lower left corner. The spectral sequences at (a) upper right and (b) lower right are respectively computed from (a) one cycle of the periodic summation of s(t) and (b) one cycle of the periodic summation of the s(nT) sequence. The respective formulas are (a) the Fourier series integral and (b) the **DFT summation**. Its similarities to the original transform, S(f), and its relative computational ease are often the motivation for computing a DFT sequence.

Contents

Definition

The *discrete Fourier transform* transforms a <u>sequence</u> of N <u>complex numbers</u> $\{\mathbf{x}_n\} := x_0, x_1, \ldots, x_{N-1}$ into another sequence of complex numbers, $\{\mathbf{X}_k\} := X_0, X_1, \ldots, X_{N-1}$, which is defined by

$$
\begin{aligned}
X_k &= \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} \\
&= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right) \right],
\end{aligned}
$$

(Eq.1)

where the last expression follows from the first one by <u>Euler's formula</u>.

The transform is sometimes denoted by the symbol $\mathcal{F}$, as in $\mathbf{X} = \mathcal{F}\{\mathbf{x}\}$ or $\mathcal{F}(\mathbf{x})$ or $\mathcal{F}\mathbf{x}$.[A]

Motivation

Eq.1 can also be evaluated outside the domain $k \in [0, N-1]$, and that extended sequence is N-<u>periodic</u>. Accordingly, other sequences of N indices are sometimes used, such as $\left[-\frac{N}{2}, \frac{N}{2} - 1\right]$ (if N is even) and $\left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$ (if N is odd), which amounts to swapping the left and right halves of the result of the transform.[4]

Eq.1 can be interpreted or derived in various ways, for example:

- It completely describes the <u>discrete-time Fourier transform</u> (DTFT) of an N-periodic sequence, which comprises only discrete frequency components. (<u>Using the DTFT with periodic data</u>)
- It can also provide uniformly spaced samples of the continuous DTFT of a finite length sequence. (<u>§ Sampling the DTFT</u>)
- It is the <u>cross correlation</u> of the *input* sequence, x_n, and a complex sinusoid at frequency $\frac{k}{N}$. Thus it acts like a <u>matched filter</u> for that frequency.
- It is the discrete analog of the formula for the coefficients of a <u>Fourier series</u>:

$$
x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}, \quad n \in \mathbb{Z},
$$

(Eq.2)

which is also N-periodic. In the domain $n \in [0, N-1]$, this is the **inverse transform** of **Eq.1**. In this interpretation, each X_k is a complex number that encodes both amplitude and phase of a complex sinusoidal component $\left(e^{i2\pi kn/N}\right)$ of function x_n. (see <u>Discrete Fourier series</u>) The sinusoid's <u>frequency</u> is k cycles per N samples. Its amplitude and phase are:

$$
\begin{aligned}
\frac{1}{N}|X_k| &= \frac{1}{N}\sqrt{\operatorname{Re}(X_k)^2 + \operatorname{Im}(X_k)^2} \\
\arg(X_k) &= \operatorname{atan2}\left(\operatorname{Im}(X_k), \operatorname{Re}(X_k)\right) = -i \cdot \ln\left(\frac{X_k}{|X_k|}\right),
\end{aligned}
$$

where <u>atan2</u> is the two-argument form of the <u>arctan</u> function. In polar form $X_k = |X_k|e^{i\arg(X_k)} = |X_k|\operatorname{cis}\arg(X_k)$ where <u>cis</u> is the mnemonic for cos + *i* sin.

The normalization factor multiplying the DFT and IDFT (here 1 and $\frac{1}{N}$) and the signs of the exponents are merely <u>conventions</u>, and differ in some treatments. The only requirements of these conventions are that the DFT and IDFT have opposite-sign exponents and that the product of their normalization factors be $\frac{1}{N}$. A normalization of $\sqrt{\frac{1}{N}}$ for both the DFT and IDFT, for instance, makes the transforms unitary. A discrete impulse, $x_n = 1$ at $n = 0$ and 0 otherwise; might transform to $X_k = 1$ for all k (use normalization factors 1 for DFT and $\frac{1}{N}$ for IDFT). A DC signal, $X_k = 1$ at $k = 0$ and 0 otherwise; might inversely transform to $x_n = 1$ for all n (use $\frac{1}{N}$ for DFT and 1 for IDFT) which is consistent with viewing <u>DC</u> as the <u>mean average</u> of the signal.

Example

Let $N = 4$ and

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 - i \\ -i \\ -1 + 2i \end{pmatrix}$$

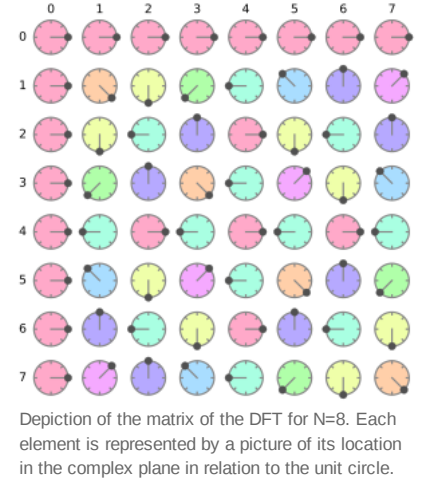Here we demonstrate how to calculate the DFT of $\mathbf{x}$ using Eq.1:

$$X_0 = e^{-i2\pi 0 \cdot 0/4} \cdot 1 + e^{-i2\pi 0 \cdot 1/4} \cdot (2 - i) + e^{-i2\pi 0 \cdot 2/4} \cdot (-i) + e^{-i2\pi 0 \cdot 3/4} \cdot (-1 + 2i) = 2$$

$$X_1 = e^{-i2\pi 1 \cdot 0/4} \cdot 1 + e^{-i2\pi 1 \cdot 1/4} \cdot (2 - i) + e^{-i2\pi 1 \cdot 2/4} \cdot (-i) + e^{-i2\pi 1 \cdot 3/4} \cdot (-1 + 2i) = -2 - 2i$$

$$X_2 = e^{-i2\pi 2 \cdot 0/4} \cdot 1 + e^{-i2\pi 2 \cdot 1/4} \cdot (2 - i) + e^{-i2\pi 2 \cdot 2/4} \cdot (-i) + e^{-i2\pi 2 \cdot 3/4} \cdot (-1 + 2i) = -2i$$

$$X_3 = e^{-i2\pi 3 \cdot 0/4} \cdot 1 + e^{-i2\pi 3 \cdot 1/4} \cdot (2 - i) + e^{-i2\pi 3 \cdot 2/4} \cdot (-i) + e^{-i2\pi 3 \cdot 3/4} \cdot (-1 + 2i) = 4 + 4i$$

$$\mathbf{X} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 - 2i \\ -2i \\ 4 + 4i \end{pmatrix}$$

Depiction of the matrix of the DFT for N=8. Each element is represented by a picture of its location in the complex plane in relation to the unit circle.

Inverse transform

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. Its inverse is known as Inverse Discrete Fourier Transform (IDFT). 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.

The inverse transform is given by:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i\frac{2\pi}{N}kn} \qquad \text{(Eq.3)}$$

Properties

Linearity

The DFT is a linear transform, i.e. if $\mathcal{F}(\{x_n\})_k = X_k$ and $\mathcal{F}(\{y_n\})_k = Y_k$, then for any complex numbers a, b:

$$\mathcal{F}(\{ax_n + by_n\})_k = aX_k + bY_k$$

Time and frequency reversal

Reversing the time (i.e. replacing n by $N - n$)[B] in x_n corresponds to reversing the frequency (i.e. k by $N - k$).[5]:p.421 Mathematically, if $\{x_n\}$ represents the vector $\mathbf{x}$ then

if $\mathcal{F}(\{x_n\})_k = X_k$
then $\mathcal{F}(\{x_{N-n}\})_k = X_{N-k}$

Conjugation in time

If $\mathcal{F}(\{x_n\})_k = X_k$ then $\mathcal{F}(\{x_n^*\})_k = X_{N-k}^*$.[5]:p.423

Real and imaginary part

This table shows some mathematical operations on x_n in the time domain and the corresponding effects on its DFT X_k in the frequency domain.

Property	Time domain x_n	Frequency domain X_k
Real part in time	$\Re(x_n)$	$\frac{1}{2}\left(X_k + X_{N-k}^*\right)$
Imaginary part in time	$\Im(x_n)$	$\frac{1}{2i}\left(X_k - X_{N-k}^*\right)$
Real part in frequency	$\frac{1}{2}\left(x_n + x_{N-n}^*\right)$	$\Re(X_k)$
Imaginary part in frequency	$\frac{1}{2i}\left(x_n - x_{N-n}^*\right)$	$\Im(X_k)$

Orthogonality

The vectors $u_k = \left[e^{\frac{i2\pi}{N}kn} \,\middle|\, n = 0, 1, \dots, N-1 \right]^{\mathsf{T}}$ form an <u>orthogonal basis</u> over the set of N-dimensional complex vectors:

$$u_k^{\mathsf{T}} u_{k'}^* = \sum_{n=0}^{N-1}\left(e^{\frac{i2\pi}{N}kn}\right)\left(e^{\frac{i2\pi}{N}(-k')n}\right) = \sum_{n=0}^{N-1}e^{\frac{i2\pi}{N}(k-k')n} = N\,\delta_{kk'}$$

where $\delta_{kk'}$ is the <u>Kronecker delta</u>. (In the last step, the summation is trivial if $k = k'$, where it is $1 + 1 + \cdots = N$, and otherwise is a <u>geometric series</u> that can be explicitly summed to obtain zero.) This orthogonality condition can be used to derive the formula for the IDFT from the definition of the DFT, and is equivalent to the unitarity property below.

The Plancherel theorem and Parseval's theorem

If X_k and Y_k are the DFTs of x_n and y_n respectively then the <u>Parseval's theorem</u> states:

$$\sum_{n=0}^{N-1}x_n y_n^* = \frac{1}{N}\sum_{k=0}^{N-1}X_k Y_k^*$$

where the star denotes <u>complex conjugation</u>. <u>Plancherel theorem</u> is a special case of the Parseval's theorem and states:

$$\sum_{n=0}^{N-1}|x_n|^2 = \frac{1}{N}\sum_{k=0}^{N-1}|X_k|^2.$$

These theorems are also equivalent to the unitary condition below.

Periodicity

The periodicity can be shown directly from the definition:

$$X_{k+N} \triangleq \sum_{n=0}^{N-1}x_n e^{-\frac{i2\pi}{N}(k+N)n} = \sum_{n=0}^{N-1}x_n e^{-\frac{i2\pi}{N}kn}\underbrace{e^{-i2\pi n}}_{1} = \sum_{n=0}^{N-1}x_n e^{-\frac{i2\pi}{N}kn} = X_k.$$

Similarly, it can be shown that the IDFT formula leads to a periodic extension.

Shift theorem

Multiplying x_n by a *linear phase* $e^{\frac{i2\pi(n-1)}{N}m}$ 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 <u>modulo</u> 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

$$\begin{aligned} &\text{if } \mathcal{F}(\{x_n\})_k = X_k \\ &\text{then } \mathcal{F}\left(\left\{x_n \cdot e^{\frac{i2\pi}{N}nm}\right\}\right)_k = X_{k-m} \\ &\text{and } \mathcal{F}(\{x_{n-m}\})_k = X_k \cdot e^{-\frac{i2\pi}{N}km} \end{aligned}$$

Circular convolution theorem and cross-correlation theorem

The <u>convolution theorem</u> for the <u>discrete-time Fourier transform</u> (DTFT) indicates that a convolution of two sequences can be obtained as the inverse transform of the product of the individual transforms. An important simplification occurs when one of sequences is N-periodic, denoted here by y_N, because $\mathbf{DTFT}\{y_N\}$ is non-zero at only discrete frequencies (see <u>DTFT § Periodic data</u>), and therefore so is its product with the continuous function $\mathbf{DTFT}\{x\}$. That leads to a considerable simplification of the inverse transform.

$$x * y_N = \mathbf{DTFT}^{-1}\left[\mathbf{DTFT}\{x\} \cdot \mathbf{DTFT}\{y_N\}\right] = \mathbf{DFT}^{-1}\left[\mathbf{DFT}\{x_N\} \cdot \mathbf{DFT}\{y_N\}\right],$$

where x_N is a <u>periodic summation</u> of the x sequence: $(x_N)_n \triangleq \sum_{m=-\infty}^{\infty} x_{(n-mN)}.$

Customarily, the DFT and inverse DFT summations are taken over the domain $[0, N-1]$. Defining those DFTs as X and Y, the result is:

$$(x * y_N)_n \triangleq \sum_{\ell=-\infty}^{\infty} x_\ell \cdot (y_N)_{n-\ell} = \underbrace{\mathcal{F}^{-1}}_{\text{DFT}^{-1}} \{X \cdot Y\}_n.$$

In practice, the x sequence is usually length N or less, and y_N is a periodic extension of an N-length y-sequence, which can also be expressed as a *circular function*:

$$(y_N)_n = \sum_{p=-\infty}^{\infty} y_{(n-pN)} = y_{(n \bmod N)}, \quad n \in \mathbb{Z}.$$

Then the convolution can be written as:

$$\mathcal{F}^{-1}\{X \cdot Y\}_n = \sum_{\ell=0}^{N-1} x_\ell \cdot y_{(n-\ell) \bmod N}$$

which gives rise to the interpretation as a *circular* convolution of x and y.[6][7] It is often used to efficiently compute their linear convolution. (see <u>Circular convolution</u>, <u>Fast convolution algorithms</u>, and <u>Overlap-save</u>)

Similarly, the <u>cross-correlation</u> of x and y_N is given by:

$$(x \star y_N)_n \triangleq \sum_{\ell=-\infty}^{\infty} x_\ell^* \cdot (y_N)_{n+\ell} = \mathcal{F}^{-1}\{X^* \cdot Y\}_n.$$

Convolution theorem duality

It can also be shown that:

$$\mathcal{F}\{\mathbf{x} \cdot \mathbf{y}\}_k \triangleq \sum_{n=0}^{N-1} x_n \cdot y_n \cdot e^{-i\frac{2\pi}{N}kn}$$

$$= \frac{1}{N}(\mathbf{X} * \mathbf{Y_N})_k,$$ which is the circular convolution of $\mathbf{X}$ and $\mathbf{Y}$.

Trigonometric interpolation polynomial

The <u>trigonometric interpolation polynomial</u>

$$p(t) = \begin{cases} \frac{1}{N}\left[X_0 + X_1 e^{i2\pi t} + \cdots + X_{N/2-1}e^{i2\pi(N/2-1)t} + X_{N/2}\cos(N\pi t) + X_{N/2+1}e^{-i2\pi(N/2-1)t} + \cdots + X_{N-1}e^{-i2\pi t}\right] & N \text{ even} \\ \frac{1}{N}\left[X_0 + X_1 e^{i2\pi t} + \cdots + X_{(N-1)/2}e^{i\pi(N-1)t} + X_{(N+1)/2}e^{-i\pi(N-1)t} + \cdots + X_{N-1}e^{-i2\pi t}\right] & N \text{ odd} \end{cases}$$

where the coefficients X_k are given by the DFT of x_n above, satisfies the interpolation property $p(n/N) = x_n$ for $n = 0, \ldots, N-1$.

For even N, notice that the <u>Nyquist component</u> $\frac{X_{N/2}}{N}\cos(N\pi t)$ is handled specially.

This interpolation is *not unique*: aliasing implies that one could add N to any of the complex-sinusoid frequencies (e.g. changing e^{-it} to $e^{i(N-1)t}$) without changing the interpolation property, but giving *different* values in between the x_n points. The choice above, however, is typical because it has two useful properties. First, it consists of sinusoids whose frequencies have the smallest possible magnitudes: the interpolation is <u>bandlimited</u>. Second, if the x_n are real numbers, then $p(t)$ is real as well.

In contrast, the most obvious trigonometric interpolation polynomial is the one in which the frequencies range from 0 to $N-1$ (instead of roughly $-N/2$ to $+N/2$ as above), similar to the inverse DFT formula. This interpolation does *not* minimize the slope, and is *not* generally real-valued for real x_n; its use is a common mistake.

The unitary DFT

Another way of looking at the DFT is to note that in the above discussion, the DFT can be expressed as the <u>DFT matrix</u>, a <u>Vandermonde matrix</u>, <u>introduced by Sylvester</u> in 1867,

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

where $\omega_N = e^{-i2\pi/N}$ 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:

$$\mathbf{U} = \frac{1}{\sqrt{N}}\mathbf{F}$$
$$\mathbf{U}^{-1} = \mathbf{U}^*$$
$$|\det(\mathbf{U})| = 1$$

where $\det()$ is the determinant function. The determinant is the product of the eigenvalues, which are always ± 1 or $\pm i$ as described below. In a real vector space, a unitary transformation can be thought of as simply a rigid rotation of the coordinate system, and all of the properties of a rigid rotation can be found in the unitary DFT.

The orthogonality of the DFT is now expressed as an orthonormality condition (which arises in many areas of mathematics as described in root of unity):

$$\sum_{m=0}^{N-1} U_{km} U_{mn}^* = \delta_{kn}$$

If $\mathbf{X}$ is defined as the unitary DFT of the vector $\mathbf{x}$, then

$$X_k = \sum_{n=0}^{N-1} U_{kn} x_n$$

and the Parseval's theorem is expressed as

$$\sum_{n=0}^{N-1} x_n y_n^* = \sum_{k=0}^{N-1} X_k Y_k^*$$

If we view the DFT as just a coordinate transformation which simply specifies the components of a vector in a new coordinate system, then the above is just the statement that the dot product of two vectors is preserved under a unitary DFT transformation. For the special case $\mathbf{x} = \mathbf{y}$, this implies that the length of a vector is preserved as well — this is just Plancherel theorem,

$$\sum_{n=0}^{N-1} |x_n|^2 = \sum_{k=0}^{N-1} |X_k|^2$$

A consequence of the circular convolution theorem is that the DFT matrix F diagonalizes any circulant matrix.

Expressing the inverse DFT in terms of the DFT

A useful property of the DFT is that the inverse DFT can be easily expressed in terms of the (forward) DFT, via several well-known "tricks". (For example, in computations, it is often convenient to only implement a fast Fourier transform corresponding to one transform direction and then to get the other transform direction from the first.)

First, we can compute the inverse DFT by reversing all but one of the inputs (Duhamel *et al.*, 1988):

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

(As usual, the subscripts are interpreted modulo N; thus, for $n = 0$, we have $x_{N-0} = x_0$.)

Second, one can also conjugate the inputs and outputs:

$$\mathcal{F}^{-1}(\mathbf{x}) = \frac{1}{N}\mathcal{F}(\mathbf{x}^*)^*$$

Third, 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 $\mathbf{swap}(x_n)$ as x_n with its real and imaginary parts swapped—that is, if $x_n = a + bi$ then $\mathbf{swap}(x_n)$ is $b + ai$. Equivalently, $\mathbf{swap}(x_n)$ equals ix_n^*. Then

$$\mathcal{F}^{-1}(\mathbf{x}) = \frac{1}{N}\mathbf{swap}(\mathcal{F}(\mathbf{swap}(\mathbf{x})))$$

That is, 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 (Duhamel *et al.*, 1988).

The conjugation trick can also be used to define a new transform, closely related to the DFT, that is involutory—that is, which is its own inverse. In particular, $T(\mathbf{x}) = \mathcal{F}(\mathbf{x}^*)/\sqrt{N}$ is clearly its own inverse: $T(T(\mathbf{x})) = \mathbf{x}$. A closely related involutory transformation (by a factor of $\frac{1+i}{\sqrt{2}}$) is $H(\mathbf{x}) = \mathcal{F}((1+i)\mathbf{x}^*)/\sqrt{2N}$, since the $(1+i)$ factors in $H(H(\mathbf{x}))$ cancel the 2. For real inputs $\mathbf{x}$, the real part of $H(\mathbf{x})$ is none other than the discrete Hartley transform, which is also involutory.

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

$$\mathbf{U}_{m,n} = \frac{1}{\sqrt{N}} \omega_N^{(m-1)(n-1)} = \frac{1}{\sqrt{N}} e^{-\frac{i2\pi}{N}(m-1)(n-1)}.$$

This matrix satisfies the matrix polynomial equation:

$$\mathbf{U}^4 = \mathbf{I}.$$

This can be seen from the inverse properties above: 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 the equation:

$$\lambda^4 = 1.$$

Therefore, the eigenvalues of $\mathbf{U}$ are the fourth roots of unity: λ is +1, −1, +i, or −i.

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 eigenvalue. (There are N independent eigenvectors; a unitary matrix is never defective.)

The problem of their multiplicity was solved by McClellan and Parks (1972), although it was later shown to have been equivalent to a problem solved by Gauss (Dickinson and Steiglitz, 1982). The multiplicity depends on the value of N modulo 4, and is given by the following table:

size N	$\lambda = +1$	$\lambda = -1$	$\lambda = -i$	$\lambda = +i$
$4m$	$m + 1$	m	m	$m - 1$
$4m + 1$	$m + 1$	m	m	m
$4m + 2$	$m + 1$	$m + 1$	m	m
$4m + 3$	$m + 1$	$m + 1$	$m + 1$	m

Multiplicities of the eigenvalues λ of the unitary DFT matrix $\mathbf{U}$ as a function of the transform size N (in terms of an integer m).

Otherwise stated, the characteristic polynomial of $\mathbf{U}$ is:

$$\det(\lambda I - \mathbf{U}) = (\lambda - 1)^{\left\lfloor \frac{N+4}{4} \right\rfloor} (\lambda + 1)^{\left\lfloor \frac{N+2}{4} \right\rfloor} (\lambda + i)^{\left\lfloor \frac{N+1}{4} \right\rfloor} (\lambda - i)^{\left\lfloor \frac{N-1}{4} \right\rfloor}.$$

No simple analytical formula for general eigenvectors is known. Moreover, the eigenvectors are not unique because any linear combination of eigenvectors for the same eigenvalue is also an eigenvector for that eigenvalue. Various researchers have proposed different choices of eigenvectors, selected to satisfy useful properties like orthogonality and to have "simple" forms (e.g., McClellan and Parks, 1972; Dickinson and Steiglitz, 1982; Grünbaum, 1982; Atakishiyev and Wolf, 1997; Candan *et al.*, 2000; Hanna *et al.*, 2004; Gurevich and Hadani, 2008).

A straightforward approach is to discretize an eigenfunction of the continuous Fourier transform, of which the most famous is the Gaussian function. Since periodic summation of the function means discretizing its frequency spectrum and discretization means periodic summation of the spectrum, the discretized and periodically summed Gaussian function yields an eigenvector of the discrete transform:

- $$F(m) = \sum_{k \in \mathbb{Z}} \exp\left(-\frac{\pi \cdot (m + N \cdot k)^2}{N}\right).$$

The closed form expression for the series can be expressed by Jacobi theta functions as

- $$F(m) = \frac{1}{\sqrt{N}} \vartheta_3\left(\frac{\pi m}{N}, \exp\left(-\frac{\pi}{N}\right)\right).$$

Two other simple closed-form analytical eigenvectors for special DFT period N were found (Kong, 2008):

For DFT period $N = 2L + 1 = 4K + 1$, where K is an integer, the following is an eigenvector of DFT:

- $$F(m) = \prod_{s=K+1}^{L} \left[\cos\left(\frac{2\pi}{N}m\right) - \cos\left(\frac{2\pi}{N}s\right) \right]$$

For DFT period $N = 2L = 4K$, where K is an integer, the following is an eigenvector of DFT:

- $$F(m) = \sin\left(\frac{2\pi}{N}m\right) \prod_{s=K+1}^{L-1} \left[\cos\left(\frac{2\pi}{N}m\right) - \cos\left(\frac{2\pi}{N}s\right) \right]$$

The choice of eigenvectors of the DFT matrix has become important in recent years in order to define a discrete analogue of the fractional Fourier transform—the DFT matrix can be taken to fractional powers by exponentiating the eigenvalues (e.g., Rubio and Santhanam, 2005). For the continuous Fourier transform, the natural orthogonal eigenfunctions are the Hermite functions, so various discrete analogues of these have been employed as the eigenvectors of the DFT, such as the Kravchuk polynomials (Atakishiyev and Wolf, 1997). The "best" choice of eigenvectors to define a fractional discrete Fourier transform remains an open question, however.

Uncertainty principles

Probabilistic uncertainty principle

If the random variable X_k is constrained by

$$\sum_{n=0}^{N-1} |X_n|^2 = 1,$$

then

$$P_n = |X_n|^2$$

may be considered to represent a discrete probability mass function of n, with an associated probability mass function constructed from the transformed variable,

$$Q_m = N|x_m|^2.$$

For the case of continuous functions $P(x)$ and $Q(k)$, the Heisenberg uncertainty principle states that

$$D_0(X)D_0(x) \geq \frac{1}{16\pi^2}$$

where $D_0(X)$ and $D_0(x)$ are the variances of $|X|^2$ and $|x|^2$ respectively, with the equality attained in the case of a suitably normalized Gaussian distribution. Although the variances may be analogously defined for the DFT, an analogous uncertainty principle is not useful, because the uncertainty will not be shift-invariant. Still, a meaningful uncertainty principle has been introduced by Massar and Spindel.[8]

However, the Hirschman entropic uncertainty will have a useful analog for the case of the DFT.[9] The Hirschman uncertainty principle is expressed in terms of the Shannon entropy of the two probability functions.

In the discrete case, the Shannon entropies are defined as

$$H(X) = -\sum_{n=0}^{N-1} P_n \ln P_n$$

and

$$H(x) = -\sum_{m=0}^{N-1} Q_m \ln Q_m,$$

and the entropic uncertainty principle becomes[9]

$$H(X) + H(x) \geq \ln(N).$$

The equality is obtained for P_n equal to translations and modulations of a suitably normalized Kronecker comb of period A where A is any exact integer divisor of N. The probability mass function Q_m will then be proportional to a suitably translated Kronecker comb of period $B = N/A$.[9]

Deterministic uncertainty principle

There is also a well-known deterministic uncertainty principle that uses signal sparsity (or the number of non-zero coefficients).[10] Let $\|x\|_0$ and $\|X\|_0$ be the number of non-zero elements of the time and frequency sequences $x_0, x_1, \ldots, x_{N-1}$ and $X_0, X_1, \ldots, X_{N-1}$, respectively. Then,

$$N \leq \|x\|_0 \cdot \|X\|_0.$$

As an immediate consequence of the inequality of arithmetic and geometric means, one also has $2\sqrt{N} \leq \|x\|_0 + \|X\|_0$. Both uncertainty principles were shown to be tight for specifically-chosen "picket-fence" sequences (discrete impulse trains), and find practical use for signal recovery applications.[10]

DFT of real and purely imaginary signals

- If $x_0, \ldots, x_{N-1}$ are real numbers, as they often are in practical applications, then the DFT $X_0, \ldots, X_{N-1}$ is even symmetric:

  $x_n \in \mathbb{R} \quad \forall n \in \{0, \ldots, N-1\} \implies X_k = X_{-k \mod N}^* \quad \forall k \in \{0, \ldots, N-1\}$, where X^* denotes complex conjugation.

It follows that for even N X_0 and $X_{N/2}$ are real-valued, and the remainder of the DFT is completely specified by just $N/2 - 1$ complex numbers.

- If $x_0, \ldots, x_{N-1}$ are purely imaginary numbers, then the DFT $X_0, \ldots, X_{N-1}$ is odd symmetric:

  $x_n \in i\mathbb{R} \quad \forall n \in \{0, \ldots, N-1\} \implies X_k = -X_{-k \mod N}^* \quad \forall k \in \{0, \ldots, N-1\}$, where X^* denotes complex conjugation.

Generalized DFT (shifted and non-linear phase)

It is possible to shift the transform sampling in time and/or frequency domain by some real shifts a and b, respectively. This is sometimes known as a **generalized DFT** (or **GDFT**), also called the **shifted DFT** or **offset DFT**, and has analogous properties to the ordinary DFT:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N}(k+b)(n+a)} \qquad k = 0, \ldots, N-1.$$

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 O² 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 (see above) have equal multiplicities (Rubio and Santhanam, 2005)[11]

The term GDFT is also used for the non-linear phase extensions of DFT. Hence, GDFT method provides a generalization for constant amplitude orthogonal block transforms including linear and non-linear phase types. GDFT is a framework to improve time and frequency domain properties of the traditional DFT, e.g. auto/cross-correlations, by the addition of the properly designed phase shaping function (non-linear, in general) to the original linear phase functions (Akansu and Agirman-Tosun, 2010).[12]

The discrete Fourier transform can be viewed as a special case of the z-transform, evaluated on the unit circle in the complex plane; more general z-transforms correspond to *complex* shifts *a* and *b* above.

Multidimensional DFT

The ordinary DFT transforms a one-dimensional sequence or array x_n that is a function of exactly one discrete variable *n*. The multidimensional DFT of a multidimensional array $x_{n_1, n_2, \ldots, n_d}$ that is a function of *d* discrete variables $n_\ell = 0, 1, \ldots, N_\ell - 1$ for ℓ in $1, 2, \ldots, d$ is defined by:

$$X_{k_1, k_2, \ldots, k_d} = \sum_{n_1=0}^{N_1-1} \left(\omega_{N_1}^{k_1 n_1} \sum_{n_2=0}^{N_2-1} \left(\omega_{N_2}^{k_2 n_2} \cdots \sum_{n_d=0}^{N_d-1} \omega_{N_d}^{k_d n_d} \cdot x_{n_1, n_2, \ldots, n_d} \right) \right),$$

where $\omega_{N_\ell} = \exp(-i2\pi/N_\ell)$ as above and the *d* output indices run from $k_\ell = 0, 1, \ldots, N_\ell - 1$. This is more compactly expressed in vector notation, where we define $\mathbf{n} = (n_1, n_2, \ldots, n_d)$ and $\mathbf{k} = (k_1, k_2, \ldots, k_d)$ as *d*-dimensional vectors of indices from 0 to $\mathbf{N} - 1$, which we define as $\mathbf{N} - 1 = (N_1 - 1, N_2 - 1, \ldots, N_d - 1)$:

$$X_{\mathbf{k}} = \sum_{\mathbf{n}=0}^{\mathbf{N}-1} e^{-i2\pi \mathbf{k} \cdot (\mathbf{n}/\mathbf{N})} x_{\mathbf{n}},$$

where the division $\mathbf{n}/\mathbf{N}$ is defined as $\mathbf{n}/\mathbf{N} = (n_1/N_1, \ldots, n_d/N_d)$ to be performed element-wise, and the sum denotes the set of nested summations above.

The inverse of the multi-dimensional DFT is, analogous to the one-dimensional case, given by:

$$x_{\mathbf{n}} = \frac{1}{\prod_{\ell=1}^{d} N_\ell} \sum_{\mathbf{k}=0}^{\mathbf{N}-1} e^{i2\pi \mathbf{n} \cdot (\mathbf{k}/\mathbf{N})} X_{\mathbf{k}}.$$

As the one-dimensional DFT expresses the input x_n as a superposition of sinusoids, the multidimensional DFT expresses the input as a superposition of plane waves, or multidimensional sinusoids. The direction of oscillation in space is $\mathbf{k}/\mathbf{N}$. The amplitudes are $X_{\mathbf{k}}$. This decomposition is of great importance for everything from digital image processing (two-dimensional) to solving partial differential equations. The solution is broken up into plane waves.

The multidimensional DFT can be computed by the composition of a sequence of one-dimensional DFTs along each dimension. In the two-dimensional case x_{n_1, n_2} the N_1 independent DFTs of the rows (i.e., along n_2) are computed first to form a new array y_{n_1, k_2}. Then the N_2 independent DFTs of *y* along the columns (along n_1) are computed to form the final result X_{k_1, k_2}. Alternatively the columns can be computed first and then the rows. The order is immaterial because the nested summations above commute.

An algorithm to compute a one-dimensional DFT is thus sufficient to efficiently compute a multidimensional DFT. This approach is known as the *row-column* algorithm. There are also intrinsically multidimensional FFT algorithms.

The real-input multidimensional DFT

For input data $x_{n_1, n_2, \ldots, n_d}$ consisting of real numbers, the DFT outputs have a conjugate symmetry similar to the one-dimensional case above:

$$X_{k_1, k_2, \ldots, k_d} = X^*_{N_1-k_1, N_2-k_2, \ldots, N_d-k_d},$$

where the star again denotes complex conjugation and the ℓ-th subscript is again interpreted modulo N_ℓ (for $\ell = 1, 2, \ldots, d$).

Applications

The DFT has seen wide usage across a large number of fields; we only sketch a few examples below (see also the references at the end). All applications of the DFT depend crucially on the availability of a fast algorithm to compute discrete Fourier transforms and their inverses, a fast Fourier transform.

Spectral analysis

When the DFT is used for signal spectral analysis, the $\{x_n\}$ sequence usually represents a finite set of uniformly spaced time-samples of some signal $x(t)$, where t represents time. The conversion from continuous time to samples (discrete-time) changes the underlying Fourier transform of $x(t)$ into a discrete-time Fourier transform (DTFT), which generally entails a type of distortion called aliasing. Choice of an appropriate sample-rate (see *Nyquist rate*) is the key to minimizing that distortion. Similarly, the conversion from a very long (or infinite) sequence to a manageable size entails a type of distortion called *leakage*, which is manifested as a

loss of detail (a.k.a. resolution) in the DTFT. Choice of an appropriate sub-sequence length is the primary key to minimizing that effect. When the available data (and time to process it) is more than the amount needed to attain the desired frequency resolution, a standard technique is to perform multiple DFTs, for example to create a spectrogram. If the desired result is a power spectrum and noise or randomness is present in the data, averaging the magnitude components of the multiple DFTs is a useful procedure to reduce the variance of the spectrum (also called a periodogram in this context); two examples of such techniques are the Welch method and the Bartlett method; the general subject of estimating the power spectrum of a noisy signal is called spectral estimation.
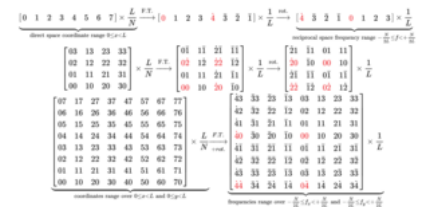
Discrete transforms embedded in time & space.

A final source of distortion (or perhaps *illusion*) is the DFT itself, because it is just a discrete sampling of the DTFT, which is a function of a continuous frequency domain. That can be mitigated by increasing the resolution of the DFT. That procedure is illustrated at § Sampling the DTFT.

- The procedure is sometimes referred to as *zero-padding*, which is a particular implementation used in conjunction with the fast Fourier transform (FFT) algorithm. The inefficiency of performing multiplications and additions with zero-valued "samples" is more than offset by the inherent efficiency of the FFT.
- As already stated, leakage imposes a limit on the inherent resolution of the DTFT, so there is a practical limit to the benefit that can be obtained from a fine-grained DFT.

Optics, diffraction, and tomography

The discrete Fourier transform is widely used with spatial frequencies in modeling the way that light, electrons, and other probes travel through optical systems and scatter from objects in two and three dimensions. The dual (direct/reciprocal) vector space of three dimensional objects further makes available a three dimensional reciprocal lattice, whose construction from translucent object shadows (via the Fourier slice theorem) allows tomographic reconstruction of three dimensional objects with a wide range of applications e.g. in modern medicine.

Filter bank

See § FFT filter banks and § Sampling the DTFT.

Data compression

The field of digital signal processing relies heavily on operations in the frequency domain (i.e. on the Fourier transform). For example, several lossy image and sound compression methods employ the discrete Fourier transform: the signal is cut into short segments, each is transformed, and then the Fourier coefficients of high frequencies, which are assumed to be unnoticeable, are discarded. The decompressor computes the inverse transform based on this reduced number of Fourier coefficients. (Compression applications often use a specialized form of the DFT, the discrete cosine transform or sometimes the modified discrete cosine transform.) Some relatively recent compression algorithms, however, use wavelet transforms, which give a more uniform compromise between time and frequency domain than obtained by chopping data into segments and transforming each segment. In the case of JPEG2000, this avoids the spurious image features that appear when images are highly compressed with the original JPEG.

Partial differential equations

Discrete Fourier transforms are often used to solve partial differential equations, where again the DFT is used as an approximation for the Fourier series (which is recovered in the limit of infinite N). The advantage of this approach is that it expands the signal in complex exponentials e^{inx}, which are eigenfunctions of differentiation: $\mathrm{d}(e^{inx})/\mathrm{d}x = in e^{inx}$. Thus, in the Fourier representation, differentiation is simple—we just multiply by in. (However, the choice of n is not unique due to aliasing; for the method to be convergent, a choice similar to that in the trigonometric interpolation section above should be used.) A linear differential equation with constant coefficients is transformed into an easily solvable algebraic equation. One then uses the inverse DFT to transform the result back into the ordinary spatial representation. Such an approach is called a spectral method.

Polynomial multiplication

Suppose we wish to compute the polynomial product $c(x) = a(x) \cdot b(x)$. The ordinary product expression for the coefficients of c involves a linear (acyclic) convolution, where indices do not "wrap around." This can be rewritten as a cyclic convolution by taking the coefficient vectors for $a(x)$ and $b(x)$ with constant term first, then appending zeros so that the resultant coefficient vectors $\mathbf{a}$ and $\mathbf{b}$ have dimension $d > \deg(a(x)) + \deg(b(x))$. Then,

$$\mathbf{c} = \mathbf{a} * \mathbf{b}$$

Where $\mathbf{c}$ is the vector of coefficients for $c(x)$, and the convolution operator $*$ is defined so

$$c_n = \sum_{m=0}^{d-1} a_m b_{n-m \bmod d} \qquad n = 0, 1 \ldots, d-1$$

But convolution becomes multiplication under the DFT:

$$\mathcal{F}(\mathbf{c}) = \mathcal{F}(\mathbf{a})\mathcal{F}(\mathbf{b})$$

Here the vector product is taken elementwise. Thus the coefficients of the product polynomial $c(x)$ are just the terms 0, ..., $\deg(a(x)) + \deg(b(x))$ of the coefficient vector

$$\mathbf{c} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{a})\mathcal{F}(\mathbf{b})).$$

With a fast Fourier transform, the resulting algorithm takes $O(N \log N)$ arithmetic operations. Due to its simplicity and speed, the Cooley–Tukey FFT algorithm, which is limited to composite sizes, is often chosen for the transform operation. In this case, d should be chosen as the smallest integer greater than the sum of the input polynomial degrees that is factorizable into small prime factors (e.g. 2, 3, and 5, depending upon the FFT implementation).

Multiplication of large integers

The fastest known <u>algorithms</u> for the multiplication of very large <u>integers</u> use the polynomial multiplication method outlined above. Integers can be treated as the value of a polynomial evaluated specifically at the number base, with the coefficients of the polynomial corresponding to the digits in that base (ex. $123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$). After polynomial multiplication, a relatively low-complexity carry-propagation step completes the multiplication.

Convolution

When data is <u>convolved</u> with a function with wide support, such as for downsampling by a large sampling ratio, because of the <u>Convolution theorem</u> and the FFT algorithm, it may be faster to transform it, multiply pointwise by the transform of the filter and then reverse transform it. Alternatively, a good filter is obtained by simply truncating the transformed data and re-transforming the shortened data set.

Some discrete Fourier transform pairs

		Some DFT pairs
$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i 2\pi k n / N}$	$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k n / N}$	**Note**
$x_n e^{i 2\pi n \ell / N}$	$X_{k-\ell}$	Frequency shift theorem
$x_{n-\ell}$	$X_k e^{-i 2\pi k \ell / N}$	Time shift theorem
$x_n \in \mathbb{R}$	$X_k = X_{N-k}^*$	Real DFT
a^n	$\begin{cases} N & \text{if } a = e^{i 2\pi k / N} \\ \frac{1-a^N}{1 - a\, e^{-i 2\pi k / N}} & \text{otherwise} \end{cases}$	from the <u>geometric progression</u> formula
$\binom{N-1}{n}$	$\left(1 + e^{-i 2\pi k / N}\right)^{N-1}$	from the <u>binomial theorem</u>
$\begin{cases} \frac{1}{W} & \text{if } 2n < W \text{ or } 2(N-n) < W \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } k = 0 \\ \frac{\sin\left(\frac{\pi W k}{N}\right)}{W \sin\left(\frac{\pi k}{N}\right)} & \text{otherwise} \end{cases}$	x_n is a rectangular <u>window function</u> of W points centered on n=0, where W is an <u>odd integer</u>, and X_k is a <u>sinc</u>-like function (specifically, X_k is a <u>Dirichlet kernel</u>)
$\sum_{j \in \mathbb{Z}} \exp\left(-\frac{\pi}{cN} \cdot (n + N \cdot j)^2\right)$	$\sqrt{cN} \cdot \sum_{j \in \mathbb{Z}} \exp\left(-\frac{\pi c}{N} \cdot (k + N \cdot j)^2\right)$	<u>Discretization</u> and <u>periodic summation</u> of the scaled <u>Gaussian functions</u> for $c > 0$. Since either c or $\frac{1}{c}$ is larger than one and thus warrants fast convergence of one of the two series, for large c you may choose to compute the frequency spectrum and convert to the time domain using the discrete Fourier transform.

Generalizations

Representation theory

The DFT can be interpreted as a complex-valued <u>representation</u> of the finite <u>cyclic group</u>. In other words, a sequence of n complex numbers can be thought of as an element of n-dimensional complex space $\mathbb{C}^n$ or equivalently a function f from the finite cyclic group of order n to the complex numbers, $\mathbb{Z}_n \mapsto \mathbb{C}$. So f is a <u>class function</u> on the finite cyclic group, and thus can be expressed as a linear combination of the irreducible characters of this group, which are the roots of unity.

From this point of view, one may generalize the DFT to representation theory generally, or more narrowly to the <u>representation theory of finite groups</u>.

More narrowly still, one may generalize the DFT by either changing the target (taking values in a field other than the complex numbers), or the domain (a group other than a finite cyclic group), as detailed in the sequel.

Other fields

Many of the properties of the DFT only depend on the fact that $e^{-\frac{i 2\pi}{N}}$ is a <u>primitive root of unity</u>, sometimes denoted ω_N or W_N (so that $\omega_N^N = 1$). Such properties include the completeness, orthogonality, Plancherel/Parseval, periodicity, shift, convolution, and unitarity properties above, as well as many FFT algorithms. For this reason, the discrete Fourier transform can be defined by using roots of unity in <u>fields</u> other than the complex numbers, and such generalizations are commonly called *number-theoretic transforms* (NTTs) in the case of <u>finite fields</u>. For more information, see <u>number-theoretic transform</u> and <u>discrete Fourier transform (general)</u>.

Other finite groups

The standard DFT acts on a sequence x_0, x_1, ..., x_{N-1} of complex numbers, which can be viewed as a function $\{0, 1, ..., N-1\} \to \mathbb{C}$. The multidimensional DFT acts on multidimensional sequences, which can be viewed as functions

$$\{0, 1, \ldots, N_1 - 1\} \times \cdots \times \{0, 1, \ldots, N_d - 1\} \to \mathbb{C}.$$

This suggests the generalization to <u>Fourier transforms on arbitrary finite groups</u>, which act on functions $G \to \mathbb{C}$ where G is a <u>finite group</u>. In this framework, the standard DFT is seen as the Fourier transform on a <u>cyclic group</u>, while the multidimensional DFT is a Fourier transform on a direct sum of cyclic groups.

Further, Fourier transform can be on cosets of a group.

Alternatives

There are various alternatives to the DFT for various applications, prominent among which are wavelets. The analog of the DFT is the discrete wavelet transform (DWT). From the point of view of time–frequency analysis, a key limitation of the Fourier transform is that it does not include *location* information, only *frequency* information, and thus has difficulty in representing transients. As wavelets have location as well as frequency, they are better able to represent location, at the expense of greater difficulty representing frequency. For details, see comparison of the discrete wavelet transform with the discrete Fourier transform.

See also

- Companion matrix
- DFT matrix
- Fast Fourier transform
- FFTPACK
- FFTW
- Generalizations of Pauli matrices
- Least-squares spectral analysis
- List of Fourier-related transforms
- Multidimensional transform
- Zak transform
- Quantum Fourier transform

Notes

A. As a linear transformation on a finite-dimensional vector space, the DFT expression can also be written in terms of a DFT matrix; when scaled appropriately it becomes a unitary matrix and the X_k can thus be viewed as coefficients of *x* in an orthonormal basis.

B. Time reversal for the DFT means replacing n by $N - n$ and not n by $-n$ to avoid negative indices.

References

1. Strang, Gilbert (May–June 1994). "Wavelets". *American Scientist*. **82** (3): 250–255. JSTOR 29775194 (https://www.jstor.org/stable/29775194). "This is the most important numerical algorithm of our lifetime..."

2. Sahidullah, Md.; Saha, Goutam (Feb 2013). "A Novel Windowing Technique for Efficient Computation of MFCC for Speaker Recognition". *IEEE Signal Processing Letters*. **20** (2): 149–152. arXiv:1206.2437 (https://arxiv.org/abs/1206.2437). Bibcode:2013ISPL...20..149S (https://ui.adsabs.harvard.edu/abs/2013ISPL...20..149S). doi:10.1109/LSP.2012.2235067 (https://doi.org/10.1109%2FLSP.2012.2235067). S2CID 10900793 (https://api.semanticscholar.org/CorpusID:10900793).

3. J. Cooley, P. Lewis, and P. Welch (1969). "The finite Fourier transform". *IEEE Transactions on Audio and Electroacoustics*. **17** (2): 77–85. doi:10.1109/TAU.1969.1162036 (https://doi.org/10.1109%2FTAU.1969.1162036).

4. "Shift zero-frequency component to center of spectrum – MATLAB fftshift" (http://www.mathworks.com/help/matlab/ref/fftshift.html). *mathworks.com*. Natick,MA 01760: The MathWorks, Inc. Retrieved 10 March 2014.

5. Proakis, John G.; Manolakis, Dimitri G. (1996), *Digital Signal Processing: Principles, Algorithms and Applications* (https://archive.org/details/digitalsignalpro00proa) (3 ed.), Upper Saddle River,NJ: Prentice-Hall International, Bibcode:1996dspp.book.....P (https://ui.adsabs.harvard.edu/abs/1996dspp.book.....P), ISBN 9780133942897, sAcfAQAAIAAJ

6. Oppenheim, Alan V.; Schafer, Ronald W.; Buck, John R. (1999). *Discrete-time signal processing* (https://archive.org/details/discretetimesign00alan) (2nd ed.). Upper Saddle River, N.J.: Prentice Hall. p. 571 (https://archive.org/details/discretetimesign00alan/page/571). ISBN 0-13-754920-2. Also available at https://d1.amobbs.com/bbs_upload782111/files_24/ourdev_523225.pdf

7. McGillem, Clare D.; Cooper, George R. (1984). *Continuous and Discrete Signal and System Analysis* (2 ed.). Holt, Rinehart and Winston. pp. 171–172. ISBN 0-03-061703-0.

8. Massar, S.; Spindel, P. (2008). "Uncertainty Relation for the Discrete Fourier Transform". *Physical Review Letters*. **100** (19): 190401. arXiv:0710.0723 (https://arxiv.org/abs/0710.0723). Bibcode:2008PhRvL.100s0401M (https://ui.adsabs.harvard.edu/abs/2008PhRvL.100s0401M). doi:10.1103/PhysRevLett.100.190401 (https://doi.org/10.1103%2FPhysRevLett.100.190401). PMID 18518426 (https://pubmed.ncbi.nlm.nih.gov/18518426). S2CID 10076374 (https://api.semanticscholar.org/CorpusID:10076374).

9. DeBrunner, Victor; Havlicek, Joseph P.; Przebinda, Tomasz; Özaydin, Murad (2005). "Entropy-Based Uncertainty Measures for $L^2(\mathbb{R}^n), \ell^2(\mathbb{Z})$, and $\ell^2(\mathbb{Z}/N\mathbb{Z})$ With a Hirschman Optimal Transform for $\ell^2(\mathbb{Z}/N\mathbb{Z})$" (http://redwood.berkeley.edu/w/images/9/95/2002-26.pdf) (PDF). *IEEE Transactions on Signal Processing*. **53** (8): 2690. Bibcode:2005ITSP...53.2690D (https://ui.adsabs.harvard.edu/abs/2005ITSP...53.2690D). doi:10.1109/TSP.2005.850329 (https://doi.org/10.1109%2FTSP.2005.850329). Retrieved 2011-06-23.

10. Donoho, D.L.; Stark, P.B (1989). "Uncertainty principles and signal recovery". *SIAM Journal on Applied Mathematics*. **49** (3): 906–931. doi:10.1137/0149053 (https://doi.org/10.1137%2F0149053). S2CID 115142886 (https://api.semanticscholar.org/CorpusID:115142886).

11. Santhanam, Balu; Santhanam, Thalanayar S. "*Discrete Gauss-Hermite functions and eigenvectors of the centered discrete Fourier transform*" (http://thamakau.usc.edu/Proceedings/ICASSP%202007/pdfs/0301385.pdf), Proceedings of the 32nd IEEE *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP 2007, SPTM-P12.4), vol. III, pp. 1385-1388.

12. Akansu, Ali N.; Agirman-Tosun, Handan "*Generalized Discrete Fourier Transform With Nonlinear Phase*" (http://web.njit.edu/~akansu/PAPERS/AkansuIEEE-TSP2010.pdf), IEEE *Transactions on Signal Processing*, vol. 58, no. 9, pp. 4547–4556, Sept. 2010.

Further reading

- Brigham, E. Oran (1988). *The fast Fourier transform and its applications*. Englewood Cliffs, N.J.: Prentice Hall. ISBN 978-0-13-307505-2.
- Smith, Steven W. (1999). "Chapter 8: The Discrete Fourier Transform" (http://www.dspguide.com/ch8/1.htm). *The Scientist and Engineer's Guide to Digital Signal Processing* (Second ed.). San Diego, Calif.: California Technical Publishing. ISBN 978-0-9660176-3-2.
- Cormen, Thomas H.; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2001). "Chapter 30: Polynomials and the FFT". *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill. pp. 822 (https://archive.org/details/introductiontoal00corm_691/page/n844)–848. ISBN 978-0-262-03293-3. esp. section 30.2: The DFT and FFT, pp. 830–838.
- P. Duhamel; B. Piron; J. M. Etcheto (1988). "On computing the inverse DFT". *IEEE Transactions on Acoustics, Speech, and Signal Processing*. **36** (2): 285–286. doi:10.1109/29.1519 (https://doi.org/10.1109%2F29.1519).

- J. H. McClellan; T. W. Parks (1972). "Eigenvalues and eigenvectors of the discrete Fourier transformation". *IEEE Transactions on Audio and Electroacoustics*. **20** (1): 66–74. doi:10.1109/TAU.1972.1162342 (https://doi.org/10.1109%2FTAU.1972.1162342).
- Bradley W. Dickinson; Kenneth Steiglitz (1982). "Eigenvectors and functions of the discrete Fourier transform" (http://www.cs.princeton.edu/~ken/Eigenvectors82.pdf) (PDF). *IEEE Transactions on Acoustics, Speech, and Signal Processing*. **30** (1): 25–31. CiteSeerX 10.1.1.434.5279 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.434.5279). doi:10.1109/TASSP.1982.1163843 (https://doi.org/10.1109%2FTASSP.1982.1163843). (Note that this paper has an apparent typo in its table of the eigenvalue multiplicities: the +i/−i columns are interchanged. The correct table can be found in McClellan and Parks, 1972, and is easily confirmed numerically.)
- F. A. Grünbaum (1982). "The eigenvectors of the discrete Fourier transform" (https://doi.org/10.1016%2F0022-247X%2882%2990199-8). *Journal of Mathematical Analysis and Applications*. **88** (2): 355–363. doi:10.1016/0022-247X(82)90199-8 (https://doi.org/10.1016%2F0022-247X%2882%2990199-8).
- Natig M. Atakishiyev; Kurt Bernardo Wolf (1997). "Fractional Fourier-Kravchuk transform". *Journal of the Optical Society of America A*. **14** (7): 1467–1477. Bibcode:1997JOSAA..14.1467A (https://ui.adsabs.harvard.edu/abs/1997JOSAA..14.1467A). doi:10.1364/JOSAA.14.001467 (https://doi.org/10.1364%2FJOSAA.14.001467).
- C. Candan; M. A. Kutay; H. M.Ozaktas (2000). "The discrete fractional Fourier transform" (http://www.ee.bilkent.edu.tr/~haldun/publications/ozaktas166.pdf) (PDF). *IEEE Transactions on Signal Processing*. **48** (5): 1329–1337. Bibcode:2000ITSP...48.1329C (https://ui.adsabs.harvard.edu/abs/2000ITSP...48.1329C). doi:10.1109/78.839980 (https://doi.org/10.1109%2F78.839980). hdl:11693/11130 (https://hdl.handle.net/11693%2F11130).
- Magdy Tawfik Hanna, Nabila Philip Attalla Seif, and Waleed Abd El Maguid Ahmed (2004). "Hermite-Gaussian-like eigenvectors of the discrete Fourier transform matrix based on the singular-value decomposition of its orthogonal projection matrices". *IEEE Transactions on Circuits and Systems I: Regular Papers*. **51** (11): 2245–2254. doi:10.1109/TCSI.2004.836850 (https://doi.org/10.1109%2FTCSI.2004.836850). S2CID 14468134 (https://api.semanticscholar.org/CorpusID:14468134).
- Shamgar Gurevich; Ronny Hadani (2009). "On the diagonalization of the discrete Fourier transform". *Applied and Computational Harmonic Analysis*. **27** (1): 87–99. arXiv:0808.3281 (https://arxiv.org/abs/0808.3281). doi:10.1016/j.acha.2008.11.003 (https://doi.org/10.1016%2Fj.acha.2008.11.003). S2CID 14833478 (https://api.semanticscholar.org/CorpusID:14833478). preprint at.
- Shamgar Gurevich; Ronny Hadani; Nir Sochen (2008). "The finite harmonic oscillator and its applications to sequences, communication and radar". *IEEE Transactions on Information Theory*. **54** (9): 4239–4253. arXiv:0808.1495 (https://arxiv.org/abs/0808.1495). Bibcode:2008arXiv0808.1495G (https://ui.adsabs.harvard.edu/abs/2008arXiv0808.1495G). doi:10.1109/TIT.2008.926440 (https://doi.org/10.1109%2FTIT.2008.926440). S2CID 6037080 (https://api.semanticscholar.org/CorpusID:6037080). preprint at.
- Juan G. Vargas-Rubio; Balu Santhanam (2005). "On the multiangle centered discrete fractional Fourier transform". *IEEE Signal Processing Letters*. **12** (4): 273–276. Bibcode:2005ISPL...12..273V (https://ui.adsabs.harvard.edu/abs/2005ISPL...12..273V). doi:10.1109/LSP.2005.843762 (https://doi.org/10.1109%2FLSP.2005.843762). S2CID 1499353 (https://api.semanticscholar.org/CorpusID:1499353).
- F.N. Kong (2008). "Analytic Expressions of Two Discrete Hermite-Gaussian Signals". *IEEE Transactions on Circuits and Systems II: Express Briefs*. **55** (1): 56–60. doi:10.1109/TCSII.2007.909865 (https://doi.org/10.1109%2FTCSII.2007.909865). S2CID 5154718 (https://api.semanticscholar.org/CorpusID:5154718).

External links

- Interactive explanation of the DFT (https://jackschaedler.github.io/circles-sines-signals/)
- Matlab tutorial on the Discrete Fourier Transformation (https://www.nbtwiki.net/doku.php?id=tutorial:the_discrete_fourier_transformation_dft) Archived (https://web.archive.org/web/20160304200056/http://www.nbtwiki.net/doku.php?id=tutorial:the_discrete_fourier_transformation_dft) 2016-03-04 at the Wayback Machine
- Interactive flash tutorial on the DFT (https://web.archive.org/web/20160523205720/http://www.fourier-series.com/fourierseries2/DFT_tutorial.html)
- Mathematics of the Discrete Fourier Transform by Julius O. Smith III (http://ccrma.stanford.edu/~jos/mdft/mdft.html)
- FFTW: Fast implementation of the DFT - coded in C and under General Public License (GPL) (http://www.fftw.org)
- General Purpose FFT Package: Yet another fast DFT implementation in C & FORTRAN, permissive license (http://www.kurims.kyoto-u.ac.jp/~ooura/fft.html)
- Explained: The Discrete Fourier Transform (http://web.mit.edu/newsoffice/2009/explained-fourier.html)
- Discrete Fourier Transform (https://web.archive.org/web/20171010173422/http://en.dsplib.org/content/dft.html)
- Indexing and shifting of Discrete Fourier Transform (https://web.archive.org/web/20171113135838/http://en.dsplib.org/content/dft_freq.html)
- Discrete Fourier Transform Properties (https://web.archive.org/web/20171106092801/http://en.dsplib.org/content/dft_prop.html)
- Generalized Discrete Fourier Transform (GDFT) with Nonlinear Phase (https://web.njit.edu/~akansu/gdft.htm)