

# Discrete Fourier transform

From Wikipedia, the free encyclopedia

In mathematics, the **discrete Fourier transform (DFT)** is a specific kind of discrete transform, used in Fourier analysis. It transforms one function into another, which is called the *frequency domain* representation, or simply the *DFT*, of the original function (which is often a function in the time domain). The DFT requires an input function that is *discrete*. Such inputs are often created by sampling a continuous function, such as the amplitude of a person's voice over time. The discrete input function must also have a limited (*finite*) duration, such as one period of a periodic sequence or a windowed segment of a longer sequence. Unlike the discrete-time Fourier transform (DTFT), the DFT only evaluates enough frequency components to reconstruct the finite segment that was analyzed. The inverse DFT cannot reproduce the entire time domain, unless the input happens to be periodic. Therefore it is often said that the DFT is a transform for Fourier analysis of finite-domain discrete-time functions.

The input to the DFT is a finite sequence of real or complex numbers (with more abstract generalizations discussed below), making the DFT ideal for processing information stored in computers. In particular, the DFT 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 or multiplying large integers. A key enabling factor for these applications is the fact that the DFT can be computed efficiently in practice using a fast Fourier transform (FFT) algorithm.

FFT algorithms are so commonly employed to compute DFTs that the term "FFT" is often used to mean "DFT" in colloquial settings. Formally, there is a clear distinction: "DFT" refers to a mathematical transformation or function, regardless of how it is computed, whereas "FFT" refers to a specific family of algorithms for computing DFTs. The terminology is further blurred by the (now rare) synonym finite Fourier transform for the DFT, which apparently predates the term "fast Fourier transform" (Cooley et al., 1969) but has the same initialism.

## Contents

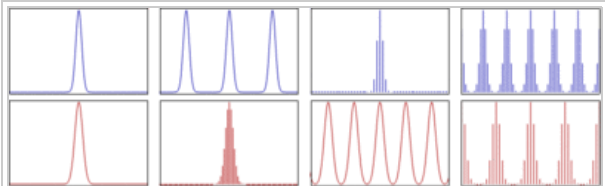
- 1 Definition
- 2 Properties
  - 2.1 Completeness
  - 2.2 Orthogonality
  - 2.3 The Plancherel theorem and Parseval's theorem
  - 2.4 Periodicity
  - 2.5 The shift theorem
  - 2.6 Circular convolution theorem and cross-correlation theorem
  - 2.7 Convolution theorem duality
  - 2.8 Trigonometric interpolation polynomial
  - 2.9 The unitary DFT
  - 2.10 Expressing the inverse DFT in terms of the DFT
  - 2.11 Eigenvalues and eigenvectors
  - 2.12 Uncertainty principle
  - 2.13 The real-input DFT
- 3 Generalized DFT (shifted and non-linear phase)
- 4 Multidimensional DFT
  - 4.1 The real-input multidimensional DFT
- 5 Applications
  - 5.1 Spectral analysis
  - 5.2 Filter bank
  - 5.3 Data compression
  - 5.4 Partial differential equations
  - 5.5 Polynomial multiplication
    - 5.5.1 Multiplication of large integers
    - 5.5.2 Convolution
- 6 Some discrete Fourier transform pairs
- 7 Generalizations
  - 7.1 Representation theory
  - 7.2 Other fields
  - 7.3 Other finite groups
- 8 Alternatives
- 9 See also
- 10 Notes
- 11 Citations
- 12 References
- 13 External links

### Fourier transforms

- Continuous Fourier transform
- Fourier series
- Discrete-time Fourier transform

### Discrete Fourier transform

- Fourier Analysis
- Related transforms



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.

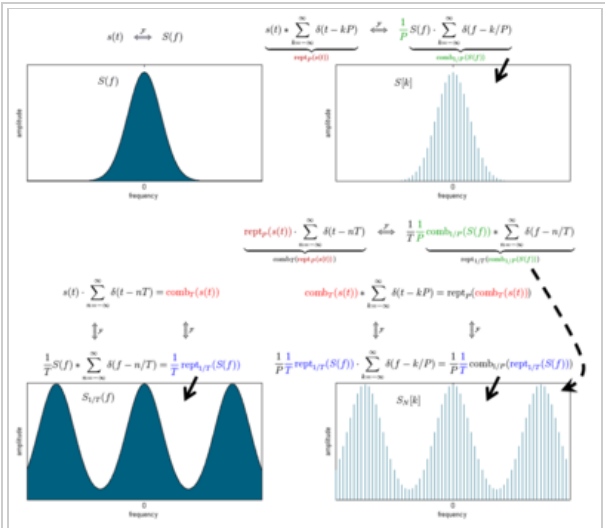


Illustration of using Dirac comb functions and the convolution theorem to model the effects of sampling and/or periodic summation. The graphs on the right side depict the (finite) coefficients that modulate the infinite amplitudes of a comb function whose *teeth* are spaced at the reciprocal of the time-domain periodicity. The coefficients in the upper figure are computed by the Fourier series integral. The **DFT** computes the coefficients in the lower figure. Its similarities to the original transform,  $S(f)$ , and its relative computational ease are often the motivation for computing a DFT. The lower left graph represents a discrete-time Fourier transform (DTFT).

## Definition

The sequence of  $N$  complex numbers  $x_0, \dots, x_{N-1}$  is transformed into another sequence of  $N$  complex numbers according to the DFT formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}. \quad (\text{Eq.1})$$

The transform is sometimes denoted by the symbol  $\mathcal{F}$ , as in  $\mathbf{X} = \mathcal{F}\{\mathbf{x}\}$  or  $\mathcal{F}(\mathbf{x})$  or  $\mathcal{F}_{\mathbf{x}}$ <sup>[note 1]</sup>.

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

- It completely describes the discrete-time Fourier transform (DTFT) of an  $N$ -periodic sequence, which comprises only discrete frequency components. (Discrete-time Fourier transform#Periodic data)
- It can also provide uniformly spaced samples of the continuous DTFT of a finite length sequence. (Sampling the DTFT)
- It is the discrete analog of the formula for the coefficients of a Fourier series:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi \frac{k}{N} n}, \quad (\text{Eq.2})$$

which is the inverse DFT (**IDFT**). Each  $X_k$  is a complex number that encodes both amplitude and phase of a sinusoidal component of function  $x_n$ .

The sinusoid's frequency is  $\frac{k}{N}$  cycles per sample. Its amplitude and phase are:

$$\begin{aligned} |X_k|/N &= \sqrt{\text{Re}(X_k)^2 + \text{Im}(X_k)^2}/N \\ \arg(X_k) &= \text{atan2}(\text{Im}(X_k), \text{Re}(X_k)), \end{aligned}$$

where atan2 is the two-argument form of the arctan function. The normalization factor multiplying the DFT and IDFT (here 1 and 1/N) and the signs of the exponents are merely conventions, 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 1/N. A normalization of  $\sqrt{1/N}$  for both the DFT and IDFT makes the transforms unitary, which has some theoretical advantages. But it is often more practical in numerical computation to perform the scaling all at once as above (and a unit scaling can be convenient in other ways).

(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",  $+\frac{k}{N}$ . 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.)

In the following discussion the terms "sequence" and "vector" will be considered interchangeable.

## Properties

### Completeness

The discrete Fourier transform is an invertible, linear transformation

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

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.

### Orthogonality

The vectors  $u_k = \left[ e^{\frac{2\pi i}{N} kn} \mid n = 0, 1, \dots, N-1 \right]^T$  form an orthogonal basis over the set of  $N$ -dimensional complex vectors:

$$u_k^T u_{k'}^* = \sum_{n=0}^{N-1} \left( e^{\frac{2\pi i}{N} kn} \right) \left( e^{\frac{2\pi i}{N} (-k')n} \right) = \sum_{n=0}^{N-1} e^{\frac{2\pi i}{N} (k-k')n} = N \delta_{kk'}$$

where  $\delta_{kk'}$  is the Kronecker delta. (In the last step, the summation is trivial if  $k = k'$ , where it is  $1+1+\dots=N$ , and otherwise is a geometric series 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 Plancherel theorem 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 complex conjugation. Parseval's theorem is a special case of the Plancherel 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

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:

$$X_{k+N} \stackrel{\text{def}}{=} \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}(k+N)n} = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \underbrace{e^{-2\pi i n}}_1 = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} = X_k.$$

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

## 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  $\mathbf{x}$  then

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

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

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

## Circular convolution theorem and cross-correlation theorem

The convolution theorem for the discrete-time Fourier transform indicates that a convolution of two infinite sequences can be obtained as the inverse transform of the product of the individual transforms. An important simplification occurs when the sequences are of finite length,  $N$ . In terms of the DFT and inverse DFT, it can be written as follows:

$$\mathcal{F}^{-1}\{\mathbf{X} \cdot \mathbf{Y}\}_n = \sum_{l=0}^{N-1} x_l \cdot (y_N)_{n-l} \stackrel{\text{def}}{=} (\mathbf{x} * \mathbf{y}_N)_n,$$

which is the convolution of the  $\mathbf{x}$  sequence with a  $\mathbf{y}$  sequence extended by periodic summation:

$$(\mathbf{y}_N)_n \stackrel{\text{def}}{=} \sum_{p=-\infty}^{\infty} y_{(n-pN)} = y_{n(\text{mod}N)}.$$

Similarly, the cross-correlation of  $\mathbf{x}$  and  $\mathbf{y}_N$  is given by:

$$\mathcal{F}^{-1}\{\mathbf{X}^* \cdot \mathbf{Y}\}_n = \sum_{l=0}^{N-1} x_l^* \cdot (y_N)_{n+l} \stackrel{\text{def}}{=} (\mathbf{x} \star \mathbf{y}_N)_n.$$

A direct evaluation of either summation (above) requires  $\mathcal{O}(N^2)$  operations for an output sequence of length  $N$ . An indirect method, using transforms, can take advantage of the  $\mathcal{O}(N \log N)$  efficiency of the fast Fourier transform (FFT) to achieve much better performance. Furthermore, convolutions can be used to efficiently compute DFTs via Rader's FFT algorithm and Bluestein's FFT algorithm.

Methods have also been developed to use **circular convolution** as part of an efficient process that achieves normal (non-circular) convolution with an  $\mathbf{x}$  or  $\mathbf{y}$  sequence potentially much longer than the practical transform size ( $N$ ). Two such methods are called overlap-save and overlap-add.<sup>[1]</sup>

## Convolution theorem duality

It can also be shown that:

$$\begin{aligned} \mathcal{F}\{\mathbf{x} \cdot \mathbf{y}\}_k &\stackrel{\text{def}}{=} \sum_{n=0}^{N-1} x_n \cdot y_n \cdot e^{-\frac{2\pi i}{N}kn} \\ &= \frac{1}{N}(\mathbf{X} * \mathbf{Y}_N)_k, \text{ which is the circular convolution of } \mathbf{X} \text{ and } \mathbf{Y}. \end{aligned}$$

## Trigonometric interpolation polynomial

The trigonometric interpolation polynomial

$$p(t) = \frac{1}{N} \left[ X_0 + X_1 e^{2\pi i t} + \dots + X_{N/2-1} e^{(N/2-1)2\pi i t} + X_{N/2} \cos(Nt/2) + X_{N/2+1} e^{(-N/2+1)2\pi i t} + \dots + X_{N-1} e^{-2\pi i t} \right]$$

for  $N$  even,

$$p(t) = \frac{1}{N} \left[ X_0 + X_1 e^{2\pi i t} + \dots + X_{\lfloor N/2 \rfloor} e^{\lfloor N/2 \rfloor 2\pi i t} + X_{\lfloor N/2 \rfloor + 1} e^{-\lfloor N/2 \rfloor 2\pi i t} + \dots + X_{N-1} e^{-2\pi i t} \right] \text{ for } N \text{ odd,}$$

where the coefficients  $X_k$  are given by the DFT of  $x_n$  above, satisfies the interpolation property  $p(2\pi n/N) = x_n$  for  $n = 0, \dots, N-1$ .

For even  $N$ , notice that the Nyquist component  $\frac{X_{N/2}}{N} \cos(Nt/2)$  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 bandlimited. 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 a Vandermonde matrix:

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

where

$$\omega_N = e^{-2\pi i/N}$$

is a primitive  $N$ th 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:

$$\begin{aligned} \mathbf{U} &= \mathbf{F}/\sqrt{N} \\ \mathbf{U}^{-1} &= \mathbf{U}^* \\ |\det(\mathbf{U})| &= 1 \end{aligned}$$

where  $\det()$  is the determinant function. The determinant is the product of the eigenvalues, which are always  $\pm 1$  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 Plancherel 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 Parseval's theorem:

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

## 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 the inputs:

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

(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:

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

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

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

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  $(1+i)/\sqrt{2}$ ) 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 **U** defined above for the DFT of length  $N$ , where

$$U_{m,n} = \frac{1}{\sqrt{N}} \omega_N^{(m-1)(n-1)} = \frac{1}{\sqrt{N}} e^{-\frac{2\pi i}{N}(m-1)(n-1)}.$$

This matrix satisfies the matrix polynomial equation:

$$U^4 = I.$$

This can be seen from the inverse properties above: operating **U** twice gives the original data in reverse order, so operating **U** four times gives back the original data and is thus the identity matrix. This means that the eigenvalues  $\lambda$  satisfy the equation:

$$\lambda^4 = 1.$$

Therefore, the eigenvalues of **U** are the fourth roots of unity:  $\lambda$  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. (Note that 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:

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

size N	λ = +1	λ = -1	λ = -i	λ = +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

Otherwise stated, the characteristic polynomial of **U** is:

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

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 the eigenfunction of the continuous Fourier transform, namely 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:

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

A closed form expression for the series is not known, but it converges rapidly.

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:

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

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

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

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 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. Nevertheless, a meaningful uncertainty principle has been introduced by Massar and Spindel.<sup>[2]</sup>

However, the Hirschman uncertainty will have a useful analog for the case of the DFT.<sup>[3]</sup> 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 Hirschman uncertainty principle becomes:<sup>[3]</sup>

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

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  $B=N/A$ .<sup>[3]</sup>

## The real-input DFT

If  $x_0, \dots, x_{N-1}$  are real numbers, as they often are in practical applications, then the DFT obeys the symmetry:

$$X_{N-k} \equiv X_{-k} = X_k^*, \text{ where } X^* \text{ denotes complex conjugation.}$$

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

## 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:

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

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  $O^2$  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 (see above) have equal multiplicities (Rubio and Santhanam, 2005)<sup>[4]</sup>

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).<sup>[5]</sup>

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  $x_{n_1, n_2, \dots, n_d}$  that is a function of  $d$  discrete variables  $n_\ell = 0, 1, \dots, N_\ell - 1$  for  $\ell$  in  $1, 2, \dots, d$  is defined by:

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

where  $\omega_{N_\ell} = \exp(-2\pi i/N_\ell)$  as above and the  $d$  output indices run from  $k_\ell = 0, 1, \dots, N_\ell - 1$ . This is more compactly expressed in vector notation, where we define  $\mathbf{n} = (n_1, n_2, \dots, n_d)$  and  $\mathbf{k} = (k_1, k_2, \dots, k_d)$  as  $d$ -dimensional vectors of indices from 0 to  $\mathbf{N} - \mathbf{1}$ , which we define as  $\mathbf{N} - \mathbf{1} = (N_1 - 1, N_2 - 1, \dots, N_d - 1)$ :

$$X_{\mathbf{k}} = \sum_{\mathbf{n}=0}^{\mathbf{N}-1} e^{-2\pi i \mathbf{k} \cdot (\mathbf{n}/\mathbf{N})} x_{\mathbf{n}},$$

where the division  $\mathbf{n}/\mathbf{N}$  is defined as  $\mathbf{n}/\mathbf{N} = (n_1/N_1, \dots, n_d/N_d)$  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:

$$x_{\mathbf{n}} = \frac{1}{\prod_{\ell=1}^d N_\ell} \sum_{\mathbf{k}=0}^{\mathbf{N}-1} e^{2\pi i \mathbf{n} \cdot (\mathbf{k}/\mathbf{N})} X_{\mathbf{k}}.$$

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

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

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

# 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 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 (aka 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.

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 noted, 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.

## 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:  $d/dx e^{inx} = in e^{inx}$ . Thus, in the Fourier representation, differentiation is simple—we just multiply by  $in$ . (Note, however, that 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

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

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  $0, \dots, \deg(a(x)) + \deg(b(x))$  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 algorithms for the multiplication of very large integers 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. After polynomial multiplication, a relatively low-complexity carry-propagation step completes the multiplication.

Convolution

When data is convolved with a function with wide support, such as for downsampling by a large sampling ratio, because of the Convolution theorem 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		
$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i2\pi kn/N}$	$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$	Note
$x_n e^{i2\pi n\ell/N}$	$X_{k-\ell}$	Shift theorem
$x_{n-\ell}$	$X_k e^{-i2\pi k\ell/N}$	
$x_n \in \mathbb{R}$	$X_k = X_{N-k}^*$	Real DFT
$a^n$	$\begin{cases} N & \text{if } a = e^{i2\pi k/N} \\ \frac{1-a^N}{1-a e^{-i2\pi k/N}} & \text{otherwise} \end{cases}$	from the geometric progression formula
$\binom{N-1}{n}$	$\left(1 + e^{-i2\pi k/N}\right)^{N-1}$	from the binomial theorem
$\begin{cases} \frac{1}{W} & \text{if } 2n < W \text{ or } 2(N-n) < W \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } k = 0 \\ \frac{\sin(\frac{\pi W k}{N})}{W \sin(\frac{\pi k}{N})} & \text{otherwise} \end{cases}$	$x_n$ is a rectangular window function of $W$ points centered on $n=0$ , where $W$ is an odd integer, and $X_k$ is a sinc-like function (specifically, $X_k$ is a Dirichlet kernel)
$\sum_{j \in \mathbb{Z}} \exp\left(-\frac{\pi}{cN} \cdot (n + N \cdot j)^2\right)$	$\sqrt{cN} \cdot \sum_{j \in \mathbb{Z}} \exp\left(-\frac{\pi c}{N} \cdot (k + N \cdot j)^2\right)$	Discretization and periodic summation of the scaled Gaussian functions 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

For more details on this topic, see Representation theory of finite groups#Discrete Fourier transform.

The DFT can be interpreted as the complex-valued representation theory of the finite cyclic group. In other words, a sequence of  $n$  complex numbers can be thought of as an element of  $n$ -dimensional complex space  $\mathbf{C}^n$  or equivalently a function from the finite cyclic group of order  $n$  to the complex numbers,  $\mathbf{Z}_n \rightarrow \mathbf{C}$ . This latter may be suggestively written  $\mathbf{C}^{\mathbf{Z}_n}$  to emphasize that this is a complex vector space whose coordinates are indexed by the  $n$ -element set  $\mathbf{Z}_n$ .

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

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

Main articles: Discrete Fourier transform (general) and Number-theoretic transform

Many of the properties of the DFT only depend on the fact that  $e^{-\frac{2\pi i}{N}}$  is a primitive root of unity, sometimes denoted  $\omega_N$  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 fields other than the complex numbers, and such generalizations are commonly called *number-theoretic transforms* (NTTs) in the case of finite fields. For more information, see number-theoretic transform and discrete Fourier transform (general).

Other finite groups

Main article: Fourier transform on finite groups

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

$$\{0, 1, \dots, N_1 - 1\} \times \dots \times \{0, 1, \dots, N_d - 1\} \rightarrow \mathbb{C}.$$

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

## Alternatives

*Main article: Discrete wavelet transform*

*For more details on this topic, see Discrete wavelet transform#Comparison with Fourier transform.*

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

- DFT matrix
- Fast Fourier transform
- List of Fourier-related transforms
- FFTW
- FFTPACK
- Generalizations of Pauli matrices
- companion matrix

## Notes

- <sup>^</sup> 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.

## Citations

- <sup>^</sup> T. G. Stockham, Jr., "High-speed convolution and correlation," in 1966 *Proc. AFIPS Spring Joint Computing Conf.* Reprinted in Digital Signal Processing, L. R. Rabiner and C. M. Rader, editors, New York: IEEE Press, 1972.
- <sup>^</sup> Massar, S.; Spindel, P. (2008). "Uncertainty Relation for the Discrete Fourier Transform". *Physical Review Letters* **100** (19). doi:10.1103/PhysRevLett.100.190401 (<http://dx.doi.org/10.1103%2FPhysRevLett.100.190401>) .
- <sup>^</sup> <sup>a</sup> <sup>b</sup> <sup>c</sup> 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>) . *IEEE Transactions on Signal Processing* **53** (8): 2690. <http://redwood.berkeley.edu/w/images/9/95/2002-26.pdf>. Retrieved 2011-06-23.
- <sup>^</sup> Santhanam, Balu; Santhanam, Thalanayar S. "Discrete Gauss-Hermite functions and eigenvectors of the centered discrete Fourier transform" (<http://thamaku.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.
- <sup>^</sup> 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.

## References

- Brigham, E. Oran (1988). *The fast Fourier transform and its applications*. Englewood Cliffs, N.J.: Prentice Hall. ISBN 0-13-307505-2.
- Oppenheim, Alan V.; Schaffer, R. W.; and Buck, J. R. (1999). *Discrete-time signal processing*. Upper Saddle River, N.J.: Prentice Hall. ISBN 0-13-754920-2.
- 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 0-9660176-3-3. <http://www.dspguide.com/ch8/1.htm>.
- Cormen, Thomas H.; Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001). "Chapter 30: Polynomials and the FFT". *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill. pp. 822–848. ISBN 0-262-03293-7. esp. section 30.2: The DFT and FFT, pp. 830–838.
- P. Duhamel, B. Piron, and J. M. Etcheto (1988). "On computing the inverse DFT". *IEEE Trans. Acoust., Speech and Sig. Processing* **36** (2): 285–286. doi:10.1109/29.1519 (<http://dx.doi.org/10.1109%2F29.1519>) .
- J. H. McClellan and T. W. Parks (1972). "Eigenvalues and eigenvectors of the discrete Fourier transformation". *IEEE Trans. Audio Electroacoust.* **20** (1): 66–74. doi:10.1109/TAU.1972.1162342 (<http://dx.doi.org/10.1109%2FTAU.1972.1162342>) .
- Bradley W. Dickinson and Kenneth Steiglitz (1982). "Eigenvectors and functions of the discrete Fourier transform". *IEEE Trans. Acoust., Speech and Sig. Processing* **30** (1): 25–31. doi:10.1109/TASSP.1982.1163843 (<http://dx.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". *J. Math. Anal. Appl.* **88** (2): 355–363. doi:10.1016/0022-247X(82)90199-8 (<http://dx.doi.org/10.1016%2F0022-247X%2882%2990199-8>) .
- Natig M. Atakishiyev and Kurt Bernardo Wolf (1997). "Fractional Fourier-Kravchuk transform". *J. Opt. Soc. Am. A* **14** (7): 1467–1477. doi:10.1364/JOSAA.14.001467 (<http://dx.doi.org/10.1364%2FJOSAA.14.001467>) .
- C. Candan, M. A. Kutay and H. M. Ozaktas (2000). "The discrete fractional Fourier transform". *IEEE Trans. on Signal Processing* **48** (5): 1329–1337.

doi:10.1109/78.839980 (<http://dx.doi.org/10.1109%2F78.839980>) .

- 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 Trans. Circ. Syst. I* **51** (11): 2245–2254. doi:10.1109/TCSI.2004.836850 (<http://dx.doi.org/10.1109%2FTCSI.2004.836850>) .
- Shamgar Gurevich and Ronny Hadani (2009). "On the diagonalization of the discrete Fourier transform". *Applied and Computational Harmonic Analysis* **27** (1): 87–99. arXiv:0808.3281 (<http://arxiv.org/abs/0808.3281>) . doi:10.1016/j.acha.2008.11.003 (<http://dx.doi.org/10.1016%2Fj.acha.2008.11.003>) . preprint at.
- Shamgar Gurevich, Ronny Hadani, and 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 (<http://arxiv.org/abs/0808.1495>) . doi:10.1109/TIT.2008.926440 (<http://dx.doi.org/10.1109%2FTIT.2008.926440>) . preprint at.
- Juan G. Vargas-Rubio and Balu Santhanam (2005). "On the multiangle centered discrete fractional Fourier transform". *IEEE Sig. Proc. Lett.* **12** (4): 273–276. doi:10.1109/LSP.2005.843762 (<http://dx.doi.org/10.1109%2FLSP.2005.843762>) .
- J. Cooley, P. Lewis, and P. Welch (1969). "The finite Fourier transform". *IEEE Trans. Audio Electroacoustics* **17** (2): 77–85. doi:10.1109/TAU.1969.1162036 (<http://dx.doi.org/10.1109%2FTAU.1969.1162036>) .
- F.N. Kong (2008). "Analytic Expressions of Two Discrete Hermite-Gaussian Signals". *IEEE Trans. Circuits and Systems –II: Express Briefs.* **55** (1): 56–60. doi:10.1109/TCSII.2007.909865 (<http://dx.doi.org/10.1109%2FTCSII.2007.909865>) .

## External links

- Matlab tutorial on the Discrete Fourier Transformation ([http://www.nbtwiki.net/doku.php?id=tutorial:the\\_discrete\\_fourier\\_transformation\\_dft](http://www.nbtwiki.net/doku.php?id=tutorial:the_discrete_fourier_transformation_dft))
- Interactive flash tutorial on the DFT ([http://www.fourier-series.com/fourierseries2/DFT\\_tutorial.html](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>)
- Fast implementation of the DFT - coded in C and under General Public License (GPL) (<http://www.fftw.org>)
- The DFT “à Pied”: Mastering The Fourier Transform in One Day (<http://www.dspdimension.com/admin/dft-a-pied/>)
- Explained: The Discrete Fourier Transform (<http://web.mit.edu/newsoffice/2009/explained-fourier.html>)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Discrete\_Fourier\_transform&oldid=522376268"

Categories: Fourier analysis | Digital signal processing | Numerical analysis | Discrete transforms | Unitary operators

- 
- This page was last modified on 10 November 2012 at 20:08.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.