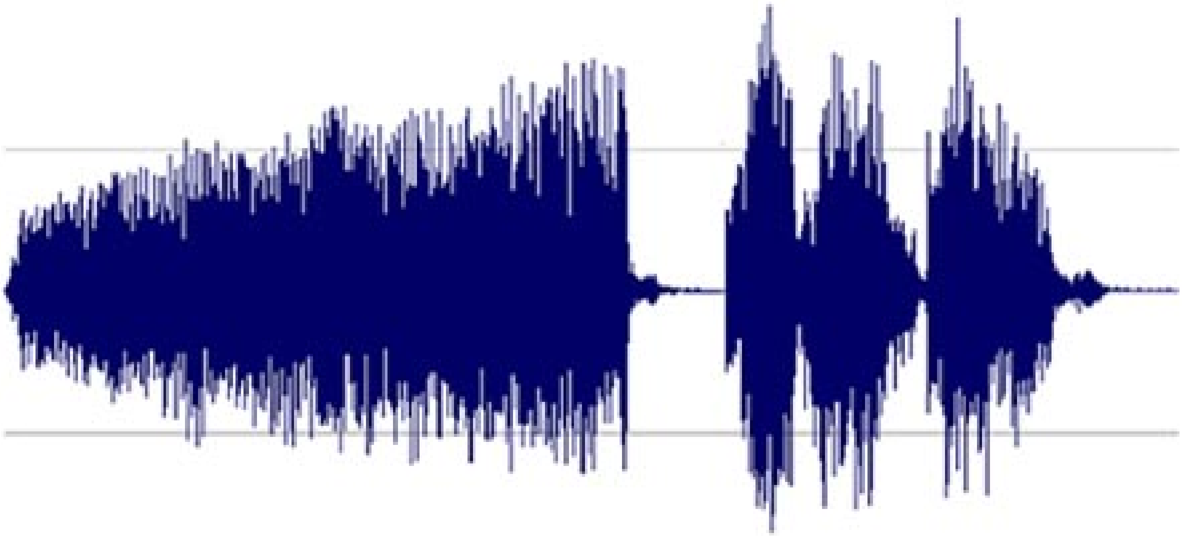
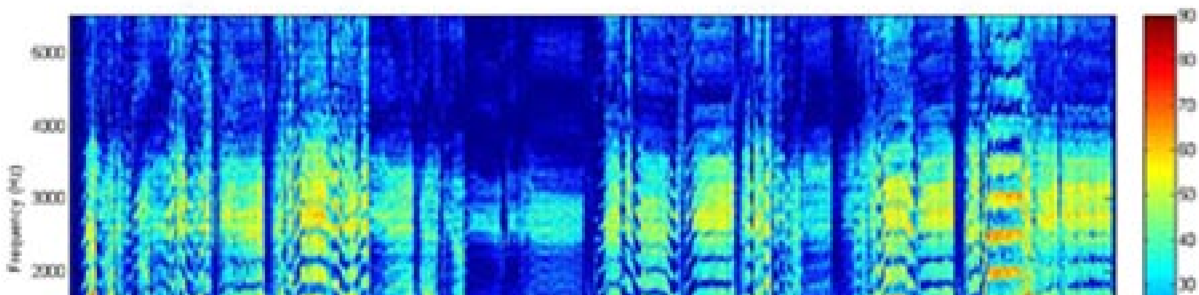
Transform Analysis in Mathematics and Signal Processing





Clemente Gamez

First Edition, 2012

ISBN 978-81-323-3686-0

Published by:
University Publications
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Bilinear Transform

The **bilinear transform** (also known as **Tustin's method**) is used in digital signal processing and discrete-time control theory to transform continuous-time system representations to discrete-time and vice versa.

The bilinear transform is a special case of a conformal mapping (namely, the Möbius transformation), often used to convert a transfer function $H_a(s)$ of a linear, time-invariant (LTI) filter in the continuous-time domain (often called an analog filter) to a transfer function $H_d(z)$ of a linear, shift-invariant filter in the discrete-time domain (often called a digital filter although there are analog filters constructed with switched capacitors that are discrete-time filters). It maps positions on the $j\omega$ axis, $Re[s] = 0$, in the s-plane to the unit circle, $|z| = 1$, in the z-plane. Other bilinear transforms can be used to warp the frequency response of any discrete-time linear system (for example to approximate the non-linear frequency resolution of the human auditory system) and are implementable in the discrete domain by replacing a system's unit delays $\left(z^{-1}\right)$ with first order all-pass filters.

The transform preserves stability and maps every point of the frequency response of the continuous-time filter, $H_a(j\omega_a)$ to a corresponding point in the frequency response of the discrete-time filter, $H_d(e^{j\omega_d T})$ although to a somewhat different frequency, as shown in the Frequency warping section below. This means that for every feature that one sees in the frequency response of the analog filter, there is a corresponding feature, with identical gain and phase shift, in the frequency response of the digital filter but, perhaps, at a somewhat different frequency. This is barely noticeable at low frequencies but is quite evident at frequencies close to the Nyquist frequency.

Discrete-time approximation

The bilinear transform is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane. When the Laplace transform is performed on a discrete-time signal (with each element of the discrete-time sequence attached to a correspondingly delayed unit impulse), the result is precisely the Z transform of the discrete-time sequence with the substitution of

$$\begin{aligned} z &= e^{sT} \\ &= \frac{e^{sT/2}}{e^{-sT/2}} \\ &\approx \frac{1 + sT/2}{1 - sT/2} \end{aligned}$$

where T is the sample time (the reciprocal of the sampling frequency) of the discrete-time filter. The above bilinear approximation can be solved for s or a similar approximation for $s = (1/T)\ln(z)$ can be performed.

The inverse of this mapping (and its first-order bilinear approximation) is

$$\begin{aligned} s &= \frac{1}{T}\ln(z) \\ &= \frac{2}{T}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 + \cdots\right] \\ &\approx \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \end{aligned}$$

The bilinear transform essentially uses this first order approximation and substitutes into the continuous-time transfer function, $H_a(s)$

$$s \leftarrow \frac{2}{T}\frac{z-1}{z+1}.$$

That is

$$H_d(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}} = H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right).$$

Stability and minimum-phase property preserved

A continuous-time causal filter is stable if the poles of its transfer function fall in the left half of the complex s-plane. A discrete-time causal filter is stable if the poles of its transfer function fall inside the unit circle in the complex z-plane. The bilinear transform maps the left half of the complex s-plane to the interior of the unit circle in the z-plane. Thus filters designed in the continuous-time domain that are stable are converted to filters in the discrete-time domain that preserve that stability.

Likewise, a continuous-time filter is minimum-phase if the zeros of its transfer function fall in the left half of the complex s-plane. A discrete-time filter is minimum-phase if the zeros of its transfer function fall inside the unit circle in the complex z-plane. Then the same mapping property assures that continuous-time filters that are minimum-phase are converted to discrete-time filters that preserve that property of being minimum-phase.

Example

As an example take a simple low-pass RC filter. This continuous-time filter has a transfer function

$$\begin{aligned} H_a(s) &= \frac{1/sC}{R+1/sC} \\ &= \frac{1}{1+RCs}. \end{aligned}$$

If we wish to implement this filter as a digital filter, we can apply the bilinear transform by substituting for *s* the formula above; after some reworking, we get the following filter representation:

$$\begin{aligned} H_d(z) &= H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right) \\ &= \frac{1}{1+RC\left(\frac{2}{T}\frac{z-1}{z+1}\right)} \\ &= \frac{1+z}{(1-2RC/T)+(1+2RC/T)z}. \\ &= \frac{1+z^{-1}}{(1+2RC/T)+(1-2RC/T)z^{-1}}. \end{aligned}$$

The coefficients of the denominator are the 'feed-backward' coefficients and the coefficients of the numerator are the 'feed-forward' coefficients used to implement a real-time digital filter.

Frequency warping

To determine the frequency response of a continuous-time filter, the transfer function $H_a(s)$ is evaluated at $s = j\omega$ which is on the $j\omega$ axis. Likewise, to determine the frequency response of a discrete-time filter, the transfer function $H_d(z)$ is evaluated at $z = e^{j\omega T}$ which is on the unit circle, $|z| = 1$. When the actual frequency of ω is input to the discrete-time filter designed by use of the bilinear transform, it is desired to know at what frequency, ω_a, for the continuous-time filter that this ω is mapped to.

$$
\begin{aligned}
H_d(z) &= H_a\left(\frac{2}{T}\frac{z-1}{z+1}\right) \\
H_d(e^{j\omega T}) &= H_a\left(\frac{2}{T}\frac{e^{j\omega T}-1}{e^{j\omega T}+1}\right) \\
&= H_a\left(\frac{2}{T}\cdot\frac{e^{j\omega T/2}\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)}{e^{j\omega T/2}\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)}\right) \\
&= H_a\left(\frac{2}{T}\cdot\frac{\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)}{\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\frac{\left(e^{j\omega T/2}-e^{-j\omega T/2}\right)/(2j)}{\left(e^{j\omega T/2}+e^{-j\omega T/2}\right)/2}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\frac{\sin(\omega T/2)}{\cos(\omega T/2)}\right) \\
&= H_a\left(j\frac{2}{T}\cdot\tan\left(\omega\frac{T}{2}\right)\right) \\
&= H_a\left(j\omega_a\right).
\end{aligned}
$$

This shows that every point on the unit circle in the discrete-time filter z-plane, $z = e^{j\omega T}$ is mapped to a point on the $j\omega$ axis on the continuous-time filter s-plane, $s = j\omega_a$. That is, the discrete-time to continuous-time frequency mapping of the bilinear transform is

$$\omega_a = \frac{2}{T}\tan\left(\omega\frac{T}{2}\right)$$

and the inverse mapping is

$$\omega = \frac{2}{T} \arctan \left(\omega_a \frac{T}{2} \right).$$

The discrete-time filter behaves at frequency ω the same way that the continuous-time filter behaves at frequency $(2/T) \tan(\omega T/2)$. Specifically, the gain and phase shift that the discrete-time filter has at frequency ω is the same gain and phase shift that the continuous-time filter has at frequency $(2/T) \tan(\omega T/2)$. This means that every feature, every "bump" that is visible in the frequency response of the continuous-time filter is also visible in the discrete-time filter, but at a different frequency. For low frequencies (that is, when $\omega \ll 2/T$ or $\omega_a \ll 2/T$), $\omega \approx \omega_a$.

One can see that the entire continuous frequency range

$$-\infty < \omega_a < +\infty$$

is mapped onto the fundamental frequency interval

$$-\frac{\pi}{T} < \omega < +\frac{\pi}{T}.$$

The continuous-time filter frequency $\omega_a = 0$ corresponds to the discrete-time filter frequency $\omega = 0$ and the continuous-time filter frequency $\omega_a = \pm\infty$ correspond to the discrete-time filter frequency $\omega = \pm\pi/T.$

One can also see that there is a nonlinear relationship between ω_a and ω. This effect of the bilinear transform is called ***frequency warping***. The continuous-time filter can be designed to compensate for this frequency warping by setting $\omega_a = \frac{2}{T} \tan \left(\omega \frac{T}{2} \right)$ for every frequency specification that the designer has control over (such as corner frequency or center frequency). This is called ***pre-warping*** the filter design.

The main advantage of the warping phenomenon is the absence of aliasing distortion of the frequency response characteristic, such as observed with Impulse invariance. It is necessary, however, to compensate for the frequency warping by pre-warping the given frequency specifications of the continuous-time system. These pre-warped specifications may then be used in the bilinear transform to obtain the desired discrete-time system

Chapter- 2

Fourier Transform

The **Fourier transform** is a mathematical operation that decomposes a signal into its constituent frequencies. Thus the Fourier transform of a musical chord is a mathematical representation of the amplitudes of the individual notes that make it up. The original signal depends on time, and therefore is called the *time domain* representation of the signal, whereas the Fourier transform depends on frequency and is called the *frequency domain* representation of the signal. The term Fourier transform refers both to the frequency domain representation of the signal and the process that transforms the signal to its frequency domain representation.

In mathematical terms, the Fourier transform transforms one complex-valued function of a real variable into another. In effect, the Fourier transform decomposes a function into oscillatory functions. The Fourier transform and its generalizations are the subject of Fourier analysis. In this specific case, both the time and frequency domains are unbounded linear continua. It is possible to define the Fourier transform of a function of several variables, which is important for instance in the physical study of wave motion and optics. It is also possible to generalize the Fourier transform on discrete structures such as finite groups. The efficient computation of such structures, by fast Fourier transform, is essential for high-speed computing.

Definition

There are several common conventions for defining the Fourier transform of an integrable function $f : \mathbf{R} \rightarrow \mathbf{C}$ (Kaiser 1994). Here we will use the definition:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\ e^{-2\pi i x \xi}\,dx,$$ for every real number ξ.

When the independent variable *x* represents *time* (with SI unit of seconds), the transform variable ξ represents frequency (in hertz). Under suitable conditions, *f* can be reconstructed from $\hat{f}$ by the **inverse transform**:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\ e^{2\pi i x \xi}\,d\xi,$$ for every real number x.

For other common conventions and notations, including using the angular frequency ω instead of the frequency ξ, see Other conventions and Other notations below. The Fourier transform on Euclidean space is treated separately, in which the variable x often represents position and ξ momentum.

Introduction

The motivation for the Fourier transform comes from the study of Fourier series. In the study of Fourier series, complicated functions are written as the sum of simple waves mathematically represented by sines and cosines. Due to the properties of sine and cosine it is possible to recover the amount of each wave in the sum by an integral. In many cases it is desirable to use Euler's formula, which states that $e^{2\pi i\theta} = \cos 2\pi\theta + i \sin 2\pi\theta$, to write Fourier series in terms of the basic waves $e^{2\pi i\theta}$. This has the advantage of simplifying many of the formulas involved and providing a formulation for Fourier series that more closely resembles the definition followed here. This passage from sines and cosines to complex exponentials makes it necessary for the Fourier coefficients to be complex valued. The usual interpretation of this complex number is that it gives both the amplitude (or size) of the wave present in the function and the phase (or the initial angle) of the wave. This passage also introduces the need for negative "frequencies". If θ were measured in seconds then the waves $e^{2\pi i\theta}$ and $e^{-2\pi i\theta}$ would both complete one cycle per second, but they represent different frequencies in the Fourier transform. Hence, frequency no longer measures the number of cycles per unit time, but is closely related.

There is a close connection between the definition of Fourier series and the Fourier transform for functions f which are zero outside of an interval. For such a function we can calculate its Fourier series on any interval that includes the interval where f is not identically zero. The Fourier transform is also defined for such a function. As we increase the length of the interval on which we calculate the Fourier series, then the Fourier series coefficients begin to look like the Fourier transform and the sum of the Fourier series of f begins to look like the inverse Fourier transform. To explain this more precisely, suppose that T is large enough so that the interval $[-T/2,T/2]$ contains the interval on which f is not identically zero. Then the n-th series coefficient c_n is given by**:**

$$c_n = \int_{-T/2}^{T/2} f(x)\ e^{-2\pi i (n/T) x}\,dx.$$

Comparing this to the definition of the Fourier transform it follows that $c_n = \hat{f}(n/T)$ since $f(x)$ is zero outside $[-T/2,T/2]$. Thus the Fourier coefficients are just the values of the Fourier transform sampled on a grid of width $1/T$. As T increases the Fourier coefficients more closely represent the Fourier transform of the function.

Under appropriate conditions the sum of the Fourier series of f will equal the function f. In other words f can be written**:**

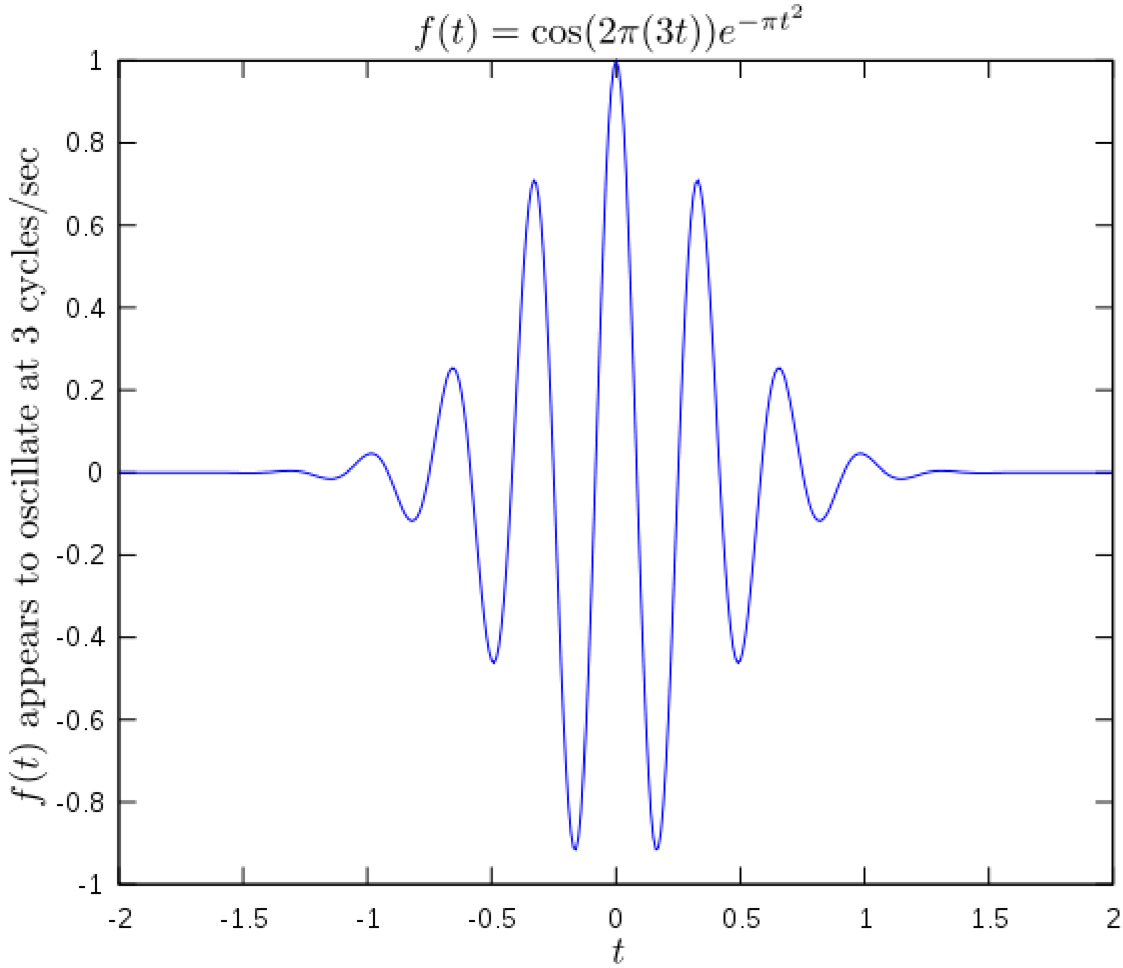
$$f(x) = \frac{1}{T}\sum_{n=-\infty}^{\infty} \hat{f}(n/T)\; e^{2\pi i(n/T)x} = \sum_{n=-\infty}^{\infty} \hat{f}(\xi_n)\; e^{2\pi i\xi_n x}\Delta\xi,$$

where the last sum is simply the first sum rewritten using the definitions $\xi_n = n/T$, and $\Delta\xi = (n + 1)/T - n/T = 1/T$.

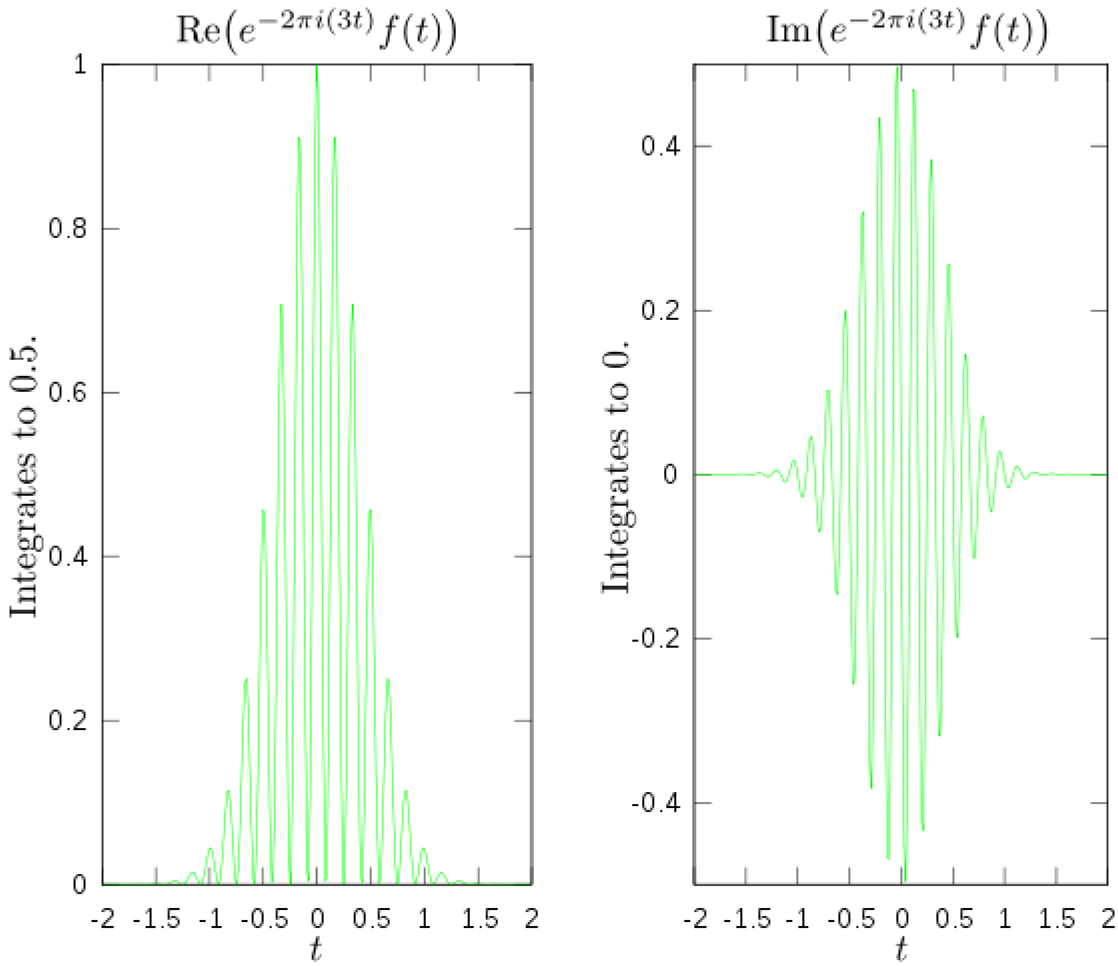
This second sum is a Riemann sum, and so by letting $T \to \infty$ it will converge to the integral for the inverse Fourier transform given in the definition section. Under suitable conditions this argument may be made precise (Stein & Shakarchi 2003).

In the study of Fourier series the numbers c_n could be thought of as the "amount" of the wave in the Fourier series of f. Similarly, as seen above, the Fourier transform can be thought of as a function that measures how much of each individual frequency is present in our function f, and we can recombine these waves by using an integral (or "continuous sum") to reproduce the original function.

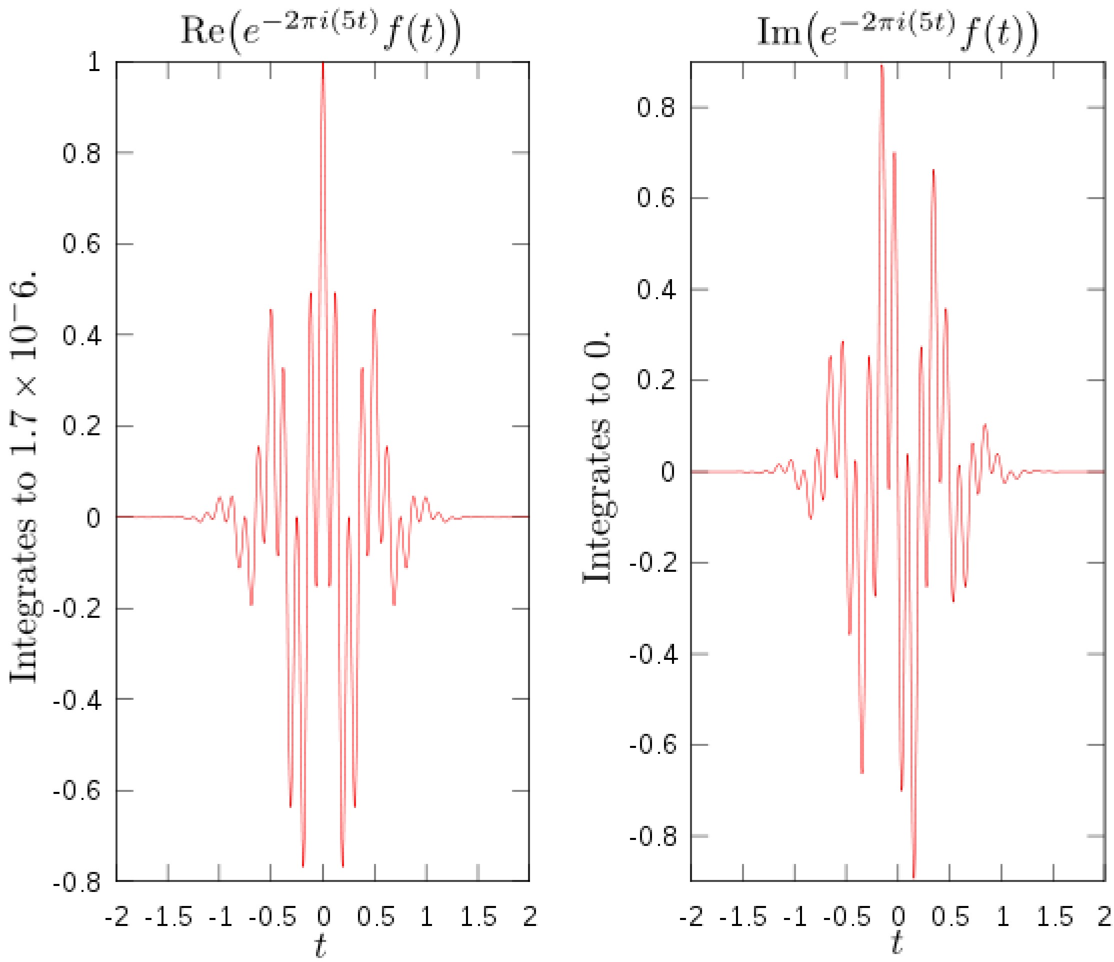
The following images provide a visual illustration of how the Fourier transform measures whether a frequency is present in a particular function. The function depicted $f(t) = \cos(6\pi t)e^{-\pi t^2}$ oscillates at 3 hertz (if t measures seconds) and tends quickly to 0. This function was specially chosen to have a real Fourier transform which can easily be plotted. The first image contains its graph. In order to calculate $\hat{f}(3)$ we must integrate $e^{-2\pi i(3t)}f(t)$. The second image shows the plot of the real and imaginary parts of this function. The real part of the integrand is almost always positive, this is because when $f(t)$ is negative, then the real part of $e^{-2\pi i(3t)}$ is negative as well. Because they oscillate at the same rate, when $f(t)$ is positive, so is the real part of $e^{-2\pi i(3t)}$. The result is that when you integrate the real part of the integrand you get a relatively large number (in this case 0.5). On the other hand, when you try to measure a frequency that is not present, as in the case when we look at $\hat{f}(5)$, the integrand oscillates enough so that the integral is very small. The general situation may be a bit more complicated than this, but this in spirit is how the Fourier transform measures how much of an individual frequency is present in a function $f(t)$.



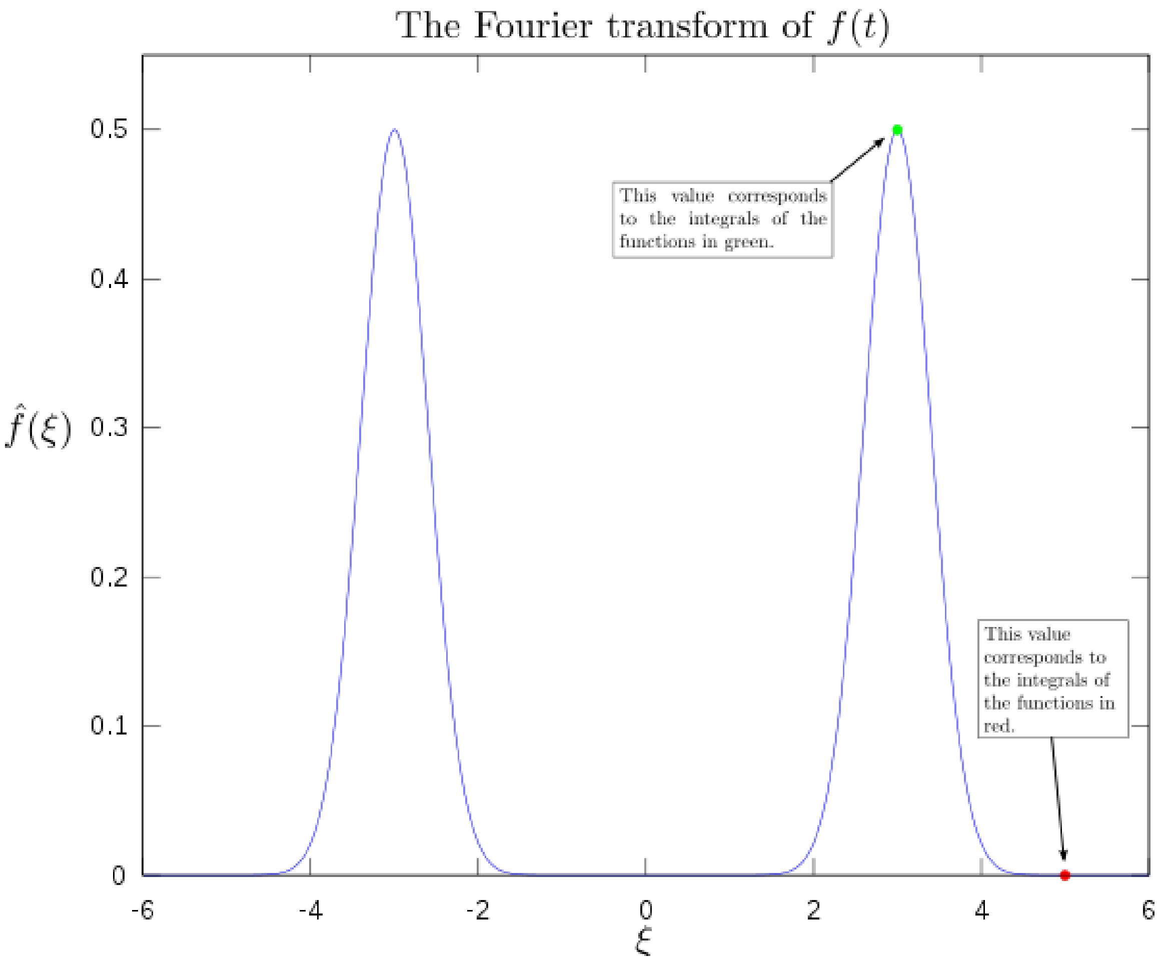
Original function showing oscillation 3 hertz.



Real and imaginary parts of integrand for Fourier transform at 3 hertz



Real and imaginary parts of integrand for Fourier transform at 5 hertz



Fourier transform with 3 and 5 hertz labeled.

Properties of the Fourier transform

An *integrable function* is a function f on the real line that is Lebesgue-measurable and satisfies

$$\int_{-\infty}^{\infty} |f(x)|\, dx < \infty.$$

Basic properties

Given integrable functions $f(x)$, $g(x)$, and $h(x)$ denote their Fourier transforms by $\hat{f}(\xi)$, $\hat{g}(\xi)$, and $\hat{h}(\xi)$ respectively. The Fourier transform has the following basic properties (Pinsky 2002).

Linearity

For any complex numbers a and b, if $h(x) = af(x) + bg(x)$, then $\hat{h}(\xi) = a \cdot \hat{f}(\xi) + b \cdot \hat{g}(\xi)$.

Translation

For any real number x_0, if $h(x) = f(x - x_0)$, then $\hat{h}(\xi) = e^{-2\pi i x_0 \xi} \hat{f}(\xi)$.

Modulation

For any real number ξ_0, if $h(x) = e^{2\pi i x \xi_0} f(x)$, then $\hat{h}(\xi) = \hat{f}(\xi - \xi_0)$.

Scaling

For a non-zero real number a, if $h(x) = f(ax)$, then $\hat{h}(\xi) = \frac{1}{|a|}\hat{f}\left(\frac{\xi}{a}\right)$. The case $a = -1$ leads to the *time-reversal* property, which states: if $h(x) = f(-x)$, then $\hat{h}(\xi) = \hat{f}(-\xi)$.

Conjugation

If $h(x) = \overline{f(x)}$, then $\hat{h}(\xi) = \overline{\hat{f}(-\xi)}$.

In particular, if f is real, then one has the *reality condition* $\hat{f}(-\xi) = \overline{\hat{f}(\xi)}$.

And if f is purely imaginary, then $\hat{f}(-\xi) = -\overline{\hat{f}(\xi)}$.

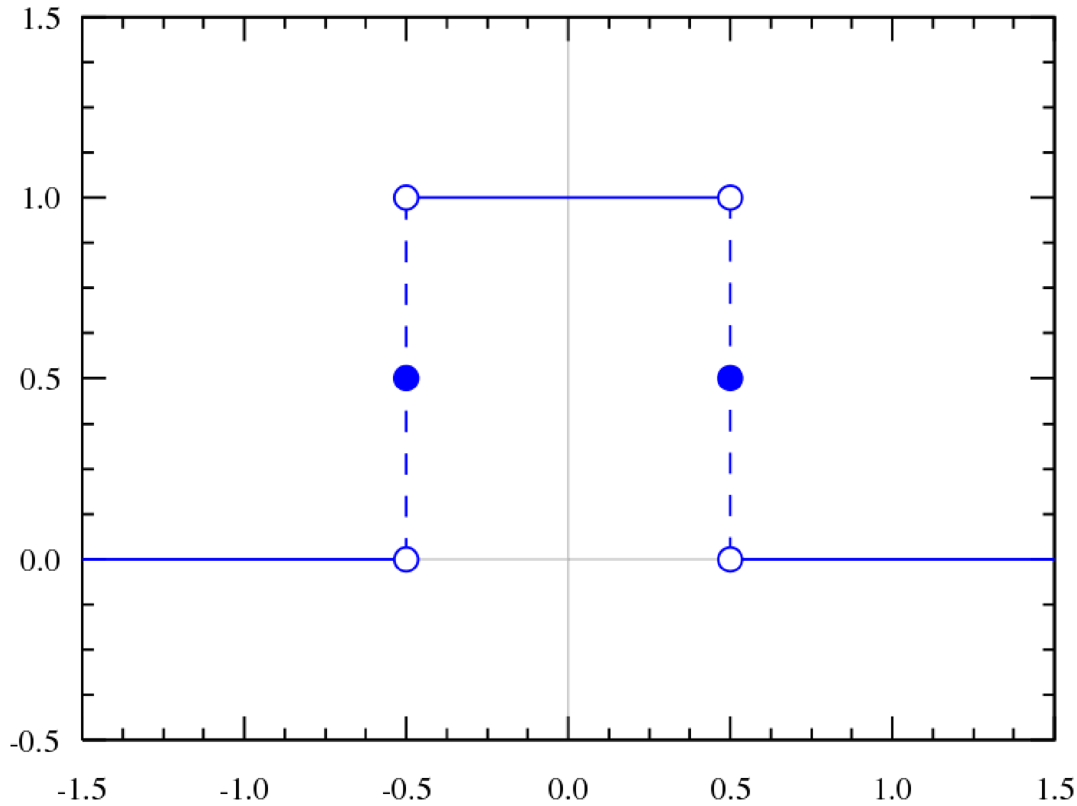
Duality

If $h(x) = \hat{f}(x)$ then $\hat{h}(\xi) = f(-\xi)$.

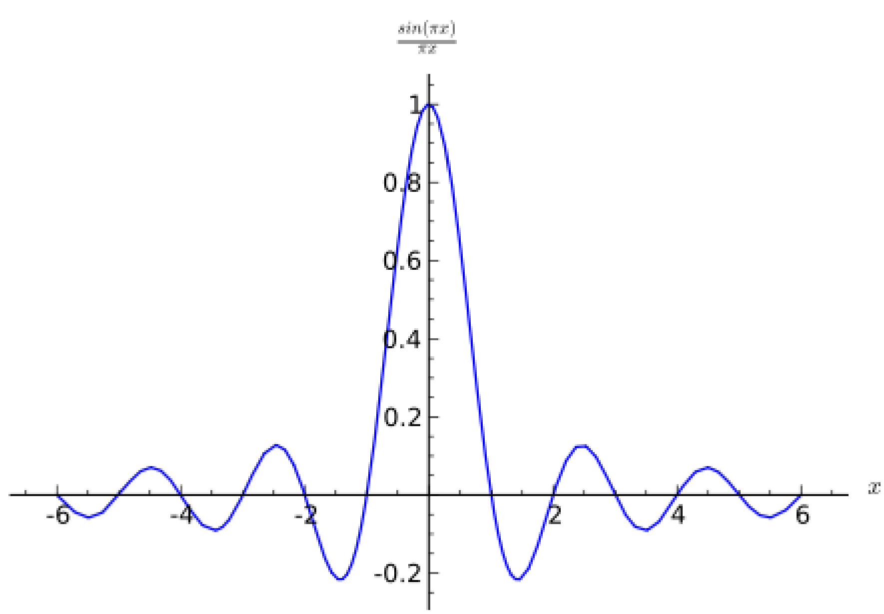
Convolution

If $h(x) = (f * g)(x)$, then $\hat{h}(\xi) = \hat{f}(\xi) \cdot \hat{g}(\xi)$.

Uniform continuity and the Riemann–Lebesgue lemma



The rectangular function is Lebesgue integrable.



The sinc function, which is the Fourier transform of the rectangular function, is bounded and continuous, but not Lebesgue integrable.

The Fourier transform of integrable functions have additional properties that do not always hold. The Fourier transforms of integrable functions *f* are uniformly continuous and $\|\hat{f}\|_\infty \leq \|f\|_1$ (Katznelson 1976). The Fourier transform of integrable functions also satisfy the *Riemann–Lebesgue lemma* which states that (Stein & Weiss 1971)

$$\hat{f}(\xi) \to 0 \text{ as } |\xi| \to \infty.$$

The Fourier transform $\hat{f}$ of an integrable function *f* is bounded and continuous, but need not be integrable – for example, the Fourier transform of the rectangular function, which is a step function (and hence integrable) is the sinc function, which is not Lebesgue integrable, though it does have an improper integral: one has an analog to the alternating harmonic series, which is a convergent sum but not absolutely convergent.

It is not possible in general to write the *inverse transform* as a Lebesgue integral. However, when both *f* and $\hat{f}$ are integrable, the following inverse equality holds true for almost every *x*:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2i\pi x\xi}\, d\xi.$$

Almost everywhere, *f* is equal to the continuous function given by the right-hand side. If *f* is given as continuous function on the line, then equality holds for every *x*.

A consequence of the preceding result is that the Fourier transform is injective on $L^1(\mathbf{R})$.

The Plancherel theorem and Parseval's theorem

Let *f*(*x*) and *g*(*x*) be integrable, and let $\hat{f}(\xi)$ and $\hat{g}(\xi)$ be their Fourier transforms. If *f*(*x*) and *g*(*x*) are also square-integrable, then we have Parseval's theorem (Rudin 1987, p. 187)**:**

$$\int_{-\infty}^{\infty} f(x)\overline{g(x)}\, dx = \int_{-\infty}^{\infty} \hat{f}(\xi)\overline{\hat{g}(\xi)}\, d\xi,$$

where the bar denotes complex conjugation.

The Plancherel theorem, which is equivalent to Parseval's theorem, states (Rudin 1987, p. 186)**:**

$$\int_{-\infty}^{\infty} |f(x)|^2 \, dx = \int_{-\infty}^{\infty} \left|\hat{f}(\xi)\right|^2 \, d\xi.$$

The Plancherel theorem makes it possible to define the Fourier transform for functions in $L^2(\mathbf{R})$, as described in Generalizations below. The Plancherel theorem has the interpretation in the sciences that the Fourier transform preserves the energy of the original quantity. It should be noted that depending on the author either of these theorems might be referred to as the Plancherel theorem or as Parseval's theorem.

Poisson summation formula

The Poisson summation formula provides a link between the study of Fourier transforms and Fourier Series. Given an integrable function f we can consider the periodic summation of f given by:

$$\bar{f}(x) = \sum_{k\in\mathbb{Z}} f(x+k),$$

where the summation is taken over the set of all integers k. The Poisson summation formula relates the Fourier series of $\bar{f}$ to the Fourier transform of f. Specifically it states that the Fourier series of $\bar{f}$ is given by:

$$\bar{f}(x) \sim \sum_{k\in\mathbb{Z}} \hat{f}(k) e^{2\pi i k x}.$$

Convolution theorem

The Fourier transform translates between convolution and multiplication of functions. If $f(x)$ and $g(x)$ are integrable functions with Fourier transforms $\hat{f}(\xi)$ and $\hat{g}(\xi)$ respectively, then the Fourier transform of the convolution is given by the product of the Fourier transforms $\hat{f}(\xi)$ and $\hat{g}(\xi)$ (under other conventions for the definition of the Fourier transform a constant factor may appear).

This means that if**:**

$$h(x) = (f * g)(x) = \int_{-\infty}^{\infty} f(y) g(x-y) \, dy,$$

where ∗ denotes the convolution operation, then**:**

$$\hat{h}(\xi) = \hat{f}(\xi) \cdot \hat{g}(\xi).$$

In linear time invariant (LTI) system theory, it is common to interpret $g(x)$ as the impulse response of an LTI system with input $f(x)$ and output $h(x)$, since substituting the unit impulse for $f(x)$ yields $h(x) = g(x)$. In this case, $\hat{g}(\xi)$ represents the frequency response of the system.

Conversely, if $f(x)$ can be decomposed as the product of two square integrable functions $p(x)$ and $q(x)$, then the Fourier transform of $f(x)$ is given by the convolution of the respective Fourier transforms $\hat{p}(\xi)$and $\hat{q}(\xi)$.

Cross-correlation theorem

In an analogous manner, it can be shown that if $h(x)$ is the cross-correlation of $f(x)$ and $g(x)$:

$$h(x) = (f \star g)(x) = \int_{-\infty}^{\infty} \overline{f(y)}\, g(x+y)\, dy$$

then the Fourier transform of $h(x)$ is:

$$\hat{h}(\xi) = \overline{\hat{f}(\xi)}\, \hat{g}(\xi).$$

As a special case, the autocorrelation of function $f(x)$ is:

$$h(x) = (f \star f)(x) = \int_{-\infty}^{\infty} \overline{f(y)} f(x+y)\, dy$$

for which

$$\hat{h}(\xi) = \overline{\hat{f}(\xi)}\, \hat{f}(\xi) = |\hat{f}(\xi)|^2.$$

Eigenfunctions

One important choice of an orthonormal basis for $L^2(\mathbf{R})$ is given by the Hermite functions

$$\psi_n(x) = \frac{2^{1/4}}{\sqrt{n!}}\, e^{-\pi x^2} H_n(2x\sqrt{\pi}),$$

where $H_n(x)$ are the "probabilist's" Hermite polynomials, defined by $H_n(x) = (-1)^n \exp(x^2/2)\, D^n \exp(-x^2/2)$. Under this convention for the Fourier transform, we have that

$$\hat{\psi}_n(\xi) = (-i)^n \psi_n(\xi).$$

In other words, the Hermite functions form a complete orthonormal system of eigenfunctions for the Fourier transform on $L^2(\mathbf{R})$ (Pinsky 2002). However, this choice of eigenfunctions is not unique. There are only four different eigenvalues of the Fourier transform (±1 and $\pm i$) and any linear combination of eigenfunctions with the same eigenvalue gives another eigenfunction. As a consequence of this, it is possible to decompose $L^2(\mathbf{R})$ as a direct sum of four spaces H_0, H_1, H_2, and H_3 where the Fourier transform acts on H_k simply by multiplication by i^k. This approach to define the Fourier transform is due to N. Wiener (Duoandikoetxea 2001). The choice of Hermite functions is convenient because they are exponentially localized in both frequency and time domains, and thus give rise to the fractional Fourier transform used in time-frequency analysis (Boashash 2003).

Fourier transform on Euclidean space

The Fourier transform can be in any arbitrary number of dimensions n. As with the one-dimensional case there are many conventions, for an integrable function $f(x)$ this article takes the definition**:**

$$\hat{f}(\xi) = \mathcal{F}(f)(\xi) = \int_{\mathbb{R}^n} f(x) e^{-2\pi i x\cdot\xi}\, dx$$

where x and ξ are n-dimensional vectors, and $x \cdot \xi$ is the dot product of the vectors. The dot product is sometimes written as $\langle x, \xi \rangle$.

All of the basic properties listed above hold for the n-dimensional Fourier transform, as do Plancherel's and Parseval's theorem. When the function is integrable, the Fourier transform is still uniformly continuous and the Riemann–Lebesgue lemma holds. (Stein & Weiss 1971)

Uncertainty principle

Generally speaking, the more concentrated $f(x)$ is, the more spread out its Fourier transform $\hat{f}(\xi)$ must be. In particular, the scaling property of the Fourier transform may be seen as saying: if we "squeeze" a function in x, its Fourier transform "stretches out" in ξ. It is not possible to arbitrarily concentrate both a function and its Fourier transform.

The trade-off between the compaction of a function and its Fourier transform can be formalized in the form of an **Uncertainty Principle** by viewing a function and its Fourier transform as conjugate variables with respect to the symplectic form on the time–frequency domain: from the point of view of the linear canonical transformation, the Fourier transform is rotation by 90° in the time–frequency domain, and preserves the symplectic form.

Suppose $f(x)$ is an integrable and square-integrable function. Without loss of generality, assume that $f(x)$ is normalized:

$$\int_{-\infty}^{\infty} |f(x)|^2 \, dx = 1.$$

It follows from the Plancherel theorem that $\hat{f}(\xi)$ is also normalized.

The spread around $x = 0$ may be measured by the *dispersion about zero* (Pinsky 2002) defined by

$$D_0(f) = \int_{-\infty}^{\infty} x^2 |f(x)|^2 \, dx.$$

In probability terms, this is the second moment of $|f(x)|^2$ about zero.

The Uncertainty principle states that, if $f(x)$ is absolutely continuous and the functions $x \cdot f(x)$ and $f'(x)$ are square integrable, then

$$D_0(f) D_0(\hat{f}) \geq \frac{1}{16\pi^2}$$ (Pinsky 2002).

The equality is attained only in the case $f(x) = C_1 \, e^{-\pi x^2/\sigma^2}$ (hence $\hat{f}(\xi) = \sigma C_1 \, e^{-\pi\sigma^2\xi^2}$) where $\sigma > 0$ is arbitrary and C_1 is such that f is L^2–normalized (Pinsky 2002). In other words, where f is a (normalized) Gaussian function, centered at zero.

In fact, this inequality implies that:

$$\left(\int_{-\infty}^{\infty} (x - x_0)^2 |f(x)|^2 \, dx \right) \left(\int_{-\infty}^{\infty} (\xi - \xi_0)^2 |\hat{f}(\xi)|^2 \, d\xi \right) \geq \frac{1}{16\pi^2}$$

for any x_0, ξ_0 in **R** (Stein & Shakarchi 2003).

In quantum mechanics, the momentum and position wave functions are Fourier transform pairs, to within a factor of Planck's constant. With this constant properly taken into account, the inequality above becomes the statement of the Heisenberg uncertainty principle (Stein & Shakarchi 2003).

Spherical harmonics

Let the set of homogeneous harmonic polynomials of degree k on $\mathbf{R}^n$ be denoted by $\mathbf{A}_k$. The set $\mathbf{A}_k$ consists of the solid spherical harmonics of degree k. The solid spherical harmonics play a similar role in higher dimensions to the Hermite polynomials in dimension one. Specifically, if $f(x) = e^{-\pi|x|2}P(x)$ for some $P(x)$ in $\mathbf{A}_k$, then $\hat{f}(\xi) = i^{-k} f(\xi)$. Let the set $\mathbf{H}_k$ be the closure in $L^2(\mathbf{R}^n)$ of linear combinations of functions of the form $f(|x|)P(x)$ where $P(x)$ is in $\mathbf{A}_k$. The space $L^2(\mathbf{R}^n)$ is then a direct sum of the spaces $\mathbf{H}_k$ and the Fourier transform maps each space $\mathbf{H}_k$ to itself and is possible to characterize the action of the Fourier transform on each space $\mathbf{H}_k$ (Stein & Weiss 1971). Let $f(x) = f_0(|x|)P(x)$ (with $P(x)$ in $\mathbf{A}_k$), then $\hat{f}(\xi) = F_0(|\xi|)P(\xi)$ where

$$F_0(r) = 2\pi i^{-k} r^{-(n+2k-2)/2} \int_0^\infty f_0(s) J_{(n+2k-2)/2}(2\pi rs) s^{(n+2k)/2}\, ds.$$

Here $J_{(n+2k-2)/2}$ denotes the Bessel function of the first kind with order $(n + 2k - 2)/2$. When $k = 0$ this gives a useful formula for the Fourier transform of a radial function (Grafakos 2004).

Restriction problems

In higher dimensions it becomes interesting to study *restriction problems* for the Fourier transform. The Fourier transform of an integrable function is continuous and the restriction of this function to any set is defined. But for a square-integrable function the Fourier transform could be a general *class* of square integrable functions. As such, the restriction of the Fourier transform of an $L^2(\mathbf{R}^n)$ function cannot be defined on sets of measure 0. It is still an active area of study to understand restriction problems in L^p for $1 < p < 2$. Surprisingly, it is possible in some cases to define the restriction of a Fourier transform to a set S, provided S has non-zero curvature. The case when S is the unit sphere in $\mathbf{R}^n$ is of particular interest. In this case the Tomas-Stein restriction theorem states that the restriction of the Fourier transform to the unit sphere in $\mathbf{R}^n$ is a bounded operator on L^p provided $1 \leq p \leq (2n + 2) \ / \ (n + 3)$.

One notable difference between the Fourier transform in 1 dimension versus higher dimensions concerns the partial sum operator. Consider an increasing collection of measurable sets E_R indexed by $R \in (0,\infty)$: such as balls of radius R centered at the origin, or cubes of side $2R$. For a given integrable function f, consider the function f_R defined by:

$$f_R(x) = \int_{E_R} \hat{f}(\xi) e^{2\pi i x\cdot\xi}\, d\xi, \quad x \in \mathbb{R}^n.$$

Suppose in addition that f is in $L^p(\mathbf{R}^n)$. For $n = 1$ and $1 < p < \infty$, if one takes $E_R = (-R, R)$, then f_R converges to f in L^p as R tends to infinity, by the boundedness of the Hilbert transform. Naively one may hope the same holds true for $n > 1$. In the case that E_R is taken to be a cube with side length R, then convergence still holds. Another natural candidate is the Euclidean ball $E_R = \{\xi : |\xi| < R\}$. In order for this partial sum operator to converge, it is necessary that the multiplier for the unit ball be bounded in $L^p(\mathbf{R}^n)$. For

$n \geq 2$ it is a celebrated theorem of Charles Fefferman that the multiplier for the unit ball is never bounded unless $p = 2$ (Duoandikoetxea 2001). In fact, when $p \neq 2$, this shows that not only may f_R fail to converge to f in L^p, but for some functions $f \in L^p(\mathbf{R}^n)$, f_R is not even an element of L^p.

Generalizations

Fourier transform on other function spaces

It is possible to extend the definition of the Fourier transform to other spaces of functions. Since compactly supported smooth functions are integrable and dense in $L^2(\mathbf{R})$, the Plancherel theorem allows us to extend the definition of the Fourier transform to general functions in $L^2(\mathbf{R})$ by continuity arguments. Further $\mathcal{F}: L^2(\mathbf{R}) \to L^2(\mathbf{R})$ is a unitary operator (Stein & Weiss 1971, Thm. 2.3). Many of the properties remain the same for the Fourier transform. The Hausdorff–Young inequality can be used to extend the definition of the Fourier transform to include functions in $L^p(\mathbf{R})$ for $1 \leq p \leq 2$. Unfortunately, further extensions become more technical. The Fourier transform of functions in L^p for the range $2 < p < \infty$ requires the study of distributions (Katznelson 1976). In fact, it can be shown that there are functions in L^p with $p>2$ so that the Fourier transform is not defined as a function (Stein & Weiss 1971).

Fourier–Stieltjes transform

The Fourier transform of a finite Borel measure μ on $\mathbf{R}^n$ is given by (Pinsky 2002):

$$\hat{\mu}(\xi) = \int_{\mathbb{R}^n} \mathrm{e}^{-2\pi i x\cdot\xi}\, d\mu.$$

This transform continues to enjoy many of the properties of the Fourier transform of integrable functions. One notable difference is that the Riemann–Lebesgue lemma fails for measures (Katznelson 1976). In the case that $d\mu = f(x)\, dx$, then the formula above reduces to the usual definition for the Fourier transform of f. In the case that μ is the probability distribution associated to a random variable X, the Fourier-Stieltjes transform is closely related to the characteristic function, but the typical conventions in probability theory take $e^{ix\cdot\xi}$ instead of $e^{-2\pi ix\cdot\xi}$ (Pinsky 2002). In the case when the distribution has a probability density function this definition reduces to the Fourier transform applied to the probability density function, again with a different choice of constants.

The Fourier transform may be used to give a characterization of continuous measures. Bochner's theorem characterizes which functions may arise as the Fourier–Stieltjes transform of a measure (Katznelson 1976).

Furthermore, the Dirac delta function is not a function but it is a finite Borel measure. Its Fourier transform is a constant function (whose specific value depends upon the form of the Fourier transform used).

Tempered distributions

The Fourier transform maps the space of Schwartz functions to itself, and gives a homeomorphism of the space to itself (Stein & Weiss 1971). Because of this it is possible to define the Fourier transform of tempered distributions. These include all the integrable functions mentioned above, as well as well-behaved functions of polynomial growth and distributions of compact support, and have the added advantage that the Fourier transform of any tempered distribution is again a tempered distribution.

The following two facts provide some motivation for the definition of the Fourier transform of a distribution. First let f and g be integrable functions, and let $\hat{f}$ and $\hat{g}$ be their Fourier transforms respectively. Then the Fourier transform obeys the following multiplication formula (Stein & Weiss 1971),

$$\int_{\mathbb{R}^n} \hat{f}(x)g(x)\,dx = \int_{\mathbb{R}^n} f(x)\hat{g}(x)\,dx.$$

Secondly, every integrable function f defines a distribution T_f by the relation

$$T_f(\varphi) = \int_{\mathbb{R}^n} f(x)\varphi(x)\,dx$$ for all Schwartz functions φ.

In fact, given a distribution T, we define the Fourier transform by the relation

$$\hat{T}(\varphi) = T(\hat{\varphi})$$ for all Schwartz functions φ.

It follows that

$$\hat{T}_f = T_{\hat{f}}.$$

Distributions can be differentiated and the above mentioned compatibility of the Fourier transform with differentiation and convolution remains true for tempered distributions.

Locally compact abelian groups

The Fourier transform may be generalized to any locally compact abelian group. A locally compact abelian group is an abelian group which is at the same time a locally compact Hausdorff topological space so that the group operations are continuous. If G is a locally compact abelian group, it has a translation invariant measure μ, called Haar measure. For a locally compact abelian group G it is possible to place a topology on the set of characters $\hat{G}$ so that $\hat{G}$ is also a locally compact abelian group. For a function f in $L^1(G)$ it is possible to define the Fourier transform by (Katznelson 1976):

$$\hat{f}(\xi) = \int_G \xi(x) f(x)\, d\mu \qquad \text{for any } \xi \in \hat{G}.$$

Locally compact Hausdorff space

The Fourier transform may be generalized to any locally compact Hausdorff space, which recovers the topology but loses the group structure.

Given a locally compact Hausdorff topological space X, the space $A=C_0(X)$ of continuous complex-valued functions on X which vanish at infinity is in a natural way a commutative C*-algebra, via pointwise addition, multiplication, complex conjugation, and with norm as the uniform norm. Conversely, the characters of this algebra A, denoted Φ_A, are naturally a topological space, and can be identified with evaluation at a point of x, and one has an isometric isomorphism $C_0(X) \to C_0(\Phi_A)$. In the case where $X=\mathbf{R}$ is the real line, this is exactly the Fourier transform.

Non-abelian groups

The Fourier transform can also be defined for functions on a non-abelian group, provided that the group is compact. Unlike the Fourier transform on an abelian group, which is scalar-valued, the Fourier transform on a non-abelian group is operator-valued (Hewitt & Ross 1971, Chapter 8). The Fourier transform on compact groups is a major tool in representation theory (Knapp 2001) and non-commutative harmonic analysis.

Let G be a compact Hausdorff topological group. Let Σ denote the collection of all isomorphism classes of finite-dimensional irreducible unitary representations, along with a definite choice of representation $U^{(\sigma)}$ on the Hilbert space H_σ of finite dimension d_σ for each $\sigma \in \Sigma$. If μ is a finite Borel measure on G, then the Fourier–Stieltjes transform of μ is the operator on H_σ defined by

$$\langle \hat{\mu}\xi, \eta \rangle_{H_\sigma} = \int_G \langle \overline{U}^{(\sigma)}_g \xi, \eta \rangle \, d\mu(g)$$

where $\overline{U}^{(\sigma)}$ is the complex-conjugate representation of $U^{(\sigma)}$ acting on H_σ. As in the abelian case, if μ is absolutely continuous with respect to the left-invariant probability measure λ on G, then it is represented as

$$d\mu = f d\lambda$$

for some $f \in L^1(\lambda)$. In this case, one identifies the Fourier transform of f with the Fourier–Stieltjes transform of μ.

The mapping $\mu \mapsto \hat{\mu}$ defines an isomorphism between the Banach space $M(G)$ of finite Borel measures and a closed subspace of the Banach space $\mathbf{C}_\infty(\Sigma)$ consisting of all sequences $E = (E_\sigma)$ indexed by Σ of (bounded) linear operators $E_\sigma : H_\sigma \to H_\sigma$ for which the norm

$$\|E\| = \sup_{\sigma \in \Sigma} \|E_\sigma\|$$

is finite. The "convolution theorem" asserts that, furthermore, this isomorphism of Banach spaces is in fact an isomorphism of C^* algebras into a subspace of $\mathbf{C}_\infty(\Sigma)$, in which $M(G)$ is equipped with the product given by convolution of measures and $\mathbf{C}_\infty(\Sigma)$ the product given by multiplication of operators in each index σ.

The Peter-Weyl theorem holds, and a version of the Fourier inversion formula (Plancherel's theorem) follows: if $f \in L^2(G)$, then

$$f(g) = \sum_{\sigma \in \Sigma} d_\sigma \operatorname{tr}(\hat{f}(\sigma) U_g^{(\sigma)})$$

where the summation is understood as convergent in the L^2 sense.

The generalization of the Fourier transform to the noncommutative situation has also in part contributed to the development of noncommutative geometry. In this context, a categorical generalization of the Fourier transform to noncommutative groups is Tannaka-Krein duality, which replaces the group of characters with the category of representations. However, this loses the connection with harmonic functions.

Alternatives

In signal processing terms, a function (of time) is a representation of a signal with perfect *time resolution,* but no frequency information, while the Fourier transform has perfect *frequency resolution,* but no time information: the magnitude of the Fourier transform at a point is how much frequency content there is, but location is only given by phase (argument of the Fourier transform at a point), and standing waves are not localized in time – a sine wave continues out to infinity, without decaying. This limits the usefulness of the Fourier transform for analyzing signals that are localized in time, notably transients, or any signal of finite extent.

As alternatives to the Fourier transform, in time-frequency analysis, one uses time-frequency transforms or time-frequency distributions to represent signals in a form that has some time information and some frequency information – by the uncertainty principle, there is a trade-off between these. These can be generalizations of the Fourier transform, such as the short-time Fourier transform or fractional Fourier transform, or can use different functions to represent signals, as in wavelet transforms and chirplet transforms, with the wavelet analog of the (continuous) Fourier transform being the continuous wavelet transform. (Boashash 2003). For a variable time and frequency resolution, the De Groot Fourier Transform can be considered.

Applications

Analysis of differential equations

Fourier transforms and the closely related Laplace transforms are widely used in solving differential equations. The Fourier transform is compatible with differentiation in the following sense: if $f(x)$ is a differentiable function with Fourier transform $\hat{f}(\xi)$, then the Fourier transform of its derivative is given by $2\pi i\xi\hat{f}(\xi)$. This can be used to transform differential equations into algebraic equations. Note that this technique only applies to problems whose domain is the whole set of real numbers. By extending the Fourier transform to functions of several variables partial differential equations with domain $\mathbf{R}^n$ can also be translated into algebraic equations.

Fourier transform spectroscopy

The Fourier transform is also used in nuclear magnetic resonance (NMR) and in other kinds of spectroscopy, e.g. infrared (FTIR). In NMR an exponentially-shaped free induction decay (FID) signal is acquired in the time domain and Fourier-transformed to a Lorentzian line-shape in the frequency domain. The Fourier transform is also used in magnetic resonance imaging (MRI) and mass spectrometry.

Domain and range of the Fourier transform

It is often desirable to have the most general domain for the Fourier transform as possible. The definition of Fourier transform as an integral naturally restricts the domain to the space of integrable functions. Unfortunately, there is no simple characterizations of which functions are Fourier transforms of integrable functions (Stein & Weiss 1971). It is possible to extend the domain of the Fourier transform in various ways, as discussed in generalizations above. The following list details some of the more common domains and ranges on which the Fourier transform is defined.

- The space of Schwartz functions is closed under the Fourier transform. Schwartz functions are rapidly decaying functions and do not include all functions which are relevant for the Fourier transform. More details may be found in (Stein & Weiss 1971).

- The space L^p maps into the space L^q, where $1/p + 1/q = 1$ and $1 \le p \le 2$ (Hausdorff–Young inequality).

- In particular, the space L^2 is closed under the Fourier transform, but here the Fourier transform is no longer defined by integration.

- The space L^1 of Lebesgue integrable functions maps into C_0, the space of continuous functions that tend to zero at infinity – not just into the space L^∞ of bounded functions (the Riemann–Lebesgue lemma).

- The set of tempered distributions is closed under the Fourier transform. Tempered distributions are also a form of generalization of functions. It is in this generality that one can define the Fourier transform of objects like the Dirac comb.

Other notations

Other common notations for $\hat{f}(\xi)$ are these:

$$\tilde{f}(\xi),\ \tilde{f}(\omega),\ F(\xi),\ \mathcal{F}(f)(\xi),\ (\mathcal{F}f)(\xi),\ \mathcal{F}(f),\ \mathcal{F}(\omega),\ \mathcal{F}(j\omega),\ \mathcal{F}\{f\},\ \mathcal{F}(f(t))$$

Though less commonly other notations are used. Denoting the Fourier transform by a capital letter corresponding to the letter of function being transformed (such as $f(x)$ and $F(\xi)$) is especially common in the sciences and engineering. In electronics, the omega (ω) is often used instead of ξ due to its interpretation as angular frequency, sometimes it is written as $F(j\omega)$, where j is the imaginary unit, to indicate its relationship with the Laplace transform, and sometimes it is written informally as $F(2\pi f)$ in order to use ordinary frequency.

The interpretation of the complex function $\hat{f}(\xi)$ may be aided by expressing it in polar coordinate form**:** $\hat{f}(\xi) = A(\xi)e^{i\varphi(\xi)}$ in terms of the two real functions $A(\xi)$ and $\varphi(\xi)$ where**:**

$$A(\xi) = |\hat{f}(\xi)|,$$

is the amplitude and

$$\varphi(\xi) = \arg\left(\hat{f}(\xi)\right),$$

is the phase.

Then the inverse transform can be written**:**

$$f(x) = \int_{-\infty}^{\infty} A(\xi)\ e^{i(2\pi\xi x + \varphi(\xi))}\, d\xi,$$

which is a recombination of all the **frequency components** of $f(x)$. Each component is a complex sinusoid of the form $e^{2\pi ix\xi}$ whose amplitude is $A(\xi)$ and whose initial phase angle (at $x = 0$) is $\varphi(\xi)$.

The Fourier transform may be thought of as a mapping on function spaces. This mapping is here denoted $\mathcal{F}$ and $\mathcal{F}(f)$ is used to denote the Fourier transform of the function *f*. This mapping is linear, which means that $\mathcal{F}$ can also be seen as a linear transformation on the function space and implies that the standard notation in linear algebra of applying a linear transformation to a vector (here the function *f*) can be used to write $\mathcal{F}f$ instead of $\mathcal{F}(f)$. Since the result of applying the Fourier transform is again a function, we can be interested in the value of this function evaluated at the value ξ for its variable, and this is denoted either as $\mathcal{F}(f)(\xi)$ or as $(\mathcal{F}f)(\xi)$. Notice that in the former case, it is implicitly understood that $\mathcal{F}$ is applied first to *f* and then the resulting function is evaluated at ξ, not the other way around.

In mathematics and various applied sciences it is often necessary to distinguish between a function *f* and the value of *f* when its variable equals *x*, denoted $f(x)$. This means that a notation like $\mathcal{F}(f(x))$ formally can be interpreted as the Fourier transform of the values of *f* at *x*. Despite this flaw, the previous notation appears frequently, often when a particular function or a function of a particular variable is to be transformed. For example, $\mathcal{F}(\mathrm{rect}(x)) = \mathrm{sinc}(\xi)$ is sometimes used to express that the Fourier transform of a rectangular function is a sinc function, or $\mathcal{F}(f(x+x_0)) = \mathcal{F}(f(x))e^{2\pi i\xi x_0}$ is used to express the shift property of the Fourier transform. Notice, that the last example is only correct under the assumption that the transformed function is a function of *x*, not of x_0.

Other conventions

The Fourier transform can also be written in terms of angular frequency**:** $\omega = 2\pi\xi$ whose units are radians per second.

The substitution $\xi = \omega/(2\pi)$ into the formulas above produces this convention**:**

$$\hat{f}(\omega) = \int_{\mathbb{R}^n} f(x)e^{-i\omega\cdot x}\, dx.$$

Under this convention, the inverse transform becomes:

$$f(x) = \frac{1}{(2\pi)^n}\int_{\mathbb{R}^n} \hat{f}(\omega)e^{i\omega\cdot x}\, d\omega.$$

Unlike the convention followed here, when the Fourier transform is defined this way, it is no longer a unitary transformation on $L^2(\mathbf{R}^n)$. There is also less symmetry between the formulas for the Fourier transform and its inverse.

Another convention is to split the factor of $(2\pi)^n$ evenly between the Fourier transform and its inverse, which leads to definitions**:**

$$\hat{f}(\omega) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x) e^{-i\omega\cdot x}\, dx$$
$$f(x) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} \hat{f}(\omega) e^{i\omega\cdot x}\, d\omega.$$

Under this convention, the Fourier transform is again a unitary transformation on $L^2(\mathbf{R}^n)$. It also restores the symmetry between the Fourier transform and its inverse.

Variations of all three conventions can be created by conjugating the complex-exponential kernel of both the forward and the reverse transform. The signs must be opposites. Other than that, the choice is (again) a matter of convention.

Summary of popular forms of the Fourier transform

ordinary frequency ξ (hertz)	**unitary**	$\hat{f}_1(\xi) \stackrel{\text{def}}{=} \int_{\mathbb{R}^n} f(x) e^{-2\pi i x\cdot\xi}\, dx = \hat{f}_2(2\pi\xi) = (2\pi)^{n/2} \hat{f}_3(2\pi\xi)$ $f(x) = \int_{\mathbb{R}^n} \hat{f}_1(\xi) e^{2\pi i x\cdot\xi}\, d\xi$
angular frequency ω (rad/s)	**non-unitary**	$\hat{f}_2(\omega) \stackrel{\text{def}}{=} \int_{\mathbb{R}^n} f(x) e^{-i\omega\cdot x}\, dx = \hat{f}_1\left(\frac{\omega}{2\pi}\right) = (2\pi)^{n/2}\ \hat{f}_3(\omega)$ $f(x) = \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \hat{f}_2(\omega) e^{i\omega\cdot x}\, d\omega$
	unitary	$\hat{f}_3(\omega) \stackrel{\text{def}}{=} \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} f(x)\, e^{-i\omega\cdot x}\, dx = \frac{1}{(2\pi)^{n/2}} \hat{f}_1\left(\frac{\omega}{2\pi}\right) = \frac{1}{(2\pi)^{n/2}} \hat{f}_2(\omega)$ $f(x) = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} \hat{f}_3(\omega) e^{i\omega\cdot x}\, d\omega$

The ordinary-frequency convention is the one most often found in the mathematics literature. In the physics literature, the two angular-frequency conventions are more commonly used.

As discussed above, the characteristic function of a random variable is the same as the Fourier–Stieltjes transform of its distribution measure, but in this context it is typical to take a different convention for the constants. Typically characteristic function is defined $E(e^{it\cdot X}) = \int e^{it\cdot x} d\mu_X(x)$. As in the case of the "non-unitary angular frequency" convention above, there is no factor of 2π appearing in either of the integral, or in the exponential. Unlike any of the conventions appearing above, this convention takes the opposite sign in the exponential.

Tables of important Fourier transforms

The following tables record some closed form Fourier transforms. For functions $f(x)$, $g(x)$ and $h(x)$ denote their Fourier transforms by $\hat{f}$, $\hat{g}$, and $\hat{h}$ respectively. Only the three most common conventions are included. It may be useful to notice that entry 105 gives a relationship between the Fourier transform of a function and the original function, which can be seen as relating the Fourier transform and its inverse.

Functional relationships

The Fourier transforms in this table may be found in (Erdélyi 1954) or the appendix of (Kammler 2000).

	Function	Fourier transform unitary, ordinary frequency	Fourier transform unitary, angular frequency	Fourier transform non-unitary, angular frequency	Remarks
	$f(x)$	$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi}\, dx$	$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x}\, dx$	$\hat{f}(\nu) = \int_{-\infty}^{\infty} f(x) e^{-i\nu x}\, dx$	Definition
101	$a \cdot f(x) + b \cdot g(x)$	$a \cdot \hat{f}(\xi) + b \cdot \hat{g}(\xi)$	$a \cdot \hat{f}(\omega) + b \cdot \hat{g}(\omega)$	$a \cdot \hat{f}(\nu) + b \cdot \hat{g}(\nu)$	Linearity
102	$f(x-a)$	$e^{-2\pi i a \xi} \hat{f}(\xi)$	$e^{-ia\omega} \hat{f}(\omega)$	$e^{-ia\nu} \hat{f}(\nu)$	Shift in time domain
103	$e^{2\pi i a x} f(x)$	$\hat{f}(\xi - a)$	$\hat{f}(\omega - 2\pi a)$	$\hat{f}(\nu - 2\pi a)$	Shift in frequency domain, dual of 102
104	$f(ax)$	$\frac{1}{\|a\|} \hat{f}\left(\frac{\xi}{a}\right)$	$\frac{1}{\|a\|} \hat{f}\left(\frac{\omega}{a}\right)$	$\frac{1}{\|a\|} \hat{f}\left(\frac{\nu}{a}\right)$	Scaling in the time domain. If $\|a\|$ is large, then $f(ax)$ is concentrated around 0 and $\frac{1}{\|a\|} \hat{f}\left(\frac{\omega}{a}\right)$ spreads out and flattens.
105	$\hat{f}(x)$	$f(-\xi)$	$f(-\omega)$	$2\pi f(-\nu)$	Duality. Here $\hat{f}$ needs to be calculated using the same method as Fourier transform column. Results from swapping "dummy" variables of x and ξ or ω or ν.
106	$\frac{d^n f(x)}{dx^n}$	$(2\pi i \xi)^n \hat{f}(\xi)$	$(i\omega)^n \hat{f}(\omega)$	$(i\nu)^n \hat{f}(\nu)$	
107	$x^n f(x)$	$\left(\frac{i}{2\pi}\right)^n \frac{d^n \hat{f}(\xi)}{d\xi^n}$	$i^n \frac{d^n \hat{f}(\omega)}{d\omega^n}$	$i^n \frac{d^n \hat{f}(\nu)}{d\nu^n}$	This is the dual of 106

108	$(f * g)(x)$	$\hat{f}(\xi)\hat{g}(\xi)$	$\sqrt{2\pi}\hat{f}(\omega)\hat{g}(\omega)$	$\hat{f}(\nu)\hat{g}(\nu)$	The notation $f * g$ denotes the convolution of f and g — this rule is the convolution theorem
109	$f(x)g(x)$	$(\hat{f} * \hat{g})(\xi)$	$\frac{(\hat{f} * \hat{g})(\omega)}{\sqrt{2\pi}}$	$\frac{1}{2\pi}(\hat{f} * \hat{g})(\nu)$	This is the dual of 108
110	For $f(x)$ a purely real	$\hat{f}(-\xi) = \overline{\hat{f}(\xi)}$	$\hat{f}(-\omega) = \overline{\hat{f}(\omega)}$	$\hat{f}(-\nu) = \overline{\hat{f}(\nu)}$	Hermitian symmetry. $\overline{z}$ indicates the complex conjugate.
111	For $f(x)$ a purely real even function	$\hat{f}(\omega)$, $\hat{f}(\xi)$ and $\hat{f}(\nu)$ are purely real even functions.			
112	For $f(x)$ a purely real odd function	$\hat{f}(\omega)$, $\hat{f}(\xi)$ and $\hat{f}(\nu)$ are purely imaginary odd functions.			

Square-integrable functions

The Fourier transforms in this table may be found in (Campbell & Foster 1948), (Erdélyi 1954), or the appendix of (Kammler 2000).

	Function	**Fourier transform unitary, ordinary frequency**	**Fourier transform unitary, angular frequency**	**Fourier transform non-unitary, angular frequency**	**Remarks**
	$f(x)$	$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}\,dx$	$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)e^{-i\omega x}\,dx$	$\hat{f}(\nu) = \int_{-\infty}^{\infty} f(x)e^{-i\nu x}\,dx$	
201	$\operatorname{rect}(ax)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{sinc}\left(\frac{\xi}{a}\right)$	$\frac{1}{\sqrt{2\pi a^2}}\cdot\operatorname{sinc}\left(\frac{\omega}{2\pi a}\right)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{sinc}\left(\frac{\nu}{2\pi a}\right)$	The rectangular pulse and the *normalized* sinc function, here defined as sinc(*x*) = sin(*πx*)/(*πx*)
202	$\operatorname{sinc}(ax)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{rect}\left(\frac{\xi}{a}\right)$	$\frac{1}{\sqrt{2\pi a^2}}\cdot\operatorname{rect}\left(\frac{\omega}{2\pi a}\right)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{rect}\left(\frac{\nu}{2\pi a}\right)$	Dual of rule 201. The rectangular function is an ideal low-pass filter, and the sinc function is the non-causal impulse response of such a filter.
203	$\operatorname{sinc}^2(ax)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{tri}\left(\frac{\xi}{a}\right)$	$\frac{1}{\sqrt{2\pi a^2}}\cdot\operatorname{tri}\left(\frac{\omega}{2\pi a}\right)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{tri}\left(\frac{\nu}{2\pi a}\right)$	The function tri(*x*) is the triangular function
204	$\operatorname{tri}(ax)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{sinc}^2\left(\frac{\xi}{a}\right)$	$\frac{1}{\sqrt{2\pi a^2}}\cdot\operatorname{sinc}^2\left(\frac{\omega}{2\pi a}\right)$	$\frac{1}{\lvert a\rvert}\cdot\operatorname{sinc}^2\left(\frac{\nu}{2\pi a}\right)$	Dual of rule 203.

205	$e^{-ax}u(x)$	$\frac{1}{a+2\pi i\xi}$	$\frac{1}{\sqrt{2\pi}(a+i\omega)}$	$\frac{1}{a+i\nu}$	The function $u(x)$ is the Heaviside unit step function and $a>0$.
206	$e^{-\alpha x^2}$	$\sqrt{\frac{\pi}{\alpha}} \cdot e^{-\frac{(\pi\xi)^2}{\alpha}}$	$\frac{1}{\sqrt{2\alpha}} \cdot e^{-\frac{\omega^2}{4\alpha}}$	$\sqrt{\frac{\pi}{\alpha}} \cdot e^{-\frac{\nu^2}{4\alpha}}$	This shows that, for the unitary Fourier transforms, the Gaussian function $\exp(-\alpha x^2)$ is its own Fourier transform for some choice of α. For this to be integrable we must have $\mathrm{Re}(\alpha)>0$.
207	$\mathrm{e}^{-a\|x\|}$	$\frac{2a}{a^2+4\pi^2\xi^2}$	$\sqrt{\frac{2}{\pi}} \cdot \frac{a}{a^2+\omega^2}$	$\frac{2a}{a^2+\nu^2}$	For *a>0*. That is, the Fourier transform of a decaying exponential function is a Lorentzian function.
208	$\frac{J_n(x)}{x}$	$\frac{2i}{n}(-i)^n \cdot U_{n-1}(2\pi\xi) \cdot \sqrt{1-4\pi^2\xi^2}\,\mathrm{rect}(\pi\xi)$	$\sqrt{\frac{2}{\pi}}\frac{i}{n}(-i)^n \cdot U_{n-1}(\omega) \cdot \sqrt{1-\omega^2}\,\mathrm{rect}\left(\frac{\omega}{2}\right)$	$\frac{2i}{n}(-i)^n \cdot U_{n-1}(\nu) \cdot \sqrt{1-\nu^2}\,\mathrm{rect}\left(\frac{\nu}{2}\right)$	The functions $J_n(x)$ are the *n*-th order Bessel functions of the first kind. The functions $U_n(x)$ are the Chebyshev polynomial of the second kind.
209	$\mathrm{sech}(ax)$	$\frac{\pi}{a}\mathrm{sech}\left(\frac{\pi^2}{a}\xi\right)$	$\frac{1}{a}\sqrt{\frac{\pi}{2}}\mathrm{sech}\left(\frac{\pi}{2a}\omega\right)$	$\frac{\pi}{a}\mathrm{sech}\left(\frac{\pi}{2a}\nu\right)$	Hyperbolic secant is its own Fourier transform

Two-dimensional functions

Functions (400 to 402)	**Fourier transform unitary, ordinary frequency**	**Fourier transform unitary, angular frequency**	**Fourier transform non-unitary, angular frequency**
$f(x,y)$	$\hat{f}(\xi_x,\xi_y) = \iint f(x,y)e^{-2\pi i(\xi_x x+\xi_y y)}\,dx\,dy$	$\hat{f}(\omega_x,\omega_y) = \frac{1}{2\pi}\iint f(x,y)e^{-i(\omega_x x+\omega_y y)}\,dx\,dy$	$\hat{f}(\nu_x,\nu_y) = \iint f(x,y)e^{-i(\nu_x x+\nu_y y)}\,dx\,dy$
$e^{-\pi\left(a^2x^2+b^2y^2\right)}$	$\frac{1}{\|ab\|}e^{-\pi\left(\xi_x^2/a^2+\xi_y^2/b^2\right)}$	$\frac{1}{2\pi\cdot\|ab\|}e^{\frac{-\left(\omega_x^2/a^2+\omega_y^2/b^2\right)}{4\pi}}$	$\frac{1}{\|ab\|}e^{\frac{-\left(\nu_x^2/a^2+\nu_y^2/b^2\right)}{4\pi}}$
$\mathrm{circ}(\sqrt{x^2+y^2})$	$\frac{J_1\left(2\pi\sqrt{\xi_x^2+\xi_y^2}\right)}{\sqrt{\xi_x^2+\xi_y^2}}$	$\frac{J_1\left(\sqrt{\omega_x^2+\omega_y^2}\right)}{\sqrt{\omega_x^2+\omega_y^2}}$	$\frac{2\pi J_1\left(\sqrt{\nu_x^2+\nu_y^2}\right)}{\sqrt{\nu_x^2+\nu_y^2}}$

Remarks

To 400: The variables ξ_x, ξ_y, ω_x, ω_y, ν_x and ν_y are real numbers. The integrals are taken over the entire plane.

To 401: Both functions are Gaussians, which may not have unit volume.

To 402: The function is defined by circ(r)=1 $0 \le r \le 1$, and is 0 otherwise. This is the Airy distribution, and is expressed using J_1 (the order 1 Bessel function of the first kind). (Stein & Weiss 1971, Thm. IV.3.3)

Formulas for general *n*-dimensional functions

	Function	Fourier transform unitary, ordinary frequency	Fourier transform unitary, angular frequency	Fourier transform non-unitary, angular frequency
500	$f(x)$	$\hat{f}(\xi) = \int_{\mathbb{R}^n} f(x) e^{-2\pi i x\cdot\xi}\, d^n x$	$\hat{f}(\omega) = \frac{1}{(2\pi)^{(n/2)}} \int_{\mathbb{R}^n} f(x) e^{-i\omega\cdot x}\, d^n x$	$\hat{f}(\nu) = \int_{\mathbb{R}^n} f(x) e^{-ix\cdot\nu}\, d^n x$
501	$\chi_{[0,1]}(\lvert x\rvert)(1-\lvert x\rvert^2)^\delta$	$\pi^{-\delta}\Gamma(\delta+1)\lvert\xi\rvert^{-(n/2)-\delta} \cdot J_{n/2+\delta}(2\pi\lvert\xi\rvert)$	$2^{-\delta}\Gamma(\delta+1)\,\lvert\omega\rvert^{-(n/2)-\delta} \cdot J_{n/2+\delta}(\lvert\omega\rvert)$	$\pi^{-\delta}\Gamma(\delta+1)\left\lvert\frac{\nu}{2\pi}\right\rvert^{-(n/2)-\delta} \cdot J_{n/2+\delta}(\lvert\nu\rvert)$
502	$\lvert x\rvert^{-\alpha}, \quad 0 < \mathrm{Re}\alpha < n.$	$c_\alpha \lvert\xi\rvert^{-(n-\alpha)}$		

Remarks

To 501: The function $\chi_{[0,1]}$ is the indicator function of the interval [0,1]. The function $\Gamma(x)$ is the gamma function. The function $J_{n/2+\delta}$ is a Bessel function of the first kind, with order $n/2+\delta$. Taking $n = 2$ and $\delta = 0$ produces 402. (Stein & Weiss 1971, Thm. 4.13)

To 502. The formula also holds for all $\alpha \neq -n, -n-1, \ldots$ by analytic continuation, but then the function and its Fourier transforms need to be understood as suitably regularized tempered distributions.

Chapter- 3

Discrete Cosine Transform

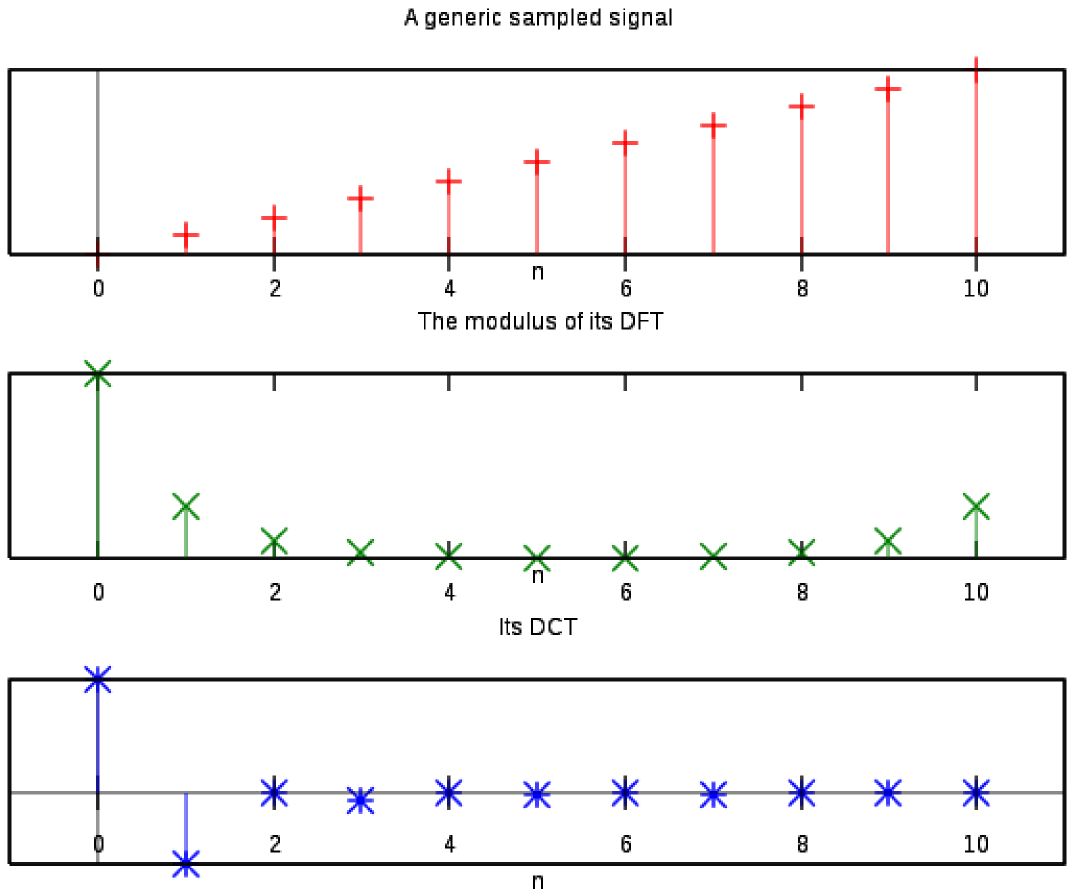
A **discrete cosine transform** (**DCT**) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient (as explained below, fewer are needed to approximate a typical signal), whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of *overlapping* data.

Applications

The DCT, and in particular the DCT-II, is often used in signal and image processing, especially for lossy data compression, because it has a strong "energy compaction" property (Rao and Yip, 1990): most of the signal information tends to be concentrated in a few low-frequency components of the DCT, approaching the Karhunen-Loève transform (which is optimal in the decorrelation sense) for signals based on certain limits of Markov processes. As explained below, this stems from the boundary conditions implicit in the cosine functions.



DCT-II (bottom) compared to the DFT (middle) of an input signal (top).

A related transform, the *modified* discrete cosine transform, or MDCT (based on the DCT-IV), is used in AAC, Vorbis, WMA, and MP3 audio compression.

DCTs are also widely employed in solving partial differential equations by spectral methods, where the different variants of the DCT correspond to slightly different even/odd boundary conditions at the two ends of the array.

DCTs are also closely related to Chebyshev polynomials, and fast DCT algorithms (below) are used in Chebyshev approximation of arbitrary functions by series of Chebyshev polynomials, for example in Clenshaw–Curtis quadrature.

JPEG

The DCT is used in JPEG image compression, MJPEG, MPEG, DV, and Theora video compression. There, the two-dimensional DCT-II of $N \times N$ blocks are computed and the results are quantized and entropy coded. In this case, *N* is typically 8 and the DCT-II formula is applied to each row and column of the block. The result is an 8 × 8 transform coefficient array in which the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies.

Informal overview

Like any Fourier-related transform, discrete cosine transforms (DCTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DCT operates on a function at a finite number of discrete data points. The obvious distinction between a DCT and a DFT is that the former uses only cosine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DCT implies different boundary conditions than the DFT or other related transforms.

The Fourier-related transforms that operate on a function over a finite domain, such as the DFT or DCT or a Fourier series, can be thought of as implicitly defining an *extension* of that function outside the domain. That is, once you write a function $f(x)$ as a sum of sinusoids, you can evaluate that sum at any x, even for x where the original $f(x)$ was not specified. The DFT, like the Fourier series, implies a periodic extension of the original function. A DCT, like a cosine transform, implies an even extension of the original function.

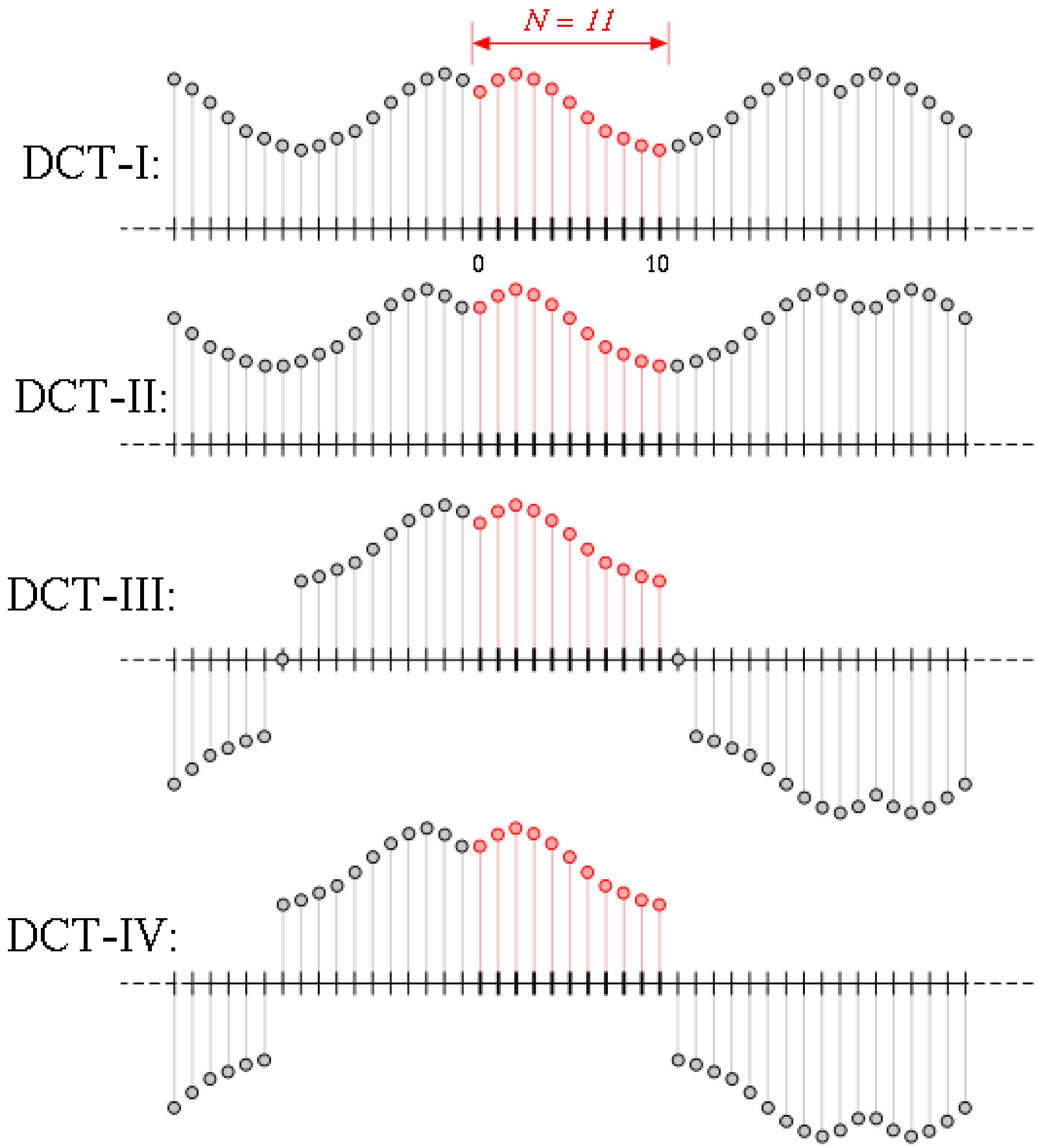


Illustration of the implicit even/odd extensions of DCT input data, for *N*=11 data points (red dots), for the four most common types of DCT (types I-IV).

However, because DCTs operate on *finite*, *discrete* sequences, two issues arise that do not apply for the continuous cosine transform. First, one has to specify whether the function is even or odd at *both* the left and right boundaries of the domain (i.e. the min-*n* and max-*n* boundaries in the definitions below, respectively). Second, one has to specify around *what point* the function is even or odd. In particular, consider a sequence *abcd* of four equally spaced data points, and say that we specify an even *left* boundary. There are two sensible possibilities: either the data is even about the sample *a*, in which case the even extension is *dcbabcd*, or the data is even about the point *halfway* between *a* and the previous point, in which case the even extension is *dcbaabcd* (*a* is repeated).

These choices lead to all the standard variations of DCTs and also discrete sine transforms (DSTs). Each boundary can be either even or odd (2 choices per boundary) and can be symmetric about a data point or the point halfway between two data points (2 choices per boundary), for a total of $2 \times 2 \times 2 \times 2 = 16$ possibilities. Half of these possibilities, those where the *left* boundary is even, correspond to the 8 types of DCT; the other half are the 8 types of DST.

These different boundary conditions strongly affect the applications of the transform, and lead to uniquely useful properties for the various DCT types. Most directly, when using Fourier-related transforms to solve partial differential equations by spectral methods, the boundary conditions are directly specified as a part of the problem being solved. Or, for the MDCT (based on the type-IV DCT), the boundary conditions are intimately involved in the MDCT's critical property of time-domain aliasing cancellation. In a more subtle fashion, the boundary conditions are responsible for the "energy compaction" properties that make DCTs useful for image and audio compression, because the boundaries affect the rate of convergence of any Fourier-like series.

In particular, it is well known that any discontinuities in a function reduce the rate of convergence of the Fourier series, so that more sinusoids are needed to represent the function with a given accuracy. The same principle governs the usefulness of the DFT and other transforms for signal compression: the smoother a function is, the fewer terms in its DFT or DCT are required to represent it accurately, and the more it can be compressed. (Here, we think of the DFT or DCT as approximations for the Fourier series or cosine series of a function, respectively, in order to talk about its "smoothness".) However, the implicit periodicity of the DFT means that discontinuities usually occur at the boundaries: any random segment of a signal is unlikely to have the same value at both the left and right boundaries. (A similar problem arises for the DST, in which the odd left boundary condition implies a discontinuity for any function that does not happen to be zero at that boundary.) In contrast, a DCT where *both* boundaries are even *always* yields a continuous extension at the boundaries (although the slope is generally discontinuous). This is why DCTs, and in particular DCTs of types I, II, V, and VI (the types that have two even boundaries) generally perform better for signal compression than DFTs and DSTs. In practice, a type-II DCT is usually preferred for such applications, in part for reasons of computational convenience.

Formal definition

Formally, the discrete cosine transform is a linear, invertible function $F : \mathbf{R}^N \to \mathbf{R}^N$ (where **R** denotes the set of real numbers), or equivalently an invertible $N \times N$ square matrix. There are several variants of the DCT with slightly modified definitions. The N real numbers $x_0, ..., x_{N-1}$ are transformed into the N real numbers $X_0, ..., X_{N-1}$ according to one of the formulas:

DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1.$$

Some authors further multiply the x_0 and x_{N-1} terms by √2, and correspondingly multiply the X_0 and X_{N-1} terms by 1/√2. This makes the DCT-I matrix orthogonal, if one further multiplies by an overall scale factor of $\sqrt{2/(N-1)}$, but breaks the direct correspondence with a real-even DFT.

The DCT-I is exactly equivalent (up to an overall scale factor of 2), to a DFT of $2N - 2$ real numbers with even symmetry. For example, a DCT-I of N=5 real numbers *abcde* is exactly equivalent to a DFT of eight real numbers *abcdedcb* (even symmetry), divided by two. (In contrast, DCT types II-IV involve a half-sample shift in the equivalent DFT.)

Note, however, that the DCT-I is not defined for N less than 2. (All other DCT types are defined for any positive N.)

Thus, the DCT-I corresponds to the boundary conditions: x_n is even around n=0 and even around n=N-1; similarly for X_k.

DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right) k\right] \qquad k = 0, \ldots, N-1.$$

The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT".

This transform is exactly equivalent (up to an overall scale factor of 2) to a DFT of $4N$ real inputs of even symmetry where the even-indexed elements are zero. That is, it is half of the DFT of the $4N$ inputs y_n, where $y_{2n} = 0$, $y_{2n+1} = x_n$ for $0 \le n < N$, and $y_{4N-n} = y_n$ for $0 < n < 2N$.

Some authors further multiply the X_0 term by 1/√2 and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$ (see below for the corresponding change in DCT-III). This makes the DCT-II matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted input.

The DCT-II implies the boundary conditions: x_n is even around n=-1/2 and even around n=N-1/2; X_k is even around k=0 and odd around k=N.

DCT-III

$$X_k = \frac{1}{2} x_0 + \sum_{n=1}^{N-1} x_n \cos \left[\frac{\pi}{N} n \left(k+\frac{1}{2}\right) \right] \quad \quad k = 0, \dots, N-1.$$

Because it is the inverse of DCT-II (up to a scale factor, see below), this form is sometimes simply referred to as "the inverse DCT" ("IDCT").

Some authors further multiply the x_0 term by $\sqrt{2}$ and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$ (see above for the corresponding change in DCT-II), so that the DCT-II and DCT-III are transposes of one another. This makes the DCT-III matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted output.

The DCT-III implies the boundary conditions: x_n is even around n=0 and odd around n=N; X_k is even around k=-1/2 and even around k=N-1/2.

DCT-IV

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n+\frac{1}{2}\right) \left(k+\frac{1}{2}\right) \right] \quad \quad k = 0, \dots, N-1.$$

The DCT-IV matrix becomes orthogonal (and thus, being clearly symmetric, its own inverse) if one further multiplies by an overall scale factor of $\sqrt{2/N}$.

A variant of the DCT-IV, where data from different transforms are *overlapped*, is called the modified discrete cosine transform (MDCT) (Malvar, 1992).

The DCT-IV implies the boundary conditions: x_n is even around n=-1/2 and odd around n=N-1/2; similarly for X_k.

DCT V-VIII

DCT types I-IV are equivalent to real-even DFTs of even order (regardless of whether N is even or odd), since the corresponding DFT is of length 2(N−1) (for DCT-I) or 4N (for DCT-II/III) or 8N (for DCT-VIII). In principle, there are actually four additional types of discrete cosine transform (Martucci, 1994), corresponding essentially to real-even DFTs of logically odd order, which have factors of N±½ in the denominators of the cosine arguments.

Equivalently, DCTs of types I-IV imply boundaries that are even/odd around either a data point for both boundaries or halfway between two data points for both boundaries. DCTs of types V-VIII imply boundaries that even/odd around a data point for one boundary and halfway between two data points for the other boundary.

However, these variants seem to be rarely used in practice. One reason, perhaps, is that FFT algorithms for odd-length DFTs are generally more complicated than FFT algorithms for even-length DFTs (e.g. the simplest radix-2 algorithms are only for even lengths), and this increased intricacy carries over to the DCTs as described below.

(The trivial real-even array, a length-one DFT (odd length) of a single number *a*, corresponds to a DCT-V of length $N=1$.)

Inverse transforms

Using the normalization conventions above, the inverse of DCT-I is DCT-I multiplied by $2/(N-1)$. The inverse of DCT-IV is DCT-IV multiplied by $2/N$. The inverse of DCT-II is DCT-III multiplied by $2/N$ and vice versa.

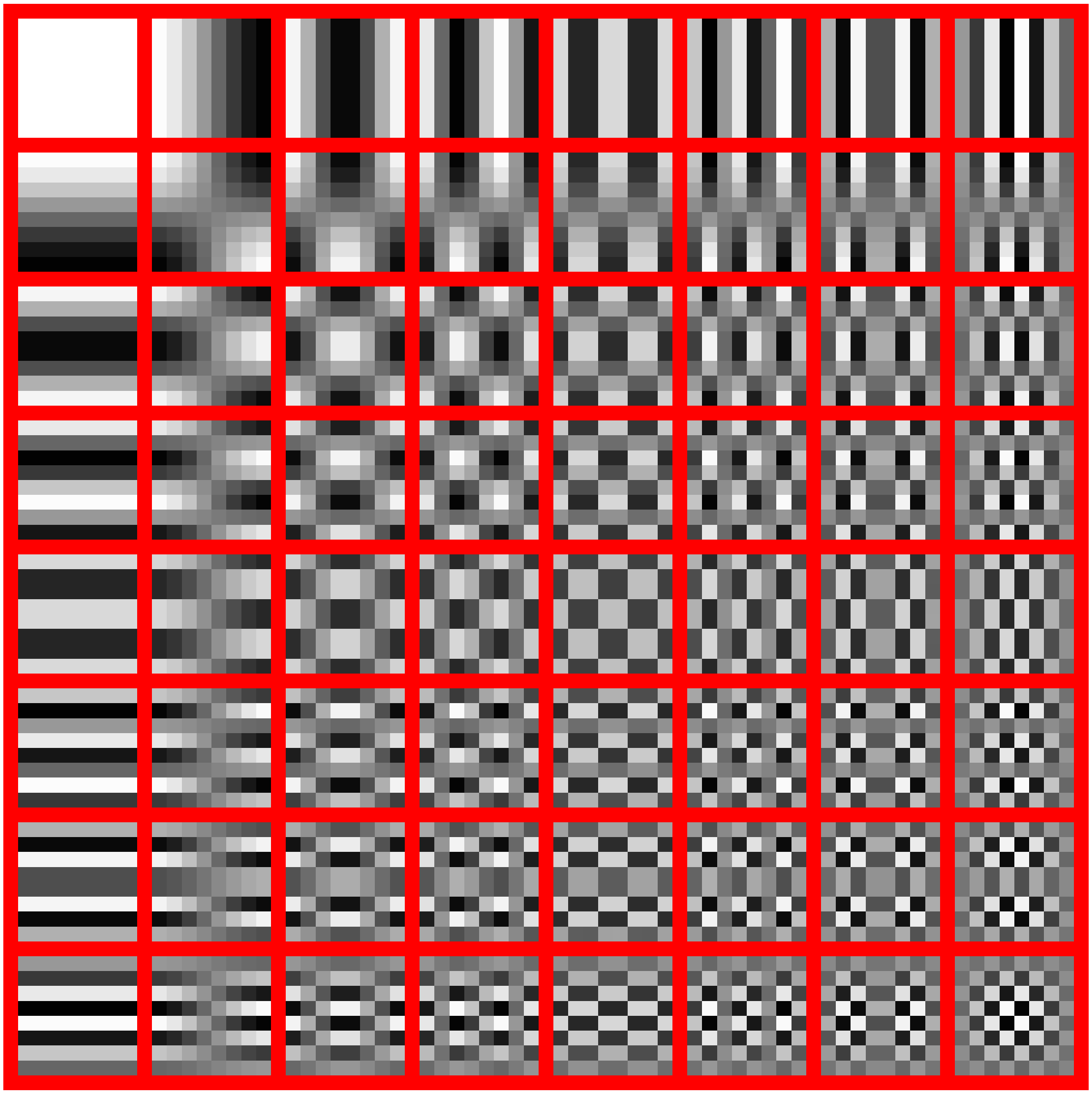
Like for the DFT, the normalization factor in front of these transform definitions is merely a convention and differs between treatments. For example, some authors multiply the transforms by $\sqrt{2/N}$ so that the inverse does not require any additional multiplicative factor. Combined with appropriate factors of √2 (see above), this can be used to make the transform matrix orthogonal.

Multidimensional DCTs

Multidimensional variants of the various DCT types follow straightforwardly from the one-dimensional definitions: they are simply a separable product (equivalently, a composition) of DCTs along each dimension.

For example, a two-dimensional DCT-II of an image or a matrix is simply the one-dimensional DCT-II, from above, performed along the rows and then along the columns (or vice versa). That is, the 2d DCT-II is given by the formula (omitting normalization and other scale factors, as above):

$$\begin{aligned} X_{k_1,k_2} &= \sum_{n_1=0}^{N_1-1} \left(\sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right) \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \\ &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]. \end{aligned}$$



Two-dimensional DCT frequencies

Technically, computing a two- (or multi-) dimensional DCT by sequences of one-dimensional DCTs along each dimension is known as a *row-column* algorithm (after the two-dimensional case). As with multidimensional FFT algorithms, however, there exist other methods to compute the same thing while performing the computations in a different order (i.e. interleaving/combining the algorithms for the different dimensions).

The inverse of a multi-dimensional DCT is just a separable product of the inverse(s) of the corresponding one-dimensional DCT(s) (see above), e.g. the one-dimensional inverses applied along one dimension at a time in a row-column algorithm.

The image to the right shows combination of horizontal and vertical frequencies for an 8 x 8 ($N_1 = N_2 = 8$) two-dimensional DCT. Each step from left to right and top to bottom is an increase in frequency by 1/2 cycle. For example, moving right one from the top-left square yields a half-cycle increase in the horizontal frequency. Another move to the right yields two half-cycles. A move down yields two half-cycles horizontally and a half-cycle vertically. The source data (8x8) is transformed to a linear combination of these 64 frequency squares.

Computation

Although the direct application of these formulas would require O(N^2) operations, it is possible to compute the same thing with only O(*N* log *N*) complexity by factorizing the computation similarly to the fast Fourier transform (FFT). One can also compute DCTs via FFTs combined with O(*N*) pre- and post-processing steps. In general, O(*N* log *N*) methods to compute DCTs are known as **fast cosine transform** (FCT) algorithms.

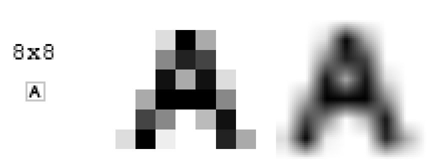
The most efficient algorithms, in principle, are usually those that are specialized directly for the DCT, as opposed to using an ordinary FFT plus O(*N*) extra operations (see below for an exception). However, even "specialized" DCT algorithms (including all of those that achieve the lowest known arithmetic counts, at least for power-of-two sizes) are typically closely related to FFT algorithms—since DCTs are essentially DFTs of real-even data, one can design a fast DCT algorithm by taking an FFT and eliminating the redundant operations due to this symmetry. This can even be done automatically (Frigo & Johnson, 2005). Algorithms based on the Cooley–Tukey FFT algorithm are most common, but any other FFT algorithm is also applicable. For example, the Winograd FFT algorithm leads to minimal-multiplication algorithms for the DFT, albeit generally at the cost of more additions, and a similar algorithm was proposed by Feig & Winograd (1992) for the DCT. Because the algorithms for DFTs, DCTs, and similar transforms are all so closely related, any improvement in algorithms for one transform will theoretically lead to immediate gains for the other transforms as well (Duhamel & Vetterli, 1990).

While DCT algorithms that employ an unmodified FFT often have some theoretical overhead compared to the best specialized DCT algorithms, the former also have a distinct advantage: highly optimized FFT programs are widely available. Thus, in practice, it is often easier to obtain high performance for general lengths *N* with FFT-based algorithms. (Performance on modern hardware is typically not dominated simply by arithmetic counts, and optimization requires substantial engineering effort.) Specialized DCT algorithms, on the other hand, see widespread use for transforms of small, fixed sizes such as the 8×8DCT-II used in JPEG compression, or the small DCTs (or MDCTs) typically used in audio compression. (Reduced code size may also be a reason to use a specialized DCT for embedded-device applications.)

In fact, even the DCT algorithms using an ordinary FFT are sometimes equivalent to pruning the redundant operations from a larger FFT of real-symmetric data, and they can even be optimal from the perspective of arithmetic counts. For example, a type-II DCT is equivalent to a DFT of size 4*N* with real-even symmetry whose even-indexed elements are zero. One of the most common methods for computing this via an FFT (e.g. the method used in FFTPACK and FFTW) was described by Narasimha & Peterson (1978) and Makhoul (1980), and this method in hindsight can be seen as one step of a radix-4 decimation-in-time Cooley–Tukey algorithm applied to the "logical" real-even DFT corresponding to the DCT II. (The radix-4 step reduces the size $4N$ DFT to four size-N DFTs of real data, two of which are zero and two of which are equal to one another by the even symmetry, hence giving a single size-N FFT of real data plus $O(N)$ butterflies.) Because the even-indexed elements are zero, this radix-4 step is exactly the same as a split-radix step; if the subsequent size-N real-data FFT is also performed by a real-data split-radix algorithm (as in Sorensen et al., 1987), then the resulting algorithm actually matches what was long the lowest published arithmetic count for the power-of-two DCT-II ($2N\log_2 N - N + 2$ real-arithmetic operations). So, there is nothing intrinsically bad about computing the DCT via an FFT from an arithmetic perspective—it is sometimes merely a question of whether the corresponding FFT algorithm is optimal. (As a practical matter, the function-call overhead in invoking a separate FFT routine might be significant for small N, but this is an implementation rather than an algorithmic question since it can be solved by unrolling/inlining.)

Example of IDCT

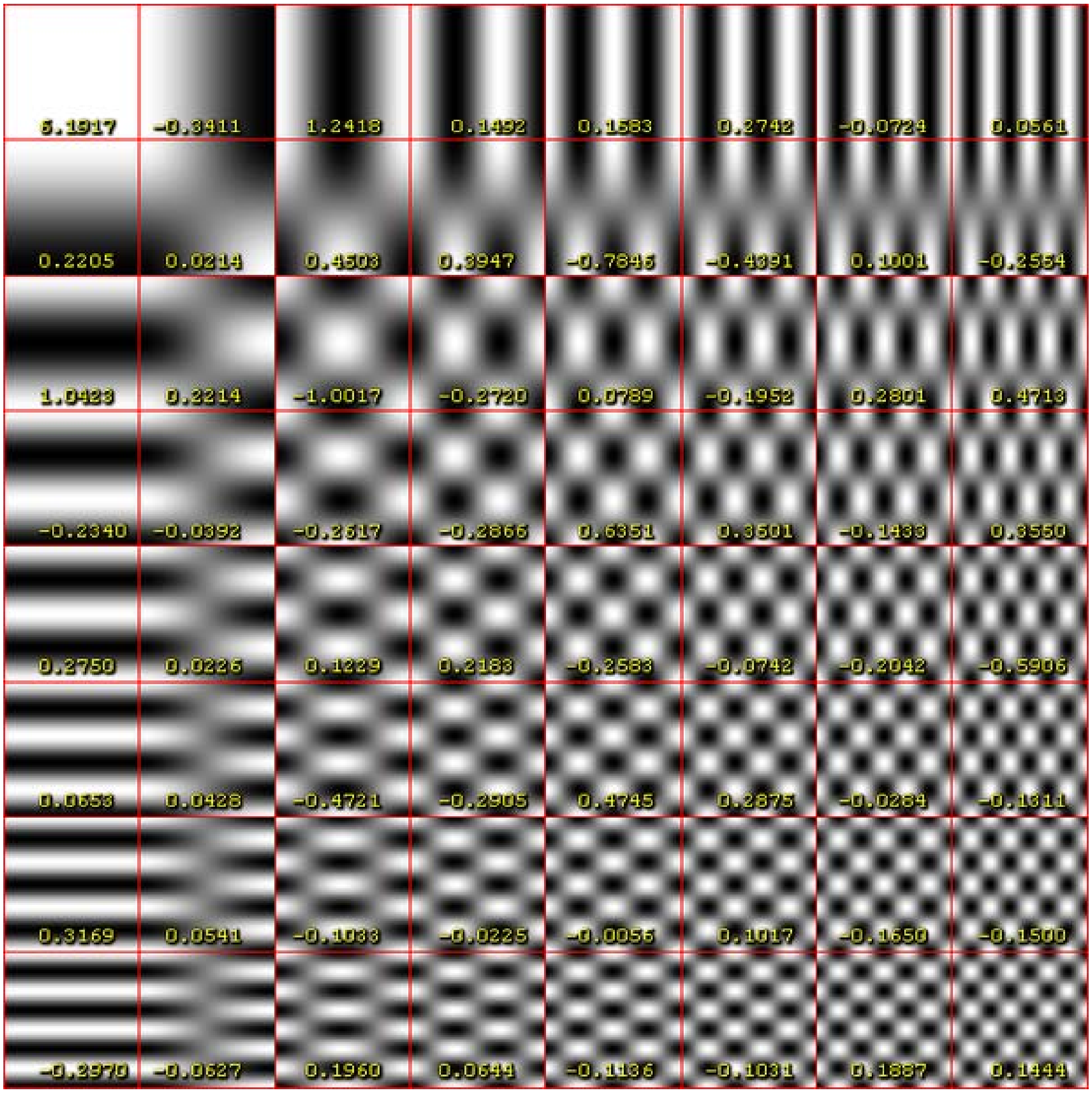
Consider this 8x8 grayscale image of capital letter A.



Original size, scaled 10x (nearest neighbor), scaled 10x (bilinear).

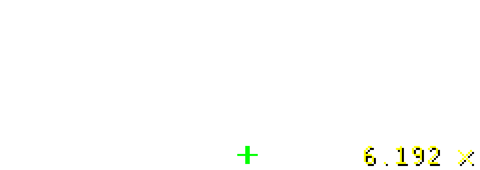
DCT of the image.

$$\begin{bmatrix} 6.1917 & -0.3411 & 1.2418 & 0.1492 & 0.1583 & 0.2742 & -0.0724 & 0.0561 \\ 0.2205 & 0.0214 & 0.4503 & 0.3947 & -0.7846 & -0.4391 & 0.1001 & -0.2554 \\ 1.0423 & 0.2214 & -1.0017 & -0.2720 & 0.0789 & -0.1952 & 0.2801 & 0.4713 \\ -0.2340 & -0.0392 & -0.2617 & -0.2866 & 0.6351 & 0.3501 & -0.1433 & 0.3550 \\ 0.2750 & 0.0226 & 0.1229 & 0.2183 & -0.2583 & -0.0742 & -0.2042 & -0.5906 \\ 0.0653 & 0.0428 & -0.4721 & -0.2905 & 0.4745 & 0.2875 & -0.0284 & -0.1311 \\ 0.3169 & 0.0541 & -0.1033 & -0.0225 & -0.0056 & 0.1017 & -0.1650 & -0.1500 \\ -0.2970 & -0.0627 & 0.1960 & 0.0644 & -0.1136 & -0.1031 & 0.1887 & 0.1444 \end{bmatrix}$$



Basis functions of the discrete cosine transformation with corresponding coefficients (specific for our image).

Each basis function is multiplied by its coefficient and then this product is added to the final image.



On the left is final image. In the middle is weighted function (multiplied by coefficient) which is added to the final image. On the right is the current function and corresponding coefficient. Images are scaled (using bilinear interpolation) by factor 10x.

Chapter- 4

Discrete Fourier Transform

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). But the DFT requires an input function that is *discrete* and whose non-zero values have a limited (*finite*) duration. Such inputs are often created by sampling a continuous function, like a person's voice. Unlike the discrete-time Fourier transform (DTFT), it only evaluates enough frequency components to reconstruct the finite segment that was analyzed. Using the DFT implies that the finite segment that is analyzed is one period of an infinitely extended periodic signal; if this is not actually true, a window function has to be used to reduce the artifacts in the spectrum. For the same reason, the inverse DFT cannot reproduce the entire time domain, unless the input happens to be periodic (forever). Therefore it is often said that the DFT is a transform for Fourier analysis of finite-domain discrete-time functions. The sinusoidal basis functions of the decomposition have the same properties.

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.

Definition

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

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

where i is the imaginary unit and $e^{\frac{2\pi i}{N}}$ is a primitive N'th root of unity. (This 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.)

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

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

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

A simple description of these equations is that the complex numbers X_k represent the amplitude and phase of the different sinusoidal components of the input "signal" x_n. The DFT computes the X_k from the x_n, while the IDFT shows how to compute the x_n as a sum of sinusoidal components $(1/N) X_k e^{\frac{2\pi i}{N} kn}$ with frequency k / N cycles per sample. By writing the equations in this form, we are making extensive use of Euler's formula to express sinusoids in terms of complex exponentials, which are much easier to manipulate. In the same way, by writing X_k in polar form, we obtain the sinusoid amplitude A_k / N and phase φ_k from the complex modulus and argument of X_k, respectively:

$$A_k = |X_k| = \sqrt{\operatorname{Re}(X_k)^2 + \operatorname{Im}(X_k)^2},$$
$$\varphi_k = \arg(X_k) = \operatorname{atan2}\big(\operatorname{Im}(X_k), \operatorname{Re}(X_k)\big),$$

where atan2 is the two-argument form of the arctan function. Note that 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 $1/\sqrt{N}$ 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" $2\pi k / N$. Equivalently, the DFT is often thought of as a matched filter: when looking for a frequency of +1, one correlates the incoming signal with a frequency of −1.)

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

Properties

Completeness

The discrete Fourier transform is an invertible, linear transformation

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

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

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

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

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

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

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 continuous and discrete time Fourier transforms indicates that a convolution of two infinite sequences can be obtained as the inverse transform of the product of the individual transforms. With sequences and transforms of length N, a circularity arises**:**

$$\begin{aligned}
\mathcal{F}^{-1}\left\{\mathbf{X}\cdot\mathbf{Y}\right\}_n &\stackrel{\text{def}}{=} \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot Y_k \cdot e^{\frac{2\pi i}{N}kn} \\
&= \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{l=0}^{N-1} x_l e^{-\frac{2\pi i}{N}kl}\right)\cdot\left(\sum_{m=0}^{N-1} y_m e^{-\frac{2\pi i}{N}km}\right)\cdot e^{\frac{2\pi i}{N}kn} \\
&= \sum_{l=0}^{N-1} x_l \sum_{m=0}^{N-1} y_m \left(\frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{2\pi i}{N}k(n-l-m)}\right).
\end{aligned}$$

The quantity in parentheses is 0 for all values of m except those of the form $n - l - pN$, where p is any integer. At those values, it is 1. It can therefore be replaced by an infinite sum of Kronecker delta functions, and we continue accordingly. Note that we can also extend the limits of m to infinity, with the understanding that the x and y sequences are defined as 0 outside [0,N-1]**:**

$$\begin{aligned}\mathcal{F}^{-1}\left\{\mathbf{X}\cdot\mathbf{Y}\right\}_n &= \sum_{l=0}^{N-1} x_l \sum_{m=-\infty}^{\infty} y_m \left(\sum_{p=-\infty}^{\infty} \delta_{m,(n-l-pN)}\right)\\ &= \sum_{l=0}^{N-1} x_l \sum_{p=-\infty}^{\infty} \underbrace{\left(\sum_{m=-\infty}^{\infty} y_m \cdot \delta_{m,(n-l-pN)}\right)}_{y_{n-l-pN}}\\ &= \sum_{l=0}^{N-1} x_l \left(\sum_{p=-\infty}^{\infty} y_{n-l-pN}\right) \stackrel{\text{def}}{=} (\mathbf{x} * \mathbf{y_N})_n \ ,\end{aligned}$$

which is the convolution of the $\mathbf{x}$ sequence with a periodically extended $\mathbf{y}$ sequence defined by**:**

$$(\mathbf{y_N})_n \stackrel{\text{def}}{=} \sum_{p=-\infty}^{\infty} y_{(n-pN)} = y_{n(modN)}.$$

Similarly, it can be shown that**:**

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

which is the cross-correlation of $\mathbf{x}$ and $\mathbf{y_N}$.

A direct evaluation of the convolution or correlation summation (above) requires $O(N^2)$ operations for an output sequence of length N. An indirect method, using transforms, can take advantage of the $O(N\log N)$ 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.

Convolution theorem duality

It can also be shown that:

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

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

Trigonometric interpolation polynomial

The trigonometric interpolation polynomial

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

for N even ,

$$p(t) = \frac{1}{N}\left[X_0 + X_1 e^{it} + \cdots + X_{\lfloor N/2\rfloor} e^{\lfloor N/2\rfloor it} + X_{\lfloor N/2\rfloor+1} e^{-\lfloor N/2\rfloor it} + \cdots + X_{N-1} e^{-it}\right]$$

for N 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 $n = 0, \ldots, N-1$.

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, and therefore minimizes the mean-square slope $\int |p'(t)|^2 dt$of the interpolating function. 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:

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

where

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

is a primitive Nth root of unity. The inverse transform is then given by the inverse of the above matrix:

$$\mathbf{F}^{-1} = \frac{1}{N}\mathbf{F}^*$$

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

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

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

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

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 involutary—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 involutary transformation (by a factor of (1+*i*) /√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 involutary.

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

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

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. (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:

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 **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 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:

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

  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:

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

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.

The real-input DFT

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

$$X_k=X_{N-k}^*.$$

The star denotes complex conjugation. The subscripts are interpreted modulo *N*.

Therefore, the DFT output for real inputs is half redundant, and one obtains the complete information by only looking at roughly half of the outputs $X_0,\ldots,X_{N-1}$. In this case, the "DC" element X_0 is purely real, and for even *N* the "Nyquist" element $X_{N/2}$ is also real, so there are exactly *N* non-redundant real numbers in the first half + Nyquist element of the complex output *X*.

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

Generalized/shifted DFT

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

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

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 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)

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 ℓ in $1, 2, \dots, d$ is defined by:

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

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 $\mathbf{N} - 1$, which we define as $\mathbf{N} - 1 = (N_1 - 1, N_2 - 1, \dots, N_d - 1)$:

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

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 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 ℓ-th subscript is again interpreted modulo N_ℓ (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. 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 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 in the discrete-time Fourier transform article.

- 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.

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.)

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 $i\ n$. (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 **a** and **b** have dimension $d > \deg(a(x)) + \deg(b(x))$. Then,

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

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

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

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 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.

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
$\begin{cases} \frac{1}{W} & \text{if } 2n < W \text{ or } 2(N-n) < W \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } k = 0 \\ \frac{\sin\left(\frac{\pi W k}{N}\right)}{W\sin\left(\frac{\pi k}{N}\right)} & \text{otherwise} \end{cases}$	x_n is a rectangular window function of W points centered on x_0, where W is an odd integer, and X_k is a sinc-like function
$\sum_{j\in\mathbb{Z}} \exp\left(-\frac{\pi}{cN}\cdot(n+N\cdot j)^2\right)$	$cN\cdot\sum_{j\in\mathbb{Z}} \exp\left(-\frac{\pi c}{N}\cdot(k+N\cdot j)^2\right)$	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.

Derivation as Fourier series

The DFT can be derived as a truncation of the Fourier series of a periodic sequence of impulses.

Generalizations

Representation theory

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, $\mathbf{Z}/n\mathbf{Z} \to \mathbf{C}$. This latter may be suggestively written $\mathbf{C}^{\mathbf{Z}/n\mathbf{Z}}$ to emphasize that this is a complex vector space whose coordinates are indexed by the *n*-element set $\mathbf{Z}/n\mathbf{Z}$.

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

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 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 fields other than the complex numbers, and such generalizations are commonly called *number-theoretic transforms* (NTTs) in the case of finite fields.

Other finite groups

The standard DFT acts on a sequence $x_0, x_1, \ldots, x_{N-1}$ of complex numbers, which can be viewed as a function $\{0, 1, \ldots, N-1\} \to \mathbf{C}$. 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 Fourier transforms on arbitrary finite groups, which act on functions $G \to \mathbf{C}$ 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

As with other Fourier transforms, 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.

Chapter- 5

Fast Fourier Transform

A **fast Fourier transform** (**FFT**) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory; this article gives an overview of the available techniques and some of their general properties, while the specific algorithms are described in subsidiary articles linked below.

A DFT decomposes a sequence of values into components of different frequencies. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly: computing a DFT of N points in the naive way, using the definition, takes $O(N^2)$ arithmetical operations, while an FFT can compute the same result in only $O(N \log N)$ operations. The difference in speed can be substantial, especially for long data sets where N may be in the thousands or millions—in practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to $N / \log(N)$. This huge improvement made many DFT-based algorithms practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers.

The most well known FFT algorithms depend upon the factorization of N, but (contrary to popular misconception) there are FFTs with $O(N \log N)$ complexity for all N, even for prime N. Many FFT algorithms only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is an Nth primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms.

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

Definition and speed

An FFT computes the DFT and produces exactly the same result as evaluating the DFT definition directly; the only difference is that an FFT is much faster. (In the presence of round-off error, many FFT algorithms are also much more accurate than evaluating the DFT definition directly, as discussed below.)

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

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}} \qquad k = 0, \dots, N-1.$$

Evaluating this definition directly requires $O(N^2)$ operations: there are N outputs X_k, and each output requires a sum of N terms. An FFT is any method to compute the same results in $O(N \log N)$ operations. More precisely, all known FFT algorithms require $\Theta(N \log N)$ operations (technically, O only denotes an upper bound), although there is no proof that better complexity is impossible.

To illustrate the savings of an FFT, consider the count of complex multiplications and additions. Evaluating the DFT's sums directly involves N^2 complex multiplications and $N(N-1)$ complex additions [of which $O(N)$ operations can be saved by eliminating trivial operations such as multiplications by 1]. The well-known radix-2 Cooley–Tukey algorithm, for N a power of 2, can compute the same result with only $(N/2) \log_2 N$ complex multiplies (again, ignoring simplifications of multiplications by 1 and similar) and $N \log_2 N$ complex additions. In practice, actual performance on modern computers is usually dominated by factors other than arithmetic and is a complicated subject (see, e.g., Frigo & Johnson, 2005), but the overall improvement from $\Theta(N^2)$ to $\Theta(N \log N)$ remains.

Algorithms

Cooley–Tukey algorithm

By far the most common FFT is the Cooley–Tukey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1N_2$ into many smaller DFTs of sizes N_1 and N_2, along with $O(N)$ multiplications by complex roots of unity traditionally called twiddle factors (after Gentleman and Sande, 1966).

This method (and the general idea of an FFT) was popularized by a publication of J. W. Cooley and J. W. Tukey in 1965, but it was later discovered (Heideman & Burrus, 1984) that those two authors had independently re-invented an algorithm known to Carl Friedrich Gauss around 1805 (and subsequently rediscovered several times in limited forms).

The most well-known use of the Cooley–Tukey algorithm is to divide the transform into two pieces of size $N / 2$ at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general (as was known to both Gauss and Cooley/Tukey). These are called the **radix-2** and **mixed-radix** cases, respectively (and other variants such as the split-radix FFT have their own names as well). Although the basic idea is recursive, most traditional implementations rearrange the algorithm to avoid explicit recursion. Also, because the Cooley–Tukey algorithm breaks the DFT into smaller DFTs, it can be combined arbitrarily with any other algorithm for the DFT, such as those described below.

Other FFT algorithms

There are other FFT algorithms distinct from Cooley–Tukey. For $N = N_1N_2$ with coprime N_1 and N_2, one can use the Prime-Factor (Good-Thomas) algorithm (PFA), based on the Chinese Remainder Theorem, to factorize the DFT similarly to Cooley–Tukey but without the twiddle factors. The Rader-Brenner algorithm (1976) is a Cooley–Tukey-like factorization but with purely imaginary twiddle factors, reducing multiplications at the cost of increased additions and reduced numerical stability; it was later superseded by the split-radix variant of Cooley–Tukey (which achieves the same multiplication count but with fewer additions and without sacrificing accuracy). Algorithms that recursively factorize the DFT into smaller operations other than DFTs include the Bruun and QFT algorithms. (The Rader-Brenner and QFT algorithms were proposed for power-of-two sizes, but it is possible that they could be adapted to general composite n. Bruun's algorithm applies to arbitrary even composite sizes.) Bruun's algorithm, in particular, is based on interpreting the FFT as a recursive factorization of the polynomial $z^N - 1$, here into real-coefficient polynomials of the form $z^M - 1$ and $z^{2M} + az^M + 1$.

Another polynomial viewpoint is exploited by the Winograd algorithm, which factorizes $z^N - 1$ into cyclotomic polynomials—these often have coefficients of 1, 0, or −1, and therefore require few (if any) multiplications, so Winograd can be used to obtain minimal-multiplication FFTs and is often used to find efficient algorithms for small factors. Indeed, Winograd showed that the DFT can be computed with only $O(N)$ irrational multiplications, leading to a proven achievable lower bound on the number of multiplications for power-of-two sizes; unfortunately, this comes at the cost of many more additions, a tradeoff no longer favorable on modern processors with hardware multipliers. In particular, Winograd also makes use of the PFA as well as an algorithm by Rader for FFTs of *prime* sizes.

Rader's algorithm, exploiting the existence of a generator for the multiplicative group modulo prime N, expresses a DFT of prime size n as a cyclic convolution of (composite) size $N - 1$, which can then be computed by a pair of ordinary FFTs via the convolution theorem (although Winograd uses other convolution methods). Another prime-size FFT is due to L. I. Bluestein, and is sometimes called the chirp-z algorithm; it also re-expresses a DFT as a convolution, but this time of the *same* size (which can be zero-padded to a power of two and evaluated by radix-2 Cooley–Tukey FFTs, for example), via the identity $nk = -(k - n)^2 / 2 + n^2 / 2 + k^2 / 2$.

FFT algorithms specialized for real and/or symmetric data

In many applications, the input data for the DFT are purely real, in which case the outputs satisfy the symmetry

$$X_{N-k} = X_k^*,$$

and efficient FFT algorithms have been designed for this situation. One approach consists of taking an ordinary algorithm (e.g. Cooley–Tukey) and removing the redundant parts of the computation, saving roughly a factor of two in time and memory. Alternatively, it is possible to express an *even*-length real-input DFT as a complex DFT of half the length (whose real and imaginary parts are the even/odd elements of the original real data), followed by O(N) post-processing operations.

It was once believed that real-input DFTs could be more efficiently computed by means of the discrete Hartley transform (DHT), but it was subsequently argued that a specialized real-input DFT algorithm (FFT) can typically be found that requires fewer operations than the corresponding DHT algorithm (FHT) for the same number of inputs. Bruun's algorithm (above) is another method that was initially proposed to take advantage of real inputs, but it has not proved popular.

There are further FFT specializations for the cases of real data that have even/odd symmetry, in which case one can gain another factor of (roughly) two in time and memory and the DFT becomes the discrete cosine/sine transform(s) (DCT/DST). Instead of directly modifying an FFT algorithm for these cases, DCTs/DSTs can also be computed via FFTs of real data combined with O(N) pre/post processing.

Computational issues

Bounds on complexity and operation counts

A fundamental question of longstanding theoretical interest is to prove lower bounds on the complexity and exact operation counts of fast Fourier transforms, and many open problems remain. It is not even rigorously proved whether DFTs truly require $\Omega(N\log N)$ (i.e., order $N\log N$ or greater) operations, even for the simple case of power of two sizes, although no algorithms with lower complexity are known. In particular, the count of arithmetic operations is usually the focus of such questions, although actual performance on modern-day computers is determined by many other factors such as cache or CPU pipeline optimization.

Following pioneering work by Winograd (1978), a tight $\Theta(N)$ lower bound *is* known for the number of real multiplications required by an FFT. It can be shown that only $4N - 2\log_2^2 N - 2\log_2 N - 4$ irrational real multiplications are required to compute a DFT of power-of-two length $N = 2^m$. Moreover, explicit algorithms that achieve this count are known (Heideman & Burrus, 1986; Duhamel, 1990).

Unfortunately, these algorithms require too many additions to be practical, at least on modern computers with hardware multipliers.

A tight lower bound is *not* known on the number of required additions, although lower bounds have been proved under some restrictive assumptions on the algorithms. In 1973, Morgenstern proved an $\Omega(N \log N)$ lower bound on the addition count for algorithms where the multiplicative constants have bounded magnitudes (which is true for most but not all FFT algorithms). Pan (1986) proved an $\Omega(N \log N)$ lower bound assuming a bound on a measure of the FFT algorithm's "asynchronicity", but the generality of this assumption is unclear. For the case of power-of-two N, Papadimitriou (1979) argued that the number $N \log_2 N$ of complex-number additions achieved by Cooley–Tukey algorithms is *optimal* under certain assumptions on the graph of the algorithm (his assumptions imply, among other things, that no additive identities in the roots of unity are exploited). (This argument would imply that at least $2N \log_2 N$ real additions are required, although this is not a tight bound because extra additions are required as part of complex-number multiplications.) Thus far, no published FFT algorithm has achieved fewer than $N \log_2 N$ complex-number additions (or their equivalent) for power-of-two N.

A third problem is to minimize the *total* number of real multiplications and additions, sometimes called the "arithmetic complexity" (although in this context it is the exact count and not the asymptotic complexity that is being considered). Again, no tight lower bound has been proven. Since 1968, however, the lowest published count for power-of-two N was long achieved by the split-radix FFT algorithm, which requires $4N \log_2 N - 6N + 8$ real multiplications and additions for $N > 1$. This was recently reduced to $\sim \frac{34}{9} N \log_2 N$ (Johnson and Frigo, 2007; Lundy and Van Buskirk, 2007).

Most of the attempts to lower or prove the complexity of FFT algorithms have focused on the ordinary complex-data case, because it is the simplest. However, complex-data FFTs are so closely related to algorithms for related problems such as real-data FFTs, discrete cosine transforms, discrete Hartley transforms, and so on, that any improvement in one of these would immediately lead to improvements in the others (Duhamel & Vetterli, 1990).

Accuracy and approximations

All of the FFT algorithms discussed below compute the DFT exactly (in exact arithmetic, i.e. neglecting floating-point errors). A few "FFT" algorithms have been proposed, however, that compute the DFT *approximately*, with an error that can be made arbitrarily small at the expense of increased computations. Such algorithms trade the approximation error for increased speed or other properties. For example, an approximate FFT algorithm by Edelman et al. (1999) achieves lower communication requirements for parallel computing with the help of a fast multipole method. A wavelet-based approximate FFT by Guo and Burrus (1996) takes sparse inputs/outputs (time/frequency localization) into account more efficiently than is possible with an exact FFT. Another algorithm for approximate computation of a subset of the DFT outputs is due to Shentov et al. (1995). Only the Edelman algorithm works equally well for sparse and non-sparse data, however, since it is based on the compressibility (rank deficiency) of the Fourier matrix itself rather than the compressibility (sparsity) of the data.

Even the "exact" FFT algorithms have errors when finite-precision floating-point arithmetic is used, but these errors are typically quite small; most FFT algorithms, e.g. Cooley–Tukey, have excellent numerical properties. The upper bound on the relative error for the Cooley–Tukey algorithm is $O(\varepsilon \log N)$, compared to $O(\varepsilon N^{3/2})$ for the naïve DFT formula (Gentleman and Sande, 1966), where ε is the machine floating-point relative precision. In fact, the root mean square (rms) errors are much better than these upper bounds, being only $O(\varepsilon \sqrt{\log N})$ for Cooley–Tukey and $O(\varepsilon \sqrt{N})$ for the naïve DFT (Schatzman, 1996). These results, however, are very sensitive to the accuracy of the twiddle factors used in the FFT (i.e. the trigonometric function values), and it is not unusual for incautious FFT implementations to have much worse accuracy, e.g. if they use inaccurate trigonometric recurrence formulas. Some FFTs other than Cooley–Tukey, such as the Rader-Brenner algorithm, are intrinsically less stable.

In fixed-point arithmetic, the finite-precision errors accumulated by FFT algorithms are worse, with rms errors growing as $O(\sqrt{N})$ for the Cooley–Tukey algorithm (Welch, 1969). Moreover, even achieving this accuracy requires careful attention to scaling in order to minimize the loss of precision, and fixed-point FFT algorithms involve rescaling at each intermediate stage of decompositions like Cooley–Tukey.

To verify the correctness of an FFT implementation, rigorous guarantees can be obtained in $O(N \log N)$ time by a simple procedure checking the linearity, impulse-response, and time-shift properties of the transform on random inputs (Ergün, 1995).

Multidimensional FFTs

As defined in the multidimensional DFT article, the multidimensional DFT

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

transforms an array $x_{\mathbf{n}}$ with a d-dimensional vector of indices $\mathbf{n} = (n_1, n_2, \ldots, n_d)$ by a set of d nested summations (over $n_j = 0 \ldots N_j - 1$ for each j), where the division $\mathbf{n}/\mathbf{N}$, defined as $\mathbf{n}/\mathbf{N} = (n_1/N_1, \cdots, n_d/N_d)$, is performed element-wise. Equivalently, it is simply the composition of a sequence of d sets of one-dimensional DFTs, performed along one dimension at a time (in any order).

This compositional viewpoint immediately provides the simplest and most common multidimensional DFT algorithm, known as the **row-column** algorithm (after the two-dimensional case, below). That is, one simply performs a sequence of d one-dimensional FFTs (by any of the above algorithms): first you transform along the n_1 dimension, then along the n_2 dimension, and so on (or actually, any ordering will work). This method is easily shown to have the usual $O(N \log N)$ complexity, where $N = N_1 N_2 \cdots N_d$ is the total number of data points transformed. In particular, there are N / N_1 transforms of size N_1, etcetera, so the complexity of the sequence of FFTs is:

$$\frac{N}{N_1} O(N_1 \log N_1) + \cdots + \frac{N}{N_d} O(N_d \log N_d)$$
$$= O\left(N \left[\log N_1 + \cdots + \log N_d\right]\right) = O(N \log N).$$

In two dimensions, the $x_{\mathbf{k}}$ can be viewed as an $n_1 \times n_2$ matrix, and this algorithm corresponds to first performing the FFT of all the rows and then of all the columns (or vice versa), hence the name.

In more than two dimensions, it is often advantageous for cache locality to group the dimensions recursively. For example, a three-dimensional FFT might first perform two-dimensional FFTs of each planar "slice" for each fixed n_1, and then perform the one-dimensional FFTs along the n_1 direction. More generally, an asymptotically optimal cache-oblivious algorithm consists of recursively dividing the dimensions into two groups $(n_1, \cdots, n_{d/2})$ and $(n_{d/2+1}, \cdots, n_d)$ that are transformed recursively (rounding if d is not even). Still, this remains a straightforward variation of the row-column algorithm that ultimately requires only a one-dimensional FFT algorithm as the base case, and still has $O(N \log N)$ complexity. Yet another variation is to perform matrix transpositions in between transforming subsequent dimensions, so that the transforms operate on contiguous data; this is especially important for out-of-core and distributed memory situations where accessing non-contiguous data is extremely time-consuming.

There are other multidimensional FFT algorithms that are distinct from the row-column algorithm, although all of them have $O(N \log N)$ complexity. Perhaps the simplest non-row-column FFT is the vector-radix FFT algorithm, which is a generalization of the ordinary Cooley–Tukey algorithm where one divides the transform dimensions by a vector $\mathbf{r} = (r_1, r_2, \cdots, r_d)$ of radices at each step. (This may also have cache benefits.) The simplest case of vector-radix is where all of the radices are equal (e.g. vector-radix-2 divides *all* of the dimensions by two), but this is not necessary. Vector radix with only a single non-unit radix at a time, i.e. $\mathbf{r} = (1, \cdots, 1, r, 1, \cdots, 1)$, is essentially a row-column algorithm. Other, more complicated, methods include polynomial transform algorithms due to Nussbaumer (1977), which view the transform in terms of convolutions and polynomial products.

Other generalizations

An $O(N^{5/2} \log N)$ generalization to spherical harmonics on the sphere S^2 with N^2 nodes was described by Mohlenkamp (1999), along with an algorithm conjectured (but not proven) to have $O(N^2 \log^2 N)$ complexity; Mohlenkamp also provides an implementation in the libftsh library. A spherical-harmonic algorithm with $O(N^2 \log N)$ complexity is described by Rokhlin and Tygert (2006).

Various groups have also published "FFT" algorithms for non-equispaced data, as reviewed in Potts *et al.* (2001). Such algorithms do not strictly compute the DFT (which is only defined for equispaced data), but rather some approximation thereof (a non-uniform discrete Fourier transform, or NDFT, which itself is often computed only approximately).

Chapter- 6

Discrete Sine Transform

In mathematics, the **discrete sine transform** (DST) is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using a purely real matrix. It is equivalent to the imaginary parts of a DFT of roughly twice the length, operating on real data with odd symmetry (since the Fourier transform of a real and odd function is imaginary and odd), where in some variants the input and/or output data are shifted by half a sample.

A related transform is the discrete cosine transform (DCT), which is equivalent to a DFT of real and *even* functions.

Applications

DSTs are widely employed in solving partial differential equations by spectral methods, where the different variants of the DST correspond to slightly different odd/even boundary conditions at the two ends of the array.

Informal overview

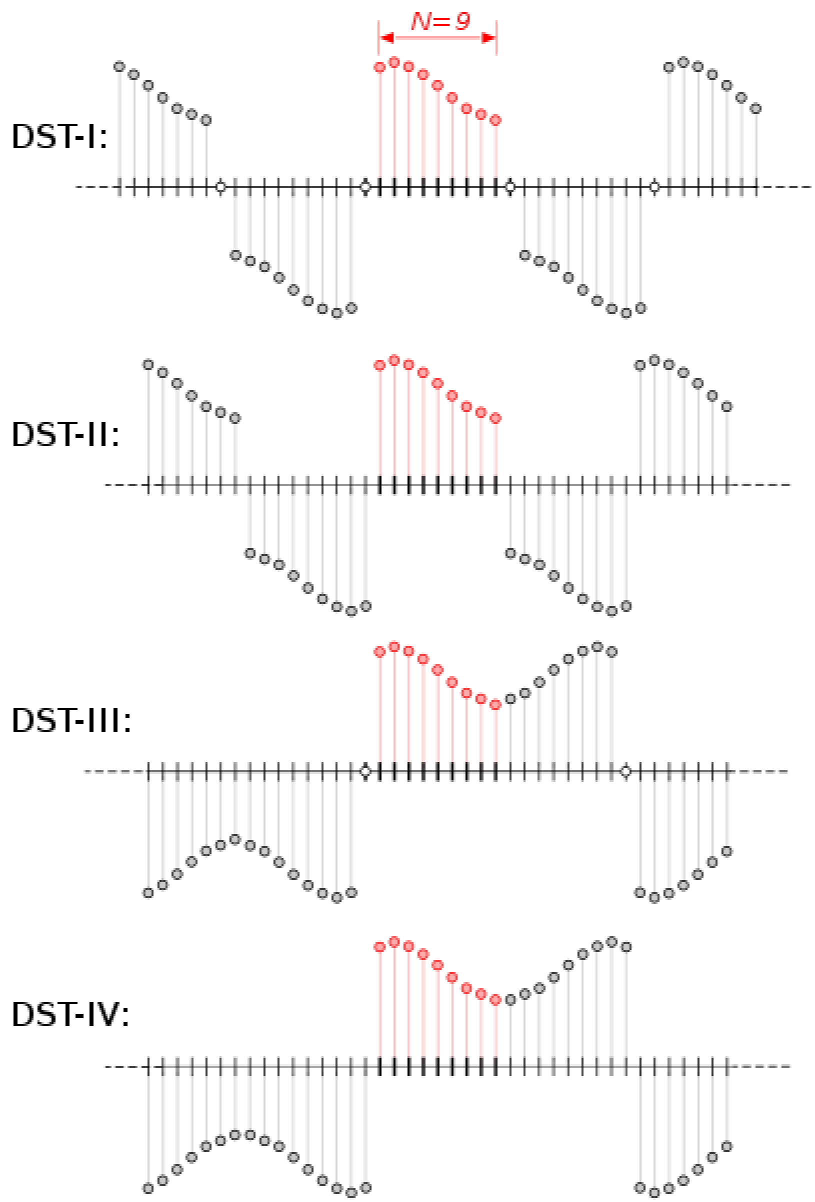


Illustration of the implicit even/odd extensions of DST input data, for N=9 data points (red dots), for the four most common types of DST (types I-IV).

Like any Fourier-related transform, discrete sine transforms (DSTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DST operates on a function at a finite number of discrete data points. The obvious distinction between a DST and a DFT is that the former uses only sine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DST implies different boundary conditions than the DFT or other related transforms.

The Fourier-related transforms that operate on a function over a finite domain, such as the DFT or DST or a Fourier series, can be thought of as implicitly defining an *extension* of that function outside the domain. That is, once you write a function $f(x)$ as a sum of sinusoids, you can evaluate that sum at any x, even for x where the original $f(x)$ was not specified. The DFT, like the Fourier series, implies a periodic extension of the original function. A DST, like a sine transform, implies an odd extension of the original function.

However, because DSTs operate on *finite*, *discrete* sequences, two issues arise that do not apply for the continuous sine transform. First, one has to specify whether the function is even or odd at *both* the left and right boundaries of the domain (i.e. the min-n and max-n boundaries in the definitions below, respectively). Second, one has to specify around *what point* the function is even or odd. In particular, consider a sequence (a,b,c) of three equally spaced data points, and say that we specify an odd *left* boundary. There are two sensible possibilities: either the data is odd about the point *prior* to a, in which case the odd extension is $(-c,-b,-a,0,a,b,c)$, or the data is odd about the point *halfway* between a and the previous point, in which case the odd extension is $(-c,-b,-a,a,b,c)$

These choices lead to all the standard variations of DSTs and also discrete cosine transforms (DCTs). Each boundary can be either even or odd (2 choices per boundary) and can be symmetric about a data point or the point halfway between two data points (2 choices per boundary), for a total of $2 \times 2 \times 2 \times 2 = 16$ possibilities. Half of these possibilities, those where the *left* boundary is odd, correspond to the 8 types of DST; the other half are the 8 types of DCT.

These different boundary conditions strongly affect the applications of the transform, and lead to uniquely useful properties for the various DCT types. Most directly, when using Fourier-related transforms to solve partial differential equations by spectral methods, the boundary conditions are directly specified as a part of the problem being solved.

Definition

Formally, the discrete sine transform is a linear, invertible function $F : \mathbf{R}^N \to \mathbf{R}^N$ (where **R** denotes the set of real numbers), or equivalently an $N \times N$ square matrix. There are several variants of the DST with slightly modified definitions. The N real numbers x_0, ..., x_{N-1} are transformed into the N real numbers X_0, ..., X_{N-1} according to one of the formulas:

DST-I

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N+1}(n+1)(k+1)\right] \qquad k = 0, \dots, N-1$$

The DST-I matrix is orthogonal (up to a scale factor).

A DST-I is exactly equivalent to a DFT of a real sequence that is odd around the zero-th and middle points, scaled by 1/2. For example, a DST-I of N=3 real numbers (a,b,c) is exactly equivalent to a DFT of eight real numbers $(0,a,b,c,0,-c,-b,-a)$ (odd symmetry), scaled by 1/2. (In contrast, DST types II-IV involve a half-sample shift in the equivalent DFT.) This is the reason for the $N+1$ in the denominator of the sine function: the equivalent DFT has $2(N+1)$ points and has $2\pi/2(N+1)$ in its sinusoid frequency, so the DST-I has $\pi/(N+1)$ in its frequency.

Thus, the DST-I corresponds to the boundary conditions: x_n is odd around $n=-1$ and odd around $n=N$; similarly for X_k.

DST-II

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)(k+1)\right] \qquad k = 0, \dots, N-1$$

Some authors further multiply the X_{N-1} term by $1/\sqrt{2}$ (see below for the corresponding change in DST-III). This makes the DST-II matrix orthogonal (up to a scale factor), but breaks the direct correspondence with a real-odd DFT of half-shifted input.

The DST-II implies the boundary conditions: x_n is odd around $n=-1/2$ and odd around $n=N-1/2$; X_k is odd around $k=-1$ and even around $k=N-1$.

DST-III

$$X_k = \frac{(-1)^k}{2} x_{N-1} + \sum_{n=0}^{N-2} x_n \sin\left[\frac{\pi}{N}(n+1)\left(k+\frac{1}{2}\right)\right] \qquad k = 0, \dots, N-1$$

Some authors further multiply the x_{N-1} term by $\sqrt{2}$ (see above for the corresponding change in DST-II). This makes the DST-III matrix orthogonal (up to a scale factor), but breaks the direct correspondence with a real-odd DFT of half-shifted output.

The DST-III implies the boundary conditions: x_n is odd around $n=-1$ and even around $n=N-1$; X_k is odd around $k=-1/2$ and odd around $k=N-1/2$.

DST-IV

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right] \qquad k = 0, \dots, N-1$$

The DST-IV matrix is orthogonal (up to a scale factor).

The DST-IV implies the boundary conditions: x_n is odd around $n=-1/2$ and even around $n=N-1/2$; similarly for X_k.

DST V-VIII

DST types I-IV are equivalent to real-odd DFTs of even order. In principle, there are actually four additional types of discrete sine transform (Martucci, 1994), corresponding to real-odd DFTs of logically odd order, which have factors of $N+1/2$ in the denominators of the sine arguments. However, these variants seem to be rarely used in practice.

Inverse transforms

The inverse of DST-I is DST-I multiplied by $2/(N+1)$. The inverse of DST-IV is DST-IV multiplied by $2/N$. The inverse of DST-II is DST-III multiplied by $2/N$ (and vice versa).

Like for the DFT, the normalization factor in front of these transform definitions is merely a convention and differs between treatments. For example, some authors multiply the transforms by $\sqrt{2/N}$ so that the inverse does not require any additional multiplicative factor.

Computation

Although the direct application of these formulas would require $O(N^2)$ operations, it is possible to compute the same thing with only $O(N \log N)$ complexity by factorizing the computation similar to the fast Fourier transform (FFT). (One can also compute DSTs via FFTs combined with $O(N)$ pre- and post-processing steps.)

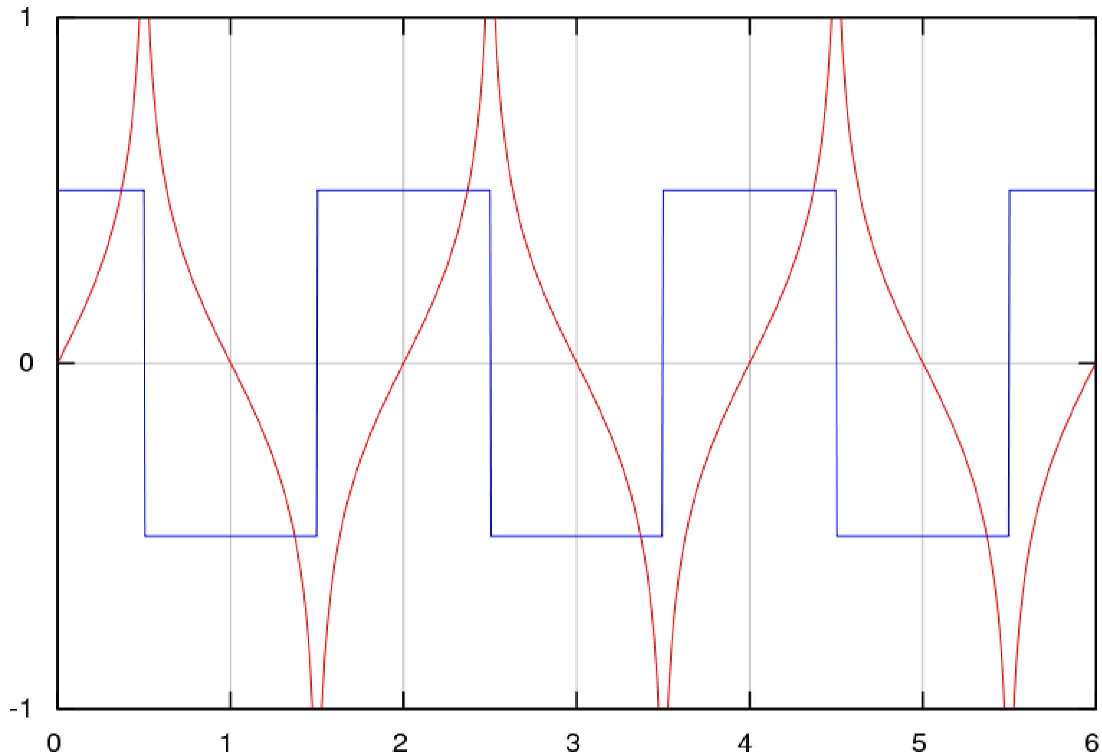
A DST-II or DST-IV can be computed from a DCT-II or DCT-IV, respectively, by reversing the order of the inputs and flipping the sign of every other output, and vice versa for DST-III from DCT-III. In this way it follows that types II–IV of the DST require exactly the same number of arithmetic operations (additions and multiplications) as the corresponding DCT types.

Chapter- 7

Hilbert Transform

In mathematics and in signal processing, the **Hilbert transform** is a linear operator which takes a function, $u(t)$, and produces a function, $H(u)(t)$, with the same domain. The Hilbert transform is named after David Hilbert, who first introduced the operator in order to solve a special case of the Riemann–Hilbert problem for holomorphic functions. It is a basic tool in Fourier analysis, and provides a concrete means for realizing the conjugate of a given function or Fourier series. Furthermore, in harmonic analysis, it is an example of a singular integral operator, and of a Fourier multiplier. The Hilbert transform is also important in the field of signal processing where it is used to derive the analytic representation of a signal $u(t)$.

The Hilbert transform was originally defined for periodic functions, or equivalently for functions on the circle, in which case it is given by convolution with the **Hilbert kernel**. More commonly, however, the Hilbert transform refers to a convolution with the **Cauchy kernel**, for functions defined on the real line **R** (the boundary of the upper half-plane). The Hilbert transform is closely related to the Paley–Wiener theorem, another result relating holomorphic functions in the upper half-plane and Fourier transforms of functions on the real line.



The Hilbert transform, in red, of a square wave, in blue

Introduction

The Hilbert transform can be thought of as the convolution of $u(t)$ with the function $h(t) = 1/(\pi t)$. Because $h(t)$ is not integrable the integrals defining the convolution do not converge. Instead, the Hilbert transform is defined using the Cauchy principal value (denoted here by p.v.) Explicitly, the Hilbert transform of a function (or signal) $u(t)$ is given by

$$H(u)(t) = \text{p.v.} \int_{-\infty}^{\infty} u(\tau)h(t-\tau)\,d\tau = \frac{1}{\pi}\ \text{p.v.} \int_{-\infty}^{\infty} \frac{u(\tau)}{t-\tau}\,d\tau$$

provided this integral exists as a principal value. This is precisely the convolution of u with the tempered distribution p.v. $1/\pi t$ (Schwartz (1950); Pandey (1996, Chapter 3)). Alternatively, by changing variables, the principal value integral can be written explicitly (Zygmund 1968, §XVI.1) as

$$H(u)(t) = -\frac{1}{\pi} \lim_{\varepsilon \downarrow 0} \int_{\varepsilon}^{\infty} \frac{u(t+\tau) - u(t-\tau)}{\tau}\,d\tau.$$

When the Hilbert transform is applied twice in succession to a function *u*, the result is minus *u*:

$$H(H(u))(t) = -u(t),$$

provided the integrals defining both iterations converge in a suitable sense. In particular, the inverse transform is $-H$. This fact can most easily be seen by considering the effect of the Hilbert transform on the Fourier transform of $u(t)$.

For an analytic function in upper half-plane the Hilbert transform describes the relationship between the real part and the imaginary part of the boundary values. That is, if $f(z)$ is analytic in the plane $\operatorname{Im} z > 0$ and $u(t) = \operatorname{Re} f(t + 0 \cdot i)$ then $\operatorname{Im} f(t + 0 \cdot i) = H(u)(t)$ up to an additive constant, provided this Hilbert transform exists.

Notation

In signal processing the Hilbert transform of $u(t)$ is commonly denoted by $\widehat{u}(t)$. However, in mathematics, this notation is already extensively used to denote the Fourier transform of $u(t)$. Occasionally, the Hilbert transform may be denoted by $\tilde{u}(t)$. Furthermore, many sources define the Hilbert transform as the negative of the one defined here.

History

The Hilbert transform arose in Hilbert's 1905 work on a problem posed by Riemann concerning analytic functions (Kress (1983); Bitsadze (2001)) which has come to be known as the Riemann–Hilbert problem. Hilbert's work was mainly concerned with the Hilbert transform for functions defined on the circle (Hilbert 1953). Some of his earlier work related to the Discrete Hilbert Transform date back to lectures he gave in Göttingen. The results were later published by Hermann Weyl in his dissertation (Hardy, Littlewood & Polya 1952). Schur improved Hilbert's results about the discrete Hilbert transform and extended them to the integral case (Hardy, Littlewood & Polya 1952). These results were restricted to the spaces L^2 and ℓ^2. In 1928, Marcel Riesz proved that the Hilbert transform can be defined for u in $L^p(\mathbf{R})$ for $1 < p < \infty$, that the Hilbert transform is a bounded operator on $L^p(\mathbf{R})$ for the same range of p, and that similar results hold for the Hilbert transform on the circle as well as the discrete Hilbert transform (Riesz 1928). The Hilbert transform was a motivating example for Antoni Zygmund and Alberto Calderón during their study of singular integrals (Calderón & Zygmund 1952). Their investigations have played a fundamental role in modern harmonic analysis. Various generalizations of the Hilbert transform, such as the bilinear and trilinear Hilbert transforms are still active areas of research today.

The early 2000s saw the development of Hilbert spectroscopy which uses Hilbert transforms to detect signatures of chemical mixtures by analyzing broad spectrum signals from gigahertz to terahertz frequency radio.

Relationship with the Fourier transform

As mentioned before, the Hilbert transform is a multiplier operator. The symbol of H is $\sigma_H(\omega) = -i\ \mathrm{sgn}(\omega)$ where sgn is the signum function. Therefore**:**

$$\mathcal{F}(H(u))(\omega) = (-i\,\mathrm{sgn}(\omega)) \cdot \mathcal{F}(u)(\omega)$$

where $\mathcal{F}$ denotes the Fourier transform. Since $\mathrm{sgn}(x) = \mathrm{sgn}(2\pi x)$, it follows that this result applies to the three common definitions of $\mathcal{F}$.

By Euler's formula,

$$\sigma_H(\omega) \quad = \begin{cases} \;\; i = e^{+i\pi/2}, & \text{for } \omega < 0 \\ \quad 0, & \text{for } \omega = 0 \\ -i = e^{-i\pi/2}. & \text{for } \omega > 0. \end{cases}$$

Therefore $H(u)(t)$ has the effect of shifting the phase of the negative frequency components of $u(t)$ by +90° (π/2 radians) and the phase of the positive frequency components by −90°. And $i{\cdot}H(u)(t)$ has the effect of restoring the positive frequency components while shifting the negative frequency ones an additional +90°, resulting in their negation.

When the Hilbert transform is applied twice the phase of the negative and positive frequency components of $u(t)$ are respectively shifted by +180° and −180°, which are equivalent amounts. The signal is negated, i.e., $H(H(u)) = -u$, because**:**

$$[\sigma_H(\omega)]^2 = e^{\pm i\pi} = -1.$$

Table of selected Hilbert transforms

Signal	Hilbert transform*
$u(t)$	$H(u)(t)$
$\sin(t)$**	$\cos(t)$
$\cos(t)$**	$-\sin(t)$
$\dfrac{1}{t^2+1}$	$\dfrac{t}{t^2+1}$
Sinc function $\dfrac{\sin(t)}{t}$	$\dfrac{1-\cos(t)}{t}$

Rectangular function $\sqcap(t)$	$\frac{1}{\pi}\ln\left\|\frac{t+\frac{1}{2}}{t-\frac{1}{2}}\right\|$
Dirac delta function $\delta(t)$	$\frac{1}{\pi t}$
Characteristic Function $\chi_{[a,b]}(x)$	$\frac{1}{\pi}\log\left\|\frac{x-a}{x-b}\right\|$

Notes

* Some authors, e.g., Bracewell, use our −*H* as their definition of the forward transform. A consequence is that the right column of this table would be negated.

** The Hilbert transform of the *sin* and *cos* functions can be defined in a distributional sense, if there is a concern that the integral defining them is otherwise conditionally convergent. In the periodic setting this result holds without any difficulty.

An extensive table of Hilbert transforms is available (King 2009). Note that the Hilbert transform of a constant is zero.

Domain of definition

It is by no means obvious that the Hilbert transform is well-defined at all, as the improper integral defining it must converge in a suitable sense. However, the Hilbert transform is well-defined for a broad class of functions, namely those in $L^p(\mathbf{R})$ for $1<p<\infty$.

More precisely, if *u* is in $L^p(\mathbf{R})$ for $1<p<\infty$, then the limit defining the improper integral

$$H(u)(t) = -\frac{1}{\pi}\lim_{\epsilon\downarrow 0}\int_{\epsilon}^{\infty}\frac{u(t+\tau)-u(t-\tau)}{\tau}\,d\tau$$

exists for almost every *t*. The limit function is also in $L^p(\mathbf{R})$, and is in fact the limit in the mean of the improper integral as well. That is,

$$-\frac{1}{\pi}\int_{\epsilon}^{\infty}\frac{u(t+\tau)-u(t-\tau)}{\tau}\,d\tau \to H(u)(t)$$

as $\varepsilon\to 0$ in the L^p-norm, as well as pointwise almost everywhere, by the Titchmarsh theorem (Titchmarsh 1948, Chapter 5).

In the case $p=1$, the Hilbert transform still converges pointwise almost everywhere, but may fail to be itself integrable even locally (Titchmarsh 1948, §5.14). In particular, convergence in the mean does not in general happen in this case. The Hilbert transform of an L^1 function does converge, however, in L^1-weak, and the Hilbert transform is a bounded operator from L^1 to $L^{1,w}$ (Stein & Weiss 1971, Lemma V.2.8). (In particular, since the Hilbert transform is also a multiplier operator on L^2, Marcinkiewicz interpolation and a duality argument furnishes an alternative proof that H is bounded on L^p.)

Properties

Boundedness

If $1<p<\infty$, then the Hilbert transform on $L^p(\mathbf{R})$ is a bounded linear operator, meaning that there exists a constant C_p such that

$$\|Hu\|_p \le C_p \|u\|_p$$

for all $u \in L^p(\mathbf{R})$. This theorem is due to Riesz (1928, VII); (1948, Theorem 101). The best constant C_p is given by

$$C_p = \begin{cases} \tan\frac{\pi}{2p} & \text{for } 1 < p \le 2 \\ \cot\frac{\pi}{2p} & \text{for } 2 < p < \infty. \end{cases}$$

This result is due to (Pichorides 1972);. The same best constants hold for the periodic Hilbert transform.

Anti-self adjointness

The Hilbert transform is an anti-self adjoint operator relative to the duality pairing between $L^p(\mathbf{R})$ and the dual space $L^q(\mathbf{R})$, where p and q are Hölder conjugates and $1 < p,q < \infty$. Symbolically,

$$\langle Hu, v\rangle = \langle u, -Hv\rangle$$

for $u \in L^p(\mathbf{R})$ and $v \in L^q(\mathbf{R})$ (Titchmarsh 1948, Theorem 102).

Inverse transform

The Hilbert transform is an anti-involution, meaning that

$$H(H(u)) = -u$$

provided each transform is well-defined. Since H preserves the space $L^p(\mathbf{R})$, this implies in particular that the Hilbert transform is invertible on $L^p(\mathbf{R})$, and that

$$H^{-1} = -H.$$

Differentiation

Formally, the derivative of the Hilbert transform is the Hilbert transform of the derivative:

$$H\left(\frac{du}{dt}\right) = \frac{d}{dt}H(u).$$

Iterating this identity,

$$H\left(\frac{d^k u}{dt^k}\right) = \frac{d^k}{dt^k}H(u).$$

This is rigorously true as stated provided *u* and its first *k* derivatives belong to $L^p(\mathbf{R})$ (Pandey 1996, §3.3).

Convolutions

The Hilbert transform can formally be realized as a convolution with the tempered distribution

$$h(t) = \text{p.v. } \frac{1}{\pi t}.$$

Thus formally,

$$H(u) = h * u.$$

However, *a priori* this may only be defined for *u* a distribution of compact support. It is possible to work somewhat rigorously with this since compactly supported functions (which are distributions *a fortiori*) are dense in L^p. Alternatively, one may use the fact that $h(t)$ is the distributional derivative of the function $\log|t|/\pi$; to wit

$$H(u)(t) = \frac{d}{dt}\left(\frac{1}{\pi}(u * \log|\cdot|)(t)\right).$$

For most operational purposes the Hilbert transform can be treated as a convolution. For example, in a formal sense, the Hilbert transform of a convolution is the convolution of the Hilbert transform on either factor:

$$H(u * v) = H(u) * v = u * H(v).$$

This is rigorously true if *u* and *v* are compactly supported distributions since, in that case,

$$h * (u * v) = (h * u) * v = u * (h * v).$$

By passing to an appropriate limit, it is thus also true if $u \in L^p$ and $v \in L^r$ provided

$$1 < \frac{1}{p} + \frac{1}{r},$$

a theorem due to Titchmarsh (1948, Theorem 104).

Invariance

The Hilbert transform has the following invariance properties.

- It commutes with translations. That is, it commutes with the operators $T_a f(x) = f(x + a)$ for all a in $\mathbf{R}^n$
- It commutes with positive dilations. That is it commutes with the operators $M_\lambda f(x) = f(\lambda x)$ for all $\lambda > 0$.
- It anticommutes with the reflection $Rf(x) = f(-x)$.

Up to a multiplicative constant, the Hilbert transform is the only L^2 bounded operator with these properties (Stein 1970, §III.1).

Extending the domain of definition

Hilbert transform of distributions

It is further possible to extend the Hilbert transform to certain spaces of distributions (Pandey 1996, Chapter 3). Since the Hilbert transform commutes with differentiation, and is a bounded operator on L^p, H restricts to give a continuous transform on the inverse limit of Sobolev spaces:

$$\mathcal{D}_{L^p} = \varprojlim_{n\to\infty} W^{n,p}(\mathbb{R}).$$

The Hilbert transform can then be defined on the dual space of $\mathcal{D}_{L^p}$, denoted $\mathcal{D}'_{L^p}$, consisting of L^p distributions. This is accomplished by the duality pairing: for $u \in \mathcal{D}'_{L^p}$, define $H(u) \in \mathcal{D}'_{L^p}$by

$$\langle Hu, v\rangle \overset{\text{def}}{=} \langle u, -Hv\rangle$$

for all $v \in \mathcal{D}_{L^p}$.

It is possible to define the Hilbert transform on the space of tempered distributions as well by an approach due to Gel'fand & Shilov (1967), but considerably more care is needed because of the singularity in the integral.

Hilbert transform of bounded functions

The Hilbert transform can be defined for functions in $L^\infty(\mathbf{R})$ as well, but it requires some modifications and caveats. Properly understood, the Hilbert transform maps $L^\infty(\mathbf{R})$ to the Banach space of bounded mean oscillation (BMO) classes.

Interpreted naively, the Hilbert transform of a bounded function is clearly ill-defined. For instance, with $u = \operatorname{sgn}(x)$, the integral defining $H(u)$ diverges almost everywhere to $\pm\infty$. To alleviate such difficulties, the Hilbert transform of an L^∞-function is therefore defined by the following regularized form of the integral

$$H(u)(t) = \text{p.v.} \int_{-\infty}^{\infty} u(\tau) \left\{h(t-\tau) - h_0(-\tau)\right\} \, d\tau$$

where as above $h(x) = 1/\pi x$ and

$$h_0(x) = \begin{cases} 0 & \text{if } |x| < 1 \\ \frac{1}{\pi x} & \text{otherwise} \end{cases}$$

The modified transform H agrees with the original transform on functions of compact support by a general result of Calderón & Zygmund (1952). The resulting integral, furthermore, converges pointwise almost everywhere, and with respect to the BMO norm, to a function of bounded mean oscillation.

A deep result of Fefferman (1971) and Fefferman & Stein (1972) is that a function is of bounded mean oscillation if and only if it has the form $f + H(g)$ for some $f, g \in L^\infty(\mathbf{R})$.

Conjugate functions

The Hilbert transform can be understood in terms of a pair of functions $f(x)$ and $g(x)$ such that the function

$$F(x) = f(x) + ig(x)$$

is the boundary value of a holomorphic function $F(z)$ in the upper half-plane. Under these circumstances, if f and g are sufficiently integrable, then one is the Hilbert transform of the other.

Suppose that $f \in L^p(\mathbf{R})$. Then, by the theory of the Poisson integral, f admits a unique harmonic extension into the upper half-plane, and this extension is given by

$$u(x+iy) = u(x,y) = \frac{1}{\pi}\int_{-\infty}^{\infty} f(s)\frac{y}{(x-s)^2+y^2}\,ds$$

which is the convolution of *f* with the Poisson kernel

$$P(x,y) = \frac{1}{\pi}\frac{y}{x^2+y^2}.$$

Furthermore, there is a unique harmonic function *v* defined in the upper half-plane such that *F*(*z*) = *u*(*z*) + i*v*(*z*) is holomorphic and

$$\lim_{y\to\infty} v(x+iy) = 0.$$

This harmonic function is obtained from *f* by taking a convolution with the **conjugate Poisson kernel**

$$Q(x,y) = \frac{1}{\pi}\frac{x}{x^2+y^2}.$$

Thus

$$v(x,y) = \frac{1}{\pi}\int_{-\infty}^{\infty} f(s)\frac{x-s}{(x-s)^2+y^2}\,ds.$$

Indeed, the real and imaginary parts of the Cauchy kernel are

$$\frac{i}{\pi z} = P(x,y) + iQ(x,y),$$

so that *F* = *u* + i*v* is holomorphic by Cauchy's theorem.

The function *v* obtained from *u* in this way is called the harmonic conjugate of *u*. The (non-tangential) boundary limit of *v*(*x*,*y*) as $y \to 0$ is the Hilbert transform of *f*. Thus, succinctly,

$$H(f) = \lim_{y\to 0} Q(-,y) \star f.$$

Titchmarsh's theorem

A theorem due to Edward Charles Titchmarsh makes precise the relationship between the boundary values of holomorphic functions in the upper half-plane and the Hilbert transform (Titchmarsh 1948, Theorem 95). It gives necessary and sufficient conditions for a complex-valued square-integrable function *F*(*x*) on the real line to be the boundary value of a function in the Hardy space $H^2(U)$ of holomorphic functions in the upper half-plane U.

The theorem states that the following conditions for a complex-valued square-integrable function $F : \mathbf{R} \to \mathbf{C}$ are equivalent:

- $F(x)$ is the limit as $z \to x$ of a holomorphic function $F(z)$ in the upper half-plane such that

$$\int_{-\infty}^{\infty} |F(x+iy)|^2 \, dx < K.$$

- $\mathrm{Im}(F)$ is the Hilbert transform of $\mathrm{Re}(F)$, where $\mathrm{Re}(F)$ and $\mathrm{Im}(F)$ are real-valued functions with $F = \mathrm{Re}(F) + i\,\mathrm{Im}(F)$.

- The Fourier transform $\mathcal{F}(F)(x)$ vanishes for $x < 0$.

A weaker result is true for functions of class L^p for $p > 1$ (Titchmarsh 1948, Theorem 103). Specifically, if $F(z)$ is a holomorphic function such that

$$\int_{-\infty}^{\infty} |F(x+iy)|^p \, dx < K$$

for all y, then there is a complex-valued function $F(x)$ in $L^p(\mathbf{R})$ such that $F(x + iy) \to F(x)$ in the L^p norm as $y \to 0$ (as well as holding pointwise almost everywhere). Furthermore,

$$F(x) = f(x) - ig(x)$$

where f is a real-valued function in $L^p(\mathbf{R})$ and g is the Hilbert transform (of class L^p) of f.

This is not true in the case $p = 1$. In fact, the Hilbert transform of an L^1 function f need not converge in the mean to another L^1 function. Nevertheless (Titchmarsh 1948, Theorem 105), the Hilbert transform of f does converge almost everywhere to a finite function g such that

$$\int_{-\infty}^{\infty} \frac{|g(x)|^p}{1+x^2} \, dx < \infty.$$

This result is directly analogous to one by Andrey Kolmogorov for Hardy functions in the disc (Duren 1970, Theorem 4.2).

Riemann–Hilbert problem

One form of the Riemann–Hilbert problem seeks to identify pairs of functions F_+ and F_- such that F_+ is holomorphic on the upper half-plane and F_- is holomorphic on the lower half-plane, such that for x along the real axis,

$$F_+(x) - F_-(x) = f(x)$$

where $f(x)$ is some given real-valued function of $x \in \mathbf{R}$. The left-hand side of this equation may be understood either as the difference of the limits of $F_\pm$ from the appropriate half-planes, or as a hyperfunction distribution. Two functions of this form are a solution of the Riemann–Hilbert problem.

Formally, if $F_\pm$ solve the Riemann–Hilbert problem

$$f(x) = F_+(x) - F_-(x),$$

then the Hilbert transform of $f(x)$ is given by

$$H(f)(x) = \frac{1}{i}(F_+(x) + F_-(x)).$$

Hilbert transform on the circle

For a periodic function f the circular Hilbert transform is defined as

$$\tilde{f}(x) = \frac{1}{2\pi}\ \text{p.v.} \int_0^{2\pi} f(t) \cot \frac{x-t}{2}\, dt.$$

The circular Hilbert transform is used in giving a characterization of Hardy space and in the study of the conjugate function in Fourier series. The kernel $\cot \frac{x-t}{2}$ is known as the **Hilbert kernel** since it was in this form the Hilbert transform was originally studied (Khvedelidze 2001).

The Hilbert kernel (for the circular Hilbert transform) can be obtained by making the Cauchy kernel $1/x$ periodic. More precisely, for $x \neq 0$

$$\frac{1}{2}\cot\left(\frac{x}{2}\right) = \frac{1}{x} + \sum_{n=1}^{\infty} \frac{1}{x+2n\pi} - \frac{1}{2n\pi}.$$

Many results about the circular Hilbert transform may be derived from the corresponding results for the Hilbert transform from this correspondence.

Hilbert transform in signal processing

Narrowband model

Bedrosian's theorem states that the Hilbert transform of the product of a low-pass and a high-pass signal with non-overlapping spectra is given by the product of the low-pass signal and the Hilbert transform of the high-pass signal, or

$$H(f_{LP}(t)f_{HP}(t)) = f_{LP}(t)H(f_{HP}(t))$$

where f_{LP} and f_{HP} are the low- and high-pass signals respectively.

Amplitude modulated signals are modeled as the product of a bandlimited "message" waveform, $u_m(t)$, and a sinusoidal "carrier":

$$u(t) = u_m(t) \cdot \cos(\omega t + \phi)$$

When $u_m(t)$ has no frequency content above the carrier frequency, $\frac{\omega}{2\pi}$ Hz, then by Bedrosian's theorem:

$$\begin{aligned}\widehat{u}(t) &\stackrel{\text{def}}{=} H(u)(t) \\ &= u_m(t) \cdot \sin(\omega t + \phi)\end{aligned}$$

So, the Hilbert transform may be as simple as a circuit that produces a 90° phase shift at the carrier frequency. Furthermore:

$$\begin{aligned}(\omega t + \phi)_{\text{mod } 2\pi} &= \text{atan2}(\ \widehat{u}(t),\ u(t)\ ) \\ &= \arg(u_a(t))\end{aligned}$$

from which one can reconstruct the carrier waveform. Then the message can be extracted from $u(t)$ by coherent demodulation.

Analytic representation

The analytic representation of a signal is defined in terms of the Hilbert transform:

$$u_a(t) = u(t) + i \cdot \widehat{u}(t).$$

For the narrowband model [above], the analytic representation is:

$$u_a(t) = u_m(t) \cdot \cos(\omega t + \phi) + i \cdot u_m(t) \cdot \sin(\omega t + \phi)$$

$$= u_m(t) \cdot [\cos(\omega t + \phi) + i \cdot \sin(\omega t + \phi)]$$

$$= u_m(t) \cdot e^{i(\omega t+\phi)}$$ (by Euler's formula) **(*Eq.1*)**

This complex heterodyne operation shifts all the frequency components of $u_m(t)$ above 0 Hz. In that case, the imaginary part of the result is a Hilbert transform of the real part. This is an indirect way to produce Hilbert transforms.

While the analytic representation of a signal is not necessarily an analytic function, $u_a(t)$ is given by the boundary values of an analytic function in the upper half-plane.

Phase/Frequency modulation

The form**:**

$$u(t) = A \cdot \cos(\omega t + \phi_m(t))$$

is called phase (or frequency) modulation. The instantaneous frequency is $\omega + \phi_m'(t).$ For sufficiently large ω, compared to ϕ_m'**:**

$$\widehat{u}(t) \approx A \cdot \sin(\omega t + \phi_m(t)),$$

and**:**

$$u_a(t) \approx A \cdot e^{i(\omega t + \phi_m(t))}.$$

Single sideband modulation (SSB)

When $u_m(t)$ in **Eq.1** is also an analytic representation (of a message waveform), that is**:**

$$u_m(t) = m(t) + i \cdot \widehat{m}(t),$$

the result is single-sideband modulation**:**

$$u_a(t) = (m(t) + i \cdot \widehat{m}(t)) \cdot e^{i(\omega t+\phi)},$$

whose transmitted component is**:**

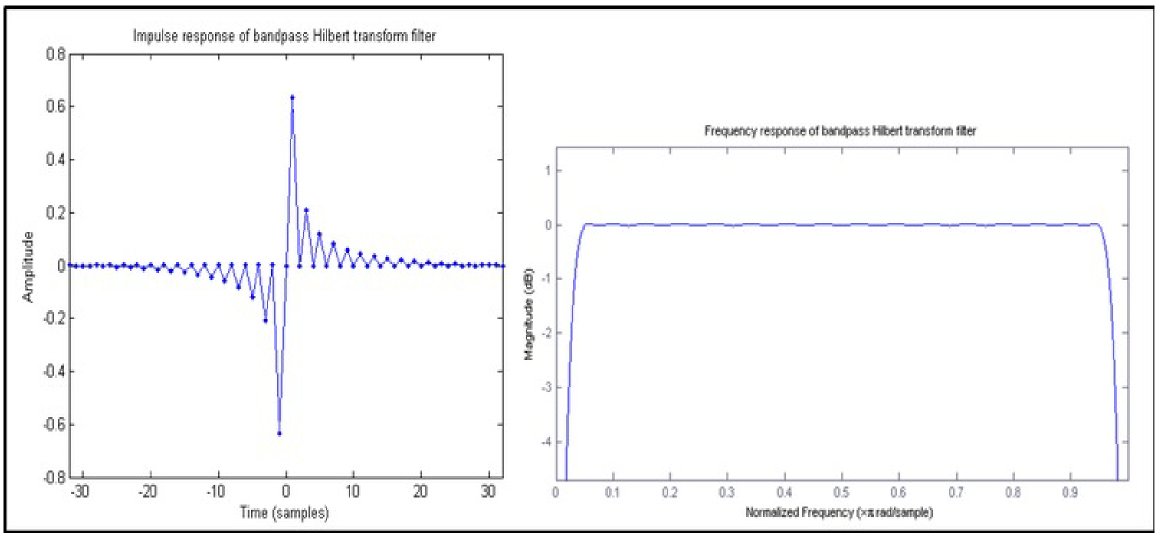
$$u(t) = \mathrm{Re}\{u_a(t)\}$$
$$= m(t) \cdot \cos(\omega t + \phi) - \widehat{m}(t) \cdot \sin(\omega t + \phi).$$

Causality

The function h with $h(t) = 1/(\pi\, t)$ is a non-causal filter and therefore cannot be implemented as is, if u is a time-dependent signal. If u is a function of a non-temporal variable, e.g., spatial, the non-causality might not be a problem. The filter is also of infinite support which may be a problem in certain applications. Another issue relates to what happens with the zero frequency (DC), which can be avoided by assuring that s does not contain a DC-component.

A practical implementation in many cases implies that a finite support filter, which in addition is made causal by means of a suitable delay, is used to approximate the computation. The approximation may also imply that only a specific frequency range is subject to the characteristic phase shift related to the Hilbert transform.

Discrete Hilbert transforms



Filter whose frequency response is bandlimited to about 95% of the Nyquist frequency

There are two objects of study which are considered discrete Hilbert transforms. The Discrete Hilbert transform of practical interest is usually described in terms of the following bandlimited transfer function**:**

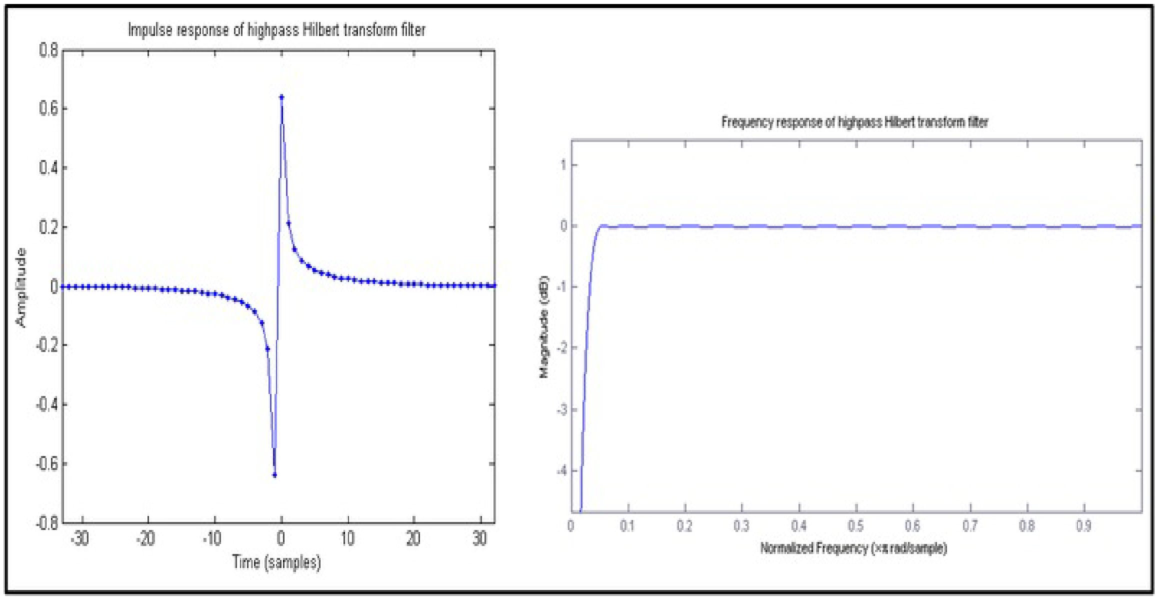
$$\sigma_H(\omega) = \begin{cases} e^{+i\pi/2}, & -\pi < \omega < 0 \\ e^{-i\pi/2}, & 0 < \omega < \pi \\ 0, & \omega = -\pi, 0, \pi, \end{cases}$$

which is the discrete-time Fourier transform of the infinite sequence**:**

$$h[n] = \begin{cases} 0, & \text{for } n \text{ even,} \\ \frac{2}{\pi n} & \text{for } n \text{ odd.} \end{cases}$$

If a signal $u(t)$ is bandlimited, then $H(u)(t)$ is bandlimited in the same way. Consequently, both these signals can be sampled according to the sampling theorem, resulting in the discrete signals $u[n]$ and $H(u)[n]$. The relation between the two discrete signals is then given by the convolution**:**

$$H(u)[n] = h[n] * u[n] \; .$$



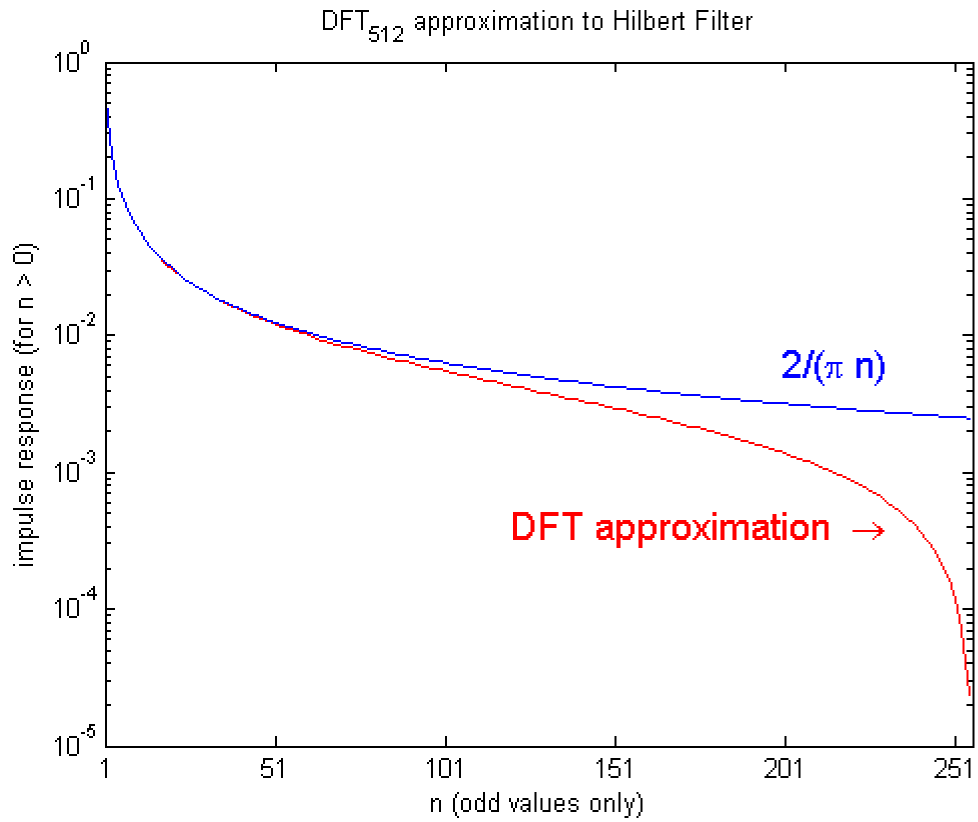
Hilbert transform filter with a highpass frequency response

When an FIR approximation is substituted for $h[n]$, we see rolloff of the passband at the low and high ends (0 and Nyquist), resulting in a bandpass filter as shown in the figure above. The high end can be restored by an FIR that more closely resembles samples of the smooth, continuous-time $h(t)$, as shown in the next figure. But as a practical matter, a properly-sampled $u[n]$ sequence has no useful components at those frequencies.

As the impulse response gets longer, the low end frequencies are also restored. Hilbert studied the transform **:**

$$H(u)[n] = \frac{1}{n} * u[n] = \sum_{m=-\infty}^{\infty} \frac{u(m)}{n-m} \qquad m \neq n,$$

and showed that for $u(n)$ in ℓ^2 the sequence $H(u)[n]$ is also in ℓ^2. An elementary proof of this fact can be found in (Grafakos 1994). This discrete Hilbert transform was used by E. C. Titchmarsh to give alternate proofs of the results of M. Riesz in the continuous case (Titchmarsh 1926; Hardy, Littlewood & Polya 1952).



We also note that a sequence similar to $h[n]$ can be generated by sampling $\sigma_H(\omega)$ and computing the inverse discrete Fourier transform. The larger the transform (i.e., more samples per 2π radians), the better the agreement (for a given value of the abscissa, n). The figure shows the comparison for a 512-point transform. (Due to odd-symmetry, only half the sequence is actually plotted.)

But that is not the actual point, because it is easier and more accurate to generate $h[n]$ directly from the formula. The point is that many applications choose to avoid the convolution by doing the equivalent frequency-domain operation**:** simple multiplication of the signal transform with $\sigma_H(\omega)$, made even easier by the fact that the real and imaginary components are 0 and ±1 respectively. The attractiveness of that approach is only apparent when the actual Fourier transforms are replaced by samples of the same, i.e., the DFT, which is an approximation and introduces some distortion. Thus, after transforming back to the time-domain, those applications have indirectly generated (and convolved with) *not* $h[n]$, but the DFT approximation to it, which is shown in the figure.

Notes on *fast convolution***:**

- Implied in the technique described above is the concept of dividing a long signal into segments of arbitrary size. The signal is filtered piecewise, and the outputs are subsequently pieced back together.

- The segment size is an important factor in controlling the amount of distortion. As the size increases, the DFT becomes denser and is a better approximation to the underlying Fourier transform. In the time-domain, the same distortion is manifested as "aliasing", which results in a type of convolution called circular. It is as if the same segment is repeated periodically and filtered, resulting in distortion that is worst at either or both edges of the original segment. Increasing the segment size reduces the number of edges in the pieced-together result and therefore reduces overall distortion.
- Another mitigation strategy is to simply discard the most badly distorted output samples, because data loss can be avoided by overlapping the input segments. When the filter's impulse response is less than the segment length, this can produce a distortion-free (non-circular) convolution (Overlap-discard method). That requires an FIR filter, which the Hilbert transform is not. So yet another technique is to design an FIR approximation to a Hilbert transform filter. That moves the source of distortion from the convolution to the filter, where it can be readily characterized in terms of imperfections in the frequency response.
- Failure to appreciate or correctly apply these concepts is probably one of the most common mistakes made by non-experts in the digital signal processing field.

Chapter- 8

Laplace Transform

In mathematics, the **Laplace transform** is a widely used integral transform. Denoted $\mathcal{L}\left\{f(t)\right\}$, it is a linear operator of a function *f*(*t*) with a real argument *t* ($t \geq 0$) that transforms it to a function *F*(*s*) with a complex argument *s*. This transformation is essentially bijective for the majority of practical uses; the respective pairs of *f*(*t*) and *F*(*s*) are matched in tables. The Laplace transform has the useful property that many relationships and operations over the originals *f*(*t*) correspond to simpler relationships and operations over the images *F*(*s*). The Laplace transform has many important applications throughout the sciences. It is named for Pierre-Simon Laplace who introduced the transform in his work on probability theory.

The Laplace transform is related to the Fourier transform, but whereas the Fourier transform resolves a function or signal into its modes of vibration, the Laplace transform resolves a function into its moments. Like the Fourier transform, the Laplace transform is used for solving differential and integral equations. In physics and engineering, it is used for analysis of linear time-invariant systems such as electrical circuits, harmonic oscillators, optical devices, and mechanical systems. In this analysis, the Laplace transform is often interpreted as a transformation from the *time-domain*, in which inputs and outputs are functions of time, to the *frequency-domain*, where the same inputs and outputs are functions of complex angular frequency, in radians per unit time. Given a simple mathematical or functional description of an input or output to a system, the Laplace transform provides an alternative functional description that often simplifies the process of analyzing the behavior of the system, or in synthesizing a new system based on a set of specifications.

History

The Laplace transform is named in honor of mathematician and astronomer Pierre-Simon Laplace, who used the transform in his work on probability theory. From 1744, Leonhard Euler investigated integrals of the form

$$z = \int X(x)e^{ax}\,dx \quad \text{and} \quad z = \int X(x)x^A\,dx$$

as solutions of differential equations but did not pursue the matter very far. Joseph Louis Lagrange was an admirer of Euler and, in his work on integrating probability density functions, investigated expressions of the form

$$\int X(x)e^{-ax}a^x\,dx,$$

which some modern historians have interpreted within modern Laplace transform theory.

These types of integrals seem first to have attracted Laplace's attention in 1782 where he was following in the spirit of Euler in using the integrals themselves as solutions of equations. However, in 1785, Laplace took the critical step forward when, rather than just looking for a solution in the form of an integral, he started to apply the transforms in the sense that was later to become popular. He used an integral of the form:

$$\int x^s\phi(s)\,dx,$$

akin to a Mellin transform, to transform the whole of a difference equation, in order to look for solutions of the transformed equation. He then went on to apply the Laplace transform in the same way and started to derive some of its properties, beginning to appreciate its potential power.

Laplace also recognised that Joseph Fourier's method of Fourier series for solving the diffusion equation could only apply to a limited region of space as the solutions were periodic. In 1809, Laplace applied his transform to find solutions that diffused indefinitely in space.

Formal definition

The Laplace transform of a function $f(t)$, defined for all real numbers $t \geq 0$, is the function $F(s)$, defined by:

$$F(s) = \mathcal{L}\left\{f(t)\right\} = \int_0^\infty e^{-st} f(t)\,dt.$$

The parameter s is a complex number:

$s = \sigma + i\omega$, with real numbers σ and ω.

The meaning of the integral depends on types of functions of interest. A necessary condition for existence of the integral is that f must be locally integrable on $[0,\infty)$. For locally integrable functions that decay at infinity or are of exponential type, the integral can be understood as a (proper) Lebesgue integral. However, for many applications it is necessary to regard it as a conditionally convergent improper integral at ∞. Still more generally, the integral can be understood in a weak sense, and this is dealt with below.

One can define the Laplace transform of a finite Borel measure μ by the Lebesgue integral

$$(\mathcal{L}\mu)(s) = \int_{[0,\infty)} e^{-st} d\mu(t).$$

An important special case is where μ is a probability measure or, even more specifically, the Dirac delta function. In operational calculus, the Laplace transform of a measure is often treated as though the measure came from a distribution function *f*. In that case, to avoid potential confusion, one often writes

$$(\mathcal{L}f)(s) = \int_{0^-}^{\infty} e^{-st} f(t)\, dt$$

where the lower limit of 0^- is short notation to mean

$$\lim_{\varepsilon \to 0^+} \int_{-\varepsilon}^{\infty} .$$

This limit emphasizes that any point mass located at 0 is entirely captured by the Laplace transform. Although with the Lebesgue integral, it is not necessary to take such a limit, it does appear more naturally in connection with the Laplace–Stieltjes transform.

Probability theory

In pure and applied probability, the Laplace transform is defined by means of an expectation value. If *X* is a random variable with probability density function *f*, then the Laplace transform of *f* is given by the expectation

$$(\mathcal{L}f)(s) = E\left[e^{-sX}\right].$$

By abuse of language, this is referred to as the Laplace transform of the random variable *X* itself. Replacing *s* by −*t* gives the moment generating function of *X*. The Laplace transform has applications throughout probability theory, including first passage times of stochastic processes such as Markov chains, and renewal theory.

Bilateral Laplace transform

When one says "the Laplace transform" without qualification, the unilateral or one-sided transform is normally intended. The Laplace transform can be alternatively defined as the *bilateral Laplace transform* or two-sided Laplace transform by extending the limits of integration to be the entire real axis. If that is done the common unilateral transform simply becomes a special case of the bilateral transform where the definition of the function being transformed is multiplied by the Heaviside step function.

The bilateral Laplace transform is defined as follows:

$$F(s) = \mathcal{L}\left\{f(t)\right\} = \int_{-\infty}^{\infty} e^{-st} f(t)\,dt.$$

Inverse Laplace transform

The inverse Laplace transform is given by the following complex integral, which is known by various names (the **Bromwich integral**, the **Fourier-Mellin integral**, and **Mellin's inverse formula**):

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\gamma - iT}^{\gamma + iT} e^{st} F(s)\,ds,$$

where γ is a real number so that the contour path of integration is in the *region of convergence* of $F(s)$. An alternative formula for the inverse Laplace transform is given by Post's inversion formula.

Region of convergence

If f is a locally integrable function (or more generally a Borel measure locally of bounded variation), then the Laplace transform $F(s)$ of f converges provided that the limit

$$\lim_{R\to\infty} \int_0^R f(t) e^{-ts}\,dt$$

exists. The Laplace transform converges absolutely if the integral

$$\int_0^\infty \left|f(t) e^{-ts}\right|\,dt$$

exists (as proper Lebesgue integral). The Laplace transform is usually understood as conditionally convergent, meaning that it converges in the former instead of the latter sense.

The set of values for which $F(s)$ converges absolutely is either of the form $\mathrm{Re}\{s\} > a$ or else $\mathrm{Re}\{s\} \geq a$, where a is an extended real constant, $-\infty \leq a \leq \infty$. (This follows from the dominated convergence theorem.) The constant a is known as the abscissa of absolute convergence, and depends on the growth behavior of $f(t)$. Analogously, the two-sided transform converges absolutely in a strip of the form $a < \mathrm{Re}\{s\} < b$, and possibly including the lines $\mathrm{Re}\{s\} = a$ or $\mathrm{Re}\{s\} = b$. The subset of values of s for which the Laplace transform converges absolutely is called the region of absolute convergence or the domain of absolute convergence. In the two-sided case, it is sometimes called the strip of absolute convergence. The Laplace transform is analytic in the region of absolute convergence.

Similarly, the set of values for which $F(s)$ converges (conditionally or absolutely) is known as the region of conditional convergence, or simply the **region of convergence** (ROC). If the Laplace transform converges (conditionally) at $s = s_0$, then it automatically converges for all s with $\mathrm{Re}\{s\} > \mathrm{Re}\{s_0\}$. Therefore the region of convergence is a half-plane of the form $\mathrm{Re}\{s\} > a$, possibly including some points of the boundary line $\mathrm{Re}\{s\} = a$. In the region of convergence $\mathrm{Re}\{s\} > \mathrm{Re}\{s_0\}$, the Laplace transform of f can be expressed by integrating by parts as the integral

$$F(s) = (s - s_0) \int_0^\infty e^{-(s-s_0)t} \beta(t)\, dt, \quad \beta(u) = \int_0^u e^{-s_0 t} f(t)\, dt.$$

That is, in the region of convergence $F(s)$ can effectively be expressed as the absolutely convergent Laplace transform of some other function. In particular, it is analytic.

A variety of theorems, in the form of Paley–Wiener theorems, exist concerning the relationship between the decay properties of f and the properties of the Laplace transform within the region of convergence.

In engineering applications, a function corresponding to a linear time-invariant (LTI) system is *stable* if every bounded input produces a bounded output. This is equivalent to the absolute convergence of the Laplace transform of the impulse response function in the region $\mathrm{Re}\{s\} \geq 0$. As a result, LTI systems are stable provided the poles of the Laplace transform of the impulse response function have negative real part.

Properties and theorems

The Laplace transform has a number of properties that make it useful for analyzing linear dynamical systems. The most significant advantage is that differentiation and integration become multiplication and division, respectively, by s (similarly to logarithms changing multiplication of numbers to addition of their logarithms). Because of this property, the Laplace variable s is also known as *operator variable* in the L domain: either *derivative operator* or (for s^{-1}) *integration operator*. The transform turns integral equations and differential equations to polynomial equations, which are much easier to solve. Once solved, use of the inverse Laplace transform reverts back to the time domain.

Given the functions *f*(*t*) and *g*(*t*), and their respective Laplace transforms *F*(*s*) and *G*(*s*):

$$f(t) = \mathcal{L}^{-1}\{F(s)\}$$
$$g(t) = \mathcal{L}^{-1}\{G(s)\}$$

the following table is a list of properties of unilateral Laplace transform:

Properties of the unilateral Laplace transform

	Time domain	's' domain	Comment
Linearity	$af(t) + bg(t)$	$aF(s) + bG(s)$	Can be proved using basic rules of integration.
Frequency differentiation	$tf(t)$	$-F'(s)$	F' is the first derivative of F.
Frequency differentiation	$t^n f(t)$	$(-1)^n F^{(n)}(s)$	More general form, n^{th} derivative of F(s).
Differentiation	$f'(t)$	$sF(s) - f(0)$	*f* is assumed to be a differentiable function, and its derivative is assumed to be of exponential type. This can then be obtained by integration by parts
Second Differentiation	$f''(t)$	$s^2F(s) - sf(0) - f'(0)$	*f* is assumed twice differentiable and the second derivative to be of exponential type. Follows by applying the Differentiation property to $f'(t)$.
General Differentiation	$f^{(n)}(t)$	$s^nF(s) - s^{n-1}f(0) - \cdots - f^{(n-1)}(0)$	*f* is assumed to be *n*-times differentiable, with n^{th} derivative of exponential type. Follow by mathematical induction.

Frequency integration	$\frac{f(t)}{t}$	$\int_s^\infty F(\sigma)\,d\sigma$	
Integration	$\int_0^t f(\tau)\,d\tau = (u * f)(t)$	$\frac{1}{s}F(s)$	$u(t)$ is the Heaviside step function. Note $(u * f)(t)$ is the convolution of $u(t)$ and $f(t)$.
Scaling	$f(at)$	$\frac{1}{a}F\left(\frac{s}{a}\right)$	where a is positive.
Frequency shifting	$e^{at}f(t)$	$F(s-a)$	
Time shifting	$f(t-a)u(t-a)$	$e^{-as}F(s)$	$u(t)$ is the Heaviside step function
Multiplication	$f(t)g(t)$	$\frac{1}{2\pi i}\lim_{T\to\infty}\int_{c-iT}^{c+iT} F(\sigma)G(s-\sigma)\,d\sigma$	the integration is done along the vertical line $Re(\sigma) = c$ that lies entirely within the region of convergence of F.
Convolution	$(f * g)(t) = \int_0^t f(\tau)g(t-\tau)\,d\tau$	$F(s)\cdot G(s)$	$f(t)$ and $g(t)$ are extended by zero for $t < 0$ in the definition of the convolution.
Periodic Function	$f(t)$	$\frac{1}{1-e^{-Ts}}\int_0^T e^{-st}f(t)\,dt$	$f(t)$ is a periodic function of period T so that $f(t) = f(t+T)$. This is the result of the time shifting property and the geometric series.

- **Initial value theorem**:

$$f(0^+) = \lim_{s\to\infty} sF(s).$$

- **Final value theorem**:

$$f(\infty) = \lim_{s\to 0} sF(s)$$, if all poles of $sF(s)$ are in the left half-plane.

The final value theorem is useful because it gives the long-term behaviour without having to perform partial fraction decompositions or other difficult algebra. If a function's poles are in the right-hand plane (e.g. e^t or sin(t)) the behaviour of this formula is undefined.

Proof of the Laplace transform of a function's derivative

It is often convenient to use the differentiation property of the Laplace transform to find the transform of a function's derivative. This can be derived from the basic expression for a Laplace transform as follows:

$$\begin{aligned}\mathcal{L}\left\{f(t)\right\} &= \int_{0^-}^{\infty} e^{-st} f(t)\,dt \\ &= \left[\frac{f(t)e^{-st}}{-s}\right]_{0^-}^{\infty} - \int_{0^-}^{\infty} \frac{e^{-st}}{-s} f'(t)\,dt \quad \text{(by parts)} \\ &= \left[-\frac{f(0)}{-s}\right] + \frac{1}{s}\mathcal{L}\left\{f'(t)\right\},\end{aligned}$$

yielding

$$\mathcal{L}\left\{f'(t)\right\} = s\cdot\mathcal{L}\left\{f(t)\right\} - f(0),$$

and in the bilateral case,

$$\mathcal{L}\left\{f'(t)\right\} = s\int_{-\infty}^{\infty} e^{-st} f(t)\,dt = s\cdot\mathcal{L}\{f(t)\}.$$

The general result

$$\mathcal{L}\left\{f^{(n)}(t)\right\} = s^n\cdot\mathcal{L}\left\{f(t)\right\} - s^{n-1}f(0) - \cdots - f^{(n-1)}(0),$$

where f^n is the n-th derivative of f, can then be established with an inductive argument.

Evaluating improper integrals

Let $\mathcal{L}\left\{f(t)\right\} = F(s)$, then (see the table above)

$$\mathcal{L}\left\{\frac{f(t)}{t}\right\} = \int_s^{\infty} F(p)\,dp,$$

or

$$\int_0^\infty \frac{f(t)}{t} e^{-st}\, dt = \int_s^\infty F(p)\, dp.$$

Let $s \longrightarrow 0$we get the identity

$$\int_0^\infty \frac{f(t)}{t}\, dt = \int_0^\infty F(p)\, dp.$$

For example,

$$\int_0^\infty \frac{\cos at - \cos bt}{t}\, dt = \int_0^\infty \left(\frac{p}{p^2+a^2} - \frac{p}{p^2+b^2}\right)\, dp = \frac{1}{2} \ln \frac{p^2+a^2}{p^2+b^2}\bigg|_0^\infty = \ln b - \ln a.$$

Another example is Dirichlet integral.

Relationship to other transforms

Laplace–Stieltjes transform

The (unilateral) Laplace–Stieltjes transform of a function $g : \mathbf{R} \to \mathbf{R}$ is defined by the Lebesgue–Stieltjes integral

$$\{\mathcal{L}^* g\}(s) = \int_0^\infty e^{-st} dg(t).$$

The function *g* is assumed to be of bounded variation. If *g* is the antiderivative of *f*:

$$g(x) = \int_0^x f(t)\, dt$$

then the Laplace–Stieltjes transform of *g* and the Laplace transform of *f* coincide. In general, the Laplace–Stieltjes transform is the Laplace transform of the Stieltjes measure associated to *g*. So in practice, the only distinction between the two transforms is that the Laplace transform is thought of as operating on the density function of the measure, whereas the Laplace–Stieltjes transform is thought of as operating on its cumulative distribution function.

Fourier transform

The continuous Fourier transform is equivalent to evaluating the bilateral Laplace transform with complex argument $s = i\omega$ or $s = 2\pi f i$:

$$\hat{f}(\omega) = \mathcal{F}\{f(t)\}$$

$$= \mathcal{L}\{f(t)\}|_{s=i\omega} = F(s)|_{s=i\omega}$$

$$= \int_{-\infty}^{\infty} e^{-\imath\omega t} f(t)\,\mathrm{d}t.$$

This expression excludes the scaling factor $1/\sqrt{2\pi}$, which is often included in definitions of the Fourier transform. This relationship between the Laplace and Fourier transforms is often used to determine the frequency spectrum of a signal or dynamical system.

The above relation is valid as stated if and only if the region of convergence (ROC) of $F(s)$ contains the imaginary axis, $\sigma = 0$. For example, the function $f(t) = \cos(\omega_0 t)u(t)$ has a Laplace transform $F(s) = s/(s^2 + \omega_0^2)$ whose ROC is $\mathrm{Re}(s) > 0$. Therefore, substituting $s = i\omega$ in $F(s)$ does not yield the Fourier transform of $f(t) = \cos(\omega_0 t)$.

However, a relation of the form

$$\lim_{\sigma \to 0^+} F(\sigma + i\omega) = \hat{f}(\omega)$$

holds under much weaker conditions. For instance, this holds for the above example provided that the limit is understood as a weak limit of measures. General conditions relating the limit of the Laplace transform of a function on the boundary to the Fourier transform take the form of Paley-Wiener theorems.

Mellin transform

The Mellin transform and its inverse are related to the two-sided Laplace transform by a simple change of variables. If in the Mellin transform

$$G(s) = \mathcal{M}\{g(\theta)\} = \int_0^{\infty} \theta^s g(\theta) \frac{d\theta}{\theta}$$

we set $\theta = e^{-t}$ we get a two-sided Laplace transform.

Z-transform

The unilateral or one-sided Z-transform is simply the Laplace transform of an ideally sampled signal with the substitution of

$$z \stackrel{\mathrm{def}}{=} e^{sT}$$

where $T = 1/f_s$ is the sampling period (in units of time e.g., seconds) and f_s is the sampling rate (in samples per second or hertz)

Let

$$\Delta_T(t) \ \stackrel{\text{def}}{=}\ \sum_{n=0}^{\infty} \delta(t - nT)$$

be a sampling impulse train (also called a Dirac comb) and

$$\begin{aligned} x_q(t) &\stackrel{\text{def}}{=} x(t)\Delta_T(t) = x(t)\sum_{n=0}^{\infty} \delta(t - nT) \\ &= \sum_{n=0}^{\infty} x(nT)\delta(t - nT) = \sum_{n=0}^{\infty} x[n]\delta(t - nT) \end{aligned}$$

be the continuous-time representation of the sampled $x(t)$

$x[n] \ \stackrel{\text{def}}{=}\ x(nT)$ are the discrete samples of $x(t)$.

The Laplace transform of the sampled signal $x_q(t)$ is

$$\begin{aligned} X_q(s) &= \int_{0^-}^{\infty} x_q(t) e^{-st}\, dt \\ &= \int_{0^-}^{\infty} \sum_{n=0}^{\infty} x[n]\delta(t - nT) e^{-st}\, dt \\ &= \sum_{n=0}^{\infty} x[n] \int_{0^-}^{\infty} \delta(t - nT) e^{-st}\, dt \\ &= \sum_{n=0}^{\infty} x[n] e^{-nsT}. \end{aligned}$$

This is precisely the definition of the unilateral Z-transform of the discrete function $x[n]$

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

with the substitution of $z \leftarrow e^{sT}$.

Comparing the last two equations, we find the relationship between the unilateral Z-transform and the Laplace transform of the sampled signal:

$$X_q(s) = X(z)\Big|_{z=e^{sT}}.$$

The similarity between the Z and Laplace transforms is expanded upon in the theory of time scale calculus.

Borel transform

The integral form of the Borel transform

$$F(s) = \int_0^\infty f(z)e^{-sz}\,dz$$

is a special case of the Laplace transform for *f* an entire function of exponential type, meaning that

$$|f(z)| \le Ae^{B|z|}$$

for some constants *A* and *B*. The generalized Borel transform allows a different weighting function to be used, rather than the exponential function, to transform functions not of exponential type. Nachbin's theorem gives necessary and sufficient conditions for the Borel transform to be well defined.

Fundamental relationships

Since an ordinary Laplace transform can be written as a special case of a two-sided transform, and since the two-sided transform can be written as the sum of two one-sided transforms, the theory of the Laplace-, Fourier-, Mellin-, and Z-transforms are at bottom the same subject. However, a different point of view and different characteristic problems are associated with each of these four major integral transforms.

Table of selected Laplace transforms

The following table provides Laplace transforms for many common functions of a single variable.

Because the Laplace transform is a linear operator:

- The Laplace transform of a sum is the sum of Laplace transforms of each term.

$$\mathcal{L}\left\{f(t)+g(t)\right\}=\mathcal{L}\left\{f(t)\right\}+\mathcal{L}\left\{g(t)\right\}$$

- The Laplace transform of a multiple of a function is that multiple times the Laplace transformation of that function.

$$\mathcal{L}\left\{af(t)\right\}=a\mathcal{L}\left\{f(t)\right\}$$

The unilateral Laplace transform takes as input a function whose time domain is the non-negative reals, which is why all of the time domain functions in the table below are multiples of the Heaviside step function, u(t). The entries of the table that involve a time delay τ are required to be causal (meaning that $\tau > 0$). A causal system is a system where the impulse response $h(t)$ is zero for all time t prior to $t = 0$. In general, the region of convergence for causal systems is not the same as that of anticausal systems.

ID	Function	Time domain $f(t)=\mathcal{L}^{-1}\left\{F(s)\right\}$	Laplace s-domain $F(s)=\mathcal{L}\left\{f(t)\right\}$	Region of convergence
1	ideal delay	$\delta(t-\tau)$	$e^{-\tau s}$	
1a	unit impulse	$\delta(t)$	1	all s
2	delayed *n*th power with frequency shift	$\frac{(t-\tau)^n}{n!}e^{-\alpha(t-\tau)}\cdot u(t-\tau)$	$\frac{e^{-\tau s}}{(s+\alpha)^{n+1}}$	$\mathrm{Re}\{s\}>-\alpha$
2a	*n*th power (for integer *n*)	$\frac{t^n}{n!}\cdot u(t)$	$\frac{1}{s^{n+1}}$	$\mathrm{Re}\{s\}>0$ $(n>-1)$
2a.1	*q*th power (for complex *q*)	$\frac{t^q}{\Gamma(q+1)}\cdot u(t)$	$\frac{1}{s^{q+1}}$	$\mathrm{Re}\{s\}>0$ $(\mathrm{Re}\{q\}>-1)$
2a.2	unit step	$u(t)$	$\frac{1}{s}$	$\mathrm{Re}\{s\}>0$
2b	delayed unit step	$u(t-\tau)$	$\frac{e^{-\tau s}}{s}$	$\mathrm{Re}\{s\}>0$
2c	ramp	$t\cdot u(t)$	$\frac{1}{s^2}$	$\mathrm{Re}\{s\}>0$
2d	*n*th power with frequency shift	$\frac{t^n}{n!}e^{-\alpha t}\cdot u(t)$	$\frac{1}{(s+\alpha)^{n+1}}$	$\mathrm{Re}\{s\}>-\alpha$
2d.1	exponential decay	$e^{-\alpha t}\cdot u(t)$	$\frac{1}{s+\alpha}$	$\mathrm{Re}\{s\}>-\alpha$

3	exponential approach	$(1-e^{-\alpha t})\cdot u(t)$	$\frac{\alpha}{s(s+\alpha)}$	$\mathrm{Re}\{s\}>0$
4	sine	$\sin(\omega t)\cdot u(t)$	$\frac{\omega}{s^2+\omega^2}$	$\mathrm{Re}\{s\}>0$
5	cosine	$\cos(\omega t)\cdot u(t)$	$\frac{s}{s^2+\omega^2}$	$\mathrm{Re}\{s\}>0$
6	hyperbolic sine	$\sinh(\alpha t)\cdot u(t)$	$\frac{\alpha}{s^2-\alpha^2}$	$\mathrm{Re}\{s\}>\|\alpha\|$
7	hyperbolic cosine	$\cosh(\alpha t)\cdot u(t)$	$\frac{s}{s^2-\alpha^2}$	$\mathrm{Re}\{s\}>\|\alpha\|$
8	Exponentially-decaying sine wave	$e^{-\alpha t}\sin(\omega t)\cdot u(t)$	$\frac{\omega}{(s+\alpha)^2+\omega^2}$	$\mathrm{Re}\{s\}>-\alpha$
9	Exponentially-decaying cosine wave	$e^{-\alpha t}\cos(\omega t)\cdot u(t)$	$\frac{s+\alpha}{(s+\alpha)^2+\omega^2}$	$\mathrm{Re}\{s\}>-\alpha$
10	*n*th root	$\sqrt[n]{t}\cdot u(t)$	$s^{-(n+1)/n}\cdot\Gamma\left(1+\frac{1}{n}\right)$	$\mathrm{Re}\{s\}>0$
11	natural logarithm	$\ln\left(\frac{t}{t_0}\right)\cdot u(t)$	$-\frac{t_0}{s}\,[\ln(t_0 s)+\gamma]$	$\mathrm{Re}\{s\}>0$
12	Bessel function of the first kind, of order *n*	$J_n(\omega t)\cdot u(t)$	$\frac{\omega^n\left(s+\sqrt{s^2+\omega^2}\right)^{-n}}{\sqrt{s^2+\omega^2}}$	$\mathrm{Re}\{s\}>0$ $(n>-1)$
13	Modified Bessel function of the first kind, of order *n*	$I_n(\omega t)\cdot u(t)$	$\frac{\omega^n\left(s+\sqrt{s^2-\omega^2}\right)^{-n}}{\sqrt{s^2-\omega^2}}$	$\mathrm{Re}\{s\}>\|\omega\|$
14	Bessel function of the second kind, of order 0	$Y_0(\alpha t)\cdot u(t)$	$-\frac{2\sinh^{-1}(s/\alpha)}{\pi\sqrt{s^2+\alpha^2}}$	$\mathrm{Re}\{s\}>0$
15	Modified Bessel function of the second kind, of order 0	$K_0(\alpha t)\cdot u(t)$		

16	Error function	$\operatorname{erf}(t) \cdot u(t)$	$\dfrac{e^{s^2/4}\left(1 - \operatorname{erf}\left(s/2\right)\right)}{s}$	$\operatorname{Re}\{s\} > 0$

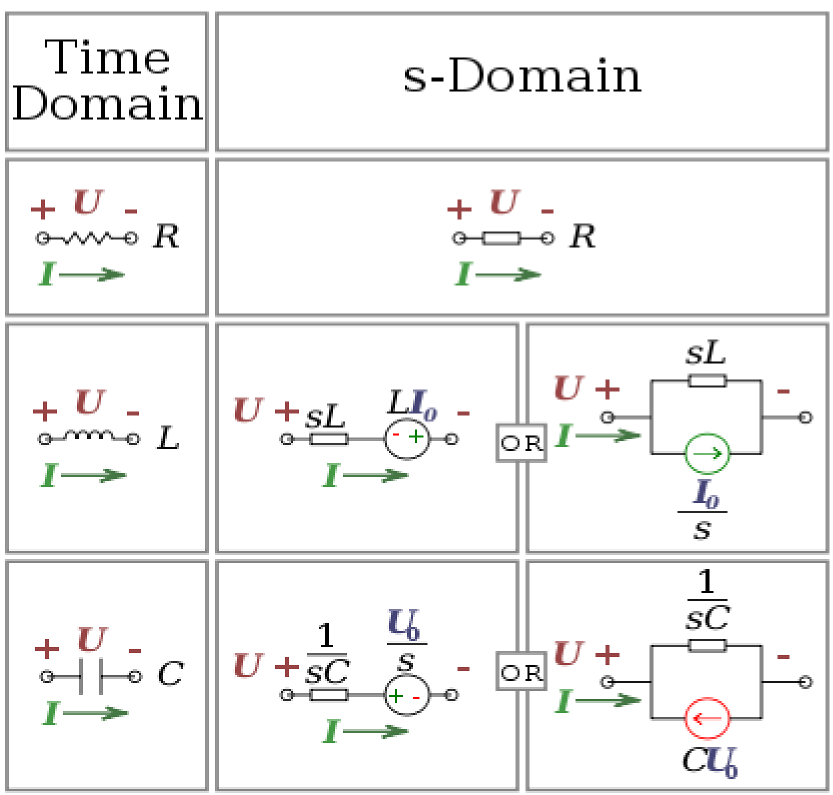
Explanatory notes:

- $u(t)$ represents the Heaviside step function.
- $\delta(t)$ represents the Dirac delta function.
- $\Gamma(z)$ represents the Gamma function.
- γ is the Euler–Mascheroni constant.
- t, a real number, typically represents *time*, although it can represent *any* independent dimension.
- s is the complex angular frequency, and $\operatorname{Re}\{s\}$ is its real part.
- α, β, τ, and ω are real numbers.
- *n* is an integer.

s-Domain equivalent circuits and impedances

The Laplace transform is often used in circuit analysis, and simple conversions to the s-Domain of circuit elements can be made. Circuit elements can be transformed into impedances, very similar to phasor impedances.

Here is a summary of equivalents:



Note that the resistor is exactly the same in the time domain and the s-Domain. The sources are put in if there are initial conditions on the circuit elements. For example, if a capacitor has an initial voltage across it, or if the inductor has an initial current through it, the sources inserted in the s-Domain account for that.

The equivalents for current and voltage sources are simply derived from the transformations in the table above.

Examples: How to apply the properties and theorems

The Laplace transform is used frequently in engineering and physics; the output of a linear time invariant system can be calculated by convolving its unit impulse response with the input signal. Performing this calculation in Laplace space turns the convolution into a multiplication; the latter being easier to solve because of its algebraic form.

The Laplace transform can also be used to solve differential equations and is used extensively in electrical engineering. The Laplace transform reduces a linear differential equation to an algebraic equation, which can then be solved by the formal rules of algebra. The original differential equation can then be solved by applying the inverse Laplace transform. The English electrical engineer Oliver Heaviside first proposed a similar scheme, although without using the Laplace transform; and the resulting operational calculus is credited as the Heaviside calculus.

Example #1: Solving a differential equation

In nuclear physics, the following fundamental relationship governs radioactive decay: the number of radioactive atoms *N* in a sample of a radioactive isotope decays at a rate proportional to *N*. This leads to the first order linear differential equation

$$\frac{dN}{dt} = -\lambda N$$

where λ is the decay constant. The Laplace transform can be used to solve this equation.

Rearranging the equation to one side, we have

$$\frac{dN}{dt} + \lambda N = 0.$$

Next, we take the Laplace transform of both sides of the equation:

$$\left(s\tilde{N}(s) - N_o\right) + \lambda\tilde{N}(s) \ = \ 0$$

where

$$\tilde{N}(s) = \mathcal{L}\{N(t)\}$$

and

$$N_o \;=\; N(0).$$

Solving, we find

$$\tilde{N}(s) = \frac{N_o}{s+\lambda}.$$

Finally, we take the inverse Laplace transform to find the general solution

$$\begin{aligned} N(t) &= \mathcal{L}^{-1}\{\tilde{N}(s)\} = \mathcal{L}^{-1}\left\{\frac{N_o}{s+\lambda}\right\} \\ &= \; N_o e^{-\lambda t}, \end{aligned}$$

which is indeed the correct form for radioactive decay.

Example #2: Deriving the complex impedance for a capacitor

In the theory of electrical circuits, the current flow in a capacitor is proportional to the capacitance and rate of change in the electrical potential (in SI units). Symbolically, this is expressed by the differential equation

$$i = C\frac{dv}{dt}$$

where C is the capacitance (in farads) of the capacitor, $i = i(t)$ is the electric current (in amperes) through the capacitor as a function of time, and $v = v(t)$ is the voltage (in volts) across the terminals of the capacitor, also as a function of time.

Taking the Laplace transform of this equation, we obtain

$$I(s) = C\left(sV(s) - V_o\right)$$

where

$$\begin{aligned} I(s) &= \mathcal{L}\{i(t)\}, \\ V(s) &= \mathcal{L}\{v(t)\}, \end{aligned}$$

and

$$V_o \;=\; v(t)|_{t=0}.$$

Solving for $V(s)$ we have

$$V(s) = \frac{I(s)}{sC} + \frac{V_o}{s}.$$

The definition of the complex impedance Z (in ohms) is the ratio of the complex voltage V divided by the complex current I while holding the initial state V_o at zero:

$$Z(s) = \left.\frac{V(s)}{I(s)}\right|_{V_o=0}.$$

Using this definition and the previous equation, we find:

$$Z(s) = \frac{1}{sC},$$

which is the correct expression for the complex impedance of a capacitor.

Example #3: Method of partial fraction expansion

Consider a linear time-invariant system with transfer function

$$H(s) = \frac{1}{(s+\alpha)(s+\beta)}.$$

The impulse response is simply the inverse Laplace transform of this transfer function:

$$h(t) = \mathcal{L}^{-1}\{H(s)\}.$$

To evaluate this inverse transform, we begin by expanding $H(s)$ using the method of partial fraction expansion:

$$\frac{1}{(s+\alpha)(s+\beta)} = \frac{P}{s+\alpha} + \frac{R}{s+\beta}.$$

The unknown constants P and R are the residues located at the corresponding poles of the transfer function. Each residue represents the relative contribution of that singularity to the transfer function's overall shape. By the residue theorem, the inverse Laplace transform depends only upon the poles and their residues. To find the residue P, we multiply both sides of the equation by $s + \alpha$ to get

$$\frac{1}{s+\beta} = P + \frac{R(s+\alpha)}{s+\beta}.$$

Then by letting $s = -\alpha$, the contribution from R vanishes and all that is left is

$$P = \left.\frac{1}{s+\beta}\right|_{s=-\alpha} = \frac{1}{\beta-\alpha}.$$

Similarly, the residue R is given by

$$R = \left.\frac{1}{s+\alpha}\right|_{s=-\beta} = \frac{1}{\alpha-\beta}.$$

Note that

$$R = \frac{-1}{\beta-\alpha} = -P$$

and so the substitution of R and P into the expanded expression for $H(s)$ gives

$$H(s) = \left(\frac{1}{\beta-\alpha}\right) \cdot \left(\frac{1}{s+\alpha} - \frac{1}{s+\beta}\right).$$

Finally, using the linearity property and the known transform for exponential decay, we can take the inverse Laplace transform of $H(s)$ to obtain:

$$h(t) = \mathcal{L}^{-1}\{H(s)\} = \frac{1}{\beta-\alpha}\left(e^{-\alpha t} - e^{-\beta t}\right),$$

which is the impulse response of the system.

Example #4: Mixing sines, cosines, and exponentials

Time function	Laplace transform
$e^{-\alpha t}\left[\cos(\omega t) + \left(\frac{\beta-\alpha}{\omega}\right)\sin(\omega t)\right]u(t)$	$\frac{s+\beta}{(s+\alpha)^2+\omega^2}$

Starting with the Laplace transform

$$X(s) = \frac{s+\beta}{(s+\alpha)^2+\omega^2},$$

we find the inverse transform by first adding and subtracting the same constant α to the numerator:

$$X(s) = \frac{s+\alpha}{(s+\alpha)^2+\omega^2} + \frac{\beta-\alpha}{(s+\alpha)^2+\omega^2}.$$

By the shift-in-frequency property, we have

$$\begin{aligned} x(t) &= e^{-\alpha t}\mathcal{L}^{-1}\left\{\frac{s}{s^2+\omega^2} + \frac{\beta-\alpha}{s^2+\omega^2}\right\} \\ &= e^{-\alpha t}\mathcal{L}^{-1}\left\{\frac{s}{s^2+\omega^2} + \left(\frac{\beta-\alpha}{\omega}\right)\left(\frac{\omega}{s^2+\omega^2}\right)\right\} \\ &= e^{-\alpha t}\left[\mathcal{L}^{-1}\left\{\frac{s}{s^2+\omega^2}\right\} + \left(\frac{\beta-\alpha}{\omega}\right)\mathcal{L}^{-1}\left\{\frac{\omega}{s^2+\omega^2}\right\}\right]. \end{aligned}$$

Finally, using the Laplace transforms for sine and cosine (see the table, above), we have

$$x(t) = e^{-\alpha t}\left[\cos(\omega t)u(t) + \left(\frac{\beta-\alpha}{\omega}\right)\sin(\omega t)u(t)\right].$$

$$x(t) = e^{-\alpha t}\left[\cos(\omega t) + \left(\frac{\beta-\alpha}{\omega}\right)\sin(\omega t)\right]u(t).$$

Example #5: Phase delay

Time function	Laplace transform
$\sin(\omega t+\phi)$	$\frac{s\sin\phi+\omega\cos\phi}{s^2+\omega^2}$
$\cos(\omega t+\phi)$	$\frac{s\cos\phi-\omega\sin\phi}{s^2+\omega^2}$

Starting with the Laplace transform,

$$X(s) = \frac{s\sin\phi+\omega\cos\phi}{s^2+\omega^2}$$

we find the inverse by first rearranging terms in the fraction:

$$X(s) = \frac{s\sin\phi}{s^2+\omega^2} + \frac{\omega\cos\phi}{s^2+\omega^2}$$
$$= (\sin\phi)\left(\frac{s}{s^2+\omega^2}\right) + (\cos\phi)\left(\frac{\omega}{s^2+\omega^2}\right).$$

We are now able to take the inverse Laplace transform of our terms:

$$x(t) = (\sin\phi)\mathcal{L}^{-1}\left\{\frac{s}{s^2+\omega^2}\right\} + (\cos\phi)\mathcal{L}^{-1}\left\{\frac{\omega}{s^2+\omega^2}\right\}$$
$$= (\sin\phi)(\cos\omega t) + (\sin\omega t)(\cos\phi).$$

This is just the sine of the sum of the arguments, yielding:

$x(t) = \sin(\omega t + \varphi)$.

We can apply similar logic to find that

$$\mathcal{L}^{-1}\left\{\frac{s\cos\phi - \omega\sin\phi}{s^2+\omega^2}\right\} = \cos(\omega t + \phi).$$

Chapter- 9

Z-Transform

In mathematics and signal processing, the **Z-transform** converts a discrete time-domain signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.

It can be considered as a discrete-time equivalent of the Laplace transform. This similarity is explored in the theory of time scale calculus.

History

The basic ideas now known as the Z-transform was known to Laplace, and re-introduced in 1947 by W. Hurewicz as a tractable way to solve linear, constant-coefficient difference equations. It was later dubbed "the z-transform" by Ragazzini and Zadeh in the sampled-data control group at Columbia University in 1952.

The name "Z-transform" was derived from the idea of the letter "z" being a sampled/digitized version of the letter "s" often used as the independent variable in Laplace transforms. This seemed appropriate since the Z-transform can be viewed as a sampled version of the Laplace transform. The naming deviates from the more commonly used scientific naming practice of associating a method or theorem with the principal investigator (i.e. Fourier, Laplace, Hartley, etc.).

The modified or advanced Z-transform was later developed and popularized by E. I. Jury.

The idea contained within the Z-transform is also known in mathematical literature as the method of generating functions which can be traced back as early as 1730 when it was introduced by DeMoivre in conjunction with probability theory. From a mathematical view the Z-transform can also be viewed as a Laurent series where one views the sequence of numbers under consideration as the (Laurent) expansion of an analytic function (the Z-transform).

Definition

The Z-transform, like many integral transforms, can be defined as either a *one-sided* or *two-sided* transform.

Bilateral Z-transform

The *bilateral* or *two-sided* Z-transform of a discrete-time signal *x[n]* is the function *X(z)* defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

where *n* is an integer, *j* is the imaginary unit, and *z* is, in general, a complex number:

$$z = Ae^{j\varphi} = A(\cos\varphi + j\sin\varphi)$$

where *A* is the magnitude of *z*, and φ is the *complex argument* (also referred to as *angle* or *phase*) in radians.

Unilateral Z-transform

Alternatively, in cases where *x*[*n*] is defined only for $n \geq 0$, the *single-sided* or *unilateral* Z-transform is defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n}.$$

In signal processing, this definition can be used to evaluate the Z-transform of the unit impulse response of a discrete-time causal system.

An important example of the unilateral Z-transform is the probability-generating function, where the component *x*[*n*] is the probability that a discrete random variable takes the value *n*, and the function *X*(*z*) is usually written as *X*(*s*), in terms of $s = z^{-1}$. The properties of Z-transforms (below) have useful interpretations in the context of probability theory.

Geophysical definition

In geophysics, the usual definition for the Z-transform is a polynomial in *z* as opposed to z^{-1}. This convention is used by Robinson and Treitel and by Kanasewich. The geophysical definition is

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_n x[n]z^n.$$

The two definitions are equivalent; however, the difference results in a number of changes. For example, the location of zeros and poles move from inside the unit circle, using one definition, to outside the unit circle, using the other definition (and vice versa). Thus, care is required to note which definition is being used by a particular author.

Inverse Z-transform

The *inverse* Z-transform is

$$x[n] = \mathcal{Z}^{-1}\{X(z)\} = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz$$

where C is a counterclockwise closed path encircling the origin and entirely in the region of convergence (ROC). The contour or path, C, must encircle all of the poles of $X(z)$.

A special case of this contour integral occurs when C is the unit circle (and can be used when the ROC includes the unit circle which is always guaranteed when $X(z)$ is stable, i.e. all the roots are within the unit circle). The inverse Z-transform simplifies to the inverse discrete-time Fourier transform:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{j\omega}) e^{j\omega n} d\omega.$$

The Z-transform with a finite range of *n* and a finite number of uniformly-spaced *z* values can be computed efficiently via Bluestein's FFT algorithm. The discrete-time Fourier transform (DTFT) (not to be confused with the discrete Fourier transform (DFT)) is a special case of such a Z-transform obtained by restricting *z* to lie on the unit circle.

Region of convergence

The region of convergence (ROC) is the set of points in the complex plane for which the Z-transform summation converges.

$$ROC = \left\{ z : \left| \sum_{n=-\infty}^{\infty} x[n] z^{-n} \right| < \infty \right\}$$

Example 1 (no ROC)

Let $x[n] = 0.5^n$. Expanding $x[n]$ on the interval $(-\infty, \infty)$ it becomes

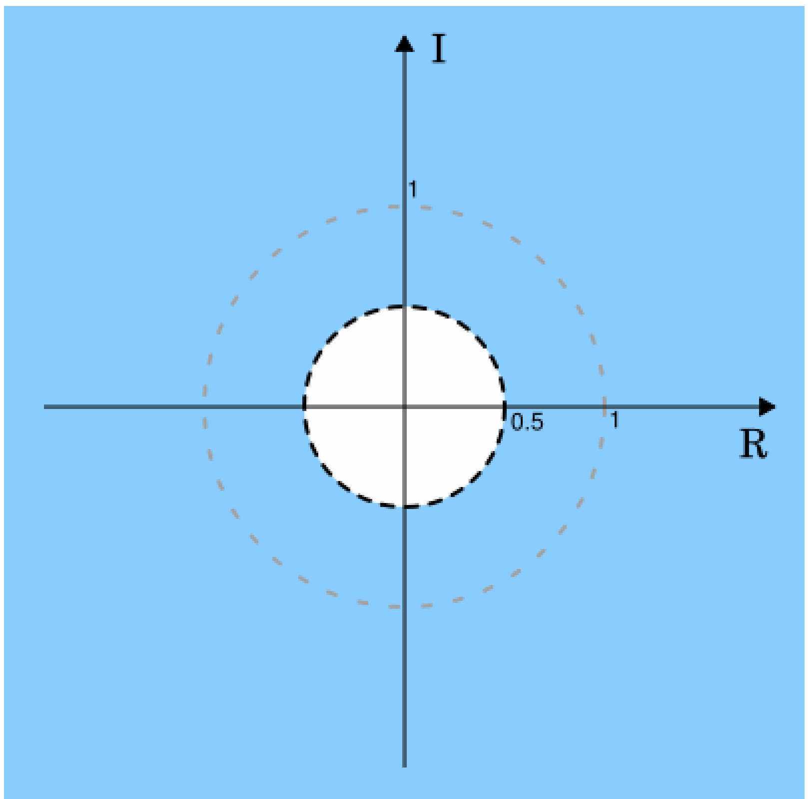
$$x[n] = \{..., 0.5^{-3}, 0.5^{-2}, 0.5^{-1}, 1, 0.5, 0.5^2, 0.5^3, ...\} = \{..., 2^3, 2^2, 2, 1, 0.5, 0.5^2, 0.5^3, ...\} .$$

Looking at the sum

$$\sum_{n=-\infty}^{\infty} x[n] z^{-n} \to \infty \ .$$

Therefore, there are no such values of z that satisfy this condition.

Example 2 (causal ROC)



ROC shown in blue, the unit circle as a dotted grey circle and the circle $|z| = 0.5$ is shown as a dashed black circle

Let $x[n] = 0.5^n u[n]$ (where *u* is the Heaviside step function). Expanding $x[n]$ on the interval $(-\infty, \infty)$ it becomes

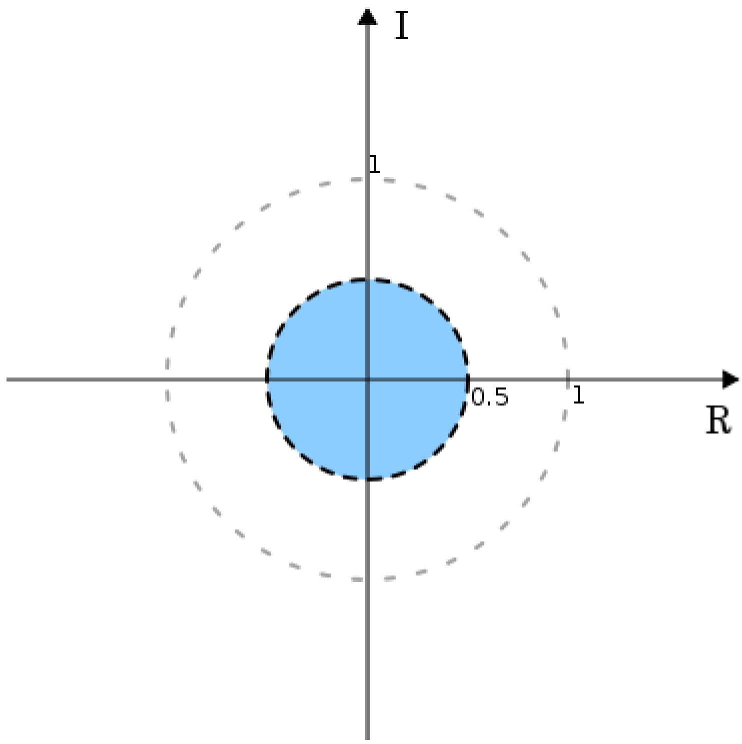
$$x[n] = \{..., 0, 0, 0, 1, 0.5, 0.5^2, 0.5^3, ...\}.$$

Looking at the sum

$$\sum_{n=-\infty}^{\infty} x[n] z^{-n} = \sum_{n=0}^{\infty} 0.5^n z^{-n} = \sum_{n=0}^{\infty} \left(\frac{0.5}{z}\right)^n = \frac{1}{1 - 0.5z^{-1}}.$$

The last equality arises from the infinite geometric series and the equality only holds if $\left|0.5z^{-1}\right| < 1$ which can be rewritten in terms of z as $|z| > 0.5$. Thus, the ROC is $|z| > 0.5$. In this case the ROC is the complex plane with a disc of radius 0.5 at the origin "punched out".

Example 3 (anticausal ROC)



ROC shown in blue, the unit circle as a dotted grey circle and the circle $|z| = 0.5$ is shown as a dashed black circle

Let $x[n] = -(0.5)^n u[-n-1]$ (where *u* is the Heaviside step function). Expanding $x[n]$ on the interval $(-\infty, \infty)$ it becomes

$$x[n] = \{..., -(0.5)^{-3}, -(0.5)^{-2}, -(0.5)^{-1}, 0, 0, 0, 0, ...\}.$$

Looking at the sum

$$\sum_{n=-\infty}^{\infty} x[n]z^{-n} = -\sum_{n=-\infty}^{-1} 0.5^n z^{-n} = -\sum_{n=-\infty}^{-1} \left(\frac{z}{0.5}\right)^{-n}$$
$$= -\sum_{m=1}^{\infty} \left(\frac{z}{0.5}\right)^{m} = -\frac{0.5^{-1}z}{1-0.5^{-1}z} = \frac{z}{z-0.5} = \frac{1}{1-0.5z^{-1}}.$$

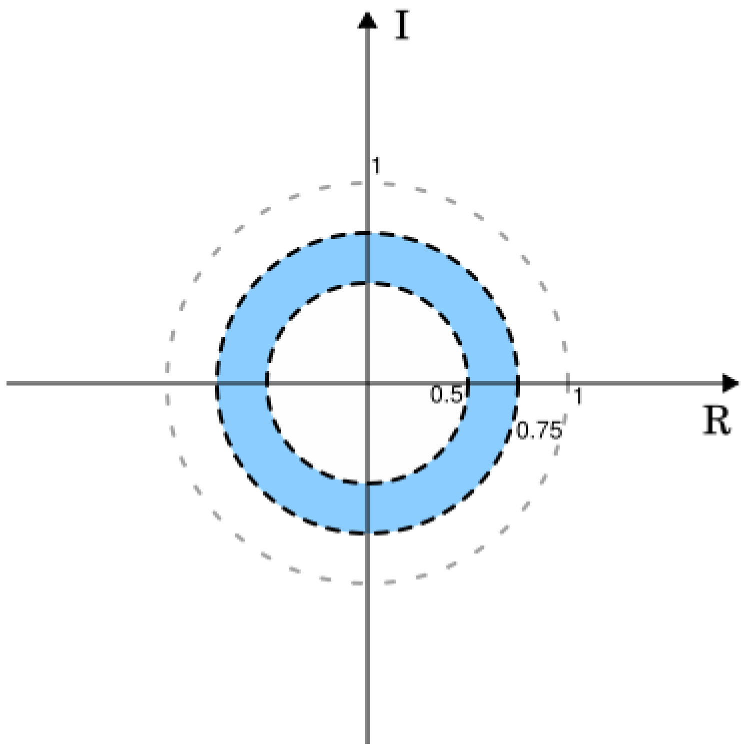
Using the infinite geometric series, again, the equality only holds if $\left|0.5^{-1}z\right| < 1$ which can be rewritten in terms of z as $|z| < 0.5$. Thus, the ROC is $|z| < 0.5$. In this case the ROC is a disc centered at the origin and of radius 0.5.

What differentiates this example from the previous example is *only* the ROC. This is intentional to demonstrate that the transform result alone is insufficient.

Examples conclusion

Examples 2 & 3 clearly show that the Z-transform $X(z)$ of $x[n]$ is unique when and only when specifying the ROC. Creating the pole-zero plot for the causal and anticausal case show that the ROC for either case does not include the pole that is at 0.5. This extends to cases with multiple poles: the ROC will *never* contain poles.

In example 2, the causal system yields an ROC that includes $|z| = \infty$ while the anticausal system in example 3 yields an ROC that includes $|z| = 0$.



ROC shown as a blue ring $0.5 < |z| < 0.75$

In systems with multiple poles it is possible to have an ROC that includes neither $|z| = \infty$ nor $|z| = 0$. The ROC creates a circular band. For example, $x[n] = 0.5^n u[n] - 0.75^n u[-n-1]$ has poles at 0.5 and 0.75. The ROC will be

$0.5 < |z| < 0.75$, which includes neither the origin nor infinity. Such a system is called a mixed-causality system as it contains a causal term $0.5^n u[n]$ and an anticausal term $-(0.75)^n u[-n-1]$.

The stability of a system can also be determined by knowing the ROC alone. If the ROC contains the unit circle (i.e., $|z| = 1$) then the system is stable. In the above systems the causal system (Example 2) is stable because $|z| > 0.5$ contains the unit circle.

If you are provided a Z-transform of a system without an ROC (i.e., an ambiguous $x[n]$) you can determine a unique $x[n]$ provided you desire the following:

- Stability
- Causality

If you need stability then the ROC must contain the unit circle. If you need a causal system then the ROC must contain infinity and the system function will be a right-sided sequence. If you need an anticausal system then the ROC must contain the origin and the system function will be a left-sided sequence. If you need both, stability and causality, all the poles of the system function must be inside the unit circle.

The unique $x[n]$ can then be found.

Table of common Z-transform pairs

Here:

- $u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$
- $\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$

	Signal, *x*[*n*]	**Z-transform, *X*(*z*)**	**ROC**
1	$\delta[n]$	1	all z
2	$\delta[n-n_0]$	z^{-n_0}	$z \neq 0$
3	$u[n]$	$\dfrac{1}{1-z^{-1}}$	$\|z\| > 1$
4	$-u[-n-1]$	$\dfrac{1}{1-z^{-1}}$	$\|z\| < 1$

5	$nu[n]$	$\frac{z^{-1}}{(1-z^{-1})^2}$	$\|z\| > 1$
6	$-nu[-n-1]$	$\frac{z^{-1}}{(1-z^{-1})^2}$	$\|z\| < 1$
7	$n^2u[n]$	$\frac{z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$	$\|z\| > 1$
8	$-n^2u[-n-1]$	$\frac{z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$	$\|z\| < 1$
9	$n^3u[n]$	$\frac{z^{-1}(1+4z^{-1}+z^{-2})}{(1-z^{-1})^4}$	$\|z\| > 1$
10	$-n^3u[-n-1]$	$\frac{z^{-1}(1+4z^{-1}+z^{-2})}{(1-z^{-1})^4}$	$\|z\| < 1$
11	$a^nu[n]$	$\frac{1}{1-az^{-1}}$	$\|z\| > \|a\|$
12	$-a^nu[-n-1]$	$\frac{1}{1-az^{-1}}$	$\|z\| < \|a\|$
13	$na^nu[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| > \|a\|$
14	$-na^nu[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| < \|a\|$
15	$n^2a^nu[n]$	$\frac{az^{-1}(1+az^{-1})}{(1-az^{-1})^3}$	$\|z\| > \|a\|$
16	$-n^2a^nu[-n-1]$	$\frac{az^{-1}(1+az^{-1})}{(1-az^{-1})^3}$	$\|z\| < \|a\|$
17	$\cos(\omega_0 n)u[n]$	$\frac{1-z^{-1}\cos(\omega_0)}{1-2z^{-1}\cos(\omega_0)+z^{-2}}$	$\|z\| > 1$
18	$\sin(\omega_0 n)u[n]$	$\frac{z^{-1}\sin(\omega_0)}{1-2z^{-1}\cos(\omega_0)+z^{-2}}$	$\|z\| > 1$
19	$a^n\cos(\omega_0 n)u[n]$	$\frac{1-az^{-1}\cos(\omega_0)}{1-2az^{-1}\cos(\omega_0)+a^2z^{-2}}$	$\|z\| > \|a\|$
20	$a^n\sin(\omega_0 n)u[n]$	$\frac{az^{-1}\sin(\omega_0)}{1-2az^{-1}\cos(\omega_0)+a^2z^{-2}}$	$\|z\| > \|a\|$

Relationship to Laplace transform

The Bilinear transform is a useful approximation for converting continuous time filters (represented in Laplace space) into discrete time filters (represented in z space), and vice versa. To do this, you can use the following substitutions in $H(s)$ or $H(z)$:

$s = \frac{2}{T}\frac{z-1}{z+1}$ from Laplace to z (Tustin transformation);

$z = \frac{2+sT}{2-sT}$ from z to Laplace.

Through the bilinear transformation, the complex s-plane (of the Laplace transform) is mapped to the complex z-plane (of the z-transform. While this mapping is (necessarily) nonlinear, it is useful in that this particular mapping maps the entire $j\Omega$ axis of the s-plane onto the unit circle in the z-plane. As such, the Fourier transform (which is the Laplace transform evaluated on the $j\Omega$ axis (assuming that the Fourier transform exists, i.e. that the $j\Omega$ axis is in the region of convergence of the Laplace transform) onto the the discrete-time Fourier transform (which is the unit circle in the z-plane).

Relationship to Fourier transform

The Z-transform is a generalization of the discrete-time Fourier transform (DTFT). The DTFT can be found by evaluating the Z-transform $X(z)$ at $z = e^{j\omega}$ or, in other words, evaluated on the unit circle. In order to determine the frequency response of the system the Z-transform must be evaluated on the unit circle, meaning that the system's region of convergence must contain the unit circle. Otherwise, the DTFT of the system does not exist.

Linear constant-coefficient difference equation

The linear constant-coefficient difference (LCCD) equation is a representation for a linear system based on the autoregressive moving-average equation.

$$\sum_{p=0}^{N} y[n-p]\alpha_p = \sum_{q=0}^{M} x[n-q]\beta_q$$

Both sides of the above equation can be divided by α_0, if it is not zero, normalizing $\alpha_0 = 1$ and the LCCD equation can be written

$$y[n] = \sum_{q=0}^{M} x[n-q]\beta_q - \sum_{p=1}^{N} y[n-p]\alpha_p.$$

This form of the LCCD equation is favorable to make it more explicit that the "current" output $y[n]$ is a function of past outputs $y[n-p]$, current input $x[n]$, and previous inputs $x[n-q]$.

Transfer function

Taking the Z-transform of the above equation (using linearity and time-shifting laws) yields

$$Y(z)\sum_{p=0}^{N} z^{-p}\alpha_p = X(z)\sum_{q=0}^{M} z^{-q}\beta_q$$

and rearranging results in

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{q=0}^{M} z^{-q}\beta_q}{\sum_{p=0}^{N} z^{-p}\alpha_p} = \frac{\beta_0 + z^{-1}\beta_1 + z^{-2}\beta_2 + \cdots + z^{-M}\beta_M}{\alpha_0 + z^{-1}\alpha_1 + z^{-2}\alpha_2 + \cdots + z^{-N}\alpha_N}.$$

Zeros and poles

From the fundamental theorem of algebra the numerator has M roots (corresponding to zeros of H) and the denominator has N roots (corresponding to poles). Rewriting the transfer function in terms of poles and zeros

$$H(z) = \frac{(1 - q_1 z^{-1})(1 - q_2 z^{-1})\cdots(1 - q_M z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})\cdots(1 - p_N z^{-1})}$$

where q_k is the k^{th} zero and p_k is the k^{th} pole. The zeros and poles are commonly complex and when plotted on the complex plane (z-plane) it is called the pole-zero plot.

In addition, there may also exist zeros and poles at $z = 0$ and $z = \infty$. If we take these poles and zeros as well as multiple-order zeros and poles into consideration, the number of zeros and poles are always equal.

By factoring the denominator, partial fraction decomposition can be used, which can then be transformed back to the time domain. Doing so would result in the impulse response and the linear constant coefficient difference equation of the system.

Output response

If such a system $H(z)$ is driven by a signal $X(z)$ then the output is $Y(z) = H(z)X(z)$. By performing partial fraction decomposition on $Y(z)$ and then taking the inverse Z-transform the output $y[n]$ can be found. In practice, it is often useful to fractionally decompose $\frac{Y(z)}{z}$ before multiplying that quantity by z to generate a form of $Y(z)$ which has terms with easily computable inverse Z-transforms

Chapter- 10

Advanced Z-Transform & Two-Sided Laplace Transform

Advanced Z-transform

In mathematics and signal processing, the **advanced Z-transform** is an extension of the Z-transform, to incorporate ideal delays that are not multiples of the sampling time. It takes the form

$$F(z,m) = \sum_{k=0}^{\infty} f(kT+m)z^{-k}$$

where

- T is the sampling period
- m (the "delay parameter") is a fraction of the sampling period $[0,T)$.

It is also known as the **modified Z-transform**.

The advanced Z-transform is widely applied, for example to accurately model processing delays in digital control.

Properties

If the delay parameter, m, is considered fixed then all the properties of the Z-transform hold for the advanced Z-transform.

Linearity

$$\mathcal{Z}\left\{\sum_{k=1}^{n} c_k f_k(t)\right\} = \sum_{k=1}^{n} c_k F(z,m).$$

Time shift

$$\mathcal{Z}\left\{u(t-nT)f(t-nT)\right\} = z^{-n}F(z,m).$$

Damping

$$\mathcal{Z}\left\{f(t)e^{-a\,t}\right\} = e^{-a\,m}F(e^{a\,T}z,m).$$

Time multiplication

$$\mathcal{Z}\left\{t^y f(t)\right\} = \left(-Tz\frac{d}{dz}+m\right)^y F(z,m).$$

Final value theorem

$$\lim_{k=\infty} f(kT+m) = \lim_{z=1}(1-z^{-1})F(z,m).$$

Example

Consider the following example where $f(t) = \cos(\omega t)$

$$\begin{aligned} F(z,m) =& \mathcal{Z}\left\{\cos\left(\omega\left(kT+m\right)\right)\right\} \\ =& \mathcal{Z}\left\{\cos(\omega kT)\cos(\omega m) - \sin(\omega kT)\sin(\omega m)\right\} \\ =& \cos(\omega m)\mathcal{Z}\left\{\cos(\omega kT)\right\} - \sin(\omega m)\mathcal{Z}\left\{\sin(\omega kT)\right\} \\ =& \cos(\omega m)\frac{z\left(z-\cos(\omega T)\right)}{z^2-2z\cos(\omega T)+1} - \sin(\omega m)\frac{z\sin(\omega T)}{z^2-2z\cos(\omega T)+1} \\ =& \frac{z^2\cos(\omega m) - z\cos(\omega(T-m))}{z^2-2z\cos(\omega T)+1} \end{aligned}$$

.

If $m = 0$ then $F(z,m)$ reduces to the Z-transform

$$F(z,0) = \frac{z^2 - z\cos(\omega T)}{z^2-2z\cos(\omega T)+1}$$

which is clearly just the Z-transform of $f(t)$.

Two-sided Laplace transform

In mathematics, the **two-sided Laplace transform** or **bilateral Laplace transform** is an integral transform closely related to the Fourier transform, the Mellin transform, and the ordinary or one-sided Laplace transform. If $f(t)$ is a real or complex valued function of the real variable t defined for all real numbers, then the two-sided Laplace transform is defined by the integral

$$\mathcal{B}\{f(t)\} = F(s) = \int_{-\infty}^{\infty} e^{-st} f(t)\, dt.$$

The integral is most commonly understood as an improper integral, which converges if and only if each of the integrals

$$\int_{0}^{\infty} e^{-st} f(t)\, dt, \quad \int_{-\infty}^{0} e^{-st} f(t)\, dt$$

exists. There seems to be no generally accepted notation for the two-sided transform, the $\mathcal{B}$used here recalls "bilateral". The two-sided transform used by some authors is

$$\mathcal{T}\{f(t)\} = s\mathcal{B}\{f\} = sF(s) = s\int_{-\infty}^{\infty} e^{-st} f(t)\, dt.$$

In pure mathematics the argument t can be any variable, and Laplace transforms are used to study how Differential operators transform the function.

In science and engineering applications, the argument t often represents time (in seconds), and the function $f(t)$ often represents a signal or waveform that varies with time. In these cases, the signals are transformed by filters, that work like a mathematical operator, but with a restriction. They have to be causal, which means that the output in a given time t cannot depend of input values in higher values of t.

When working with functions of time, $f(t)$ is called the **time domain** representation of the signal, while $F(s)$ is called the **frequency domain** representation. The inverse transformation then represents a *synthesis* of the signal as the sum of its frequency components taken over all frequencies, whereas the forward transformation represents the *analysis* of the signal into its frequency components.

Relationship to other integral transforms

If $u(t)$ is the Heaviside step function, equal to zero when t is less than zero, to one-half when t equals zero, and to one when t is greater than zero, then the Laplace transform $\mathcal{L}$may be defined in terms of the two-sided Laplace transform by

$$\mathcal{L}\left\{f(t)\right\} = \mathcal{B}\left\{f(t)u(t)\right\}.$$

On the other hand, we also have

$$\left\{\mathcal{B}f\right\}(s) = \left\{\mathcal{L}f(t)\right\}(s) + \left\{\mathcal{L}f(-t)\right\}(-s)$$

so either version of the Laplace transform can be defined in terms of the other.

The Mellin transform may be defined in terms of the two-sided Laplace transform by

$$\left\{\mathcal{M}f\right\}(s) = \left\{\mathcal{B}f(e^{-x})\right\}(s)$$

and conversely we can get the two-sided transform from the Mellin transform by

$$\left\{\mathcal{B}f\right\}(s) = \left\{\mathcal{M}f(-\ln x)\right\}(s).$$

The Fourier transform may also be defined in terms of the two-sided Laplace transform; here instead of having the same image with differing originals, we have the same original but different images. We may define the Fourier transform as

$$\mathcal{F}\left\{f(t)\right\} = F(s = i\omega) = F(\omega).$$

Note that definitions of the Fourier transform differ, and in particular

$$\left\{\mathcal{F}f\right\} = F(s = i\omega) = \frac{1}{\sqrt{2\pi}}\left\{\mathcal{B}f\right\}(s)$$

is often used instead. In terms of the Fourier transform, we may also obtain the two-sided Laplace transform, as

$$\left\{\mathcal{B}f\right\}(s) = \left\{\mathcal{F}f\right\}(-is).$$

The Fourier transform is normally defined so that it exists for real values; the above definition defines the image in a strip $a < \Im(s) < b$ which may not include the real axis.

The moment-generating function of a continuous probability density function $f(x)$ can be expressed as $\left\{\mathcal{B}f\right\}(-s)$.

Properties

It has basically the same properties of the unilateral transform with an important difference

Properties of the unilateral Laplace transform

	Time domain	unilateral-'s' domain	bilateral-'s' domain
Differentiation	$f'(t)$	$sF(s) - f(0)$	$sF(s)$
Second Differentiation	$f''(t)$	$s^2F(s) - sf(0) - f'(0)$	$s^2F(s)$

To use the bilateral transform is equivalent to assume null initial conditions. Therefore it is more suitable than the unilateral for calculating transfer functions from the differential equations, or when looking for an easy particular solution.

Causality

Bilateral transforms don't respect causality. They make sense when applied over generic functions but when working with functions of time (signals) unilateral transforms are preferred.