Numerical Transforms

Author(s): R. N. Bracewell

Source: *Science*, May 11, 1990, New Series, Vol. 248, No. 4956 (May 11, 1990), pp. 697-704

Published by: American Association for the Advancement of Science

Stable URL: https://www.jstor.org/stable/2874133

Articles

Numerical Transforms

R. N. BRACEWELL

Numerical computation of transforms is now widely practiced in science and industry and has been revolutionized by the development of fast transforms that make feasible computing projects that once could not be contemplated. The article discusses the significance of transforms in numerical work, defines the modern forms of several common transforms and their inverses, gives examples, and describes and gives references to methods of numerical evaluation.

THE BEST KNOWN TRANSFORMS ARE THOSE NAMED FOR Laplace, Fourier, Hilbert, Hankel, Mellin, and Abel, all of which continue to attract contributions to the mathematical literature. Years ago their use was mainly analytic. Indeed, the theory of Fourier series developed hand in hand with modern analysis, but with the growth of automatic computing new applications have developed. Although the Fourier transform was always used for computing, at least by a few practitioners, today computing by means of the fast Fourier transform (FFT) is a widespread activity in all technical fields by very large numbers of technicians, engineers, and scientists, including many in biology and medicine.

Speed is of the essence in certain applications of computers. An image that takes several minutes to compute becomes more interesting when it can be completed and displayed in a time as short as a heartbeat, a chemical reaction, or an industrial process. Besides discussing the state of numerical evaluation of the well-known transforms, I will treat the Radon transform, because of its relation to the exploding field of tomography, and the Hartley transform, because it is a new development in spectral analysis related to the FFT. Several other important transforms are mentioned briefly in the available space.

The Transform Concept

What is a transform? Think of a function $f(x)$, for example, $\exp(-|x|)$, on which some explicit operation $\mathbf{T}$ is carried out, leading to another function $\mathbf{T}\{f(x)\}$. Call this other function $F()$. We say that $F()$ is the such-and-such transform of the function $f()$ that we first thought of. The symbols for the independent variables are deliberately omitted in order to focus attention on the idea that the functional shape $F()$ derives from the original shape $f()$, not from either the value of the independent variable or its identity. With the Fourier transform, the operation $\mathbf{T}$ is as follows. "Multiply the function $f(x)$ by $\exp(-i2\pi sx)$, where s is the transform variable, and integrate with respect to x from $-\infty$ to ∞." Applying this operation to $f(x) = \exp(-|x|)$, we find that $\mathbf{T}\{f(x)\} = F(s) = 2/[1 + (2\pi s)^2]$, which is the Fourier transform of $\exp(-|x|)$.

It is apparent that any particular value of $F(s)$, for example, $F(2)$ which equals 0.0126, takes into account the whole range of x; that is, the value depends on the shape of $f()$ as a whole, not on any single point. Thus the Fourier operation is quite unlike the operation that converts $f(x) = \exp(-|x|)$ to $\sin[\exp(-|x|)]$. The outcome of this latter operation is referred to as a "function of a function," and each of the resulting values depends on only a single value of x, not on the whole shape of $f()$.

With some transforms—the Abel transform is an example—each transform value depends on only a part of, not all of, $f()$; with other transforms the transform variable does not necessarily have a different identity (as s is different from x) but may have the same identity (Hilbert transform).

All the transforms dealt with here are linear transforms, which are the commonest type; they all obey the superposition rule that $\mathbf{T}\{f_1(x) + f_2(x)\} = \mathbf{T}\{f_1(x)\} + \mathbf{T}\{f_2(x)\}$, for any choice of the given functions $f_1(x)$ and $f_2(x)$. An example of a nonlinear transformation is provided by $\mathbf{T}\{f(x)\} = a + bf(x)$, as may be tested by reference to the superposition definition; clearly the term "linear" in linear transform does not have the same meaning as in Cartesian geometry.

Discrete Transforms

Before defining the main transforms succinctly by their operations $\mathbf{T}$, all of which involve integration over some range, it is worth commenting on a numerical aspect. One could take the point of view, as is customary with numerical integration, that the desired integral is an entity in its own right; that the integral may on occasion be subject to precise evaluation in analytic terms, as with $F(s) = 2/[1 + (2\pi s)^2]$; and that, if numerical methods are required, a sum will be evaluated that is an approximation to the desired integral. One would then discuss the desired degree of approximation and how to reach it. This is quite unlike the customary way of thinking about the discrete Fourier transform. What we evaluate is indeed a sum, but we regard the sum as precise and not as an approximation to an integral. There are excellent reasons for this. Meanwhile, the important thing to realize is that there are both a Fourier transform and a discrete Fourier transform, each with its own definition:

$$\text{Fourier transform: } F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi sx}dx \qquad (1)$$

$$\text{Discrete Fourier transform: } F(\nu) = \frac{1}{N}\sum_{\tau=0}^{N-1} f(\tau)e^{-i2\pi\nu\tau/N} \qquad (2)$$

The word "discrete" is used in antithesis to "continuous" and in the cases discussed here means that an independent variable assumes integer values. In order to understand the discrete Fourier transform, which is exclusively what we compute when in numerical

The author is professor of electrical engineering and, by courtesy, of computer science at Stanford University, Stanford, CA 94305.

mode, it is best to forget the Fourier integral and start afresh. Instead of starting with a complex function $f(x)$ that depends on the continuous real variable x, we start with N data (complex in general, but often real) indexed by an integer serial number τ (like time) that runs from 0 to $N - 1$. In the days when FORTRAN did not accept zero as a subscript, summation from $\tau = 0$ caused much schizophrenia, but the mathematical tradition of counting from zero prevailed and is now unanimous. The quantity τ can be thought of as time that is counted in units starting from time zero, in cases where $f()$ is a wave form, as it often is. Clearly, N samples can never fully represent $\exp(-|x|)$, for two reasons: (i) the samples take no account of the function where x exceeds some finite value, and (ii) no account is taken of fine detail between the samples. Nevertheless, one may judge that, for a given particular purpose, 100 samples will suffice and the confidence to judge may be bolstered by trying whether 200 samples significantly affects the purpose in hand. Numerical intuition as developed by hand calculation has always been a feature of mathematical work but was regarded as weak compared with physical intuition. Nowadays, however, numerical intuition is so readily acquired that it has become a matter of choice whether to attack questions about the size of N by traditional analytic approaches. A new mix of tools from analysis, finite mathematics, and numerical analysis is evolving.

The discrete transform variable ν reminds us of frequency. If τ is thought of as time measured in integral numbers of seconds, then ν is measured in cycles per second and is indeed like frequency (cycles per second or hertz), but not exactly. It is ν/N that gives correct frequencies in hertz, and then only for $\nu \leq N/2$. Where ν exceeds $N/2$, we encounter a domain where the discrete approach conflicts with the continuous. When the Fourier transform is evaluated as an integral, it is quite ordinary to contemplate negative values of s, and a graph of $F(s)$ will ordinarily have the vertical $s = 0$ axis in the middle, giving equal weight to positive and negative "frequencies." (The unit of s is always cycles per unit-of-x; if x is in meters, s will be a spatial frequency in cycles per meter; if x is in seconds, s will be a temporal frequency in cycles per second, or hertz). However, the discrete Fourier transform, as conventionally defined, explicitly requires the transform variable ν to range from 0 to $N - 1$, not exhibiting negative values at all. There is nothing wrong with that, but persons coming from continuous mathematics or from physics may like to know that, when ν is in the range from $N/2$ to $N - 1$, the quantities $N - \nu$ correspond to the negative frequencies familiar to them as residing to the left of the origin on the frequency axis. This is because the discrete transform is periodic in ν, with period N. If one wishes to adopt a private definition with ν ranging from $-N/2 + 1$ to $N/2$, the formulas still work.

Just as in the familiar Fourier series the first term a_0 represents the dc or zero frequency value, so the first term $F(0)$ of the discrete Fourier transform is the average of the N data values. This is the reason for the factor $1/N$ in front of the summation sign in Eq. 2, and the factor must be remembered when checking. In practical computing it is efficient to combine the factor $1/N$ with other factors such as calibration factors and graphical scale factors, which are applied later at the display stage.

How to decide whether the discrete Fourier transform is an adequate approximation to the Fourier transform is a very interesting question. In the first place, who says it is an approximation? If I am studying cyclicity in animal populations, perhaps seasonal

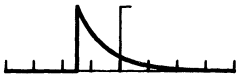
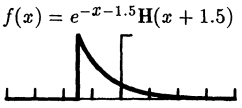
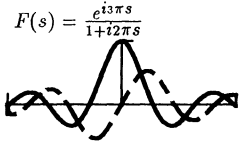
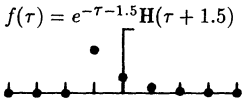
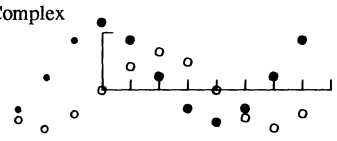
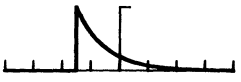
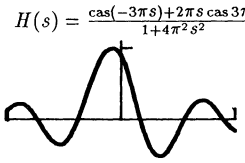
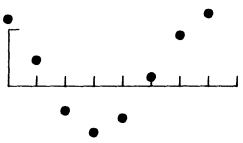
Name of transform	Nature of function domain variable	Example of function	Nature of transform variable	Nature of transform	Example of transform	Defining formula and the inverse
Laplace	Continuous, real	$f(x) = e^{-x-1.5}\mathbf{H}(x+1.5)$ 	Continuous, complex	Complex	$\frac{e^{1.5s}}{1+s}$, $-1 < \mathrm{Re}\ s$	$F_{\mathrm{L}}(s) = \int_{-\infty}^{\infty} f(x)e^{-sx}dx$ $\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} F_{\mathrm{L}}(s)e^{xs}ds$
Fourier	Continuous, real	$f(x) = e^{-x-1.5}\mathbf{H}(x+1.5)$ 	Continuous, real	Complex	$F(s) = \frac{e^{i3\pi s}}{1+i2\pi s}$ 	$F(s) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi sx}dx$ $f(x) = \int_{-\infty}^{\infty} F(s)e^{i2\pi xs}ds$
Discrete Fourier	Discrete, real	$f(\tau) = e^{-\tau-1.5}\mathbf{H}(\tau+1.5)$ 	Discrete, real	Complex		$F(\nu) = N^{-1}\sum_{\tau=0}^{N-1} f(\tau)e^{-i2\pi\nu\tau/N}$ $f(\tau) = \sum_{\nu=0}^{N-1} F(\nu)e^{i2\pi\tau\nu/N}$
Hartley	Continuous, real	$f(x) = e^{-x-1.5}\mathbf{H}(x+1.5)$ 	Continuous, real	Real	$H(s) = \frac{\mathrm{cas}(-3\pi s)+2\pi s\,\mathrm{cas}\,3\pi s}{1+4\pi^2 s^2}$ 	$H(s) = \int_{-\infty}^{\infty} f(x)\,\mathrm{cas}\,2\pi sx\,dx$ $f(x) = \int_{-\infty}^{\infty} H(s)\,\mathrm{cas}\,2\pi xs\,ds$
Discrete Hartley	Discrete, real	$f(\tau) = e^{-\tau-1.5}\mathbf{H}(\tau+1.5)$ 	Discrete, real	Real		$H(\nu) = N^{-1}\sum_{0}^{N-1} f(\tau)\,\mathrm{cas}(2\pi\nu\tau/N)$ $f(\tau) = \sum_{\nu=0}^{N-1} H(\nu)\,\mathrm{cas}(2\pi\nu\tau/N)$

Fig. 1. Transform definitions, inverses, and examples. Ticks are at unit spacing.

influence on bird migration, I may start with 365 reports of how many birds were seen each day of the year. In such a case, and in many other cases, discrete data mean that the integrals, even though convenient, are themselves the approximations. The discrete Fourier transform, given N equispaced data, is an entity in its own right. Nevertheless, if it is open to us to choose N, we may make a choice that is too large or too small. Among the bad consequences are slow computing, unwanted sensitivity to measurement error, and aliasing (the spurious appearance of frequencies that are present in the samples but not in the world sampled).

The Best Known Transforms

In Fig. 1 terse definitions of several transforms are presented for reference together with the inversion formulas that enable the original function to be regained from knowledge of the transform. In addition, examples of each transform are presented. Enough examples of discrete versions are given to illustrate the distinct status of the discrete transforms, but for the most part only the original continuous definitions are given for want of any discrete standard.

The Laplace Transform

A long and diverse history (*1*) characterizes the Laplace transform, which was in use long before Laplace, but became known to current generations mainly through its pertinence to the linear differential equations of transient behavior in electricity and heat conduction. Many tough technological problems of electric circuits that arose in connection with telegraphy, submarine cables and wireless, and related industrial process problems of thermal diffusion were cracked around the turn of the century, sometimes by novel methods such as Heaviside's, which were to be fixed subsequently to the satisfaction of academic mathematics by systematic application of the Laplace transform. Heaviside is remembered for stimulating the application of the Laplace transform to convergence of series, for Maxwell's equations, the delta function, the Heaviside layer, impedance, nonconvergent series that are useful for computing, fractional order derivatives and integrals, and operational calculus.

Figure 1 gives, as an example, the Laplace transform of $f(x) = \exp(-x - 1.5)\mathbf{H}(x + 1.5)$. The Heaviside unit step function $\mathbf{H}(x)$ jumps, as x increases, from 0 to 1, the jump being where $x = 0$; one of its uses is as a multiplying factor to allow algebraic expression of functions that switch on. The transform of $f(x)$, which is easy to verify by integration, is $\exp(1.5s)/(1 + s)$; the transform variable s may be complex but must lie among those numbers whose real parts are greater than -1 (otherwise the integral does not exist). It is rather cumbersome to exhibit the complex transform graphically on the complex plane, so an illustration is omitted. To invert the transform requires integration on the complex plane along a semicircular contour with indentations if they are necessary to circumvent points where the integrand goes to infinity (poles). The constant c in the inversion formula is to be chosen to the right of all poles.

To some extent, Laplace transforms were computed numerically but more typically development led to compilations of analytic transforms resembling the tables of integrals (*2, 3*). Programs for

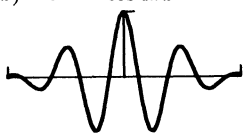
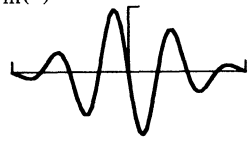
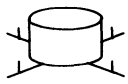
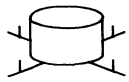
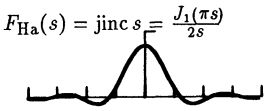
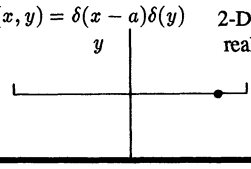
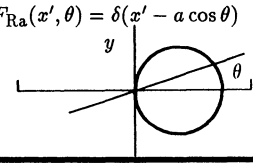
Name of transform	Nature of function domain variable	Example of function	Nature of transform variable	Nature of transform	Example of transform	Defining formula and the inverse
Mellin	Continuous, real	$f(x) = e^{-x}\mathbf{H}(x)$ 	Continuous, complex	Real	s 0 1 2 3 $F_M(s)$ 1 1 2 6	$F_M(s) = \int_0^\infty f(x)x^{s-1}dx$ $f(x) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+\infty} F_M(s)x^{-s}ds$
Hilbert	Continuous, real	$f(x) = e^{-\pi x^2}\cos 4\pi x$ 	Continuous, complex	Real	$F_{Hi}(x) = -e^{-\pi x^2}\sin 4\pi x$ 	$F_{Hi}(x) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{f(x')dx'}{x'-x}$ $f(x) = -\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{F_{Hi}(x')dx'}{x'-x}$
Abel	Continuous, real	$f(r) = 1,\ r < 0.5$ 	Continuous, complex	Real	$F_A(s) = \sqrt{1-s^2},\ s < 1,$ 	$F_A(s) = 2\int_s^\infty \frac{f(r)rdr}{\sqrt{r^2-s^2}}$ $f(r) = -\frac{1}{\pi}\int_r^\infty \frac{F'_A(x)dx}{\sqrt{x^2-r^2}}$
Hankel	Continuous, real	$f(r) = 1,\ r < 0.5$ 	Continuous, complex	Real	$F_{Ha}(s) = \text{jinc}\, s = \frac{J_1(\pi s)}{2s}$ 	$F_{Ha}(s) = 2\pi\int_0^\infty f(r)J_0(2\pi sr)rdr$ $f(r) = 2\pi\int_0^\infty F_{Ha}(s)J_0(2\pi rs)sds$
Radon	2-D, continuous, real	$f(x,y) = \delta(x-a)\delta(y)$ 	2-D, continuous, real	Real	$F_{Ra}(x',\theta) = \delta(x' - a\cos\theta)$ 	$F_{Ra}(x',\theta) = \int_{-\infty}^{\infty} f(x,y)dy'$ Complicated inverse

Fig. 1 (cont.)

deriving the Laplace transform of the impulse response from electrical networks given diagrammatically are also available. Consequently, it is hardly ever necessary to derive Laplace transforms analytically today. The analytic solution of transients in electric circuits, a subject traditionally used for sharpening the minds of electrical engineers, is obsolescent because impulse responses and transfer functions have been concisely published (*3, 4*). Furthermore, the advent of integrated circuits has meant that inductance is seldom included in new designs and that circuits containing more than two or three elements have become less common. Mature programs are also available for step-by-step integration of circuit differential equations.

On the numerical side, the Laplace transform has also been largely eroded by use of the Fourier transform. This is because frequency ω is a real quantity mathematically, and it ought to be possible to compute the behavior of an electrical, acoustical, or mechanical system without reference to the complex frequency $s = j\omega$. Certainly, the Laplace transform is computable over its strip of convergence from any single slice therein. Nevertheless, practitioners of control theory find it convenient to think on the complex plane of s in terms of poles and zeroes that are off the real frequency axis (using Bode diagrams and root-locus plots), and theirs is one tradition that keeps the complex plane alive; the convenience stems from the fact that the Laplace transform is analytic and is thus specifiable by its poles and isolated zeroes. There are problems that formerly were handled by the Laplace transform, with strict attention being paid to the strip of convergence, because the Fourier integral did not converge. These situations are now universally handled by Fourier methodology with the aid of delta function notation for impulses and their derivatives and no longer call for special treatment. When it comes to discrete computing, the impulse, and its associated spectrum reaching to indefinitely large frequencies, may in any case be forgotten. Thus, it has been wondered (*5*, p. 551) "whether the Laplace transform will keep its place in the standard mathematical methods course for very much longer," but it will never die out; a new balance between curricular segments will be struck.

Why Transforms Are Useful

Many problems can be posed in the form of a differential equation (or a difference equation, or an integral equation, or an integrodifferential equation) that has to be solved for some wanted function subject to stated boundary conditions or initial conditions. Laplace's equation in three dimensions describes the potential distribution set up by an array of electric charges, and the diffusion equation describes the heat flow distribution set up by a given distribution of heat. By applying a transformation such as the Laplace or Fourier to each term of such an equation, we arrive at a new equation that describes the transform rather than the original wanted function. The interesting thing about this procedure is that the new equation may be simpler, sometimes solvable just by algebra. We solve that equation for the transform of the solution and then invert. Not all differential equations simplify in this way; those that do are characterized by linearity and coordinate invariance (such as time invariance), and the presence of these characteristics in nature is responsible for a good deal of the numerical activity with transforms. Transfer functions, such as the frequency response curves of amplifiers, are corresponding manifestations of these same characteristics. The passage of a speech wave form through an amplifier is described by a differential equation that may be hard to solve, but, having used a transform to go to the frequency domain, we apply the transfer function, frequency by frequency, by complex multiplication to get the transform of the output. Then retransforming gives the output wave form.

There is also a differential equation that describes the bending of a beam under the influence of a load distribution, which may be thought of as analogous to an input wave form, while the curve of deflection is the output wave form. Although Hooke's law, the first of the linear laws, may apply, we do not use transform methods. If we analyze the load distribution into spatially sinusoidal components and find the bending response to each component and linearly sum the responses, we will get the desired shape of the bent beam but there is no transfer function to facilitate getting the individual responses by simple algebra. The reason is that we have linearity but not space invariance—if we shift the load, the response does not shift correspondingly without change of shape; a sinusoidal load does not produce sinusoidal deflection. If, on the contrary, we delay the input to an amplifier or a vibratory mechanical system, the response is correspondingly delayed but is unchanged as to shape; furthermore, a sinusoidal input produces a sinusoidal output.

The Fourier and Hartley Transforms

Figure 1 illustrates by example that the Fourier transform in general is a complex function of the real transform variable s; consequently, two curves must be drawn, one for the real part (solid line) and one for the imaginary part (broken line). The example $f(\tau)$ for the discrete Fourier transform is based on samples of the previous $f(x)$. Imaginary values of the discrete transform $F(\nu)$ are shown as open circles. Three features may be noted: (i) no matter how closely samples are spaced, some detail can be missed; (ii) no outlying parts beyond a finite range are represented; and (iii) the indexing convention 0 to $N - 1$ has the effect of cutting off the left side of $F(s)$, translating it to the right, and reconnecting it. To convey the nature of this third comment the points for $\tau > N/2$ have been copied back on the left.

The Hartley transform differs from the Fourier transform in that the kernel is cas $2\pi sx$ instead of $\exp(-i2\pi sx)$. The cas function, which was introduced by Hartley (*6*), is defined by $\text{cas}\, x = \cos x + \sin x$ and is simply a sinusoid of amplitude $\sqrt{2}$ shifted one-eighth of a period. The consequences of the change are that the Hartley transform is real rather than complex and that the transformation is identical to the inverse transformation. As may be obvious from the graphical example, the Hartley transform contains all the information that is in the Fourier transform and one may move freely from one to the other using the relations

$$H(s) = \text{Re}\, F(s) - \text{Im}\, F(s)$$

and

$$2F(s) = H(s) + H(N - s) - iH(s) + iH(N - s)$$

The convenience that arises from familiarity with complex algebra when one is thinking about transforms loses its value in computing. What one thinks of as one complex product still means four real multiplications to computer hardware, which must be instructed accordingly.

The Fast Fourier Transform

In about 1805, C. F. Gauss, who was then 28, was computing orbits by a technique of trigonometric sums equivalent to today's discrete Fourier synthesis. To get the coefficients from a set of a dozen regularly spaced data, he could if he wished explicitly implement the formula that we recognize as the discrete Fourier

transform. To do this he would multiply the N data values $f(\tau)$ by the weighting factors $\exp(-i2\pi\nu\tau)$, sum the products, and repeat these N multiplications N times, once for each value of ν. But he found that, in the case where N is a composite number with factors such that $N = n_1 n_2$, a computing advantage was gained by partitioning the data into n_2 sets of n_1 terms. Where N was composed of three or more factors, a further advantage could be obtained. Gauss wrote (*7*, p. 307) "illam vero methodum calculi mechanici taedium magis minuere, praxis tentatem docebit." He refers to diminishing the tedium of mechanical calculation, as practice will teach him who tries. This factoring procedure, usually into factors of 2, is the basis of the FFT algorithm, which is explained in many textbooks (*8–12*) and is available in software packages. The fast method burst on the world of signal analysis in 1965 and was for a time known as the Cooley-Tukey algorithm (*13, 14*), but, as the interesting history (*15*) of prior usage in computing circles became known, the term FFT became universal.

Most FFT programs in use take advantage of factors by adopting a choice of N that is some power P of 2, that is, $N = 2^P$. People with 365 data points simply append sufficient zeros to reach $512 = 2^9$ values. This might seem wasteful but an attendant feature is the closer spacing of the resulting transform samples, which is advantageous for visual presentation. Perhaps one could do the job faster, say, by factoring into 5×73. There are fast algorithms for 5 points and for many other small primes, but not for 73, as far as I know; it is simply not practical to store and select from lots of special programs for peculiar values of N. On the other hand, a significant speed advantage is gained if one elects more rigidity rather than more flexibility, tailors one's data collection to a total of 4^P values, and uses what is referred to as a radix-4 program. Because $1024 = 4^5$, the radix-4 approach is applicable to $N = 1024$ data samples (or to 256 for example, but not to 512 unless one appends 512 zeros, which has one desirable effect, that a polygon drawn through the then twice as closely spaced transform samples is much smoother).

Much practical technique is involved. When the sampled input is not naturally zero outside the sampled range, packing with zeros introduces unwarranted discontinuities whose effects on the transform, such as overshoot and negative-going oscillation, may be undesirable. Packing with plausible (but unobserved) data can eliminate the undesired artifacts and is probably practiced in more cases than are admitted to. Investigators often mitigate the effects of discontinuities in the data by multiplying by a tapering factor; they should then explain that they value freedom from negatives more than accuracy of amplitude values of spectral peaks or than resolution of adjacent peaks.

The FFT is carried out in P successive stages, each entailing N multiplications, for a total of NP. When NP is compared with N^2 (as for direct implementation of the defining formula), the savings are substantial for large N and make operations feasible, especially on digital images, that would otherwise be unthinkable.

The Fast Hartley Algorithm

When data values are real, which is very commonly the case, the Fourier transform is nevertheless complex. The N transform values are also redundant (if you have the results for $0 \le \nu \le N/2$, you can deduce the rest). This inefficiency was originally dealt with by the introduction of a variety of efficient unilateral algorithms, which transformed one way only but in half the time of the FFT, and can now also be handled with calling programs such as REALFT (*12*) or by dealing with the Hartley transform, which for real data is real and is not redundant. For every variant of the FFT, such as radix-4 (*16, 17*), prime-factor partitioning, parallel-processed vectors (*18*), two-dimensional (*19*), in-place computation without bit reversal (scrambling) (*20*), and so forth, there is a corresponding Hartley version. The Hartley transform is elegant and clean and takes you to the other domain, regardless of which one you are in currently. Fast Hartley algorithms have been published in BASIC (*16, 21*), FORTRAN (*16*), PASCAL (*22*), and C (*23*) programs; versions in assembler languages exist; and various supercomputer libraries contain versions. Resident programs contain a copyright statement (*24*).

When a Hartley transform is obtained, there may be a further step required to get to the more familiar complex Fourier transform. The time taken is always negligible, but, even so, the step is usually unnecessary. The reason is that, although we are accustomed to thinking in terms of complex quantities for convenience, it is never obligatory to do so. As a trivial example, suppose we want the power spectrum, which is defined by $P(\nu) = [\text{Re } F(\nu)]^2 + [\text{Im } F(\nu)]^2$. If we already have the Hartley transform $H(\nu)$, then it is not necessary to move first to the complex plane and then get the power spectrum; the desired result is obtained directly as $\{[H(\nu)]^2 + [H(N - \nu)]^2\}/2$. Likewise, phase $\phi(\nu)$, which is required much less often than $P(\nu)$, is defined by $\tan \phi(\nu) = \text{Im } F(\nu)/\text{Re } F(\nu)$. Again, one can get phase directly from $\tan[\phi(\nu) + \pi/4] = H(N - \nu)/H(\nu)$, thus circumventing the further step that would be necessary to go via the well-beaten path of real and imaginary parts.

The encoding of phase by a real transform has added a physical dimension to the interest of the Hartley transform, which has now been constructed in the laboratory with light and microwaves (*25–28*) and has suggested a new sort of hologram.

The Mellin Transform

The vast majority of transform calculations that are done every day fall into categories that have already been dealt with, and much of what has been said is applicable to the special transforms that remain to be mentioned. The Mellin transform has the property that $F_M(n + 1)$ is the nth moment of $f(x)$ when n assumes integer values 1, 2, 3, ... The special value $F_M(1)$ is the zeroth moment of, or area under, $f(x)$. But the transform variable does not have to be integral or even real, so one can think of the Mellin transform as a sort of interpolate passing through the moment values. When the scale of x is stretched or compressed, for example, when $f(x)$ is changed to $f(ax)$, the Mellin transform becomes $a^{-s} F_M(s)$, a modification that leaves the position of features on the s axis unchanged and is useful in some pattern-recognition problems.

If we plot $f(x)$ on a logarithmic scale of x, a familiar type of distortion results and we have a new function $f(e^{-x})$, whose Laplace transform is exactly the same as the Mellin transform of $f(x)$. An equally intimate relation exists with the Fourier transform. Consequently, the FFT may be applicable in numerical situations. Because of the intimate relation with moments and with spectral analysis, Mellin transforms have very wide application. A specific example is given by the solution of the two-dimensional Laplace equation expressed in polar coordinates, namely,

$$\partial^2 V/\partial r^2 + r^{-1}\partial V/\partial r + r^{-2}\partial^2 V/\partial\theta^2 = 0$$

Multiply each term by r^{s-1} and integrate with respect to r from 0 to ∞. We get

$$d^2 F_M/d\theta^2 + s^2 F_M = 0$$

Solve this for $F_M()$ and invert the transform to get the solution. In this example, a partial differential equation is converted to a simple differential equation by the transform technique.

The Hilbert Transform

As the example in Fig. 1 shows, the Hilbert transform, or quadrature function, of a cosinusoidal wave packet is a similar, but odd, wave form sharing the same envelope. But what do we mean by the envelope of an oscillation that only touches the intuitively conceived envelope at discrete points? The Hilbert transform provides an answer in the form $\{[f(x)]^2 + [f_{\mathrm{Hi}}(x)]^2\}^{1/2}$. The original wave packet reveals its phase at its zero crossings. But what is the phase at intermediate points? The Hilbert transform also supplies an instantaneous phase ϕ in the form $\tan \phi = f_{\mathrm{Hi}}(x)/f(x)$. The operation **T** for the Hilbert transform is simply convolution with $-1/\pi x$. It is known that the Fourier transform of $-1/\pi x$ is i sgn s, where sgn s is 1 for $s > 0$ and -1 for $s < 0$. Therefore, by the convolution theorem, according to which the Fourier transform of a convolution is the product of the separate Fourier transforms, it would seem that a fast Hilbert transform of $f(x)$ could be calculated as follows. Take the FFT of $f(x)$, multiply by i for $0 < \nu < N/2$ and by $-i$ for $N/2 < \nu < N$, put $F(0)$ and $F(N/2)$ to zero, and invert the FFT to obtain the Hilbert transform. This sounds straightforward but the procedure is fraught with peril, for two reasons. The first is that we are proposing to multiply a given function $f(x)$ by $-1/\pi[(x + \text{const})]$ and to integrate from $-\infty$ to ∞, but we are only given N samples. The extremities of $-1/\pi x$ approach zero and have opposite signs, but there is infinite area under these tails, no matter how far out we start. Consequently, we are asking two oppositely signed large numbers to cancel acceptably. But how can we expect satisfaction when the convolving function $-1/\pi x$ is not symmetrically situated about the extremes of the data range? The second reason is that we are asking for similar cancellation in the vicinity of the pole of $1/x$. Experience shows that satisfactory envelopes and phases only result when $f(x)$ is a rather narrow-band function. Under other circumstances an N-point discrete Hilbert transform can be defined and will give valid results free from worries about the infinities of analysis, but the outcome may not suit expectation.

Multidimensional Programs

Work with images involves two dimensions, electrostatics and x-ray crystallography involve three, and fluid dynamics involves four. All these cases can be handled with a one-dimensional FFT subprogram, or a fast Hartley, as follows. Consider an $N \times N$ data array. Take the one-dimensional (1-D) transform of each row and write the N transform values in over the data values. Now take the 1-D transform of each resulting column (*19, 24*). In three and four dimensions the procedure is analogous (*29, 30*). Further simple steps lead to the Hartley transform and to the real and imaginary parts of the Fourier transform if they are wanted, but usually they are not; more often the quadratic content (power spectrum) suffices.

When a 2-D function has circular symmetry, as commonly arises with the response functions of optical instruments, not so much work is required, as explained below in connection with the Hankel transform. Cylindrical symmetry in three dimensions is essentially the same, while spherical symmetry in three dimensions is also referred to below.

The Hankel Transform

In two dimensions, where there is circular symmetry as expressed by a given function $f(r)$, the 2-D Fourier transform is also circularly symmetrical. Call it $F_{\mathrm{Ha}}(s)$. It can be arrived at by taking the full 2-D transform as described earlier, or it can be obtained from a single 1-D Hankel transform as defined in Fig. 1. The inverse transform is identical. There is apparently no opening for the Hartley transform because, in the presence of circular symmetry, the 2-D Fourier transform of real data contains no imaginary part. The kernel for the Hankel transform is a zero-order Bessel function, which is a complication that hampers the FFT factoring approach, but there is an elegant sidestep around this that is explained below in connection with the Abel transform. In three dimensions, under spherical symmetry, a different 1-D transform applies that is defined by

$$4\pi \int_0^\infty f(r)\mathrm{sinc}(2sr)r^2 dr \qquad (3)$$

The inverse transform is identical.

The Abel Transform

Most commonly, although not always, the Abel transform arises when a 2-D function $g(x,y)$ has circular symmetry, as given by $f(r)$. The Abel transform (Fig. 1) then simplifies to

$$F_{\mathrm{A}}(x) = \int_{-\infty}^{\infty} g(x,y)dy$$

In other words, if the given $f(r)$ is represented by a square matrix of suitably spaced samples, then the Abel transform results when the columns are summed. There might not seem to be any future in trying to speed up such a basic operation, apart from the obvious step of summing only half way and doubling. However, when it is remembered that for each of $N^2/8$ matrix elements we have to calculate $(x^2 + y^2)^{1/2}$ to find r, and thence $f(r)$, it gives pause. The alternative is to proceed by equal steps in r; then the oversampling near the x axis is mitigated. But the variable element spacing in a column needs correction by a factor $r/(r^2 - s^2)^{1/2}$, which takes more time to compute than $(x^2 + y^2)^{1/2}$. This is an excellent case for decision by using the millisecond timer found on personal computers. Of course, if many runs are to be made, the factors $r/(r^2 - s^2)^{1/2}$ can be precomputed and the preparation time can be amortized over the successive runs.

With the Abel transform under control, we can now see a way of doing the Hankel transform without having to call up Bessel functions. The Abel, Fourier, and Hankel transforms form a cycle known as the FHA cycle (*31*), so that, if we take the Abel transform and then take the FFT, we get the Hankel transform (Fig. 2). The FFT required will not be complex, except in the extraordinary case of complex 2-D data; consequently, it will in fact be appropriate to use the fast Hartley to get the Hankel transform. Because of symmetry, the result will also be exactly the same as obtained with the FFT, if after taking the FFT we pay no attention to the imaginary parts that have been computed, which should all be zero or close to zero.

The Radon Transform

Consider a set of rotated coordinates (x',y') centered on the (x,y) plane. The expression

$$\int_{-\infty}^{\infty} g(x,y)dy$$

given for the Abel transform, representing a line integral in the y direction at a given value of x, would equal the line integral

$$\int_{-\infty}^{\infty} g(x,y)dy'$$

in the rotated direction provided $g(x,y)$ had circular symmetry as specified for the Abel transform. But, if $g(x,y)$ did not have symmetry, then the line-integral values would depend both on x' and on the direction of integration. The set of integrals with respect to dy' is the Radon transform (*32*) of $g(x,y)$, named after Johann Radon. Such integrals arise in computed x-ray tomography, where a needle beam of x-rays scans within a thin plane section of an organ with a view to determining the distribution of absorption coefficient in that plane. If there are N^2 pixels for which values have to be determined, and since one scan will give N data, at least N different directions of scan spaced $180°/N$ will be needed to acquire enough data to solve for the N^2 unknowns. In practice, more than $2N$ directions are needed to compensate for diminished sample density at the periphery. The computation of a Radon transform is easy; the only tricky part is summing a given matrix along inclined directions. One approach is to rotate all the matrix and interpolate onto a rotated grid, for each direction of scan; but this may be too costly. At the other extreme one sums, without weighting, the matrix values lying within inclined strips, which, independently of inclination, preserve unit width in the direction parallel to the nearer coordinate direction. How coarse the increment inclination angle may be depends on acceptability as judged by the user in the presence of actual data.

The harder problem is to invert the line integral data to retrieve the wanted absorption coefficient distribution. A solution was given by Radon (*33*). Later Cormack (*34, 35*), working in the context of x-ray scanning of a solid object, gave a solution in terms of sums of transcendental functions. Other solutions (*31, 36*) include the modified back-projection algorithm used in computer-assisted tomography scanners (*32, 37*). It works as follows (see Fig. 2). From the projection-slice theorem (*31*) we know that the Fourier transform of the projection P' (or scan) in any one direction is the central cross section or slice S' through the 2-D Fourier transform of the wanted distribution $g(x,y)$. Because the density of polar coordinate samples is inversely proportional to radius in the Fourier transform plane, a simple correction factor followed by an inverse 2-D Fourier transform will yield the solution. But a way was found (*36, 37*), based on this theoretical reasoning, to entirely avoid numerical Fourier transforms. An equivalent correction can be directly applied to each projection P' as a simple convolution with few coefficients, after which the modified scans are accumulated on the (x,y) plane by back-projection to reconstitute $g(x,y)$. Back projection means assigning the projected value at x' to all points of the (x,y) plane that, in the rotated coordinate system, have the abscissa x'. Accumulation means summing the back-projected distributions for all inclination angles.

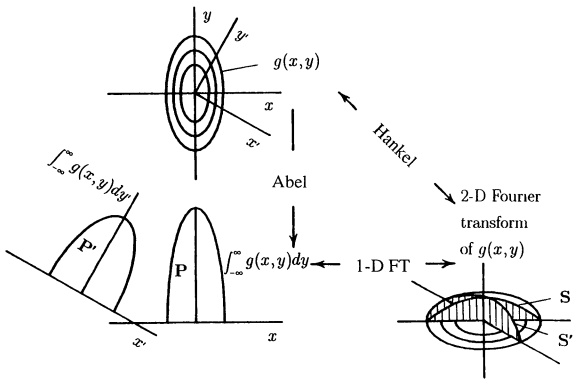


Fig. 2. The projection-slice theorem. A distribution $g(x,y)$ has a projection P, in the y direction, whose 1-D Fourier transform is the slice S through the 2-D Fourier transform of $g(x,y)$. In the presence of circular symmetry where $g(x,y) = f(r)$, the projection P is the Abel transform of $f(r)$. The 1-D Fourier transform of P is the slice S as before, but the slice S is now the Hankel transform of $f(r)$. Thus the Abel, Fourier, and Hankel transforms form a cycle (*31*). The set of projections P' for all inclination angles of the (x',y') coordinates constitutes the Radon transform.

Other Transforms

The Walsh transform. A function defined on the interval [0,1] can be expressed as a sum of sines and cosines of frequency 1,2,3, . . . but can also be expressed as a sum of many other sets of basis functions. Among the alternatives, Walsh functions (*10, 38, 39*) are particularly interesting because they oscillate between values of +1 and −1, a property that is most appropriate to digital circuits and is encountered in telecommunications and radar, though otherwise not common. Furthermore, multiplication by a Walsh function value takes much less time than multiplication by a trigonometric function. Walsh functions, not being periodic, are not to be confused with the square cosine and sine functions $\sqsubset(x) = \mathrm{sgn}(\cos x)$ and $\sqsupset(x) = \mathrm{sgn}(\sin x)$; but on $[0,2\pi]$ they do form a complete set from which any given function can be composed. They are also orthonormal (mutually orthogonal, and integral of square equal to unity, as with Fourier components), which leads to simple relations for both analysis and synthesis. The Walsh (or Walsh-Hadamard) transform has found use in digital signal and image processing and for fast spectral analysis. Fast algorithms are available that use only addition and subtraction and have been implemented in hardware. A vast, enthusiastic literature sprang into existence in the 1970s, a guide to which can be found in the text by Elliott and Rao (*10*).

The z transform. In control theory, in dealing with signals of the form

$$f(t) = \sum_{-\infty}^{\infty} a_n \delta(t - n) \tag{4}$$

and systems whose response to $\delta(t)$ is

$$h(t) = \sum_{0}^{\infty} h_n \delta(t - n) \tag{5}$$

the response $g(t)$ is the convolution integral

$$g(t) = \int_{-\infty}^{\infty} f(t')h(t - t')dt' \tag{6}$$

This response is a series of equispaced impulses whose strengths are given by $\Sigma_i a_i h_{n-i}$, an expression representable in asterisk notation for convolution by $\{g_n\} = \{a_n\} * \{h_n\}$ [in this notation the sequence $\{a_n\}$ sufficiently represents $f(t)$]. For example, a signal {1 1 1 1 1 1 . . .} applied to a system whose impulse response is {8 4 2 1} produces a response {1 1 1 1 1 1 . . .} * {8 4 2 1} = {8 12 14 15 15 15 . . .}. This is the same rule as that which produces the coefficients of the polynomial that is the product of the two polynomials $\Sigma a_n z^n$ and $\Sigma h_n z^n$, as may be verified by multiplying $1 + z + z^2 + z^3 + z^4 + z^5 + \ldots$ by $8 + 4z + 2z^2 + z^3$. The z transform of the sequence {8 4 2 1} is, by one definition, just the polynomial $8 + 4z + 2z^2 + z^3$; more often, one sees $8 + 4z^{-1} + 2z^{-2} + z^{-3}$. If conversely we ask what applied signal would produce the response {8 12 14 15 15 15 . . .}, we get the answer by long division: $(8 + 12z + 14z^2 + 15z^3 + 15z^4 + 15z^5 + \ldots)/(8 + 4z + 2z^2 + z^3)$. Occasionally, one of the polynomials may factor, or simplify, allowing cancellation of factors in the numerator and denominator. For example, the z transform of the infinite impulse response {8 4 2 1 0.5 . . .}, where successive elements are halved, simplifies to $8/(1 - z/2)$. But with measured data, or measured system responses, or both, this never happens and the z notation for a polynomial quotient is then just a waste of ink compared with straightforward

sequence notation such as $\{8\ 12\ 14\ 15\ 15\ 15\ \ldots\} * \{8\ 4\ 2\ 1\ \ldots\}^{-1}$. Whenever sampled data are operated on by a convolution operator (examples would be finite differences, finite sums, weighted running means, finite-impulse-response filters), the z transform of the outcome is expressible as a product of z transforms. Thus, to take the finite difference of a data sequence, one could multiply its z transform by $1 - z$, and the resulting polynomial would be the z transform of the desired answer; in a numerical environment one would simply convolve the data with $\{1\ -1\}$. In control theory and filter design, the complex plane of z is valued as a tool for thinking about the topology of the poles and zeroes of transfer functions.

Convolution

Sequences to be convolved may be handled directly with available subprograms for convolution and inverse convolution that operate by complex multiplication in the Fourier transform domain. When two real sequences are to be convolved, one can do it conveniently by calling the two Hartley transforms, multiplying term by term, and calling the same Hartley transform again to get the answer. Some subtleties are involved when the sequences are of unequal length (*16*). But if one of the sequences is short, having less than about 32 elements depending on the machine, then slow convolution by direct evaluation of the convolution sum may be faster, and a shorter program will suffice. When the Fourier transform is used, the multiplications are complex but half of them may be avoided because of Hermitian symmetry. Software packages such as CNVLV (*12*) are available that handle these technicalities by calling two unilateral transforms or two equivalent subprograms. Fast convolution with prime factor algorithms is also available if general purpose use is not a requisite.

REFERENCES AND NOTES

1. M. A. B. Deakin, *Math. Educ.* **1**, 24 (1985).
2. A. Erdélyi, *Tables of Integral Transforms* (McGraw-Hill, New York, 1954), vol. 1.
3. G. A. Campbell and R. M. Foster, *Fourier Integrals for Practical Applications* (Van Nostrand, New York, 1948).
4. P. A. McCollum and B. F. Brown, *Laplace Transforms, Tables and Theorems* (Holt, Rinehart and Winston, New York, 1965).
5. T. W. Körner, *Fourier Analysis* (Cambridge Univ. Press, Cambridge, 1988).
6. R. V. L. Hartley, *Proc. IRE* **30**, 144 (1942).
7. C. F. Gauss, *Werke* (Royal Society of Sciences, Göttingen, 1876), vol. 3.
8. R. N. Bracewell, *The Fourier Transform and Its Applications* (McGraw-Hill, New York, ed. 2, 1986).
9. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing* (Prentice Hall, Englewood Cliffs, NJ, 1975).
10. D. F. Elliott and K. R. Rao, *Fast Transforms: Algorithms, Analyses, Applications* (Academic Press, New York, 1982).
11. H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms* (Springer, New York, 1982).
12. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes* (Cambridge Univ. Press, Cambridge, 1986).
13. J. W. Cooley and J. W. Tukey, *Math. Comput.* **19**, 297 (1965).
14. Special issue on fast Fourier transform, *IEEE Trans. Audio Electroacoust.* **AU-2**, 43–98 (1967).
15. M. T. Heideman, D. H. Johnson, C. S. Burrus, *Arch. Hist. Exact Sci.* **34**, 265 (1985).
16. R. N. Bracewell, *The Hartley Transform* (Oxford Univ. Press, New York, 1986).
17. ———, *Electron. Lett.* **23**, 1148 (1987).
18. J. D. Villasenor and R. N. Bracewell, *ibid.* **25**, 1110 (1989).
19. R. N. Bracewell, O. Buneman, H. Hao, J. Villasenor, *Proc. IEEE* **74**, 1283 (1986).
20. Digital Signal Processing Committee, IEEE Acoustics, Speech, and Signal Processing Society, Eds., *Programs for Digital Signal Processing* (IEEE Press, New York, 1979).
21. R. N. Bracewell, *Proc. IEEE* **72**, 1010 (1984).
22. M. A. O'Neill, *Byte* **13**, 293 (1988).
23. R. Ullmann, *Dr. Dobb's J.* **9**, 8 (1984).
24. O. Buneman, *IEEE Trans. Acoust. Speech Signal Process.* **37**, 577 (1989). The copyright is held by the Board of Trustees of Leland Stanford Junior University.
25. J. D. Villasenor and R. N. Bracewell, *Nature* **330**, 735 (1987).
26. ———, *ibid.* **335**, 617 (1988).
27. R. N. Bracewell and J. D. Villasenor, *J. Opt. Soc. Am.* **7**, 21 (1990).
28. R. N. Bracewell, *J. Atmos. Terr. Phys.* **51**, 791 (1989).
29. H. Hao and R. N. Bracewell, *Proc. IEEE* **75**, 264 (1987).
30. O. Buneman, *ibid.*, p. 267.
31. R. N. Bracewell, *Aust. J. Phys.* **9**, 198 (1956).
32. S. R. Deans, *The Radon Transform and Some of Its Applications* (Wiley, New York, 1983).
33. J. Radon, *Ber. Sächsische Akad. Wiss. Leipzig. Math. Nat. Kl.* **69**, 262 (1917).
34. A. M. Cormack, *Med. Phys.* **7**, 277 (1980).
35. ———, *J. Appl. Phys.* **34**, 2722 (1963); *ibid.* **35**, 2908 (1964).
36. R. N. Bracewell and A. C. Riddle, *Astrophys. J.* **150**, 427 (1961).
37. R. A. Brooks and G. Di Chiro, *Phys. Med. Biol.* **21**, 689 (1976).
38. J. L. Walsh, *Am. J. Math.* **55**, 5 (1923).
39. C.-Y. Hsu and J.-L. Wu, *Electron. Lett.* **23**, 466 (1987).

40. This work was supported by Office of Naval Research contract N00014-85-K-0544.