Digital Fourier Transforms: a revision note

John Coffey, Cheshire, UK.

2020

Fourier series, Fourier transform, discrete transform, FFT, convolution, deconvolution

1 Fourier series of continuous functions

There is nothing original in this article: I merely wished to refresh my understanding of the continuous and discrete versions of the Fourier transform and its calculation with the Fast Fourier Transform (FFT) algorithm. We start with real continuous functions.

It is well established that the sine and cosine functions are orthogonal, meaning that

$$\int_{\theta_0}^{\theta_0+2\pi} \cos(j\theta)\cos(k\theta)\,d\theta \;=\; \begin{cases}\pi, & \text{if } j = k\\ 0 & \text{if } j \neq k\end{cases} \tag{1}$$

$$\int_{\theta_0}^{\theta_0+2\pi} \sin(j\theta)\sin(k\theta)\,d\theta \;=\; \begin{cases}\pi & \text{if } j = k\\ 0 & \text{if } j \neq k\end{cases}$$

$$\int_{\theta_0}^{\theta_0+2\pi} \cos(j\theta)\sin(k\theta)\,d\theta \;=\; 0 \text{ for all } j,k.$$

for any value of θ_0, where j, k are integers. They therefore form a basis for expanding any given 'well behaved' periodic function $F(\theta)$ with period 2π as an infinite series:

$$F(\theta) \;=\; \sum_{k=0}^{\infty} \left(a_k \cos k\theta + b_k \sin k\theta\right).$$

Since b_0 could be arbitrary, and because a factor 2 appears with a_0 most texts write this as

$$F(\theta) \;=\; \frac{a_0}{2} \;+\; \sum_{k=1}^{\infty} \left(a_k \cos k\theta + b_k \sin k\theta\right). \tag{2}$$

Figure 1 gives as example. The coefficients here are $a_0 = 2$ (the constant term), $a_1 = 1, a_2 = 2$, $b_1 = 0{\cdot}5$, $b_2 = -0{\cdot}7, b_5 = 0{\cdot}8$ and all others are zero. Since the cosine function is even, a sum of cosines alone can represent any even function, and similarly a sum of sines can represent any odd function. However, for a general function with no symmetry, two independent sets of coefficients are needed.

The coefficients are found using the orthogonality relations. Multiplying by $\cos jx$ and integrating over 2π will 'knock out' all terms except that in j and so isolate the coefficient a_j.

$$\int_0^{2\pi} F(\theta)\cos j\theta\,d\theta \;=\; \sum_{k=0}^{\infty}\left(a_k \int_0^{2\pi} \cos k\theta.\cos j\theta\,d\theta + b_k \int_0^{2\pi} \sin k\theta.\cos j\theta\,d\theta\right) \;=\; \pi a_j \tag{3}$$

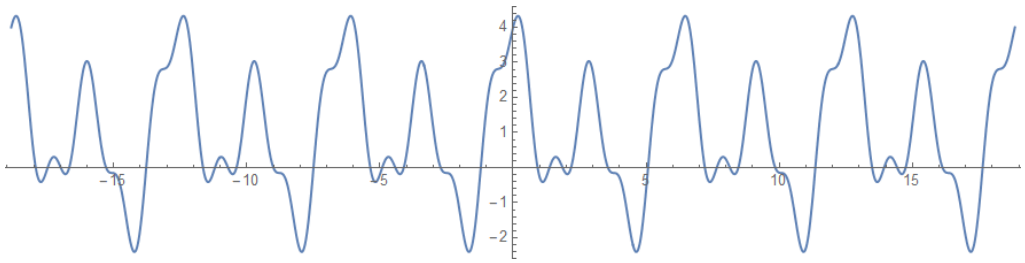

Figure 1: A periodic function, period 2π built by summing a few cosine and sine functions.

and similarly for the b_j. Note that this applies also when $j = 0$ on account of the factor $\frac{1}{2}$ defining the constant term a_0. For a general function, especially one with corners in its graph, or one which is only piecewise continuous, the coefficients a_j, b_j could extend indefinitely, $j \to \infty$. A graph of a_j against j, and another of b_j against j, would be a sequence of discrete points, one at each value of j, under an overall envelope. For instance, Figure 2 shows an approximation to the square-wave function $H(x) = +\pi/4,\ 0 < x < \pi,\ H(x) = -\pi/4,\ \pi < x < 2\pi$ by the sine series

$$F(x) \ = \ \sum_0^\infty \frac{1}{2j+1} \sin(2j+1)x.$$

The figure marks the left and right panels as graphs in two spaces – an object space plotting the given function, and an image space plotting the coefficients of its series representation. All the even coefficients are zero. 15 non-zero terms have been added to create the graph in the object space. The spiky rapid oscillations there are referred to as Gibb's phenomenon and are an unavoidable consequence of trying to fit continuous functions to a discontinuity, either in the function or its gradient.

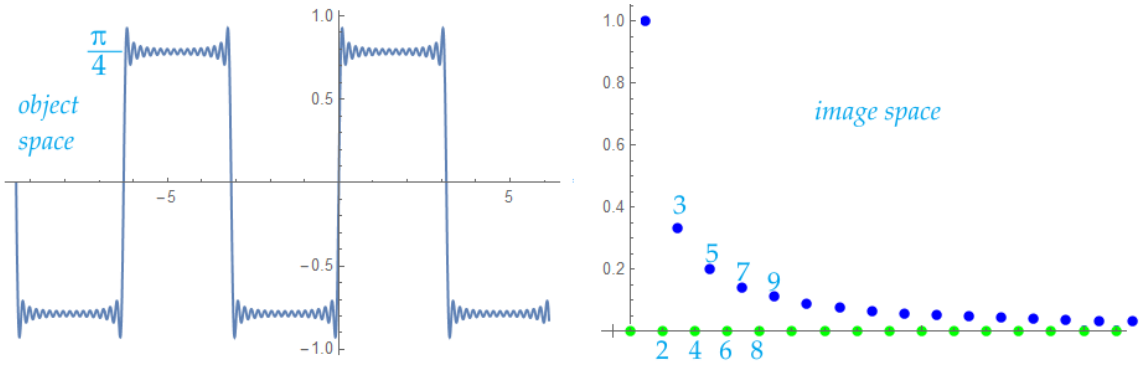

Figure 2: A square wave (left) and the coefficients b_k of its Fourier series.

It is commonplace in mathematical physics to represent periodic behaviour with the complex exponential function rather than cosines and sines. The instantaneous amplitude and phase of a real physical quantity can then be represented in one complex number. The complex equivalent of Eq 2 is

$$F(\xi) \ = \ \frac{C_0}{2} \ + \ \sum_{k=-\infty, k\neq 0}^{\infty} C_k \exp ik\xi, \qquad \xi = \theta + i\eta,$$

$$e^{ik\xi} = \cos k\xi + i \sin k\xi \ = \ e^{-\eta}(\cos\theta + i \sin\theta).$$

Note that for a general complex function of the complex variable ξ four sets of independent coefficients are required, two for the real part and two for the imaginary. This is attained here by adding terms

of the form $a_{-k}\exp(-ik\xi)$, which is equivalent to extending the lower limit of summation to $-\infty$. If the $e^{-\eta}$ factor is incorporated into c_k so that $C_k e^{-i\eta} = c_k$, the expression simplifies to

$$F(\xi) \;=\; \sum_{k=-\infty}^{\infty} c_k \exp ik\theta. \tag{4}$$

C_0 has been absorbed into the doubly infinite summation, giving an attractively compact expression. The complex coefficients are found using the orthogonality relation

$$\int_0^{2\pi} \exp(k-j)\theta \, d\theta \;=\; \begin{cases} 2\pi & \text{if } j = k \\ 0 & \text{if } j \neq k. \end{cases} \tag{5}$$

For a real function Eq 2 can be written in terms of complex exponentials using the relations

$$\cos\theta = \frac{1}{2}(e^{i\theta} + e^{i-\theta}), \qquad \sin\theta = \frac{-i}{2}(e^{i\theta} - e^{-i\theta}).$$

$$\text{Then} \qquad F(\theta) \;=\; \frac{a_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left(a_k(e^{ik\theta} + e^{-ik\theta}) - ib_k(e^{ik\theta} - e^{-ik\theta})\right).$$

$$= \frac{a_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left((a_k - ib_k)e^{ik\theta} + (a_k + ib_k)e^{-ik\theta}\right) \tag{6a}$$

$$= \sum_{k=-\infty}^{\infty} c_k e^{ik\theta}. \tag{6b}$$

The coefficients $c_{\pm k}$ for $k < 0$, $k > 0$, therefore, are paired as complex conjugates. This reduces the number of independent sets of coefficients to two – the real and imaginary parts of the c_k – as appropriate to a real function.

So far the variable has been the angular measure θ with period 2π. The physical quantity being represented might be an acoustic waveform or other time series with period T, or a function of distance x, so we make the change of variable

$$\theta \to \frac{2\pi t}{T} \quad \text{or} \quad \theta \to \frac{2\pi x}{L}.$$

Eq. 3 and 6b then become

$$F(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(\frac{2\pi ikt}{T}\right), \qquad c_k = \frac{1}{T}\int_{T_0}^{T+T_0} F(t)\exp\left(\frac{-2\pi ikt}{T}\right) dt. \tag{7}$$

There are some annoying factors of 2 which need to be taken care of. As well as those in Eq 6a, Eq 2 with the above change of variable is

$$F(t) \;=\; \frac{a_0}{2} \;+\; \sum_{k=1}^{\infty}\left\{a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right)\right\},$$

$$a_j \;=\; \frac{2}{T}\int_{T_0}^{T_0+T} F(t)\cos\left(\frac{2\pi jt}{T}\right) dt, \qquad b_j \;=\; \frac{2}{T}\int_{T_0}^{T_0+T} F(t)\sin\left(\frac{2\pi jt}{T}\right) dt. \tag{8}$$

T_0 is often taken to be 0 or $-T/2$. To add to the confusion some authors take the wavelength or period of the given function to be $2T$ instead of just T.

2 Fourier transforms

Fourier series represent strictly periodic functions. Suppose, however, that we are interested in a real function $F(t)$ which is not periodic. Can this also be represented as a trigonometric or complex exponential series? The answer is a qualified Yes, but only in the limit of the period tending to infinity. We take the given non-periodic function and isolate it on the real number line by placing large spaces either side at which the value is zero, then calculate the Fourier series. The series will model an infinite number of copies $F(t)$ separated by the blank spaces. In the limit of the blank spaces extending either side of the given function to negative infinity and positive infinity, the trigonometric or exponential series will model $F(t)$ exactly. The coefficients become increasingly close together as the amount of blank space is increased, and in the limit merge into a continuous curve. This is the Fourier transform of the given non-periodic function.

I will illustrate the approach to this limit with the function shown in Figure 3. This has been chosen because it is continuous and compact, with no infinite tails, and is described by the simple equations

$$F(t) \;=\; \begin{cases} 1+t & \text{if } -1 \le t \le 0 \\ 1-t^2 & \text{if } 0 \le t \le 1 \\ 0 & \text{otherwise.} \end{cases}$$

The integrals in Eq 8 can be obtained in closed form as

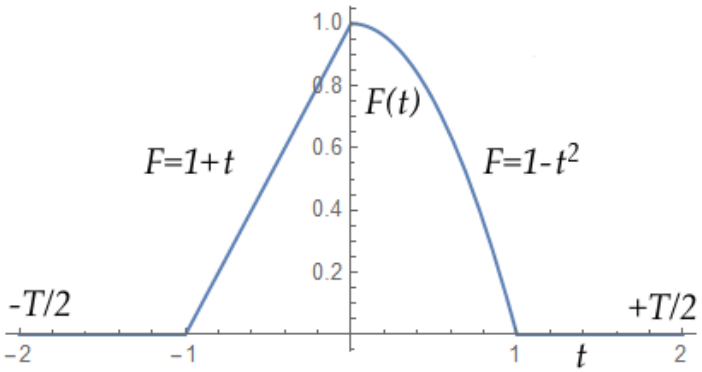

Figure 3: Function to demonstrate the limit from Fourier series to transform for the case $T = 4$.

$$\int_{-T/2}^{T/2} F(t)\cos\left(\frac{2\pi kt}{T}\right)\,dt \;=\; \frac{T^3}{4\pi^3k^3}\left\{\frac{\pi k}{T}\left(1-3\cos\left(\frac{2\pi k}{T}\right)\right)+\sin\left(\frac{2\pi k}{T}\right)\right\},$$

$$\int_{-T/2}^{T/2} F(t)\sin\left(\frac{2\pi kt}{T}\right)\,dt \;=\; \frac{T^3}{4\pi^3k^3}\left\{1\;-\;\cos\left(\frac{2\pi k}{T}\right)+\frac{\pi k}{T}\sin\left(\frac{2\pi k}{T}\right)\right\}. \qquad (9)$$

From these, taking $T = 8$, $a_1 = 0{\cdot}2753$, $b_1 = 0{\cdot}0157$, $a_2 = 0{\cdot}2303$, $b_2 = 0{\cdot}0277$, etc. By Eq 6a the complex equivalent is $c_1 = 0{\cdot}1377 - i0{\cdot}0079$, $c_2 = 0{\cdot}1152 - i0{\cdot}01384$. The behaviour is illustrated in Figure 4 where the complex Fourier coefficients c_k in the series for our test function are plotted against index k for $T = 8$, 16 and 64 using Eq 7. Since $F(t)$ is real, $G(q)$ for $q > 0$ is the complex conjugate of $G(q)$ for $q < 0$, so $\Re G(q)$ is symmetrical and $\Im G(q)$ is antisymmetric. If T is doubled, cosine and sine terms with twice the previous wavelength are needed to described it; this means the introduction of a frequency at $k/2$ for every previous k. We can see in Figure 4, therefore, that if the ratio k/T remains the same, a higher coefficient for large T will be equal to a one of lower index for smaller T. For example, with T doubled to 16, $\Re c_2 = 0{\cdot}0688$ and $16/8 \times 0{\cdot}0688 = 0{\cdot}1377$ which is $\Re c_1$ when

$T = 8$. The effect of doubling T is to move the position of coefficient a_{2k} to that previously taken by a_k. In particular note that in all cases the first zero in the imaginary part is at $k = T$. Moreover the amplitudes shrink as $1/T$, and so the whole graph of c_k against k shrinks by a factor of 2 along both axes. The intermediate coefficients have values which fit between the previously plotted points, so making the envelope of the graph more clearly defined. The envelope has the shape of the Fourier transform, described below.

Figure 5 shows the result of summing terms up to $k = 16$ for $T = 8$. Repetition of the elementary shape continues indefinitely every 8 units of t to left and to right. Reproduction of the shape of the original function is fair, though the series struggles to cope with the corners at $t = -1$, 0 and 1.

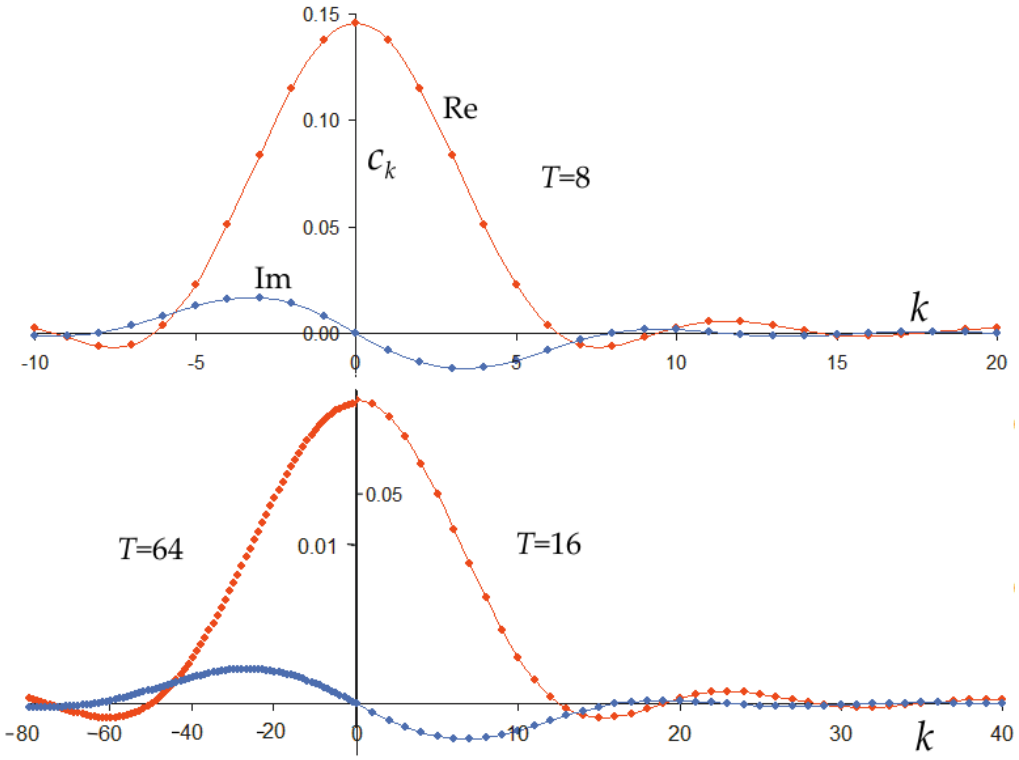

Figure 4: Real and imaginary parts of the complex Fourier coefficients for three values of period T. $Re(c_k) = a_k/2$, $Im(c_k) = -b_k/2$.

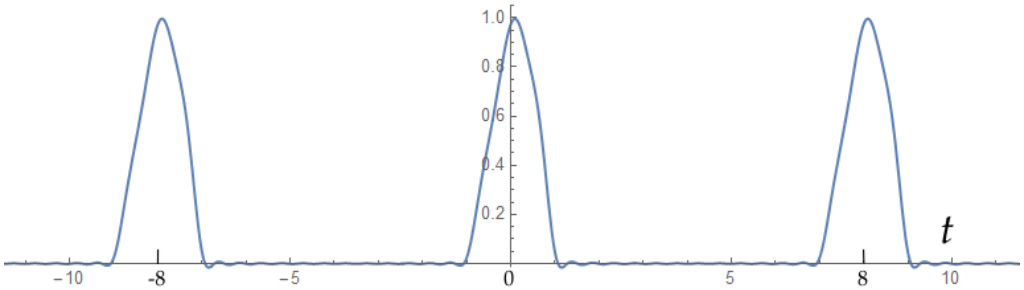

Figure 5: The given function reconstructed from the first 17 terms of its Fourier series for $T = 8$.

To determine the envelope curve, which is the Fourier Transform, we take the limit $T \to \infty$, but change the image-space variable from k to k/T since Figure 4 shows that all the discrete points

will lie on the same curve if plotted against k/T. Following Mathematica I use the image-space variable $q = 2\pi k/T$. The sum over k has been replaced by an integral over q, and the coefficients c_k by the continuous function $G(q)$. Then

$$F(t) \;=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} G(q) \exp(+iqt)\, dq, \tag{10a}$$

$$G(q) \;=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(t) \exp(-iqt)\, dt. \tag{10b}.$$

Fourier transforms come in pairs: forwards $F(t) \to G(q)$, and inverse or reverse, $G(q) \to F(t)$. The normalisation factor of $1/\sqrt{2\pi}$ is introduced to give symmetry to forwards and reverse transforms. Eqs 7, 8, 9 together give the Fourier transform of our function in Figure 3 to be[1]

$$\text{Real part:} \quad \frac{1}{q^3}\left[q - 3q\cos q + 2\sin q\right], \qquad \text{Imag part :} \quad \frac{1}{q^3}\left[-2 + 2\cos q + q\sin q\right]. \tag{11}$$

Its graph is in Figure 6. Note that the full width at half height of $F(t)$ in Figure 3 is about 1 unit in t and the corresponding width of the transform is about $2\pi \approx 6$. Reciprocal widths are a distinctive features of transform pairs. Looking back at the Fourier coefficients in Figure 4, they

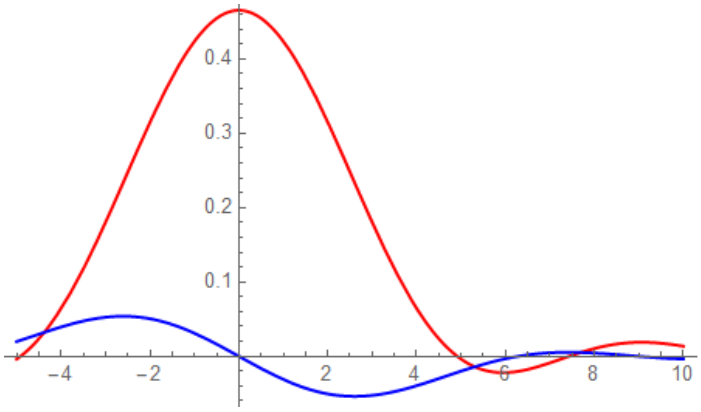

Figure 6: The Fourier transform of the function in Figure 3.

sample the Fourier transform at equal intervals, and differ only from the transform in Figure 5 by scaling factors in the two axes. Note also that the first zero in the imaginary part is at 2π in the transform, consistent with it being at $k = T$ in the Fourier series. The peak value of $\Re G(q)$ at $q = 0$ is $7/6 \times 1/\sqrt{2\pi}$, and in each panel of Figure 4 $\Re c_0$ is at $7/6 \times 1/T$.

The Fourier transform as defined in Eq 10 can be evaluated for any well behaved bounded function, not just ones like that in Figure 3 which is zero outside a finite interval. Thus the Gaussian curve $\exp(-\alpha x^2)$ has a Fourier transform of the form $\exp(-q^2/\alpha)$. In general the broader the graph of the function in object space, the narrower its transform in image space. An important function is the Dirac delta function $\delta(t - t_0)$ which is the limiting form of almost any 'spike' function which has unit area under the curve centred very narrowly around $t = t_0$, and is sensibly zero elsewhere. The absolute value of its Fourier transform will be unity for all values of the image-space variable q. Only the phase difference between real and imaginary parts will indicate the position of t_0.

[1] I find that the Mathematica `FourierTransform` operation differs from this in that the sign of the imaginary part is reversed. Their imaginary part corresponds to $b_k/2$ instead of $-b_k/2$.

3 Sampled data and convolution

We have seen that a non-periodic continuous function $F(t)$ has a non-periodic continuous transform $G(q)$. A periodic continuous function has a transform, Eq 7, which is a list of discrete coefficients $c_k,\ -\infty < k < \infty$ of the constituent complex exponentials indexed by k. As Figure 4 illustrates, these lie on the Fourier transform of the limiting non-periodic version of the given function, and in this sense are a discrete sampling of the Fourier transform. This might prompt us to ask 'what would be the transform of a sampled version of the given continuous function $F(t)$?'. Figure 7 illustrates this for the demonstration function in Figure 3. If sampling of the transform corresponds with periodicity of the function in object space, might sampling in the object space correspond with periodicity in the transform/image space?

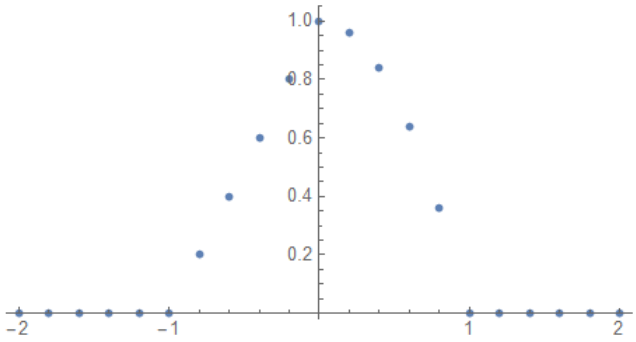

Figure 7: The curve in Figure 3 sampled at intervals of 0·2.

One approach to answering these conjectures involves two devices:

- the convolution theorem of Fourier series: that the transform of the convolution of two functions in object space corresponds with the product of their transforms in image space, and *vice versa*.
- the 'comb' function, which is a sequence of spikes at equal intervals along the axis.

Convolution of real functions $F_1(t)$, $F_2(t)$, written $F_1(t) * F_2(t)$ is defined as

$$C(t) \;=\; \int_{-\infty}^{\infty} F_1(u)F_2(t-u)\,du, \;=\; \int_{-\infty}^{\infty} F_2(u)F_1(t-u)\,du. \tag{12}$$

A picture of what this means is to imagine an experiment in which an instrument is recording the spectrum from some source. Even if the spectrum itself has very sharp lines or peaks in it, these will be smeared out in the recording if the instrument does not have an equally fine aperture. The resolving power of the instrument determines the observed line or peak width. Roughly, the recorded width is the sum of the widths of the true spectral line and the aperture width of the instrument. In two dimensions convolution occurs in the blurring of an image by an out-of-focus camera. Suppose $F_1(u)$, $F_2(u)$ are centrally positioned about the origin. Mirror F_2 in the vertical axis as $F_2(-u)$ and displace it through t to the right as $F_2(t-u)$. As t increases F_2 is dragged to the right across F_1 and the value recorded at t is the area under the product of the two functions. An exaggerated illustration is given in Figure 8. On the left at A is our demonstration function from Figure 3, and at B is a contrived 'aperture function' which has two spikes with rising edges at $t = 0$ and $2{\cdot}5$. At C is the convolution of the 'signal' A with B. Note the creation of a second peak at about $t = 2{\cdot}8$ and the slewing of the larger peak to the right, since the centroid of the larger triangle in B is near $T = 0{\cdot}6$. Is is also remarkable how smooth the convolution is compared with its parent functions, showing how readily detail is lost.

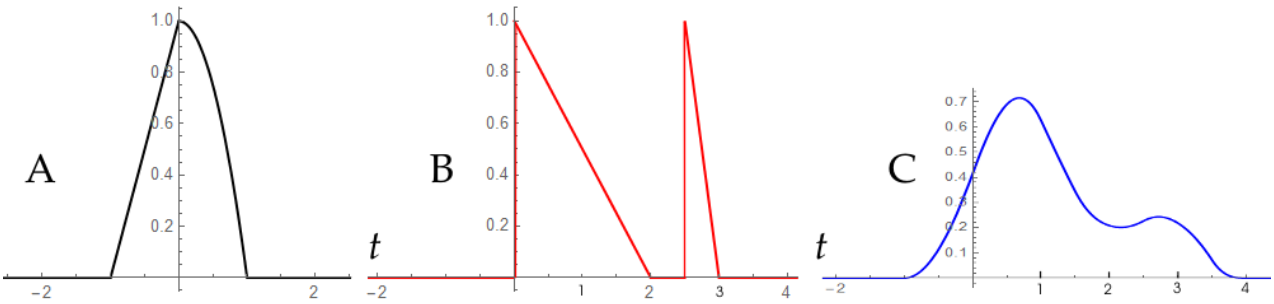

Figure 8: C is the convolution of functions A and B.

A comment is perhaps in order regarding the mirroring of one function in the convolution before it is dragged across the other. At first glance this may seem wrong, but it is correct and can be understood in this way. Regard $F_1(u)$ as a fixed function, the spectrum from some physical source, and regard $F_2(u)$ as the detector aperture which scans steadily across the spectrum, u being the 'distance', be it an actual distance, a voltage, the reading on a dial, or whatever. We expect to see the spectrum smeared out by the finite aperture, so expect to see the result in the orientation in Figure 8C, with the smaller peak on the right. Now picture the spiky aperture in Figure 8B starting well to the left of A and moving to teh right. Multiply A and B as they overlap, integrate the area and record the outcome at the instantaneous position of the moving aperture. Since the narrow spike in B would overlap A first, the small peal would be recorded first. Then as the aperture moved right, the wider triangle would overlap A and give a larger integral and hence the larger peak. Clearly the larger peak would then be on the right side of the small one, the opposite of what is required. Hence the mirroring.

If $F_1(u)$, $F_2(u)$ have Fourier transforms $G_1(q)$, $G_2(q)$, the transform of their convolution is $G_1(q).G_2(q)$. This is such am important result that I recite the proof. We suppose that the convolution $C(t)$ has a transform $D(q)$ given by the forwards Fourier formula Eq 10a:

$$D(q) \;=\; \int_{-\infty}^{\infty} C(t)e^{-iqt}\,dt \;=\; \int_t \left\{ \int_u F_1(u)F_2(t-u)du \right\} e^{-iqt}\,dt$$

where the subscripts on the integral signs indicate the variable involved. Change the variable from t to $v = t - u$ and interchange the order of integration.

$$\int_v \left\{ \int_u F_1(u)F_2(v)du \right\} e^{-iq(u+v)}\,dv \;=\; \left\{ \int_u F_1(u)e^{-iqu}\,du \right\} \left\{ \int_v F_2(v)e^{-iqv}\,dv \right\} \;=\; G_1(q)G_2(q). \quad (13a)$$

The converse also holds:

$$\int_q \left\{ \int_p G_1(p)G_2(q-p)dp \right\} e^{+iqt}\,dq \;=\; F_1(t)F_2(t). \quad (13b)$$

The second device to examine is the comb function $B(t)$, so called because its graph is like a hair comb, with many narrow prongs separated by relatively wide spaces. Mathematically it is an array of delta-functions. One representation of comb (and there are many) is simply a sum of cosines $\sum \cos kt$ for $k \to \infty$. Figure 9 shows the addition to $k = 50$. Its period is 2π. Clearly by definition, its Fourier series is simply $\Re c_k = 1$, $\Im c_k = 0$ for all k, and its transform is just a series of spikes at unit intervals in q; that is, the transform of $B(t)$ is $B(q)$ with reciprocal spacing of the spikes. Comb transform to comb.

The reasons for introducing comb is that to multiply any continuous function $F(t)$ by $B(t)$ is to sample $F(t)$ at discrete points. The transform of this sampled product will be the convolution of the transform of $F(t)$ with the transform of comb: $\int G(p)B(q-p)\,dp$. This will be multiple copies of $G(q)$ separated along the q axis at the prong spacing of $B(q)$ and added where they overlap each other. This confirms the conjecture above that sampling the object-space function induces periodicity into its transform in the image space.

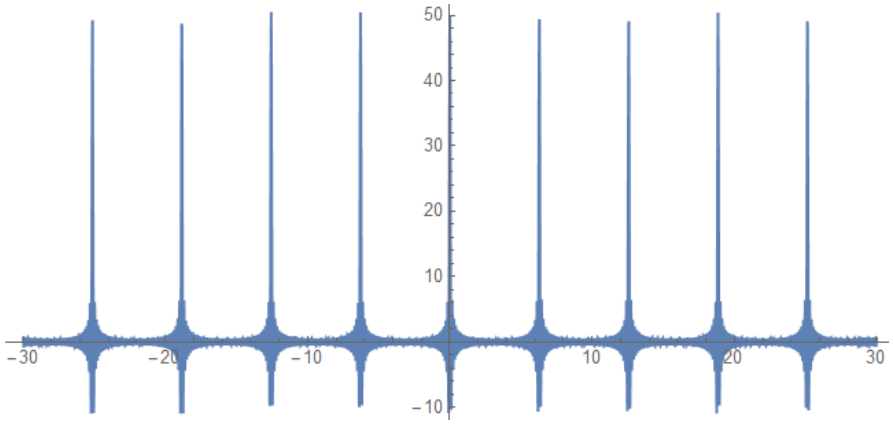

Figure 9: An approximation to the comb function $B(t)$ by $\sum_0^{50} \cos kt$.

Taking this a stage further, if the convolution in the image space is itself sampled by being multiplied by comb, its reverse transform in object space will also have periodicity induced. So sampling and periodicity are two halves of the same coin. If the sampling in either space is coarse, one unit cell of the periodic pattern is likely to overlaps its neighbours to left and right. There the functions add and so fail to represent faithfully the intended function, and this in turn brings distortion into the transform. This overlap is sometime called 'folding' and the distorted, false representation is called 'aliasing'.

The development of a sampled, periodic function and its discrete periodic transform from a non-periodic continuous function $F(t)$ is illustrated in the panels of Figures 10 to 13. For this illustration I use a simple representation of comb, as a sum of triangular spikes of half-width a and unit height spaced d apart, as in Figure 10a for the case $a = 0{\cdot}02$, $d = 0{\cdot}2$ spanning $[-1, 1]$. Since the period T here is d, the spikes in the transform, Figure 10b, are $2\pi/d = 10\pi \approx 31{\cdot}4$ apart. The triangular envelope of the transform is the transform of the narrow unit triangle and has half width $2\pi/a = 100\pi$. Clearly if the spikes in the image space are made narrower, this triangular envelope will widen reciprocally and approach the flat envelope of Figure 9.

Our reference function $F(t)$ of Figure 3 is now multiplied by the comb of Figure 10a, sampling it every $d = 0{\cdot}2$ in t. The Fourier transform of this product function is shown in Figure 11. The right panel shows that it is the convolution of the transforms of its two constituent factors. Each peak in Figure 11b looks like the central unit shown in the left panel. Note that this is very close to the transform $G(q)$ of the non-periodic $F(t)$ in Figure 6. The difference due to folding between the transform of the sampled $F(t)$ and the non-sampled is clear in Figure 12b. This corresponds to loss of high frequency detail and introduction of spurious structure.

The final stage of this demonstration is to sample in image space, perform the reverse transform back to object space, and see how the resulting version of $F(t)$ compares with the original

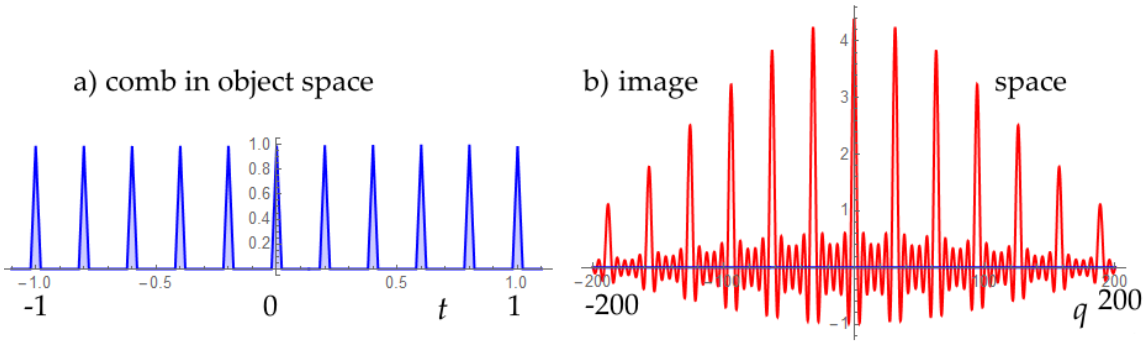

Figure 10: A comb function made of triangles (a) and its Fourier transform. (b)

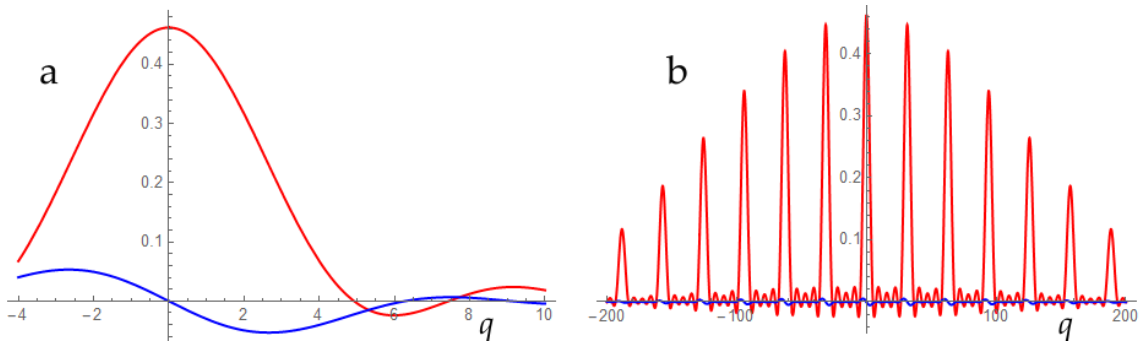

Figure 11: Transform $D(q)$ of the product of $F(t)$ with comb. a): detail of the central unit, b) wide view showing periodic repetition of the central unit.

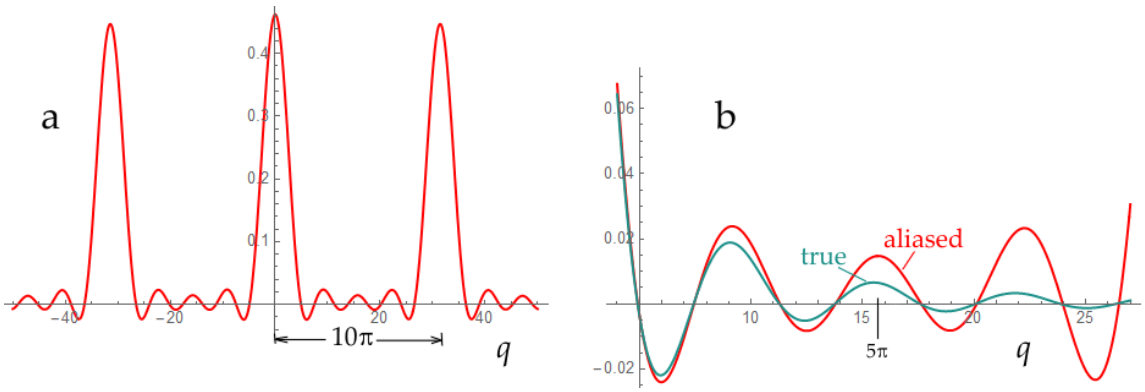

Figure 12: The transform of Figure 11 in close-up. a) the central 3 units, b) comparison of the aliased transform with the true one.

sampled one of Figure 7. How should this sampling be done efficiently? The sampling in object space has used only 11 points. We cannot expect to get back more information than we put in, so there seems no point in doing a close sampling in image space. The highest frequency which can be represented by two sample points d apart is the one which has these points corresponding to the maximum and minimum of a cosine; that is a wavelength of $2d$, frequency π/d, which is 5π in our example. $q = 5\pi$ is the position midway between adjacent peaks in the transform. Moreover, we know that the function $D(q)$ in Figures 11 and 12 is periodic, so can be represented by one period of its cyclic structure, and its transform will be the discrete coefficients of a Fourier series. Also each unit cell of the complex exponential transform has a symmetric real part and antisymmetric

imaginary part. This all means that if a real function $F(t)$ is sampled at intervals d apart, all valid information is projected into the transform between $q = 0$ and $q = \pi/d$.

If we limit ourselves to 11 sample points in image space, 6 could be used for the real part up to the mid point between adjacent cells (5π in this case), and 5 for the imaginary part, $\Im D(0)$ being zero. Regrettably I have bean unable to coax Mathematica to carry out the reverse integration of this sampled convolution in image space. To see what it must look like I must appeal to the convolution theorem: it must be the periodic version of the sampled $F(t)$ in Figure 7 repeated at intervals reciprocally related to the spacing of the sample spacing in image space (π), meaning they are $2\pi/\pi = 2$ units in t apart. In other words, they are touching. Figure 11 shows the expected form as the width of the sampling triangular comb spikes tends to zero.. Figures 11 and 13 make a discrete Fourier transform pair. If the sampling interval in image space were made smaller, the unit cells in Figure 13 would move apart to show that the function is zero outside $[-1, 1]$. If $F(t)$ were not a truncated function but instead had tails to either side, these would be folded in the reverse sampled transform and errors arise through aliasing.

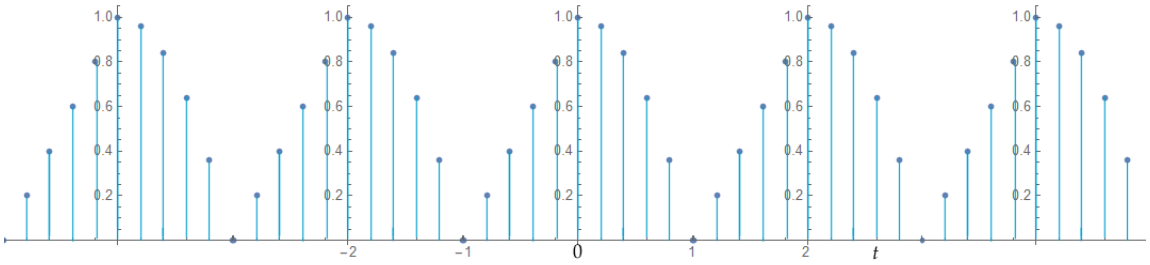

Figure 13: Expected limiting form of the reverse transform of the sampled image-space convolution in Figures 11 and 12.

4 Discrete Fourier series

As we now live in a digital age, continuous functions have been replaced by sequences of point samplings, and the Fourier series and Fourier transform need a digital formulation. It will be clear from the previous section that sampled functions will have periodic transforms, and sampled periodic transforms will have periodic patterns of points in object space representing multiple adjacent copies of the unit object space function. Also we expect loss of quality through aliasing unless the sampling rate is sufficiently high. Naturally there will be a trade-off between quality of data and computational time and effort. In this section we look at the Discrete Fourier Transform (DFT) and its implementation by the remarkable Fast Fourier Transform (FFT) algorithm developed by Cooley and Tukey in the mid 1960s for efficient calculation of the DFT.

Suppose that in object space $F(t)$ with period T is sampled N times per cycle at spacing d, so $Nd = T$. In the image space let $G(q)$ have period U and be sampled M times per cycle at spacing h, so $Mh = U$. These quantities are related:

$$U = \frac{2\pi}{d} \text{ and } h = \frac{2\pi}{T} \quad \text{so} \quad h = \frac{2\pi}{Nd} = \frac{U}{N} = \frac{U}{M} \quad \text{making} \quad M = N.$$

For $F(t)$ complex, the N samples are complex numbers requiring $2N$ independent values in each space. As we have seen, for a real function $\Im F(t) = 0$, and $\Re G(q)$ is symmetric, $\Im G(q)$ antisymmetric, so the N points sampled on $F(t)$ can be split in the image space into $N/2$ on $\Re G(q)$, $0 \le q \le \pi/d$, and $N/2$ on $\Im G(q)$ over the same interval. No information captured in the sampling is then lost.

In making the transition to the Discrete Fourier Transform[2] $F(t) \to F_t$ where t is now an integer and $T \to N$. Then

$$F_t \;=\; A \sum_{k=0}^{N-1} G_k \, e^{2\pi i k t/N}, \qquad G_k \;=\; B \sum_{t=0}^{N-1} F_t \, e^{-2\pi i k t/N} \tag{14}$$

where A, B are normalising constants whose product is $1/N$. The sum can in fact be taken over any successive N points, for instance from $-N/2$ to $N/2-1$, but this is merely equivalent to redefining the unit cell of the repeated pattern.

The Appendix gives an account of the Fast Fourier Transform algorithm. I will not say anything about it here, expect to note that it is implemented as a standard libary function in many software packages, both the forwards transform and the reverse. Here we compare how various computer packages calculate the DFT of 16 equally spaced sample points of the function in Figure 3, and how the values they return are related to each other and to the Fourier series in Eq 9 and the Fourier transform in Eq 11. The values are at $0 \cdot 2$ intervals over $[-1.4, 1.6]$ as listed below. The bottom line is the index, the middle the t value and the top line F_t. All the maths software

0	0	0	0·2	0·4	0·6	0·8	1	0·96	0·84	0·64	0·36	0	0	0	0
−1·4	−1·2	−1	−0·8	−0·6	−0·4	−0·2	0	0·2	0·4	0·6	0·8	1	1·2	1·4	1·6
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 1: 16 sampled points on the function $F(t)$ in Figure 3.

packages correctly recovered the given input values when the transform was reverse transformed, so in this sense they all work correctly. However the amplitudes differ and so do some scale factors. All transform values G_k have symmetry about index $N/2 = 8$; the real part is symmetric and the imaginary part antisymmetric. Therefore all information is contained in transform values with indices 0 to 8.

The software examined is

1. direct solution of the 16-point equivalent of Eq 15a by matrix inversion (using Reduce),
2. Wolfram Mathematica Alpha,
3. Maxima
4. Octave (similar to MatLab)
5. code provided at Rosetta Code in the C language.

The sum of the given F_t is $5 \cdot 8$, and G_0 is this value, perhaps with a scale factor. Octave and direct calculation give identical values. $G_0 = 0 \cdot 3625 = 5 \cdot 8/16$. Maxima gives the same numerical values, but the imaginary parts all have the opposite sign. Mathematica has the same signs as Maxima, but the values are scaled differently; its $G_0 = 3 \cdot 6346 = 5 \cdot 8 \times \sqrt{8/\pi}$.

The most noticeable difference between the DFT values from these programs and those in Figures 4 and 6 is seen by plotting them against k, as in Figure 14. This is the transform of the list

[2] It would probably better be called the Discrete Fourier Series

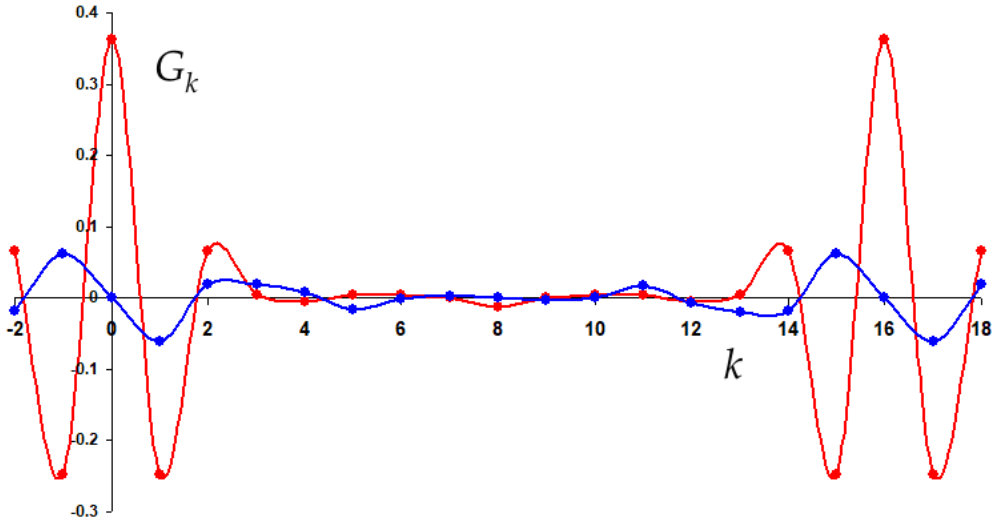

Figure 14: Discrete Fourier Transform of $F(t)$ of data in Table 1 obtained by direct matrix inversion and from Octave. Red: real part. Blue: Imaginary.

in Table 1. At first sight it looks as if it comes from a completely different function, but this is an illusion caused by phase differences between Figures 4 and 6 and the DFT values in Table 1. If the amplitudes of the G_k are calculated, they will be found to be the same as those in Figures 4 and 6. The phase angle, being $\arctan(\Im G_k/\Re G_k)$, depends on which data point is taken to be F_0. In Figure 3 the peak, 1, is at $t = 0$, but in Table 1 it is at 7. Shifting by 1 place multiplies all the G_k by $\exp(-2\pi i/N)$. If the data list is rearranged by putting the block of points 0 to 6 at the end, the phase difference is zero and the DFT graphs match Figures 4 and 6.

I finally wrote my own version of the FFT in BBC Basic, having found that the version on the Rosetta Code website only works for some input data and seems to give all zeros for more general input. To illustrate convolution of sampled data using the DFT, as implemented with the Fast FT algorithm, we can use the functions in Figure 8. Figure 15 shows the two functions, blue and green, sampled every $0 \cdot 1$ units in t (or u) over 64 points each. The values are 0 except over the narrow intervals illustrated. For this figure both A and B begin their rise from 0 at $u = 0$. The convolution, in red, then also starts to rise at $t = 0$. The repetition after 64 points is very apparent. The calculation was performed using FFT program which I wrote myself. The transforms are taken of A and B individually, multiplied point by point as complex numbers, and the inverse transform calculated. The result agrees well with Figure 8C. The result needs normalising by multiplying by h/N where h is the notional spacing of the data points in the t domain. This effectively gives the area under the sampled convolution. If some of the zeros which pad the functions out to 64 points are moved from, say, the end of the list to the beginning, the effect is to shift the convolution relative to the other two functions without changing its shape or size. In this sense the absolute position of the convolution is ambiguous.

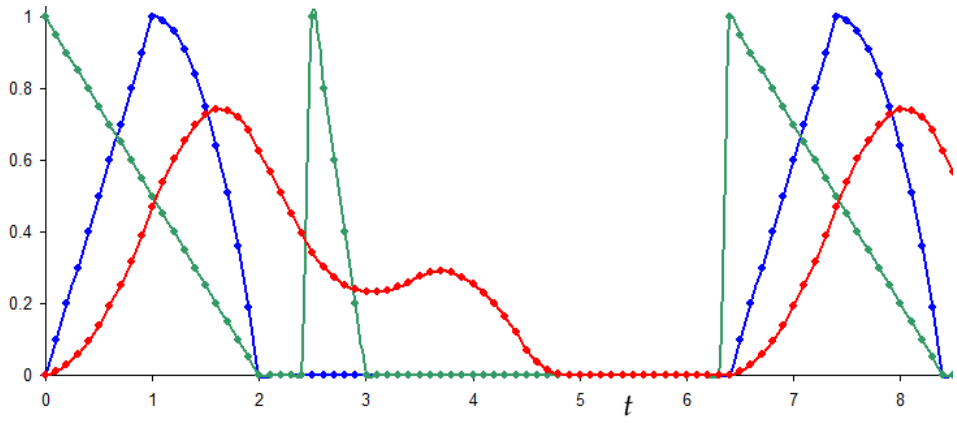

Figure 15: Two sample functions A (blue) and B (green) convolved into C (red) by the multiplication of their DFTs.

5 Noise filtering and deconvolution

One major application of the DFT and FFT algorithm is to the processing of 1-D audio and 2-D graphics images in an attempt to improve their quality. Two major defects to correct are noise and blurring.

5.1 The deconvolution dream

Physical measurement often involves scanning through a spectrum using an instrument with an aperture of finite width. This blurs the observed spectrum so the recorded signal is the convolution of the true spectrum with the aperture function. We saw in §3 that the transform of this convolution is the product of the transforms of the spectrum and instrumental aperture. We might think, therefore, that if the transform of the convolution were divided by the transform of the aperture function, we could recover the fine detail of the true spectrum using the inverse transform. This has been attempted by many scientists and engineers over the years. It does not work at all well because random noise is always present in real world data. If the transform of the aperture function has values close to zero, division by these will grotesquely magnify any noise present, and the resulting reverse transform could have structure which is totally spurious.

In view of this it would seem obvious to filter out the noise before attempting deconvolution. This is easier said than done. Removing noise from a signal is one of nature's almost irreversible processes, along with unmixing a cake and factorising a large integer. Where time-varying signals are involved, in the days of analogue electronics data would be smoothed to remove noise using some electronic circuit involving capacitors, resistors and inductors. A whole industry grew up to design and manufacture so-called passive and active filters which would remove unwanted parts of the input in selected frequency bands. Digital filtering essentially involves calculating the DFT of the input and attenuating or even setting to zero certain parts of the spectrum by multiply the transform of input signal by a filtering function in the frequency domain. This supposes that the user has adequate knowledge of the noise to design a suitably selective filter. In some applications noise can be sampled at a nominally quiet part of the signal and its characteristics determined. Much noise has a Gaussian (normal) distribution in amplitude in t, and its spectrum in k also has a Gaussian distribution of the coefficients G_k of each frequency. Typically most of the structure is contained in only the first few frequency components. Attenuating the higher frequencies k does reduce the fine scale noise in the t domain, but inevitably loses the true fine detail. I have carried out some trials of deconvolution of

the functions in Figures 8 and 15 in the presence in artificially added Gaussian noise. The results have been disappointing. Even small amounts of noise – say with only rms 1% of the peak signal – frustrates useful recovery of the true signal, and instead produces completely spurious structure made by the addition of randomly exaggerated sine and cosine curves. Deconvolution schemes are intrinsically unstable, though that has not stopped some researchers from pressing ahead, perhaps using intuition to cut off the transforms above a threshold value of k at which noise is judged to be too great.

5.2 Removal of echoes in audio

One application I have been interested in has been to see whether deconvolution could remove echoes and reverberation from an audio recording made in an echoing building. I have in mind a simplified model of a church which is 18 m wide, 36 m long and about 14 m high. The voices of clergy are amplified from radio microphones worn on their chests, and fed to two loudspeakers situated either side of the central aisle, about 5 m apart from each other and 3 m above ground. Suppose a recording microphone is placed 1 m away from one loudspeaker, pointing directly at it. A rough calculation of the impulse signal (that heard at the microphone due to a sudden short sound such as a stick hitting a piece of wood) can be made using the velocity of sound in air of 340 m/sec. The recording microphone receives the direct sound from each speaker plus plus echoes from the four walls, roof and floor. Ignoring second reflections, the impulse response function (taking the place of the aperture function in earlier discussion) will consist of 14 closely spaced spikes at delays from 12 msec to 115 msec relative to the direct sound from the nearer speaker. The relative amplitudes are not readily calculated because they depend on the fall off with distance and the angular variation in amplitude of the speakers and sensitivity of the microphone. In a space with no reflections fall off from a small source with range R is as $1/R^2$, while reflection from a wall is as $1/R$. The second sound to arrive is the direct output from the other speaker 5 m away, and could be perhaps 20 dB to 26 dB less (1/10 to 1/20) of the primary sound. The others are much weaker. As an illustration the first five signals as listed in this table. The direct signals from the two speakers should be taken as true

delay (msec)	amplitude %	function
0	100	δ
12	5	δ
15	2	
20	1	
32	0·8	

δ-functions with only one data point non-zero. The convolution with these will therefore be the exact reproduction of the given recording. The later three echoes will be smoothed by minor local structure on the walls and furniture. In a real situation the impulse would best be recorded as some nominally instantaneous crack.

Audio processing using Fourier transforms cannot be done on the whole 30 or 60 minute long recording. It must be divided into manageably short samples by a window which moves in time steps from start the end. The window function is zero outside a certain width, typically bell shape within that width, and the given signal is multiplied by it. Sampling rates for speech could be as low as 11 kHz, one quarter of the rate used for CDs. This is 11 samples every 1 msec. The sample length must cover the duration of the impulse response, with a margin either end to limit aliasing distortion. So the window needs to be about 200 msec wide, and so would contain 2200 samples. In terms of a

Fourier transform the best length would therefore be 2^{11} = 2048 samples, a large but manageable number. Clearly if more echoes were included, the number of samples would have to be doubled.

There are many varieties of windowing function used in industry, all attempting to limit distortion of the signal by limiting aliasing. To do so their Fourier transforms should not have zeros, so a square window would be a poor choice as its transform is the oscillatory 'sinc' function $\sin(k)/k$. Names such as Hann, Hamming, Blackman-Harris and Kaiser have been given to effective functions proposed by these scientists. The window moves in steps with typically 75% overlap.

I do not know of any software, commercial or otherwise, which attempt this deconvolution to remove echoes. The problem is regarded as ill-posed and intractable. Perhaps the nearest is the noise removal facility of GoldWave (*www.goldwave.com*) which uses the Fourier transform to remove noise, a sample of which is selected from a nominally quiet part of the recording.

John Coffey, July 2020

6 Appendix: The FFT Algorithm

The FFT is a computationally highly efficient way of computing the discrete Fourier transform of equally spaced sample points F_t of a possibly continuous function $F(t)$. Each of the relations in Eq 14, defining the reverse and forwards discrete Fourier transforms, is a set of linear equations, so the transform operation amounts to solving these equations. The FFT algorithm makes great use of symmetries amongst the coefficients when N is a power of 2. It is available as a library function in many software packages.

6.1 The DFT in matrix form

Each of the DFT definitions in Eq 14 can be written in matrix form and in principle be solved for F_t or G_k by matrix multiplication. Suppose the G_k are known and examine the matrix for F_t. Write $e^{2\pi i/N} = w$, an Nth root of 1, and reduce the indices mod N. For $N = 8$ the equation is

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 & w^4 & w^5 & w^6 & w^7 \\ 1 & w^2 & w^4 & w^6 & 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w & w^4 & w^7 & w^2 & w^5 \\ 1 & w^4 & 1 & w^4 & 1 & w^4 & 1 & w^4 \\ 1 & w^5 & w^2 & w^7 & w^4 & w & w^6 & w^3 \\ 1 & w^6 & w^4 & w^2 & 1 & w^6 & w^4 & w^2 \\ 1 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w \end{pmatrix} \begin{pmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \\ G_6 \\ G_7 \end{pmatrix}. \tag{15a}$$

In matrix notation

$$\mathcal{F}_8 = \mathcal{V}_8 \mathcal{G}_8$$

where the subscript now denotes the size of the matrix. The square matrix $\mathcal{V}_8$ is a Vandermonde matrix. It has a high degree of symmetry. Here are the actual values with $w = \exp(2\pi i/8) = (1+i)/\sqrt{2}$, $w^5 = -w$, $w^3 = -w^7 = -w^*$.

$$\mathcal{V}_8 = \left(\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & i & -w^* & -1 & -w & -i & w^* \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & -w^* & -i & w & -1 & w^* & i & -w \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -w & i & w^* & -1 & w & -i & -w^* \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & w^* & -i & -w & -1 & -w^* & i & w \\ \hline 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{15b}$$

The bottom line gives the sum of each column. Each column except the first is a sum of all N roots of unity for $N = 8, 4, 8, 4, 2, 8, 4, 8$ respectively, all of which are zero[3]. Using this we can demonstrate that F_t and G_k are indeed inverses.

$$B \sum_{t=0}^{N-1} F_t \, e^{-2\pi i k t/N} = B \sum_{t=0}^{N-1} \left[A \sum_{j=0}^{N-1} G_j \, e^{2\pi i j t/N} \right] e^{-2\pi i k t/N}$$

$$= AB \sum_{j=0}^{N-1} G_j \left[\sum_{t=0}^{N-1} e^{2\pi i (j-k) t/N} \right] = NAB \, G_k = G_k$$

[3] If each root is represented by a point mass placed around the unit circle, their centre of mass would be at the origin.

since $AB = 1/N$ by design.

As we would expect from the formula for the forwards DFT, the inverse of this matrix is

$$\mathcal{V}_8{}^{-1} = \frac{1}{8}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w^* & -i & -w & -1 & -w^* & i & w \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & -w & i & w^* & -1 & w & -i & -w^* \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -w^* & -i & w & -1 & w^* & i & -w \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & w & i & -w^* & -1 & -w & -i & w^* \end{pmatrix} \tag{15c}$$

which is just its complex conjugate, $\mathcal{V}_8^*$. So not only is the matrix symmetric but it is unitary – its inverse is its complex conjugate. This holds true for all values of N as can be seen by evaluating a typical element in the product $\mathcal{V}_N\mathcal{V}_N^*$. Element 33 is

$$1 + w^3 w^{3*} + w^6 w^{6*} + \ .... \ + w^{3(N-2)} w^{3(N-2)*} + w^{3(N-1)} w^{3(N-1)*}$$
$$= 1 + |w^3|^2 + |w^6|^2 + \ .... \ + |w^{3(N-1)}|^2 \ = \ N$$

since all w^k have unit modulus. In contrast the element 34 is

$$1 + w^3 w^{4*} + w^6 w^{8*} + \ .... \ + w^{3(N-2)} w^{4(N-2)*} + w^{3(N-1)} w^{4(N-1)*}$$
$$= \ 1 + w^3 w^{4*} + w^6 w^{8*} + \ .... \ + w^{-6} w^{-8*} + w^{-3} w^{-4*}$$

But $w^{k*} = w^{-k}$ so the previous line equals

$$= \ 1 + w^{-1} + w^{-2} + w^{-3} \ .... \ + w^3 + w^2 + w \ = \ 0.$$

Since $w^{-k} = w^{N-k}$, this is the sum of all the N roots of unity[4]. Hence the product $\mathcal{V}_N\mathcal{V}_N^* = \mathcal{I}_N$, the identity matrix.

Solving the N simultaneous linear equations for the G_k is in principle achievable even for several thousand rows and columns, but becomes computationally prohibitive if N is very large. A more efficient way of finding the G_k was developed by Cooley and Tukey. Perhaps they were struck by observing that if N is even and the columns are indexed from 0 to $N-1$, the even rows contain only even powers of w and the odd rows only odd powers, and the submatrix formed from just the even rows is essentially the matrix $\mathcal{V}_{N/2}$. Therefore an $N \times N$ matrix could be factored into two $N/2 \times N/2$ ones and thereby gain greatly in computing efficiency. In Eq 15a, b only the odd columns contain w; the even ones have ± 1 or $\pm i$ which are the 4th roots of 1, not the 8ths. Shuffle the columns of $\mathcal{V}_8$ to bring the 'w' columns together and note that it then has a 4-by-4 block structure. The G_t must be correspondingly reordered to preserve the linear equations. Clear notation is needed here. I propose to call the shuffled version of $\mathcal{V}_N$ '$\mathcal{W}_N$', and the shuffled version of $\mathcal{G}_N$ '$\mathcal{H}_N$'.

$$\mathcal{F}_8 = \mathcal{V}_8\mathcal{G}_8 \ = \ \mathcal{W}_8\mathcal{H}_8 \ = \ \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i & w & -w* & -w & w^* \\ 1 & -1 & 1 & -1 & i & -i & i & -i \\ 1 & -i & -1 & i & -w^* & w & w^* & -w \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & i & -1 & -i & -w & w^* & w & -w^* \\ 1 & -1 & 1 & -1 & -i & i & -i & i \\ 1 & -i & -1 & i & w^* & -w & -w^* & w \end{array}\right)\begin{pmatrix} G_0 \\ G_2 \\ G_4 \\ G_6 \\ G_1 \\ G_3 \\ G_5 \\ G_7 \end{pmatrix}$$

[4] This is strictly true only for some elements. Other elements may be formed by repeated sums such as $1 + w^2 + w^6 + w^8 + 1 + w^2 + w^6 + w^8$ which is also zero.

$$= \left(\begin{array}{c|c} \mathcal{V}_4 & \mathcal{D}_4\mathcal{V}_4 \\ \hline \mathcal{V}_4 & -\mathcal{D}_4\mathcal{V}_4 \end{array}\right)\mathcal{H}_8 = \begin{pmatrix}\mathcal{H}_{4a} \\ \mathcal{H}_{4b}\end{pmatrix}. \tag{17}$$

$\mathcal{H}_{4a}$ is the column vector $(G_0 \;\; G_2 \;\; G_4 \;\; G_6)^T$ and $\mathcal{H}_{4b}$ is similarly the bottom half of $\mathcal{H}_8$. This is remarkable. Things become even more remarkable when we note that

1. $\mathcal{V}_4$ is another unitary matrix, the Vandermonde matrix for the 4×4 Discrete Fourier Transform counterpart of Eq 15a,

2. as the notation $\mathcal{D}_4\mathcal{V}_4$ implies, this 4-by-4 block can be factorised.

$\mathcal{D}_4$ is found by right-multiplying the upper right block by $\mathcal{V}_4^*$, the inverse of $\mathcal{V}_4$. The result is

$$\mathcal{D}_4 = \begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & -w*\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & w^3\end{pmatrix}, \tag{17}$$

the diagonal matrix of $\exp(2\pi ik/8)$, $0 \le k < 4$, the 8th root of unity. Once we know that this is the case for any even N, $\mathcal{D}_{N/2}$ can be written down without further calculation. Using the elements $\mathcal{V}_4$ and $\mathcal{D}_4$, $\mathcal{W}_8$ itself can be factorised as I show in subsection §6.2 below.

$$\mathcal{W}_8 = \left(\begin{array}{c|c} \mathcal{I}_4 & \mathcal{D}_4 \\ \hline \mathcal{I}_4 & -\mathcal{D}_4 \end{array}\right)\left(\begin{array}{c|c} \mathcal{V}_4 & \\ \hline & \mathcal{V}_4 \end{array}\right) \tag{18}$$

where $\mathcal{I}_4$ is the identity matrix and the zero matrix. The partitioning of the matrix on the right means that the calculation separates into two independent 4-by-4 sets of simultaneous linear equation, one for the even-indexed sample points, the other for the odd. The even G_k elements form one building block of the solution for $\mathcal{F}_8$, and the odd elements another. In turn $\mathcal{V}_4\mathcal{H}_{4a}$, $\mathcal{V}_4\mathcal{H}_{4b}$ can each be split into two 2-by-2 matrices involving the even and odd elements of $\mathcal{H}_{4a}$ or $\mathcal{H}_{4b}$, at which stage the building blocks are all of the form $G_j \pm G_k$ as §6.3 explains. James Cooley and John Tukey saw that this could be coded recursively to fragment large matrices where $N = 2^m$ for some integer m and so vastly reduce the number of multiplications and additions needed.

6.2 Factorisation of matrix $\mathcal{W}_8$

We are looking for a factorisation of $\mathcal{W}_8$ which has as many zero elements as possible because that will simplify subsequent operations. Use the property that block matrices can be manipulated in much the same way as ordinary matrices, the blocks behaving as elements, though multiplication is not generally commutative. Suppose that the 2×2 block matrix in Eq 15 is factored into two 2×2 blocks and look to see what the elements might be.

$$\begin{pmatrix}\mathcal{V} & \mathcal{D}\mathcal{V} \\ \mathcal{V} & -\mathcal{D}\mathcal{V}\end{pmatrix} = \begin{pmatrix}a & b \\ c & d\end{pmatrix}\begin{pmatrix}p & q \\ r & s\end{pmatrix} = \begin{pmatrix}ap+br & aq+bs \\ cp+dr & cq+ds\end{pmatrix}$$

Matching elements

$$r = -\frac{p(a-c)}{b-d}.$$

If $c = a$, $r = 0$ and one zero element has been obtained. It now looks like

$$\begin{pmatrix} \mathcal{V} & \mathcal{D}\mathcal{V} \\ \mathcal{V} & -\mathcal{D}\mathcal{V} \end{pmatrix} = \begin{pmatrix} a & b \\ a & d \end{pmatrix} \begin{pmatrix} p & q \\ 0 & s \end{pmatrix} = \begin{pmatrix} ap & aq+bs \\ ap & aq+ds \end{pmatrix}$$

There are no additions in the original matrix, so it is reasonable to let $q = 0$. Also $p = \mathcal{V}/a$. a looks as if it could be chosen from a wide range of values, so simply take $a = 1$. Then

$$\begin{pmatrix} \mathcal{V} & \mathcal{D}\mathcal{V} \\ \mathcal{V} & -\mathcal{D}\mathcal{V} \end{pmatrix} = \begin{pmatrix} 1 & b \\ 1 & d \end{pmatrix} \begin{pmatrix} \mathcal{V} & 0 \\ 0 & s \end{pmatrix} = \begin{pmatrix} \mathcal{V} & bs \\ \mathcal{V} & ds \end{pmatrix}$$

It is now clear that $bs = DW$, $ds = -DW$. There are four ways this could be achieved as listed below:

$$\begin{pmatrix} \mathcal{V} & \mathcal{D}\mathcal{V} \\ \mathcal{V} & -\mathcal{D}\mathcal{V} \end{pmatrix} = \begin{pmatrix} \mathcal{I} & \mathcal{D} \\ \mathcal{I} & -\mathcal{D} \end{pmatrix} \begin{pmatrix} \mathcal{V} & 0 \\ 0 & \mathcal{V} \end{pmatrix}. \qquad (Copy\ of\ Eq\ 18)$$

and also these other choices of sign:

$$\begin{pmatrix} \mathcal{I} & -\mathcal{D} \\ \mathcal{I} & \mathcal{D} \end{pmatrix} \begin{pmatrix} \mathcal{V} & 0 \\ 0 & -\mathcal{V} \end{pmatrix}, \quad \begin{pmatrix} \mathcal{I} & \mathcal{V} \\ \mathcal{I} & -\mathcal{V} \end{pmatrix} \begin{pmatrix} \mathcal{V} & 0 \\ 0 & \mathcal{D} \end{pmatrix}, \quad \begin{pmatrix} \mathcal{I} & -\mathcal{V} \\ \mathcal{I} & \mathcal{V} \end{pmatrix} \begin{pmatrix} \mathcal{V} & 0 \\ 0 & -\mathcal{D} \end{pmatrix}.$$

I have checked that Eq 18 and the first of the three others work when converted into 4-by-4 blocks, but the other two do not, possibly because the matrix blocks do not commute.

6.3 Recursion in the FFT algorithm

The concept is that a large matrix with $N = 2^m$ can be dissected into two building blocks of half the size; these each in turn can be dissected giving four matrices each a quarter the size, and so on until there are $N/2$ pairs of the form $G_j + G_k$, $G_j - G_k$. The process is then reversed to build 4-element blocks from the two 2-element ones, 8-element blocks from 4-element ones, and so up to $\mathcal{F}_N$ from two at $N/2$. As two blocks A, B, are compounded into one twice the size, the operation is always of the form $A + w^h B$ where w^h is one of the Nth roots of unity. There are two particular questions to answer:

1. what are the specific indices j, k and values G_j, G_k which make up the pairs, 4-sets, 8-sets, etc.?
2. what are the multipliers $w^h = e^{2\pi i h/N}$ at each stage?

Both are determined by the even-odd shuffling of the matrices at the $m - 1$ stages of dissection by which an $N = 2^m$ transform is calculated.

Here are some examples which show the operation for $N = 2$, 4, 8 and 16.

N=**2**. When a large vandermonde matrix with $N = 2^m$ has been fully dissected, one reaches a set of $N/2$ 2-by-2 matrices each with the form

$$\mathcal{V}_2\mathcal{H}_2 = W_2 \begin{pmatrix} G_A \\ G_B \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} G_A \\ G_B \end{pmatrix}$$

since $\mathcal{V}_1 = \mathcal{D}_1 = 1$. Here G_A and G_B are the G_k values in the upper and lower halves of the column matrix $\mathcal{H}_2$. We will look shortly at just what these value are, but first multiply this out to see that

$$\mathcal{V}_2\mathcal{H}_2 = \begin{pmatrix} G_A + G_B \\ G_A - G_B \end{pmatrix}.$$

If in fact $N = 2$ (a very small transform!), the solution is

$$F_0 = G_0 + G_1$$
$$F_1 = G_0 - G_1.$$

No swapping is required and the multiplying factors w^h on the second element are +1 and −1 = $\exp(2\pi i/2)$. So $h = 0$ or $h = 1$.

N=**4**. The general matrix relations at the 4-element stage are

$$\mathcal{V}_4\mathcal{H}_4 \;=\; W_4\begin{pmatrix}X\\Y\end{pmatrix} \;=\; \begin{pmatrix}1&0&1&0\\0&1&0&i\\1&0&-1&0\\0&1&0&-i\end{pmatrix}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}\begin{pmatrix}X\\Y\end{pmatrix}$$

since

$$\mathcal{V}_2 \;=\; \begin{pmatrix}1&1\\1&-1\end{pmatrix} \quad\text{and}\quad \mathcal{D}_2 \;=\; \begin{pmatrix}1&0\\0&i\end{pmatrix}.$$

Here X and Y are upper and lower pairs of $G_A{+}G_B$, $G_A{-}G_B$ in the matrix $\mathcal{H}_4$. We shall identify their specific values shortly. Multiplying this out generates $(G_A{+}G_B){+}(G_C{+}G_D)$, $(G_A{-}G_B){+}i(G_C{-}G_D)$, $(G_A+G_B)-(G_C+G_D)$, $(G_A-G_B)-i(G_C-G_D)$. If $N = 4$ the transform is

$$\begin{aligned}F_0 &= (G_0+G_2)+(G_1+G_3)\\F_1 &= (G_0-G_2)+i(G_1-G_3)\\F_2 &= (G_0+G_2)-(G_1+G_3)\\F_3 &= (G_0-G_2)-i(G_1-G_3)\end{aligned}.$$

Elements G_1 and G_2 have been swapped to form the upper and lower pairs with indices {0, 2}, {1, 3}. In alternate rows the sign within each pair alternated from +1 to −1 corresponding to $N = 2$, while the multiplying factors on the second pair in each set of four are 1, i, −1, $-i$, consecutive powers of $\exp(2\pi i/4)$. So at this stage $h = 0, 1, 2, 3$. These arise directly from the matrix $\mathcal{D}_4$, Eq 17.

N=**8**. There are two stages of even-odd swapping. The order of the elements of $\mathcal{G}$ becomes in turn

At start	(0	1	2	3	4	5	6	7)	Multipliers 1, $w = e^{2\pi i/8}$, w^2, ... w^7,
In 4-sets	(0	2	4	6)	(1	3	5	7)	Multipliers 1, i, −1, $-i$
In pairs	(0	4)	(2	6)	(1	5)	(3	7)	Multiplier 1, −1

Each value of F_t will be made of pairs with indices $(0,+4)$, $(0,-4)$, $(2,+6)$, $(2,-6)$, etc. (bottom row of table). These will be compounded in 4-element blocks with form $((0,\pm4)+w^h((2,\pm6))$ with w^h coming from teh middle row. Finally the odd index 4-element blocks will multiplied by a further factor (top row) and added with the even index 4-element blocks.

Here is a prescription for calculating the F_t. Write four columns. In the first put on alternate rows $(0,+4)$, $(0,-4)$. In column 2 write $(2,+6)$, $(2,-6)$ on alternate rows, in column 3 write $(1,+5)$, $(1-5)$, and in the fourth write $(3,+7)$, $(3,-7)$. Next go down column 2 and place in front if each pair in turn the multipliers 1, $i = w^2$, $-1 = w^4$ and $-i = w^6$, and repeat this sequence on the lower set of four rows. Do the same with column 4. Now bracket the elements of columns 3 and 4 together,

and prefix each bracket with the multipliers 1, $w = e^{2\pi i/8}$, w^2, ... w^7 in turn down the column. The sum in each row is now the respective F_t. Here are the first 4 rows:

$$\begin{array}{rcllll} F_0 &=& (G_0+G_4) & +1(G_2+G_6) & +1[(G_1+G_5) & +1(G_3+G_7)] \\ F_1 &=& (G_0-G_4) & +i(G_2-G_6) & +w[(G_1-G_5) & +i(G_3-G_7)] \\ F_2 &=& (G_0+G_4) & -1(G_2+G_6) & +w^2[(G_1+G_5) & -1(G_3+G_7)] \\ F_3 &=& (G_0-G_4) & -i(G_2-G_6) & +w^3[(G_1-G_5) & -i(G_3-G_7)] \end{array}$$

Multiplying out and using $w^3.(-i) = w$

$$F_3 \;=\; G_0 - G_4 - iG_2 + iG_6 + w^3G_1 - w^3G_5 + wG_5 - wG_7\,.$$

N=**16**. Three stages of shuffling are needed to bring even-index and odd-index elements together. This means that the F_t are constructed from 8 pairs such as $G_0 \pm G_8$, $G_5 \pm G_{13}$, etc. combined into 4 sets of 4 with the second pairs multiplied by 1, i, -1 or $-i$, and then into 2 sets of 8.

At start	(0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15)	Multipliers 1, $w = e^{2\pi i/16}$, w^2, ... w^{15}
In 8-sets	(0	2	4	6	8	10	12	14)	(1	3	5	7	9	11	13	15)	Multipliers 1, w^2, w^4, ... w^{14}
In 4-sets	(0	4	8	12)	(2	6	10	14)	(1	5	9	13)	(3	7	11	15)	Multipliers 1, i, -1, $-i$
In pairs	(0	8)	(4	12)	(2	10)	(6	14)	(1	9)	(5	13)	(3	11)	(7	15)	Multiplier 1, -1.

It is possible to write down any F_t by inspection using the recipe used for $N = 8$ above. For instance, in index notation F_5 is

$$[(0,-8)+i(4,-12)] + w^{10}[(2,-10)+i(6,-14)] + w^5\{[(1,-9)+i(5,-13)] + w^{10}[(3,-11)+i(7,-15)]\}$$
$$= G_0 - G_8 + iG_4 - iG_{12} - w^2G_2 + w^2G_{10} - w^6G_6 + w^6G_{14} + w^5G_1 - w^5G_9 - wG_5 + wG_{13} + w^7G_3 - w^7G_{11} + w^3G_7 - w^3G_{15}\,.$$

The above is essentially the FFT algorithm, though some extra features are included in the published versions. One important feature is the ability to write down the order of the G_k in the bottom row, for 2-element blocks, without having to create the whole table through $m-1$ levels (not that that would be a great computational task.) Someone many years ago spotted that if a chosen position along the bottom row is written in binary, the binary number whose digits are the reverse is the index k of the G_k value at that position. Perhaps they had first noticed that some elements do not move. Certainly the ones with index 0 and $N-1$, but also $N/2+1$. These are in positions which are symmetric in binary. The table below gives all results for $N = 16$ where $0 \le k \le 15$. The position - index values are interchangeable. Check these against the bottom row for $N = 16$ above.

position	*binary*	*reverse*	*index*
0	0000	0000	0
1	0001	1000	8
2	0010	0100	4
3	0011	1100	12
4	0100	0010	2
5	0101	1010	10
6	0110	0110	6
7	0111	1110	14
9	1001	1001	9
11	1011	1101	13
15	1111	1111	15