Mathematical tools: Convolution and the Fourier Transform

This material is abstracted from a chapter in an fMRI book still being written, thus there is a repeated focus on MRI examples.

Introduction

A few mathematical methods are so commonly used in neuroimaging that it is a practical necessity for the researcher to have a strong intuitive understanding of them, as well as working knowledge of their applications. In this section, we introduce two of these: **convolution** and the **Fourier transform**. We will first discuss these from a primarily intuitive and non-mathematical perspective, and then develop some of the formal derivations. Finally, we will consider a few applications relevant to functional neuroimaging.

Linear Time Invariant Systems

A system is said to be linear if $f(Ax) = Af(x)$ when A is a constant. This is a very general idea. If, for example, x is the weight of a single penny, f(x) is the weight of a group of pennies and A is the number of stones, the intuitive understanding is that the weight scales *linearly* with the number of stones. If $g(x)$ is the weight of a box of stones, we might expect that $g(Ax) = Af(x) + b$, which is also a linear equation (b being the weight of the box).

Linear systems are extremely important in all manner of engineering, primarily because there are a huge number of theorems and results for linear systems, and because in most cases linear operations are readily invertible – if we know the output we can infer the input. An example of a nonlinear system might be the area, A, of a square in relation to the length of its sides, d, in which $A = d^2$. Equations in which variables are multiplied by one another, or by themselves in the previous example, are non-linear.

Time invariance means, essentially, that the response to an input at time τ, will be a time-shifted version of the response at $\tau+\delta$. Time invariance is an important extension of linearity. A Linear Time Invariant (LTI) system is one in which the linear result is the same, even if the data were acquired at different times. LTI systems also obey the property that: $f(A+B) = f(A) + f(B)$, which also follows from the above.

The Brain Response are Not *Linear*

There are innumerable reasons that measured brain responses, such as the fMRI of EEG response, must be non-linear. For example, in MRI the relationship between signal intensity (SI) and T2* (which is the marker for oxygen concentration in such data) is manifestly non-linear: $\mathrm{SI} = ke^{-t/\mathrm{T2}^*}$ (except when t = 0). The relationship of T2* itself to oxygen concentration, $[O_2]$ is non-linear: $T_2^* = 1/(R + \chi[\mathrm{O}_2])$. As noted above, the brain blood flow clearly cannot increase without limit as the stimulus intensity increases. Given all of this, how and when can we treat brain responses as a linear system?

Essentially, there are two answers. One is to apply the truism, justified below, that "(*almost*) everything is linear to first order." The second is take steps to *linearize* the system.

(Almost) Everything is Linear to First Order

Consider these statements,

> "The faster you drive, the quicker you will get there."

"If you pull a rubber band twice as hard, it will become twice as long."

"If you hit a ball twice as hard, it will go twice as far."

Strictly, none of these are true. If you drive fast enough, you will most certainly lose control, destroy your car or get in an accident and perhaps never get there. If you pull a spring hard enough it will break, and if you hit a ball hard enough it will explode (other factors, such as friction also come into play). Nevertheless, each of these is true over a limited range. Mathematically, this can be shown to be true for almost any well-behaved function by appealing to the Taylor series (with which we assume the reader is familiar), which states that:

$$\begin{aligned} f(x) &= f(a)(x-a)^0 + \frac{df(a)}{dx}\frac{(x-a)^1}{1!} + \frac{d^2 f(a)}{dx^2}\frac{(x-a)^2}{2!} + \ldots + \frac{d^n f(a)}{dx^n}\frac{(x-a)^n}{n!} + \ldots \\ &= \sum_0^\infty f^{(n)}(a)\frac{(x-a)^n}{n!} \end{aligned} \qquad \text{(Eq. 6)}$$

In words, the Taylor series states that a function is equal to its value at a point, *a*, plus the weighted sums of its derivatives at that point. Notice that the second term in the series is the first derivative (the slope) at *a* multiplied by *x*. That term, simply put, is *linear*. Note also that all of the higher order terms are weighted by the factor 1/n!, so that they generally contribute less to the value of $f(x)$. Strictly speaking, the Taylor expansion of a function does not always exist, however, and is available only if the function has infinite derivatives and if the series converges over the interval in which it is applied. For most physical systems, these conditions are met to a very good degree of approximation. There are also cases where the first order term is 0; for example, the function cos(x) has a slope of 0 at x=0 so that technically that function is *not* linear to first order at that point or at integer multiples of 2π.

As we deviate further from the point *a*, the $(x\text{-}a)^n$ term becomes more significant, and the first order approximation to linearity becomes less accurate. It follows that the smaller the deviation in our starting, *a*, the more closely $f(x)$ is approximated by:

$$f(x) \approx f(a) + (x-a)\frac{df(a)}{dx}.$$

When we are analyzing small changes in magnetic resonance signal intensity (typically less than 1%), light reflection, action potential rates, etc… resulting from small perturbations in neuronal activity, the linear approximation is accurate enough for most cases. *This is a succinct expression of the commonsense observation that everything is linear to first order*. The smaller the perturbations, the less impact the higher order terms in the Taylor series have on the estimate of the function values.

There is good reason to believe that the neuronal firing rate by the way, as measured by synaptic activity, bears a near linear relationship to the observed MRI signal intensity.

Linearization

Often it is possible to *linearize* a function by transforming its output values in some manner. Consider the scatter of data points in figure A3:

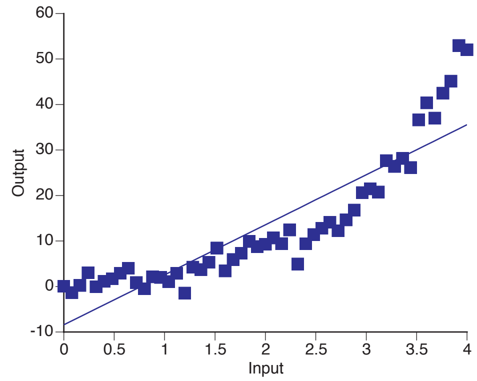


Figure A3. A nonlinear functional relationship.

Clearly, these data are not well-fitted by the "best-fit" straight line that appears in that figure, and the system is not behaving in linear manner. However, it is possible to linearize the relationship between input and output in this case by plotting the logarithm of the output against the input (Figure A4), where the linear fit is an excellent approximation. One can often linearize a system by looking at a non-linear transform of the input.

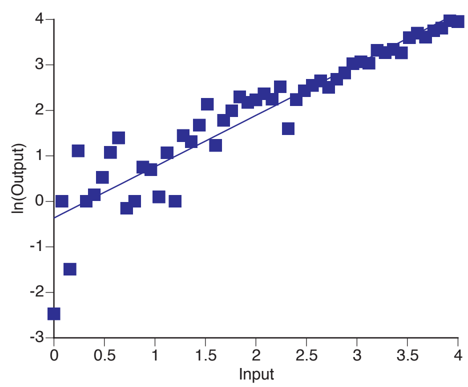


Figure A4. Log transform of the prior figure.

One way in which this might be appropriate in functional neuroimaging by any method is to consider fitting the observed signal strength to a secondary variable that might bear a non-linear or even non-monotonic relationship to the input function. For example, it is well known that when subjects listen to sounds of varying intensity, they typically rate the apparent relative loudness approximately the logarithm of the relative sound energies: When sound levels are increased by ten-fold, subjects on average report that the loudness has doubled. It might therefore make sense to evaluate the relationship between the logarithm of the sound intensity and the imaging signal strength. Some investigators have successfully linearized the observed MRI signal response by comparing it to the subjective difficulty of a task or the reaction times, which typically bear a complex relationship to fundamental stimulus parameters.

Human and Brain Responses are not *Time-invariant*

It is only in the rarest cases that people respond to a stimulus twice in exactly the same way. We accommodate to repetitive stimuli, usually by responding less to monotonously repeated trials. Our reaction times generally decrease when a cognitive task is repeated. Further, our responses are often variable as a result of hidden latent states, such as tiredness or sleep.

The typical strategies that investigators employ to address this challenge are, on the one hand, allowing the subject's response to reach a steady state by either ignoring the first few trials, overtraining the subjects or other similar strategies, or to increase the novelty of the stimuli by randomizing the stimulus order, the intertrial interval, or other stimulus parameters that might serve to diminish the extent to which subjects accommodate. In both cases, the intent is to ensure that the response to the stimulus is approximately the same each time. In these cases averaging, for example, can be an efficient way to remove noise variation from the measured responses. Experimental designs often must take this into account to produce sensible results. Here we note particularly that the assumptions of linearity that are involved in MRI signal detection depend sensitively on the details of the actual task design strategy and timing.

Convolution

Though the name may be new to the reader, convolution is a mathematical description of many common *filtering* processes. For example, when light casts the shadow of an object, the blur of that shadow is the convolution of the light source with the object it passes by. When stockbrokers create a *moving average* of share prices, this is a form of convolution. When light passes through the lens in a telescope, the focusing of the light beam is a convolution process. Interestingly, the process of convolution is generally approximately *invertible*, as long as the noise level is very well controlled; when two signals are convolved together and produce an output, if we know the output, and the value of one of those two signals, we can determine the other one. The inversion of the convolution operation is known as *deconvolution.* When the Hubble space telescope was launched, and the primary mirror was determined to be defective, *deconvolution* of the blurred astronomical images became possible, because the scientists involved had both the images the Hubble created (the output) and new the exact characteristics of the improperly manufactured mirror (one of the two inputs). They were able to successfully invert the convolution operation of the mirror to create the second input – the corrected images of the galaxies.

One of the simplest, and most familiar, forms of convolution is the *moving average*. If we are presented with a times series that has a great deal of noise, it is often possible to smooth the data by replacing each point with the average, or weighted average, of its neighbors. Figure A1 shows a typical result. Here, the time series consists of a sinusoidally varying data set, where each point has added to it a random value. In the resulting noisy time series the underlying sinusoidal fluctuation is somewhat difficult to see. The bottom curve in that figure has each point replaced by the average of itself and the four adjacent points. This simple transformation works because the sinusoidal function varies much more slowly than once every five points. The moving average is a filter that clearly tends to suppress rapidly changing elements in the data series, and we consider it to be a "low-pass" filter, because low frequencies pass by with little alteration, while high frequencies are attenuated. This commonly used form of *unweighted* moving average, in which each point contributes equally, has some unfortunate characteristics as a filter however, which we will consider in the section on the Fourier transform.

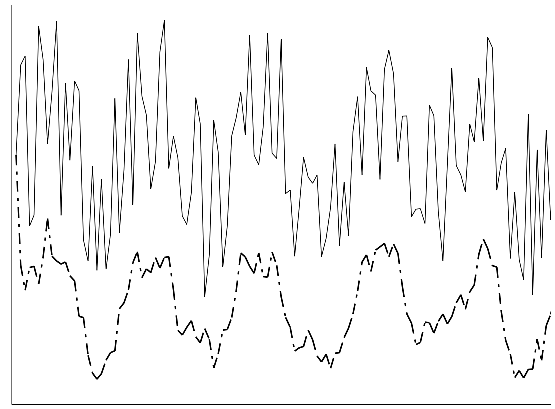


Figure A1. Noisy data smoothed by convolution.

For the moment, we will consider convolution of discrete data sets, consisting of a series of values, as opposed to continuous functions (*i.e.,* instead of looking at the continuous function, sin(x) we would look at sin($x[nt]$), where n is set of integer values. The square bracket notation is used commonly to indicate discrete values. After developing intuitions about it, and showing a few examples of its value we will consider convolution in the limit of infinitely small time steps to solve for continuous functions. Such a discrete set might be the heights of each individual in a room, whereas a continuous variable might be the height of a single individual at any point in time; we will consider continuous data sets below. For discrete data it is convenient to express the convolution process using the summation operator. For the five point unweighted moving average, the output data points, $y[n]$, are formed from the input data points, $g[n]$ (which can take any form), as follows:

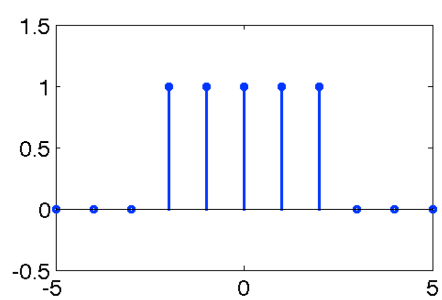
$$y_n = \sum_{n=-2}^{2} g_n \text{ or, } y[n] = \sum_{n=-2}^{2} g[n]. \qquad \text{(Eq. 1)}$$

This type of operation is expressed more broadly as the convolution of two functions with each other. One of these is the *input* series, $g[n]$, and the other is the *filter* series, $h[n]$:

$$h[n] = \ldots 0,0,0,1,1,1,1,1,0,0,0\ldots, \text{ or}$$

$h[-2] = h[-1] = h[0] = h[1] = h[2] = 1$ and is 0 at all other times. That filter series often is calaled the convolution kernel.

Graphically in this case, $h[n]$ looks like:



We say that,

$$y[n] = g[n] \otimes h[n]\,, \qquad \text{(Eq. 2)}$$

where the ⊗ operator denotes convolution. Many texts will use the asterisk, *, to indicate convolution, but this may be confusing as many computer programming languages use * to represent multiplication. The more general way to express discrete convolutions is this:

$$y[n] = g[n] \otimes h[n] = \sum_{k=-\infty}^{\infty} g[n]h[n-k]\,. \qquad \text{(Eq. 3)}$$

You can and should verify easily for yourself that when $h[n]$ is defined as in the example above, the operation of equation 3 yields the same result as equation 1. Notice that the summation limits are from -∞ to ∞, rather than the five point filter function above, allowing the convolution of infinitely extended functions for g and h if data exist for those points.

Convolution filters are used frequently in image processing. The figures below show the results of applying spatial convolution filters to the reference image, where the weightings given the filter are as shown. In this case, the pixel value at each location is replaced by the weighted average of its neighbors (ordinarily, the weightings are normalized, so that the sum of all included weights is equal to 1. In the first example, which is a "boxcar" filter similar to the one dimensional filter $h[n]$ discussed above, the 13 ones would be replaced by 1/13 instead). The second is a sharpening filter, sometimes called a Laplacian. The final example in this figure shows "Gaussian" smoothing, which has a very natural appearance, and of which we will learn more later.

Reference Image

		1		
	1	1	1	
1	1	1	1	1
	1	1	1	
		1		

	-1	
-1	5	-1
	-1	

4	9	12	9	4
6	12	16	12	6
7	16	20	16	7
6	12	16	12	6
4	9	12	9	4

Gaussian

The Impulse Response Function

The term *linear* means formally that the response of a system to two signals is the same as sum of the response to each of those signals presented independently. Multiplication is a good example of a linear function. If my system simply multiplies an input by some factor, *m*, then the response to an input *x* is simply *mx*. Likewise, the response to input *y* is *my*. If *x* and *y* are present together as the input, *x* + *y*, then the system output will be *m(x* + *y)* = *mx* + *my*, the sum of these inputs presented separately. The square root function is a good example of a *nonlinear* system; if the output of system to input *x* is $\sqrt{x}$ and to input y is $\sqrt{y}$ the output to *x* + *y* together, $\sqrt{x+y}$, which generally is *not* the sum of $\sqrt{x}$ + $\sqrt{y}$.

Linear systems have extremely important features, not the least of which is that they tend to be more mathematically tractable than non-linear systems. One extremely important property is that if we know the system response to a single isolated input and if the system is "time-invariant", meaning that its response to the same input does not vary with when the input is presented, we can predict its response, by convolution, to any arbitrary input. This is relatively easy to see for a system acting upon discrete data. We call the response of the system to a single input of unit amplitude at time 0 its *impulse response, h*[*t*]. For example, the response might fall of exponentially with time, such that $h[t] = e^{-t/\tau}$, which turns out to be a relatively common impulse response. This form of *h*[*t*] is shown graphically in figure A2.

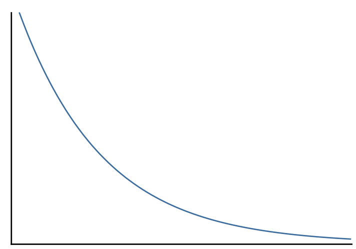


Figure A2. Exponential impulse response, y axis is amplitude, x axis is time

If we scale the input, making it *g* times as large, then the impulse response will be $gh(t) = ge^{-t/\tau}$, a scaled version of the unit impulse response. If, on the other hand, that input is delivered at a later point in time, t_0, then the response will be $gh(t-t_0) = ge^{-(t-t_0)/\tau}$. In the case, therefore, that our input consists of a time series of values, the response to that input is simply the sum of the appropriately scaled and shifted impulse responses. If the input series *g*[*t*] is made up of the values:

$$g[t] = g_0 + g_1[1] + g_2[2] + g_3[3] + \ldots + g_n[n] + \ldots \qquad \text{(Eq. 4)}$$

then the output from our linear, time-invariant, system will be:

$$\begin{aligned} y[t] &= g_0 h[t] + g_1[1]h[t-1] + g_2[2]h[t-2] + g_3[3]h[t-3] + \ldots + g_n[n]h[t-n] + \ldots \\ &= g[t] \otimes h[t], \end{aligned} \qquad \text{, (Eq. 5)}$$

the convolution of *x*[*t*] with the impulse response, *h*[*t*].

We see this expressed frequently in fMRI, for example, in the prediction of the observed response to a known input, which allows us to search for that predicted response in the resulting

MRI image series. The idea is to determine an *impulse response* for the blood flow of the brain as measured by fMRI, and then to model the response to an arbitrary input as the summed responses to a series of scaled and shifted impulses. Conventionally, this brain impulse response is called the hemodynamic response function, or *hrf*. Formally, if our input is modeled as a series of values $x[t]$ as in equation 4, the expected response in the fMRI signal is:

$$y[t] = x[t] \otimes hrf[t]. \qquad \text{(Eq. 6)}$$

This idea is fine but for 3 problems:

1. We don't know the exact form of the *hrf.*
2. The brain response is not linear. It is obvious, for example, that the brain response to an arbitrarily large input is not itself arbitrarily large.
3. The brain response is not time-invariant.
   For example, as people learn, we can expect the response to change.

Because response modeling is so fundamental to fMRI, the general outline bears repeating.

The general approach to finding areas of brain activity by MRI or PET is to:

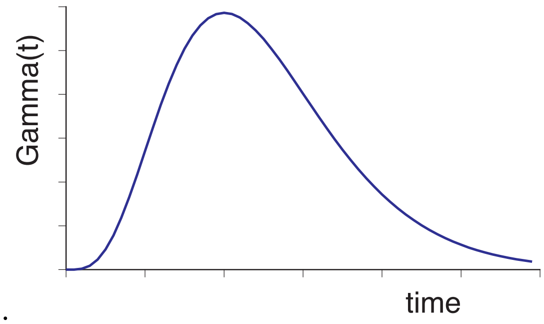
> **Observe the signal changes in signal intensity while subjects perform a task of some sort.**
>
> **Predict the signal response to that task by modeling, almost always by convolving the task timing with a nominal hemodynamic response function (and radioactive decay function, etc…, for PET.**
>
> **Use statistical methods to search for signal intensity fluctuations in the MRI data that correlate with our modeled responses.**

Everything depends on the accuracy of our model. In most cases, this depends upon assumptions of linearity. Fortunately, there are a few simplifying truisms that we can apply to the problem.

We don't know the exact form of the hrf.

While it is impossible at present to predict the *hrf* from first principles, it is possible to make plausible measurements of it. One way to do so is to observe the fMRI response experimentally to inputs that approximate an impulse. A good example comes from the work of Savoy, et al., reproduced by permission below. Savoy and his colleagues simply observed the MRI signal following brief (<1 second) light flashes, averaging these to form a good estimate of the natural response. The observed response develops relatively slowly, reaching a peak in seven seconds or so, then falls back to baseline over about 15 seconds. Although one could use the observed response directly, it is somewhat easier to use an approximating function. One candidate is the "gamma variate" (Γ) function which has a shape that can be matched closely to the observed brain response using only three parameters:

$$\Gamma(t) = \begin{cases} kt^A e^{-Bt} & t \geq 0 \\ 0 & t < 0 \end{cases}.$$



The interpretation of the three parameters might be that k represents an overall gain or scaling, A controls the rate of rise, and B controls the rate of fall. Here, the important point is that if this adequately models the *hrf*, the convolution process allows us to model the brain response to an input of essentially arbitrary time course.

Convolution in the Limit: Continuous functions

While we presented convolution in the domain of a discrete time series of inputs where it is perhaps easier to gain an intuitive understanding, the operation is readily extended to the domain of continuous functions. Looking once again at equation 3:

$$y[n] = g[n] \otimes h[n] = \sum_{k=-\infty}^{\infty} g[n]h[n-k]. \qquad \text{(Eq. 3)}$$

we see that in the discrete case, it is just the sum of the point by point multiplication of one function, $g[n]$ and time reverse of a second function, $g[n]$. Considering these as time series, the function $x[t]$ for example, represents a value applied for an interval, Δt, that is the time between successive points, for example a light intensity applied for a time of 1 second. The total stimulus energy during that interval would be the intensity of the light pulse multiplied by Δt. As the interval becomes smaller we simply require more points to cover the same total period. Convolution in the continuous domain is simply the limit of this sum as the interval between points, Δt, goes to zero:

$$g(t) \otimes h(t) = \int_{-\infty}^{\infty} g(\tau)h(t-\tau)d\tau \qquad \text{(Eq. 7)}$$

A Few Properties of Convolution

Convolution is Commutative. By substitution of the variable, $\tau = t - u$, into equation 7, so that $du=-d\tau$, we can see that:

$$\begin{aligned} f(t) \otimes g(t) &= \int_{\tau=-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \\ &= \int_{t-u=-\infty}^{\infty} f(t-u) g(u) d(t-u) \\ &= -\int_{u=\infty}^{-\infty} f(t-u) g(u) du \\ &= \int_{-\infty}^{\infty} g(u) f(t-u) du \end{aligned} \qquad \text{(Eq. 8)}$$

The last equation is of course the definition of $g(t) \otimes f(t)$, proving the commutativity of convolution. For our purposes this means, for example, that when we consider the modeled time series as the convolution of the stimulus, $s(t)$ by the *hrf*, we can present these elements in either order:

$$s(t) \otimes hrf = hrf \otimes s(t). \qquad \text{(Eq. 9)}$$

Convolution is distributive, as can be seen easily:

$$\begin{aligned} f(t) \otimes (g(t) + h(t)) &= \int_{-\infty}^{\infty} f(\tau) g(t-\tau) + h(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(\tau) g(t-\tau) + \int_{-\infty}^{\infty} f(\tau) h(t-\tau) d\tau \\ &= f(t) \otimes g(t) + f(t) \otimes h(t). \end{aligned} \qquad \text{(Eq. 10)}$$

as well as associative:

$$\begin{aligned} f(t) \otimes (g(t) \otimes h(t)) &= f(t) \otimes \int_{-\infty}^{\infty} g(\tau) h(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(\varphi) \int_{-\infty}^{\infty} g(\tau) h(t-\tau-\varphi)\, d\tau\, d\varphi \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\varphi) h(t-\tau-\varphi)\, d\varphi g(\tau) d\tau \\ &= \int_{-\infty}^{\infty} f(\varphi) h(t-\varphi) d\varphi \otimes g(t) \\ &= (f(t) \otimes g(t)) \otimes h(t). \end{aligned}$$

These properties of convolution allow us a great deal of freedom in modeling the responses of a complex system such that we can chain together different elements, each of which performs a separate convolution operation on the output data, and still predict output in a simple manner. One convolution step might be the transformation of neuronal activity $A(t)$ to blood flow hrf:

$s(t) = A(t) \otimes hrf$, and another might be the convolution of those data by the spatial blurring of the imager, $I(t)$. The total response $R(t)$ would be $R(t) = s(t) \otimes I(t) = A(t) \otimes hrf \otimes I(t)$.

The Fourier Transform

Basics

The Fourier transform, and its newer relatives such as the wavelet transform, play fundamental roles in the analysis of time varying systems and in image processing. Especially in magnetic resonance imaging, it is impossible to gain a strong understanding of the data acquisition, processing and artifacts, without a set of equivalently strong intuitions in the Fourier transform. That being said, you are unlikely to have to *solve* Fourier equations as part of your work in neuroimaging; the primary goal should be to develop an intuitive understanding of the method.

The Fourier transform typically is interpreted as the ability to express an (almost completely) arbitrary function as a series of sinusoids at different *amplitudes*, *frequencies*, and *phases*. Importantly, the Fourier transform is *invertible* in a special way: It is essentially its own inverse (actually its reflection – more on that later). Applying the transform twice returns the original function, in much the same that the operation y=1/x applied twice returns x. A function expressed as values that change with time is transformed to a set of frequencies; these can in turn be expressed as the original time function. In the imaging system MRI data are received as a time-varying signal. The spatial encoding process results in their being transformed such that their frequency depends on location. This means that the natural state of the raw MRI signal is the Fourier transform of the image. What we see as a spatial map is actually a map of the intensity of the magnetic resonance signal as a function of frequency. The Fourier transform (or "FT") has many useful consequences, in that the analysis of signals in the "frequency" (or transformed) domain is often easier than the study of the time domain signal. Conversely, we sometimes receive data in the frequency domain and can analyze it easier in time.

> The Fourier transform is named after its discoverer, Jean Baptiste Joseph Fourier, a brilliant mathematician whose personal life was caught up in the upheavals of France under Napoleon, for whom he served somewhat unwillingly, originally as scientific advisor in 1798 in the failed campaign to invade Egypt. While he wished to pursue his mathematical research, Napoleon had other plans for him, and Fourier was appointed as a prefect in Grenoble. Though he had broad administrative and political duties, it was here that he developed the transform as part of his research into heat propagation. The ideas were so novel and so ground-breaking that they became sources of controversy immediately. The greatest mathematicians of his time, his former teachers Lagrange and Laplace disputed these results and held back publication of this work. Later, Laplace, Poisson and Biot each laid claim to priority in the discovery; such is the way of scientific progress.

Fortunately, the FT is not too hard to understand. It is based very fundamentally on the Euler relation, which we will discuss first.

The Euler relation

The so-called Euler relation is one of the most remarkable results in mathematics, as it embodies some of the most fundamental constants in math, specifically, "*e*" the root of natural logarithms, π and *i,* the (imaginary) square root of –1 and the trigonometric ratios. It is basic to many forms of engineering, such as circuit analysis. Succinctly, the Euler relation is this:

$$e^{ix} = \cos x + i \sin x\,, \qquad \text{(Eq. 12)}$$

where i is the *imaginary* number: the square root of negative one.

Optional: Deriving the Taylor and Maclaurin series (informally)

We will try to express $f(x)$ as a polynomial of order n:

$$f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \ldots + f_n x^n + R_{n+1}(x).$$

Our task is to find $\{f_i\}_{i=0}^{n}$ that minimizes the error.

Assume that $R_{n+1}(x) = 0$ for all x, if the values of f_i are correct.

At $x = 0$ we can find the minimum error:

$$f(0) = f_0.$$

Taking the derivative the original polynomial:

$$f'(x) = f_1 + 2f_2 x + 3f_3 x^2 + \ldots + nf_n x^{n-1} + R_{n+1}(x)\,,$$

but remember that $R_{n+1}(x) = 0$ (our original boundary assumption). Now look at higher derivatives:

$$\begin{aligned} f'(0) &= f_1 \\ f''(0) &= 2f_2 \\ f'''(0) &= 3 \times 2 f_3 \\ &\ldots \\ f^{(n)}(0) &= n! f_n. \end{aligned}$$

Solving all at $x = 0$:

$$\begin{aligned} f_o &= f(0) \\ f_1 &= f'(0) \\ f_2 &= \frac{f''(0)}{2} \\ f_3 &= \frac{f'''(0)}{3 \times 2} \\ &\ldots \\ f_n &= \frac{f^n(0)}{n!}. \end{aligned}$$

Therefore,

$$f(x) \approx \sum_{n=1}^{\infty} \frac{f^{(n)}(0)}{n!} x^n .$$

Note that this leaves behind the sticky problem of proving that $R_{n+1}(x)$ is small over a sufficiently large range of x. An exercise for the reader

We can derive the Euler relation starting with the Taylor series as we reminded you earlier. It is possible to express a function as a series of derivatives about a point. Let $f(x)$ be infinitely differentiable at a. The Taylor series expansion for $f(x)$ is:

$$\begin{aligned} f(x) &= f(a)(x-a)^0 + \frac{df(a)}{dx}\frac{(x-a)^1}{1!} + \frac{d^2 f(a)}{dx^2}\frac{(x-a)^2}{2!} + \ldots + \frac{d^n f(a)}{dx^n}\frac{(x-a)^n}{n!} + \ldots \\ &= \sum_0^{\infty} f^{(n)}(a)\frac{(x-a)^n}{n!} \end{aligned} \quad , \quad \text{(Eq. 13)}$$

where $f^{(n)}$ is the n^{th} derivative of f.

If we set a equal to zero, expanding the function $f(x)$ around its value at zero (resulting in the special case of the Maclaurin series), this can be written more compactly as:

$$\begin{aligned} f(x) &= f(0) + \frac{df(0)}{dx}x + \frac{d^2 f(0)}{dx^2}\frac{x^2}{2} + \frac{d^3 f(0)}{dx^3}\frac{x^3}{3!} + \ldots + \frac{d^n f(0)}{dx^3}\frac{x^n}{n!} + \ldots \\ f(x) &= \sum_{n=0}^{\infty} \frac{f^{(n)} x^n}{n!}. \end{aligned} \quad . \quad \text{(Eq. 14)}$$

Let's look at this for $f(x) = \sin(x)$, recalling that the derivative of $\sin(x)dx = \cos(x)dx$ and the derivative of $\cos(x)dx = -\sin(x)dx$:

$$\begin{aligned} \sin(x) &= \sin(0) + x\cos(0) - \frac{x^2 \sin(0)}{2} - \frac{x^3 \cos(0)}{3!} + \frac{x^4 \sin(0)}{4!} + \frac{x^5 \cos(0)}{5!} + \ldots \\ &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \end{aligned} \quad \text{(Eq. 15)}$$

Similarly, $\cos(x)$ can be expressed as:

$$\begin{aligned} \cos(x) &= \cos(0) - x\sin(0) - \frac{x^2 \cos(0)}{2} + \frac{x^3 \sin(0)}{3!} + \frac{x^4 \cos(0)}{4!} - \frac{x^5 \sin(0)}{5!} + \ldots \\ &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots \end{aligned} \quad \text{(Eq. 16)}$$

Now, let's look at the expansion of e^u, noting that the derivative of e^u is $e^u du$:

$$e^u = 1 + u + \frac{u^2}{2!} + \frac{u^3}{3!} + \frac{u^4}{4!} + \frac{u^5}{5!} + \ldots \quad \text{(Eq. 17)}$$

Examining the expansions of e^u, sin(x) and cos(x), you can see that the terms in the sine and cosine expansion seem to sum in an interesting way to the expansion of e^u. In fact, when you substitute $u = ix$, where i is the square root of –1, you can see that e^{ix} is expanded as:

$$\begin{aligned} e^{ix} &= 1 + ix - \frac{x^2}{2!} - i\frac{x^3}{3!} + \frac{x^4}{4!} + i\frac{x^5}{5!} - \frac{x^6}{6!} - i\frac{x^7}{7!} + \ldots \\ &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ i\left[x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots\right] \\ &= \cos(x) + i\sin(x) \end{aligned} \qquad \text{(Eq. 18)}$$

which *is* the Euler relation. That is, *e* raised to an imaginary exponent can be seen to be the sum of a series of (real) cosines and (imaginary) cosines.

The Fourier Transform

Stated mathematically, the Fourier transform of $f(x)$, which we will call $\boldsymbol{F}(s)$, is:

$$\begin{aligned} \mathscr{F}(s) &= \int_{-\infty}^{\infty} f(x)[\cos(2\pi sx) - i\sin(2\pi sx)]\,dx \\ &= \int_{-\infty}^{\infty} e^{-2\pi isx} f(x)\,dx. \end{aligned} \qquad \text{(Eq. 19)}$$

The second form simply applies the Euler relation to the first, to make a more compact expression. Usually it is expressed in the second form, as the function multiplied by a complex (and arbitrary-looking) exponential. The first form may be easier to understand, as it implies finding the product of the function $f(x)$ at each x value with both a sine and a cosine series having a frequency dependent on s, with the sinusoids multiplied by the imaginary number, i. This expression is structurally similar to the Euler relation, as well. The variable names, x and s, are arbitrary of course. What is notable, however, is that the Fourier transform of one variable becomes a function of another. The two are related, of course, with one acting like the inverse of the other. For example, if the input is a function of time, t, its transform is a function of frequency, behaving as $1/t$. This produces an odd looking glass character to Fourier transform space, which we will see as we continue. Note that we s may, in general, be a complex number, which we express as $s = \sigma + j\omega$.

The inverse Fourier transform is defined similarly. The concept of an inverse of course is that applying it gets you back to the original form. If we use $\mathscr{F}(s)$ to indicate the Fourier transform of $f(t)$, and $\mathscr{F}^{-1}$ is its inverse: $\mathscr{F}^{-1}[\mathscr{F}(s)] = f(t)$.

$$\begin{aligned} \mathscr{F}^{-1}[\mathscr{F}(s)] = f(t) &= \int_{-\infty}^{\infty} f(t)e^{+2\pi ist}\,ds \\ &= \int_{-\infty}^{\infty} f(t)[\cos(2\pi st) + i\sin(2\pi st)]\,dt \end{aligned} \qquad \text{(Eq. 20)}$$

The only break in symmetry between the inverse and forward transform is the sign of the complex exponential, $+2\pi ist$, rather than $-2\pi ist$.

A common interpretation is that the parameter *s* contains only an imaginary component and is expressed as frequency. As is clear from the first form in Equation 19, changes in *s* appear as changes in the frequency of the sinusoidal components. Many people, physicists in particular, choose to express frequency in angular (radian) measure: $\omega = 2\pi f$. Radian measure really *is* simpler. In this case,

$$F(j\omega) = \mathscr{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt. \qquad \text{(Eq. 21)}$$

The inverse transform therefore must become:

$$\mathscr{F}^{-1}\{F(j\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(\omega) e^{+j\omega t}\, d\omega. \qquad \text{(Eq. 22)}$$

This breaks the symmetry of the forward and inverse transform, which is unsightly. Therefore, you will sometimes see the forward and inverse transform expressed in an alternate form:

$$F(j\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t) e^{-j\omega t}\, dt, \text{ and}$$

$$\mathscr{F}^{-1}\{F(j\omega)\} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(\omega) e^{+j\omega t}\, d\omega. \qquad \text{(Eq. 22)}$$

Notice that we express this in *t* and ω to emphasize that we might be dealing with time domain functions. To develop an intuition of how this works, consider a periodic square wave function with a cycle time of one second, and its products with cosine and sine functions at 1, 2, 3, 4 & 5 cycles/second, as shown in figure A5.

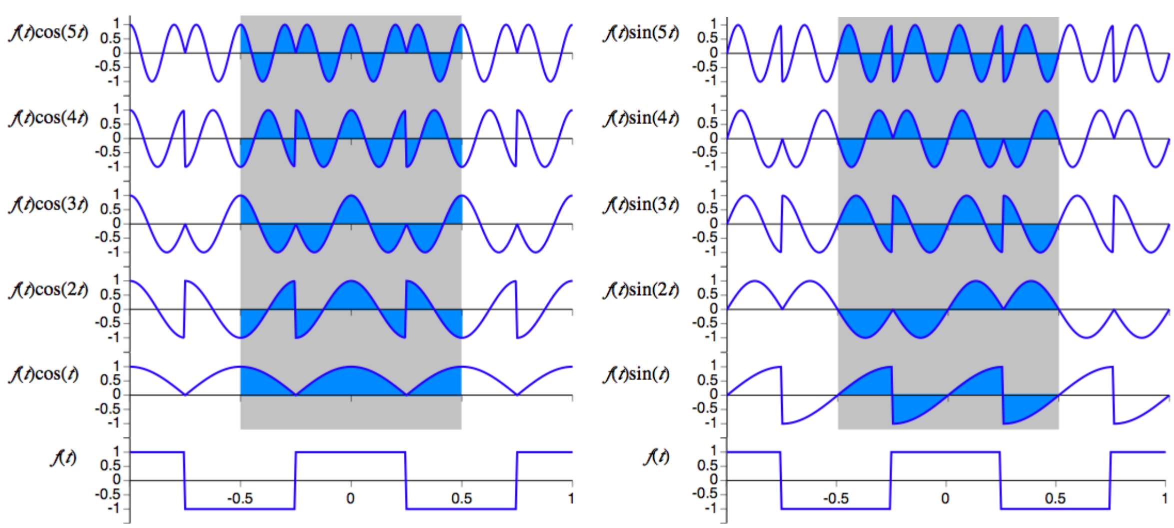


Figure A5. The function *f*(*t*), at the bottom on left and right is a square wave with a period of 1 second. Shown above are the products of *f*(*t*) with cosine (left) and sine (right) functions at frequencies of 1, 2, 3, 4 and 5 Hz (cycles/second). The shaded regions are the *area* over just one cycle of *f*(*t*). Note that the area of the even numbered cosine waves and all of the sine waves is equal to zero over the one cycle period. The Fourier transform of *f*(*t*), which is the sum of all of these products, contains *only* the cosine products at the odd multiples of the frequency of *f*(*t*).

In that figure, a one Hertz (Hertz or Hz = one cycle/second) square wave is shown at bottom as $f(t)$. Above are the products of f(t) with cosine functions (*left*) and sine functions (*right*) at frequencies of 1Hz, 2Hz, 3Hz, 4Hz and 5Hz; the shaded areas are the products over one cycle of $f(t)$. It should be obvious form looking at this drawing that the areas of all of the products with sine waves (right) have an area of zero over one cycle. It should be equally clear that the areas of the products with the *even* cosine frequencies are also zero. This holds true for arbitrarily high frequencies, by the way. The Fourier transform of $f(t)$ is the sum of all of these products, where the sine products are multiplied by i. We can see that for this particular function, $f(t)$, the Fourier transform, $\boldsymbol{F}\{f(\mathrm{t})\}$ has the form:

$$\boldsymbol{F}\{f(\mathrm{t})\} = A_1\cos(t) + A_3\cos(3t) + A_5\cos(5t) + \ldots + A_{2N+1}\cos((2N+1)t) + \ldots$$

for N=0, 1, 2, 3, 4, etc….

A couple of things have been buried here. The first is that we showed the areas over just one cycle of $f(t)$. This is fair only if we understand that the sums must be computed symmetrically about t=0. Secondly, we specified only integer multiples of the frequency, forming the *discrete* Fourier series. As it happens, the integrals of frequencies that are *not* odd multiples of 1 Hz will be equal to zero over a sufficiently long time interval (we leave that to your imagination.)

Fourier transform examples

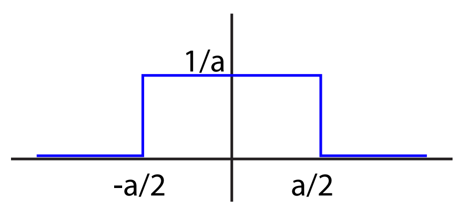
Looking again at the Fourier integral, we will derive a few transform pairs and then consider some general properties of the Fourier transform.

Starting with the definition:

$$\mathscr{F}(s) = \int_{-\infty}^{\infty} f(x)e^{-2\pi isx}\,dx.$$

we will look first at the Fourier transform of a square function in time (substituting t for x in the definition.) This expresses, for example, a signal that starts and ends at distinct time, such as a light flash might. Specifically, we have a pulse of unit area and duration, a. We can define the function analytically as:

$$f(t) = \begin{cases} \frac{1}{a} \text{ for } -\frac{a}{2} < t < \frac{a}{2} \\ 0 \text{ otherwise} \end{cases}.$$



Let $u = -i2\pi st$, and $du = -i2\pi s$. From which we can calculate the transform:

$$
\begin{aligned}
\mathscr{F}(s) &= \frac{1}{a}\int_{-a/2}^{a/2} e^{-2\pi ist}\,dt \\
&= \frac{1}{2\pi ias}\int_{-a/2}^{a/2} e^{u}\,du \\
&= -\left(e^{-2\pi ias/2} - e^{2\pi ias/2}\right)/\,2\pi ias \\
&= \left(e^{2\pi ias/2} - e^{-2\pi ias/2}\right)/\,2\pi ias \\
&= \frac{\cos(\pi as) + i\sin(\pi as) - \cos(-\pi as) + i\sin(\pi as))}{2\pi ias} \\
&= \frac{2i\sin(\pi as)}{2\pi ias} \\
&= \frac{\sin(\pi as)}{\pi as}.
\end{aligned}
$$

The result, sin(π*as*)/πa*s* is called a "sinc" function, and appears as shown in figure A6 below. Observe that this function has infinite tails into the highest and lowest frequencies. The implication is that a finite duration pulse (for example, if the square function is expressed as a function of *time*) contains an *infinite* range of frequencies. Since the FT is invertible, we could also imagine the square function above to represent a range of frequencies (for example, the kind of frequency range you would want to perform slice-selective excitation in an MRI device.) The transform would be the time-domain signal required to produce those frequency characteristics; it is infinitely extensive in time.

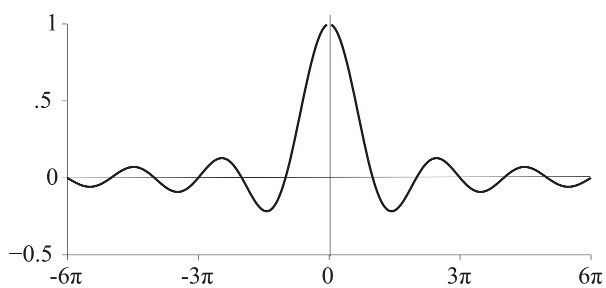


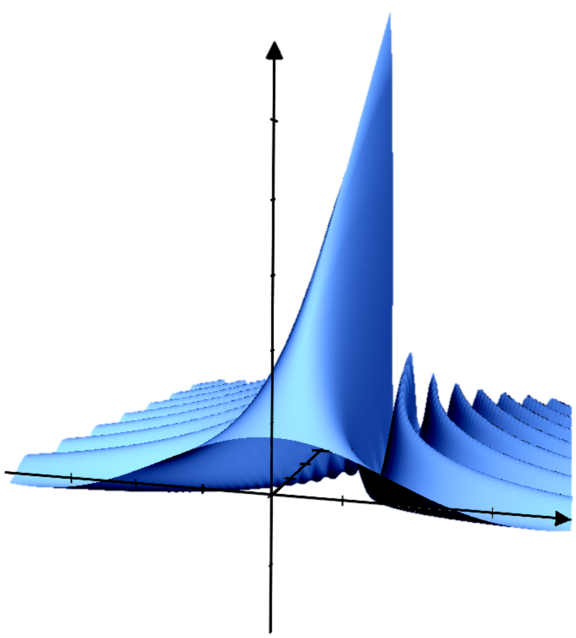
Figure A6. The sinc function, $\sin(\pi as)/\pi as$, with a=1

This particular transform pair: *Square* to *Sinc*, or *Sinc* to *Square*, occurs time and again in signal and image processing and analysis, so that it is worth considering a few of its properties. First, as noted, while the function is discrete in one domain, it is infinite in the other. Speaking somewhat loosely, this reflects the fact that the square function is defined to change instantaneously from 0 to its maximum. Such transitions require energy at infinitely high frequencies.

Another notable property is that only a modest fraction of the energy in the sinc function is contained in its center peak. The area from 0.5 to 1, for example, is about one-quarter that of the area from 0 to 0.5. This means that truncated approximations to the sinc function tend to be very poor, as so much of the energy is in the tails of that function. This is the cause of substantial artifacts in real world imaging. For example, as expanded on in the sections concerning the MRI imaging process itself, gathering signal from a single *slice* of tissue requires creating a waveform that has energy in a well defined band of frequencies. To create a square slice profile, where we receive all of the signal from only a narrow slice in the brain would require us to generate a sinc-shaped waveform that is infinite in time. In practice, we are forced to approximate the infinite time series with a waveform that lasts only a few milliseconds. This results in a far from ideal slice selection.

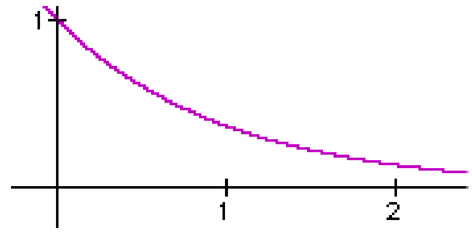
In a similar vein, notice that the amplitude of much of the sinc function is negative. In fact, almost as much of the area is negative as is positive if we integrate from further out in the tails. This may have important implications as we try to understand activation magnitude in the context of functional MRI.

What happens if we change the width of the square function (by altering the value of a in figure A6? Looking at the form of the sinc function $\boldsymbol{F}(s) = \sin(2\pi as)/\pi as$, increasing a increases the frequency of the sine function in the denominator, which has the effect of condensing the shape of $\boldsymbol{F}(s)$ and of increasing the peak height of the central lobes. Figure A7 is a plot of the sinc function as a is increased over a ten-fold interval. Practically speaking, if we imagine the square function to be, for example, a brief impulse such as a light flash or a sound, we see that making the impulse briefer (reducing a) results in a widening of the frequency content of the signal. In the limit of an instantaneous pulse the Fourier transform, $\boldsymbol{F}(s)$, contains equal energy at all frequencies.



A7. Sinc dilation

For a decaying exponential (like a decaying optical, PET or MRI signal):



$$\text{let } f(x) = e^{-a|x|}.$$

$$\mathscr{F}(s) = \int_{-\infty}^{\infty} e^{-a|x|} e^{-2\pi isx}\, dx$$

$$= \int_{-\infty}^{\infty} e^{-a|x|} \left[\cos(-2\pi isx) + i\sin(-2\pi isx)\right] dx$$

$$= \int_{0}^{\infty} e^{-a|x|} \cos(-2\pi isx)\, dx = \int_{0}^{\infty} e^{-a|x|} \cos(2\pi isx)\, dx$$

$$= 2\,\mathrm{Re}\left[\int_{0}^{\infty} e^{-a|x|} e^{2\pi isx}\, dx\right]$$

$$= 2\,\mathrm{Re}\left[\int_{0}^{\infty} e^{(2\pi is - a)x}\, dx\right].$$

$$\text{let } u = (2\pi is - a)x,\ du = 2\pi is - a:$$

$$\mathscr{F}(s) = 2\,\mathrm{Re}\left[\frac{1}{2\pi is - a}\int_{0}^{\infty} e^{u}\, du\right]$$

$$= 2\,\mathrm{Re}\left[\frac{1}{2\pi is - a} e^{(2\pi is - a)x}\right]_{x=0}^{x=\infty}$$

$$= 2\,\mathrm{Re}\left[\frac{1}{2\pi is - a} e^{2\pi isx} e^{-ax}\right]_{x=0}^{x=\infty}$$

$$= 2\,\mathrm{Re}\left[\frac{1}{2\pi is - a}(-1)\right]$$

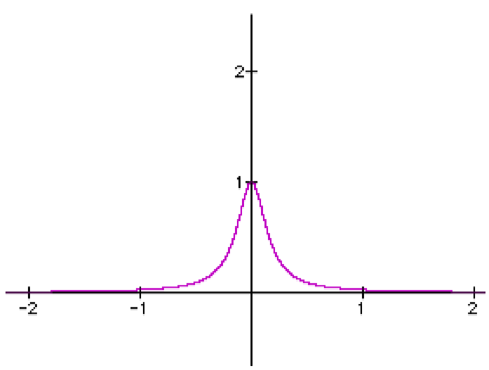
$$= 2\,\mathrm{Re}\left[\frac{-1}{2\pi is - a}\frac{2\pi is + a}{2\pi is + a}\right]$$

$$= \mathrm{Re}\left[\frac{-4\pi is - 2a}{-4\pi^2 s^2 - a^2}\right]$$

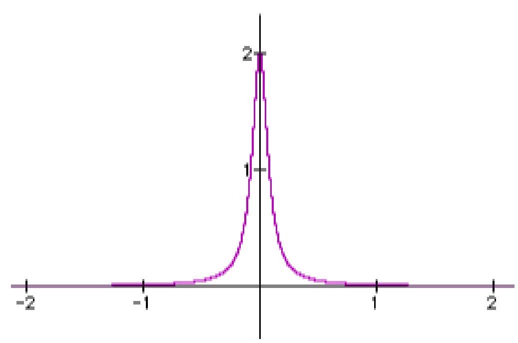
$$\mathscr{F}(s) = \frac{2a}{4\pi^2 s^2 + a^2}, \text{ which is known as a Lorentzian.}$$

A special consideration for MRI: If we substitute T2 or T2* for a, we see that the frequency width (spectral width) of a long T2 sample is narrower than that of a short T2 sample. *Thought question*: what does this imply about our ability to use frequency selection to find slices in a short T2 sample?

a=1:



a=2:



Gaussian shapes are ubiquitous, appearing in many cases in which there is measurement uncertaintly. Finding the Fourier transform of a Gaussian, $f(x) = e^{-ax^2}$ is tricky.

$$\text{Let } \pi u^2 = ax^2$$

$$u^2 = {ax^2}/{\pi}$$

$$u = {x\sqrt{a}}/{\sqrt{\pi}}$$

$$x = {u\sqrt{\pi}}/{\sqrt{a}}$$

Substituting for x:

$$\mathscr{F}(s) = \int_{-\infty}^{\infty} e^{-\pi u^2} e^{-i2\pi sx}\, dx$$

$$= \int_{-\infty}^{\infty} e^{-\pi u^2} e^{-i2\pi su\sqrt{\pi}/\sqrt{a}}\, dx$$

$$= \int_{-\infty}^{\infty} e^{-\pi(u^2 + i2\pi su\sqrt{\pi}/\sqrt{a} - \pi s^2/a)} e^{-\pi s^2/a}$$

$$= e^{-\pi s^2/a} \int_{-\infty}^{\infty} e^{-\pi(u^2 + i2\pi su\sqrt{\pi}/\sqrt{a} - \pi s^2/a}$$

$$= e^{-\pi s^2/a} \int_{-\infty}^{\infty} e^{-\pi(u^2 + is\sqrt{\pi}/\sqrt{a})^2}\, dx.$$

Note that:

$$\frac{d(u + is\sqrt{\pi}/\sqrt{a})}{dx} = \frac{d(x\sqrt{a}/\sqrt{\pi} + is\sqrt{\pi}/\sqrt{a})}{dx}$$

$$= {x\sqrt{a}}/{\sqrt{\pi}},$$

so that:

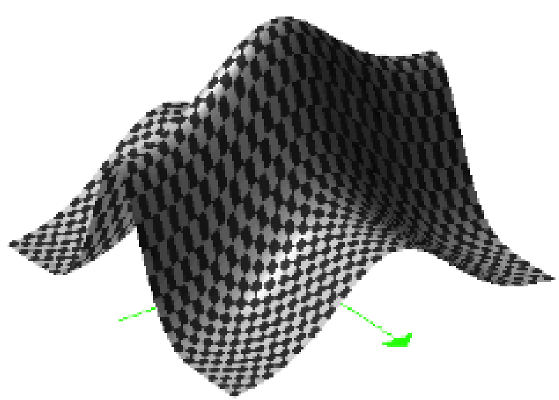
$$\mathscr{F}(s) = e^{-\pi s^2/a} \frac{\sqrt{\pi}}{\sqrt{a}} \int_{-\infty}^{\infty} e^{-\pi(u^2 + is\sqrt{\pi}/\sqrt{a})^2}\, du$$

It is known that:

$$\int_{-\infty}^{\infty} e^{-\pi^2 x^2}\, dx = 1,$$

thus:

$$\mathscr{F}(s) = \frac{\sqrt{\pi} e^{-\pi s^2/a}}{\sqrt{a}}.$$



The FT of a gaussian is a gaussian!

The two axes of this graph represent $f(\mathrm{x})$ and $\mathscr{F}(\mathrm{s})$. Note that as a increases $f(x)$ becomes narrower and $\mathscr{F}(\mathrm{s})$ becomes broader.

Odd and Even

An arbitrary function can be broken down into is Even (E(x)) and Odd (O(x)) components, where the even components have the same values for both positive and negative values of *x*, whereas the values of the odd components have the opposite sign for positive and negative *x*:

$$E(x) = \{ f(x) + f(-x) \}/2$$

$$O(x) = \{ f(x) - f(-x) \}/2.$$

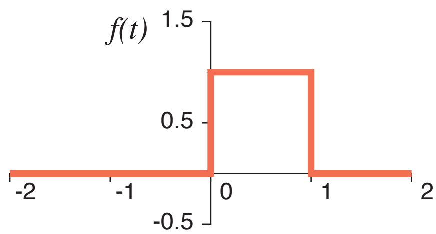
These are unique.

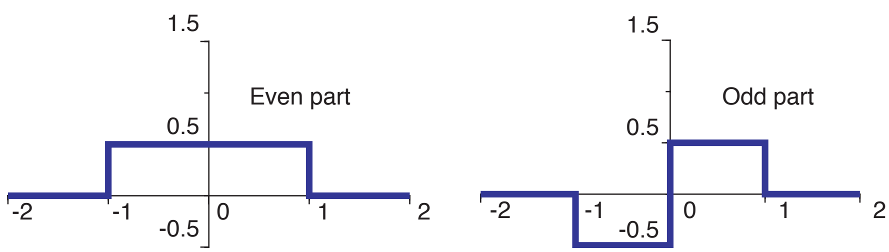
Proof: if $f(x) = E_1(x) + O_1(x) = E_2(x) + O_2(x)$, then $E_1(x) - E_2(x) = O_1(x) - O_2(x)$.

But $E_1(x) - E_2(x)$ is even, and $O_1(x) - O_2(x)$ is odd. Therefore $E_1(x) - E_2(x) = O_1(x) - O_2(x) = 0$.

For example, suppose we have a function, *f*(t), defined as $f(t) = \begin{cases} 1 \text{ when } 0 < t < 1 \\ 0: \quad \text{otherwise} \end{cases}$.

That function appears in red below. You can use the equations above to show easily, that the even and odd parts of *f*(t) are as shown in blue below:





Notice that the function *f(t)* is clearly the sum of the even and odd parts. If the function were shifted to the left by one half, it would be purely even, of course.

The oddness (and even-ness) of the Fourier Transform:

Let f(x) = E(x) + O(x). From which we note that:

if a function is:	*its Fourier transform is:*
Real and Even	Real and Even
Real and Odd	Imaginary and Odd
Imaginary and Even	Imaginary and Even
Complex* and Even	Complex and Even
Complex and Odd	Complex and Odd
Real and Asymmetrical	Complex and Hermitian
Imaginary and Asymmetrical	Complex and anti-Hermitian
Real Even + Imaginary Odd	Real
Real Odd + Imaginary Even	Imaginary
Even	Even
Odd	Odd

**we didn't say this earlier, but the Fourier transform works the same with* complex *inputs.*

Hermitian means if $f(x) = a+ib$, then $f(-x) = a-ib$

Another way to think about this is that the cosine part of the FT is even, and the sine part is odd. Because of the way the Euler relation shows up, we call the sine part *imaginary* and the cosine part *real*.

The Shift theorem

Suppose that we have a function, f, of t. If f is shifted in time, it will take on values $y = f(t-a)$. If we take the Fourier transform of this shifted input, we have:

$$\begin{aligned}\mathscr{F}[f(t-a)](s) &= \int_{-\infty}^{\infty} f(t-a)e^{-2\pi ist}\,dt \\ &= e^{-2\pi isa}\int_{-\infty}^{\infty} f(t-a)e^{-2\pi ist}e^{2\pi isa}\,dt \\ &= e^{-2\pi isa}\int_{-\infty}^{\infty} f(t-a)e^{-2\pi is(t-a)}\,dt.\end{aligned}$$

Let $u = t-a$ and $du = dt$: $\mathscr{F}$

$$\begin{aligned}&= e^{-2\pi isa}\int_{-\infty}^{\infty} f(u)e^{-2\pi isu}du \\ &= e^{-2\pi isa}\mathrm{F}[f(t)](s) \\ &= (\cos(2\pi sa) - i\sin(2\pi sa)F[f(t)](s)\end{aligned}$$

$$\begin{aligned}&= e^{-2\pi isa}\int_{-\infty}^{\infty} f(u)e^{-2\pi isu}du \\ &= e^{-2\pi isa}\mathrm{F}[f(t)](s) \\ &= (\cos(2\pi sa) - i\sin(2\pi sa)F[f(t)](s)\end{aligned}$$

What this means is that shifting the input function, $f(t)$, in time results in a *frequency-dependent phase shift*. That is, the Fourier transform of a time-shifted input contains the same frequencies, but at a phase that is a smooth function of the frequency.

There are many applications for this result. In MRI for example, interpreting t as *time,* if we time shift the acquisition (e.g. a non-centered readout), then the image (which is its FT) will have a frequency-dependent phase shift. That is, the phase will vary smoothly with position. For this reason, we collect the Magnitude image in most cases, as the phase is in general not of interest (and is not affected by this time shift).

Looking Glass Land

The fact that the FT expresses a time-domain signal as frequency ($1/t$) gives it some oddly inverted properties. For example, something that changes very slowly in time tends to have a very narrow range of frequencies (makes sense, doesn't it?). If we think about how the function and its transform look as a graph, however, the time domain signal looks broad (it changes slowly) and the transformed, frequency domain signal is narrow. This inverse similarity principle is very general (and is proven in later pages of this handout). It can be useful in forming intuitions about the Fourier transform pairs.

The Similarity Theorem of the Fourier transform:

The derivation:

$$\mathscr{F}[ax](s) = \int_{-\infty}^{\infty} f(ax)e^{-2\pi ixs}\, dx$$

$$= \frac{1}{|a|}\int_{-\infty}^{\infty} f(ax)e^{-2\pi ixs/a} d(ax)$$

$$= \frac{1}{|a|}\mathscr{F}\left(\frac{s}{a}\right).$$

shows that as the input function dilates, its Fourier transform contracts, and its amplitude increases so that the area remains constant.

The Fourier Convolution Theorem

The Fourier transform and convolution bear an important and deep relationship. Namely, the Fourier transform of the convolution of two functions is equal to the product of the Fourier transforms of each of the two functions. The inverse is true, as well: The Fourier transform of the product of two functions is equal to the convolution of the Fourier transforms of the two functions. Succinctly:

$$\mathscr{F}(f(x)\otimes g(x)) = \mathscr{F}(f(x))\mathscr{F}(g(x))$$

$$\mathscr{F}(f(x)g(x)) = \mathscr{F}(f(x))\otimes\mathscr{F}(g(x)).$$

Proof:

$$\mathscr{F}(f(x)\otimes g(x)) = \mathscr{F}\left(\int_{-\infty}^{\infty} f(x)g(u-x)\,dx\right)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-2\pi isu} f(x)g(u-x)\,dx\,du$$

Substituting $w = u-x,\ du = dw$ yields:

$$\begin{aligned}\mathscr{F}(f(x)\otimes g(x)) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-2\pi is(x+w)} f(x)g(w)\,dx\,dw \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-2\pi isx} f(x)e^{-2\pi isw} g(w)\,dx\,dw\end{aligned}$$

Note that the left part of the integrand is a function only of x and the right is a function only of w:

$$\begin{aligned}\mathscr{F}(f(x)\otimes g(x)) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-2\pi is(x+w)} f(x)g(w)\,dx\,dw \\ &= \int_{-\infty}^{\infty} e^{-2\pi isx} f(x)\,dx \int_{-\infty}^{\infty} e^{-2\pi isw} g(w)\,dw \\ &= \mathscr{F}(f(x))\mathscr{F}(g(x))\end{aligned}$$

The latter is just the products of the Fourier transforms of f and g, which is our proof of the first statement of the convolution theorem.

To prove the second, inverse, statement, we start with the result above. For convenience, we will define the following:

$$\begin{aligned}\mathbf{A} = \mathscr{F}(a),\ a = \mathscr{F}^{-1}(\mathbf{A}) \\ \mathbf{B} = \mathscr{F}(b),\ b = \mathscr{F}^{-1}(\mathbf{B}).\end{aligned}$$

Then:

$$\begin{aligned}\mathscr{F}\left[\mathscr{F}^{-1}(\mathbf{A})\otimes\mathscr{F}^{-1}(\mathbf{B})\right] &= \mathbf{AB} \\ \mathscr{F}^{-1}(\mathbf{A})\otimes\mathscr{F}^{-1}(\mathbf{B}) &= \mathscr{F}^{-1}(\mathbf{AB}) \\ \mathscr{F}(ab) &= \mathscr{F}(a)\otimes\mathscr{F}(a)\end{aligned}$$

Why is this cool?

Rapid Convolution

Except for simple cases, performing numerical convolutions is very slow, especially if both of the functions to be convolved contain many points. Computing the Fourier transform, however, is very rapid by comparison. Convolution is used often to smooth or to sharpen an image as suggested in the introduction. If we want to convolve a function (or an image) $x(n)$ by a smoothing function $s(n)$ (*i.e.*, to compute $s(n) * x(n)$) it is often better to compute the Fourier transforms of $s(n)$ and $x(n)$, take their product, and then compute the inverse transform. This yields the convolution.

Deconvolution

Suppose that we know $h(t)$, which we call the system impulse response (or perhaps the *hemodynamic response function*...) Computing its Fourier transform is a simple mechanical process (on a computer), which yields $H(s)$. Further, in a typical fMRI experiment, we *measure*

$$y(t) = x(t)\otimes h(t),$$

which we understand to be the convolution of the neural activity in response to a stimulus with the hemodynamic response function (see Villringer's chapter in "Functional MRI" (*1*) or Cohen's paper on Linear Systems Analysis in fMRI).

Calculating the Fourier transform of the observed response (e.g., on a computer):

$$\begin{aligned}\mathscr{F}[y(t)](s) &= \mathscr{F}[x(t)\otimes h(t)](s)\\ &= \mathscr{F}[x(t)](s)\mathscr{F}[h(t)](s)\\ &= \mathscr{F}[x(t)](s)H(s).\end{aligned}$$

We can rearrange this equation, dividing both sides by $H(s)$:

$$\frac{\mathscr{F}[x(t)\otimes h(t)](s)}{H(s)} = \frac{\mathscr{F}[y(t)](s)}{H(s)} = \mathscr{F}[x(t)](s)$$

$$\begin{aligned}\mathscr{F}^{-1}\left[\frac{\mathscr{F}[y(t)](s)}{H(s)}\right](t) &= \mathscr{F}^{-1}\left[\mathscr{F}[x(t)](s)\right](t)\\ &= x(t).\end{aligned}$$

The remarkable implication here is that, for example, from the observed BOLD imaging data, if we know the hemodynamic response function (known affectionately as the *hrf*), we can obtain at least the timing of the neural process in the brain.

Sadly, however, we do not know the *hrf* very accurately and, to make matters worse, the observed signal, $y(t)$, is not really $x(t) * h(t)$. Instead, there is always noise present, such that we observe:

$$y(t) = x(t) * h(t) + \sigma.$$

The inverse Fourier transform of $y(t)/H(s)$ usually is a bad approximation to $x(t)$. Furthermore, at points where $H(s)$ is zero, the left side of equation 3 becomes singular, and the problem is ill-posed.

The Fast Fourier Transform

Typically, we do not work with continuous functions these days, as those that have been described above. Instead, data are expressed as a series of discrete samples, for example the digital value of the NMR signal at a series of time points. Computing the Fourier transform of a set of samples is not done by solving some equation symbolically, as above, but instead is performed on digital computers.

Performing the Fourier transform in the most brute force way on a computer can be shown to take time that is approximately proportional to square of the number of support points that describe the input function. One of the most remarkable breakthroughs in computing was the work of Tukey, who showed that there was a very clever way to perform the transform that made it computable in a time proportional to $N\log(N)$, where N is the number of support points, an algorithm know as the *Fast Fourier Transform* or “FFT”. The difference between N^2 and $N\log(N)$ is huge for large data sets. No one has really worked out anything that is much faster, though with great effort, there have been some incremental advances.