3. FREQUENCY DOMAIN (FOURIER) ANALYSIS

3.1 Why Fourier Analysis?

Ultimately, what we have in a time series is a string of numbers corresponding to regular intervals. For most seasonal adjustment, that's all we have and all we use – dates and the number that matches each date. And yet somehow, we turn one string of numbers into three. We separate the stuff that repeats each year from the stuff that is evolving long term and the stuff that blips and fades in the short term.

The filters we use aren't terribly complicated or fancy, though there's a deal of maths and research underneath them. If we strip out the seasonal component and average what's left, that ought to remove the transient stuff and leave the long term behaviour. If we average the same time-point every year for several years, we'll focus on the seasonal component. (If you think about that further, you'll realise it has the very long term stuff in too, so the final process is a bit more complicated.) Intuitively, this ought to make sense to many people.

But if you want to understand why the filters work, and which one to use when, and (which is pretty important in practice) what their weaknesses are, then you need to delve deeper into mathematics. If you do not need to know these things, you can skip this chapter and the next. If you do, then we have found that the effects of filters can be conveniently summarized by their Gain Functions (and phase functions). But to interpret those gain functions, and to understand what can make filters give poor results, you must have at least a basic qualitative grasp of Fourier Transforms.

If you already are familiar with Fourier Transforms or Fourier Analysis, you can skip straight to the next chapter – or read section 3.4 on Fourier transforms of real series and then skip ahead.

3.1 Definition and properties of the Fourier Transform

3.1.1 What is Fourier Analysis?

The fundamental idea behind Fourier Analysis is that we can take any arbitrary signal – like our string of dates and numbers – and express exactly the same information as a bunch of sine and cosine waves[1], with different frequencies. Then we can consider this transformed data in terms of what's going on at the different frequencies, instead of what's going on at particular times.

Now think about how this applies to seasonal patterns, for monthly or quarterly time series. It doesn't matter what the seasonal pattern actually looks like – whether the series shows a

[1] This works because the set of all piecewise-continuous functions f(t) forms a Hilbert Space, and sine and cosine waves form an orthogonal set of vectors that span this space. We could (and some people do) use other orthogonal sets of "vectors" – such as Wavelets – the results can be called "generalized Fourier Transforms."

sharp spike each December, a slow rolling evolution with the seasons, or anything more complicated. We can still express that pattern in terms of just a few key frequencies – the ones that repeat every 12, 6, 4, 3, 2.4 and 2 months. (In other words, 1, 2, 3, 4, 5, or 6 times per year.) Pick out those frequencies, and the rest must be trend or irregular. And hey, if the trend is long term stuff, it should all be up one end of the frequency spectrum, and the short term irregular stuff should all (or at least mostly) be up the other end…

Then we can look at what real-world events are like, and how these pass through the filters we apply, and get quite a feel for how distortions echo through the system… if we first develop a feel for the relationship between a time series and its Fourier Transform.

Fourier Analysis is used across many disciplines and in many different forms. We're most interested in the Discrete, Finite Fourier Transform. We're most interested in the amplitude and phase view of the coefficients. But we will touch on the others.

Fundamentally it comes down to this: we can write any function $f(t)$ in terms of a sum of sine waves of varying amplitudes and phases, like so:

$$f(t) = \sum_i A_i \sin(w_i t + \phi_i)$$

Where A_i is the amplitude of the wave, and $\phi_i$is its phase. (See Section 3B. "Proving the transform" if the assertion that this works worries you.)

The set of amplitudes and phases is called the Fourier Transform of the series. There's more than one way to express or display this information – see Section 3.2.1 "Other forms of The Fourier Transform" if you want to think carefully about the others.

So let's get specific about the form we're actually going to use. We will label our time points 0, 1, 2 – the units might be months, quarters, or something else; it doesn't change the maths, so long as the points are evenly spaced.

Definition 1.1a

Let us take a Discrete, Finite time series $f(t) = y_0, y_1, \dots, y_{n-1}$. Then the operator F acts on $f(t)$ to produce the set of coefficients $C(\omega_i)$ and $S(\omega_i)$, defined at $\omega_j = \frac{2\pi j}{n}$ for j taking integer values in the range $0 \le \mathrm{j} \le \frac{n}{2}$ and the coefficients taking the values

$$C(\omega_j) = \sum_{t=0}^{n-1} y_t \cos(\omega_j t) \text{ and } S(\omega_j) = \sum_{t=0}^{n-1} y_t \sin(\omega_j t)$$

We may write this as $\mathsf{F}\left(f(t)\right) = C(\omega_j) + iS(\omega_j)$

We may also write this as $\mathsf{F}\left(f(t)\right) = A(\omega_j)\cos(\omega_j t + \Phi(\omega_j))$ where

$$A(\omega_j) = \sqrt{C^2(\omega_j) + S^2(\omega_j)} \text{ and } \Phi(\omega_j) = \arctan\left(\frac{S(\omega_j)}{C(\omega_j)}\right)$$

Well, mostly. For every possible value of $\tan(\theta)$ there are two possible values of θ that can give us that output. So arctan's output doesn't span 0-2□. To get $\Phi(\omega)$ really right, we have to check through all these corner cases:

If $C(\omega) > 0$ then $\Phi(\omega) = \arctan(\frac{S(\omega)}{C(\omega)})$

If $C(\omega) < 0$ and $S(\omega) \geq 0$ then $\Phi(\omega) = \arctan(\frac{S(\omega)}{C(\omega)}) + \pi$

If $C(\omega) < 0$ and $S(\omega) < 0$ then $\Phi(\omega) = \arctan(\frac{S(\omega)}{C(\omega)}) - \pi$

If $C(\omega) = 0$ and $S(\omega) > 0$ then $\Phi(\omega) = \frac{\pi}{2}$

If $C(\omega) = 0$ and $S(\omega) < 0$ then $\Phi(\omega) = -\frac{\pi}{2}$

If $C(\omega) = 0$ and $S(\omega) = 0$ then $\Phi(\omega)$ is undefined

Definition 1.1b

In this Discrete, Finite case, the inverse Fourier transform operator F^{-1} acts on the set of coefficients $C(\omega_j)$ and $S(\omega_j)$ to produce a function $g(t)$ according to:

$$g(t) = \frac{1}{n}C(\omega_0) + \frac{2}{n}\sum_{j=1}^{j\leq(n-1)/2}\left(C(\omega_j)\cos\omega_j t + S(\omega_j)\sin\omega_j t\right) + \frac{1}{n}C(\omega_{n/2})\cos\omega_{n/2}t$$

Where the last term is included only if n is even. (Otherwise $n/2$ isn't an integer.)[2]

Assertion 1.1

We get the same function back that we put in: $g(t) = f(t)$ In other words, F^{-1} actually is an inverse. For details, see appendix 3.B Proving the Transform.

3.1.2 Linearity

One of the really important things about the Fourier Transform is that it is a linear operator. If we add two series together, and then transform them, we get the same result as if we did the transform first, and then added the series up. If we multiply a series by a constant, the Fourier transform of that series comes out multiplied by that same constant. We know this works because it works at every step of the process of computing a Fourier transform. It works when we multiply each function by the same $\cos\omega_j t$; it works when we add up all the

[2] There are several ways to think about why you need different coefficients for C_0 and $C_{n/2}$. One of them is that all the 'energy' at these frequencies must be concentrated in the cosine wave; the sine wave is, at these frequencies only, identically zero. Another is to consider what happens when you try and pick out these frequencies using the inner product; part of the equation is

$\cos(\omega_j t)\cos(\omega_k t) = \frac{1}{2}\left(\cos(\omega_j - \omega_k)t + \cos(\omega_j + \omega_k)t\right)$ and we add this up over a full cycle of 2□ radians; normally only the first term survives that, when $\omega_j = \omega_k$; but at these special frequencies the second term is also always 1.

results across all the points in the function; it works when we multiply by our normalization constant.

In equation form:

If $\mathsf{F}(f(t)) = C(\omega_j) + iS(\omega_j)$ does $\mathsf{F}(af(t)) = aC(\omega_j) + iaS(\omega_j) = a\mathsf{F}(f(t))$? Well,

$$C'(\omega_j) = \sum_{t=0}^{n-1} ay_t \cos(\omega_j t)$$
$$= a\sum_{t=0}^{n-1} y_t \cos(\omega_j t)$$
$$= aC(\omega_j)$$

And similarly for $S(\omega)$.

Similarly $\mathsf{F}(f(t) + g(t)) = \mathsf{F}(f(t)) + \mathsf{F}(g(t))$

Key result:

You can add or subtract functions, or multiply by constants, and when you take the Fourier Transform you get exactly the corresponding sum or multiple.

$$\mathsf{F}\left(af(t) + bg(t)\right) = a\mathsf{F}\left(f(t)\right) + b\mathsf{F}\left(g(t)\right)$$

In other words, the Fourier Transform is a Linear Operator.

3.1.3 Symmetry

There is a lot of symmetry between the Fourier Transform and its Inverse. They're practically the same operation – whichever way you're going, you're multiplying by a wave and integrating (or summing) over some span.

So if you transform a function A to get some output B, then if you were to start with B and transform it, you ought to get A. Making this maths work precisely in our usual notation may involve such things as assuming that A has an implicit phase of 0 at all frequencies, and assuming that B is twice as long as the coefficients we usually bother calculating, with the second half a mirror image of the first.

In the complex exponential notation the symmetry is glaringly obvious. See Section 3.2.1, subsection Complex Exponentials.

Key result:

If $\mathsf{F}(f(t)) = g(\omega)$ Then $\mathsf{F}(g(t)) = f*(\omega)$ where $*$ denotes the complex conjugate. (triple-checking required)

In other words, the Fourier Transform is symmetric with its inverse.

3.1.4 Frequencies vs. periods:

Roughly speaking, a frequency measures how many times a given cycle goes through in a unit of time. Throughout this course, we're going to assume one cycle corresponds to 2□ radians; we're going to measure our frequency □ in radians per unit time (angular frequency), not cycles per unit time (oscillation frequency).

Imagine something that happens every p units of time. (*p* does not need to be an integer.) We call *p* the period, and we relate it to our frequency □ by: $\omega = \frac{2\pi}{p}$ or equivalently

$$p = \frac{2\pi}{\omega}$$

So if our units of time are months, and we're looking at something that happens once every year, then *p*=12 months, and $\omega = \frac{2\pi}{p} = \frac{\pi}{6}$. If it happens twice a year, $\omega = \frac{\pi}{3}$ Three times a year, $\omega = \frac{\pi}{2}$, etc.

If our units of time are quarters, then one year corresponds to $\omega = \frac{\pi}{2}$, and the only other 'seasonal' frequency we can reasonably measure is "twice a year" (2 quarters) for which $\omega = \pi$.

In Discrete Fourier Analysis, $\omega = \pi$ is always going to be the fastest frequency we can observe. That's something that is up one measurement and down the next. In monthly or quarterly time series, this is going to be seasonal (two months or two quarters per cycle, precisely) – but nearby frequencies will be irregular.

$\omega = 0$ is always infinite period – the constant term – at □ near and including 0, we find the trend.

Some people like to work with negative frequencies. Apply the transformations $\sin(-x) = -\sin x$ and $\cos(-x) = \cos x$ and there should be no problem here… a negative frequency doesn't mean a negative period, it just means the phase changes differently as time runs forward. In Discrete situations, a frequency –□ is precisely equivalent to a frequency of 2□-□– see Section 3.6.1 "Aliasing" for more details.

3.2 Technical Explorations (Optional)

3.2.1 Other forms of the Fourier Transform

3.1 Sine and Cosine coefficients

	Transform	*Inverse Transform*
Continuous Infinite Transform (Function to function)	$C(\omega)=\int_{-\infty}^{\infty} f(t)\cos(\omega t)dt$ $S(\omega)=\int_{-\infty}^{\infty} f(t)\sin(\omega t)dt$ $0\le\omega$	$f(t)=\int_{\omega=0}^{\infty} C(\omega)\cos(\omega t)+S(\omega)\sin(\omega t)d\omega$ Note: Normalization constants have not been checked Inverse may be only roughly right.
Continuous Finite Transform (A periodic (perfectly repeating) wave)	$C(0)=\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$ $C(n)=\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx$ $S(n)=\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx$	$f(x)=\frac{1}{2}C(0)+\sum_{n=1}^{\infty}C(n)\cos(nx)+\sum_{n=1}^{\infty}S(n)\sin(nx)$ http://mathworld.wolfram.com/FourierSeries.html Assumes f(x) is defined on $\left[-\pi,\pi\right]$ (which is a bit different to our other cases.) This is a Fourier Series.
Discrete Infinite Transform (Dots from here to eternity)	$C(\omega)=\sum_{t=-\infty}^{\infty} y_t\cos(\omega t)$ $S(\omega)=\sum_{t=-\infty}^{\infty} y_t\sin(\omega t)$ $0\le\omega\le\pi$	$f(t)=\int_{\omega=0}^{\pi} C(\omega)\cos(\omega t)+S(\omega)\sin(\omega t)d\omega$ Assumes t takes integer values Note: Normalization constants have not been checked Inverse may be only roughly right.
Discrete Finite Transform (dot to dot)	$C(\omega_j)=\sum_{t=0}^{n-1} y_t\cos(\omega_j t)$ $S(\omega_j)=\sum_{t=0}^{n-1} y_t\sin(\omega_j t)$	$y_t=\frac{1}{n}C(\omega_0)+\frac{2}{n}\sum_{j=1}^{j\le(n-1)/2}\left(C(\omega_j)\cos\omega_j t+S(\omega_j)\sin\omega_j t\right)+\frac{1}{n}C(\omega_{n/2})\cos\omega_{n/2}t$ As given earlier in the chapter. Assumes t takes integer values between 0 and n-1

There's more than one way to do it and more than one way to view it.

Those normalization constants – different sources put those constants in different places. They can be all in the transform, all in the inverse, or shared between them. For the purposes of our calculations, we would like our Fourier Transforms to give us nice gain functions, which we have when the normalization is chosen so that when we simply add up (or integrate over) the input function (f(t) or y(t)) we get C(0).

Amplitude and phase coefficients

One of the classic results of Trigonometry is that if you have two waves of the same frequency, and you add them together, you get another wave at the same frequency, but with possibly different amplitude and phase. In the Fourier case this is particularly simple; where we have a sine wave and a co-sine wave, we could instead be writing

$$C(\omega)\cos\omega t + S(\omega)\sin\omega t = A(\omega)\cos\left(\omega t + \Phi(\omega)\right)$$

Where:

$$A^2(\omega) = C^2(\omega) + S^2(\omega)$$

$$\tan\left(\Phi(\omega)\right) = \frac{S(\omega)}{C(\omega)}$$

This is a really useful view of a Fourier Transform. It lets us think about how much strength is present at a given frequency, and worry about the phase later if at all – sometimes the phase isn't important; sometimes it's really important. Many of the graphs in this chapter and the next show amplitude only.

Complex exponentials:

The maths for Fourier series and Fourier Transforms simplifies dramatically if we work not with sin(□t) and cos(□t) but with $e^{i\omega t}$ and $e^{-i\omega t}$ - remember that $e^{i\omega t} = \cos\omega t + i\sin\omega t$; the equivalence is pretty straightforward. $F(\omega) = C(\omega) + iS(\omega)$ and $F(-\omega) = C(\omega) - iS(\omega)$

This naturally implies the use of negative frequencies.

To take the continuous infinite case as our example:

$$\mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\,dt$$

$$\mathcal{F}^{-1}(F(\omega)) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}\,d\omega$$

(If we used as our variable $\nu = \frac{\omega}{2\pi}$ the normalization coefficient would evaporate, leaving gorgeous symmetry behind. See for example http://mathworld.wolfram.com/FourierTransform.html)

The maths of convolution shows at its best when using this notation.

Although we can work with Fourier Transforms purely in real numbers, everything behaves perfectly nicely if you put complex numbers in and allow yourself to get complex coefficients out.

Constraints:

1) The Discrete Transforms assume that the input data points are evenly spaced. Push that assumption a little, and we find ourselves making "Length of Month" and "Trading Day" corrections. If you try to generalize any further, you'd better know exactly what you're doing, or you'll introduce massive noise etc.

2) The Continuous transforms only work for piece-wise continuous functions. It's quite possible to come up with fuzzy nasty functions for which the Fourier Transform breaks down. For example, we can't meaningfully transform a function i(x) such that i(x)=1 if x is a rational number and i(x)=0 if x is irrational. But that's *not* piecewise continuous.

3) The Infinite transforms, if they're to stick to finite values, require that the input function tail off to zero, 'fast enough' – it will behave if $\lim_{T\to\infty}\int_0^T |f(t)|\,dt$ converges to something finite (and the same on the negative side; this is of course the criterion for the continuous case. I haven't checked this completely; a weaker condition might do.) Otherwise we can end up with infinities in the transformed domain … but if those infinities are Dirac Delta functions, a peculiarly well-behaved infinity also called the Impulse function, then the mathematics still holds together.

4) Generally, Fourier Transforms only work for functions that confine themselves to finite values. The Dirac Delta function is a very special exception.

The Continuous Fourier Transform can cope with a few discontinuities. For example, we can replicate a step function. If we are using approximations instead of a perfect Fourier Transform, there are imperfections – see "The Gibbs Phenomenon" in Section 3.3.1.

3.2.2 How the different Fourier Transforms relate to one another – Continuous and Discrete, Finite and Infinite.

This discussion is largely qualitative. It doesn't have a lot to do with how Fourier Transforms help us understand Gain Functions which help us understand (and pick and choose between) filters. Personally I think it's just beautiful mathematics, and it's really nice to see how the different possibilities are consistent with one another. But to do this, we need to work with limits, and with infinities. I hope your calculus is up to date …

Remember limits? For a continuous function,

$$\lim_{x\to a} f(x) = f(a)$$

But for calculus we used:

$$\frac{df(x)}{dx} = \lim_{\Delta x\to 0}\frac{f(x+\Delta x)-f(x)}{\Delta x}$$

Which is often a very nicely behaved limit, once you get over the fact that you're dividing zero by zero.

Now, remember infinity? If you multiply infinity by 2, you still have exactly the same infinity, right? Well, in the world of "Generalized functions" there are exceptions to that, and one of these is the Dirac Delta function. It's zero everywhere but at one point – nevertheless, if you integrate over that point, you get 1. And if we double the function, the integral doubles

– in other words, it's an infinity that does behave itself. (It is also sometimes described as an 'impulse' function.)

We can get it using limits. For example:

$$\delta(x) = \lim_{a \to 0} \frac{1}{a\sqrt{\pi}} e^{-x^2/a^2}$$

This approaches zero everywhere except at x=0 – where it approaches infinity. If you integrate over the whole line, you always get 1 exactly, and that still holds in the limit. Apply a constant – and it scales perfectly; you can integrate over it and get that constant back. It's kind of like the transition between the Reals and the Complex numbers – it has much in common with i.

Just to go through some of the places Dirac Delta turns up:

If you apply the Continuous Infinite Fourier Transform (CIFT) to sin(□t) (or cos □t) you get a Dirac Delta function at □. (Phase differs, of course.)

If you apply the CIFT to a Dirac Delta function at t=a, you get a perfectly flat Fourier Transform – all frequencies are present at equal strength. (If that sounds like white noise to you, you'd be right – but with white noise, the phase is random; with a Delta-function, the phase follows a specific pattern depending on a – that is, when the blip actually occurs.) (See graph 3.6, later.)

If you have a regularly spaced infinite sequence of Dirac Delta Functions (a 'comb'), and you transform it, you get back another regularly spaced infinite sequence of Dirac Delta Functions – only the spacing is different; the narrower the spacing on the original deltas, the wider it is on the transformed deltas, in a classic T -> 1/T relationship (classic for Fourier Transforms, anyway; any time you play with the width of the input you get a 1/T response on the width of the output.) See graph 3.5, later.

If you do a CIFT on any perfectly repeating, continuous wave, you get back a string of Dirac Delta functions with varying coefficients. The blips appear at exactly those frequencies we'd use for a Continuous Finite transform on one repeat of the wave, and the coefficients on the Dirac Delta functions are exactly the coefficients we'd get out of the Continuous Finite transform.

Now let's see how the Continuous Infinite Fourier Transform contains within it the Discrete Infinite Fourier Transform. For our input function, we're going to take the discrete data, and multiply it (scalar style) by the Dirac Delta. So where we had f(i) defined at integer values, the new function is

$f(t) = \delta(0) f(i)$ where $t = i$

$f(t) = 0$ elsewhere

(It's a comb function where the teeth have heights defined by f(i).)

Because we started with a comb function, the output is going to be a wave with a repeat inversely proportional to the space between the tines. Each tine $\delta(0)f(i)$ adds in a sine or cosine wave with amplitude f(i) – just exactly as the Discrete transform does.

To set up Discrete Finite Fourier Transform, we proceed as for the Discrete Infinite version, but we repeat f(i) infinitely many times. So, if there are n values, then:

$f(t)=\delta(0)f(i)$ where $t \equiv i \bmod n$

$f(t)=0$ elsewhere

That gives us output that repeats infinitely many times – as before – but also that is a comb function itself, because it comes from a repeating function.

So – that should illustrate how all the other Fourier Transforms can be done as special cases of the Continuous Infinite one. What about the other way?

Take your Finite Fourier Transform, and consider doing it over longer and longer spans. If your input signal is a pure sine wave, then the coefficient you accrue for it over your span gets larger and larger (as does your normalization coefficient for the return transform – it looks suspiciously like the 'd□' you're going to need …)– and in the limit of infinite span, becomes a Dirac Delta function. (If your input signal tails off at large t, then your Infinite Fourier transform is the limit that your finite Fourier transforms converge to at arbitrarily large T; the finite transforms settle down to look like the infinite one.)

Take your Discrete Fourier Transform, and consider doing it over finer and finer intervals. (We're going to have to work with intervals of time □t, unlike the □=1 assumed throughout most of this text; there is a corresponding normalization coefficient.) If the input function is changing appreciably between observations, that indicates very high frequencies are present – which we won't be able to resolve until we can 'see' high enough frequencies and aliasing ceases to be a problem. For well-behaved (i.e. continuous) functions we can probably show that the Fourier transform coefficients at low frequencies converge to well-defined values as the interval drops to zero.

3.2.3 Information Theory, Decomposing Time Series, and Discrete Finite Fourier Transforms

In the original Time Series, every number has something all its own. We cannot be certain of *any* number in the series unless we have that number itself – even if we have 119 numbers from ten years of collecting monthly data, we do not know the missing one. It has something all its own – it's irregular value, if nothing else. We might be able to guess roughly where it would be – but not exactly.

(When we decompose a time series, we shed a lot of that. The three strings of numbers we end up with, each contain less information than the original. Knowing what filters have been applied to obtain each series, we should be able to obtain some internal values in each series from the others.)

When we do a discrete finite Fourier transform, we take N numbers and turn them into amplitude and phase coefficients for N/2 +1 frequencies. Does this really always work? Isn't there too much information?

One of the frequencies is always 0 – constant, not oscillating. Sin 0t is always 0, so that coefficient contains no information. So if N is odd, we have 1 coefficient C(0) and 2 sets of (N-1)/2 coefficients, C(□)and S(□) for $\omega_j = \frac{2\pi j}{n}$ for j less than n/2. That gives us exactly the same number of numbers as we put in.

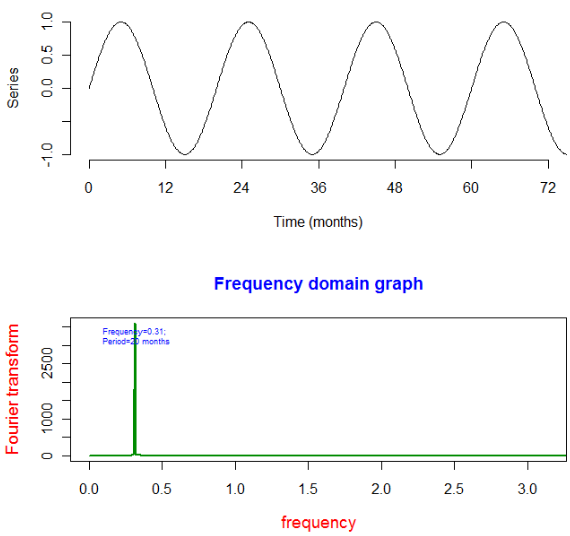
Importantly, if N is even, then $\sin\left(2\pi \frac{N}{2} t\right)$ is also always 0, so S(□) has no information and C(□) has one number worth of information. (In the amplitude and phase view, the phase term is always □/2 for □=□) Again, we get out exactly as much information as we put in.

3.3 Analytical Fourier Transforms

3.3.1 Simple examples

If we're trying to write a function in terms of the sine waves that make it up, then if the function is only made from one or two sine waves, the situation should be really simple. Let's look at: y=sin □x/10

Figure 3.1 y=sin □x/10



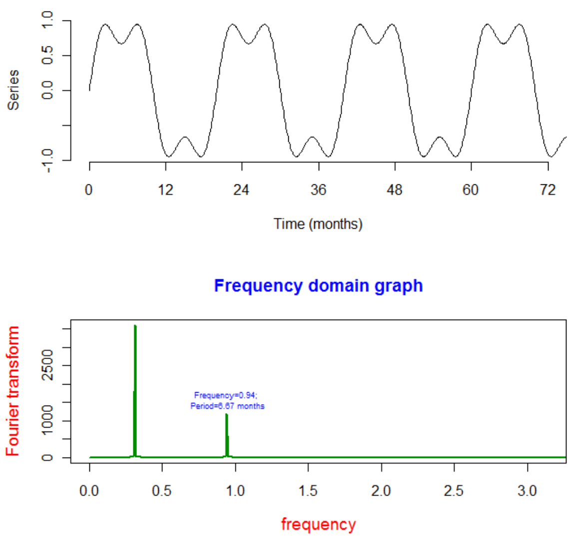
There's nothing in the signal but a single frequency – so of course the Fourier Transform is nothing but a single spike.

If we were to reduce the frequency (make the input signal take a longer time T to go through a cycle) then the spike would move closer to the □ axis according to 1/T; this inverse scaling behaviour (you get narrower if I get wider) is classic for all input signals, pure waves or not.

> Exercise:
>
> If we put in a cosine wave instead of a sine wave, A(□) spikes by the same amount at the same frequency. So what is different?

Now let’s look at a close relative. We’ll modify the above input signal by adding a second, higher frequency, at a lower strength.

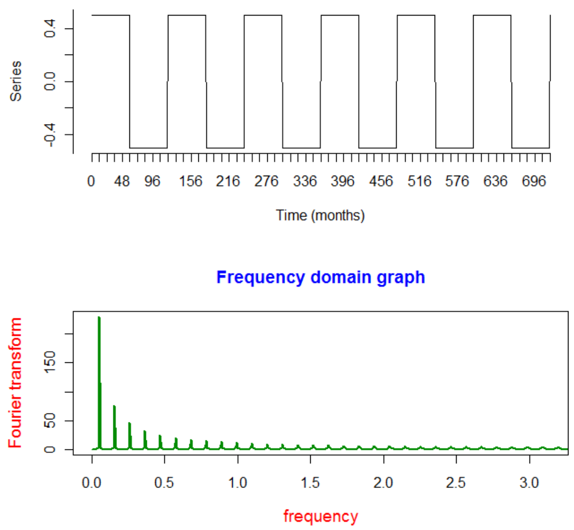
Figure 3.2 y = sin □x/10 + 1/3 sin 3□x/10



Notice that the higher frequencies (shorter periods) create far more abrupt transitions. We’ve got wild swings in the level – and then it abruptly settles down a bit; in other words we have much higher curvature etc. This kind of dramatic shift in level, rapid change in direction, can loosely be described as a ‘fast’ movement.

Okay, let’s take one of the fastest movements there is – a square wave!

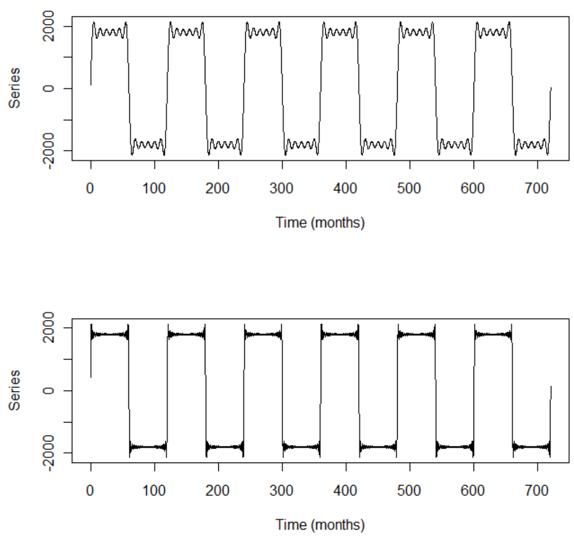
Figure 3.3 A square wave



This is a close relative of the above series – and its Fourier transform shows strength at arbitrarily high frequencies. For a discrete finite transformation (this one is only based on 720 points) we can't observe the high frequencies; but for a continuous transform, the infinitesimal signals at arbitrarily high frequencies are still significant.

Aside: Gibbs Phenomenon

Figure 3.4 Gibbs Phenomenon



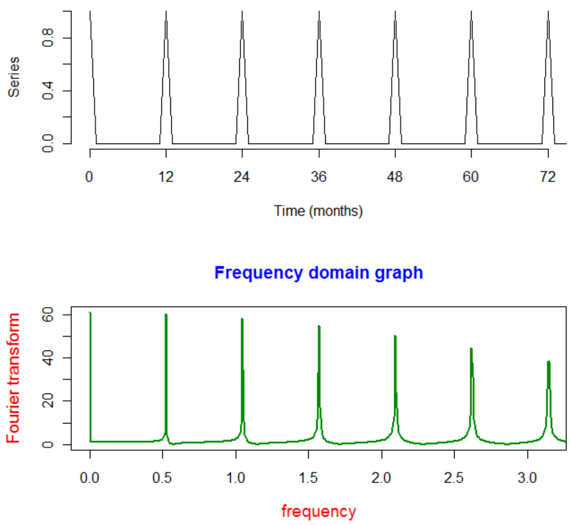
We can do a Continuous Fourier Transform on any piece-wise continuous function. But if there are discontinuities in the function, then the Fourier Transform will have infinitesimal signals at arbitrarily high frequencies.

If we chop off the Fourier transform above a given frequency, and apply the inverse Fourier Transform to take that back to the original ('time') domain, we get back an approximation to the original function. For original functions which are continuous, these approximations can be very good indeed.

For original functions which have discontinuities, the approximations are good everywhere except right close to the discontinuities. How close depends on how high the frequency cut-off is; two such approximations to the square wave are shown. The 'ears' or Ringing artefacts get narrower as the frequency cut-off goes up – but the height never shrinks beyond a certain limit – about 9% of the discontinuous jump.

More examples of Fourier Transforms

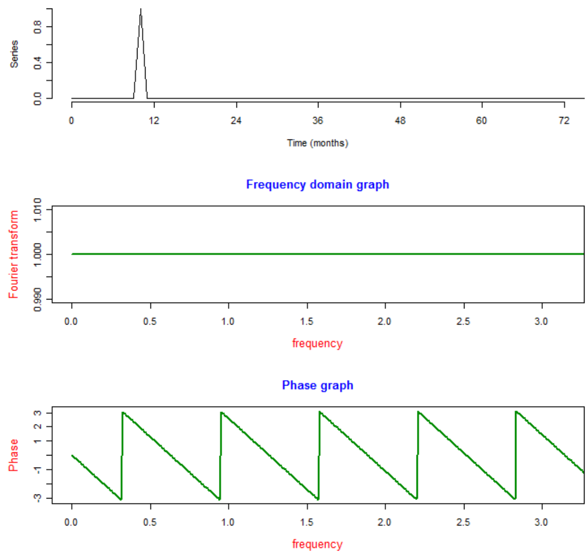
Figure 3.5 Series with regular spike



And yet we can get faster still. What if the series is only momentarily high? A regular blip … well, in this case the high frequency components are a great deal stronger relative to the low frequency ones. This kind of series can loosely be described as a comb.

Suppose we took instead a single blip (also described as an 'impulse function' or a Delta function – in this example a Kronecker Delta.) When you consider the symmetry between the Fourier Transform and its inverse, you might predict that the Fourier Transform of an impulse would be a sine wave. But – phase matters. So far we've only been plotting the strength of the signal – the coefficient A(□), which is $\sqrt{C(\omega)^2 + S(\omega)^2}$ - now let's include $\Phi(\omega)$, the phase information.

Figure 3.6 Wave with phase information



You see that A(□) is flat to the limits of the computer's ability to calculate it. This is in fact characteristic of an impulse function – their spectrum is 'white'; they have the same signal strength at all frequencies. Whatever strength doesn't go into S(□) does go into C(□) (because $\sin^2 x + \cos^2 x = 1$).

Exercise:

Compute the Fourier Transform for a blip appearing at t=0, or a blip appearing at t=20. How would you describe the changes to the phase function?

3.3.2 Some Analytical Fourier Transforms

Continuous Infinite Fourier Transforms,

From http://mathworld.wolfram.com/FourierTransform.html

(Uses e-iwt notation)

Figure 3.7 A selection of analytical Fourier Transforms

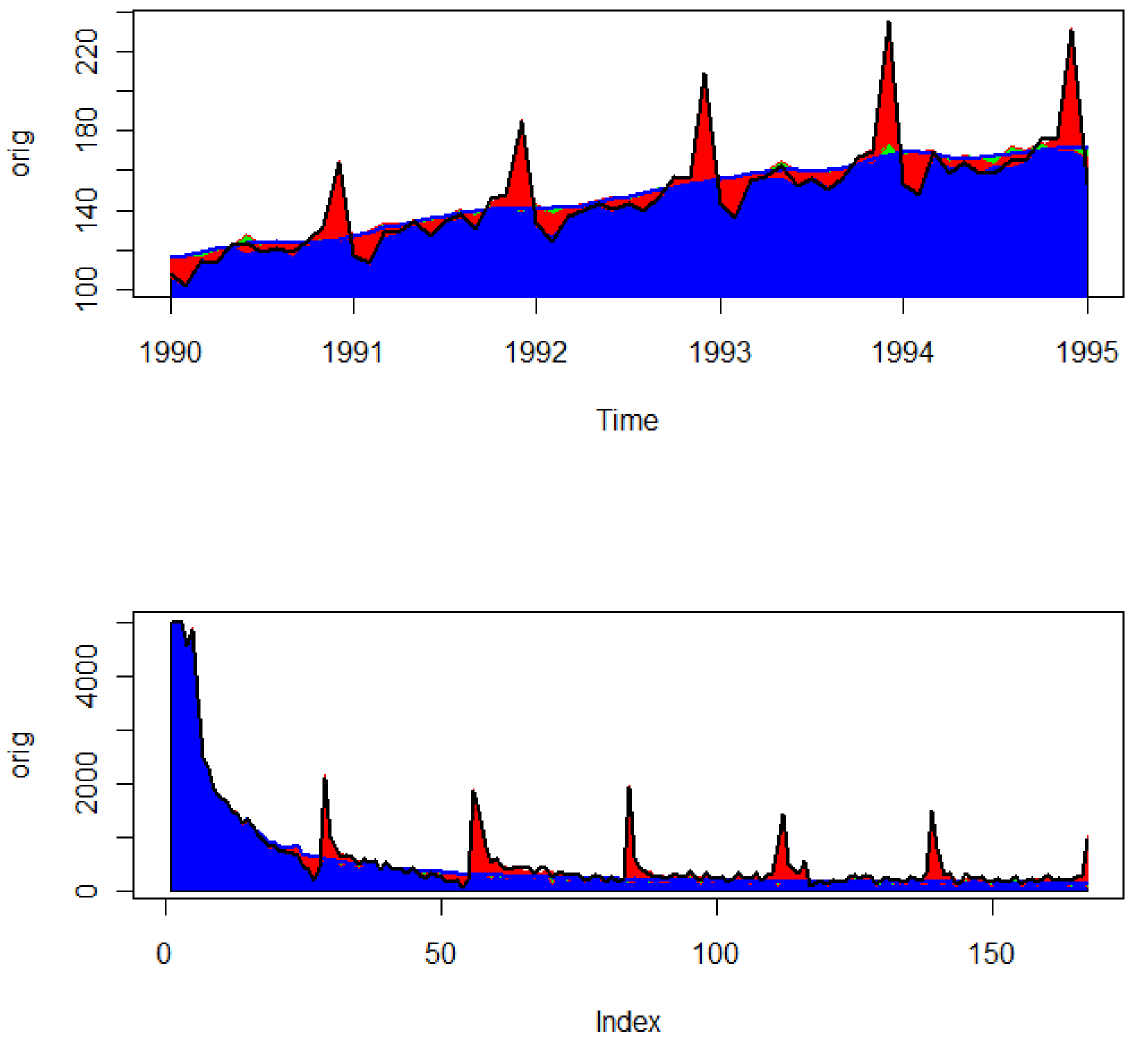
function	$f(x)$	$F(k) = \mathcal{F}_x[f(x)](k)$
Fourier transform--1	1	$\delta(k)$
Fourier transform--cosine	$\cos(2\pi k_0 x)$	$\frac{1}{2}[\delta(k-k_0)+\delta(k+k_0)]$
Fourier transform--delta function	$\delta(x-x_0)$	$e^{-2\pi i k x_0}$
Fourier transform--exponential function	$e^{-2\pi k_0 \lvert x \rvert}$	$\frac{1}{\pi}\frac{k_0}{k^2+k_0^2}$

Fourier transform--Gaussian	$e^{-a x^2}$	$\sqrt{\frac{\pi}{a}}\ e^{-\pi^2 k^2/a}$
Fourier transform--Heaviside step function	$H(x)$	$\frac{1}{2}\left[\delta(k) - \frac{i}{\pi k}\right]$
Fourier transform--inverse function	$-PV\ \frac{1}{\pi x}$	$i\,[1 - 2\,H(-k)]$
Fourier transform--Lorentzian function	$\frac{1}{\pi}\ \frac{\frac{1}{2}\Gamma}{(x-x_0)^2+\left(\frac{1}{2}\Gamma\right)^2}$	$e^{-2\pi i k x_0 - \Gamma\pi \lvert k\rvert}$
Fourier transform--ramp function	$R(x)$	$\pi\, i\, \delta'(2\pi k) - \frac{1}{4\pi^2 k^2}$
Fourier transform--sine	$\sin(2\pi k_0 x)$	$\frac{1}{2}\, i\, [\delta(k+k_0) - \delta(k-k_0)]$

3.4 Fourier transforms of real time series

Of course a real time series has a far more complex Fourier transform. There's a lot more going on. Let's have a look at Retail sales.

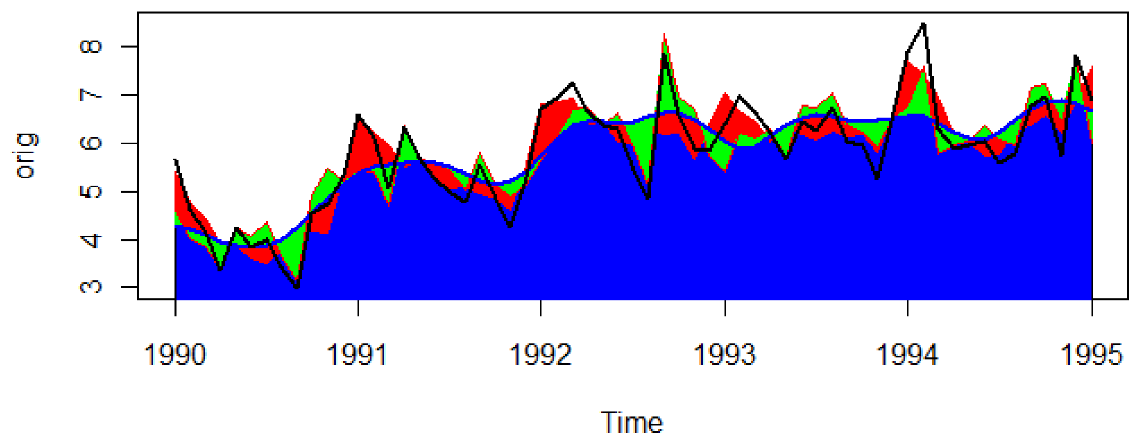
Figure 3.8 ACT Retail Sales, Original (upper panel and Fourier Transform (lower panel). Red shows the seasonal variation about the trend (blue) – the irregular variation (green) is barely visible.) For the Fourier Transform, the vertical scale is truncated; at □=0 the Fourier Transform peaks above 60,000. The Fourier Transform has been computed on Retail Trade from 1982 to 2009; and the vertical scale is essentially arbitrary.

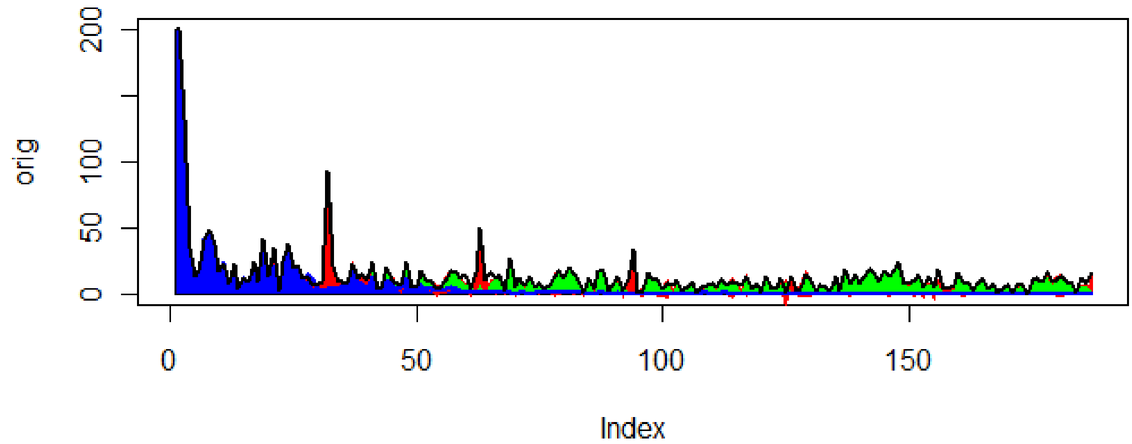


This figure shows ACT retail trade (total across all the Retail sector.) The trend (blue) is extremely smooth, showing steady long-term growth; as a result, its Fourier Transform is massively concentrated at the very lowest frequencies. The seasonal variation (red) is concentrated in spikes once a year; so the Fourier Transform is strong at all the frequencies that fit into 12 months. (In this example, trading day variation is included in the red component; it does not fit the 12 month frequencies so well.) There is minimal irregular (green).

Now let's look at a noisier example.

Figure 3.9 Male unemployment rate in the ACT. The peak of the trend at w=0 is not shown.

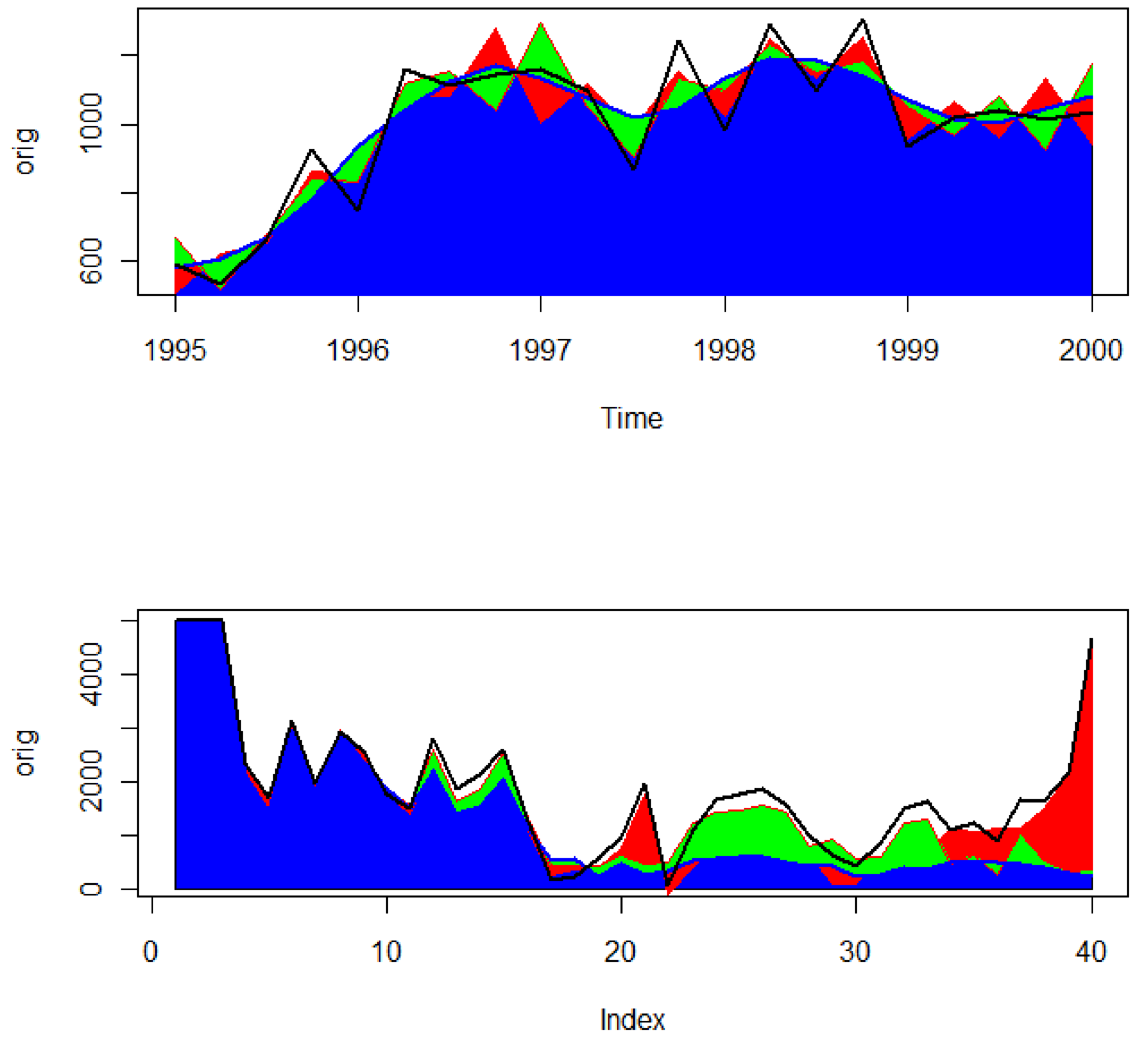




For Male unemployment in the ACT, we see the trend is not nearly so steady – but it is still concentrated near □=0. The seasonal pattern is for highs in late summer, autumn, and a bit lower the rest of the year – so we see plenty of strength around period 12, but little or none at periods 2, 2.4 and 3. This series has plenty of irregular (green). There is no trading day component.

Now let's have a look at a quarterly series.

Figure 3.10 Capital Expenditure on Buildings, NSW. Like most quarterly series, we don't have that many data points available in the first place.



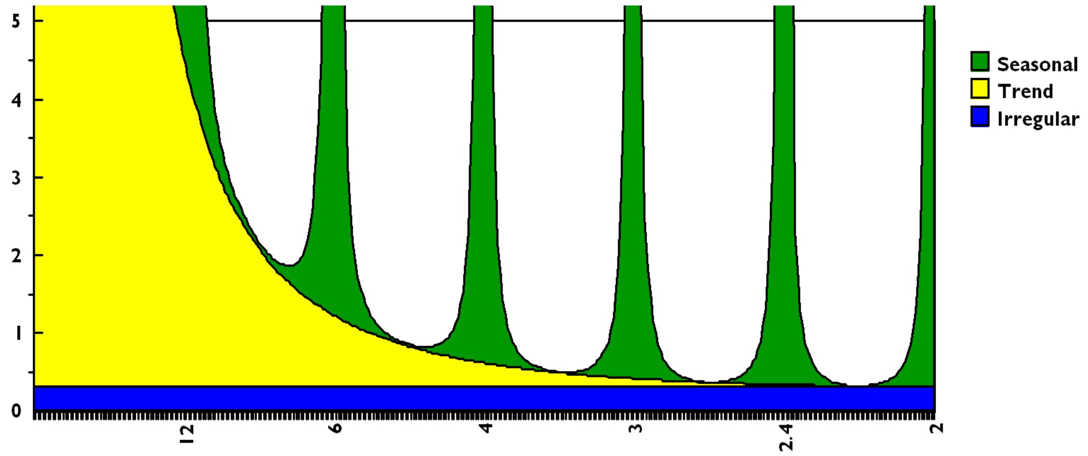
Capital Expenditure on Buildings, in NSW – has a fairly consistent seasonal pattern (though it evolves over the history of the series, leading to broader peaks.) The trend shows plenty of evolution in its own right – people might call that the Business Cycle – and its Fourier Transform extends to 'higher' frequencies. Seasonal frequencies are only really strong at period = 4 (that's once a year!) and period = 2.

These are empirical Fourier Transforms. We see hints that other frequencies are present in the seasonal component, when theoretically there shouldn't be any. Some of this may be due to the imperfections in our filters – in the next chapter we can show you what those imperfections are. And some of this may be due to weaknesses in the Discrete Finite Fourier Transform.

I've picked some of the cleanest examples, but the standard transform has weaknesses (of resolution, aliasing, and sampling) that make the above transforms less useful for day-to-day analysis.

So what should an ideal, real fourier transform be like?

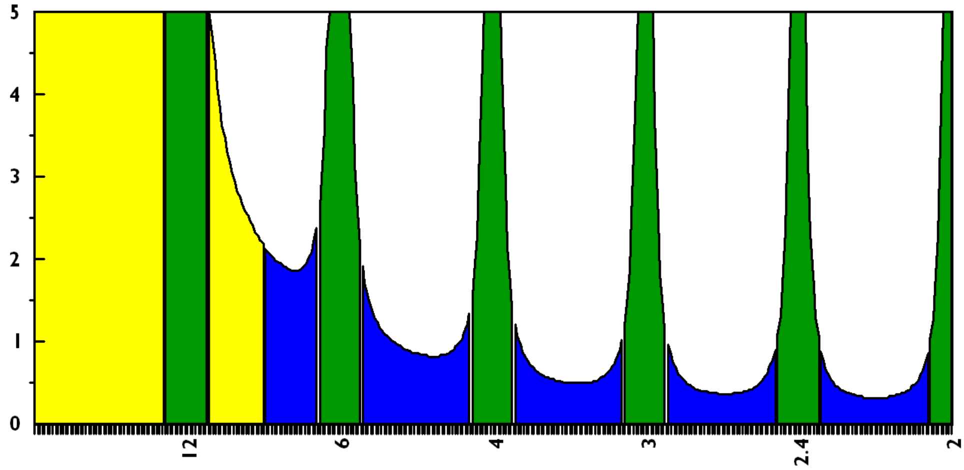
Figure 3.11 Theoretical Fourier Transform of a real series – model-based, monthly series



If the Irregular is genuine white noise, it ought to be present in equal amounts at all frequencies (blue). The seasonal component (green) ought to be present chiefly at the seasonal frequencies (12 months, 6, 4, 3, 2.4 and 2 months) but if the seasonal pattern evolves then these spikes get a little wider. The trend – in this view, 'everything else' – is present mainly at low frequencies.

If we were doing a model-based decomposition, fine-tuned for the individual series, we might actually manage to decompose the time series like that. But we have to be more practical than that – don't have time to fine tune 2800 series individually. If we simply cut up the Fourier transform by frequency, we'd get something that was at least roughly right, and it would look like this:

Figure 3.12 Theoretical Fourier transform of a real series, frequency-based apportionment between trend (yellow), seasonal (green) and irregular (blue)



If you look at what's happening at period 6 months, for example, you'll see that this isn't going to give the same results. If the first model is correct, then we've labelled as seasonal a bit of signal that really ought to be trend, and a bit of irregular as well. At period=3 months, we've labelled as seasonal some stuff that ought to be irregular, and we've missed the edges of the seasonal peak (our computed seasonal pattern isn't evolving as it should) and classed stuff as irregular instead.

The reality of the X11 filter-based decomposition is, of course, somewhere between these ideals. We **do** only separate the components based on frequency, but the transition from 'in' to 'out' is gradual, with the transition depending on the filter.

With an X11 decomposition there is still some mixing between irregular and seasonal, irregular and trend – and a little between trend and seasonal. We apportion what's present at each frequency to the different components in some ratio, which you should see in the filters' gain functions. See next chapter. The "true" seasonal component may have a different phase to the "true" irregular component – they're mixed, that's tough, and frankly we might not have managed to preserve the phase perfectly anyway.

If all that sounds awful – well, how bad it is really depends on the series. Most series have a much stronger seasonal component than they do noise; they're really pretty easy to separate and you'll get much the same results whichever way you do it. If there's a lot of noise, but the seasonality is very stable, it's still possible to do a decent (fairly robust) job of identifying the seasonal component. If the series is swamped with noise then it really doesn't matter what you try.[3]

3.5 Convolutions and Gain Functions

Why have we been going into this? We're going to analyse our filters in terms of their Fourier Transforms. Let's analyse why this works.

[3] Garbage In, Garbage Out.

3.5.1 Convolution

When we multiply one function by another function, the relationship between the Fourier Transforms of the parts and that of the product is fairly complex.

$$\mathsf{F}\left(f(t)g(t)\right) = \mathsf{F}\left(f(t)\right) \otimes \mathsf{F}\left(g(t)\right)$$

and, thanks to symmetry in the Fourier Transform,

$$\mathsf{F}\left(f(t) \otimes g(t)\right) = \mathsf{F}\left(f(t)\right)\mathsf{F}\left(g(t)\right)$$

Where $\otimes$ denotes the Convolution operation.

In the continuous case, the convolution is defined as:

$$\left[f \otimes g\right](t) = \int_{-\infty}^{\infty} f(x)g(t-x)dx$$

In the discrete case, it is defined as:

$$\left[f \otimes g\right](t) = \sum_{-\infty}^{\infty} f(x)g(t-x)$$

This is a lot like simply multiplying f(t) by g(t). But – it's multiplying f(t) by g(t) at every possible offset □ (that's a linear operation), integrating (adding up, in the discrete case) the results from top to tail (also a linear operation), and keeping *all* those values as the new function.

You might be thinking about Correlation measures; there are commonalities. You might reasonably think the convolution is measuring how much these functions 'overlap' at certain offsets. (See http://mathworld.wolfram.com/Convolution.html for some lovely animations.) But – see how the dummy variable x has different signs for f(x) and g(t-x)? g(t) has been 'flipped'; the other common name for the convolution operation is 'folding'.

Like multiplication, convolution is commutative (it doesn't matter which comes first out of f & g). It is linear.

Here's the wonderful thing. When we apply a filter w(i) to a signal y(t), that is mathematically equivalent to a convolution operation. If the filter has nonzero values from w(-m) to w(n) then the result is:

$$z(t) = \sum_{k=-m}^{n} y(t+k)w(k)$$

So define f(t) = w(t), noting that w(t)=0 for t outside –m to n; define g(t)=y(-t), and we get $f \otimes g(t) = z(-t)$. Doublecheck those equivalences and how FT reacts to flipping)

Now look at the Fourier Transforms …

$$\mathsf{F}\left(y(t) \otimes w(t)\right) = \mathsf{F}\left(y(t)\right)\mathsf{F}\left(w(t)\right)$$

What this means is that we can understand what a filter does to an input signal – any input signal! – just by looking at the Fourier transform of the filter. We call it the Gain Function (or Gain & Phase function). This is a Discrete Infinite Fourier Transform (infinite supply of zeroes on either side of the filter …) and it shows how the filter affects frequencies from 0 to □ (assuming our classic □t=1)

This one single result is almost the only reason we include Fourier Analysis in this course. Gain functions are hugely informative when it comes to deciding which filter to use in which situation, and how the results from a filter are affected in the presence of freak values, changes in the real world, or simply by noisy data … but you can't interpret a gain function if you don't have at least a qualitative feel for Fourier transforms.

3.5.2 Gain functions, brief:

For a filter □m to □n we work out the coefficients C(□) and S(□) according to:

$$C(\omega) = \sum_{i=-m}^{n} w_i \cos(\omega t)$$

$$S(\omega) = \sum_{i=-m}^{n} w_i \sin(\omega t)$$

$$0 \le \omega \le \pi$$

And then compute G(□) and □(□) according to:

$$G(\omega) = \sqrt{C^2(\omega) + S^2(\omega)}$$

$$\Phi(\omega) = \arctan\left(\frac{S(\omega)}{C(\omega)}\right)$$

(though of course you have to pay careful attention to the edge cases where C(□) is 0 or negative, to get the right □(□).)

If the gain function G(□) is zero at some frequency, the filter removes those frequencies from the input signal. If it's merely small, that frequency is merely attenuated. If the gain function equals 1, that frequency is going through at full strength (now check the phase function to see if it's being shifted in time.) If the gain function is greater than one, that frequency is being amplified. For more details, and a mountain of examples, see the next chapter.

3.6 Limitations of the Discrete Finite Fourier Transform

There are differences in the properties of the Discrete Finite Fourier Transform and its idealised relative, the Continuous Infinite Transform. The finite span presents problems we call Resolution; the discrete sampling presents problems we describe as Aliasing.

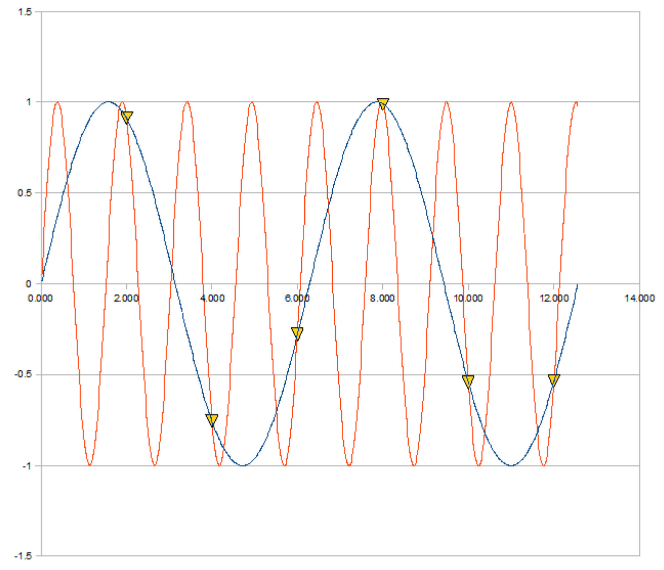
3.6.1 Aliasing

This only affects Discrete Fourier Transforms – in other words, data that is observed at fixed intervals.

Suppose you were doing inventories in a shop every month – but deliveries turned up on a three week cycle. What kind of pattern would you observe in the monthly data? Suppose you could observe the brightness of a variable star every Saturday night at 9pm – but the star actually had a 6 day variable cycle. What would you see?

If the processes generating the input signal go through more than one cycle per two observations, it is not possible to get that frequency out of the Fourier Transform directly. The Fourier Transform doesn't go to frequencies that high. The signal will instead show up at a lower frequency – we call it Aliasing.

Figure 3.13 Demonstration of Aliasing



If we are observing data at the time-points marked by the yellow triangles in this graph, then we cannot distinguish the two frequencies shown.

Mathematically, if a wave goes through a whole cycle and a bit between observations, it's only the bit that counts:

$$\omega + 2\pi k \equiv \omega$$

If a wave goes through just a little bit *less* than a full cycle between observations, it is still only that little bit that counts. But the phase changes – negative □, if you will.

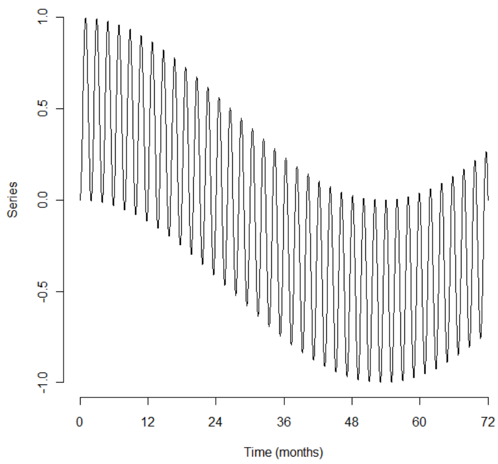
3.6.2 Resolution

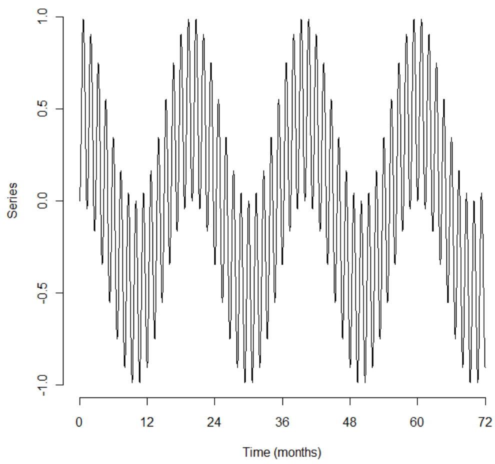
Our Discrete Finite Fourier Transform is defined at $\omega_j = \frac{2\pi j}{n}$ for j taking integer values in the range $0 \le j \le \frac{n}{2}$. This implies that the closest two frequencies can be, that we can still distinguish, is $\Delta\omega = \frac{2\pi}{n}$ and this is usually called the resolution. Clearly resolution gets better as n gets bigger, and for infinite span we can actually treat □ as continuous.

This is a reflection of a deeper problem. We can only compute the Fourier Transform for certain specific frequencies defined by the span of the data. If we add a bit more data, the frequencies we can measure at are different! And yet the frequencies actually present in the signal should be the same … what does this mean?

Two frequencies cannot be distinguished properly (the waves are not orthogonal) unless the product of the corresponding waves go through a whole cycle in the available span of data. See Figure 3.14 below for two examples of inconvenient overlaps.

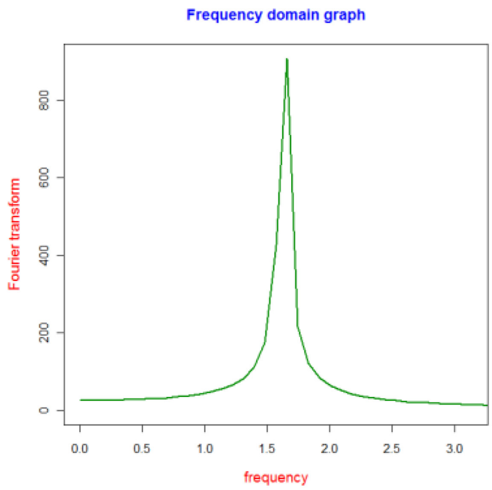
Figure 3.14 Examples of inconvenient overlaps





Let's take an example. We'll work with a span of 72 months. Suppose we're measuring what's happening with the wave $\sin(\frac{2\pi}{4}t)$, which goes through exactly 18 cycles in this span. If the input data actually has a signal at $\sin(\frac{2.075\pi}{4}t)$ then to compute the Fourier transform at $\frac{2\pi}{4}$ we will be adding up 18 and a bit cycles of the actual input signal … the waves will not quite cancel out (not be orthogonal) and we'll observe strong Fourier signal at frequencies quite close to the true input frequency. We will also observe weak Fourier signal at frequencies quite a long way from the true input frequency.

Figure 3.15 Observed signal



Generally, if the underlying input frequency doesn't fit the data span, a standard Finite Fourier Transform is going to give poor results. There are a variety of techniques for working around this, each with limitations.

Working around resolution problems

Spectrum: the Auto-correlation function shows most of the same frequencies, (though phase info is lost) but self-attenuates. This is rather handy given that the real world changes. See section 3.7 for details.

Windowing. Apply an artificial attenuation. We get convolution effects on the output FT – but the 'edge' effects disappear.

3.7 Spectral analysis

If we actually want to know what frequencies are present in a living time series, we may be ill-served by taking the Fourier Transform of the series directly. (E.g. if we take the transform of 14.5 years of data, all our seasonal peaks are going to be distorted.) Instead, we can take the auto-covariance of the data – then the Fourier Transform of *that*. This is called the spectrum. (It seems harsh that the direct Fourier transform *isn't* called the spectrum – but no; *this* is the spectrum.)

The auto-covariance (see Section 5.8 Auto-covariance) is going to have the same frequencies present. If your input data is highly seasonal then you'll have a strong spike at auto-covariance lag 12 – and lag 24, and so on. (If you had a cycle of length 5 in your raw data, you'd have high autocorrelation at lag 5, 10, etc.) But the spikes tail off in strength – the auto-covariance function provides its own attenuation – so adding a few data points on the end doesn't give significant modifications to the resulting Fourier Transform.

Now if you have an auto-correlation function that *doesn't* tail off (isn't absolutely summable), this isn't going to work. But that means you have a time series which is impressively predictable – once you've got enough data to start with (probably a year or two, three at the outside) you can forecast arbitrarily many years into the future, with at most finite loss in precision – and that doesn't happen for real time series.

The spectrum of a stationary process is the Fourier transform of the absolutely summable autocovariance function of the process.

Defining the Spectrum

Suppose Y_t is a stationary process with an absolutely summable autocovariance sequence γ_k, i.e.

$$\sum_{k=0}^{\infty} |\gamma_k| < \infty$$

This means that the Fourier transform of the sequence γ_k exists and it is called the spectrum of the process. See Section 5.8 for more on the auto-covariance function.

Substituting the autocovariance γ_k into (4.5) gives the spectrum of the process Y_t:

$$\begin{aligned} f(\omega) &= \sum_{k=-\infty}^{\infty} \gamma_k e^{-i\omega k} \\ &= \sum_{k=-\infty}^{\infty} \gamma_k \left(\cos(-\omega k) + i\sin(-\omega k)\right) \qquad (4.6) \\ &= \gamma_0 + 2\sum_{k=1}^{\infty} \gamma_k \cos(\omega k) \end{aligned}$$

where $\gamma_k = \gamma_{-k}$ and $0 \le \omega \le \pi$. Also, $\sin(0) = 0$, $\sin(\omega(-k)) = -\sin(\omega k)$, $\cos(\omega(-k)) = \cos(\omega k)$, and $e^{-i\omega k} = \cos(-\omega k) + i\sin(-\omega k)$. Note that the relationship $\cos(\omega k) = \frac{1}{2}\left(e^{i\omega k} + e^{-i\omega k}\right)$ can be used for an alternative derivation of (4.6).

Recall that the autocovariance generating function is defined by (4.2) as

$$\gamma(B) = \sum_{k=-\infty}^{\infty} \gamma_k B^k$$

By equating (4.6) and (4.2) the spectra and the autocovariance generating function can be shown to be related by

$$f(\omega) = \gamma\left(e^{-i\omega k}\right) \qquad (4.7)$$

This means that once the autocovariance generating function is known for a process the above relationship can then be used to derive the spectrum for that process.

As an aside, the sequence γ_k can be recovered through the inverse Fourier transform defined as

$$\gamma(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\omega)e^{i\omega k}d\omega \tag{4.8}$$

By setting $k = 0$ this gives the variance for the process Y_t as

$$Var\left(Y_t\right) = \gamma_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\omega)\,d\omega$$

This shows that the spectrum $f\left(\omega\right)$ can be interpreted as the decomposition of the stationary process variance. A peak in the spectrum indicates an important contribution to the variance from the components at frequencies in the corresponding interval.

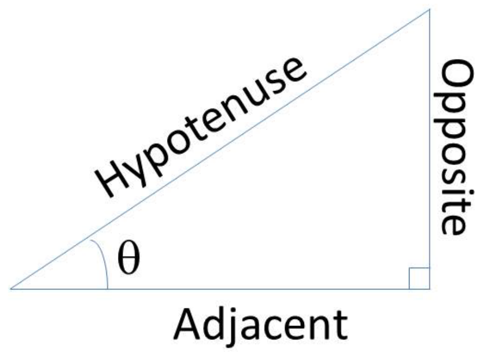
3.A Trigonometric Identities.

Fourier is a dance of trigonometric waves. If we want to prove very much about the properties of Fourier Transforms, we're likely to need some of these identities. If you aren't trying to prove anything, then the only thing you'd need any of these for would be working out a gain function in precise analytical terms … which is useful sometimes. If you're not doing that either, you can skip this stuff.

These are provided without proofs, though qualitative justification is included in some cases.

3.A.1 The very basics

Note: we are working in Radians; $2\pi = 360^\circ$

<table>
<tr>
<td rowspan="2">
</td>
<td>$\sin\theta = O/H$
$\cos\theta = A/H$ SOH CAH TOA
$\tan\theta = O/A = \dfrac{\sin\theta}{\cos\theta}$</td>
</tr>
<tr>
<td>$\sin^2\theta + \cos^2\theta = 1$
$\tan^2\theta + 1 = \sec^2\theta = 1/\cos^2\theta$</td>
</tr>
</table>

$\sin 0 = 0$ $\sin\frac{\pi}{2} = \sin 90^\circ = 1$ $\sin\frac{\pi}{3} = \sin 60^\circ = \frac{\sqrt{3}}{2}$ $\sin\frac{\pi}{4} = \sin 45^\circ = \frac{1}{\sqrt{2}}$	$\cos 0 = 1$ $\cos\frac{\pi}{2} = \cos 90^\circ = 0$ $\cos\frac{\pi}{3} = \cos 60^\circ = \frac{1}{2}$ $\cos\frac{\pi}{4} = \cos 45^\circ = \frac{1}{\sqrt{2}}$

In a broader context

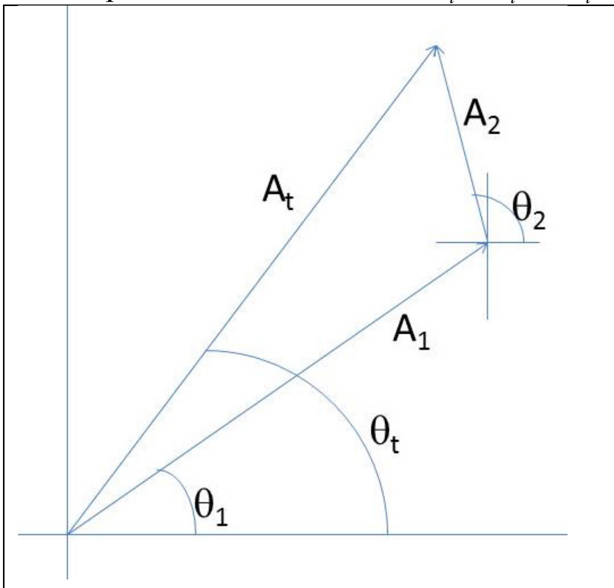
$\frac{d\sin x}{dx} = \cos x$	$\frac{d\cos x}{dx} = -\sin x$	$e^{ix} = \cos x + i\sin x$

Manipulating the angles

$\sin(x \pm y) = \sin x\cos y \pm \sin y\cos x$ $\cos(x \pm y) = \cos x\cos y \mp \sin x\sin y$ $\tan(x \pm y) = \frac{\tan x \pm \tan y}{1 \mp \tan x\tan y}$	$\sin 2x = 2\sin x\cos x$ $\cos 2x = \cos^2 x - \sin^2 x$	$\cos^2 x = \frac{1}{2}(1 + \cos 2a)$ $\sin^2 x = \frac{1}{2}(1 - \cos 2a)$
$2\sin x\cos y = (\sin(x+y) + \sin(x-y))$ $2\sin x\sin y = (\cos(x-y) - \cos(x+y))$ $2\cos x\cos y = (\cos(x+y) + \cos(x-y))$	$\sin\left(\frac{\pi}{2} + \theta\right) = \cos\theta$ $\cos\left(\frac{\pi}{2} - \theta\right) = \sin\theta$ $\tan\left(\frac{\pi}{2} - \theta\right) = \frac{1}{\tan\theta}$	$\sin(-x) = -\sin x$ $\cos(-x) = \cos x$ $\tan(-x) = -\tan x$

3.A.2 Adding two waves together

If we take any two vectors, $\vec{V_1} = A_1\cos\theta_1 + iA_1\sin\theta_1$ and $\vec{V_2} = A_2\cos\theta_2 + iA_2\sin\theta_2$, we can add them up to obtain a third vector $\vec{V_t} = A_t\cos\theta_t + iA_t\sin\theta_t$



$$A_t^2 = A_1^2 + A_2^2 + 2A_1A_2\cos(\theta_1 - \theta_2)$$

(Cosine Rule)

$$\tan\theta_t = \frac{A_1\sin\theta_1 + A_2\sin\theta_2}{A_1\cos\theta_1 + A_2\cos\theta_2}$$

Perhaps more usefully …

$$\tan(\theta_t - \theta_1) = \frac{A_2\sin(\theta_2 - \theta_1)}{A_1 + A_2\cos(\theta_2 - \theta_1)}$$

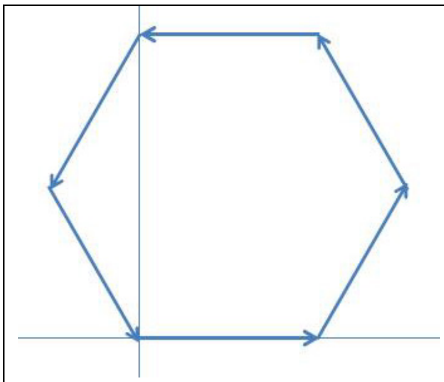
This gets particularly useful when considered in the Fourier context. If we're adding up two functions such as $C(\omega)\cos\omega t$ and $S(\omega)\sin\omega t$, then we can look at this as the x axis projection of the vectors $\overrightarrow{V_1} = C(\omega)\cos\omega t + iC(\omega)\sin\omega t$ and $\overrightarrow{V_2} = S(\omega)\cos\left(\frac{\pi}{2} - \omega t\right) + iS(\omega)\sin\left(\frac{\pi}{2} - \omega t\right)$.

Then $\theta_1 - \theta_2 = \frac{\pi}{2}$ and the equations simplify right down to

$$C(\omega)\cos\omega t + S(\omega)\sin\omega t = A(\omega)\cos\left(\omega t + \Phi(\omega)\right)$$
$$A^2(\omega) = C^2(\omega) + S^2(\omega)$$
$$\tan\left(\Phi(\omega)\right) = \frac{S(\omega)}{C(\omega)}$$

3.A.3 Adding lots of waves together

Discrete case

<table>
<tr>
<td rowspan="2">
Consider an n-gon on the complex plane …
Note that this is off-centre!</td>
<td>$\sum_{j=0}^{n-1}\left(\cos\frac{2\pi j}{n}\right)\begin{cases} = 0 \text{ if } n > 1 \\ = n \text{ if } n = 1 \end{cases}$</td>
</tr>
<tr>
<td>$\sum_{j=0}^{n-1}\left(\cos\frac{2\pi jk}{n}\right)\begin{cases} = 0 \text{ if not k} \equiv 0 \bmod n \\ = n \text{ if k} \equiv 0 \bmod n \end{cases}$</td>
</tr>
</table>

Continuous case (to be added)

3.B Proving the Transform

We'll use the Discrete Finite Fourier Transform.

In High maths terms, the set of all finitely valued functions f(t) defined at a fixed and finite set of evenly spaced points t_0 … t_{n-1}, forms a Hilbert space using a metric defined on the inner product $f(t) \bullet g(t) = \sum_{i=0}^{n-1} f(t_i)g(t_i)$ and we seek to show that the set of functions $\cos(\omega_j t)$ and $\sin(\omega_j t)$ for $\omega_j = \frac{2\pi j}{n}$ $\quad 0 \le j \le \frac{n}{2}$ form an orthogonal set of vectors that span the space. (Note that they are not unit vectors in this definition and they do not all share the same 'length'.)

In lower brow terms, we want to show that we can extract the coefficients of our sine waves and cosine waves using the given formula, and that we actually can regenerate any function we feel like by choosing the right set of coefficients.

Either way, we're relying heavily on the fact that the Fourier Transform is Linear.
$\mathcal{F}\left(af(t)+bg(t)\right)=a\mathcal{F}\left(f(t)\right)+b\mathcal{F}\left(g(t)\right)$

We will first show that our chosen set of functions is orthogonal.

(Not that this is hard but it seems I haven't typed it up yet!)

Next, we will show that our set of functions does span the space. Consider the function $y_t=\delta(t-a)$ (the Kronecker Delta function, zero everywhere except at t=a, where it takes the value 1.) It's pretty obvious that $\{\delta(t-t_i):0\le i<n\}$ spans the space; we can make any arbitrary function $f(t)$ according to $f(t)=\sum_{a=t_0}^{a=t_{n-1}} f(a)\delta(t-a)$. So if our Fourier Transform can make $\delta(t-t_i)$ for any i, we can make anything!

What coefficients would we get for y_t?

$$C(\omega_j)=\sum_{t=0}^{n-1} y_t\cos(\omega_j t)=\cos(\omega_j a)$$
$$S(\omega_j)=\sum_{t=0}^{n-1} y_t\sin(\omega_j t)=\sin(\omega_j a)$$

Now let's see if applying those coefficients in the inverse transform actually re-generates y_t.

$$g(t)=\frac{1}{n}C(\omega_0)+\frac{2}{n}\sum_{j=1}^{j\le(n-1)/2}\left(C(\omega_j)\cos\omega_j t+S(\omega_j)\sin\omega_j t\right)+\frac{1}{n}C(\omega_{n/2})\cos\omega_{n/2}t^{\#}$$

($^{\#}$ this term only included if n is even. Note that in this case $\omega_{n/2}=\dfrac{2\pi(n/2)}{n}=\pi$ and so $\sin\omega_{n/2}t=0\,\forall t$)

$$ng(t)=1+2\sum_{j=1}^{(n-1)/2}\left(\cos\omega_j a\cos\omega_j t+\sin\omega_j a\sin\omega_j t\right)+\cos\omega_{n/2}a\cos\omega_{n/2}t^{\#}$$

$$ng(t)=1+2\sum_{j=1}^{j\le(n-1)/2}\left(\cos\omega_j(a-t)\right)+(\cos\omega_{n/2}(a-t))^{\#}$$

$$ng(t)=1+\sum_{j=1}^{j\le(n-1)/2}\left(\cos\frac{2\pi j}{n}(a-t)\right)+\sum_{j'\ge(n+1)/2}^{j'=n-1}\left(\cos(\frac{2\pi(n-j')}{n}(a-t)\right)+(\cos\omega_{n/2}(a-t))^{\#}$$

Exploiting the identities $\cos(2\pi-x)=\cos x$ and that if n is even, that final term plugs the $j=n/2$ hole perfectly, we collapse that into the 'nth root of 1' trig identity:

$$ng(t)=\sum_{j=0}^{n-1}\left(\cos\frac{2\pi j}{n}(a-t)\right)\begin{cases}=0 \text{ if } t\ne a\\ =n \text{ if } t=a\end{cases}$$

Which shows that $g(t)=\delta(t-a)$ as required.