Stuart Riffle

Follow @StuartRiffle

I'm a graphics/systems programmer with 14 years in the industry, and my credits include Madden, Bond, and Need for Speed. I'm currently doing the indie thing. My specialties are optimization, rendering, concurrency, and compilers. I am delighted by all the pretty colors.

Understanding the Fourier transform

Yes, I realize that after reading the title of this post, 99% of potential readers just kept scrolling. So to the few of you who clicked on it, welcome! Don't worry, this won't take long.

A very long time ago, I was curious how to detect the strength of the bass and treble in music, in order to synchronize some graphical effects. I had no idea how to do such a thing, so I tried to figure it out, but I didn't get very far. Eventually I learned that I needed something called a Fourier transform, so I took a trip to the library and looked it up (which is what we had to do back in those days).

What I found was the Discrete Fourier Transform (DFT), which looks like this:

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

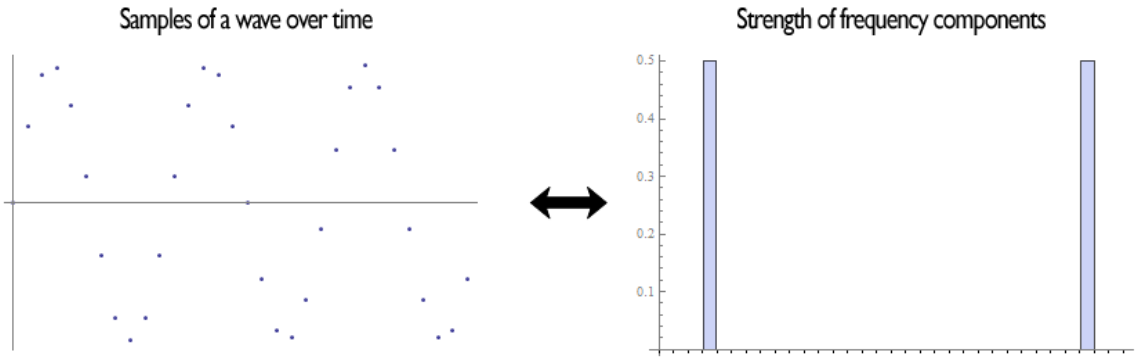
This formula, as anyone can see, makes no sense at all. I decided that Fourier must have been speaking to aliens, because if you gave me all the time and paper in the world, I would not have been able to come up with that.

Eventually, I was able to visualize how it works, which was a bit of a lightbulb for me. That's what I want to write about today: an intuitive way to picture the Fourier transform. This may be obvious to you, but it wasn't to me, so if you work with audio or rendering, I hope there's something here you find useful.

Disclaimer: my math skills are pitch-patch at best, and this is just intended to be an informal article, so please don't expect a rigorous treatment. However, I will do my best not to flat-out *lie* about anything, and I'm sure people will set me straight if I get something wrong.

A quick bit of background – what does the Fourier transform *do*? It translates between two different ways to represent a signal:

- The time domain representation (a series of evenly spaced samples over time)
- The frequency domain representation (the strength and phase of waves, at different frequencies, that can be used to reconstruct the signal)

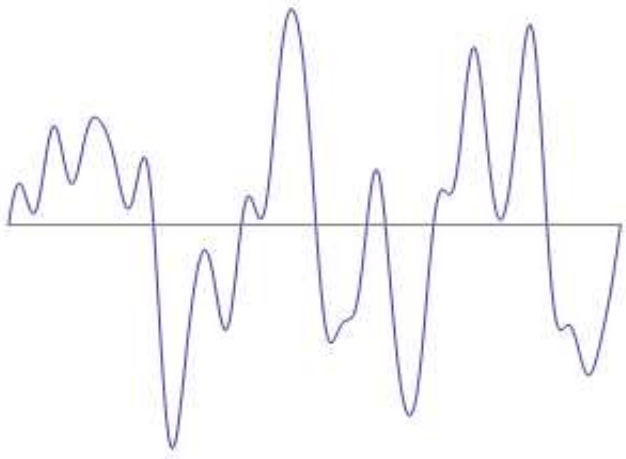


The picture on the left shows 3 cycles of a sine wave, and the picture on the right shows the Fourier transform of those samples. The output bars show energy at 3 cycles (and, confusingly enough, *negative* 3 cycles... more on that below).

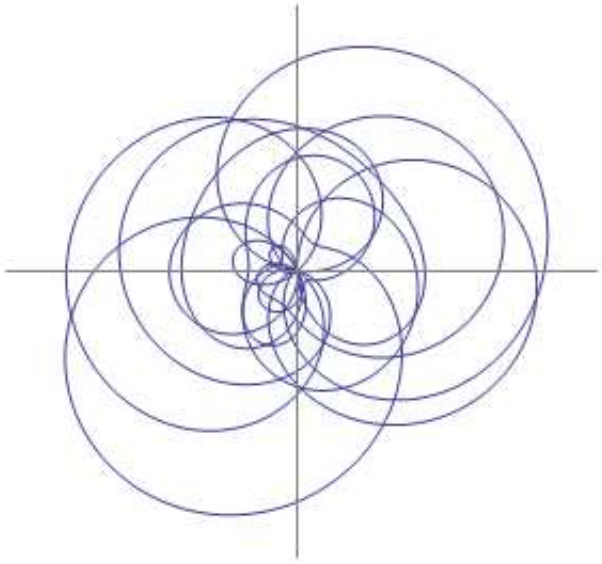
The inputs and outputs are actually complex numbers, so to feed a real signal (like some music) into the Fourier transform, we just set all the imaginary components to zero. And to check the strength of the frequency information, we just look at the magnitude of the outputs, and ignore the phase. But let's never mind all that for now.

What are we trying to accomplish? We've got a sampled signal, and we want to extract frequency information from it. The Fourier transform works on a periodic, or looping signal. This seems like a problem, since we don't actually have any signals like that. In practice, you just take a small slice of a longer signal, fade both ends to zero so that they can be joined (which is a whole topic unto itself), and pretend it's a loop.

Let's make things simple and say that our loop repeats once per second.

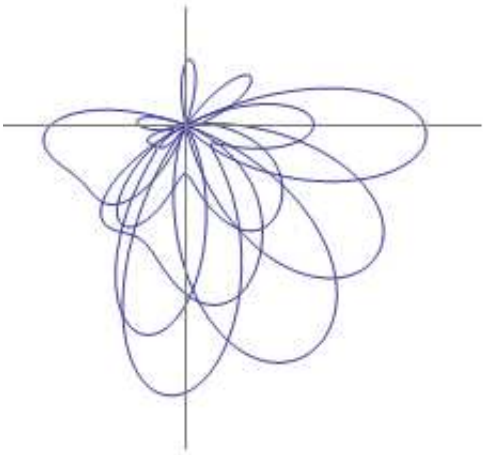


Picture it as a bead, sliding up and down along a thin rod, tracing out the signal. So as this bead is bobbing up and down, look what happens if we spin the rod at a rate of, say, 10 revolutions per second:

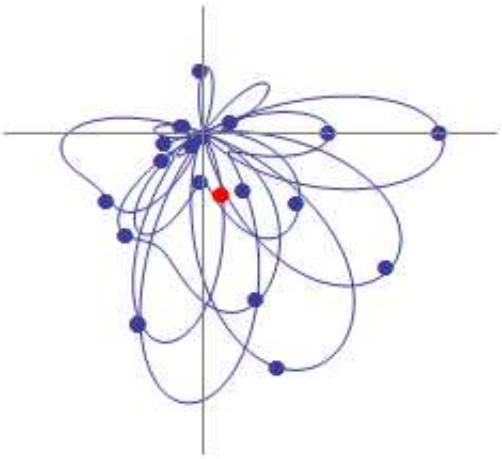


We get a scribble, as you'd expect. And it is roughly centered on the origin.

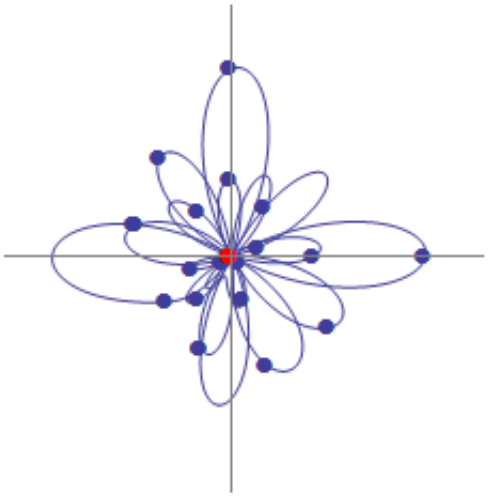
Now, let's assume we know there's some energy in the signal at 3 Hz, and we want to measure it. What that means is that on top of whatever else is causing the signal to wobble around, we've added a wave that oscillates 3 times per second. It has a high point every 1/3 of a second, and corresponding low points in between, also spaced 1/3 of a second apart. You can probably see now how we might be able to detect it... let's try spinning our signal at a matching rate of 3 revolutions per second.



Since the signal completes a rotation every 1/3 of a second, all the high points in our 3 Hz wave line up at the same part of the rotation, and this pulls the whole scribble off-center. How can we quantify that? The easiest way would be to record a bunch of points as we rotate, and average them to find their midpoint:



It makes sense that the distance of this midpoint from the origin is proportional to the strength of the signal, because as the high points in our signal get higher, they will move the scribble farther away. But what if the signal contains no energy at 3 Hz? Let's remove the 3 Hz wave and see:



Now there is nothing to pull the scribble off center, and all of the other oscillations tend to (approximately) balance each other out.

This looks promising as a way to detect energy at a given frequency. Time to translate it into math! For a looping signal of N samples:

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

To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

(Raising e to an imaginary power produces rotation around a unit circle in the complex plane, according to Euler's formula. How? Magic, as far as I can tell. But apparently it's true).

So this equation is a little different from what we started with. I've added a normalization factor of 1/N, and changed the sign of the exponent. I also rearranged the terms slightly for clarity. This form is normally called the *inverse* DFT, which is confusing, but apparently the difference between the DFT and IDFT is a matter of convention, and can depend on the application. So, let's call that close enough.

Anyway, once you can "see" what's going on in your head, a lot of the quirks of working with the DFT become much less mysterious. If you've had to work with DFT output before, you may have wondered:

- *Why does the first element in the result (k=0) contain the DC offset?* Because in that case, our samples don't spin at all, so all we're doing is averaging them.
- *Why doesn't the DC offset affect the frequency information?* Because adding a constant value to all the samples just makes the whole scribble bigger, which doesn't affect the midpoint.
- *Why does the second half of the output array contain a mirror image of the first half?* It's just our old friend aliasing. When calculating the last element (k=N-1), we're rotating by (N-1)/N at each step, which is almost all of the way around. This is the same as taking small steps (1/N) in the wrong direction. That's why the result at (k=N-1) has the same magnitude as (k=1). It's equivalent to processing a negative frequency of (k=-1).
- *Why does a sine wave with amplitude 1.0 come out of the DFT as 0.5?* When we spin the sine wave, we get a circle of diameter 1.0, but it's midpoint is only half that distance away from the origin.
- *Where is the other half of the energy then?* It's hiding in the negative frequency part!

Hopefully this was more helpful than confusing.

And if you'd like to get updates on my game development work, come subscribe to my RSS feed at pureenergygames.com!

Gustavs • 3 years ago

I made some animations to explain the spinning thing.

http://fixplz.blourp.com/blog/...

50 | 1 • Reply • Share ›

Carlos ➚ Gustavs • a year ago

This helped me understand it a lot better, mad props!

2 | • Reply • Share ›

Stuart Riffle ➚ Gustavs • 3 years ago

Cool! I think that will help a lot of people.

2 | • Reply • Share ›

MySchizoBuddy ➚ Gustavs • 2 months ago

link no longer works. can you provide an updated link

| • Reply • Share ›

blah_blah_2056 ➚ MySchizoBuddy • 2 months ago

http://web.archive.org/web/201...

| • Reply • Share ›

sigmaalgebra • a year ago

No. You are talking the discrete Fourier transform or really just Fourier series. Fourier series is for periodic waves, but the Fourier transform is not. Or you are taking a 'finite' Fourier transform, that is, on essentially a finite group or a circle instead of the whole real line.

The reason you can get by with the discrete stuff, that is, discrete in time instead of considering a continuous wave, is the fundamental theorem of interpolation theory that says that, in the case of a 'bandlimited wave', those discrete points are enough to interpolate back to the full wave form. To interpolate back, at each point in time, just put in a sin(x)/x which is essentially the Fourier transform of a box and, then, you could do convolution with the wave I mentioned, which is really what I did mention.

What the exponentials you are using are is orthogonal. That is, if you take the inner product of different pairs, then you get zero. The Fourier formula is the innner product of the signal on each of the orthogonal axes. That inner product is basically an orthogonal projection. So, you are taking the given signal and taking its orthogonal projection onto the various axes that

see more

23 | • Reply • Share ›

Cristián Arenas Ulloa • 3 years ago

Could you add a font-coded version of the formula for us color blind

people?

8 • Reply • Share ›

MySchizoBuddy ➚ Cristián Arenas Ulloa • 2 months ago

don't color blind people switch colors? so the same color will be switched on the text and the formula.

• Reply • Share ›

Fourier ➚ MySchizoBuddy • a month ago

Please, before making comments like those take your time to learn what color blindness exactly is. It depends a lot on the degree of color blindness, but the most common form of color blindness is the inability to distinguish different colors, such as red and green, because they see both colors in a brown-ish tone.

• Reply • Share ›

MySchizoBuddy ➚ Fourier • a month ago

sorry didn't know color blindness results in overreaction. lesson learned.

3 | 1 • Reply • Share ›

urtubia • 3 years ago

Thanks!!
I did something really quick on processing to sort of illustrate this...

http://hascanvas.com/fftvisual...

7 • Reply • Share ›

ahfoo ➚ urtubia • 3 years ago

Ooh, now we're getting into the good stuff. Animations.

So, how about a little explanation of what is going on in those three animations at the bottom. Why do the orbits seem to be changing when the waveform is cyclical?

1 • Reply • Share ›

urtubia ➚ ahfoo • 3 years ago

Hehe, the animation is cool to look at :)

...So the 3 animations at the bottom are the "spinning rods" spinning at different frequencies using the wave on top as their function. The interesting one is on the bottom right, which spins at the same frequency of one of the wave components of the original function. And, as you can see, the figure created is not perfectly symmetrical, which means that there is an energy level being detected at that

frequency.

• Reply • Share ›

ahfoo ➚ urtubia • 3 years ago

So what is the energy level we're seeing in the one on the right?

• Reply • Share ›

Karras ➚ ahfoo • a year ago

Over 9000.

2 • Reply • Share ›

ranza • 3 years ago

Man, I owe you a beer for this! Great work!
Although I think I would have a slight suggestion how to make the description a bit better, how about describing the rotation part like this:
Now imagine that you pull the time axis towards your direction, so that it points right into your eyes.
Secondly you point a flashlight in it's direction and a screen behind the access, so that the bead which travels on our signal gets projected on the screen.
Tracing that dot, we get the pictures you've created.

Once again, awesome work!

6 • Reply • Share ›

Chui Tey • 3 years ago

Simply brilliant. If it wasn't for your humility - your so called pitch patch knowledge of maths - I would have not persisted to the end.

6 • Reply • Share ›

the one true josh • 3 years ago

The explanation is very well done, but the colorized formula with commentary is spectacular! All math should be taught this way.

4 • Reply • Share ›

Victor Maia • 2 years ago

This was not "useful". This was the most useful article in math that I have ever read. I wish I could see this kind of explanation for everything. Thank you very much.

3 • Reply • Share ›

Tory • 3 years ago

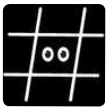
A sound of "Fourier transform" always made me tremble, but not after I read this!!!

I think what will also help people really understand what is going on are a

couple of simple graphs that show a signal with only one frequency to see how it translates to a bead/rod image.

Thanks for such a great post!

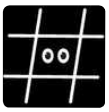
3 ▲ ▼ • Reply • Share ›



Brent Lewis • a year ago

The n/N portion isn't part of the averaging - it's used to create something like a curve parameter. At the beginning of the samples, it's 0. At the end, it's 1. Multiply this value by 2 pi k and you've got the "rotation" in radians. Unfortunately, when I searched for "understanding" I was looking for something rigorous. Nice article at any rate.

2 ▲ ▼ • Reply • Share ›



isomorphisms • 2 years ago

> Raising e to an imaginary power produces rotation around a unit circle in the complex plane, according to Euler's formula. How? Magic, as far as I can tell. But apparently it's true

Euler's formula $\exp(i \cdot \theta) = \cos\theta + i \sin\theta$. If you're rusty on trig, $\sin$ tells you the vertical coordinate at a particular angle on the circle, and $\cos$ tells you the horizontal coordinate.

(Picture a spinny green radar tracking a torpedo in a submarine movie: the rectangular grid coordinates are contrasted to the circular spinny beeper.)

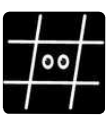
Anyway if you accept that imaginary is the vertical axis and real is the horizontal axis, then of course going around the unit circle decomposes x to cosine and y to sine. That's just what $\sin$ and $\cos$ do.

If that bothers you because "vertical" and "horizontal" are just conveniences for ourselves on paper/chalkboard...

The way to derive it is with power series. $\exp = \Sigma\ 1/n!$ and that can be decomposed into alternating odd and even components which happen to correspond to the famous circle functions $\sin$ and $\cos$. Which does seem kind of magical. I don't know if de Moivre just looked at the algebra and knew his series very well or if there was some geometrical intuition behind it.

But a cheap way to remember this $e^{i\theta} =$ circle business is that the relation $e^{i\pi} = -1$ falls out of it. π is 180° around the circle. 180° opposite of +1 is −1.

2 ▲ ▼ • Reply • Share ›

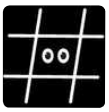


Gordon Bailey • 2 years ago

This is a link that I've bookmarked and come back to a few times - thanks a lot for giving an extremely clear and intuitive explanation of a complex

a lot for giving an extremely clear and intuitive explanation of a complex subject!

2 ▲ | ▼ • Reply • Share ›



Greg • 3 years ago

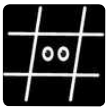
Article about Fourier transform in Wikipedia should reference this page!

2 ▲ | ▼ • Reply • Share ›

Toby Champion ➚ Greg • a year ago

It does now.

1 ▲ | ▼ • Reply • Share ›



kkk • 3 years ago

Thank you very much for your article.

One more thing that I am not very clear is "What is energy?" "what is energy at certain frequency?"

i know that one big usage of fourier transform is to do image filters, like those inside photoshop.

but given an image, what frequency should i use to apply fourier transform?

Thank you very much.

2 ▲ | ▼ • Reply • Share ›

Stuart Riffle ➚ kkk • 3 years ago

When I say there is "energy" at a certain frequency, all I mean is that the signal contains oscillations at that frequency, and we can detect them. If you scroll up a bit and read my reply to Carl, I tried to describe that a bit better.

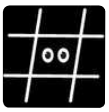
As for processing images, yes: the DFT can be used to make filters (though there are usually much more efficient ways to do it)! As for which frequency: all of them. A 2D DFT is suprisingly easy to set up... you process all of your rows independently, to get a "stack" of 1D frequency spectra. Then you do a DFT on all of the columns. What you get is a 2D frequency spectrum, which you can then play with (knock out frequencies, boost them, look for features, etc). When you are done, you do the inverse transform to reconstruct the image.

There are great (and more precise) descriptions of the 2D FFT on these pages:
http://www.cs.unm.edu/~brayer/...
http://sharp.bu.edu/~slehar/fo...

And please, don't implement the DFT as described above... use an

FFT implementation instead. It does the same thing, but much faster.

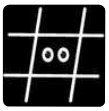
2 ▲ | ▼ • Reply • Share ›



Augustine Cost • a year ago

Here's my attempt to explain imaginary exponents and Euler's formula: http://guscost.com/2012/08/06/...

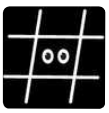
1 ▲ | ▼ • Reply • Share ›



Mediocritas • 2 years ago

This is the best explanation of the FT that I've seen to date. The rotational plots and colour-coded FT formula are strokes of genius. Good work! I hope this gets used to educate some educators.

1 ▲ | ▼ • Reply • Share ›



Michael • 3 years ago

Hey man, this was an amazing work. The only thing I might add is that e^i itself only *allows* for spin, as opposed to producing it.

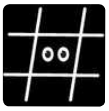
1 ▲ | ▼ • Reply • Share ›

Michael ➚ Michael • 3 years ago

Applying consistency(e) to(^) complexity(i) gives continuity (so spin can occur).

If that doesn't help you there are other mind trick approaches I can tell you about.

▲ | ▼ • Reply • Share ›



Shawn • 3 years ago

I love how you describe this great job. However when you say
"The inputs and outputs are actually complex numbers, so to feed a real signal (like some music) into the Fourier transform, we just set all the imaginary components to zero."
This is incorrect, the amplitude is the absolute value of the complex number so.
amplitude = sqrt(real^2+imaginary^2). The phase of the signal (which can be visualized by horizontally offsetting a sin wave) is stored in the direction of the complex vector so that would be arctan(imaginary/real). Taking the real component looks close but its not exactly right.

In your visual model the amplitude corresponds to the distance away from the origin and your phase corresponds to the rotational position. So if you phase shift the signal by delaying it slightly you will notice that your average midpoint will rotate around the center axis while the distance is preserved.

An interesting thing to note here is assuming that your output of your dft

has the same number of samples as your input you have preserved all the information needed to perfectly reconstruct your signal (computers will accumulate rounding errors but that isn't a problem with the math) so inverse_fft(fft(signal)) == signal.

Anyway sorry to nit pick, great article.

1 • Reply • Share ›

Stuart Riffle → Shawn • 3 years ago

Thanks Shawn! I only meant that when we use real numbers as inputs, the imaginary part is just set to zero. I didn't mean to throw away the imaginary component of the output! As you said, that won't work. I'll see what I can do to make that more clear. Thank you for pointing that out.

1 • Reply • Share ›

Shawn → Stuart Riffle • 3 years ago

Ah ok I see that now, my mind must have skipped over that second sentence a bit. I made the mistake of ignoring the imaginary component back when I first tried to use a dft so I think its a good thing to be clear about.
Cheers,
Shawn

1 • Reply • Share ›

Stuart Riffle • 3 years ago

I just want to say thank you to everyone, both for the encouraging comments, and for pointing out exactly what I didn't explain too well. I'll do my best to make the article more clear, and post a new version on my blog in a couple of days.

Shameless: if you subscribe to my feed, you'll get the update automatically!
http://feeds.feedburner.com/pu...

Cheers,
-Stuart

1 • Reply • Share ›

oas • 3 years ago

What's the difference between the first spinning and the last one? You removed the 3 Hz for the last one - why is that different from where you started?

1 • Reply • Share ›

Stuart Riffle → oas • 3 years ago

Hi oas. The first spin was at a much faster speed: 10 revolutions

per second. There's no energy at that frequency, so the scribble is pretty balanced.

Thanks for pointing out that it's not clear... I should put a little caption next to the graphs to say what they are supposed to show.

• Reply • Share ›

oas → Stuart Riffle • 3 years ago

The first one is at 10, the second at 3, the last says "Let's remove the 3 Hz wave and see" - what's left?

• Reply • Share ›

Stuart Riffle → oas • 3 years ago

Exactly. When we spin at 3 revolutions per second, we are able to detect the energy at 3 Hz because the graph is skewed. I was just trying to show that when there's no energy at 3 Hz, the graph is no longer skewed.

The first graphs are exactly the same signal, just spun at different speeds. For the last graph, I modified the signal to show what it would look like.

1 • Reply • Share ›

adam • 3 years ago

Ding! *light bulb going off in my head/*

1 • Reply • Share ›

Revolves • 3 years ago

Nice post there, Stuart. Fourier transform is useful because we normally use sinusoidal inputs to analyse the behavior of various circuit components. The output that we get from each circuit depends on the frequency of the input sinusoids.

Once we know how a circuit responds to sinusoids, we can use Fourier transform on any arbitrary signal. The Fourier transform would give us a set of frequency components and their amplitudes, which when added together would give you your original signal. Since each component is a