Complex Fourier Transform and Numerical Solutions

Complex Fourier Transform

From before, if a function is periodic in time T

$$x(t) = x(t+T)$$

it can be expressed as a sum of sine and cosine terms

$$x(t) = a_0 + \sum a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t).$$

It can also be expressed in terms of a complex exponential:

$$x(t) = c_0 + \sum c_n e^{-jn\omega_0 t}$$

where

$$c_n = \frac{1}{T}\int_T x(t) \cdot e^{-jn\omega_0 t} \cdot dt$$

The relationship between the complex Fourier transform and its sine and cosine version comes from Euler's identity

$$\cos(x) = \left(\frac{e^{jx}+e^{-jx}}{2}\right)$$

$$\sin(x) = \left(\frac{e^{jx}-e^{-jx}}{2j}\right)$$

Suppose c_n has a real and complex part

$$c_n = a_n + jb_n$$

and c_{-n} is its complex conjugate (it will be for real functions)

$$c_{-n} = a_n - jb_n$$

Then the nth harmonic of x(t) will be

$$x_n(t) = c_n e^{jn\omega_0 t} + c_{-n} e^{-jn\omega_0 t}$$

$$x_n(t) = (a_n + jb_n)e^{jn\omega_0 t} + (a_n - jb_n)e^{-jn\omega_0 t}$$

$$x_n(t) = a_n(e^{jn\omega_0 t} + e^{-jn\omega_0 t}) + jb_n(e^{jn\omega_0 t} - e^{-jn\omega_0 t})$$

$$x_n(t) = 2a_n\left(\frac{e^{jn\omega_0 t}+e^{-jn\omega_0 t}}{2}\right) + 2j^2 b_n\left(\frac{e^{jn\omega_0 t}-e^{-jn\omega_0 t}}{2j}\right)$$

$$x_n(t) = 2a_n\cos(n\omega_0 t) - 2b_n \sin(n\omega_0 t)$$

The relationship between the complex Fourier transform and its sine and cosine version is

2*real(c_n) = cosine terms

-2*imag(c_n) = sine terms

Example 1: Find the complex Fourier transform for

$$x(t) = \sum e^{jn\omega_0 t}$$

Solution:

$$c_n = \frac{1}{T}\int_T x(t)\cdot e^{-jn\omega_0 t}\cdot dt$$

$$c_n = \frac{1}{T}\int_T e^{jn\omega_0 t}\cdot e^{-jn\omega_0 t}\cdot dt$$

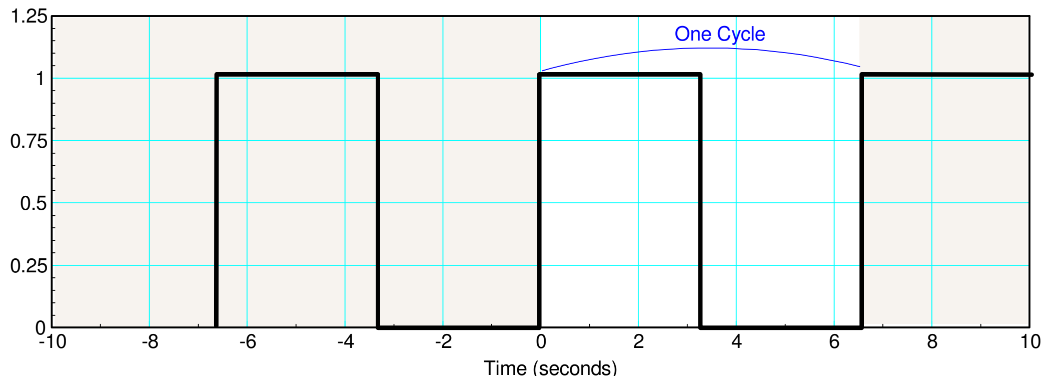
$$c_n = \frac{1}{T}\int_T 1\cdot dt$$

$$c_n = 1$$

x(t) is already in complex exponential form so there's nothing to do. This is also why you use the complex conjugate in the exp() term.

Example 2: Find the complex Fourier transform for a 1 rad/sec 50% duty cycle square wave.

$$x(t) = \begin{cases} 1 & 0 < t < \pi \\ 0 & \pi < t < 2\pi \end{cases}$$



Solution: The fundamental frequency is one

$$\omega_0 = \frac{2\pi}{T} = 1$$

so

$$c_n = \frac{1}{2\pi}\int_0^{2\pi} x(t)\cdot e^{-jnt}\cdot dt$$

$$c_n = \frac{1}{2\pi}\int_0^{\pi}\cdot e^{-jnt}\cdot dt$$

$$c_n = \frac{1}{2\pi}\cdot\left(\frac{1}{-jn}e^{-jnt}\right)_0^{\pi}$$

$$c_n = \frac{j}{2n\pi}\left((-1)^n - 1\right)$$

or

$$c_n = \begin{cases} \left(\frac{-j}{n\pi}\right) & \text{n odd} \\ 0 & \text{n even} \end{cases}$$

This is the same as we got using sine and cosine terms

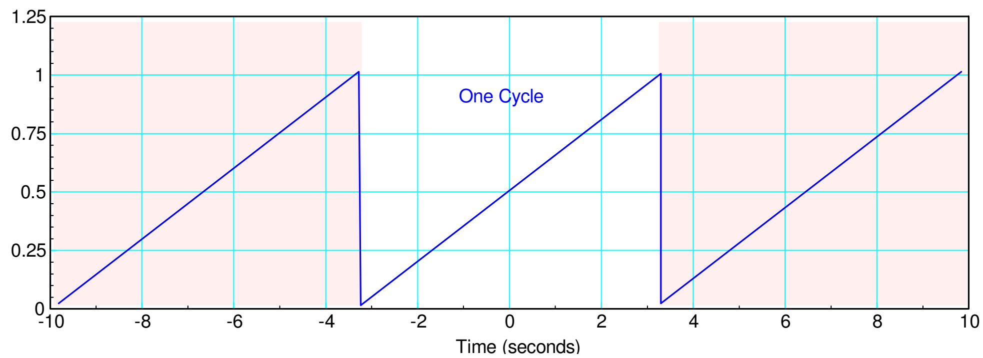
$$b_n = -2 \cdot imag(c_n)$$

$$b_n = \left(\frac{2}{n\pi}\right) \qquad \text{n odd}$$

c_n is the phasor representation for sin(), and much easier to compute.

Example 3: Find the complex Fourier transform for a saw-tooth wave:

$$x(t) = 0.5 + \frac{t}{2\pi}$$



Solution: The 0.5 affects the DC term

$$c_0 = \frac{1}{2\pi}\int_{-\pi}^{\pi} x(t) \cdot dt$$

$$c_0 = 0.5$$

The other terms come from

$$c_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{t}{2\pi} \cdot e^{-jnt} \cdot dt$$

Integrating by parts

$$c_n = \left(\frac{1}{(2\pi)^2}\right)\left(\frac{1}{-jn} \cdot t \cdot e^{-jnt} - \frac{1}{(-jn)^2}e^{-jnt}\right)_{-\pi}^{\pi}$$

$$c_n = \left(\frac{1}{(2\pi)^2}\right)\left(\left(\frac{1}{-jn} \cdot \pi \cdot (-1)^n - \frac{(-1)^n}{(jn)^2}\right) - \left(\frac{1}{-jn} \cdot (-\pi) \cdot (-1)^n - \frac{(-1)^n}{(jn)^2}\right)\right)$$

$$c_n = \left(\frac{1}{(2\pi)^2}\right)\left(\frac{2\pi}{-jn} \cdot (-1)^n\right)$$

$$c_n = \left(\frac{j(-1)^n}{2n\pi}\right)$$

This also gives the cosine and sine terms

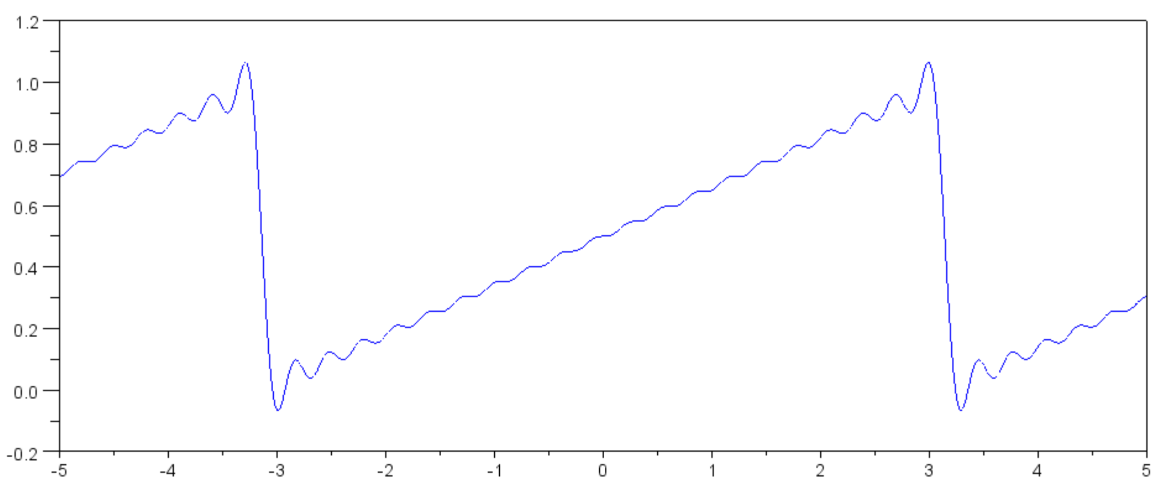
$$a_n = 2 \cdot real(c_n) = 0$$

$$b_n = -2 \cdot imag(c_n) = \left(\frac{-(-1)^n}{n\pi}\right)$$

harmonic	0	1	2	3	4	5	6	7
cn	0.5	-j0.0796	j0.0398	-j0.0265	j0.0199	-j0.0159	j0.0133	-j0.0114
an 2*real(cn)	0.5	0	0	0	0	0	0	0
bn -2*imag(cn)	0	0.1592	-0.0796	0.0531	-0.0398	0.0318	-0.0265	0.0227

Plotting this in Matlab out to 20 terms

```
bn = zeros(20,1);

for n=1:20
   bn(n) = - ((-1)^n / (n*pi));
   end

x = 0.5 + 0*t;

for n=1:20
   x = x + bn(n) * sin(n*t);
end

plot(t,x)
```



Fourier Approximation to a Sawtooth Wave taken out to 20 terms

Numerical Solutions

Hand calculations are a lot of fun, but some functions are really hard to integrate by hand. Fortunately, you can always use Matlab.

Given a function, x(t), which is periodic in time T

$$x(t) = x(t+T)$$

the sine and cosine terms can be found in Matlab as

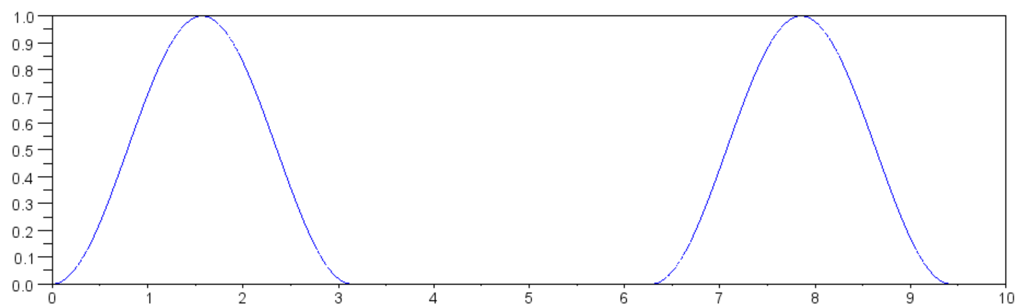
$$a_0 = mean(x)$$

$$a_n = 2 \cdot mean(x \cdot \cos(n\omega_0 t))$$

$$b_n = 2 \cdot mean(x \cdot \sin(n\omega_0 t))$$

For example, find the Fourier transform for

$$x(t) = \begin{cases} \sin^2(t) & 0 < t < \pi \\ 0 & \pi < t < 2\pi \end{cases}$$



The first 5 harmonics found using Matlab are:

DC:

```
a0 = mean(x)
    0.2499676
```

Cosine Terms

```
a1 = 2*mean(x .* cos(t))
  - 2.373D-12

a2 = 2*mean(x .* cos(2*t))
  - 0.2499676

a3 = 2*mean(x .* cos(3*t))
  - 2.373D-12

a4 = 2*mean(x .* cos(4*t))
    2.373D-12

a5 = 2*mean(x .* cos(5*t))
  - 2.373D-12
```

Sine terms

```
b1 = 2*mean(x .* sin(t))
    0.4243582

b2 = 2*mean(x .* sin(2*t))
    9.254D-15

b3 = 2*mean(x .* sin(3*t))
  - 0.0848716

b4 = 2*mean(x .* sin(4*t))
    1.857D-14

b5 = 2*mean(x .* sin(5*t))
  - 0.0121245
```

So,

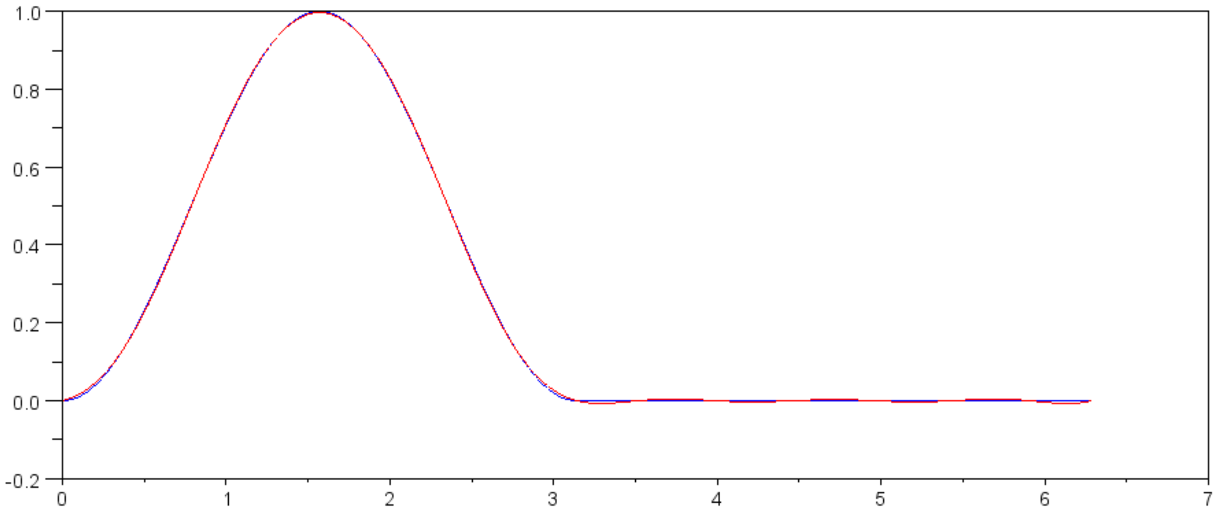
$$x(t) \approx 0.2499 + 0.4243\sin(t) - 0.2499\cos(2t) - 0.0848\sin(3t) - 0.0121\sin(5t)$$

This also works with the complex Fourier transform (but you get the cosine and sine terms at the same time)

```
-->c1 = mean(x .* exp(-j*t))

c1 =   - 1.187D-12 - 0.2121791i

-->c2 = mean(x .* exp(-j*2*t))

c2 =   - 0.1249838 - 4.627D-15i

-->c3 = mean(x .* exp(-j*3*t))

 c3  =  - 1.187D-12 + 0.0424358i

-->c4 = mean(x .* exp(-j*4*t))

 c4  =    1.187D-12 - 9.278D-15i

-->c5 = mean(x .* exp(-j*5*t))

 c5  =  - 1.186D-12 + 0.0060623i
```

Plotting x(t) and its Fourier approximation taken out to five terms together:

```
xf = a0 + b1*sin(t) + a2*cos(2*t) + b3*sin(3*t) + b5*sin(5*t);

plot(t,x,t,xf)
```



x(t) (blue) and its Fourier approximation taken out to the 5th harmonic (red)

This also works with the complex Fourier transform

```
-->c1 = mean(x .* exp(-j*t))

c1 =   - 1.187D-12 - 0.2121791i

-->c2 = mean(x .* exp(-j*2*t))

c2 =   - 0.1249838 - 4.627D-15i

-->c3 = mean(x .* exp(-j*3*t))

 c3  =  - 1.187D-12 + 0.0424358i

-->c4 = mean(x .* exp(-j*4*t))

 c4  =    1.187D-12 - 9.278D-15i

-->c5 = mean(x .* exp(-j*5*t))

 c5  =  - 1.186D-12 + 0.0060623i
```