Properties of Fourier Transforms

Background

Assume x(t), y(t), and their Fourier transforms

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

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

Linearity:

If you multiply x(t) by a constant, its Fourier coefficients are multiplied by the same constant

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

If you add two functions, the Fourier coefficients add

$$x(t) + y(t) = \sum X_n e^{jn\omega_0 t} + \sum Y_n e^{jn\omega_0 t}$$

$$= \sum (X_n + Y_n) e^{jn\omega_0 t}$$

Delay

If x(t) is delayed by time T, Xn is multiplied by $e^{-n\omega_0 T}$

$$e^{-sT} \cdot x(t) = e^{-sT} \cdot \sum X_n e^{jn\omega_0 t}$$

$$= \sum X_n \cdot e^{-sT} \cdot e^{jn\omega_0 t}$$

$$= \sum X_n \cdot e^{-n\omega_0 T} \cdot e^{jn\omega_0 t}$$

Differentiation:

$$\frac{dx}{dt} = \frac{d}{dt}\left(\sum X_n e^{jn\omega_0 t}\right)$$

$$= \sum X_n \cdot jn\omega_0 \cdot e^{jn\omega_0 t}$$

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

Integration:

$$\int x dt = \int \left(\sum X_n e^{jn\omega_0 t}\right) dt$$

$$= \sum \left(\frac{X_n}{jn\omega_0}\right) e^{jn\omega_0 t}$$

Time Scaling

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

The Fourier coefficients don't change. All that changes are the frequencies

$$\omega_0 \to a\omega_0$$

Convolution

$$\int x(\tau)y(t-\tau)d\tau = \int (\sum X_n e^{jn\omega_0\tau})(\sum Y_n e^{jn\omega_0(t-\tau)})d\tau$$

Summary

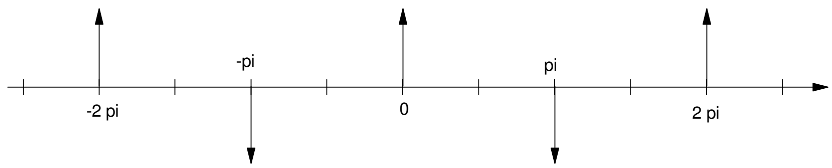
Operation	x(t)	Xn
amplitude scaling	a x(t)	$X_n \to aX_n$
addition	x(t) + y(t)	Xn + Yn
delay T seconds	x(t-T)	$X_n \to e^{-jn\omega_0}X_n$
differentiation	;$\frac{dx}{dt}$	;$X_n \to \left(\frac{1}{jn\omega_0}\right)X_n$
integration	$\int x \cdot dt$	$X_n \to (jn\omega_0)X_n$

With these properties you can derive the Fourier transform for different functions

Example 1: Delta Train:

$$x(t) = x(t + 2\pi)$$

$$x(t) = \delta(t) - \delta(t - \pi)$$



The complex Fourier transform for a delta function with a period of 2π is $\frac{1}{2\pi}$

$$\delta(t) \leftrightarrow \frac{1}{2\pi}$$

A delayed delta function becomes

$$\delta(t - \pi) \leftrightarrow e^{-jn\pi} \cdot \left(\frac{1}{2\pi}\right) = \frac{(-1)^n}{2\pi}$$

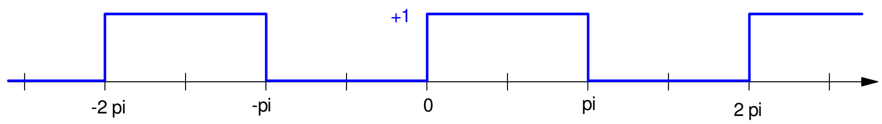
Subtracting gives the Fourier transform for x(t)

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

$$X_n = \begin{cases} \left(\frac{1}{\pi}\right) & \text{n odd} \\ 0 & \text{n even} \end{cases}$$

Example 2: Square Wave. If you integrate the previous function, you get a square wave

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



The Fourier transform for a square wave is therefore

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

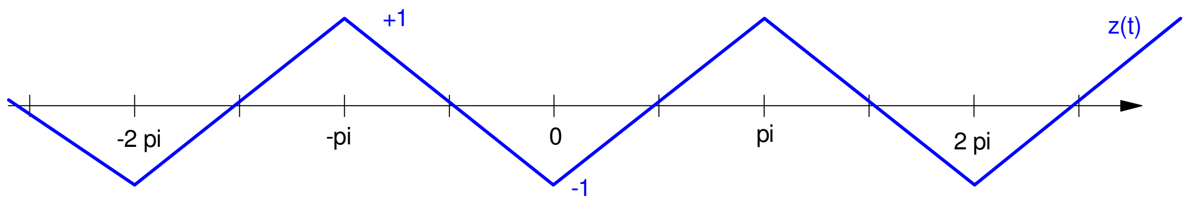
$$Y_n = \begin{cases} \left(\frac{-j}{n\pi}\right) & \text{n odd} \\ 0 & \text{n even} \end{cases}$$

This is the same result we got twice before

Example 3: Triangle Wave: If you take the previous square wave,

- Remove the DC offset (so the square wave goes from -0.5 to +0.5)
- Integrate, and
- Multiply by $\left(\frac{2}{\pi}\right)$

you get a triangle wave



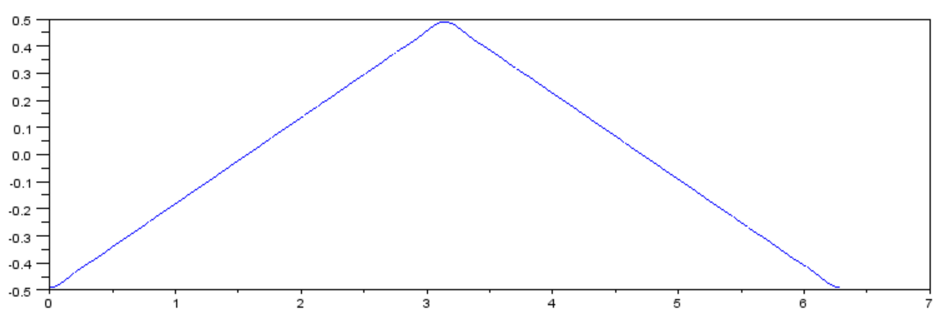
$$z(t) = \frac{32}{\pi} \int y(t) \cdot dt$$

$$Z_n = \left(\frac{32}{\pi}\right)\left(\frac{1}{jn}\right) Y_n = \left(\frac{32}{\pi}\right)\left(\frac{1}{jn}\right)\left(\frac{1-(-1)^n}{j2n\pi}\right)$$

$$Z_n = 16\left(\frac{(-1)^n - 1}{n^2\pi^2}\right)$$

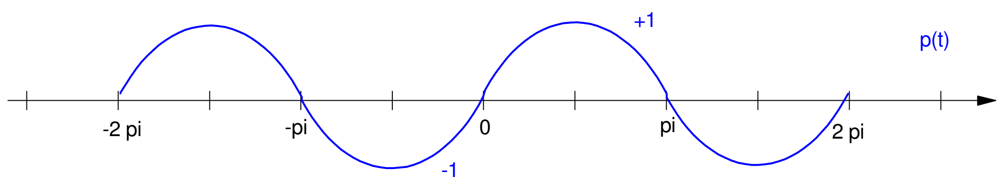
Checking in Matlab:

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

for n=1:20
   cn(n) = ((-1)^n - 1) / (n^2 * pi^2);
   end

x = 0*t;

for n=1:20
   x = x + 2*real(cn(n))*cos(n*t)  -  2*imag(cn(n))*sin(n*t);
   end

plot(t,x)
```



Sum of the first 20 terms of the Fourier series approximation to a triangle wave

Example 4: Parabolic Sine Wave: If you integrate a triangle wave, you get parabolas. Multiply by a constant to keep the peak-to-peak amplitude equal to one



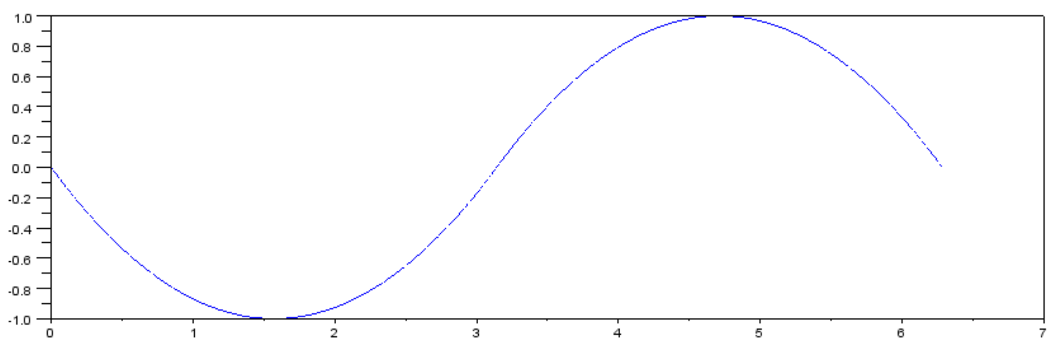
$$p(t) = \frac{1}{\pi} \int z(t) \cdot dt$$

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

$$P_n = \left(\frac{(-1)^n - 1}{j2n^3\pi^3}\right)$$

Checking in Matlab

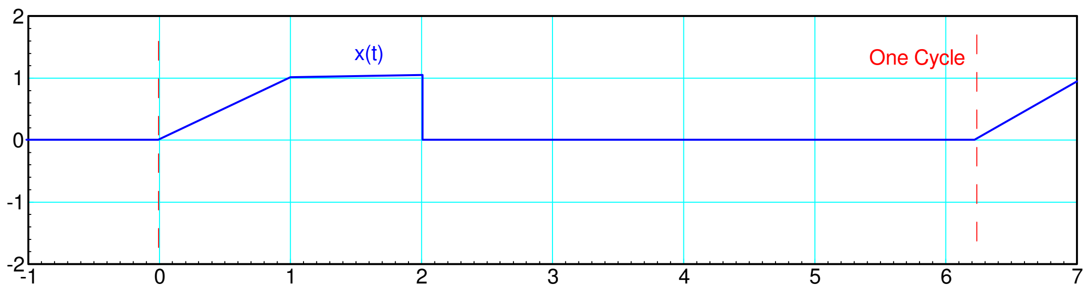
```
cn = zeros(20,1);
for n=1:20
   cn(n) = 16*((-1)^n - 1) / (2*j*n^3 * pi^3);
   end
x = 0*t;
for n=1:20
  x = x + 2*real(cn(n))*cos(n*t)  -  2*imag(cn(n))*sin(n*t);
  end
plot(t,x)
```



Parabolic Sine Wave

You can also find the Fourier transform for different functions with delays and differentiation.

Example 5: Find the Fourier Transform for the following function:



Solution: Start taking derivatives until you get delta functions. Delta functions are nice since they have a simple Fourier transform

$$\delta(t) \leftrightarrow \left(\frac{1}{2\pi}\right)$$

A delayed delta function is

$$\delta(t-T) \leftrightarrow \left(\frac{1}{2\pi}\right) e^{-jn\omega_0 T}$$

So... start taking derivatives. Note that taking a derivative is the same as multiplying the Fourier transform by $jn\omega_0$

$$\frac{dx}{dt} \leftrightarrow (jn\omega_0)X$$

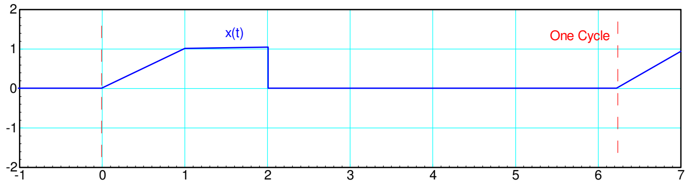
Integration (to get back to x(t)) is equivalent to dividing by

$$\int x \cdot dt \leftrightarrow \left(\frac{1}{jn\omega_0}\right)X$$

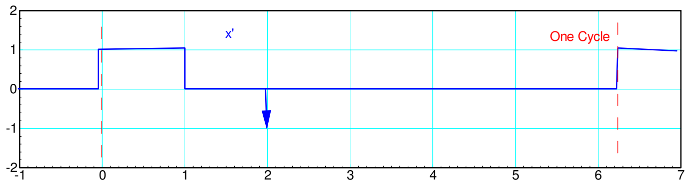
Also note that this function has a period of 2π. Hence

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

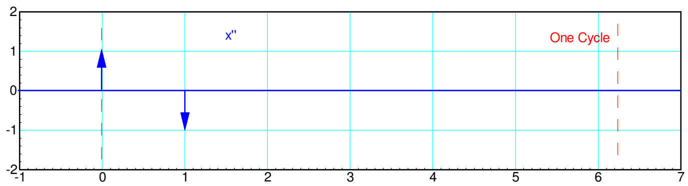
x(t) and its derivatives | Complex Fourier Transform of delta functions



$$X_0 = 0$$



$$X_1 = \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn\omega_0}\right)(-e^{-j2n\omega_0})$$



$$X_2 = \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn\omega_0}\right)^2(1-e^{-jn\omega_0})$$

This means the complex Fourier transform for x(t) is

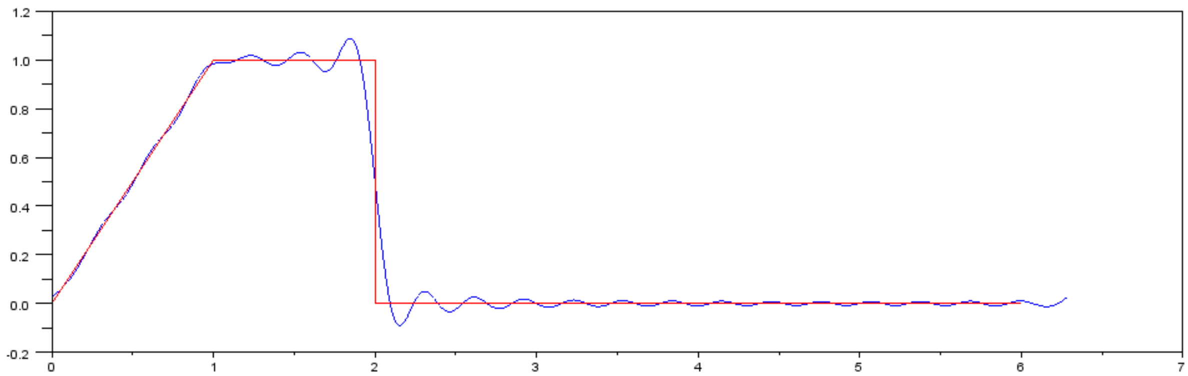
$$X = \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn\omega_0}\right)(-e^{-j2n\omega_0}) + \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn\omega_0}\right)^2(1-e^{-jn\omega_0})$$

or since $\omega_0 = 1$

$$\boxed{X = \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn}\right)(-e^{-j2n}) + \left(\frac{1}{2\pi}\right)\left(\frac{1}{jn}\right)^2(1-e^{-jn})}$$

Verifying in Matlab:

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

for n=1:20
  X(n) = (1/(2*pi)) * (1/(j*n)) * ( - exp(-j*2*n) );
  X(n) = X(n) + (1/(2*pi)) * (1/(j*n))^2 * ( 1 - exp(-j*n) );
  end

x = 0*t;

for n=1:20
  x = x + 2*real(X(n))*cos(n*t) - 2*imag(X(n))*sin(n*t);
  end

plot(t,x)
```


1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
1
2
3
4
5
6
7