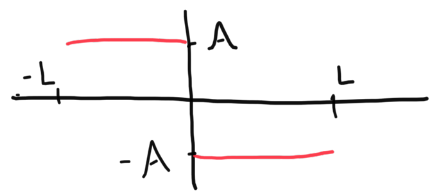
(1) In the lecture we saw that function,

$$f(x) = \begin{cases} A & \text{if } -L < x < 0, \\ -A & \text{if } 0 < x < L. \end{cases}$$

has a Fourier series

$$f(x) = \sum_{n=1}^{\infty} b_n \sin \frac{n\pi x}{L} = -\frac{4A}{\pi}\left[\sin\frac{\pi x}{L} + \frac{1}{3}\sin\frac{3\pi x}{L} + \frac{1}{5}\sin\frac{5\pi x}{L} + \cdots\right].$$

Write a `MatLab` program that will find the Fourier series in the interval $[-\pi, \pi]$. Use 5, 10, 30 terms in the sum. Plot both the original function and its Fourier series for each of these cases.



(2) `MatLab` has a built in integrator that works well for the kinds of intergals that arise in Fourier series. Lets begin with a simple case to see how this command works. You will be asked to integrate the function $f(x) = x^2$ between 0 and 1. The syntax of the `integral` command is

```
F = integral(fun,xmin,xmax)
```

where `fun` is defined using the anonymous function handle `@`. So for our example you have

```
fun = @(x) (x.^2)
```

Use `integral` to evaluate $f(x) = x^2$ between 0 and 1.

(3) Parameters can also be passed to the function `integral`. They are just added in the function definition, for example

```
fun = @(x,c) (c*x.^2)
```

Use `integral` to evaluate $f(x) = cx^2$ between 0 and 1. Set c various values, both positive and negative. You may need to read the help page on `integral` to get this work correctly.

(4) Finally, integrate the function $f(x) = cx^2 + b$ with arbitrary parameters a, b.

(5) Write a `MatLab` function to numerically evaluate the Fourier transform of the function

$$f(t) = \begin{cases} a(1 - a|t|), & |t| < \frac{1}{a} \\ \\ 0, & |t| > \frac{1}{a} \end{cases}$$

with $a = 10$. There are a few other parameters that are handy in the function `integral` Read the help page about `RelTol` and `AbsTol`. As we saw in the lecture, the Fourier transform of a function is

$$g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-i\omega t}$$

The Fourier transform is all well and good, but in physical applications, we usually don't have $f(t)$, but only a *sampling* of $f(t)$. That is, we we have measures of the function that occur at discrete points. Here we will assume that the data points are evenly sampled at intervals, Δt and that we had N such measurements. That is the measurements occur at $m\Delta t$ where $m = 0, 1, \ldots N-1$.

To arrive at the *discrete Fourier transform*, we do much as we did when derived the Fourier tranform. The basic steps are:

- start with the complex version of Fourier seris

  $$f(t) = \sum_{-\infty}^{\infty} c_n e^{in\pi t/T}$$

  with c_n now

  $$c_n = \frac{1}{T}\int_0^T f(t)e^{-in\pi t/T}$$

  What is the difference?

- Now recalling that

  $$\Delta\omega = \frac{\pi}{T},$$

  we approximate the Fourier transform as

  $$g(\omega) \approx g(n\Delta\omega) = \sum_0^{N-1} f(m\Delta t)e^{-in\Delta\omega m\Delta t} = \sum_0^{N-1} f(m\Delta t)e^{-i2\pi mn/N}$$

The inverse discrete Fourier transform is a bit involved to obtain, but it can be done using some orthogonality relations and partial sum relations. The result is that

$$f(m\Delta t) = \frac{1}{N}\sum_{n=0}^{N-1} g(n\Delta\omega)\ e^{i2\pi mn/N}.$$

There is relationship between the Fourier transform and the discrete Fourier transform. Denoting the Fourier transform by $\mathcal{F}$ we have that

$$\begin{aligned} \mathcal{F}[f(t)] &= \frac{\Delta t}{\sqrt{2\pi}} DFT[f(t)] \\ \mathcal{F}^{-1}[g(\omega)] &= \frac{\sqrt{2\pi}}{\Delta t} DFT^{-1}[g(\omega)] \end{aligned}$$

(7) Discuss in your group the differences, similarities, and when it is appropriate to use Fourier Series, the Fourier Transform, and the Discrete Fourier Transform.

(8) Find the *discrete Fourier transform* for the same function as in question (5). Use $N = 100$.