Fourier transform

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 23
Any periodic single value function

\[ y(t) = y(t + T) \]

with finite number of discontinues and for which \( \int_{0}^{T} |f(t)| \, dt \) is finite can be presented as

\[ y(t) = \frac{a_0}{2} + \sum_{1}^{\infty} (a_n \cos(n\omega_1 t) + b_n \sin(n\omega_1 t)) \]

\( T \) period

\( \omega_1 \) fundamental frequency \( \frac{2\pi}{T} \)

\[
\begin{pmatrix} a_n \\ b_n \end{pmatrix} = \frac{2}{T} \int_{0}^{T} dt \left( \begin{pmatrix} \cos(n\omega_1 t) \\ \sin(n\omega_1 t) \end{pmatrix} y(t) \right)
\]

At discontinuities series approach the mid point.
Fourier series example: $|t|$

$$y(t) = |t|, \quad -\pi < t < \pi$$

Since function is even all $b_n = 0$

$$\begin{align*}
a_0 &= \pi, \\
a_n &= 0, \quad n \text{ is even} \\
a_n &= -\frac{4}{\pi n^2}, \quad n \text{ is odd}
\end{align*}$$
Fourier series example: step function

\[
\begin{cases}
0, & -\pi < x < 0, \\
1, & 0 < x < \pi
\end{cases}
\]

Since function is odd all \( a_n = 0 \) except \( a_0 = 1 \)

\[
\begin{align*}
    b_n &= 0, & n \text{ is even} \\
    b_n &= \frac{2}{\pi n}, & n \text{ is odd}
\end{align*}
\]

Eugeniy Mikhailov (W&M)
Practical Computing
Lecture 23
Complex representation

Recall that

$$\exp(i\omega t) = \cos(\omega t) + i \sin(\omega t)$$

It can be shown that

$$y(t) = \sum_{n=-\infty}^{\infty} c_n \exp(in\omega_1 t)$$

$$c_n = \frac{1}{T} \int_{0}^{T} y(t) \exp(-i\omega_1 nt) dt$$

$$a_n = c_n + c_{-n}$$

$$b_n = i(c_n - c_{-n})$$
What to do if function is not periodic?

- $T \to \infty$
- $\sum \to \int$
- discrete spectrum $\rightarrow$ continuous spectrum
  - $c_n \rightarrow c_\omega$

\[
y(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} c_\omega \exp(i\omega t) d\omega
\]
\[
c_\omega = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} y(t) \exp(-i\omega t) dt
\]

Required: $\int_{-\infty}^{\infty} dt \ y(t)$ exist and finite
notice: rescaling of $c_\omega$ compared to $c_n$ by extra $\sqrt{2\pi}$ and $T$ is gone.
Discrete Fourier transform (DFT)

In reality we cannot have

- infinitively large interval
- infinite amount of points to calculate true integral

Assuming that $y(t)$ has a period $T$ and we took $N$ equidistant points

$$\Delta t = \frac{T}{N} \text{ samples spacing, } f_s = \frac{1}{\Delta t} \text{ sampling rate}$$

$$f_1 = \frac{1}{T} = \frac{1}{N \Delta t} \text{ smallest observed frequency, also resolution bandwidth}$$

$$t_k = \Delta t \times k$$

$$y(t_{k+N}) = y(t_k) \text{ periodicity condition}$$

$$y_k = y(t_k) \text{ shortcut notation}$$

$y_1, y_2, y_3, \cdots, y_N \text{ data set}$

We replace integral in Fourier series with the sum
\[ y_k = \frac{1}{N} \sum_{n=0}^{N-1} c_n \exp\left(i \frac{2\pi (k - 1) n}{N}\right) \quad \text{inverse Fourier transform} \]

\[ c_n = \sum_{k=1}^{N} y_k \exp\left(-i \frac{2\pi (k - 1) n}{N}\right) \quad \text{Fourier transform} \]

\[ n = 0, 1, 2, \ldots, N - 1 \]

Confusion keep increasing: where are the negative coefficients \( c_{-n} \)?

In DFT they moved to the right end of the \( c_n \) vector:

\[ c_{-n} = c_{N-n} \]
Fast Fourier transform (FFT)

Fast numerical realization of DFT is FFT. This is just smart way to do DFT. Matlab has one built in

- $y$ is a matlab vector of data points ($y_k$)
- $c = \text{fft}(y)$ Fourier transform
- $y = \text{ifft}(c)$ inverse Fourier transform

Notice that $\text{fft}$ does not normalize by $N$ so to get Fourier series $c_n$ you need to calculate $\frac{\text{fft}(y)}{N}$.

However $y = \text{ifft}(\text{fft}(y))$

Notice one more point of confusion: Matlab does not have index=0, so actual $c_n = \frac{c_{\text{matlab fft}}(n - 1)}{N}$, so $c_0 = \frac{c_{\text{matlab fft}}(1)}{N}$