Fourier series

Any periodic single value function

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

with finite number of discontinues and for which \( \int_{t} y(t) \, dt \) is finite can be presented as

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

\( T \) period

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

\( (a_n, b_n) = \frac{2}{T} \int_{0}^{T} dt \left( \cos(n\omega_1 t) \sin(n\omega_1 t) \right) y(t) \)

At discontinuities series approach the mid point

Fourier series example: \(|t|\)

\[ y(t) = \begin{cases} 0, & -\pi < t < 0, \\ 1, & 0 < t < \pi \end{cases} \]

Since function is even all \( b_n = 0 \)

\[ \begin{align*}
    a_0 &= \pi, \\
    a_n &= 0, & n \text{ is even} \\
    a_n &= -\frac{4}{n\pi}, & n \text{ is odd}
\end{align*} \]

Fourier series example: step function

\[ y(t) = \begin{cases} 0, & -\pi < x < 0, \\ 1, & 0 < x < \pi \end{cases} \]

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

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

Recall that

\[ \exp(i\omega t) = \cos(\omega t) + i\sin(\omega t) \]

It can be shown that

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

\[ c_n = \frac{1}{T} \int_{0}^{T} y(t) \exp(-in\omega_1 t) dt \]

\[ a_n = c_n + c_{-n} \]

\[ b_n = i(c_n - c_{-n}) \]

What to do if function is not periodic?

- \( T \to \infty \)
- \( \sum \to \int \)
- discrete spectrum \( \to \) continuous spectrum

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

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

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

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

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

\[ \Delta t = \frac{T}{N} \] samples spacing, \( f_s = \frac{1}{\Delta t} \) sampling rate

\[ f_1 = \frac{1}{T} = \frac{1}{N\Delta t} \] smallest observed frequency;
also resolution bandwidth

\[ f_k = \Delta t \times (k - 1) \]

\( y(t_k) \) periodicity condition

\( y_k \) \( y(t_k) \) shortcut notation

\( y_1, y_2, y_3, \ldots, y_N \) data set

We replace integral in Fourier series with the sum

DFT

\[ y_k = \frac{1}{N} \sum_{n=0}^{N-1} c_n \exp\left(\frac{2\pi i(k-1)n}{N}\right) \] \text{inverse Fourier transform} \n
\[ c_n = \sum_{k=1}^{N} y_k \exp\left(-\frac{2\pi i(k-1)n}{N}\right) \] \text{Fourier transform} \n
\[ n = 0, 1, 2, \ldots, N-1 \]

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 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 ($y_k$)
- $c=fft(y)$ Fourier transform
- $y=ifft(c)$ inverse Fourier transform

Notice that $fft$ does not normalize by $N$ so to get Fourier series $c_n$ you need to calculate $fft(y)/N$.

However $y = ifft(ifft(y))$

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