Numerical Methods

Fourier Series

Prof. Peter Clark

Fourier Series and Discrete Fourier Series

Fourier Series are a valuable tool for many applications.

They will help us understand some problems such as waves.

We shall be discussing two closely related series:

The Fourier Series – (virtually) any periodic function can be described using Fourier Series and many other functions can be made to look periodic.

The Fourier Series is **infinite.**

The Discrete Fourier Transform – This has a finite number of terms and deals with ‘sampled’ data. Thus, it has characteristics of a ‘numerical’ scheme.

Why Fourier Series

Fourier Series have very many applications.

One of the most useful features comes from the fact that Fourier Series are made up of terms like:

$$\exp(ikx) \quad \text{or} \quad \exp(i\omega t)$$

With $i = \sqrt{-1}$. The first derivative of this is given by:

$$\frac{d}{dx}\exp(ikx) = ik\exp(ikx) \quad \text{or} \quad \frac{d}{dt}\exp(i\omega t) = i\omega\exp(i\omega t)$$

so differentiation turns into multiplication.

This is extremely useful for solving some types of PDE and for analysing solution methods for more general problems.

The Fourier Series

Fourier showed that any periodic function with period T can be written:

$$f(t) = \sum_{j=-\infty}^{\infty} c_j \exp(ij\omega_0 t) \qquad \omega_0 = \frac{2\pi}{T}$$

A note on notation; many authors like to separate sines and cosines, and have a constant term:

$$f(t) = a_0 + \sum_{j=1}^{\infty} a_j \cos(j\omega_0 t) + \sum_{j=1}^{\infty} b_j \sin(j\omega_0 t)$$

Note the limits!
Using complex exponential makes life MUCH easier and I' ve never found a reason not to, so we shall do so here but note where the sums start – my '$j=0$' is the constant term, since $\exp(0) = 1$, and so:

$$a_j = c_j + c_{-j}$$
$$b_j = i(c_j - c_{-j})$$
$$j > 0$$

Orthogonality

Given two 'harmonics', j and k, we can write

$$\int_0^T \exp(ij\omega_0 t)\exp(-ik\omega_0 t)dt = \int_0^T \exp(i(j-k)\omega_0 t)dt$$

$$= \left[\frac{1}{i(j-k)\omega_0}\exp(i(j-k)\omega_0 t)\right]_0^T$$

$$= \frac{1}{i(j-k)\omega_0}\left[\exp\left(i(j-k)\frac{2\pi}{T}T\right)-1\right]$$

$$= \frac{1}{i(j-k)\omega_0}\left[\cos((j-k)2\pi)-1+i\sin((j+k)2\pi)\right]$$

$$= 0 \quad \text{if } j \neq k$$

If $j = k$, we can write:

$$\int_0^T \exp(ij\omega_0 t)\exp(-ij\omega_0 t)dt = \int_0^T 1dt = T$$

In general

$$\boxed{\int_0^T \exp(ij\omega_0 t)\exp(-ik\omega_0 t)dt = T\delta_{jk}}$$

$$\begin{aligned}\delta_{jk} &= 1; \quad j = k \\ &= 0; \quad j \neq k\end{aligned}$$

The Fourier Series

If: $$f(t) = \sum_{j=-\infty}^{\infty} c_j \exp(ij\omega_0 t) \qquad \omega_0 = \frac{2\pi}{T}$$

and: $$\int_0^T \exp(ij\omega_0 t)\exp(-ik\omega_0 t)dt = T\delta_{jk}$$

Then: $$\int_0^T f(t)\exp(-ik\omega_0 t)dt = \sum_{j=-\infty}^{\infty} c_j \int_0^T \exp(ij\omega_0 t)\exp(-ik\omega_0 t)dt$$

$$= \sum_{j=-\infty}^{\infty} c_j T\delta_{jk}$$

$$= c_k T$$

Thus: $$c_k = \frac{1}{T}\int_0^T f(t)\exp(-ik\omega_0 t)dt$$

Solving Differential Equations using Fourier Series

Suppose $$\frac{d^2y}{dx^2} = \rho(x)$$

If: $$\rho(x) = \sum_{j=-\infty}^{\infty} \rho_j \exp(ijk_0x) \qquad k_0 = \frac{2\pi}{L}$$

with $$\rho_j = \frac{1}{L}\int_0^L \rho(x)\exp(-ijk_0x)dx$$

Let $$y(x) = \sum_{j=-\infty}^{\infty} c_j \exp(ijk_0x)$$

Then $$\frac{d^2y}{dx^2} = \sum_{j=-\infty}^{\infty} c_j \frac{d^2}{dx^2}\exp(ijk_0x) = -\sum_{\substack{j=-\infty \\ j\neq o}}^{\infty} c_j j^2k_0^2 \exp(ijk_0x) = \sum_{\substack{j=-\infty \\ j\neq o}}^{\infty} \rho_j \exp(ijk_0x)$$

$$\Rightarrow c_j = -\frac{\rho_j}{j^2k_0^2}$$

Solving Differential Equations using Fourier Series

Thus
$$\frac{d^2y}{dx^2} = \rho(x)$$

Gives us
$$y(x) = \sum_{\substack{j=-\infty \\ j\neq 0}}^{\infty} \frac{\rho_j}{j^2 k_0^2} \exp(ijk_0 x)$$

The higher frequency harmonics in the load are 'low-pass filtered' – frequencies in the load $j2\pi/L$ appear in y with amplitude reduced by $1/j^2k_0{}^2$.

(This is a general property of the Laplacian operator – it acts as a smoother)

The Discrete Fourier Transform

Suppose we have a set of N uniformly spaced data $\{y_l,\ l=0,1 \ldots N\text{-}1\}$.
The Inverse Discrete Fourier Transform is given by

$$y_l = \sum_{j=0}^{N-1} Y_j \exp(ij\omega_0 l) \qquad \omega_0 = \frac{2\pi}{N}$$

Note that this can be written:

$$y_l = \sum_{j=0}^{N-1} Y_j W^{lj}$$

Where $W = \exp(i\omega_0)$ and the superscript is a power.

Orthogonality

Given two 'harmonics', j and k, we can write

$$\sum_{l=0}^{N-1} \exp(ij\omega_0 l)\exp(-ik\omega_0 l) = \sum_{l=0}^{N-1} W^{(j-k)l} = \sum_{l=0}^{N-1} \left(W^{(j-k)}\right)^l$$

Now, in general, suppose:

$$E = \sum_{l=0}^{N-1} A^l = 1 + A + A^2 + \ldots + A^{N-1}$$

Multiply by A:

$$AE = \sum_{l=0}^{N-1} A^{l+1} = A + A^2 + \ldots + A^{N-1} + A^N$$

Subtract:

$$E - AE = 1 - A^N \Rightarrow E = \frac{1 - A^N}{1 - A}$$

Thus

$$\sum_{l=0}^{N-1} \left(W^{j-k}\right)^l = \frac{1 - W^{(j-k)N}}{1 - W^{(j-k)}} = \frac{1 - \exp(-i(j-k)2\pi)}{1 - \exp(-i(j-k)\,2\pi/N)} = 0 \quad \text{if } j \neq k$$

If $j = k$, we can write:

$$\sum_{l=0}^{N-1} \exp(ij\omega_0 l)\exp(-ij\omega_0 l) = \sum_{l=0}^{N-1} 1 = N$$

In general

$$\boxed{\sum_{l=0}^{N-1} \exp(ij\omega_0 l)\exp(-ik\omega_0 l) = N\delta_{jk}}$$

The Discrete Fourier Transform

If: $$y_l = \sum_{j=0}^{N-1} Y_j \exp(ij\omega_0 l) \qquad \omega_0 = \frac{2\pi}{N}$$

and: $$\sum_{l=0}^{N-1} \exp(ij\omega_0 l)\exp(-ik\omega_0 l) = N\delta_{jk}$$

Then:

$$\sum_{l=0}^{N-1} y_l \exp(-ik\omega_0 l) = \sum_{l=0}^{N-1}\sum_{j=0}^{N-1} Y_j \exp(ij\omega_0 l)\exp(-ik\omega_0 l)$$

$$= \sum_{j=0}^{N-1} Y_j N\delta_{jk}$$

$$= Y_k N$$

Thus: $$Y_k = \frac{1}{N}\sum_{l=0}^{N-1} y_l \exp(-ik\omega_0 l)$$

This is the forward Discrete Fourier Transform (DFT)

Fourier Series vs DFT

	Fourier Series	Discrete Fourier Transform
Inverse Transform	$f(t) = \sum_{j=-\infty}^{\infty} c_j \exp(ij\omega_0 t)$	$y_l = \sum_{j=0}^{N-1} Y_j \exp(ij\omega_0 l)$
Fundamental Frequency	$\omega_0 = \frac{2\pi}{T}$	$\omega_0 = \frac{2\pi}{N}$
Orthogonality Condition	$\int_0^T \exp(ij\omega_0 t)\exp(-ik\omega_0 t)dt = T\delta_{jk}$	$\sum_{l=0}^{N-1} \exp(ij\omega_0 l)\exp(-ik\omega_0 l) = N\delta_{jk}$
Forward Transform	$c_k = \frac{1}{T}\int_0^T f(t)\exp(-ik\omega_0 t)dt$	$Y_k = \frac{1}{N}\sum_{l=0}^{N-1} y_l \exp(-ik\omega_0 l)$

Fourier Series vs DFT

The DFT very closely resembles the Fourier Series with integrals replaced by sums.

Let us suppose out time period T is split into N intervals each of width $\Delta t = T/N$

The Fourier Series coefficients can be calculated by

$$c_k = \frac{1}{T}\int_0^T f(t)\exp\left(-ik\frac{2\pi}{T}t\right)dt = \sum_{l=0}^{N-1}\int_{l\Delta t}^{(l+1)\Delta t} f(t)\exp\left(-ik\frac{2\pi}{T}t\right)dt$$

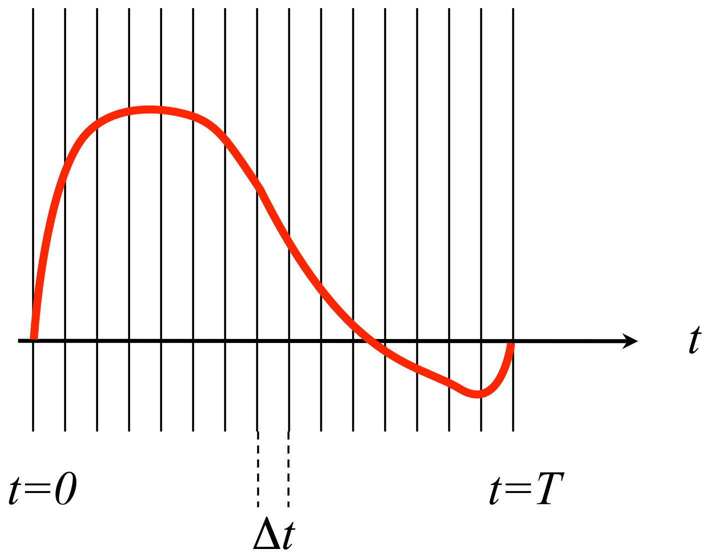
Approximate the integral over the time step

$$\int_{l\Delta t}^{(l+1)\Delta t} f(t)\exp\left(-ik\frac{2\pi}{T}t\right)dt \approx f(l\Delta t)\exp\left(-ik\frac{2\pi}{T}l\Delta t\right)\Delta t$$

$$= f_l \exp\left(-ik\frac{2\pi}{N}l\right)\frac{T}{N}$$

Thus:

$$c_k \approx \frac{1}{N}\sum_{l=0}^{N-1} f_l \exp\left(-ik\frac{2\pi}{N}l\right)$$ i.e. the DFT



Fourier Series vs DFT

What about the inverse transforms? They look rather different.

$$f(t) = \sum_{j=-\infty}^{\infty} c_j \exp\left(ij\frac{2\pi}{T}t\right) \qquad \text{vs} \qquad y_l = \sum_{j=0}^{N-1} Y_j \exp\left(ij\frac{2\pi}{N}l\right)$$

Substitute $t = l\Delta t = l\dfrac{T}{N} \Rightarrow l = \dfrac{N}{T}t$

$$f(t) = \sum_{j=-\infty}^{\infty} c_j \exp\left(ij\frac{2\pi}{T}t\right) \qquad \text{vs} \qquad y_l = \sum_{j=0}^{N-1} Y_j \exp\left(ij\frac{2\pi}{T}t\right)$$

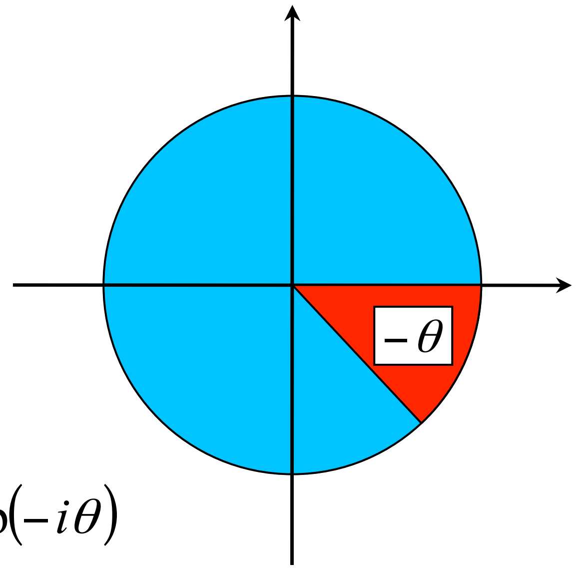
What does negative frequency mean?

First, it is easiest to think of the sums above as sums over angular frequencies which are a fraction of 2π, i.e.

$$\omega_j = \frac{j}{N}2\pi$$

Fourier Series vs DFT

What does negative frequency mean?



Compare $\exp(i\theta)$ with $\exp(-i\theta)$

$$\exp(i\theta) = \cos(\theta) + i\sin(\theta)$$

$$\exp(-i\theta) = \cos(-\theta) + i\sin(-\theta) = \cos(\theta) - i\sin(\theta)$$

Also:

$$\exp(i(2\pi - \theta)) = \cos(2\pi - \theta) + i\sin(2\pi - \theta) = \cos(\theta) - i\sin(\theta) = \exp(-i\theta)$$

Thus, we can say any frequency $\omega_j = \dfrac{j}{N}2\pi$ greater than π is equivalent to a negative frequency

$$\omega'_j = -\left(2\pi - \frac{j}{N}2\pi\right) = -\frac{N-j}{N}2\pi = -\frac{j'}{N}2\pi$$

If N is even we can write

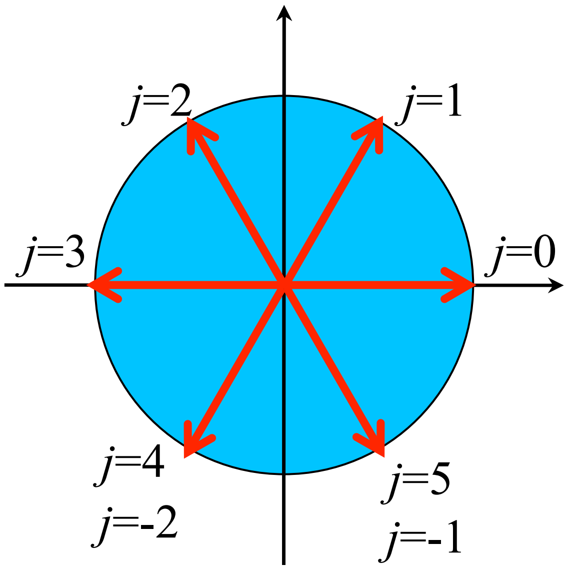
$$y_l = \sum_{j=0}^{N-1} Y_j \exp\left(ij\frac{2\pi}{T}t\right) \equiv \sum_{j=-N/2+1}^{N/2} Y_j \exp\left(ij\frac{2\pi}{T}t\right)$$

Fourier Series vs DFT

If N is even we can write

$$y_l = \sum_{j=0}^{N-1} Y_j \exp\left(ij \frac{2\pi}{T} t \right) \equiv \sum_{j=-N/2+1}^{N/2} Y_j \exp\left(ij \frac{2\pi}{T} t \right)$$

e.g. N=6

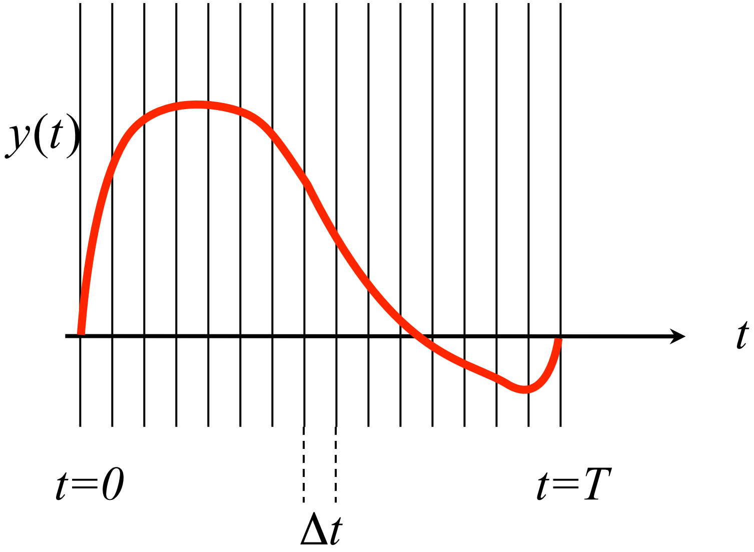


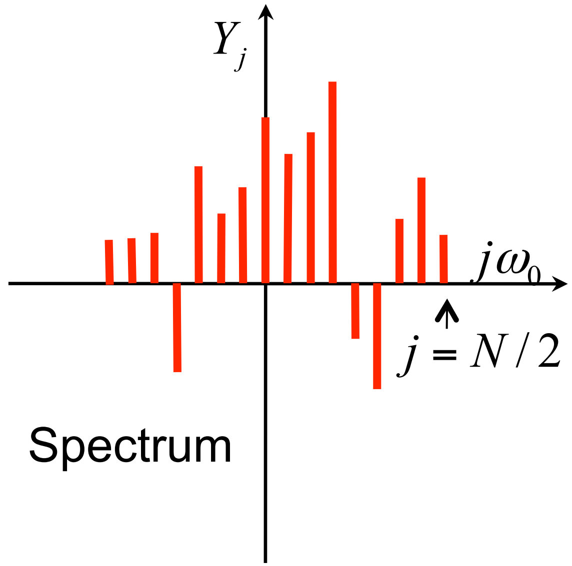
Thus, the DFT is like a truncation of the Fourier Series, including angular frequencies up to $\frac{N}{2}\frac{2\pi}{T} = \frac{1}{2}\frac{2\pi}{\Delta t}$. Note that $\frac{2\pi}{\Delta t}$ is the angular sampling frequency.

This is essentially Shannon's Sampling Theorem – the maximum frequency represented is half the sampling frequency.

For example, supposing T = 1 second, and we sample 44100 times (i.e. 44.1 kHz), then the maximum frequency we can include is half this, i.e. 22.050 kHz.

Fourier Series vs DFT





Suppose $y(t)$ actually contains frequencies $> N\omega_0/2$?

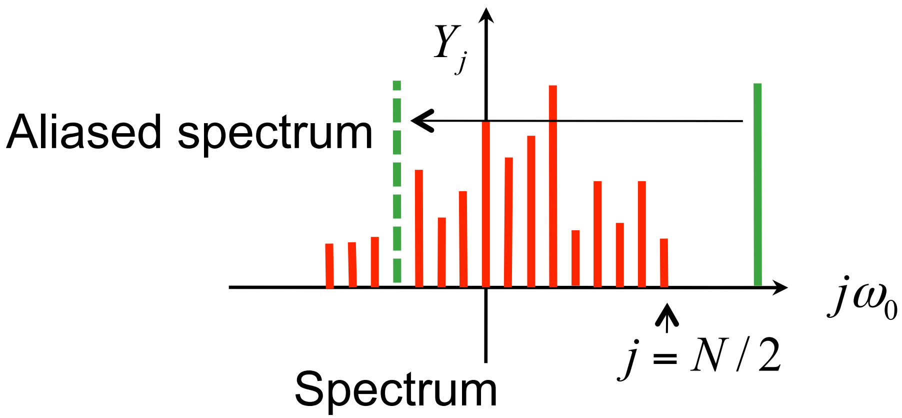
$$y_l = A\exp(im\omega_0 l) \quad m > N/2$$

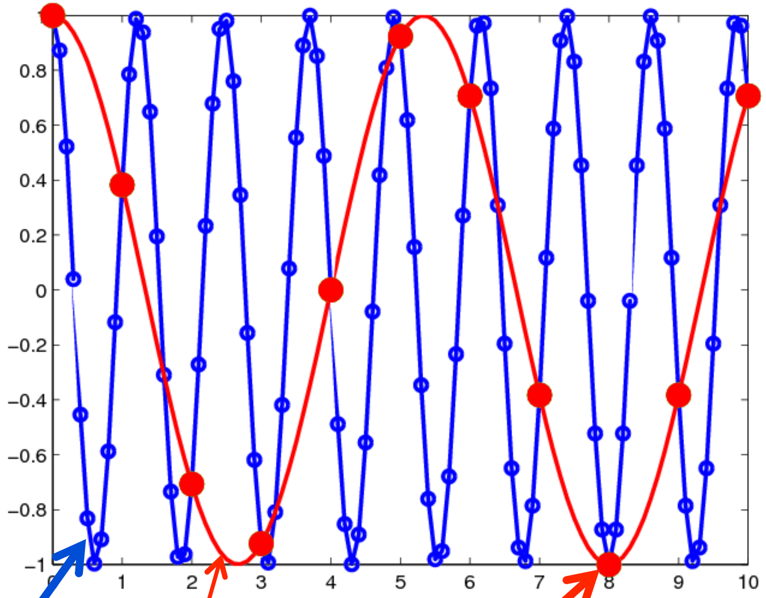
$$Y_k = \frac{1}{N}\sum_{l=0}^{N-1} y_l \exp(-ik\omega_0 l) = \frac{1}{N}\sum_{l=0}^{N-1} A\exp(im\omega_0 l)\exp(-ik\omega_0 l) = \frac{1}{N}\sum_{l=0}^{N-1} A\exp(i(m-k)\omega_0 l)$$

Y_k is non-zero whenever $(m-k)\omega_0 = 2\pi$, i.e. $(m-k)\dfrac{2\pi}{N} = 2\pi \Rightarrow k = m - N$

Aliasing

So if $y(t)$ actually contains frequencies $m\omega_0 = (N/2 + j)\omega_0$ it produces an apparent amplitude at the negative frequency $-m'\omega_0 = -(N/2 - j)\omega_0 = -(N - m)\omega_0$.





$$y = \cos\left(13\frac{2\pi}{16}t\right)$$

Sampled with $\Delta t = 0.1$

Sampled with $\Delta t = 1$

$$y = \cos\left(3\frac{2\pi}{16}t\right)$$

Apparent frequency=16-13=3

It is essential that frequencies higher than $N\omega_0/2$ are removed before the signal is sampled.

The FFT

The DFT needs a lot of computation as N increases – proportional to N^2.

A number of algorithms exist which reduce this to roughly $N\log_2 N$.

We shall not cover the details here but they are called 'Fast Fourier Transforms' of FFTs for short. Matlab has an FFT function.

Beware notation! There is an enormous variety. In particular:

1. Choice of summation limits (Matlab counts from 1 and uses (j-1), for obvious reasons).
2. Normalization. Where does the 1/N go?
3. Sign.

Always read the documentation!!!!!