Physics Coding Club

The Fast Fourier Transform

Phil Hasnip
phil.hasnip@york.ac.uk

Fourier Series

Any ‘sensible’ 1D periodic function with period L may be expressed as a Fourier series:

$$f(x) = \sum_{G=0}^{\infty} \left[a_G \sin\left(\frac{2\pi i G x}{L}\right) + b_G \cos\left(\frac{2\pi i G x}{L}\right)\right]$$

where i is the imaginary number, G is an integer and a_G and b_n are coefficients (real, for real functions).

Often convenient to rewrite as complex exponentials:

$$f(x) = \sum_{G=-\infty}^{\infty} c_G e^{\frac{2\pi i G x}{L}}$$

where c_G are the *Fourier coefficients* (usually complex).

Derivatives of periodic functions

The derivatives of Fourier series are also Fourier series:

$$
\begin{aligned}
f(x) &= \sum_{G=-\infty}^{\infty} c_G e^{\frac{2\pi i G x}{L}} \\
\Rightarrow \frac{df}{dx} &= \sum_{G=-\infty}^{\infty} \frac{2\pi i G}{L} c_G e^{\frac{i\pi G x}{L}} \\
\frac{d^2 f}{dx^2} &= \sum_{G=-\infty}^{\infty} \left(-\frac{4\pi^2 G^2}{L^2}\right) c_G e^{\frac{2\pi i G x}{L}}
\end{aligned}
$$

So what? Differential equations

$$\frac{d^2 f}{dx^2} + f = 0$$

becomes

$$\sum_{G=-\infty}^{\infty} \left[\left(-\frac{4\pi^2 G^2}{L^2} \right) + 1 \right] c_G e^{\frac{i\pi G x}{L}} = 0$$

i.e.

$$\begin{aligned} G^2 &= \frac{L^2}{4\pi^2} \\ \Rightarrow G &= \pm\frac{L}{2\pi} \end{aligned}$$

or $c_G = 0$.

Fourier transform

For a given function $f(x)$, we can find any particular c_m using:

$$c_m = \frac{1}{2\pi} \int_{-L}^{L} f(x) e^{-\frac{i\pi m x}{L}} dx$$

These coefficients are often called the *Fourier transform* of $f(x)$, and written as $\tilde{f}_G$ or F_G.

The Discrete Fourier transform

Suppose $L = 1$ and our periodic function is sampled on a regular grid with N points. Our sampled function is a discrete function and we have:

$$
\begin{aligned}
f_x &= \frac{1}{N}\sum_{G=0}^{N-1} F_G e^{2\pi i G x} \\
F_G &= \sum_{x=0}^{N-1} f_x e^{-2\pi i G x} dx
\end{aligned}
$$

So what? Convolutions

Convolutions happen a lot in science.

$$p_x = (f * h)_x = \sum_{x'} f_{x'} h_{x-x'}$$

Sum over N points for each x value, so need N^2 operations in total.

But the Fourier transform $\tilde{p}_G$ is:

$$P_G = F_G H_G$$

Only need 1 multiplication for each G value, so need N operations in total.

So if we have the Fourier transforms of f and h, we can quickly compute the Fourier transform of p...

FFT in practice

This only works if we can do the Fourier transform fast than N^2...

Example: 2-point discrete transform

Sample function f_x in real-space at 2 points. Want to Fourier transform to get F_G the 2 Fourier components.

$$
\begin{aligned}
F_1 &= f_1 + f_2 \\
F_2 &= f_1 - f_2
\end{aligned}
$$

Example: 4-point discrete transform

Sample function f_x in real-space at 2 points. Want to Fourier transform to get F_G the 2 Fourier components.

$$
\begin{aligned}
F_1 &= f_1 + f_2 + f_3 + f_4 \\
F_2 &= f_1 + if_2 - f_3 - if_4 \\
F_3 &= f_1 - f_2 + f_3 - f_4 \\
F_4 &= f_1 - if_2 - f_3 + if_4
\end{aligned}
$$

This looks like N^2 work, but if we gather terms...

Example: 4-point discrete transform

Sample function f_x in real-space at 2 points. Want to Fourier transform to get F_G the 2 Fourier components.

$$
\begin{aligned}
F_1 &= f_1 + f_2 + f_3 + f_4 \\
&= (f_1 + f_3) + (f_2 + f_4) \\
F_2 &= f_1 + if_2 - f_3 - if_4 \\
&= (f_1 - f_3) + i(f_2 - f_4) \\
F_3 &= f_1 - f_2 + f_3 - f_4 \\
&= (f_1 + f_3) - (f_2 + f_4) \\
F_4 &= f_1 - if_2 - f_3 + if_4 \\
&= (f_1 - f_3) - i(f_2 + f_4)
\end{aligned}
$$

It's a double 2 point Fourier transform!

FFT in practice

A naïve N-point Fourier transform scales as N^2, but using this divide-and-conquer style we can perform it in $\sim N \log_2 N$ time instead.

This is why it is called the *Fast* Fourier Transform (FFT)!

There are many different algorithms to do this, but they are all ‘fast’ in this sense.

What can we do with these fast Fourier transforms?

Example use: long multiplication

Pictures

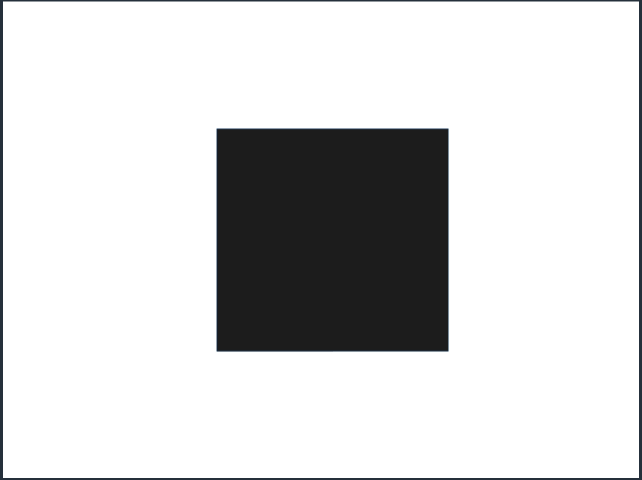
Most images aren't periodic, but we only care about a finite region.

We can define the region to be repeated through all space – in fact it's convenient to reflect about the Cartesian axes to make it an even function.

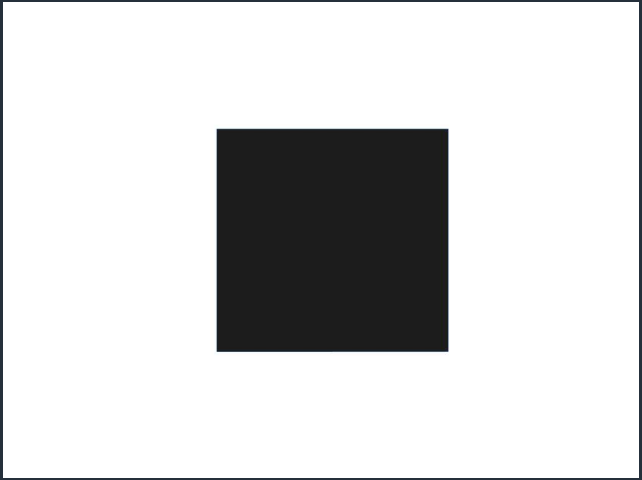
Our eyes are often insensitive to small, high-frequency changes...

- We can ignore these coefficients
- Truncate Fourier expansion
- Image compression!

Square with 100% of Fourier components



Square with 50% of Fourier components



Fun with images

'Marilyn Einstein' © Dr Aude Oliver, MIT