The Fourier Transform

CS/CME/BIOPHYS/BMI 279
Fall 2015
Ron Dror

The Fourier transform is a mathematical method that expresses a function as the sum of sinusoidal functions (sine waves). Fourier transforms are widely used in many fields of sciences and engineering, including image processing, quantum mechanics, crystallography, geoscience, etc.

Here we will use, as examples, functions with finite, discrete domains (i.e., functions defined at a finite number of regularly spaced points), as typically encountered in computational problems. However the concept of Fourier transform can be readily applied to functions with infinite or continuous domains. (We won't differentiate between "Fourier series" and "Fourier transform.")

A graphical example

Suppose we have a function $f(x)$ defined in the range $-\frac{L}{2} < x < \frac{L}{2}$ on $2N + 1$ discrete points such that $x_n = \frac{n}{2N+1} L$. By a Fourier transform we aim to express it as:

$$f(x_n) = a_0 + a_1 \cos(2\pi k_1 x + \phi_1) + a_2 \cos(2\pi k_2 x + \phi_2) + \cdots \quad (1)$$

We first demonstrate graphically how a function may be described by its Fourier components. Fig. 1 shows a function defined on 101 discrete data points integer $x = -50, -49, \ldots, 49, 50$. In fig. 2, the first

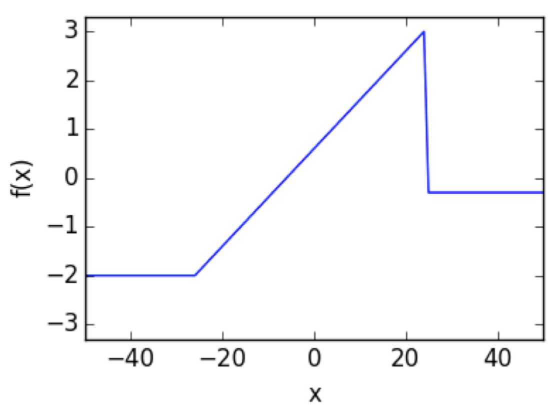


Fig. 1. The function to be Fourier transformed

few Fourier components are plotted separately (left) and added together (right) to form an approximation to the original function. Finally, fig. 3 demonstrates that by including the components up to $m = 50$, a faithful representation of the original function can be obtained.

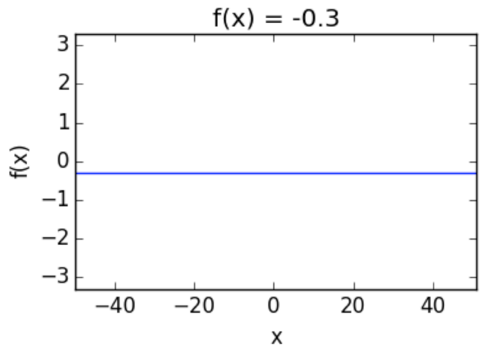
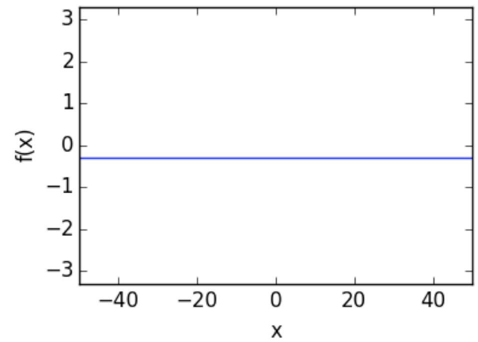
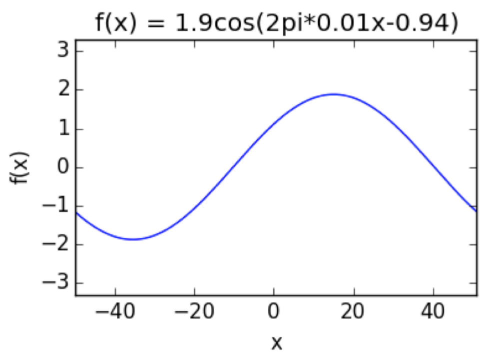
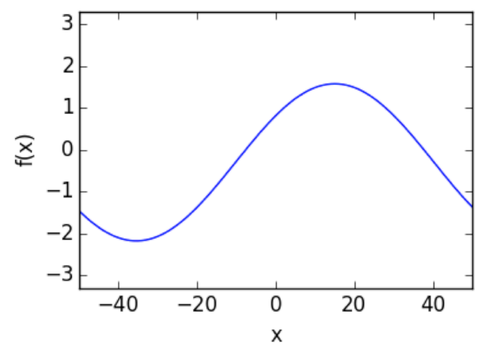
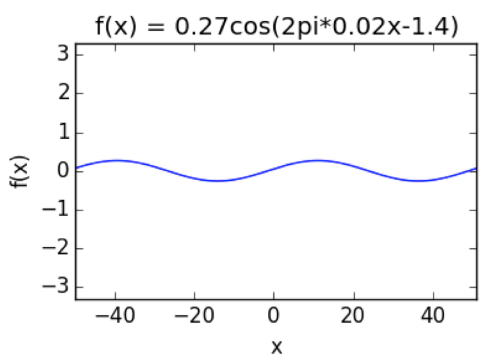
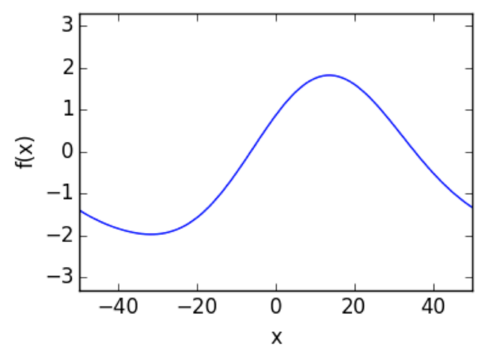
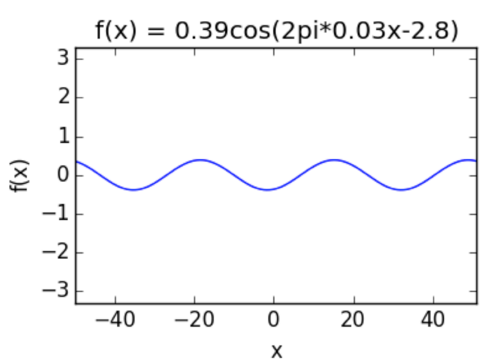
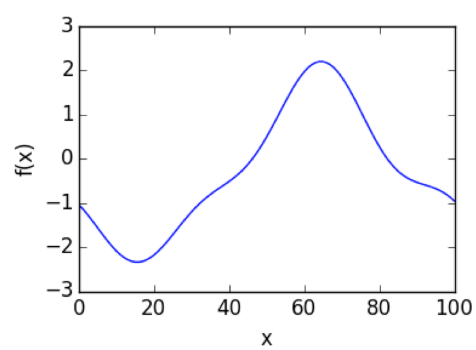
m	a_m, k_m & ϕ_m	Single components	Summing over components up to m
0	$a_m = 0.6$ $k_m = 0$ $\phi_m = 0$		
1	$a_m = 1.9$ $k_m = 0.01$ $\phi_m = 2.2$		
2	$a_m = 0.27$ $k_m = 0.02$ $\phi_m = -1.3$		
3	$a_m = 0.39$ $k_m = 0.03$ $\phi_m = 0.4$		

Fig.2. Single Fourier components (left) and combined inverse Fourier transform up to the components (right)

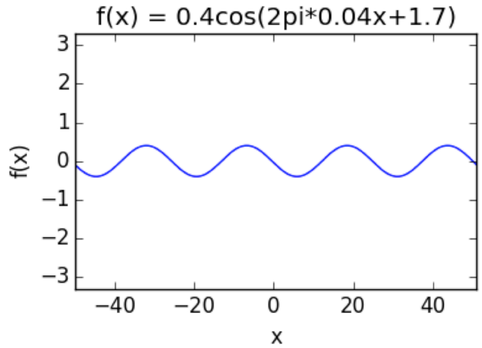
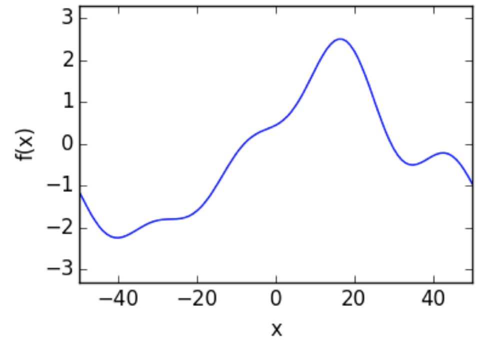
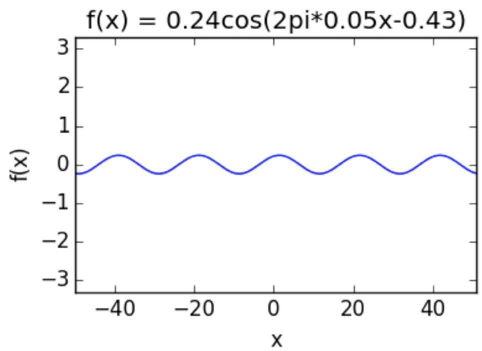
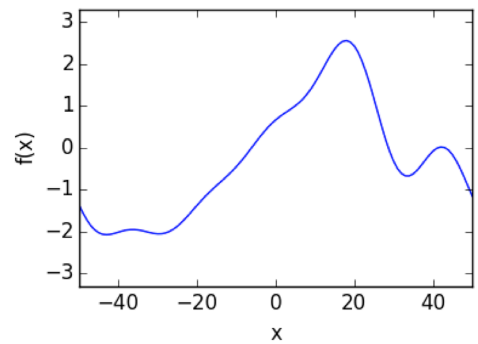
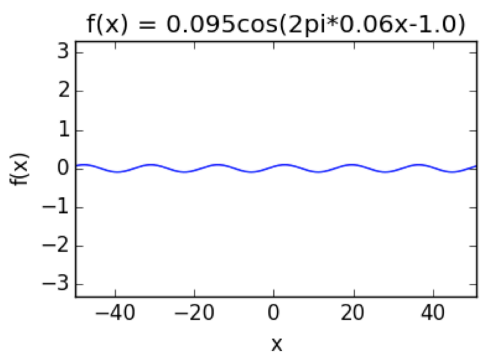
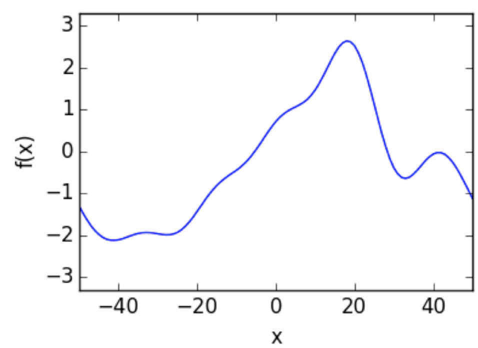
m	a_m, k_m & ϕ_m	Single component	Summing over up to m
4	$a_m = 0.4$ $k_m = 0.04$ $\phi_m = 1.8$		
5	$a_m = 0.24$ $k_m = 0.05$ $\phi_m = 2.9$		
6	$a_m = 0.095$ $k_m = 0.06$ $\phi_m = -0.83$		

Fig.2 (cont.). Single Fourier components (left) and sums of the first *m* components (right)

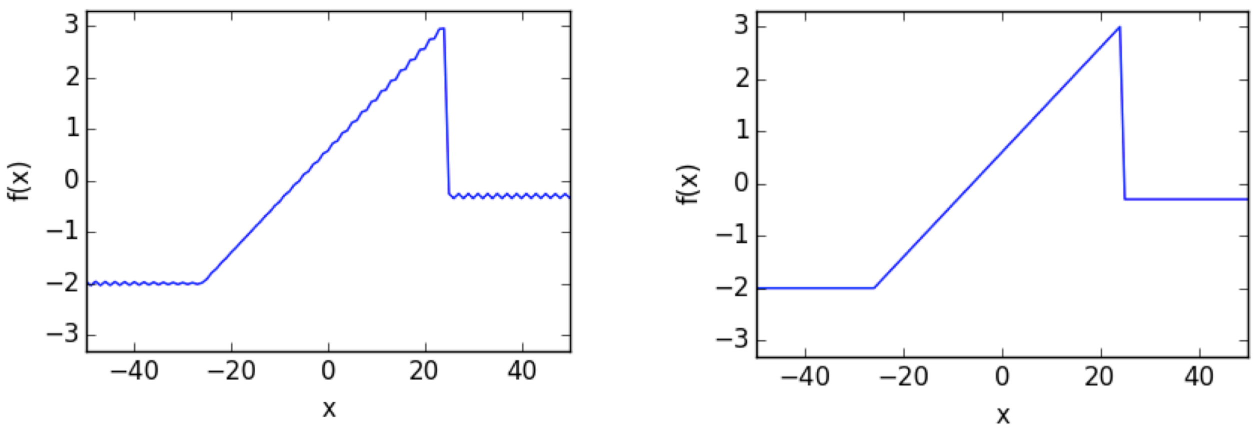


Fig. 3. Summing the Fourier components up to *m*=49 (left) and to *m*=50 (right)

Mathematical Formulae (you are not responsible for these)

More often you will see equation (1) in its more concise form with complex number notation:

$$f(x_n) = \frac{1}{2N+1} \sum_{m=-N}^{N} c(k_m) e^{2\pi i k_m x_n} \tag{2}$$

Note that $e^{i\theta} \equiv \cos(\theta) + i \cdot \sin(\theta)$. Here the numbers $c(k)$ are in general complex and the phases ϕ_m in eqn (1) are absorbed into $c(k)$. (We can think of a complex number as having real and imaginary parts, or as having a magnitude and an angle; the magnitude and angle can be used to specify the magnitude and phase of each sinusoidal component, respectively.

It turns out that we need k_m to be:

$$k_m = \frac{m}{L} \tag{3}$$

The values $c(k)$ can be calculated according to the following formula:

$$c(k_m) = \sum_{n=-N}^{N} f(x_n)\, e^{-2\pi i k_m x_n} \tag{4}$$

You may have noticed the formula 4 (the Fourier transform) is very similar to formula 2 (the inverse Fourier transform). The Fourier transform maps a function from **real space** to **Fourier (of frequency) space**; that is, it gives the phase and magnitude for each sinusoid. The inverse Fourier transform maps the other way, but is so similar that it can be computed using the same code.

As an example, sound is by nature mechanical movement as a function of time, but we often talk about sound in terms of frequencies. In this case we are effectively describing the Fourier transform of the sound wave.

Formulae (2) and (4) suggest that an algorithm for calculating a Fourier transform or its inverse transform from first principles would require an amount of computation proportional to N^2. In practice, a class of algorithms known as the "Fast Fourier Transform" (FFT) substantially reduce the computational requirements, so that the required amount of computation is proportional to $NlogN$.

Fourier Transforms in Multiple Dimensions

In multiple dimensions, instead of a single number k, each Fourier component is a **plane wave** and is characterized by a vector $\vec{k} = (k_x, k_y, \dots)$ instead of a single number. For example in two-dimensions, the plane wave $f(x, y) = \cos(2\pi \cdot 0.02x + 2\pi \cdot 0.01y)$ is characterized by $\vec{k} = (0.02, 0.01)$ and is

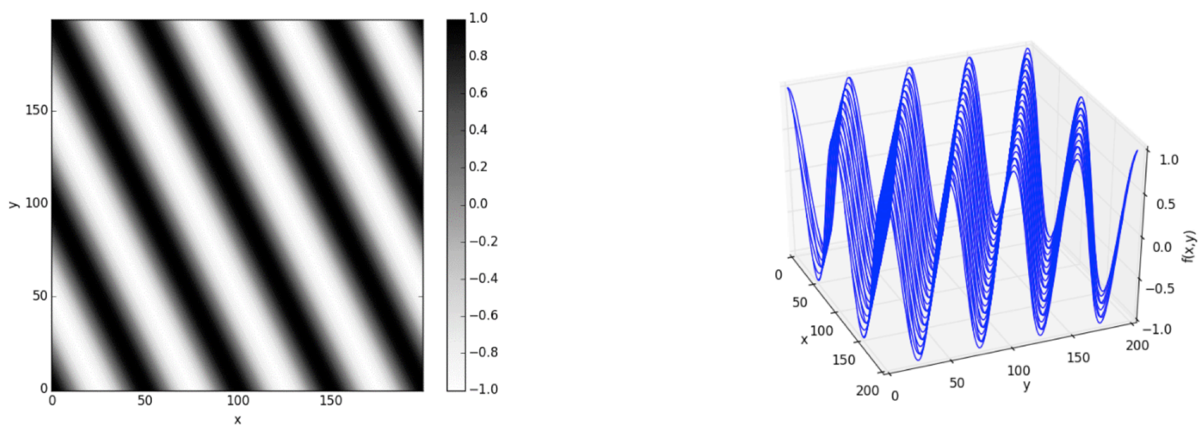


Fig. 4. A plane wave with wave-vector $\vec{k} = (0.02, 0.01)$ visualized as a color-map (left) and as a 3-D plot (right)

visualized in fig. 4.

The formulae for the Fourier transform in multiple dimensions need to incorporate all the dimensions. For example, in 2-D:

$$f(x_m, y_n) = \frac{1}{N^2} \sum_{\alpha,\beta=-N}^{N} c(k_\alpha, k_\beta) e^{2\pi i (k_\alpha x_m + k_\beta y_n)} \quad (5)$$