Numerical Computation of Fourier Transform

Xuan Liu

This article is on numerical computation of (continuous) Fourier transform of a function on $[-L/2, L/2]$ using the fast Fourier transform (FFT) in NumPy. Let $a = \{a_n : n = 0 \dots, N-1\}$ be an array. The FFT of a implemented in NumPy is the array defined by

$$\mathrm{F}[a](k) = \sum_{n=0}^{N-1} a_n e^{-\mathrm{i}2\pi kn/N}, \quad k = 0, \dots N-1.$$

This is an effective numerical scheme of the (scaled) Fourier transform of a function on $[0, 1]$. More precisely, let a be a function on $[0, 1]$ and $a_n = a(n/N)$. Then

$$\hat{a}(k) \approx N^{-1}\mathrm{F}[a](k), \quad k = 0, \dots, N-1, \tag{0.1}$$

which follows easily from

$$\hat{a}(k) = \int_0^1 a(x)e^{-\mathrm{i}2\pi kx}dx \approx N^{-1}\sum_{n=0}^{N-1} a_n e^{-\mathrm{i}2\pi kn/N}.$$

Now consider a function f defined on $[-L/2, L/2]$. Let δ_L be the dilation operator $\delta_L f(x) = f(Lx)$, and $\tau_{1/2}$ be the translation operator $\tau_{1/2}f(x) = f(x - 1/2)$. Then,

$$\begin{aligned}(e^{\mathrm{i}2\pi\lfloor N/2\rfloor x}\tau_{1/2}\delta_L f)^\wedge(k) &= (\tau_{1/2}\delta_L f)^\wedge(k - \lfloor N/2\rfloor) \\ &= (-1)^{k-\lfloor N/2\rfloor}L^{-1}\hat{f}[L^{-1}(k - \lfloor N/2\rfloor)].\end{aligned} \tag{0.2}$$

Note that $e^{\mathrm{i}2\pi\lfloor N/2\rfloor x}(\tau_{1/2}\delta_L f)(x)$ is defined on $[0, 1]$, and its Fourier transform can be computed using FFT. Therefore, by (0.1) and (0.2),

$$\hat{f}(\xi_k) \approx (-1)^{L\xi_k}LN^{-1}\mathrm{F}[e^{\mathrm{i}2\pi\lfloor N/2\rfloor x}\tau_{1/2}\delta_L f](k), \quad k = 0, \dots N-1, \tag{0.3}$$

where $\xi_k = L^{-1}(k - \lfloor N/2\rfloor)$. This gives an approximation of $\hat{f}$ on $[-L'/2, L'/2]$, where $L' = N/L$ and N is the number of sampling points in both the space and the frequency domains.

A few remarks are in demand. Firstly, the sampling distance in (0.3) in the frequency domain is $\Delta\xi = 1/L$. Therefore, to obtain a finer grid in the frequency domain, a larger integration domain $[-L/2, L/2]$in the space domain is needed. Secondly, the sampling distance for integration in the space domain is $1/L'$. Therefore, to obtain a higher integration precision, a wider output range $[-L'/2, L'/2]$ in the frequency domain is needed. Lastly, it is more natural in practice to specify L and L', and sample $N = LL'$ points from f.