ONLY FOR PYTHON
For a given vector, v, of length $2N$, call np.fft.fft and store the coefficients in a vector V, so V=np.fft.fft(v). MOST importantly the fourier coefficients, $\hat{v}_k$,

$$v = \sum_{k=-(N-1)}^{N-1} \hat{v}_k e^{2\pi i k x} = \frac{a_0}{2} + \sum_{1}^{N-1} a_k cos(2\pi k x) + b_k sin(2\pi k x) \tag{1}$$

The fourier coefficients are the ordered as follows

$$V_0 = \hat{v}_0 \tag{2}$$

$$V_k = \hat{v}_k, k \in (1, N-1) \tag{3}$$

$$V_k = \hat{v}_{k-2N}, k \in (N, 2N-1) \tag{4}$$

If one were to do spectral differentiation then the coefficients $\hat{w}_k = ik\hat{v}_k$ can be obtained as follows :

$$\hat{w}_k = ikV_k, k \in (0, N-1) \tag{5}$$

$$\hat{w}_{k-2N} = i(k-2N)V_k, k \in (N, 2N-1) \tag{6}$$

if nothing works copy this:
ik=1j*np.hstack((np.arange(0,N),np.arange(-N,0)))*2*pi/(L)