Discrete Fourier Transform

Larry Doolittle, LBNL

USPAS Houston, January 25, 2023

Lawrence Berkeley National Lab

Start

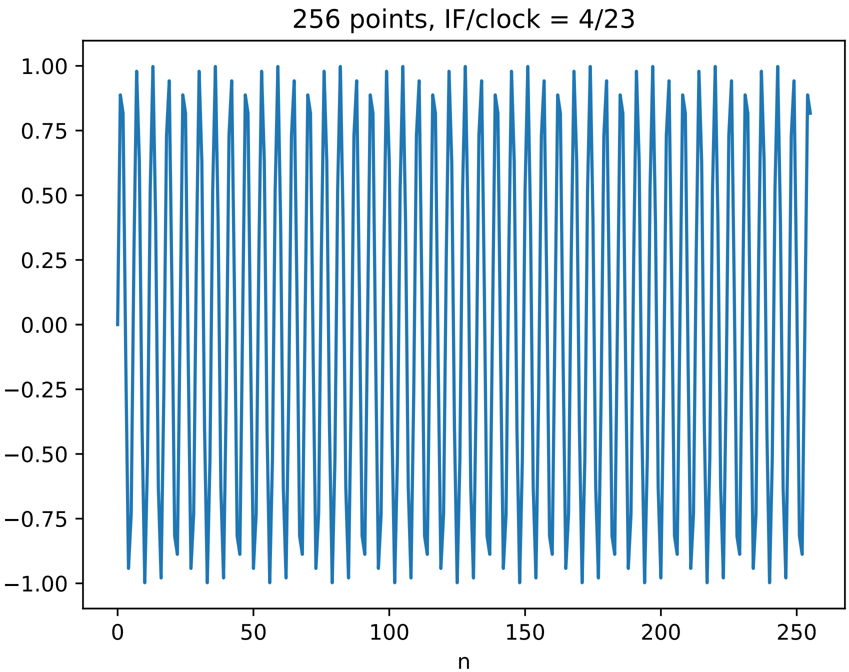
$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{2\pi j}{N} kn}$$

where x_n is the input time series of N complex numbers, and X_k are the N output frequency bins of complex numbers.

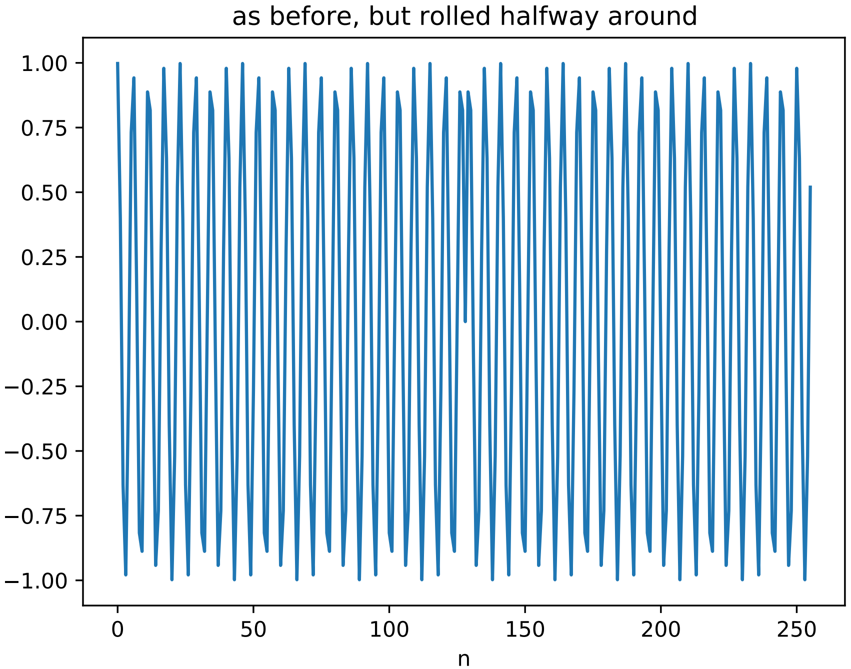
The very nature of the analysis assumes that the N input points represent one period of a periodic signal.

In many cases, the input waveform is a slice of non-periodic data. These slides explain how that shows up in the spectrum, and a couple of ways to mitigate problems.

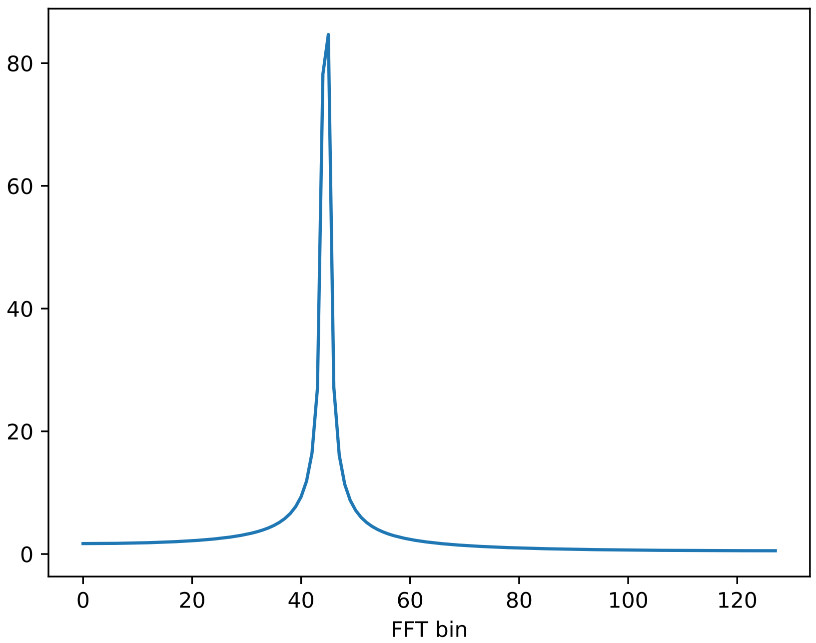
Example waveform



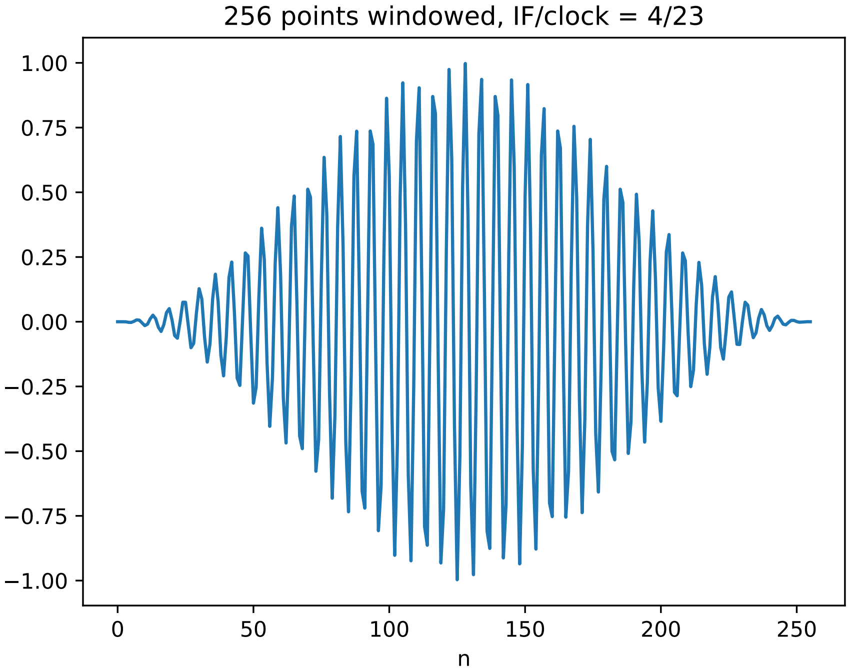
Example waveform



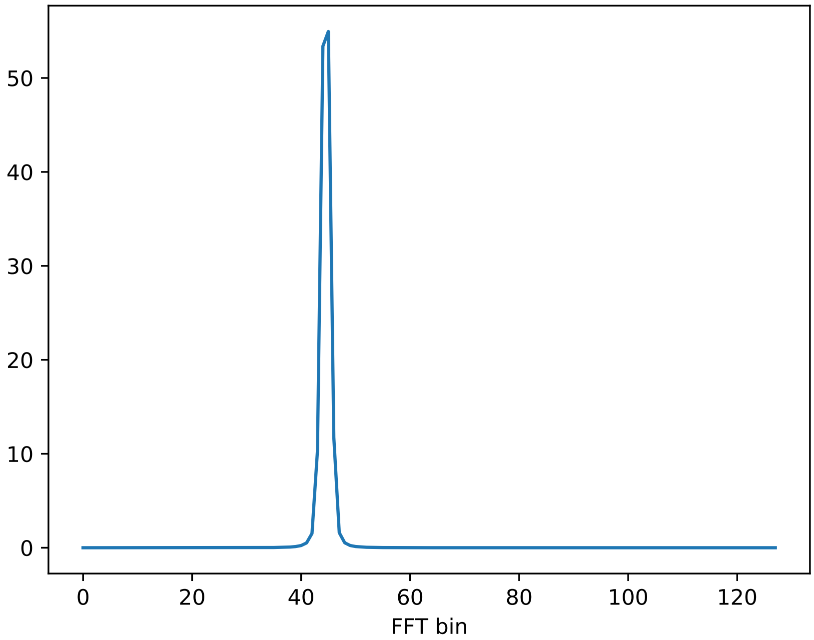
Example spectrum



Example waveform



Example waveform

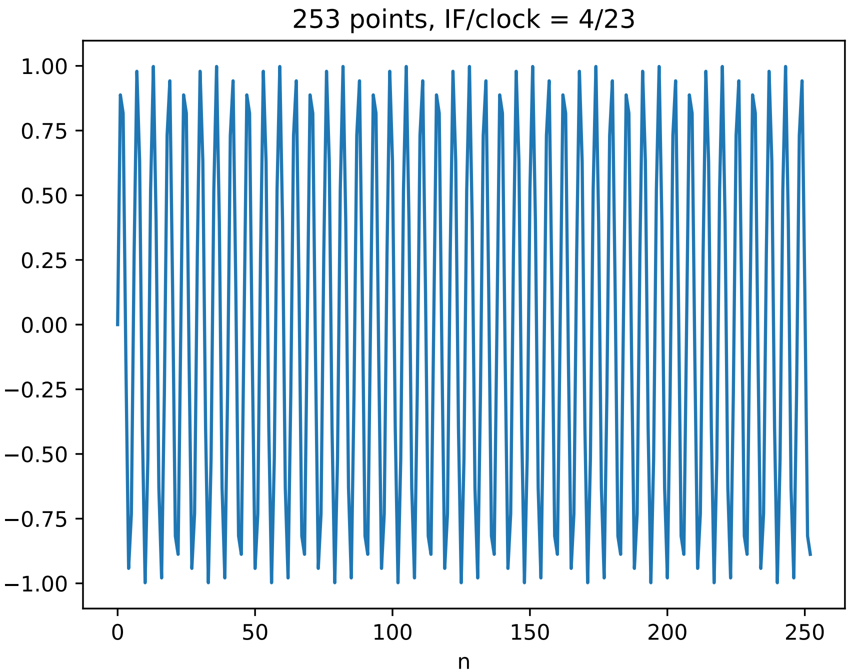


Periodicity

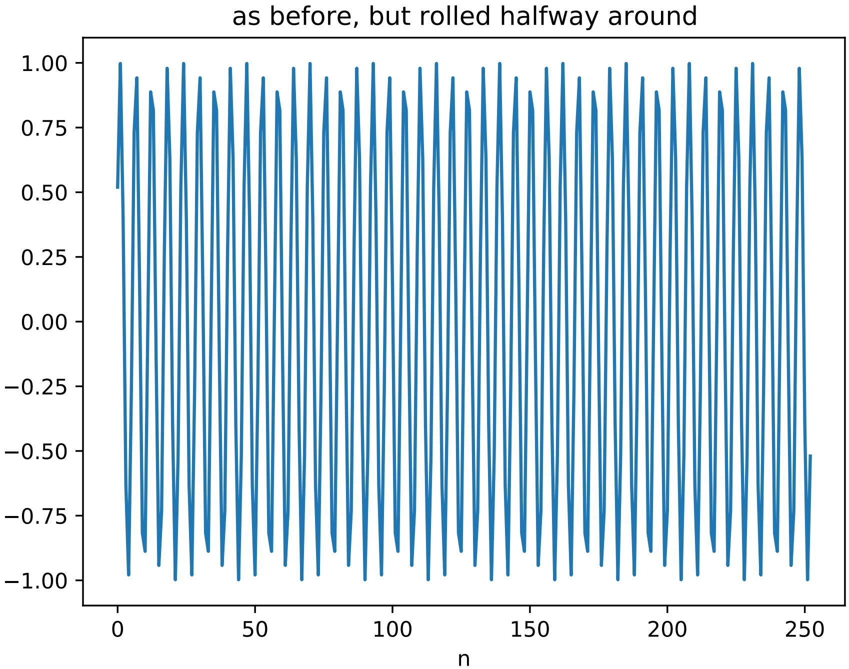
$$256/23 \approx 11.13$$

$$23 * 11 = 253$$

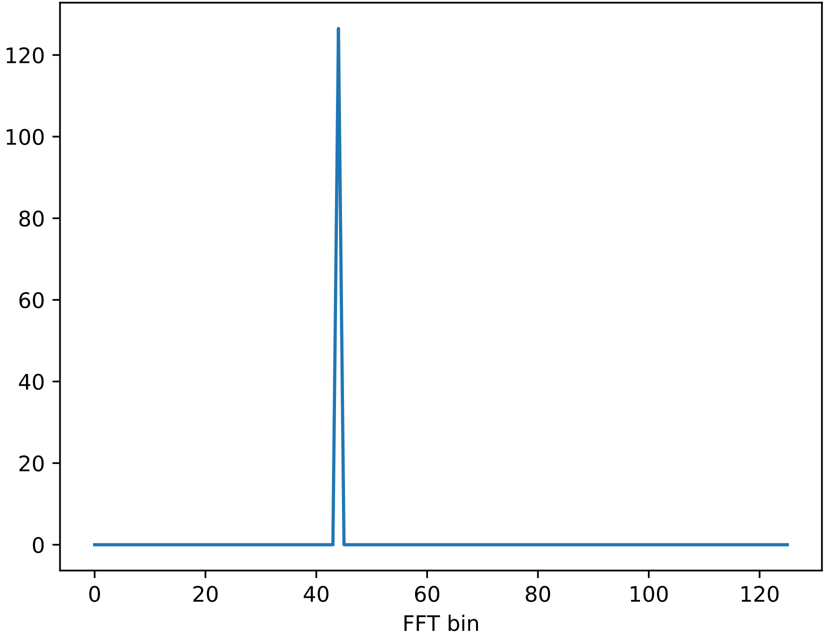
Example waveform



Example waveform



Example spectrum



Diversion: How fast is a Fast Fourier Transform?

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{2\pi j}{N} kn}$$

has n^2 multiplications as written.

FFTs come from tricky factorization of that $N \times N$ matrix. Computation effort usually written as $N \log_2 N$ when N is a power of two, That's a special case: if N has a prime factorization such that

$$N = \prod_k p_k$$

generalized FFT techniques can compute the N-point using about

$$N \cdot \sum_k p_k$$

multiplications. Examples:

- 256-point DFT: 256*256 multiplies
- 256-point FFT: 256*(2+2+2+2+2+2+2+2) = 256*16 multiplies
- 253-point FFT: 256*(23+11) = 256*34 multiplies

Some, but not all, modern DFT software takes advantage of this.