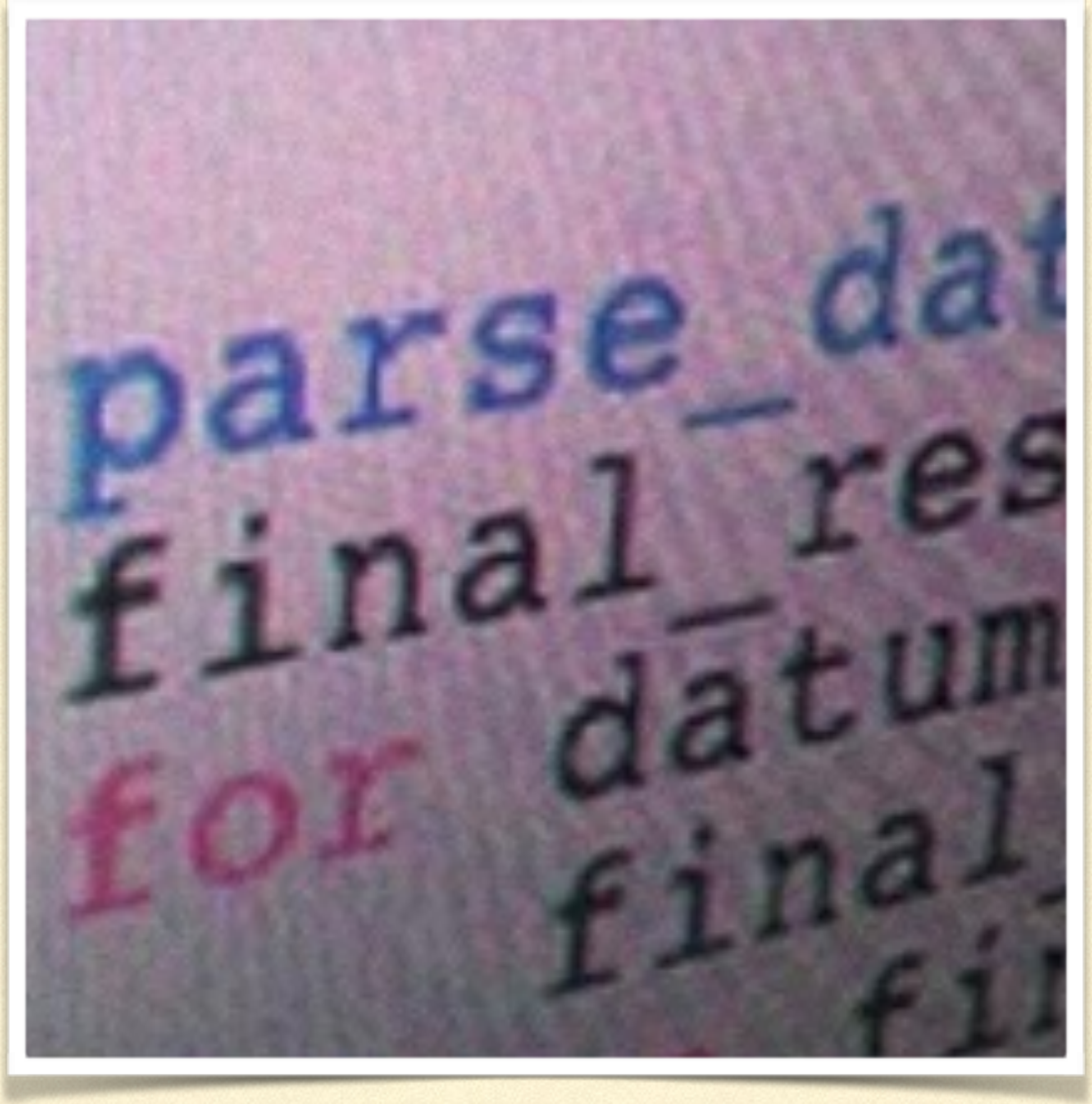
ASTROINFORMÁTICA I

AULA 08

Prof. Dr. Luciano Silva

luciano.silva@mackenzie.br

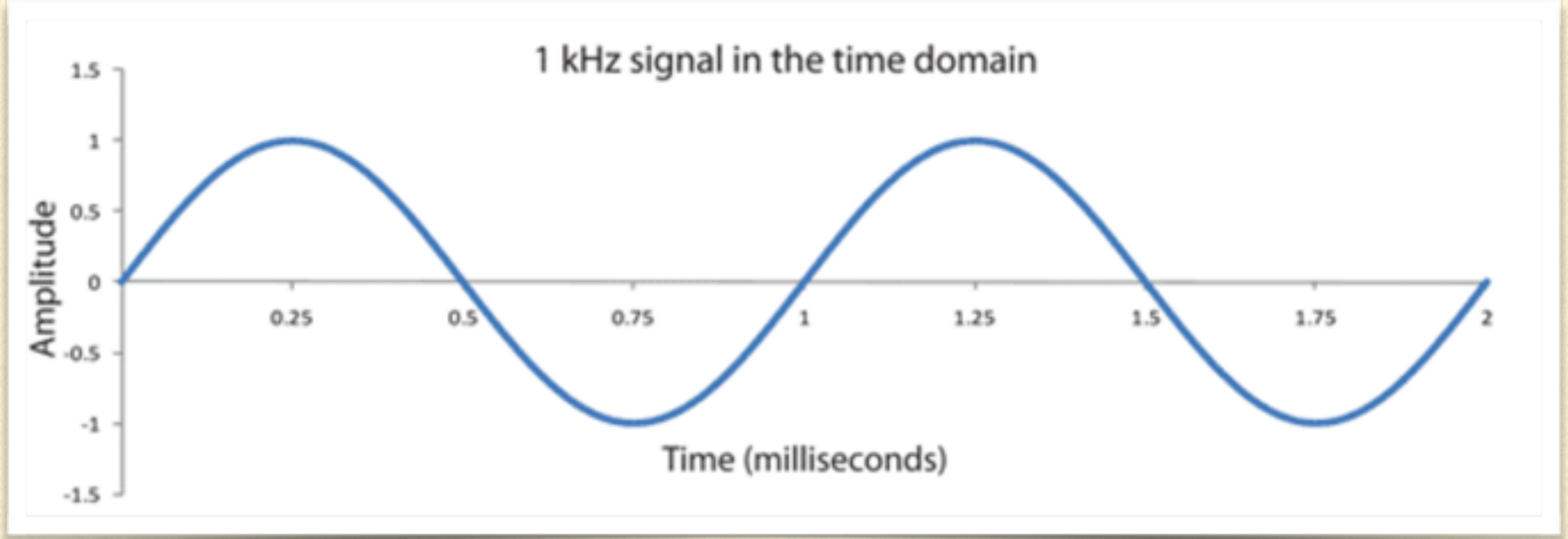
OBJETIVOS

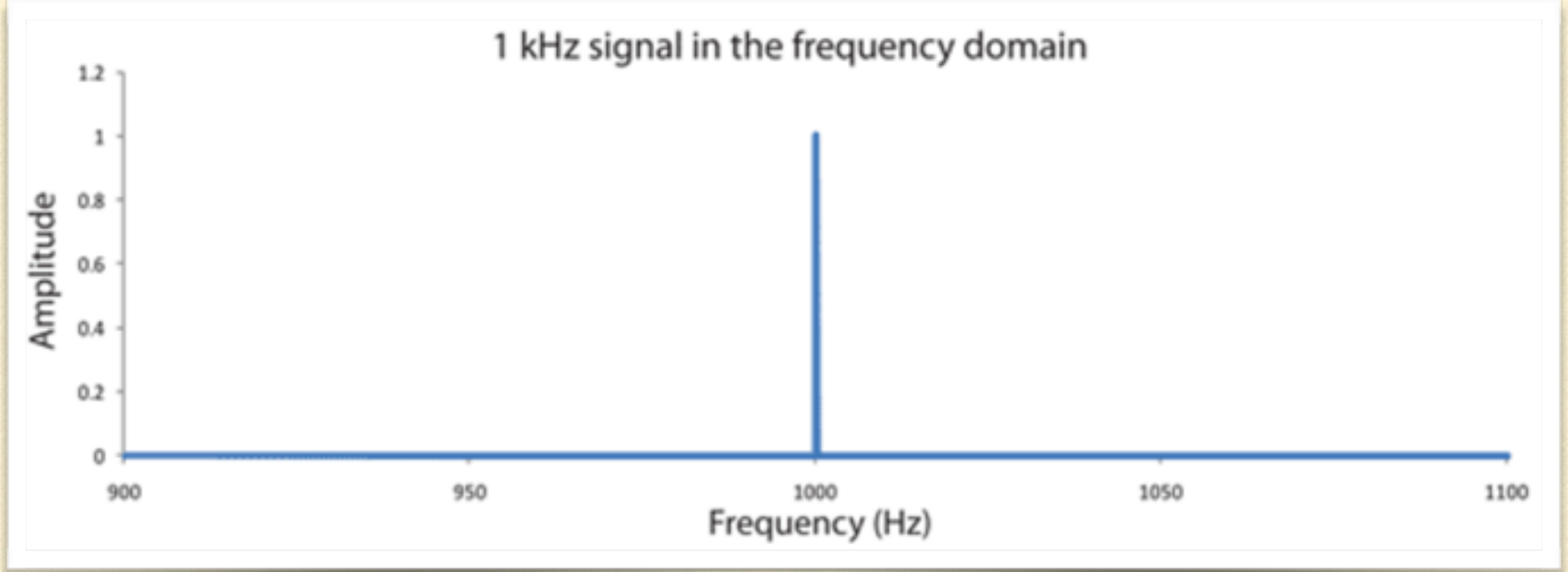


- Conhecer o processo de transformação de domínios de dados unidimensionais

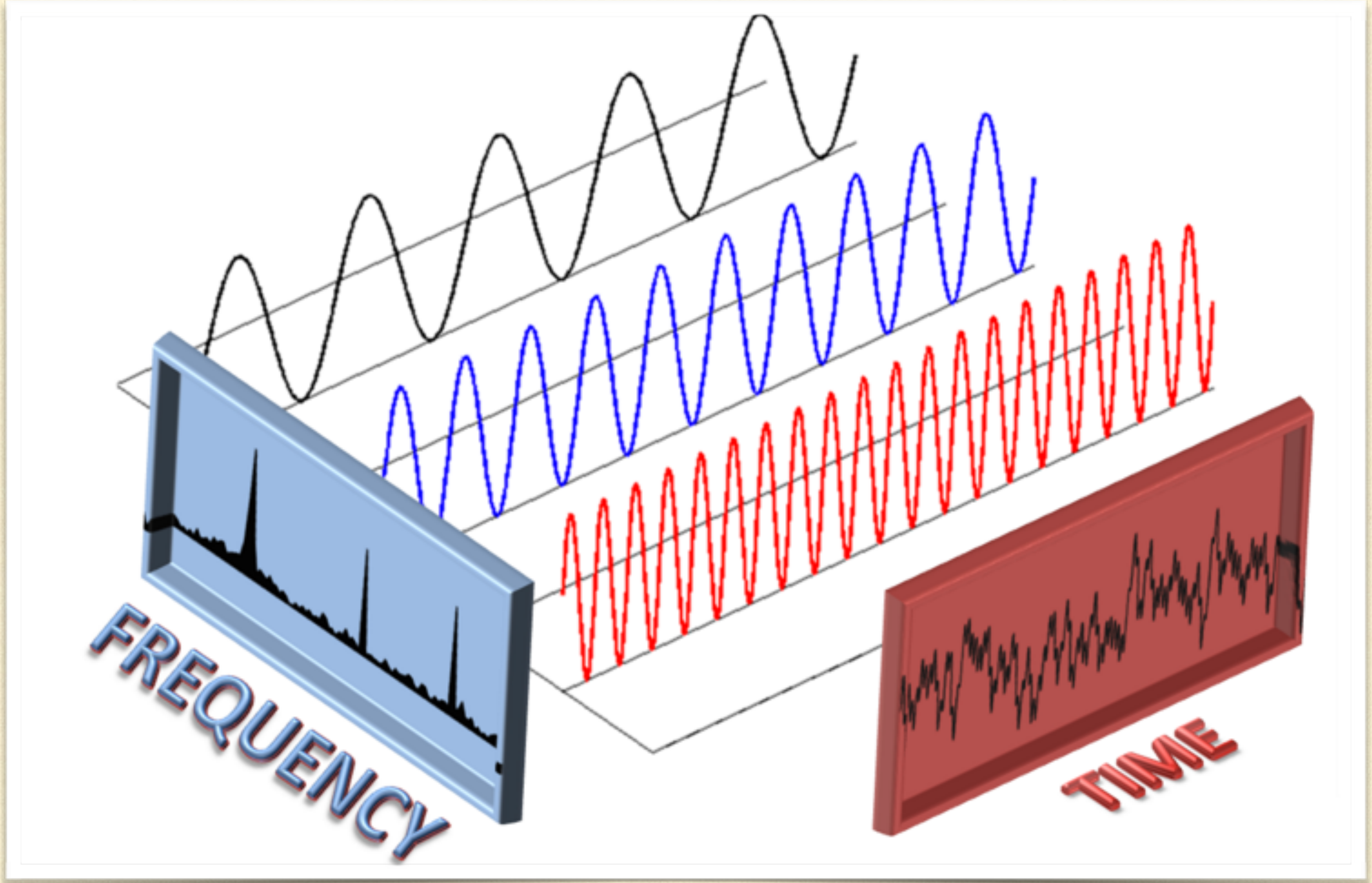
- Integrar o processo de transformação de domínios com leitura de tabelas de arquivos FITS e visualização de dados

DOMÍNIO DE TEMPO E DE FREQUENCIA

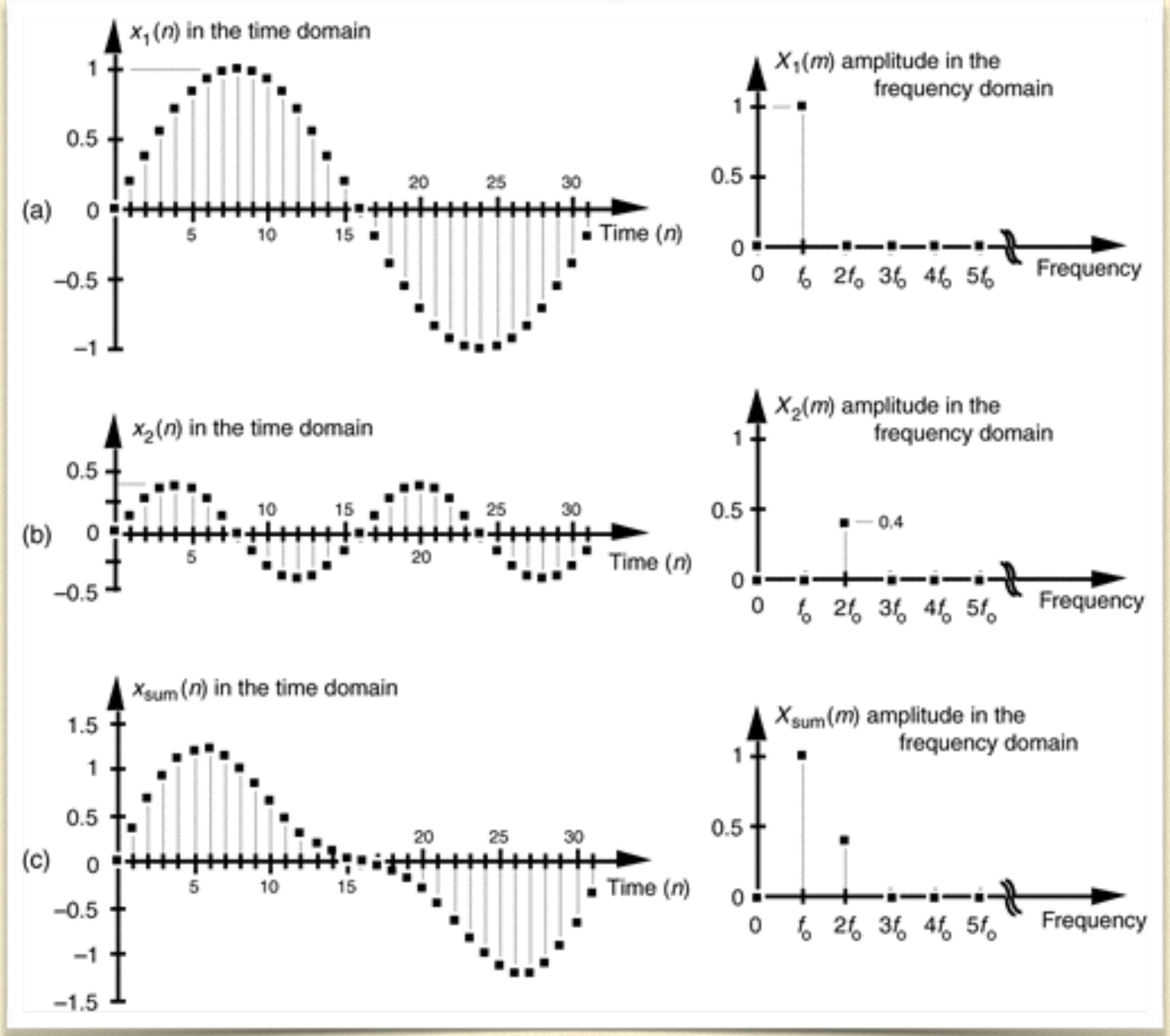




DOMÍNIO DE TEMPO E DE FREQUENCIA



DOMÍNIO DE TEMPO E DE FREQUENCIA



TRANSFORMADA DE FOURIER

DOMÍNIO DO TEMPO -> DOMÍNIO DA FREQUENCIA

$$\omega = \frac{2\pi}{T} = 2\pi f$$

$$\mathbb{F}[f(t)] = F(\omega) = \int_{-\infty}^{\infty} f(t).e^{-j\omega t}\partial t$$

$$\mathbb{F}^{-1}[F(\omega)] = f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega).e^{+j\omega t}\partial\omega$$

TRANSFORMADA DISCRETA DE FOURIER (DFT)

DOMÍNIO DO TEMPO -> DOMÍNIO DA FREQUENCIA

$$\mathbb{F}[f(t)] = F(\omega) = \int_{-\infty}^{\infty} f(t).e^{-j\omega t}\partial t$$

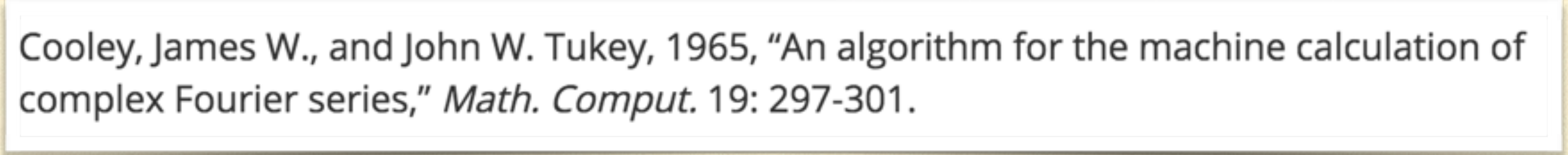
Contínua

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n].e^{-j\omega n}$$

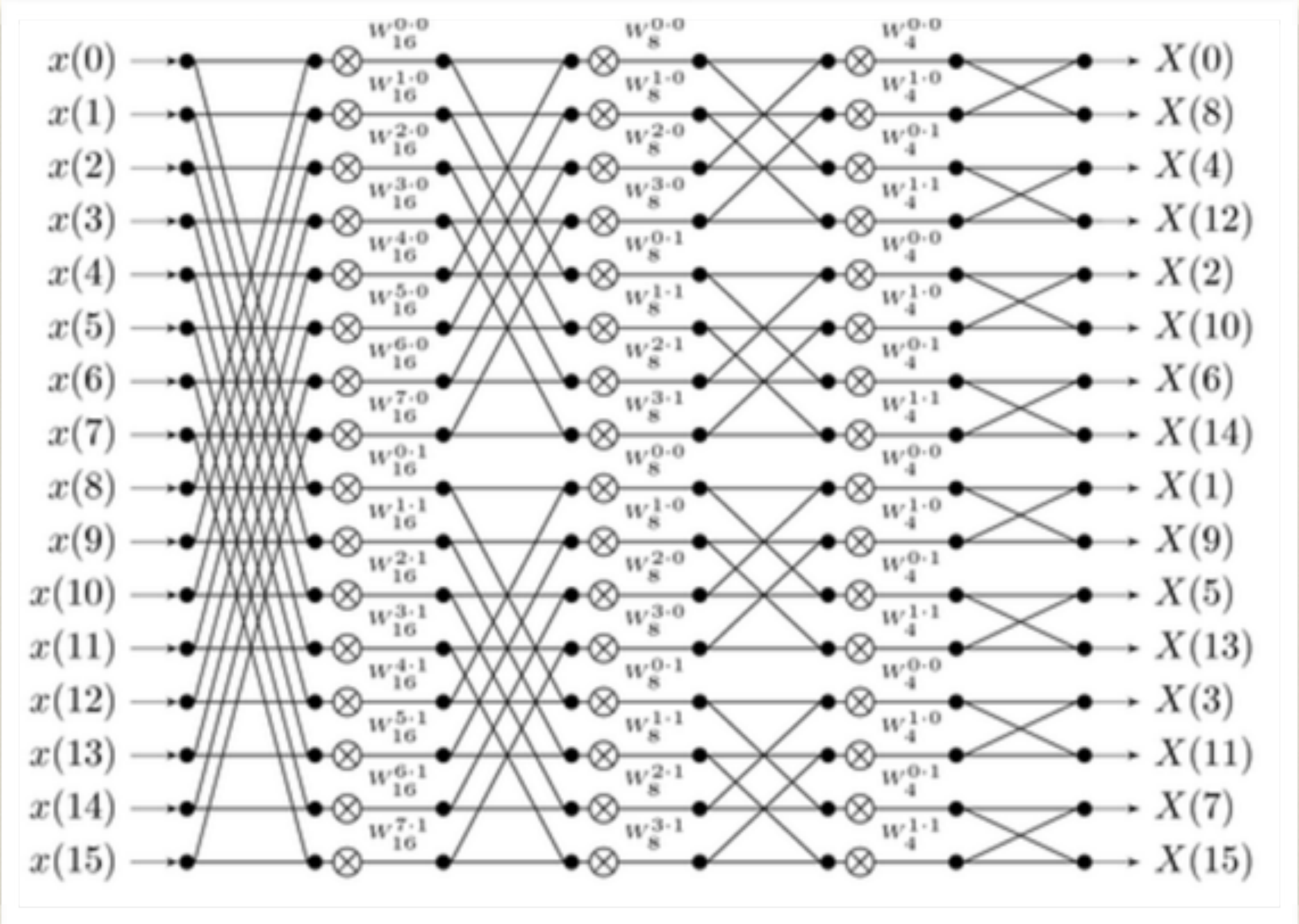
Discreta

TRANSFORMADA RÁPIDA DE FOURIER (FFT)

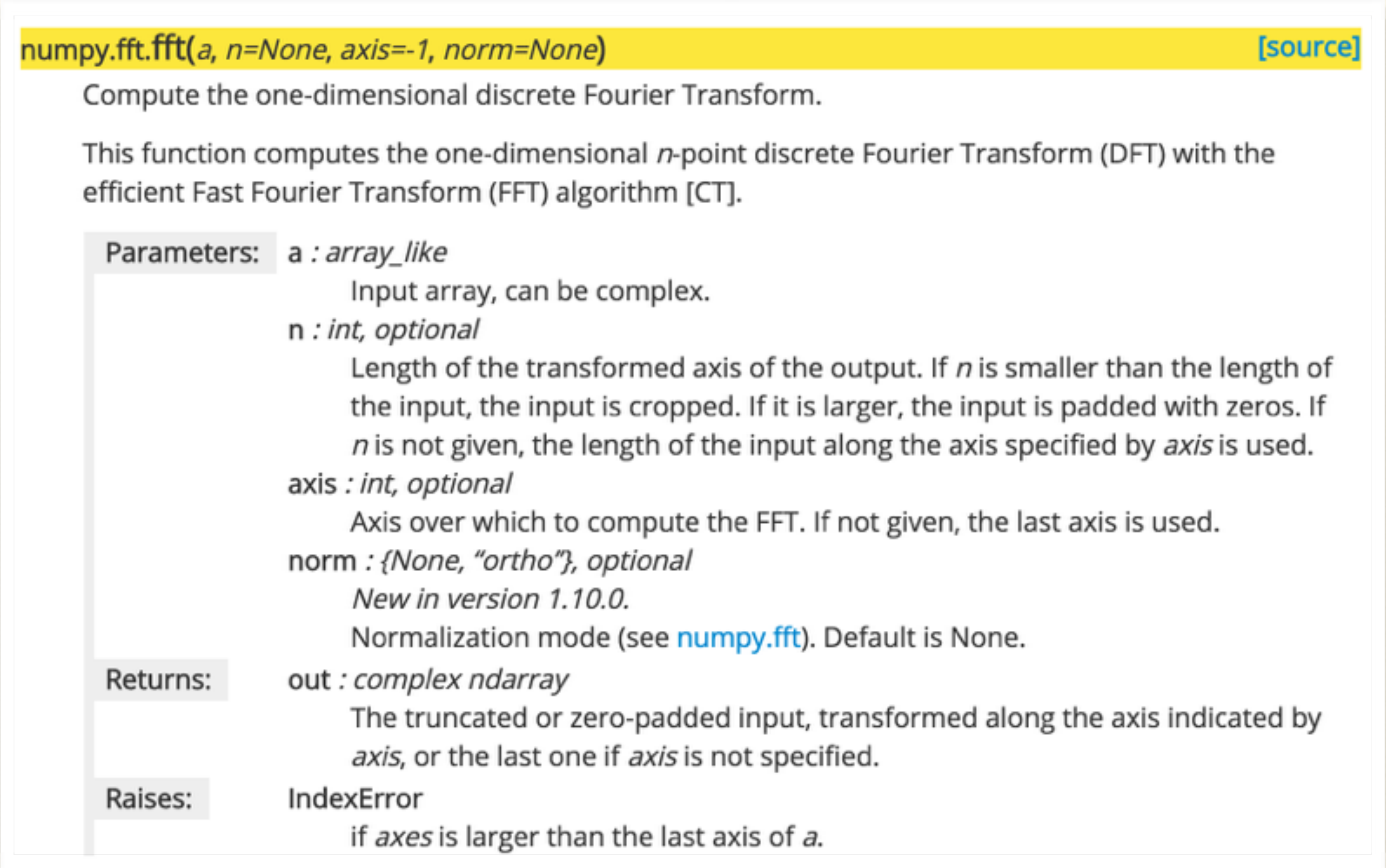
DOMÍNIO DO TEMPO -> DOMÍNIO DA FREQUENCIA



Cooley, James W., and John W. Tukey, 1965, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput.* 19: 297-301.



FFT COM SINAIS COMPLEXOS



numpy.fft.fft(*a*, *n=None*, *axis=-1*, *norm=None*) [source]

Compute the one-dimensional discrete Fourier Transform.

This function computes the one-dimensional *n*-point discrete Fourier Transform (DFT) with the efficient Fast Fourier Transform (FFT) algorithm [CT].

Parameters:

a : *array_like*
Input array, can be complex.

n : *int, optional*
Length of the transformed axis of the output. If *n* is smaller than the length of the input, the input is cropped. If it is larger, the input is padded with zeros. If *n* is not given, the length of the input along the axis specified by *axis* is used.

axis : *int, optional*
Axis over which to compute the FFT. If not given, the last axis is used.

norm : *{None, "ortho"}, optional*
New in version 1.10.0.
Normalization mode (see numpy.fft). Default is None.

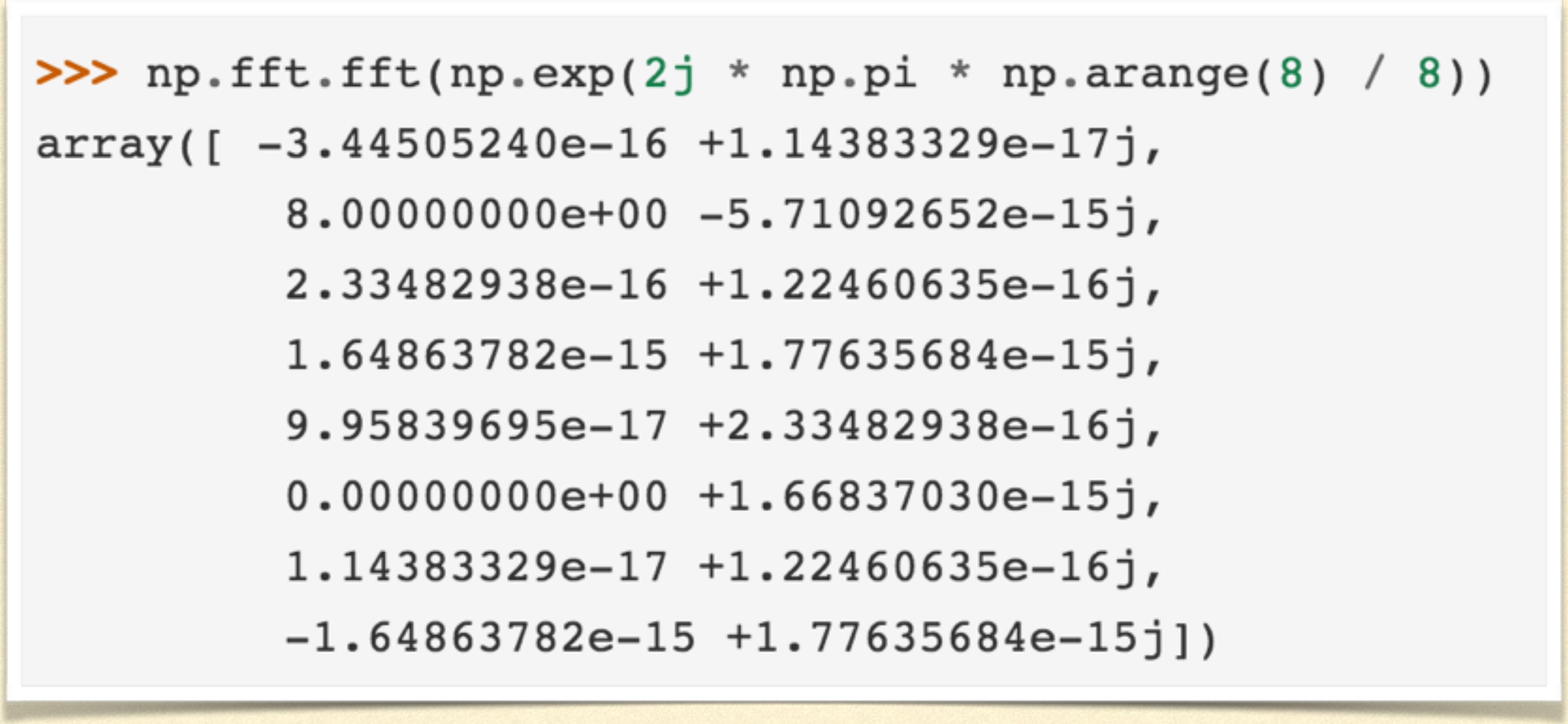
Returns:

out : *complex ndarray*
The truncated or zero-padded input, transformed along the axis indicated by *axis*, or the last one if *axis* is not specified.

Raises:

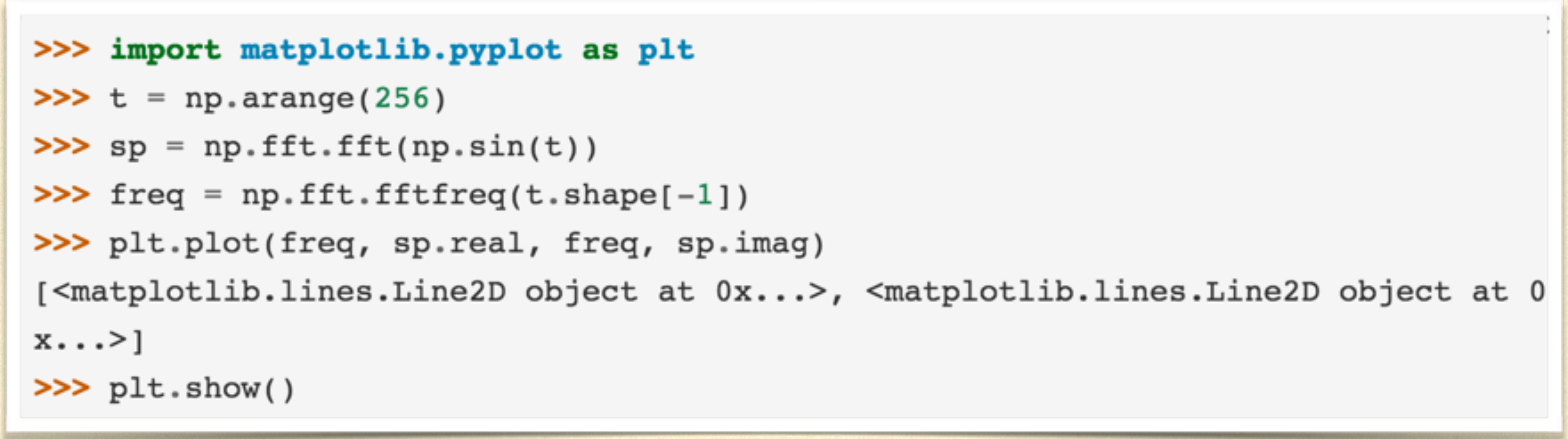
IndexError
if *axes* is larger than the last axis of *a*.

FFT EM PYTHON



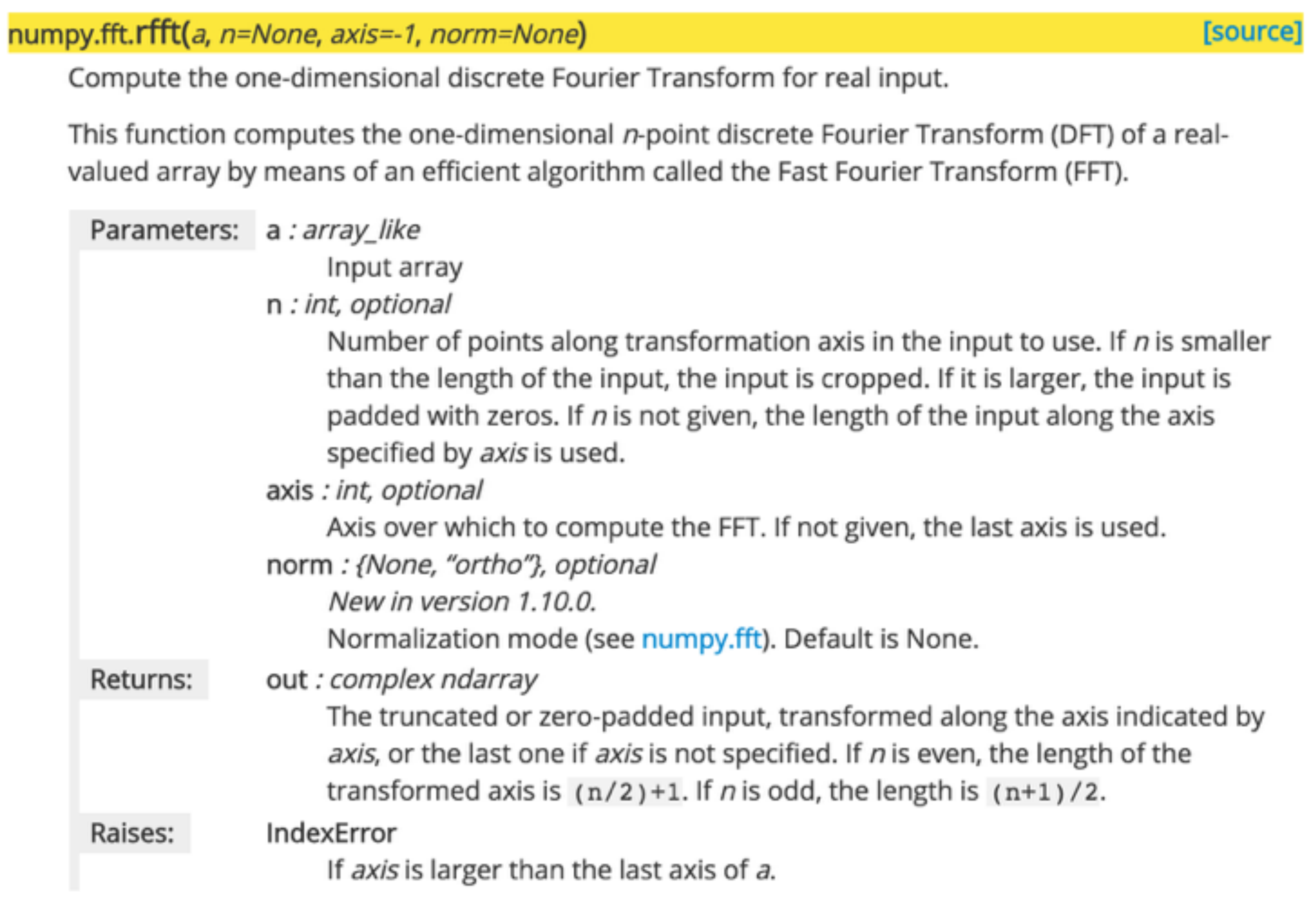
```
>>> np.fft.fft(np.exp(2j * np.pi * np.arange(8) / 8))
array([ -3.44505240e-16 +1.14383329e-17j,
         8.00000000e+00 -5.71092652e-15j,
         2.33482938e-16 +1.22460635e-16j,
         1.64863782e-15 +1.77635684e-15j,
         9.95839695e-17 +2.33482938e-16j,
         0.00000000e+00 +1.66837030e-15j,
         1.14383329e-17 +1.22460635e-16j,
         -1.64863782e-15 +1.77635684e-15j])
```

FFT EM PYTHON



```
>>> import matplotlib.pyplot as plt
>>> t = np.arange(256)
>>> sp = np.fft.fft(np.sin(t))
>>> freq = np.fft.fftfreq(t.shape[-1])
>>> plt.plot(freq, sp.real, freq, sp.imag)
[<matplotlib.lines.Line2D object at 0x...>, <matplotlib.lines.Line2D object at 0
x...>]
>>> plt.show()
```

FFT COM SINAIS REAIS



numpy.fft.**rfft**(*a*, *n=None*, *axis=-1*, *norm=None*) [source]

Compute the one-dimensional discrete Fourier Transform for real input.

This function computes the one-dimensional *n*-point discrete Fourier Transform (DFT) of a real-valued array by means of an efficient algorithm called the Fast Fourier Transform (FFT).

Parameters:

a *: array_like*

Input array

n *: int, optional*

Number of points along transformation axis in the input to use. If *n* is smaller than the length of the input, the input is cropped. If it is larger, the input is padded with zeros. If *n* is not given, the length of the input along the axis specified by *axis* is used.

axis *: int, optional*

Axis over which to compute the FFT. If not given, the last axis is used.

norm *: {None, "ortho"}, optional*

New in version 1.10.0.

Normalization mode (see numpy.fft). Default is None.

Returns:

out *: complex ndarray*

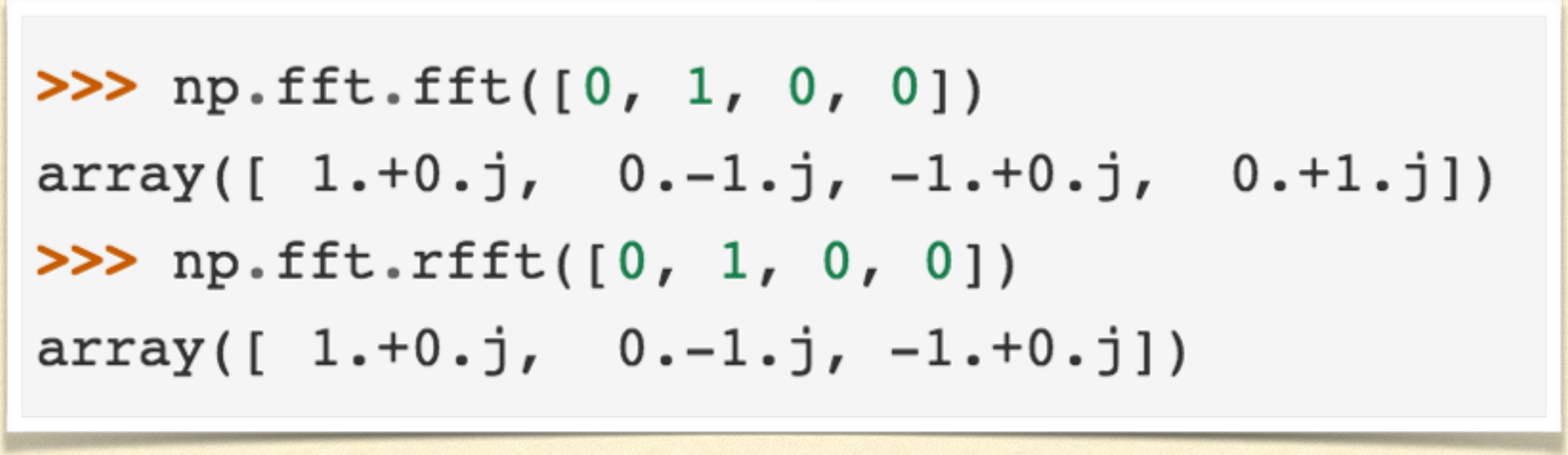
The truncated or zero-padded input, transformed along the axis indicated by *axis*, or the last one if *axis* is not specified. If *n* is even, the length of the transformed axis is `(n/2)+1`. If *n* is odd, the length is `(n+1)/2`.

Raises:

IndexError

If *axis* is larger than the last axis of *a*.

FFT COM SINAIS REAIS



```
>>> np.fft.fft([0, 1, 0, 0])
array([ 1.+0.j,  0.-1.j, -1.+0.j,  0.+1.j])
>>> np.fft.rfft([0, 1, 0, 0])
array([ 1.+0.j,  0.-1.j, -1.+0.j])
```

EXERCÍCIO (I)

Exibir os gráficos do sinal real e os resultados da Transformada de Fourier Complexa e Real calculados anteriormente.

EXERCÍCIO (II)

Verificar se há possibilidade de haver periodicidade no sinal StarMag do arquivo tabela.fits.