

Faz um espectrograma!

Copiei este código de algum lugar mas não lembro qual! Funciona!

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
```

Equação de uma onda progressiva

Displacement

Amplitude

Oscillating term

Phase

Angular frequency

Initial phase

Position

Propagation constant

$$y(x, t) = a \sin(\omega t \pm kx \pm \phi_0)$$

Que, se observada a partir de uma posição fixa, o termo com 'x' é nulo

```
In [41]: # gera um sinal qualquer...

# gera um domínio para a solução da função de onda (tempo!), 15 dias com frequencia
t = np.arange(0, 24*15, 1)

# onda 1 (onda diurna 'Solar')
amp1 = 0.3
omega1 = 2*np.pi/24 # velocidade angular

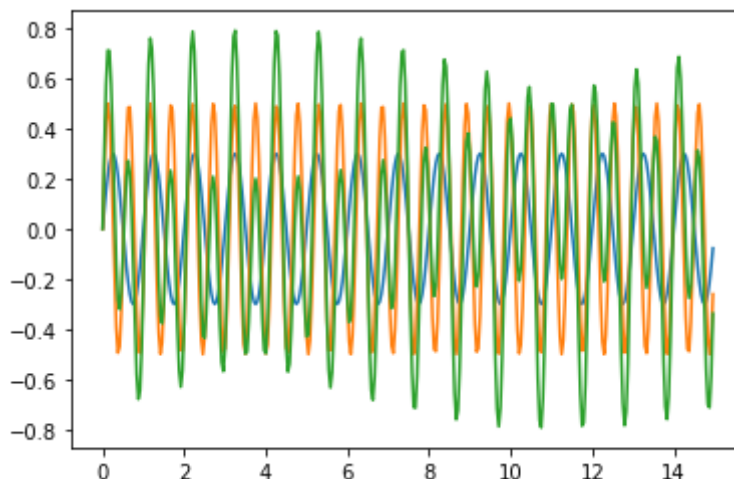
# onda 2 (onda semi-diurna 'Lunar')
amp2 = 0.5
omega2 = 2*np.pi/(12 + 25/60)

# gera as séries de oscilações
n1 = amp1 * np.sin(t*omega1)
n2 = amp2 * np.sin(t*omega2)

plt.plot(t/24, n1)
plt.plot(t/24, n2)

plt.plot(t/24, n1+n2)
```

Out[41]: [<matplotlib.lines.Line2D at 0x22760a80e80>]



```
In [40]: # faz pelo código...
sinal = n1

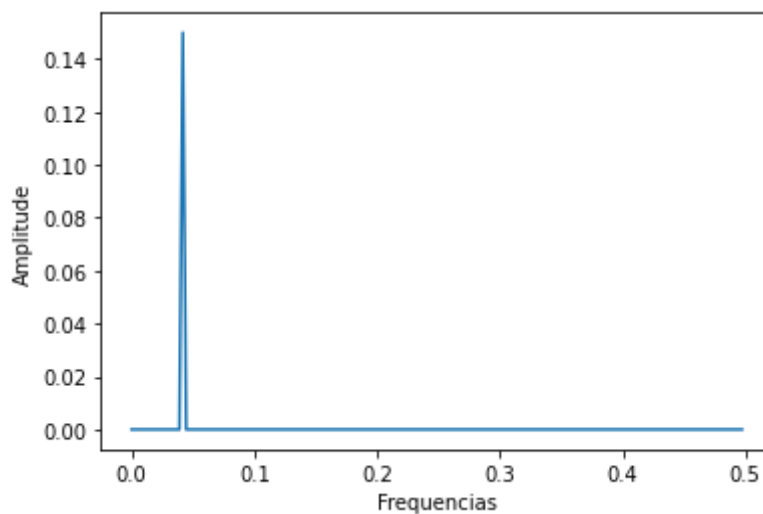
intervalo_amostral = 1
frequencia_amostral = 1/intervalo_amostral

amostras = len(sinal)

transformada_Fourier = np.fft.fft(sinal)/amostras # amplitude normalizada
transformada_Fourier = transformada_Fourier[range(int(amostras/2))]

valores = np.arange(int(amostras/2))
periodo_tempo = amostras / frequencia_amostral
frequencias = valores / periodo_tempo

plt.plot(frequencias, abs(transformada_Fourier))
plt.xlabel('Frequencias')
plt.ylabel('Amplitude')
plt.show()
```



```
In [36]: # faz por função
def faz_fft(sinal, intervalo_amostral = 1):
    # sinal = n1
    # intervalo_amostral = 1
    frequencia_amostral = 1/intervalo_amostral

    n_amostras = len(sinal)

    transformada_Fourier = np.fft.fft(sinal)/n_amostras # amplitude normaliza
    transformada_Fourier = transformada_Fourier[range(int(n_amostras/2))]

    valores = np.arange(int(n_amostras/2))
    periodo_tempo = n_amostras / frequencia_amostral
    frequencias = valores / periodo_tempo

    return frequencias, abs(transformada_Fourier)

f1, tf1 = faz_fft(n1)
f2, tf2 = faz_fft(n2)
f3, tf3 = faz_fft(n1+n2)

plt.plot(f1, tf1, lw=5)
plt.plot(f2, tf2, lw=5)
plt.plot(f3, tf3, lw=0.5, color='k')
plt.xlabel('Frequências')
```

```
plt.ylabel('Amplitude')  
# plt.yscale('log')  
# plt.xscale('log')  
plt.show()
```

