

```
# -*- coding: utf-8 -*-
import numpy as np
from scipy import fft
import numpy, matplotlib, scipy.fftpack
from pylab import plt

num = 256
norm = 2.0 / float(num) # normalization factor
sample = 4.
time = np.arange(num) / sample
dx = time[1] - time[0]
f1 = .88
a1 = 2.
kont = a1 * np.sin(2 * np.pi * f1 * time) # the input
g = fft(kont) * norm
#-----
#-- Method 1
freq = np.fft.fftfreq(kont.shape[-1])/dx
plt.plot(freq, abs(np.fft.fftshift(g)))
#-----
#-----
#-- Method 2
#freq = np.fft.fftfreq(kont.shape[-1])/dx
#plt.plot(freq, abs(np.fft.fftshift(g)))
#-----
plt.axvline(x=f1, linewidth=2, color='r')
plt.xlim(xmin=1e-6) # to have only positive frequencies
plt.xlabel(r'Frequency')
plt.ylabel(r'Normalized amplitude')
plt.title('method 1')
plt.show()
```