1D Energy Spectrum

- Takes in an 1D array and outputs wavenumber k and magnetic energy

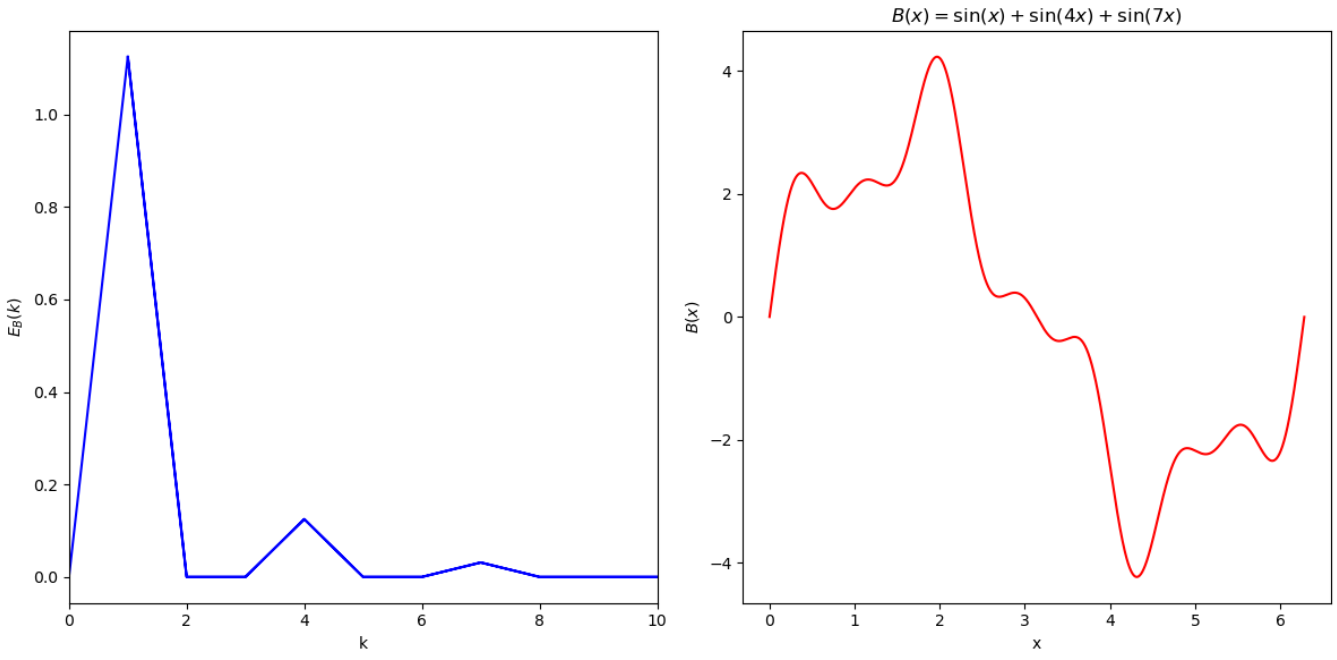
```
import numpy.fft as nf
import errno
import math
import os
import sys
import re
import pickle
import h5py as h5
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
from scipy.stats import norm
import statistics
def Spec1D(ar):
    if len(ar) == 0:
      print('No Array')
      return
    ar=ar-np.mean(ar)
    nx=len(ar)
    k= np.abs(nf.fftfreq(nx)* nx)
    far = nf.fft(ar)/(nx);
    ffteb=0.5*np.abs(far)**2
    return k, ffteb
```

Example Signal:

$$B_x = \sin(x) + \sin(4x) + \sin(7x)$$

```
# sampling rate
sr = 2000
# sampling interval
ts = 1.0/sr
t = np.arange(0,2 * np.pi,ts)

freq = 1.
x1 = 3*np.sin(freq*t)

freq = 4
```

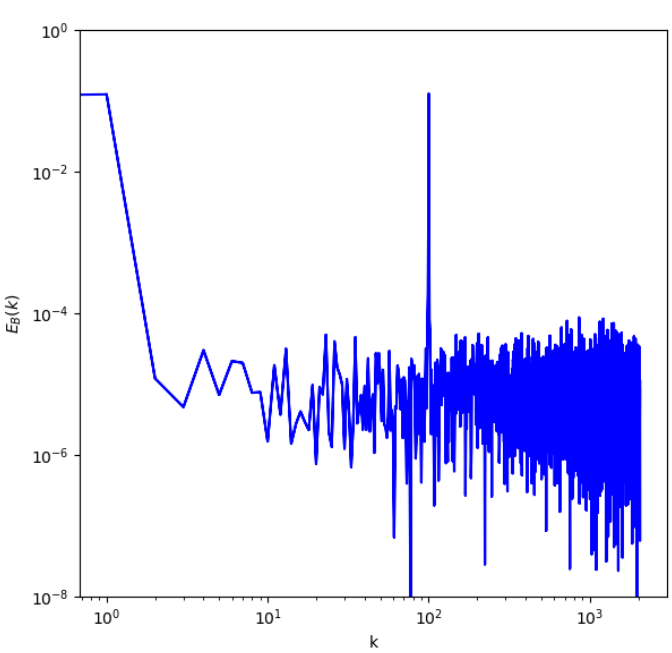
```
x2 = np.sin(freq*t)

freq = 7
x3 = 0.5* np.sin(freq*t)
k, Eb = Spec1D(x1 + x2 + x3)

plt.figure(figsize = (12, 6))
plt.subplot(121)

plt.plot(k, Eb, 'b')
plt.xlabel('k')
plt.ylabel(r'$E_B (k)$')
plt.xlim(0,10)

plt.subplot(122)
plt.title(r'$B(x) = \sin(x) + \sin(4x) + \sin(7x)$')
plt.plot(t, x1 + x2 + x3, 'r')
plt.xlabel('x')
plt.ylabel(r'$B(x)$')
plt.tight_layout()

plt.savefig('1Dspectrum_example.png')
plt.show()
```

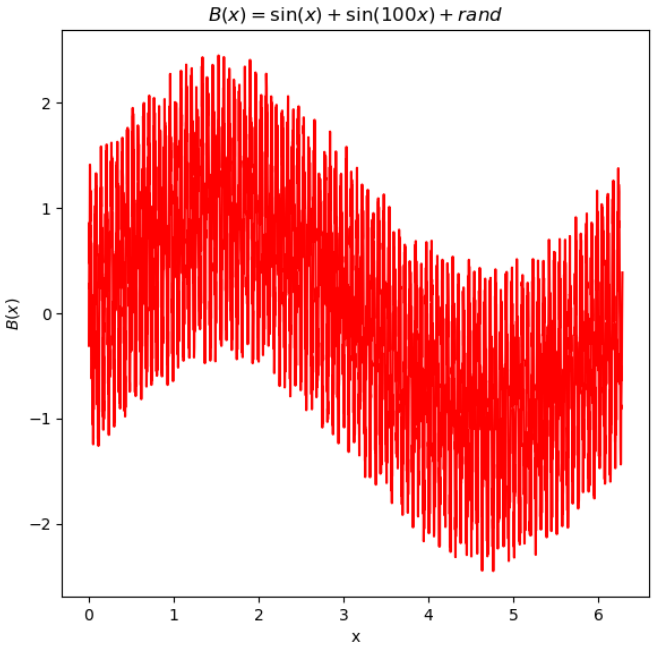


Example Signal 2:

$$B_x = \sin(x) + \sin(100x) + \phi(x)$$

```
Ns = 4096 # number of samples
np.random.seed(1234) # random seed (for repeatability)
rn = np.random.random(Ns)-0.5 # zero mean random noise

Fs = 100 # sampling frequency
dt = 1./Fs # time discretisation
tt = np.arange(Ns)*dt # time sampling
print(len(tt))

# sampling rate
sr = 4096
# sampling interval
ts = 1.0/sr
t = np.linspace(0,2 * np.pi, Ns)

freq = 1.
A = 1
sw = A*np.sin(freq*t) # sine wave


freq2 = 100
sw2 = A * np.sin(freq2*t)

ss = sw+ sw2 + rn # sample signal
k, ffteb = Spec1D(ss)

plt.figure(figsize = (12, 6))
plt.subplot(121)

plt.loglog(k, ffteb, 'b')
plt.xlabel('k')
plt.ylabel(r'$E_B (k)$')
# plt.xlim(0,10)
plt.ylim(10**(-8), 10**(0))

plt.subplot(122)
plt.title(r'$B(x) = \sin(x) + \sin(100x) + rand$')
plt.plot(t, ss, 'r')
plt.xlabel('x')
plt.ylabel(r'$B(x)$')
plt.tight_layout()

plt.savefig('1D_spectrum_ex.png')
plt.show()
```


10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
$E_B(k)$
10^0
10^1
10^2
10^3
k


$B(x) = \sin(x) + \sin(100x) + rand$
2
1
0
−1
−2
$B(x)$
0
1
2
3
4
5
6
x