

```

1  #!/usr/bin/env python3
2
3  import numpy                as np
4  import matplotlib.pyplot    as plt
5  import pandas               as pd
6  import allantools
7
8  def read_data(fname : str):
9      df = pd.read_excel(fname, 'Sheet1', parse_dtypes=['Timet(s)',
10 'Deviation(Hz)'])
11      # print(df.head())
12      var1 = df['Timet(s)'].tolist()
13      var2 = df['Deviation(Hz)'].tolist()
14      return var1, var2
15
16 def partial_average(fract_data: list,
17                     step      : int) :
18     nparts = (len(fract_data) - 1) // step
19     out     = []
20     for i in range(nparts) :
21         partial_sum = .0
22         for j in range(i, i+step + 1):
23             partial_sum += fract_data[j]
24         out.append(partial_sum / step )
25     return out
26
27 def calc_deviation(pavgs : list) :
28     sum_of_dy2 = .0
29     for i in range(len(pavgs) - 1):
30         sum_of_dy2 += (pavgs[i+1] - pavgs[i]) ** 2
31     return np.sqrt(sum_of_dy2 / (2*len(pavgs) - 2))
32
33 def main() :
34     # f_0 = 1
35     f_0 = 3e8 / 730e-9 # the frequency of 730nm light
36     timet, fract_data = read_data(r'./Ca_clock_transition_data.xls')
37     timet.pop()
38     fract_data.pop()
39
40     y = []
41     x = timet[:len(fract_data)//2]
42
43     for i in range(len(fract_data) // 2):
44         pavgs = partial_average(fract_data, i + 1)
45         devia = calc_deviation (pavgs)
46         y.append(devia / f_0)

```

```

46
47     plt.loglog(x, y)
48     plt.xlabel('$\\tau$ / sec$')
49     plt.ylabel('$\\delta f$ / f$')
50     plt.savefig('ad.png', dpi=400)
51     # plt.show()
52
53 def main_use_allantools() :
54     f_0 = 3e8 / 730e-9 # the frequency of 730nm light
55     timet, fract_data = read_data(r'./Ca_clock_transition_data.xls')
56     timet.pop()
57     fract_data.pop()
58
59     for i, elem in enumerate(fract_data) :
60         fract_data[i] = elem / f_0
61     (t2, ad, ade, adn) = allantools.oadev(fract_data, rate=11.2,
62     data_type="freq", taus = [i for i in range(1, len(timet) +
63 1)])
64     plt.loglog(t2, ad)
65     plt.savefig('ad.png', dpi=400)
66     # plt.show()
67
68 def debug_partial_average(step: int) :
69     vec = [i for i in range(1, 21)]
70     pavg = partial_average(vec, step)
71     print('step = {}, vec.size() = {}, pavg.size = {}'.
72     .format(step, len(vec), len(pavg)))
73     print(vec)
74     print(pavg, '\\n')
75
76 def debug_deviation() :
77     vec = [i for i in range(1, 21)]
78     devia = calc_deviation(vec)
79     print(vec)
80     print(devia)
81
82 if __name__ == "__main__" :
83     # for i in range(1, 10):
84     #     debug_partial_average(i)
85     #     debug_deviation()
86     main()
87     # main_use_allantools()

```