

# 11-12\_FFT\_Data\_Example

September 27, 2023

```
[1]: import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
[3]: !head 11-12_force.csv
```

```
,/ACF/telem/force
1630993378.1155696,-10.0
1630993383.4162107,-10.0
1630993396.461548,-10.100000381469727
1630993396.4934843,-4.800000190734863
1630993396.5416863,19.700000762939453
1630993396.5936322,25.200000762939453
1630993396.6417477,25.600000381469727
1630993396.6935449,26.100000381469727
1630993396.741945,25.799999237060547
```

```
[6]: data = np.loadtxt("11-12_force.csv", delimiter=",", skiprows=1)
data
```

```
[6]: array([[ 1.63099338e+09, -1.00000000e+01],
           [ 1.63099338e+09, -1.00000000e+01],
           [ 1.63099340e+09, -1.01000004e+01],
           ...,
           [ 1.63099351e+09,  2.61000004e+01],
           [ 1.63099351e+09,  2.61000004e+01],
           [ 1.63099351e+09,  2.63999996e+01]])
```

```
[8]: t = data[:,0]
force = data[:,1]
t
```

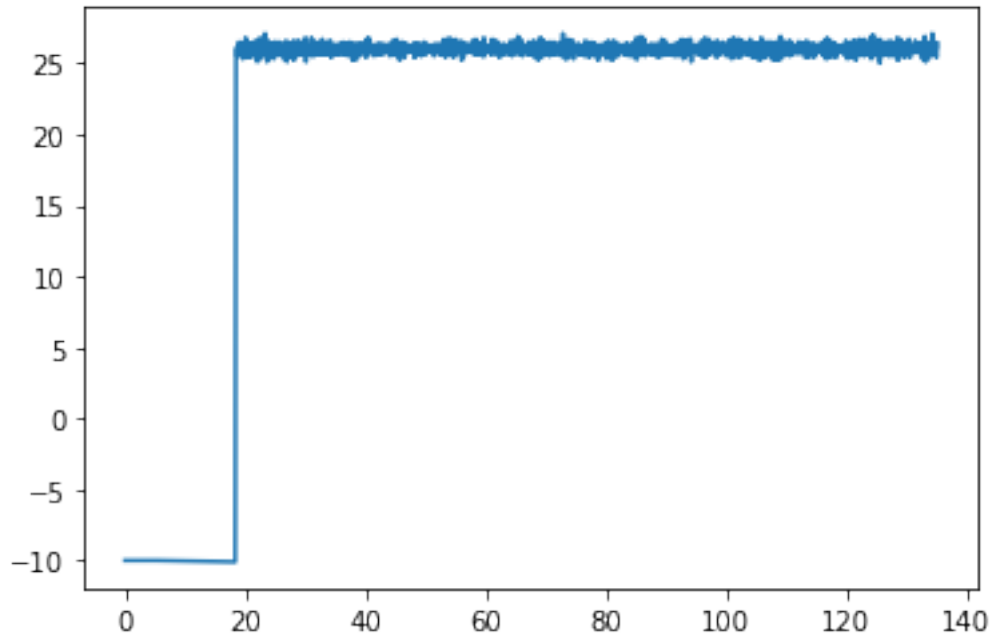
```
[8]: array([1.63099338e+09, 1.63099338e+09, 1.63099340e+09, ...,
          1.63099351e+09, 1.63099351e+09, 1.63099351e+09])
```

```
[9]: t = t - t[0]
t
```

```
[9]: array([ 0.          ,  5.30064106, 18.3459785 , ..., 135.12995982,
          135.17822385, 135.23330331])
```

```
[10]: plt.plot(t, force)
```

```
[10]: [<matplotlib.lines.Line2D at 0x7fb2586cfb50>]
```



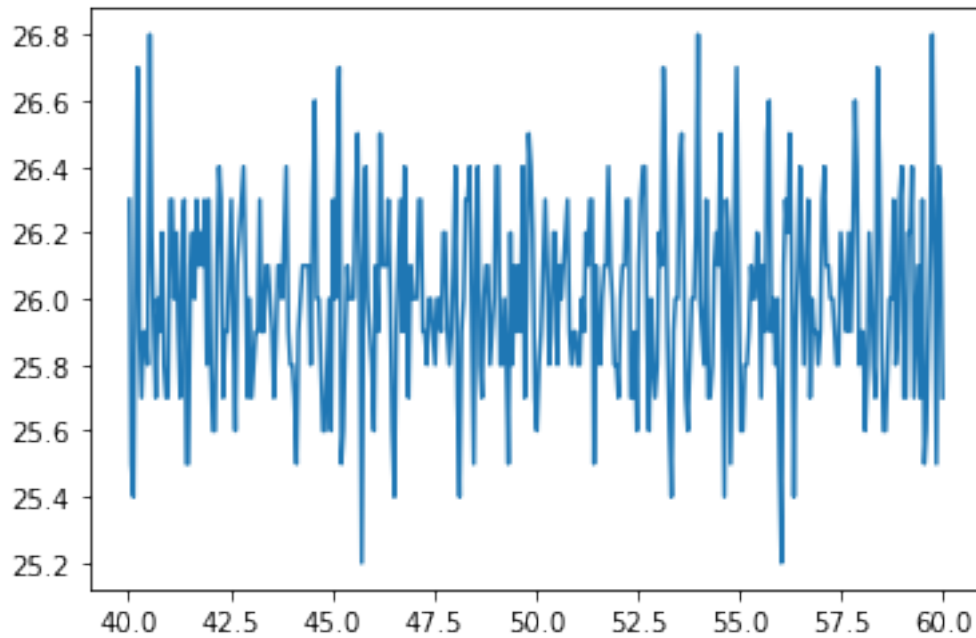
```
[12]: mask = (t > 40) & (t < 60)
      mask
```

```
[12]: array([False, False, False, ..., False, False, False])
```

```
[13]: t = t[mask]
      force = force[mask]
```

```
[14]: plt.plot(t, force)
```

```
[14]: [<matplotlib.lines.Line2D at 0x7fb258667040>]
```



```
[16]: dt = np.mean(np.diff(t))  
dt
```

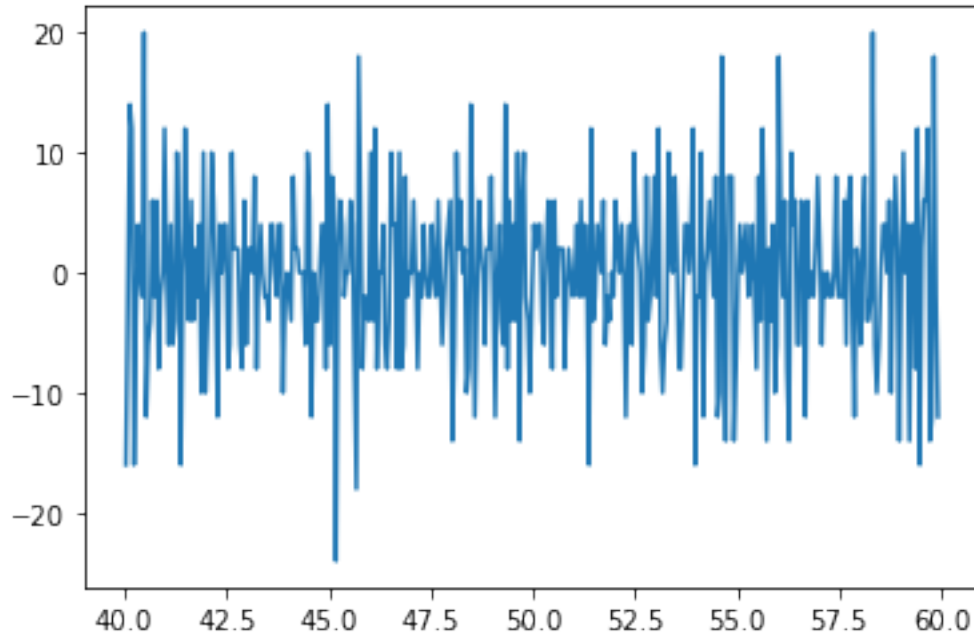
```
[16]: 0.04999916714833195
```

```
[17]: x_test = np.array([1, 2, 4, 3])  
np.diff(x_test)
```

```
[17]: array([ 1,  2, -1])
```

```
[20]: force_deriv = np.diff(force) / dt  
plt.plot(t[:-1], force_deriv)
```

```
[20]: [<matplotlib.lines.Line2D at 0x7fb2585c2820>]
```



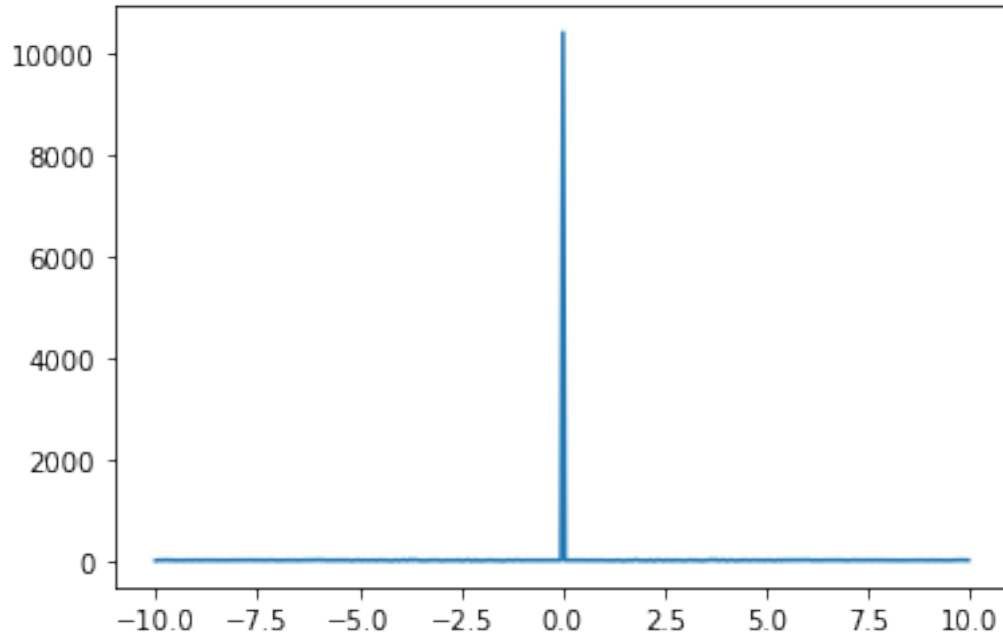
```
[21]: F = np.fft.fft(force)
```

```
[22]: omega = np.fft.fftfreq(force.size, d=dt)
```

```
[23]: order = omega.argsort()  
plt.plot(omega[order], F[order])
```

```
/usr/lib/python3/dist-packages/numpy/core/_asarray.py:85: ComplexWarning:  
Casting complex values to real discards the imaginary part  
return array(a, dtype, copy=False, order=order)
```

```
[23]: [<matplotlib.lines.Line2D at 0x7fb2585986a0>]
```

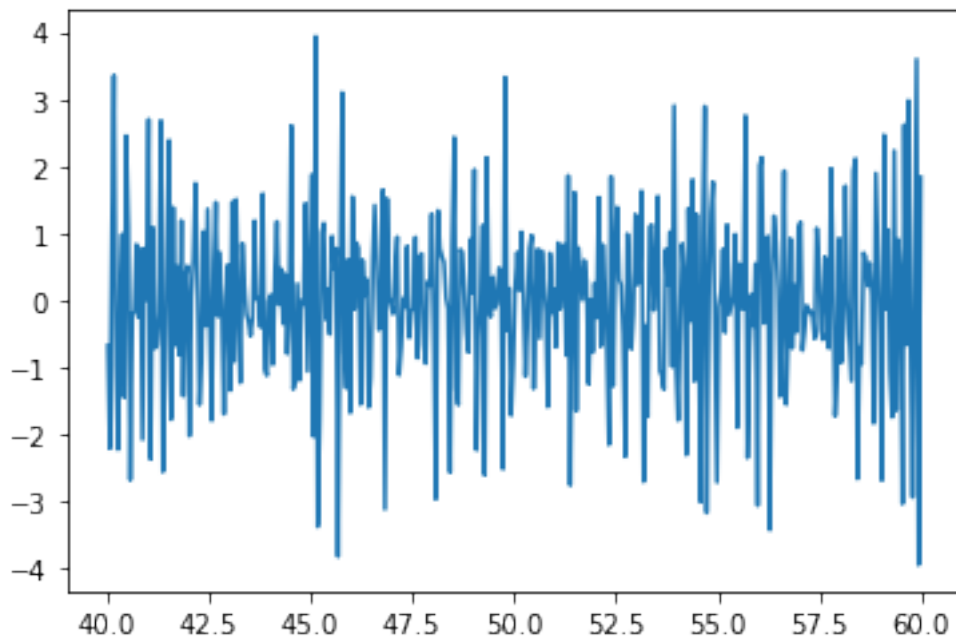


```
[24]: F_deriv = 1j * omega * F
```

```
[25]: force_deriv_fft = np.fft.ifft(F_deriv)
```

```
[26]: plt.plot(t, force_deriv_fft)
```

```
[26]: [<matplotlib.lines.Line2D at 0x7fb2584f38e0>]
```



[27]: omega

```
[27]: array([ 0.          ,  0.05000083,  0.10000167,  0.1500025 ,
 0.20000333,  0.25000416,  0.300005   ,  0.35000583,
 0.40000666,  0.4500075 ,  0.50000833,  0.55000916,
 0.60000999,  0.65001083,  0.70001166,  0.75001249,
 0.80001333,  0.85001416,  0.90001499,  0.95001582,
 1.00001666,  1.05001749,  1.10001832,  1.15001916,
 1.20001999,  1.25002082,  1.30002165,  1.35002249,
 1.40002332,  1.45002415,  1.50002499,  1.55002582,
 1.60002665,  1.65002748,  1.70002832,  1.75002915,
 1.80002998,  1.85003082,  1.90003165,  1.95003248,
 2.00003331,  2.05003415,  2.10003498,  2.15003581,
 2.20003665,  2.25003748,  2.30003831,  2.35003914,
 2.40003998,  2.45004081,  2.50004164,  2.55004248,
 2.60004331,  2.65004414,  2.70004497,  2.75004581,
 2.80004664,  2.85004747,  2.90004831,  2.95004914,
 3.00004997,  3.0500508 ,  3.10005164,  3.15005247,
 3.2000533 ,  3.25005414,  3.30005497,  3.3500558 ,
 3.40005663,  3.45005747,  3.5000583 ,  3.55005913,
 3.60005997,  3.6500608 ,  3.70006163,  3.75006246,
 3.8000633 ,  3.85006413,  3.90006496,  3.9500658 ,
 4.00006663,  4.05006746,  4.10006829,  4.15006913,
 4.20006996,  4.25007079,  4.30007163,  4.35007246,
 4.40007329,  4.45007413,  4.50007496,  4.55007579,
 4.60007662,  4.65007746,  4.70007829,  4.75007912,
 4.80007996,  4.85008079,  4.90008162,  4.95008245,
 5.00008329,  5.05008412,  5.10008495,  5.15008579,
 5.20008662,  5.25008745,  5.30008828,  5.35008912,
 5.40008995,  5.45009078,  5.50009162,  5.55009245,
 5.60009328,  5.65009411,  5.70009495,  5.75009578,
 5.80009661,  5.85009745,  5.90009828,  5.95009911,
 6.00009994,  6.05010078,  6.10010161,  6.15010244,
 6.20010328,  6.25010411,  6.30010494,  6.35010577,
 6.40010661,  6.45010744,  6.50010827,  6.55010911,
 6.60010994,  6.65011077,  6.7001116 ,  6.75011244,
 6.80011327,  6.8501141 ,  6.90011494,  6.95011577,
 7.0001166 ,  7.05011743,  7.10011827,  7.1501191 ,
 7.20011993,  7.25012077,  7.3001216 ,  7.35012243,
 7.40012326,  7.4501241 ,  7.50012493,  7.55012576,
 7.6001266 ,  7.65012743,  7.70012826,  7.75012909,
 7.80012993,  7.85013076,  7.90013159,  7.95013243,
 8.00013326,  8.05013409,  8.10013492,  8.15013576,
 8.20013659,  8.25013742,  8.30013826,  8.35013909,
 8.40013992,  8.45014075,  8.50014159,  8.55014242,
```

8.60014325,	8.65014409,	8.70014492,	8.75014575,
8.80014658,	8.85014742,	8.90014825,	8.95014908,
9.00014992,	9.05015075,	9.10015158,	9.15015241,
9.20015325,	9.25015408,	9.30015491,	9.35015575,
9.40015658,	9.45015741,	9.50015824,	9.55015908,
9.60015991,	9.65016074,	9.70016158,	9.75016241,
9.80016324,	9.85016407,	9.90016491,	9.95016574,
-10.00016657,	-9.95016574,	-9.90016491,	-9.85016407,
-9.80016324,	-9.75016241,	-9.70016158,	-9.65016074,
-9.60015991,	-9.55015908,	-9.50015824,	-9.45015741,
-9.40015658,	-9.35015575,	-9.30015491,	-9.25015408,
-9.20015325,	-9.15015241,	-9.10015158,	-9.05015075,
-9.00014992,	-8.95014908,	-8.90014825,	-8.85014742,
-8.80014658,	-8.75014575,	-8.70014492,	-8.65014409,
-8.60014325,	-8.55014242,	-8.50014159,	-8.45014075,
-8.40013992,	-8.35013909,	-8.30013826,	-8.25013742,
-8.20013659,	-8.15013576,	-8.10013492,	-8.05013409,
-8.00013326,	-7.95013243,	-7.90013159,	-7.85013076,
-7.80012993,	-7.75012909,	-7.70012826,	-7.65012743,
-7.6001266 ,	-7.55012576,	-7.50012493,	-7.4501241 ,
-7.40012326,	-7.35012243,	-7.3001216 ,	-7.25012077,
-7.20011993,	-7.1501191 ,	-7.10011827,	-7.05011743,
-7.0001166 ,	-6.95011577,	-6.90011494,	-6.8501141 ,
-6.80011327,	-6.75011244,	-6.7001116 ,	-6.65011077,
-6.60010994,	-6.55010911,	-6.50010827,	-6.45010744,
-6.40010661,	-6.35010577,	-6.30010494,	-6.25010411,
-6.20010328,	-6.15010244,	-6.10010161,	-6.05010078,
-6.00009994,	-5.95009911,	-5.90009828,	-5.85009745,
-5.80009661,	-5.75009578,	-5.70009495,	-5.65009411,
-5.60009328,	-5.55009245,	-5.50009162,	-5.45009078,
-5.40008995,	-5.35008912,	-5.30008828,	-5.25008745,
-5.20008662,	-5.15008579,	-5.10008495,	-5.05008412,
-5.00008329,	-4.95008245,	-4.90008162,	-4.85008079,
-4.80007996,	-4.75007912,	-4.70007829,	-4.65007746,
-4.60007662,	-4.55007579,	-4.50007496,	-4.45007413,
-4.40007329,	-4.35007246,	-4.30007163,	-4.25007079,
-4.20006996,	-4.15006913,	-4.10006829,	-4.05006746,
-4.00006663,	-3.9500658 ,	-3.90006496,	-3.85006413,
-3.8000633 ,	-3.75006246,	-3.70006163,	-3.6500608 ,
-3.60005997,	-3.55005913,	-3.5000583 ,	-3.45005747,
-3.40005663,	-3.3500558 ,	-3.30005497,	-3.25005414,
-3.2000533 ,	-3.15005247,	-3.10005164,	-3.0500508 ,
-3.00004997,	-2.95004914,	-2.90004831,	-2.85004747,
-2.80004664,	-2.75004581,	-2.70004497,	-2.65004414,
-2.60004331,	-2.55004248,	-2.50004164,	-2.45004081,
-2.40003998,	-2.35003914,	-2.30003831,	-2.25003748,
-2.20003665,	-2.15003581,	-2.10003498,	-2.05003415,











```

0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, 0.00000000e+00+0.00000000e+00j,
0.00000000e+00+0.00000000e+00j, -3.44137849e+00+1.04507379e+00j,
-4.02834932e+00-2.38950476e+00j, 1.86790702e+00+1.61595768e+00j,
1.12956130e+01+2.10490091e+00j, 4.74292740e+00-8.88567501e+00j,
-3.71344175e+00+3.29106242e-02j, -1.85572947e+00+6.01501265e-01j,
1.56708164e+00+2.32293509e+00j, -1.28101899e+00-4.04628418e+00j,
-1.10581433e+01-6.70895392e+00j, -3.47388143e+00-1.10563811e+00j,
-3.55708688e+00-7.99183358e+00j, 3.96083611e+00+4.58191915e-01j,
1.68469447e+00-1.56115556e+00j, -5.34637363e+00+2.20220652e+00j,
-4.66316451e+00-1.56641350e+00j, 6.70374814e+00+1.44799984e+00j,
3.61865981e+00-4.42391412e+00j, -9.01605265e-01-1.31874914e+00j,
-3.74818832e+00-3.75098792e+00j, 1.01267086e+00-9.34119840e-01j,
1.03556652e+00+4.37318416e-02j, 1.60959999e+00+1.33394013e+00j,
-1.02881871e+00+1.66314963e+00j, -9.03285100e-01+1.87601321e+00j,
2.24515248e+00+8.54714483e-02j, -1.85043774e+00+2.10234656e-01j,
-1.15541423e-01+1.06431113e+00j, -1.06234339e+00-1.62628498e+00j,
1.13172201e+00-4.89567856e-01j, 4.36127139e-01-7.03885125e-01j,
6.91740082e-01-4.76431788e-01j, 1.23266994e+00-9.19743142e-01j,
8.36699981e-01-1.50743014e-01j, -2.37504698e+00+7.33815732e-01j,
-1.01399557e+00+1.38889177e+00j, -8.98204887e-02+6.45731955e-03j,
5.23534078e-01-3.97880499e-01j, -6.70929279e-02-1.22079499e+00j])

```

```

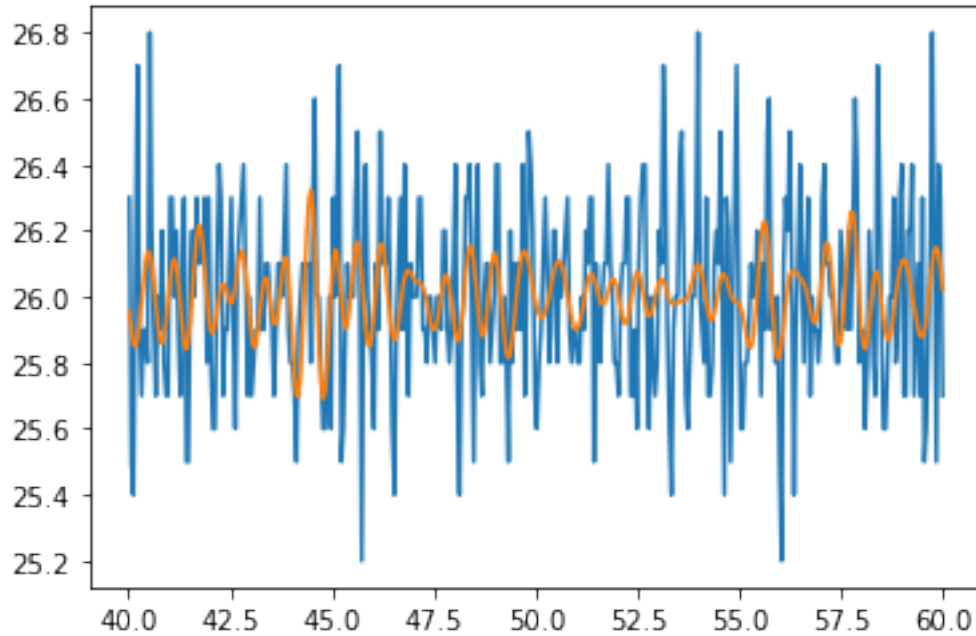
[34]: plt.plot(t, force)
      plt.plot(t, np.fft.ifft(F_filtered))

```

```

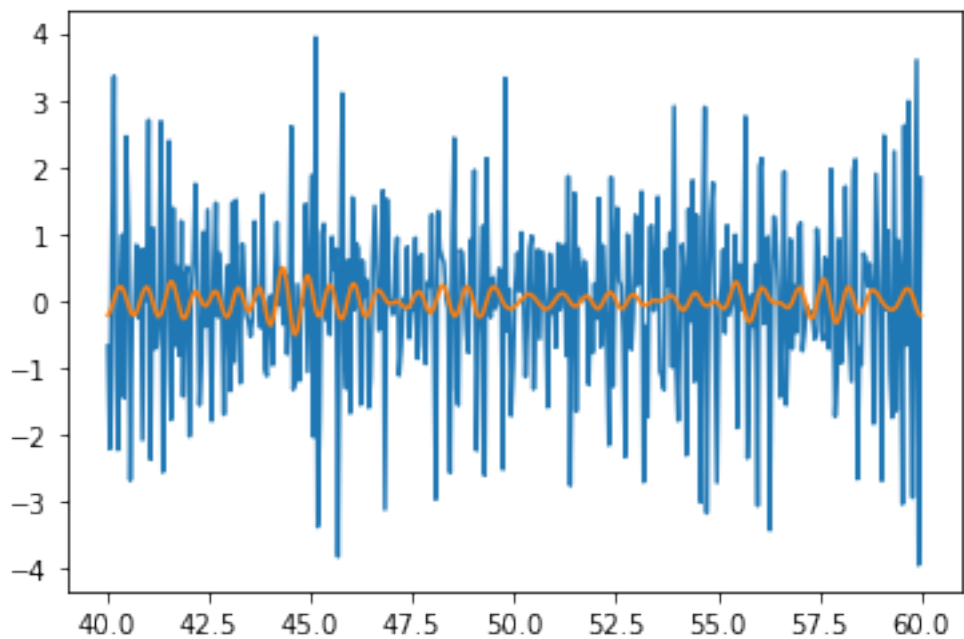
[34]: [<matplotlib.lines.Line2D at 0x7fb25842dd30>]

```



```
[36]: F_filtered_deriv = 1j * omega * F_filtered
plt.plot(t, force_deriv_fft)
plt.plot(t, np.fft.ifft(F_filtered_deriv))
```

[36]: [[matplotlib.lines.Line2D](#) at 0x7fb2583714c0>]



[ ]: