44-44-44-64-6-beta-10.85-twist-1

October 17, 2024

```
[16]: %load_ext autoreload
      %autoreload 2
      from modules import read_and_write
      from modules import polyakov
      from modules import utility
      from modules import fourier_surface
      import pandas as pd
      import os
      import glob
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

For reference with integration method the following surface tensions were computed

z_1: $\alpha_{o-o}/T^3(\beta = 10.85) = 1.2316804724774406$

z_2: $\alpha_{o-o}/T^3(\beta = 10.85) = 1.5433288477348852$

1 Load data

```
[64]: folder_names = ["../data/output-measure-surface/su4-36-36-48-6",
                      "../data/output-measure-surface/su4-44-44-64-6/beta-12-twist-2",
                      "../data/output-measure-surface/su4-44-44-64-6/beta-10.
        ↪9-twist-2",
                      "../data/output-measure-surface/su4-44-44-64-6/beta-10.
        ↪85-twist-1",
                      "../data/output-measure-surface/su4-44-44-64-6/beta-10.
        ↪85-twist-2"]
      choose_folder = 4
      fourier_profiles = {}
      folder = folder_names[choose_folder-1]
      files = glob.glob(os.path.join(folder, "fourier_profile_*"))
      for file in files:
          file_name = file.split("/")[-1]
          smearing_level = file_name.split("_")[-1]
          volume, fourier_profile = read_and_write.read_surface_data(folder,␣
        ↪file_name)
```

```
    fourier_profiles[smearing_level] = fourier_profile
fourier_profiles = dict(sorted(fourier_profiles.items(), key=lambda item:␣
 ↪int(item[0])))
```

[65]:

```
utility.display_markdown_title(folder)
```

2 SU(4), $V = [‘44’, ‘44’, ‘64’, ‘6’]$, $\beta = 10.85$, twist coeff $= 1$

2.1 Perform post processing

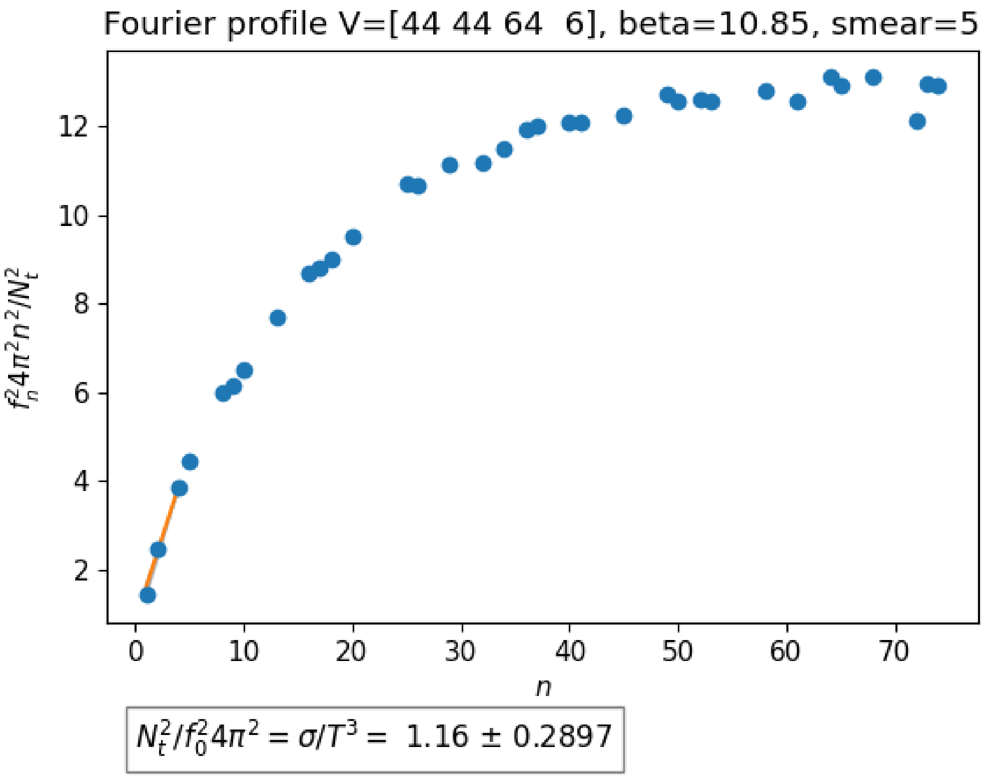
[66]:

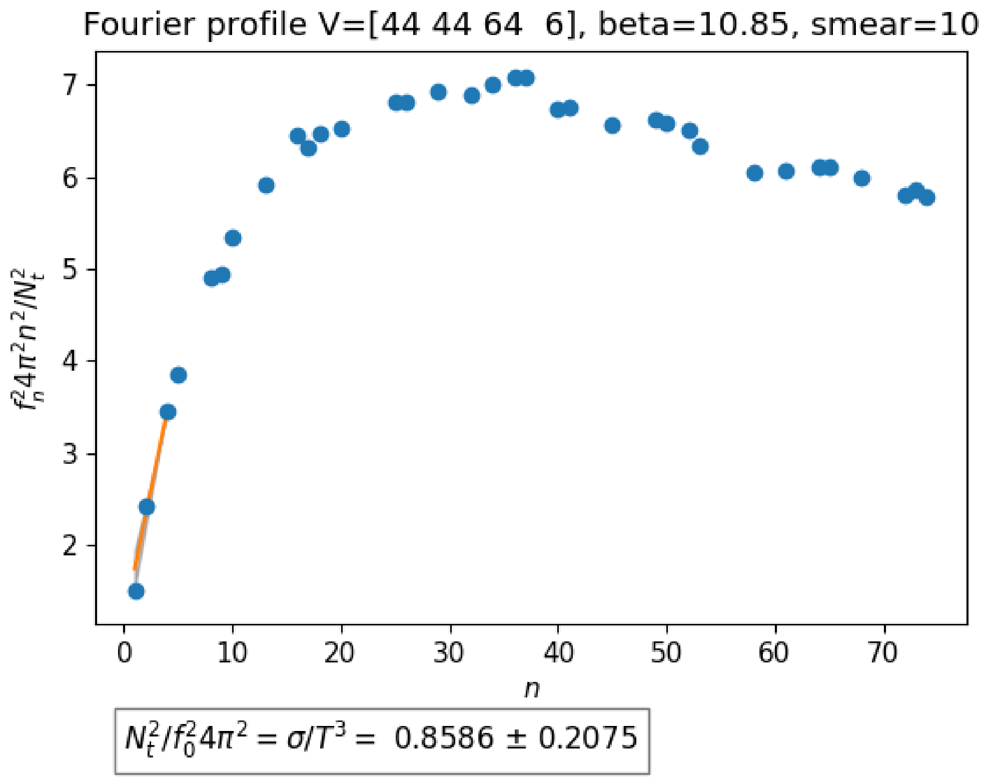
```
n_2_list = []
f_n_list = []
errors_list = []
for smearing_level, profile in fourier_profiles.items():
    n_2, f_n, errors = utility.compute_with_aa_jackknife_fourier(profile, 10,␣
 ↪thermalization=100)
    n_2_list.append(n_2)
    f_n_list.append(f_n)
    errors_list.append(errors)
```

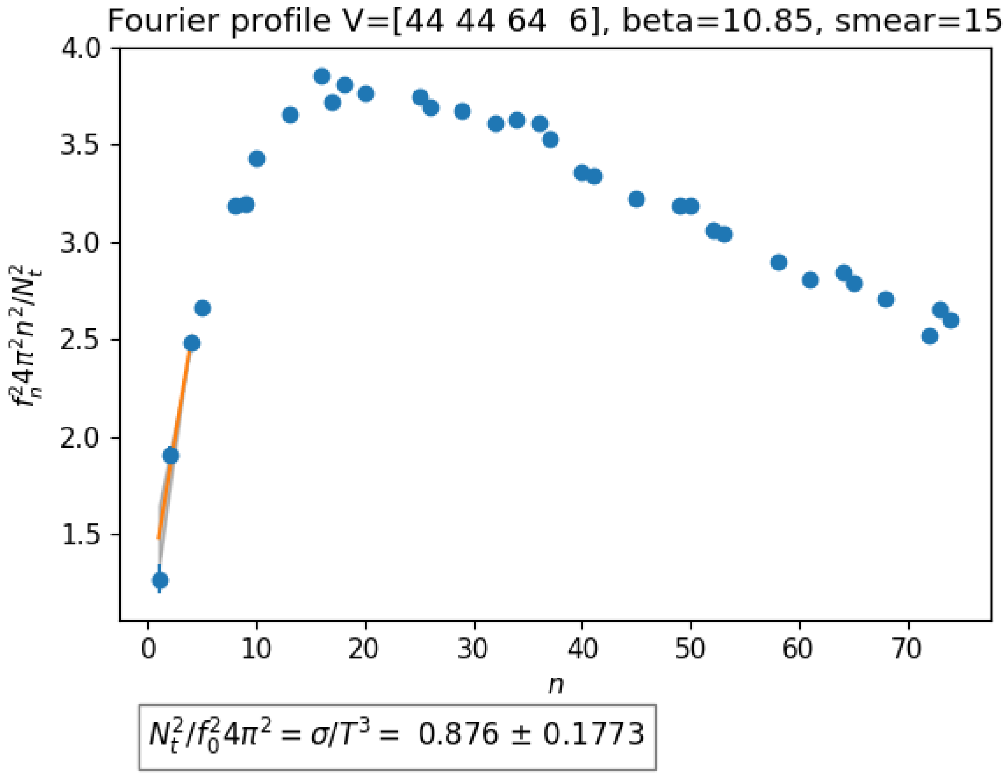
2.2 Plot Fourier modes for different smearing steps

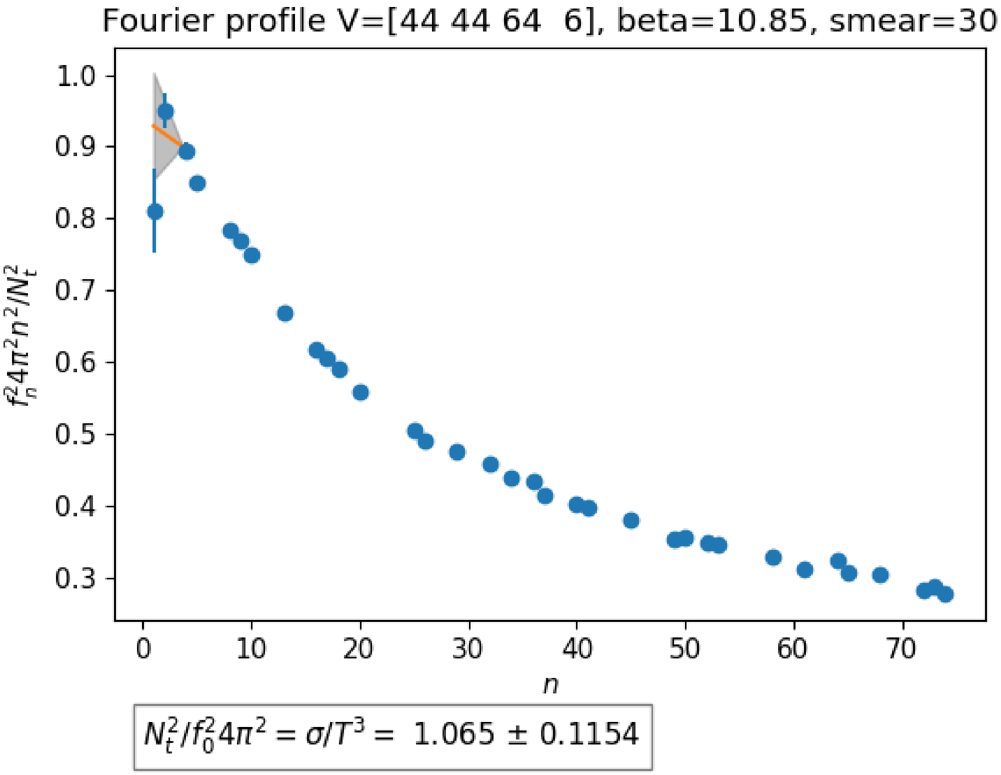
[68]:

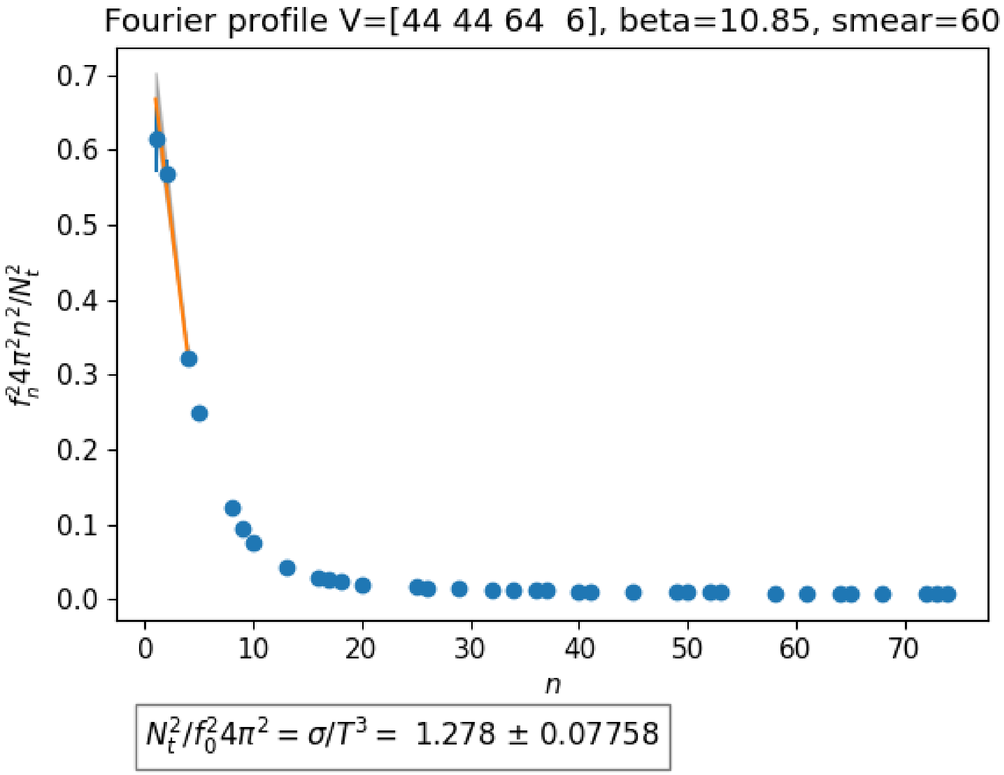
```
%matplotlib widget
smearing_levels = list(fourier_profiles.keys())
show_plot = True
data = {
    "smearing": smearing_levels,
    "linear": [
        fourier_surface.compute_fourier_profile(
            n_2, f_n, volume, errors=error, beta=10.85, fit_range=3,␣
 ↪smearing=smear, show_plot=show_plot
        ) for n_2, f_n, error, smear in zip(n_2_list, f_n_list, errors_list,␣
 ↪smearing_levels)
    ],
    "exponential": [
        fourier_surface.compute_fourier_profile_exponential_fit(
            n_2, f_n, volume, errors=error, beta=10.85, smearing=smear,␣
 ↪show_plot=show_plot
        ) for n_2, f_n, error, smear in zip(n_2_list, f_n_list, errors_list,␣
 ↪smearing_levels)
    ]
}

df = pd.DataFrame(data)
utility.print_df_as_markdown_fourier_modes(df)
```

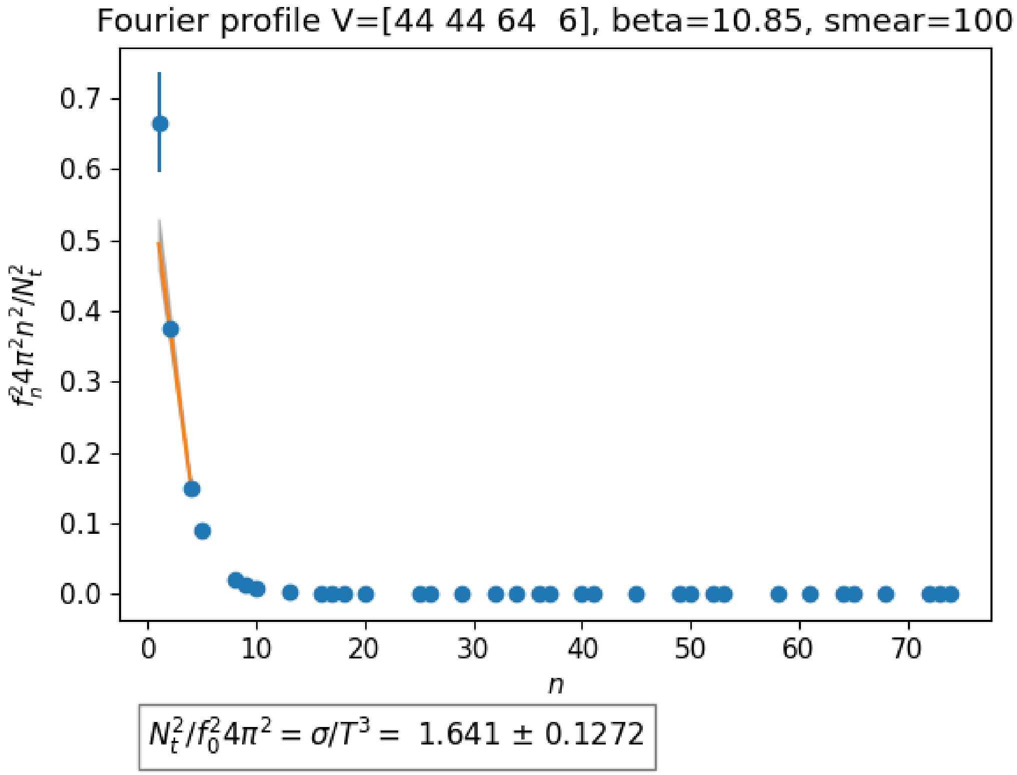


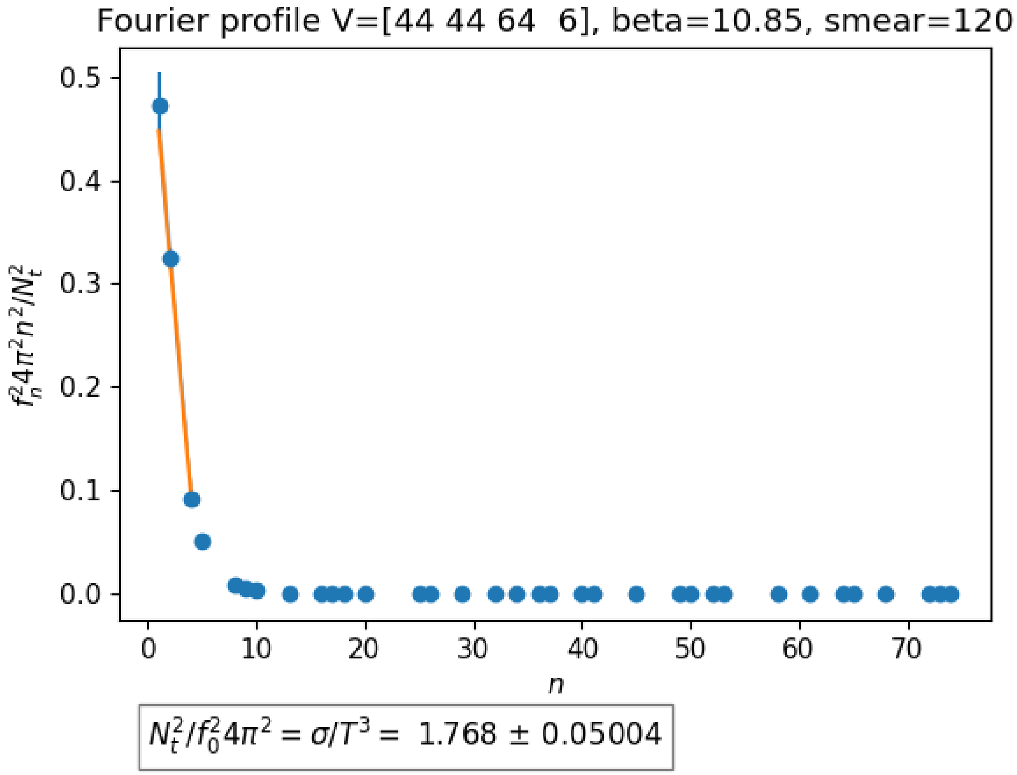

Fourier profile V=[44 44 64 6], beta=10.85, smear=10
$f_n^2 4\pi^2 n^2/N_t^2$
1
2
3
4
5
6
7
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 0.8586 ± 0.2075

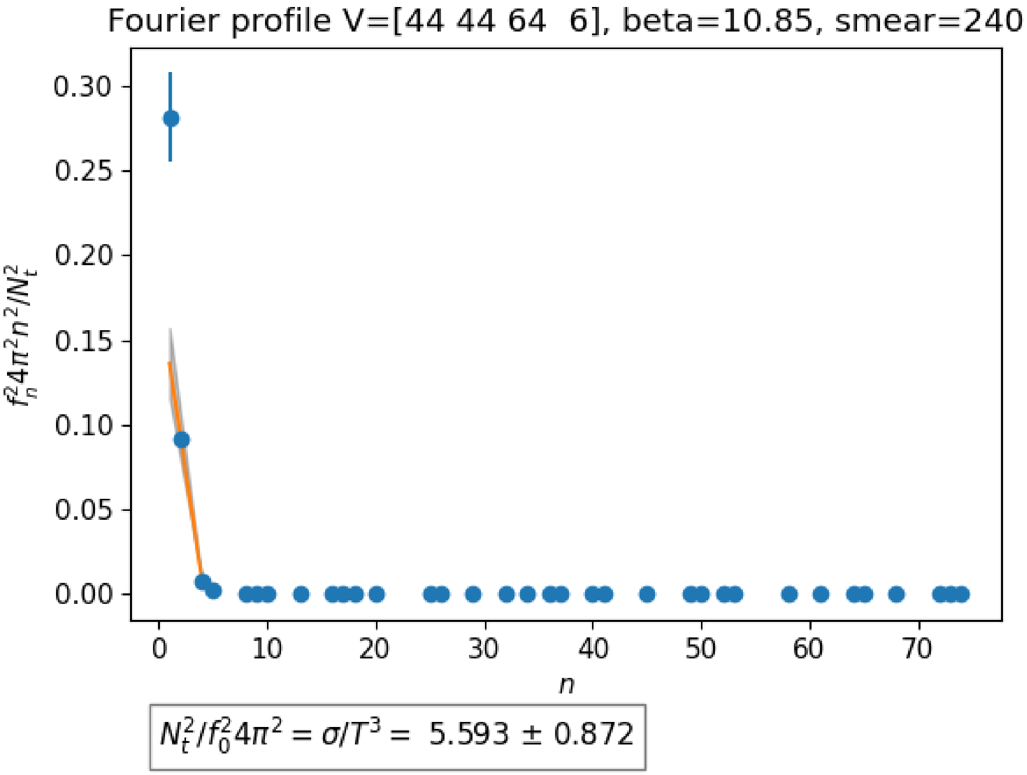

Fourier profile V=[44 44 64 6], beta=10.85, smear=15
$f_n^2 4\pi^2 n^2/N_t^2$
4.0
3.5
3.0
2.5
2.0
1.5
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 0.876 ± 0.1773

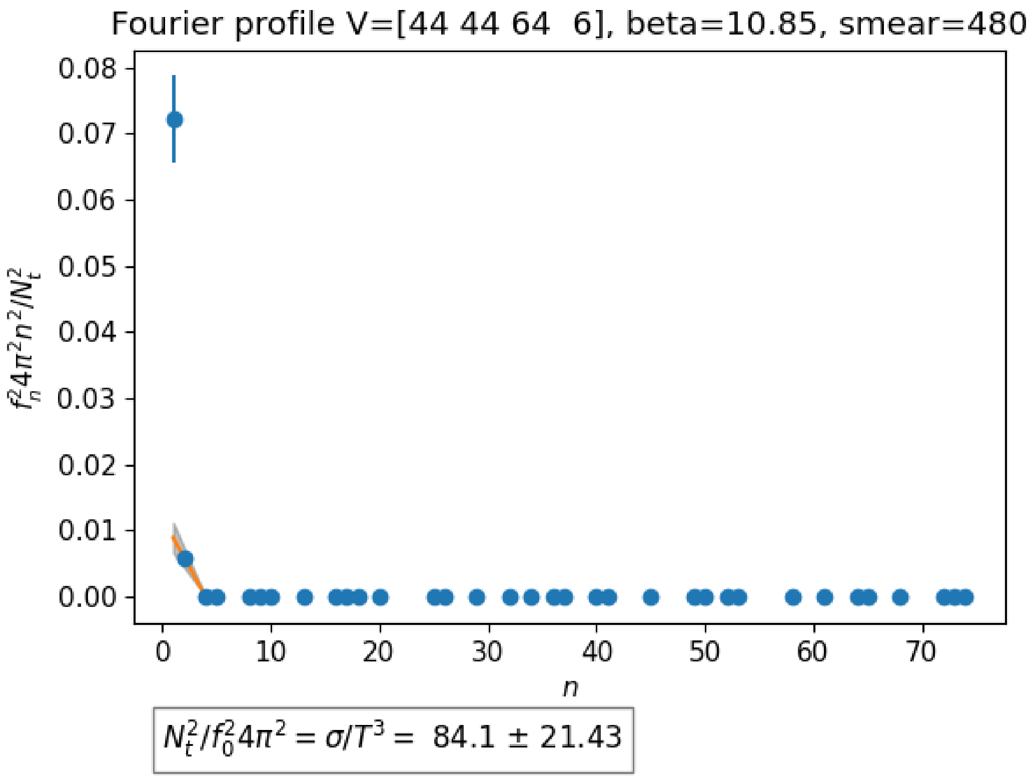

Fourier profile V=[44 44 64 6], beta=10.85, smear=30
$f_n^2 4\pi^2 n^2/N_t^2$
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.065 ± 0.1154


Fourier profile V=[44 44 64 6], beta=10.85, smear=60
$f_n^2 4\pi^2 n^2/N_t^2$
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.278 ± 0.07758

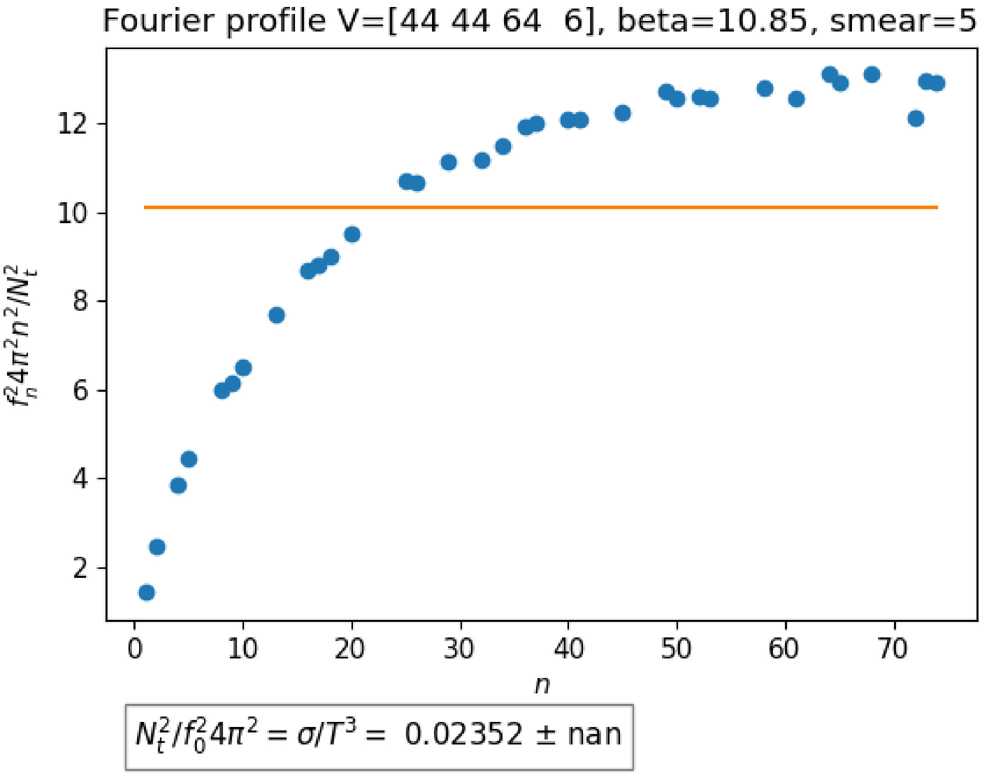

Fourier profile V=[44 44 64 6], beta=10.85, smear=100
$f_n^2 4\pi^2 n^2/N_t^2$
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
n
0
10
20
30
40
50
60
70
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.641 ± 0.1272

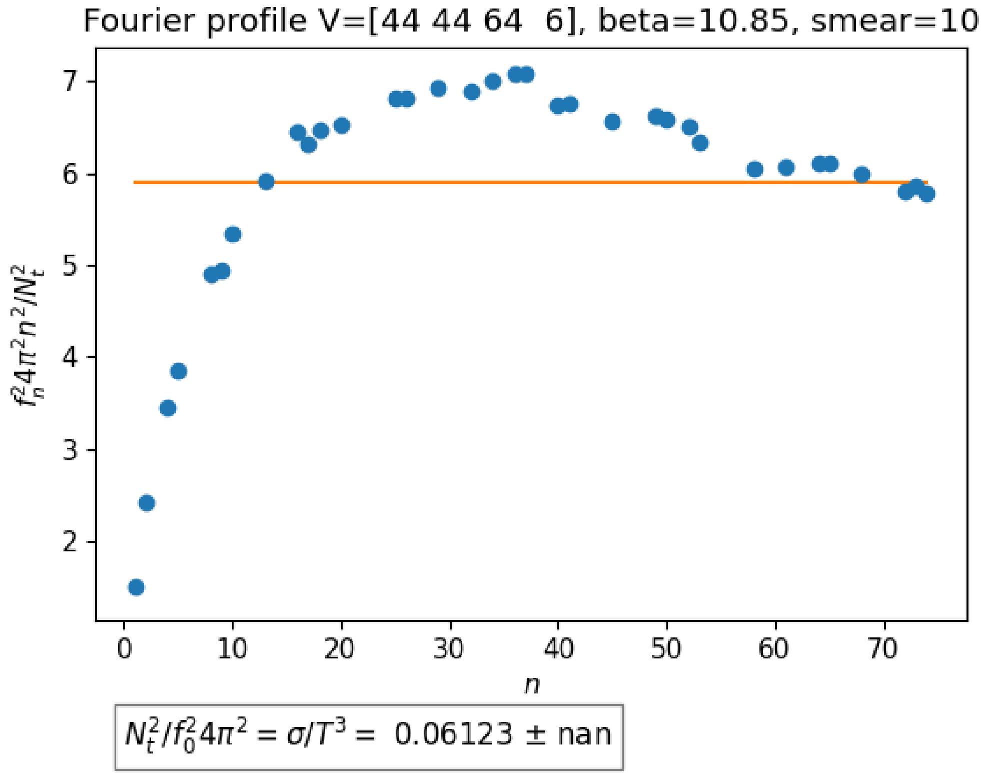

Fourier profile V=[44 44 64 6], beta=10.85, smear=120
$f_n^2 4\pi^2 n^2/N_t^2$
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.768 ± 0.05004

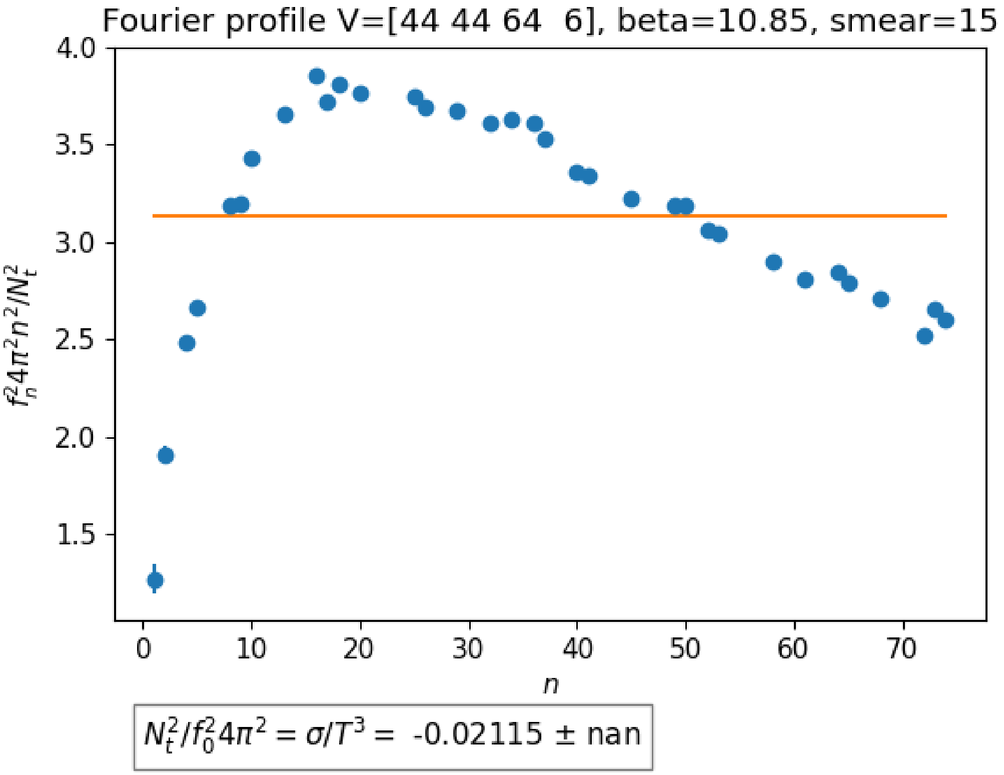

Fourier profile V=[44 44 64 6], beta=10.85, smear=240
$f_n^2 4\pi^2 n^2/N_t^2$
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 5.593 ± 0.872

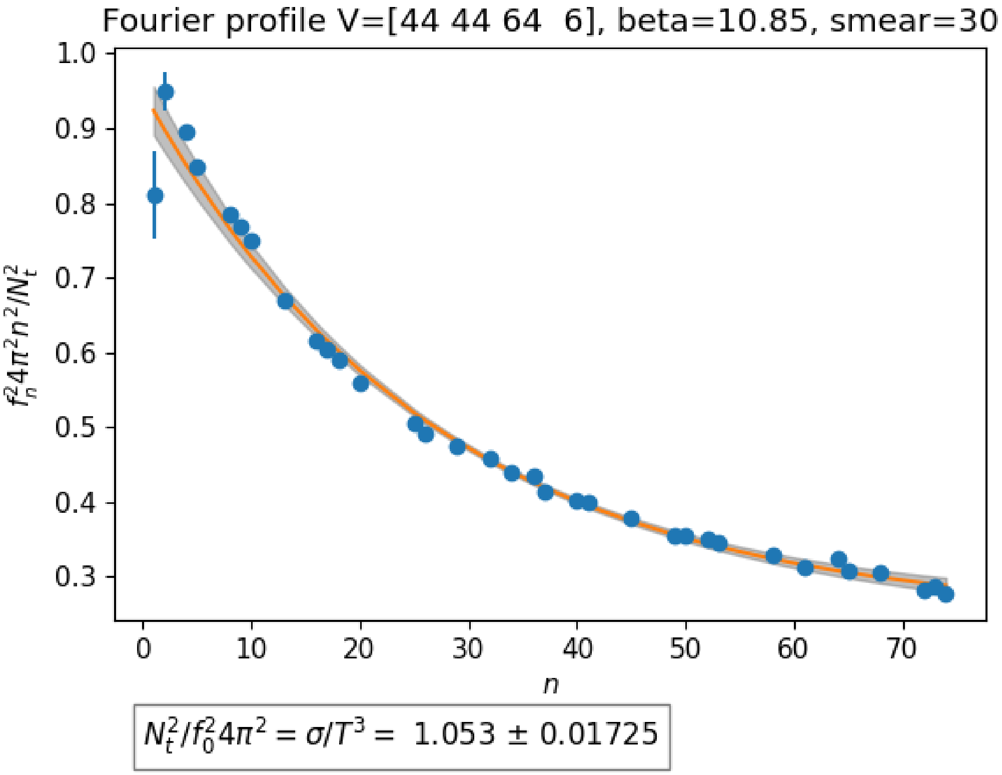


```
/home/haaaaron/SUN_twist_python_analysis/modules/fourier_surface.py:93:
RuntimeWarning: invalid value encountered in multiply
  return a * np.exp(-b * x) + c
/home/haaaaron/SUN_twist_python_analysis/modules/fourier_surface.py:153:
RuntimeWarning: invalid value encountered in matmul
```

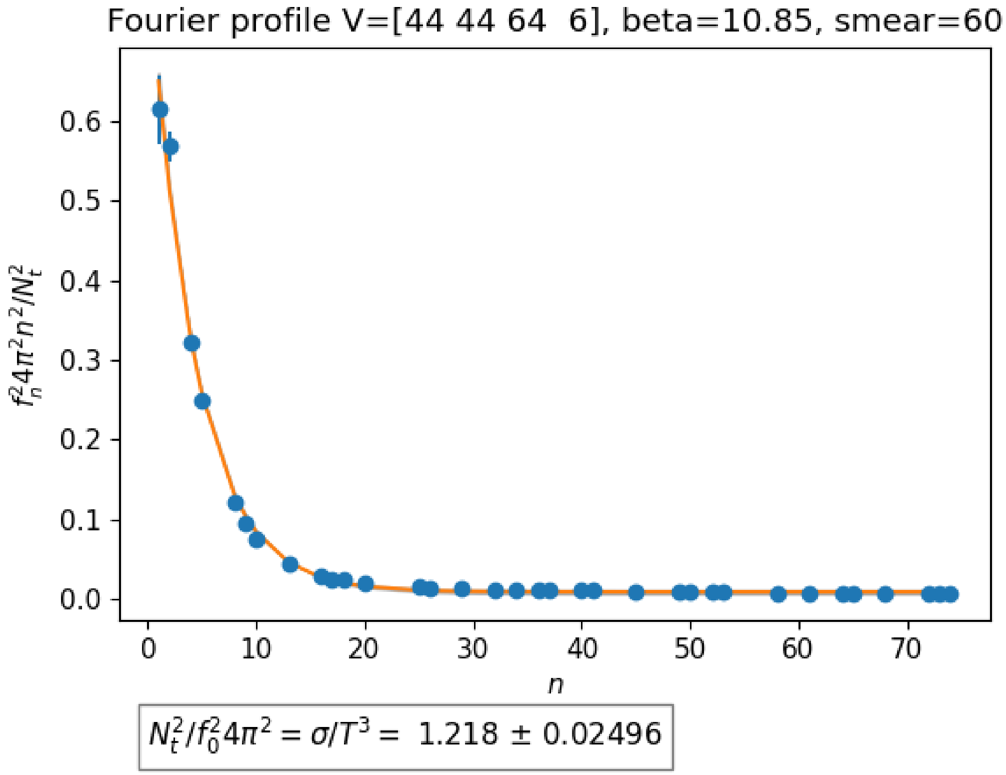

Fourier profile V=[44 44 64 6], beta=10.85, smear=5
$f_n^2 4\pi^2 n^2/N_t^2$
2
4
6
8
10
12
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 0.02352 ± nan

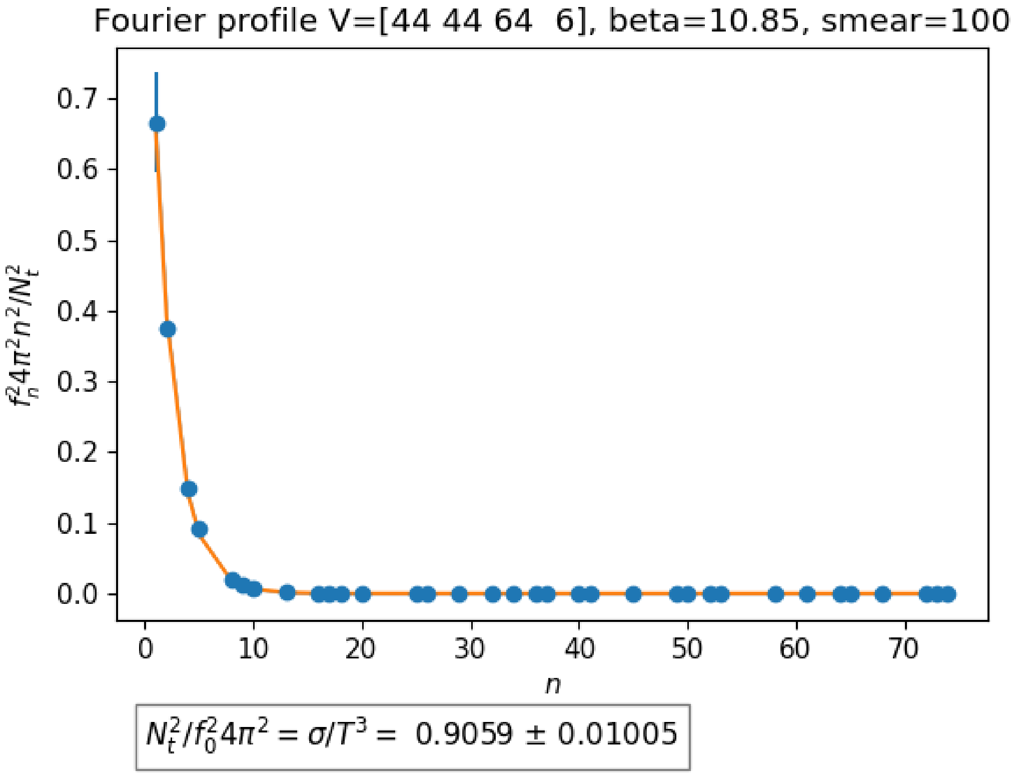

Fourier profile V=[44 44 64 6], beta=10.85, smear=10
$f_n^2 4\pi^2 n^2 / N_t^2$
7
6
5
4
3
2
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 0.06123 ± nan

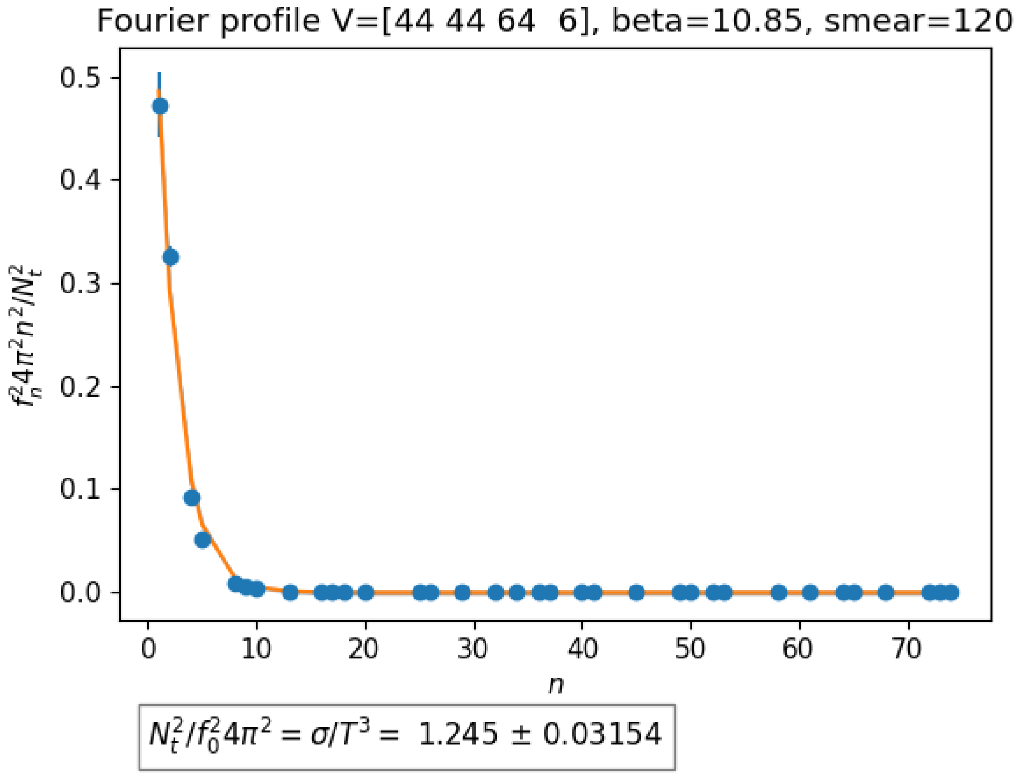

Fourier profile V=[44 44 64 6], beta=10.85, smear=15
4.0
3.5
3.0
2.5
2.0
1.5
$f_n^2 4\pi^2 n^2/N_t^2$
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ -0.02115 ± nan

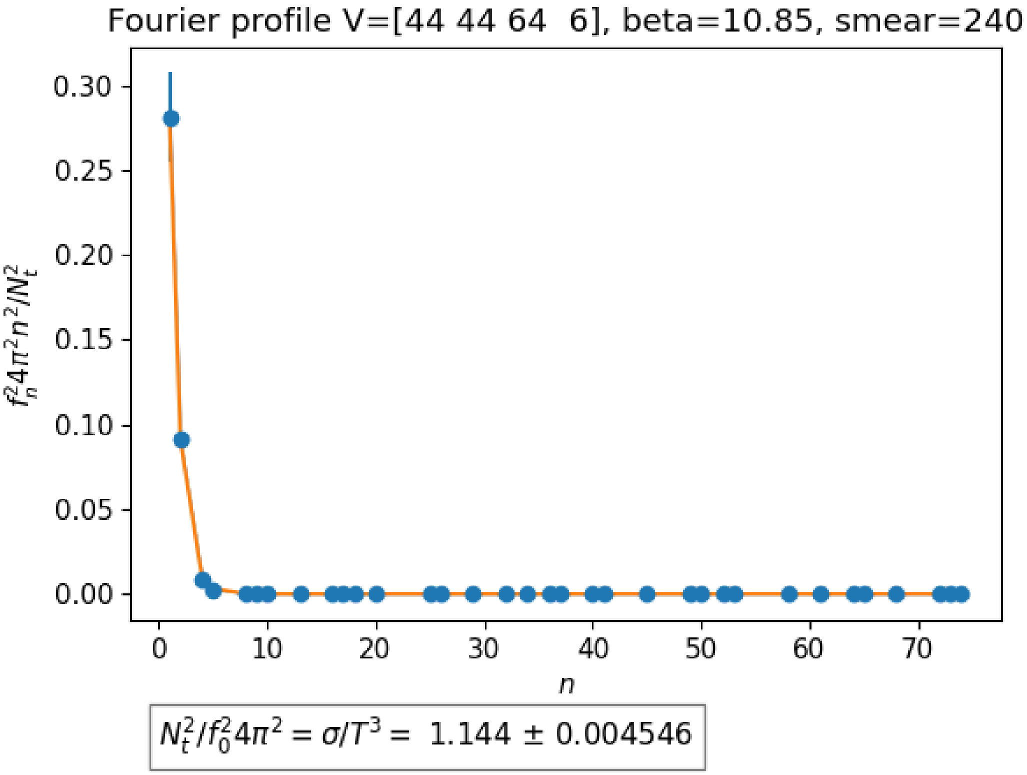


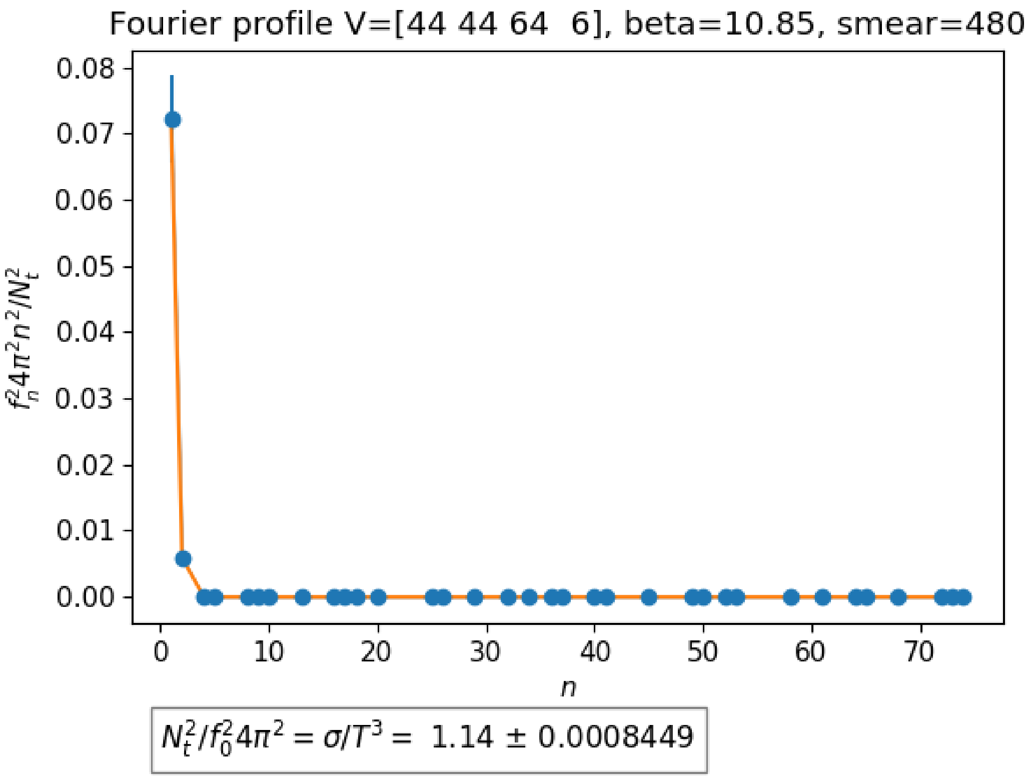
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.053 ± 0.01725


Fourier profile V=[44 44 64 6], beta=10.85, smear=60
$f_n^2 4\pi^2 n^2/N_t^2$
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.218 ± 0.02496


Fourier profile V=[44 44 64 6], beta=10.85, smear=100
$f_n^2 4\pi^2 n^2/N_t^2$
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 0.9059 ± 0.01005


Fourier profile V=[44 44 64 6], beta=10.85, smear=120
$f_n^2 4\pi^2 n^2/N_t^2$
0.0
0.1
0.2
0.3
0.4
0.5
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.245 ± 0.03154


Fourier profile V=[44 44 64 6], beta=10.85, smear=240
$f_n^2 4\pi^2 n^2/N_t^2$
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
n
$N_t^2/f_0^2 4\pi^2 = \sigma/T^3 =$ 1.144 ± 0.004546



smearing	Linear fit (σ/T^3)	Exponential fit (σ/T^3)
5	1.16 ± 0.2897	0.02352 ± nan
10	0.8586 ± 0.2075	0.06123 ± nan
15	0.876 ± 0.1773	-0.02115 ± nan
30	1.065 ± 0.1154	1.053 ± 0.01725
60	1.278 ± 0.07758	1.218 ± 0.02496
100	1.641 ± 0.1272	0.9059 ± 0.01005
120	1.768 ± 0.05004	1.245 ± 0.03154
240	5.593 ± 0.872	1.144 ± 0.004546
480	84.1 ± 21.43	1.14 ± 0.0008449

[45]:

```
import matplotlib.pyplot as plt

plt.close('all')
```

[ ]:

```
```