44-44-44-64-6-beta-10.85-twist-2

October 23, 2024

[15]:
```
%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

[3]:
```
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",
                "../data/output-measure-surface/su4-44-44-64-6/beta-10.
 ↪85-twist-1-small-smear-range"]
choose_folder = 5
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])))
utility.display_markdown_title(folder)
```

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

2.1 Perform post processing

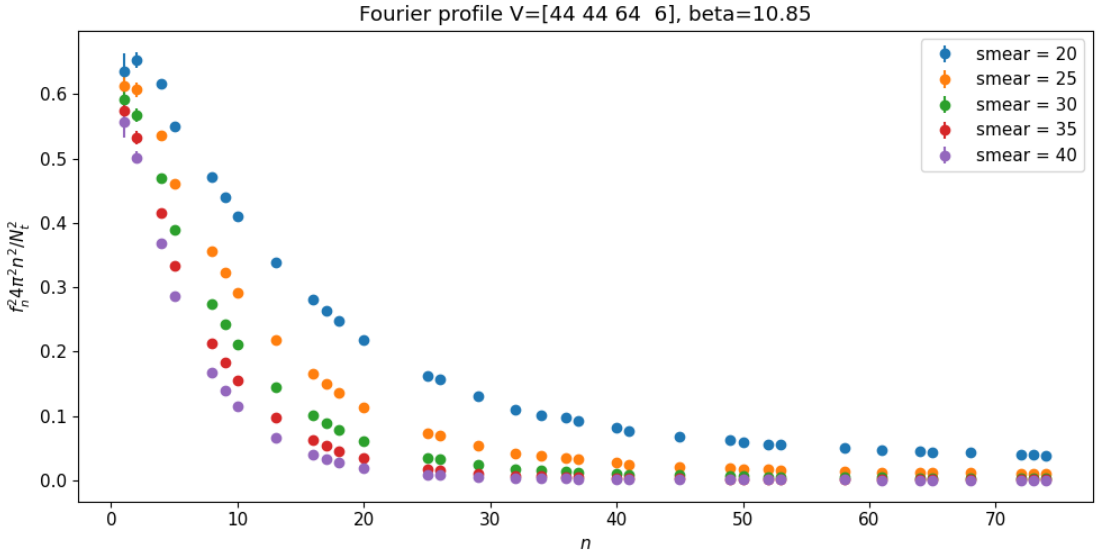
[4]:
```
n_2_list = []
f_n_list = []
errors_list = []
for smearing_level, profile in fourier_profiles.items():
    n_2, f_n, errors = utility.compute_with_aa_jackknife_fourier(profile, 10,␣
 ↪thermalization=100)
    n_2_list.append(n_2)
    f_n_list.append(f_n)
    errors_list.append(errors)
```

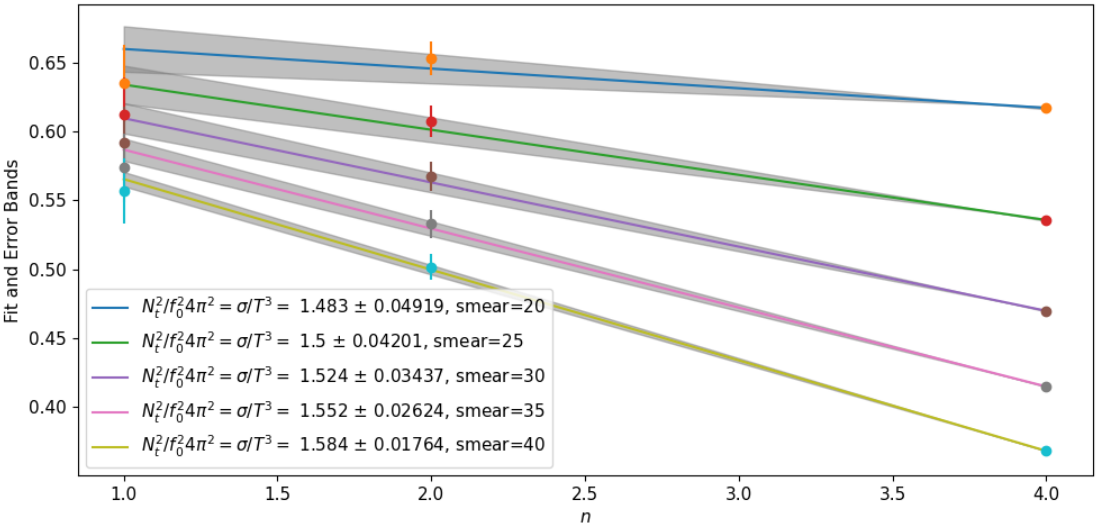
2.2 Plot Fourier modes for different smearing steps

[21]:
```
%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_surface.fig = None
```

smearing	Linear fit (σ/T^3)
20	1.483 ± 0.04919
25	1.5 ± 0.04201
30	1.524 ± 0.03437
35	1.552 ± 0.02624
40	1.584 ± 0.01764





[8]:
```
import matplotlib.pyplot as plt

plt.close('all')
```

[ ]: