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

October 30, 2024

[6]:
```
%load_ext autoreload
%autoreload 2
from modules import read_and_write
from modules import polyakov
from modules import utility
from modules import fourier_surface
from modules import surface_amplitudes as sf

import pandas as pd
import numpy as np
import os
import glob
indices = None
```

```
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

[10]:
```
from modules.globals import folder_names

choose_folder = 2
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, modes, fourier_profile = read_and_write.
 ↪read_fourier_profile(folder,file_name=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'], β = 10.85, twist coeff = 2

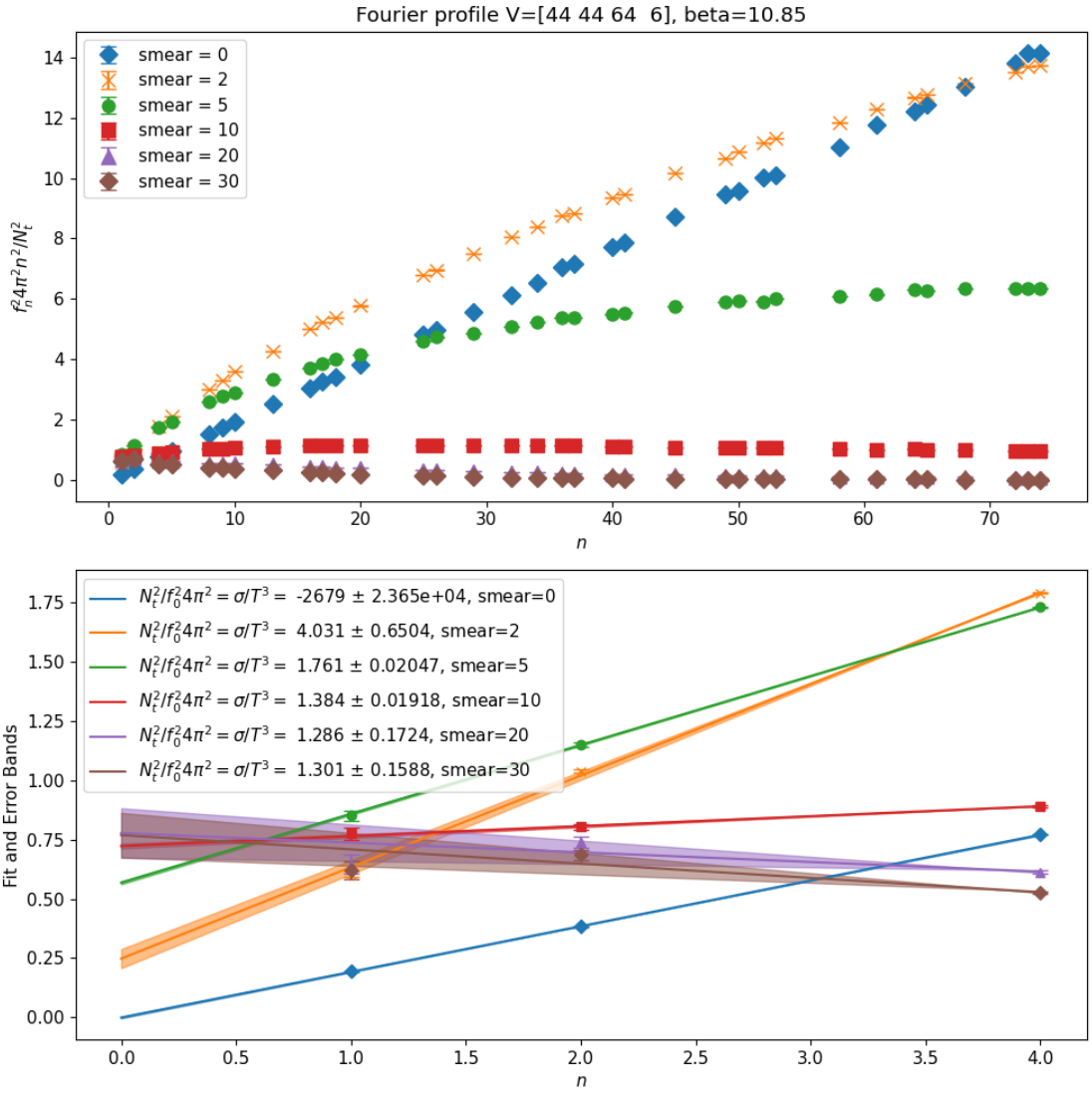
2.1 Perform post processing

[11]:
```
f_n_list = []
errors_list = []
for smearing_level, profile in fourier_profiles.items():
    if indices is not None:
        sample_size = len(profile)
        indices_set = indices[smearing_level]
        profile = np.delete(profile, list(indices_set), axis=0)
        print(f"Dropped {sample_size-len(profile)} samples")
    f_n, errors = utility.compute_with_aa_jackknife_fourier(profile, 10,␣
 ↪thermalization=1000)
    f_n_list.append(f_n)
    errors_list.append(errors)
```

2.2 Plot Fourier modes for different smearing steps

[13]:
```
%matplotlib widget
smearing_levels = list(fourier_profiles.keys())
show_plot = True
data = {
    "smearing": smearing_levels,
    "linear": [
        fourier_surface.compute_fourier_profile(
            modes, f_n, volume, errors=error, beta=10.85, fit_range=3,␣
 ↪smearing=smear, show_plot=show_plot
        ) for f_n, error, smear in zip( 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.global_fig = None
```

smearing	Linear fit (σ/T^3)
0	-2679 $\pm$ 2.365e+04
2	4.031 $\pm$ 0.6504
5	1.761 $\pm$ 0.02047
10	1.384 $\pm$ 0.01918
20	1.286 $\pm$ 0.1724
30	1.301 $\pm$ 0.1588



[ ]:

[ ]: