

44-44-44-64-6-beta-13-twist-2

November 6, 2024

```
[2]: %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
from modules import globals
import numpy as np
import matplotlib.pyplot as plt
import os
import glob
```

1 Load data

```
[22]: folders = utility.list_all_folders(globals.data_path, "test")
```

```
../data/output-measure-surface/test/beta-13-twist-2-60-60-86-6-test , index: 0
```

```
[23]: smooth_surfaces= {}
choose_folder = 0
folder = folders[choose_folder]
files = glob.glob(os.path.join(folder, "surface_smooth_*"))
for file in files:
    file_name = file.split("/")[-1]
    smearing_level = file_name.split("_")[-1]
    volume, surface = read_and_write.read_surface_data(folder, file_name)
    smooth_surfaces[smearing_level] = surface
```

```
[29]: smooth_surfaces = dict(sorted(smooth_surfaces.items(), key=lambda item:
    ↪int(item[0])))
smooth_surfaces
```

```
[29]: {'0': array([[ 0.    ,  0.    , 27.8602],
                  [ 1.    ,  0.    , -28.0798],
                  [ 2.    ,  0.    , 28.768 ]]),
      ...}
```

```

[ 57.    , 59.    , 11.2822],
[ 58.    , 59.    , 12.1477],
[ 59.    , 59.    , 21.074 ]],

[[ 0.    , 0.    , -40.1674],
 [ 1.    , 0.    , -20.1646],
 [ 2.    , 0.    , 39.4527],
 ...,
 [ 57.    , 59.    , -32.6829],
 [ 58.    , 59.    , 17.2817],
 [ 59.    , 59.    , -35.7591]],

[[ 0.    , 0.    , 24.0175],
 [ 1.    , 0.    , -17.7328],
 [ 2.    , 0.    , -22.9249],
 ...,
 [ 57.    , 59.    , -40.6091],
 [ 58.    , 59.    , 31.1378],
 [ 59.    , 59.    , 40.4098]],

...,

[[ 0.    , 0.    , 110.995 ],
 [ 1.    , 0.    , 52.0156],
 [ 2.    , 0.    , 119.535 ]],
...,
[ 57.    , 59.    , 93.3597],
[ 58.    , 59.    , 87.9487],
[ 59.    , 59.    , 117.414 ]],

[[ 0.    , 0.    , 77.9474],
 [ 1.    , 0.    , 95.7479],
 [ 2.    , 0.    , 78.1821],
 ...,
 [ 57.    , 59.    , 92.1526],
 [ 58.    , 59.    , 51.097 ],
 [ 59.    , 59.    , 78.9117]],

[[ 0.    , 0.    , 77.0499],
 [ 1.    , 0.    , 58.1037],
 [ 2.    , 0.    , 43.1244],
 ...,
 [ 57.    , 59.    , 83.8691],
 [ 58.    , 59.    , 80.8452],
 [ 59.    , 59.    , 121.702 ]]]}]

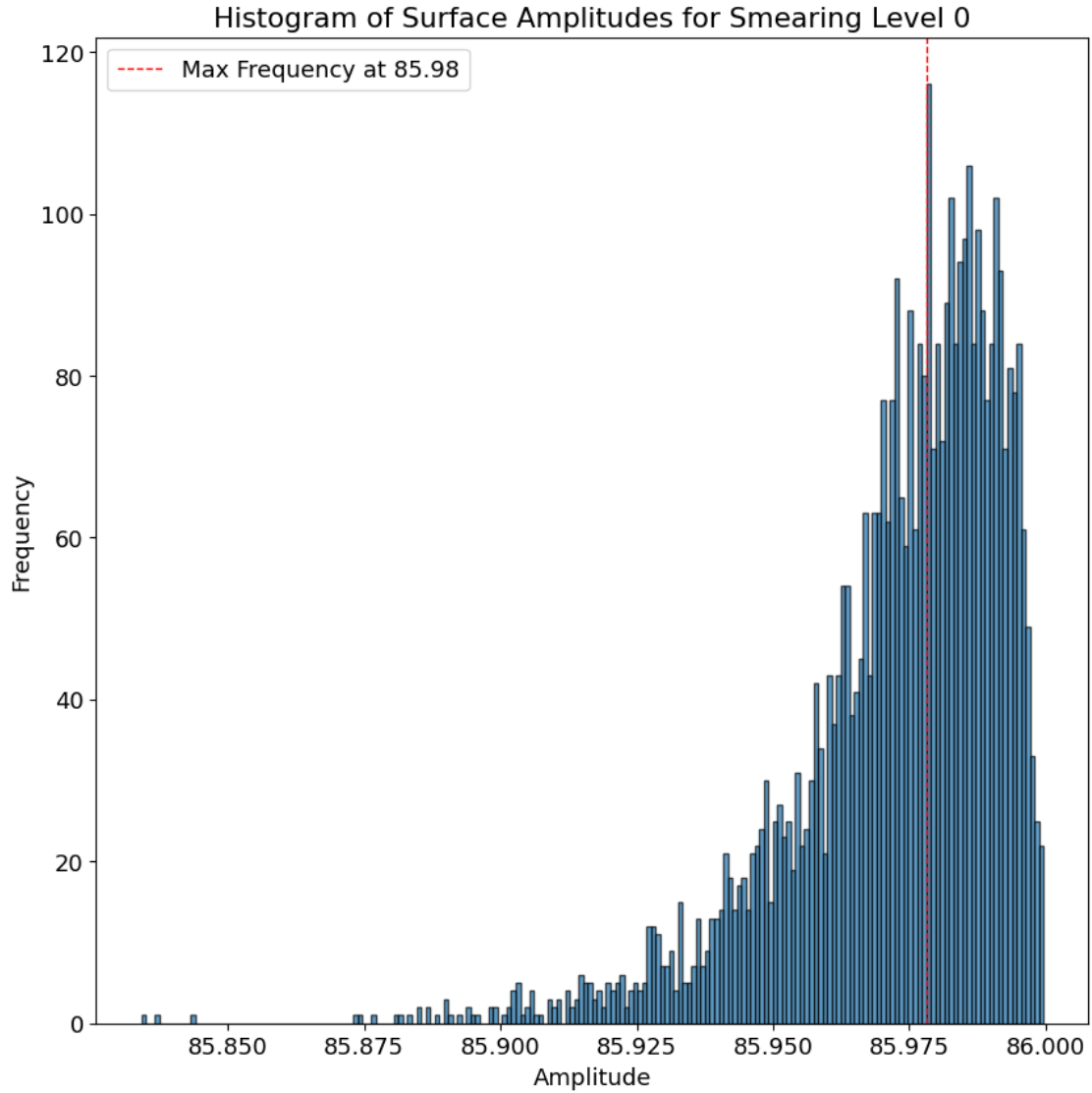
```

```
[26]: utility.display_markdown_title(folder)
```

2 $SU(s)$, $V = []$, $\beta = 13$, twist coeff = 2

```
[30]: indices = sf.surface_amplitudes(smooth_surfaces=smooth_surfaces,
    ↪return_threshold=40,thermalization=10)
```

Smearing Level: 0



```
{'0': (85.97292317565962, 85.8341, 85.9996)}
```

```
[32]: # Create a dictionary to store fluctuations for each smearing level
fluctuations_dict = {}
thermalization = 1000
plt.rcParams.update({'font.size': 13})
```

```

for smearing_level, surface_data in smooth_surfaces.items():
    post_thermalization_data = surface_data[thermalization:thermalization+10000]
    mean_z_values = np.mean(post_thermalization_data[:, :, 2], axis=1)
    fluctuations = post_thermalization_data[:, :, 2] - mean_z_values[:, np.newaxis]
    fluctuations_dict[smearing_level] = fluctuations.flatten()

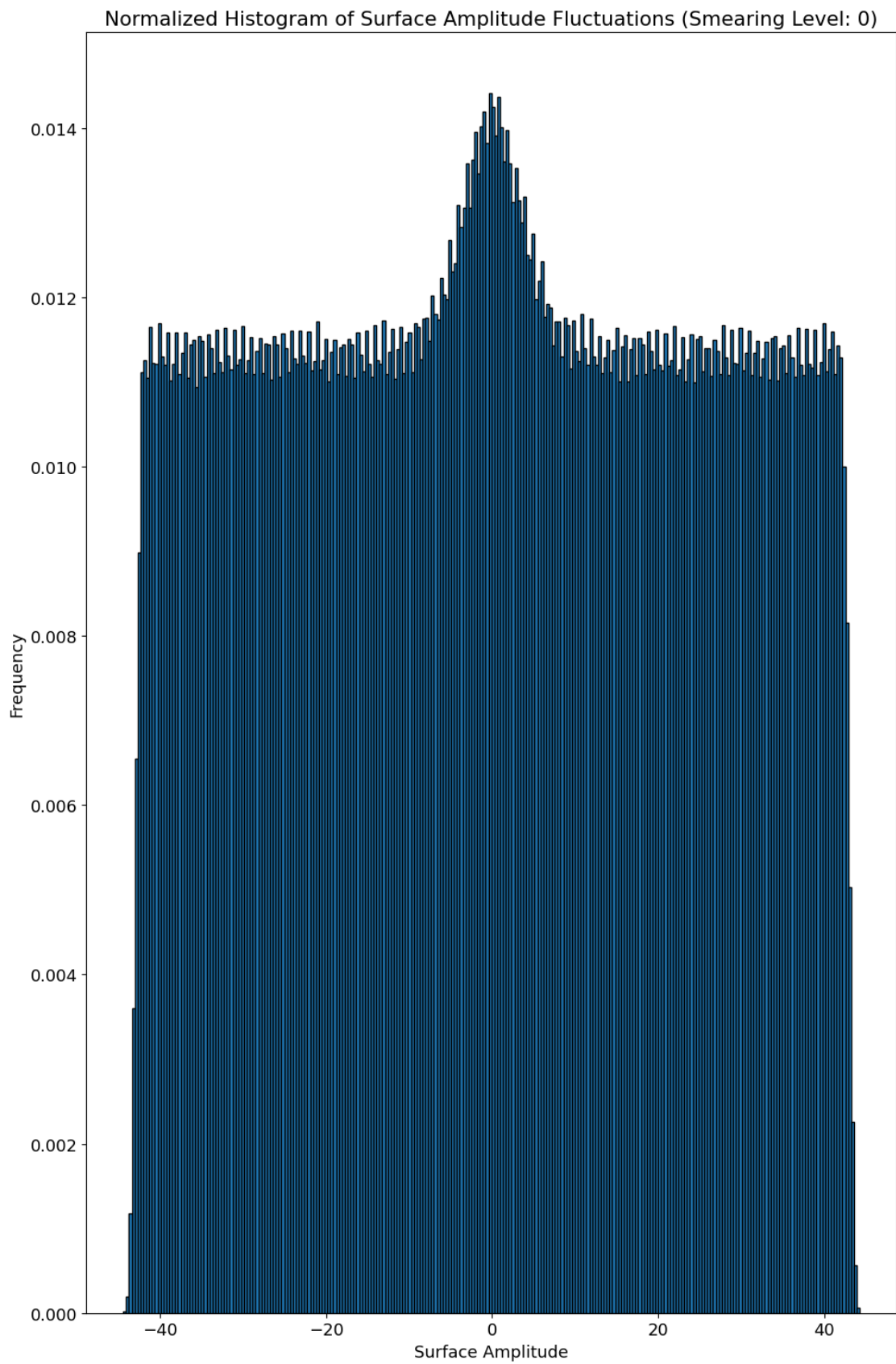
# Plotting all histograms in subplots
num_plots = len(fluctuations_dict)
fig, axes = plt.subplots(num_plots, 1, figsize=(10,15))

# If there is only one histogram to be plotted, axes is not an array
if num_plots == 1:
    axes = [axes]

for ax, (smearing_level, fluctuations) in zip(axes, fluctuations_dict.items()):
    ax.hist(fluctuations, edgecolor='black', bins=64*4, density=True)
    ax.set_xlabel('Surface Amplitude')
    ax.set_ylabel('Frequency')
    ax.set_title(f'Normalized Histogram of Surface Amplitude Fluctuations_
    ↪(Smearing Level: {smearing_level})')

plt.tight_layout()
plt.show()

```



[]:

[]: