# 44-44-44-64-6-beta-10.85-twist-1-small-smearing-range

October 22, 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
import numpy as np
import matplotlib.pyplot as plt
import os
import glob
```

## 1 Load data

[7]:
```
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"]
smooth_surfaces= {}
choose_folder = 6
folder = folder_names[choose_folder-1]
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
```

[8]:
```
smooth_surfaces = dict(sorted(smooth_surfaces.items(), key=lambda item:␣
 ↪int(item[0])))
```

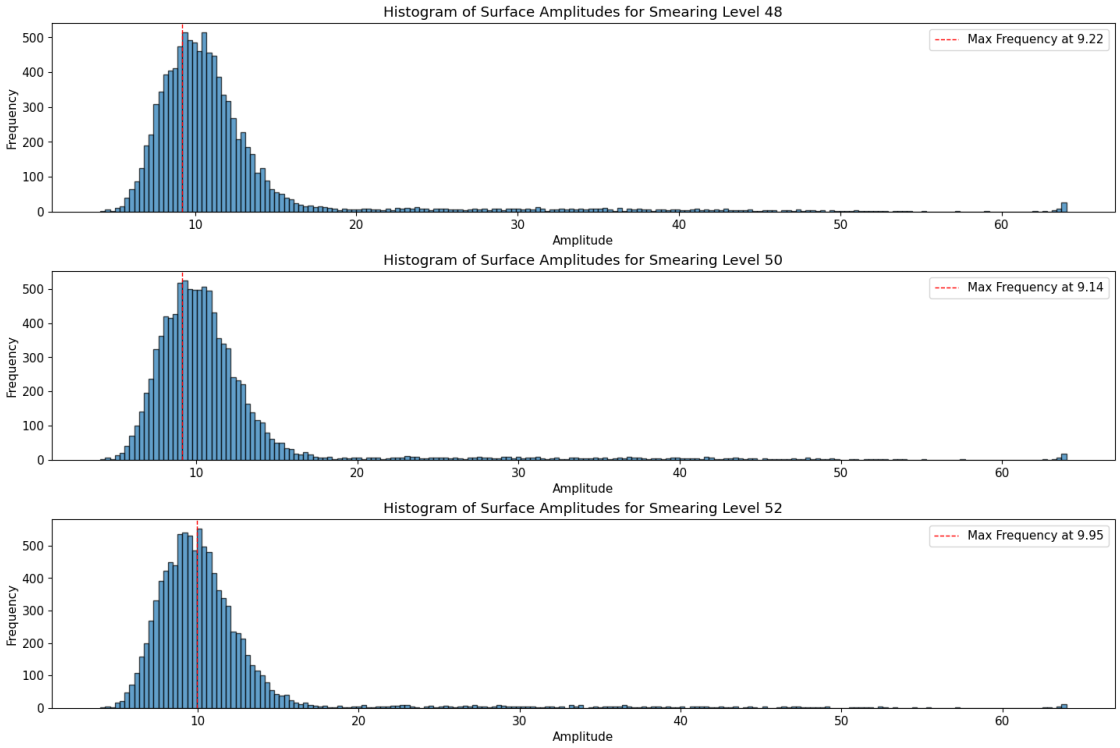
[9]:
```
utility.display_markdown_title(folder)
```

## 2 SU(4), $V = [\text{'44'}, \text{'44'}, \text{'64'}, \text{'6'}]$, $\beta = 10.85$, twist coeff $= 1$

[13]:
```
surface_amplitudes = {}
thermalization = 100
for smearing_level, surface in smooth_surfaces.items():
    print(f"Smearing Level: {smearing_level}")
    surface_amplitudes[smearing_level] = np.array([np.abs(instance["z"].max() -␣
 ↪instance["z"].min()) for instance in surface[100:]])

for level, amplitudes in surface_amplitudes.items():
    indices = np.where(amplitudes > 60)
    print(f"Smearing Level: {level}, Indices: {indices}")

fig, axs = plt.subplots(len(surface_amplitudes), figsize=(15, 10))
fig.tight_layout(pad=3.0)

for ax, (level, amplitudes) in zip(axs.flatten(), surface_amplitudes.items()):
    counts, bins, patches = ax.hist(amplitudes, bins=200, edgecolor='black',␣
 ↪alpha=0.7)
    max_bin = bins[np.argmax(counts)]
    ax.axvline(max_bin, color='r', linestyle='dashed', linewidth=1)
    ax.set_xlabel('Amplitude')
    ax.set_ylabel('Frequency')
    ax.set_title(f'Histogram of Surface Amplitudes for Smearing Level {level}')
    ax.legend([f'Max Frequency at {max_bin:.2f}'])

# Hide any unused subplots
for i in range(len(surface_amplitudes), len(axs.flatten())):
    fig.delaxes(axs.flatten()[i])

plt.show()
average_surface_amplitudes = {level: (np.mean(amplitudes),np.min(amplitudes),␣
 ↪np.max(amplitudes)) for level, amplitudes in surface_amplitudes.items()}
average_surface_amplitudes
```

```
Smearing Level: 48
Smearing Level: 50
Smearing Level: 52
Smearing Level: 48, Indices: (array([ 547,  548,  623,  811,  963, 1689, 1857,
2029, 2082, 2085, 2086,
       2868, 2876, 3003, 3044, 3372, 3505, 3508, 3626, 3875, 4361, 4362,
       4680, 5377, 5676, 5680, 5952, 5965, 5981, 6638, 7658, 7835, 8488,
       8495, 8497, 8499, 9061, 9211, 9895]),)
Smearing Level: 50, Indices: (array([ 548,  623,  811,  963, 1689, 2082, 2085,
```

```
2086, 2868, 3044, 3505,
       3508, 3626, 3875, 4362, 5676, 5680, 5952, 5981, 6638, 7658, 8495,
       8497, 8499, 9211, 9895]),)
Smearing Level: 52, Indices: (array([ 548,  623,  811,  963, 2085, 2086, 2868,
3044, 3626, 3875, 4362,
       5680, 5952, 5981, 8495, 8497, 8499, 9211, 9895]),)
```




[13]:
```
{'48': (11.918305494949495, 4.136799999999994, 63.9917),
 '50': (11.354866696969697, 4.040800000000004, 63.99159999999999),
 '52': (10.868840272727272, 3.9498999999999995, 63.9756)}
```

[ ]: