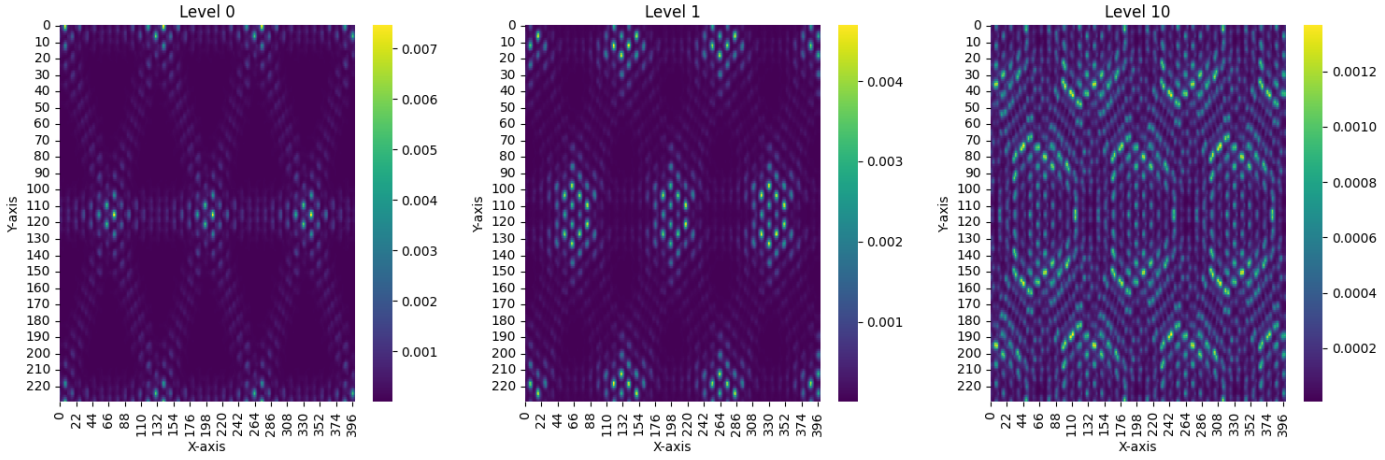
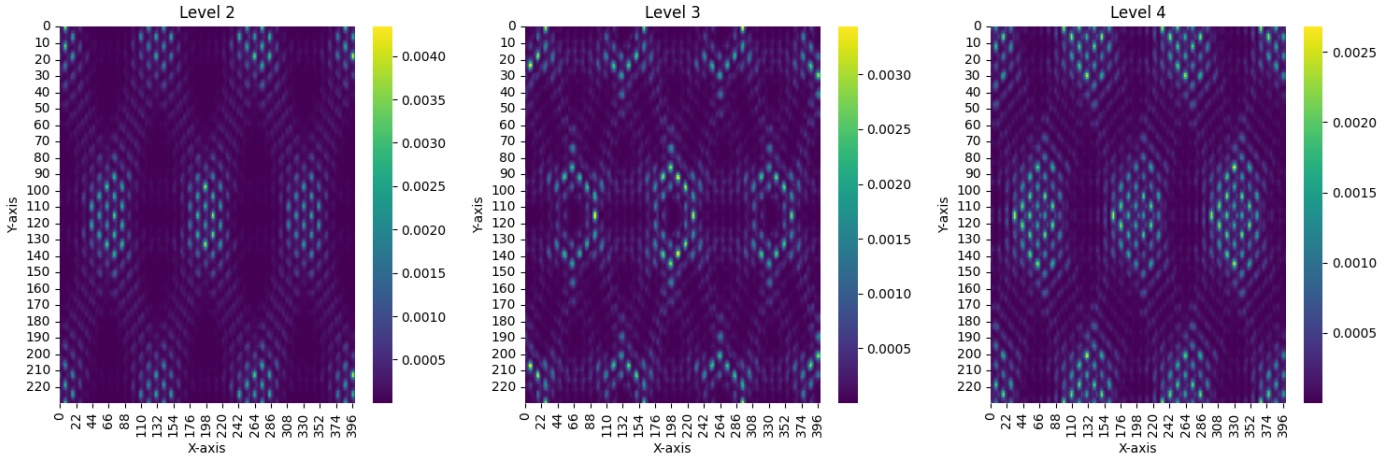
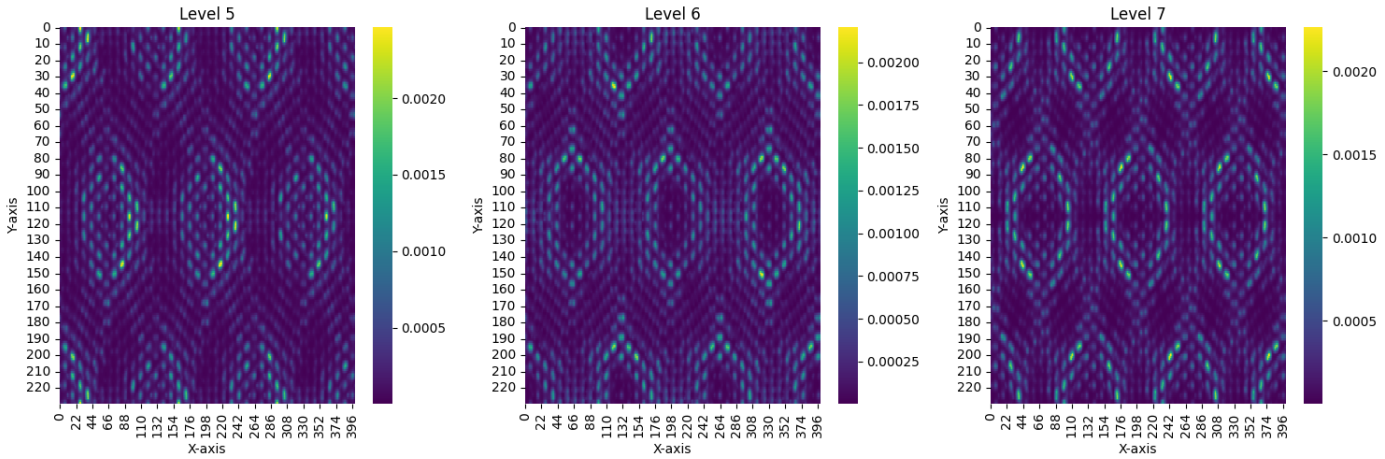
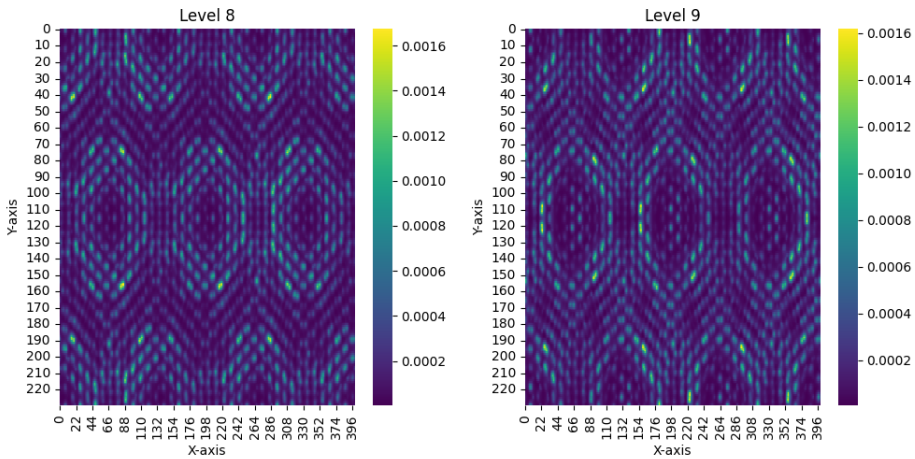
In [21]:
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
import pandas as pd
import scipy
import os
```

In [22]:
```
# Prob 2(a)
input_dir = r"C:\Users\Eric\Desktop\PHYS 129AL\PS3\Code\Local_density_of_states_near_ban
output_dir = r"C:\Users\Eric\Desktop\PHYS 129AL\PS3\Code\Local_density_of_states_near_ba
files = sorted([f for f in os.listdir(input_dir) if f.startswith("local_density_of_state
num_files = len(files)
cols = 3
rows = (num_files // cols) + (num_files % cols > 0)
fig, axes = plt.subplots(rows, cols, figsize=(15, 5 * rows))
axes = axes.flatten()

for i, file in enumerate(files):
    file_path = os.path.join(input_dir, file)
    data = np.loadtxt(file_path, delimiter=",")
    sns.heatmap(data, cmap="viridis", cbar=True, ax=axes[i])
    file_index = file.split("_")[-1].split(".")[0]
    axes[i].set_title(f"Level {file_index}")
    axes[i].set_xlabel("X-axis")
    axes[i].set_ylabel("Y-axis")
    output_path = os.path.join(output_dir, f"heatmap_level_{file_index}.png")
    plt.figure(figsize=(6, 5))
    sns.heatmap(data, cmap="viridis", cbar=True)
    plt.title(f"Local Density of States (Level {file_index})")
    plt.xlabel("X-axis")
    plt.ylabel("Y-axis")
    plt.close()

for j in range(i + 1, len(axes)):
    axes[j].axis("off")
plt.tight_layout()
plt.show()
```



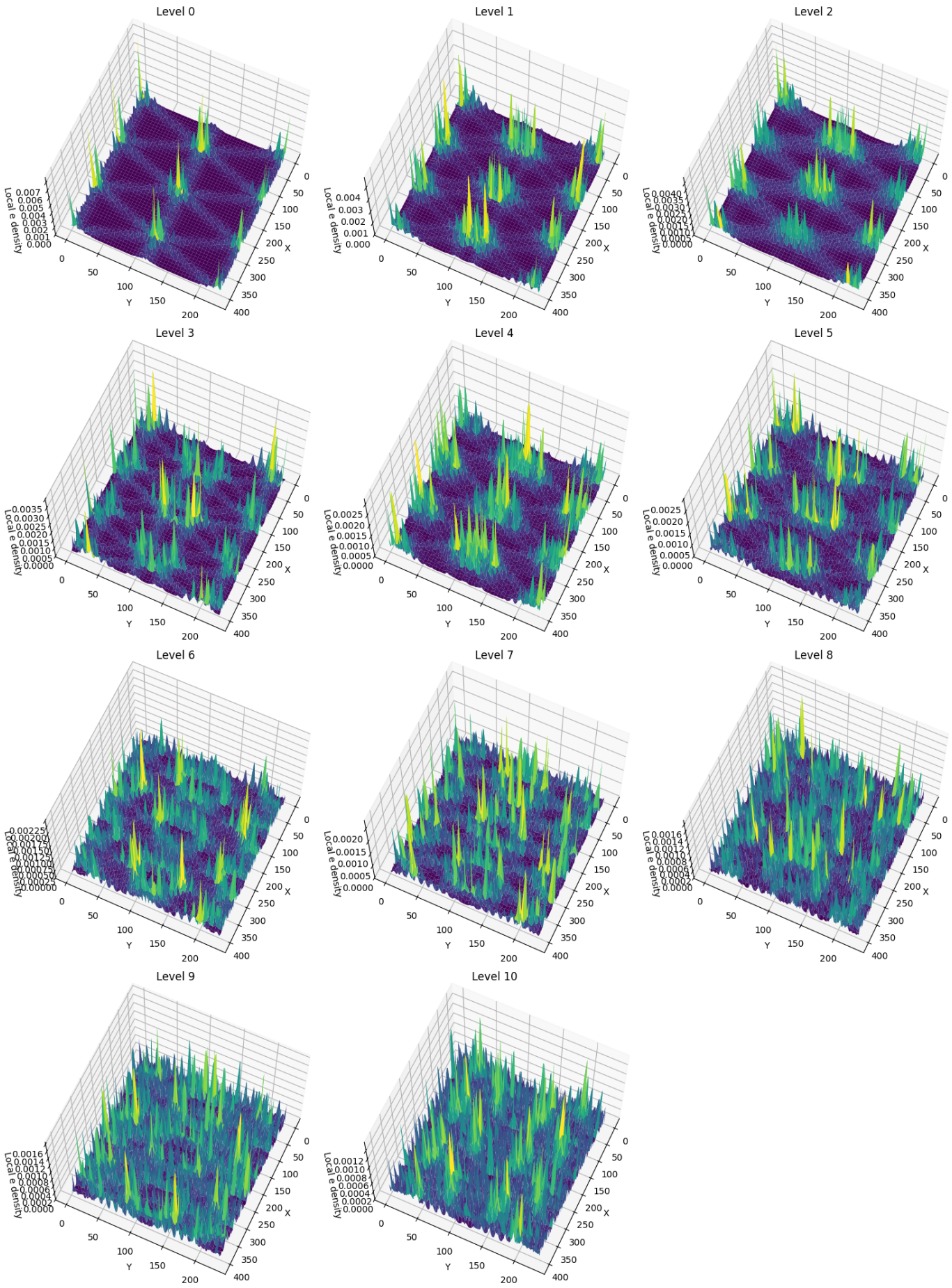






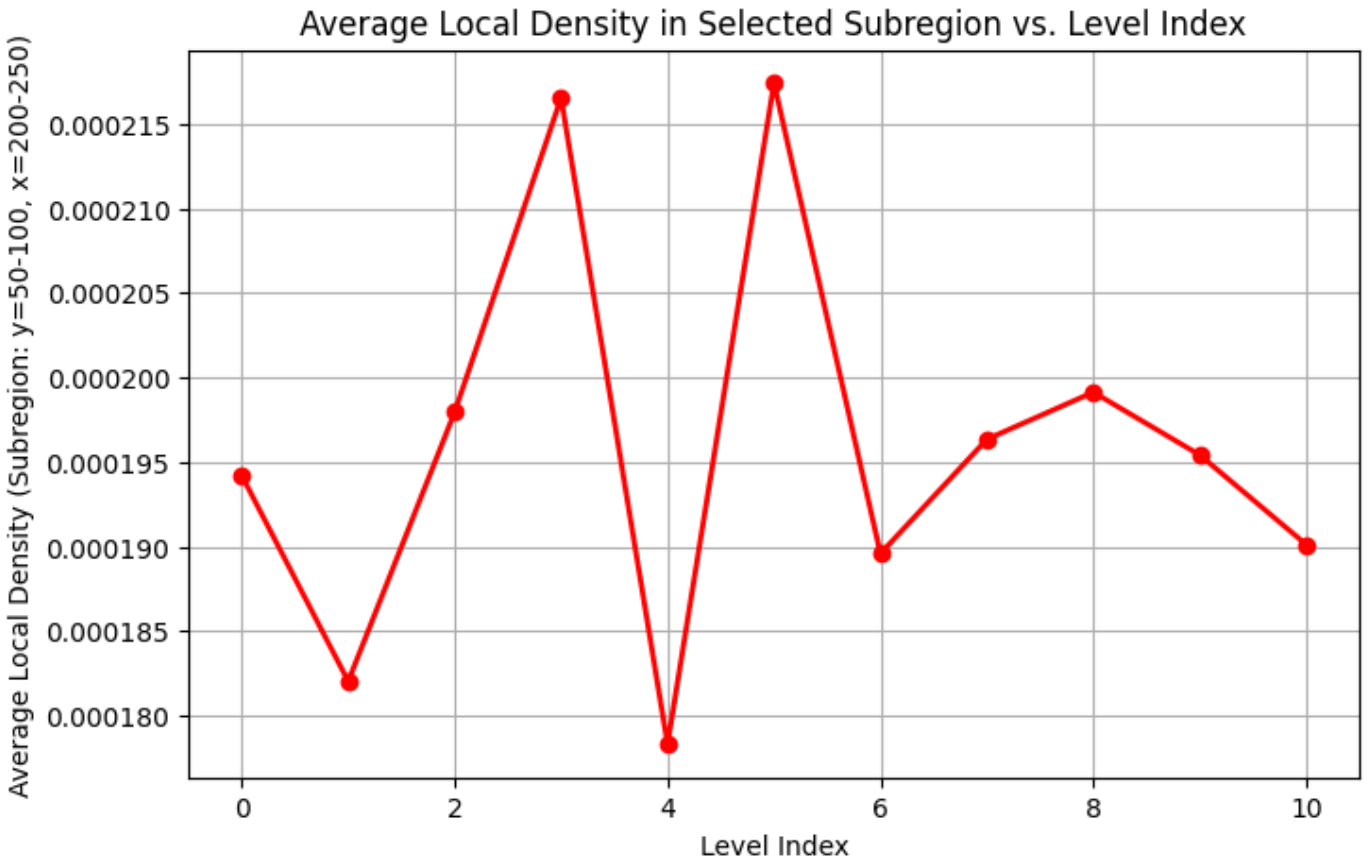
In [ ]:
```
# Prob 2(b)
input_dir = r"C:\Users\Eric\Desktop\PHYS 129AL\PS3\Code\Local_density_of_states_near_ban
output_dir = r"C:\Users\Eric\Desktop\PHYS 129AL\PS3\Code\Local_density_of_states_near_ba

os.makedirs(output_dir, exist_ok=True)
```

```
num_files = 11
cols = 3
rows = (num_files // cols) + (num_files % cols > 0)  # Compute number of rows needed

fig, axes = plt.subplots(rows, cols, figsize=(15, 5 * rows), subplot_kw={'projection': '

axes = axes.flatten()

for i in range(num_files):
    file_path = os.path.join(input_dir, f"local_density_of_states_for_level_{i}.txt")
    save_path = os.path.join(output_dir, f"level_{i}.png")

    data = np.loadtxt(file_path, delimiter=",")
    rows, cols = data.shape
    X, Y = np.meshgrid(np.arange(cols), np.arange(rows))

    ax = axes[i]
    ax.plot_surface(X, Y, data, cmap="viridis", edgecolor='none')
    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.set_zlabel("Local e density")
    ax.set_title(f"Level {i}")
    ax.view_init(elev=65, azim=25)
    plt.figure(figsize=(10, 7))
    ax_individual = plt.axes(projection='3d')
    ax_individual.plot_surface(X, Y, data, cmap="viridis", edgecolor='none')
    ax_individual.set_xlabel("X")
    ax_individual.set_ylabel("Y")
    ax_individual.set_zlabel("Local e density")
    ax_individual.set_title(f"3D Surface Plot: Local Density of States (Level {i})")
    plt.savefig(save_path)
    plt.close()

for j in range(i + 1, len(axes)):
    axes[j].axis("off")

plt.tight_layout()
plt.show()
```



In [32]:
```
# Prob 2(c)

input_dir = r"C:\Users\Eric\Desktop\PHYS 129AL\PS3\Code\Local_density_of_states_near_ban
y_min, y_max = 75, 175
x_min, x_max = 75, 275
```

```
levels = list(range(11))
avg_intensities_subregion = []

for i in levels:
    file_path = os.path.join(input_dir, f"local_density_of_states_for_level_{i}.txt")
    data = np.loadtxt(file_path, delimiter=",")
    subregion = data[y_min:y_max, x_min:x_max]
    avg_intensity = np.mean(subregion)
    avg_intensities_subregion.append(avg_intensity)
plt.figure(figsize=(8, 5))
plt.plot(levels, avg_intensities_subregion, marker='o', linestyle='-', color='r', marker
plt.xlabel("Level Index")
plt.ylabel("Average Local Density (Subregion: y=50-100, x=200-250)")
plt.title("Average Local Density in Selected Subregion vs. Level Index")
plt.grid(True)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: