Assignment 2

# Platform Description, Results and Performance

The code was run on my personal laptop with the specifications provided as under:

Graphical user interface, text, application

Description automatically generated

Numba’s automatic parallelization was utilized to demonstrate parallel speed up on a system.

The following screenshot shows the results of running the provided code with 3 different configurations:

1. Starter Code with no changes (assignment\_2\_starter\_code.py)
2. Starter Code modified to include the @jit decorator with nopython attribute set to True.
3. Starter Code modified to include the @jit decorator with both nopython and parallel attributes set to True.

Text

Description automatically generated

As you can observe, in the first case, the code was run without any modifications. It took over 11 seconds for the image to be rendered. On the other hand, in both cases (2) and (3), the precompiled version is much faster (over 20 times). There is a slight overhead associated with parallelization which is reflected in the last case (about 0.4 seconds slower), However, parallelization in combination with pre-compilation significantly brings down the time for rendering the fractal image (to about 0.02 seconds). Clearly there is significant speedup when using parallelization over just the sequential version.

Additionally, here are screenshots of the diagnostics ran on the modified code (item 3) showing exactly what parts of the code were parallelized.

Text

Description automatically generated

Text

Description automatically generated