Pthreads Implementation

The Mandelbrot set was computed using the **Pthreads** parallelization technique with varying numbers of threads. The algorithm was ran in a similar manner to the **OpenMP** version (10 times per iteration value, same number of iterations and thread counts).

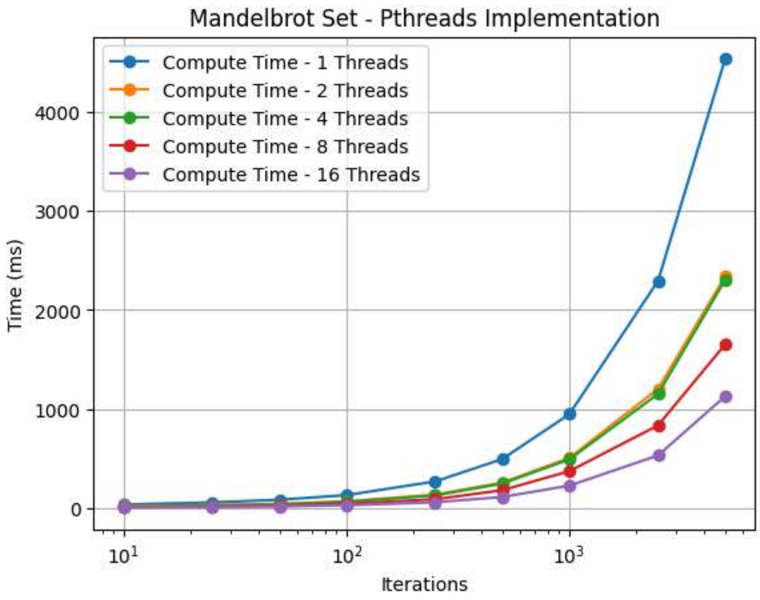
```
iteration_values = [10, 25, 50, 100, 250, 500, 1000, 2500, 5000]
```

The computation was run for a total of 10 times per iteration value, and the graph represents the **average** time taken per iteration for different thread counts.

The parallel version of the algorithm was tested with the following numbers of threads:

```
num_threads = [1, 2, 4, 8, 16]
```

Performance Metrics



Data (*5700x3D*):

Threads	Iterations	Compute Time (ms)	Save Time (ms)
1	10	34.50	23.20
1	25	55.20	20.60
1	50	81.50	21.20
1	100	134.40	22.20
1	250	268.40	21.90
1	500	491.90	22.20
1	1000	939.10	21.60
1	2500	2301.90	23.20
1	5000	4522.40	22.60
2	10	17.00	19.90
2	25	27.80	20.70
2	50	40.90	20.70
2	100	66.10	21.20

Threads	Iterations	Compute Time (ms)	Save Time (ms)
2	250	134.60	21.40
2	500	254.10	21.90
2	1000	481.10	21.30
2	2500	1168.30	22.00
2	5000	2327.10	22.00
4	10	12.30	19.00
4	25	22.10	20.40
4	50	34.50	20.80
4	100	58.40	20.60
4	250	126.80	21.20
4	500	238.40	21.70
4	1000	483.60	21.20
4	2500	1193.40	22.00
4	5000	2296.70	22.90
8	10	8.10	19.50
8	25	13.60	21.10
8	50	24.40	20.80
8	100	40.20	22.00
8	250	91.40	21.50
8	500	191.60	21.00
8	1000	335.30	22.30
8	2500	824.30	21.30
8	5000	1657.20	23.10
16	10	5.10	18.90
16	25	9.50	20.60
16	50	15.00	20.60
16	100	26.80	21.20
16	250	61.00	20.90
16	500	112.00	21.90
16	1000	220.20	21.10
16	2500	527.10	21.90
16	5000	1043.40	22.30