Mark Kaiser V00884677

CSC 360 A2 P2: L1 Approximation

For this task, I approximated the best fit L1 lines for the 2002 and 2001-2010 data sets sequentially and then in parallel with threads. Both the threaded and sequential solution gave the same results for the best L1 line, as seen in the output of the code below. The program was run on the lab computers in ECS 348, with an intel i7 processor (more hardware info needed). My threaded solution made it simple to change the number of threads. In the output below, this was done with 8 threads. Interestingly, the threaded solution ran slower than the sequential solution every time I ran it, by about 20 seconds.

The number of threads didn't really change the runtime of the parallel program. However, when tinkering with the program, it was clear that the threaded solution calculated lines and distances of points to those lines much quicker than the sequential solution (as seen with printfs that have since been removed). Thus, I speculate that this is because I included the creation of threads in the runtime of the threaded program. I thought this was necessary because the sequential solution does not require these extra expensive steps, so it would be more indicative of the actual runtime of the threaded program to include them. I did not have the time to calculate the time without the thread creation and joining.

```
./dataPar stremflow_time_series.csv

slope 2002 par: -0.816406 seq: -0.816406
slope complete par: -0.002716 seq: -0.002716
Number of threads: 8
Time elapsed for
Sequential: 276980121 ms
Parallel: 299023647 ms
```

The test is the C file itself, as I implemented the L1 approximation in its main function.