The second implementation (task parallelization) had significantly higher performance than the strictly parallel implementation for the small data sizes and a small but still present performance increase for the larger data sets. This is not readily apparent based on the execution time charts trend lines; however, if the dataset is truncated to values above 1E+06 the parallel execution trendline will have a higher power than the task trendline. The issue that I encountered during this assignment was addition errors due to out of order processing in the parallel implementation, using floating point numbers A + B may not be equal to B + A. These addition errors occurred when I used a reduction clause and allowed the program to partition the for loop using its automatic scheduling rules. To fix this issue I implemented static scheduling which forces the parallel region to execute in a specific order but also adds considerable overhead due to increased cache misses. Thus, the parallel implementation was slower due to non-optimized but necessary scheduling.