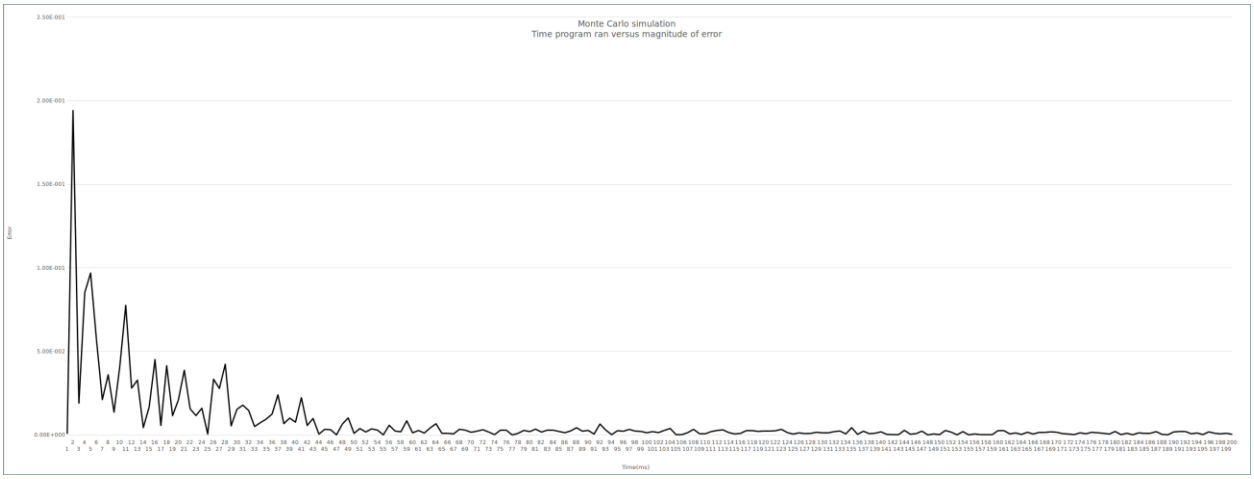
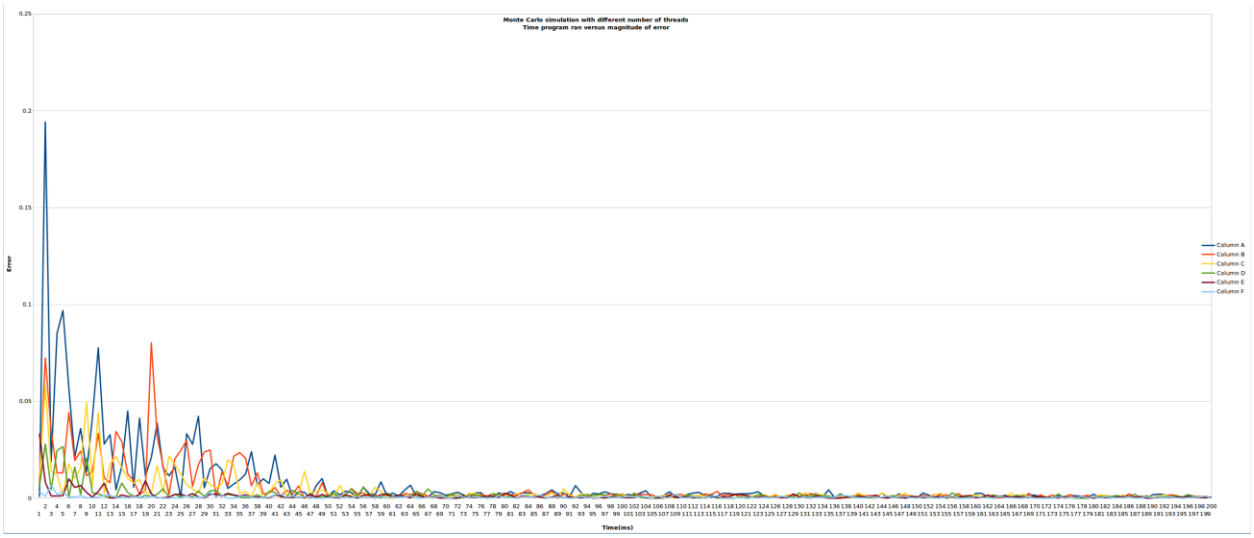
Melvin Cheng
100526486
CSCI2020u
Ken Pu

According to my results, the most optimal thread was 32. The light blue line in the second graph represents 32 threads. It starts with a small magnitude of error and consistently have a small magnitude of error. When the program ran with less than 32 threads, the first 50 results had large spikes or started with a large magnitude of error.




There is no synchronization between threads as the threads do not access a shared object. This is because we are require to keep track of the inCirc independently and find the sum of the inCirc after all the threads are complete. However, if we were to implement synchronization, I would have made a synchronized ThreadedTimeBoundedPi class object to store all the inCircs while the threads are running.