Joe DeFlorio

Assignment 1

For my assignment I implemented integration using the Monte Carlo method. To achieve this, I generated a random distribution of samples between the limits of integration to be computed with given function (sin(x)/x). With this method of integration, I achieved results that scale as I would expect. To show this scalability I used 1000000000 samples distributed across 1-8 cores/threads. As predicted, the time for execution decreased exponentially as the number of threads increased. This is supported by the linear line created by the efficiency graph.

Although these results were as expected, there are some things I might have changed if I was looking for better overall performance. First, Monte Carlo integration relies on a random distribution of numbers. In my implementation, I imported the random library and used the std::default_radnom_engine to create my distribution. I decided to use this engine and library for simplicity, however, using an alternate method of RNG might have yielded faster times or more precise results. Additionally, I generated a random sample individually in each thread. This also could have contributed to slower performance; I might have seen better times if I globally created the distribution for each thread to use.