Matthew D’Angeli
ECEC-413
20 April 2019

pThreads Assignment

Trapezoidal Integration

The parallel code submitted to compute the trapezoidal area under a curve was structured similarly to the example code provided in the vector dot product V2 source. A thread data structure was defined containing all necessary data and pointers required to effectively compute in parallel, including the chunk size (how many trapezoids are computed per core), lower bound of the integral, width and number of trapezoids being used, number of threads, assigned thread ID number, and pointers to the integral and the mutex controlling the integral. This data structure was used as the input to the worker function, trap_integrate, which is the function called by the individual threads. An array of thread data structures is created containing the same number of elements as there are threads, so each thread has its own data. Values in the thread data structure are generated depending upon the thread ID in order to effectively “chunk up” the data for ease of computing. Once the thread data has been computed, the threads are called and are given the relevant data, and each thread computes its chunk of the integral estimation. The shared integral value was initialized to the edge case (f(a) + f(b)) / 2 in order to make the common case parallel. Each thread simply computes a partial integral by looping through its chunk in the exact same manner as in the serial case, except in the parallel code, the for loops are bounded based on the thread ID; the last thread is responsible for all extra trapezoids that weren’t able to be evenly distributed to other threads. Once the partial integral is computed, the thread requests a lock on the mutex to add its part to the shared integral, then releases the mutex and is available to join back with the main threads when called to do so. Once all threads are finished and joined, the integral is then multiplied by the trapezoidal width in order to determine an area under the curve.

Table 1 – Timing analysis for lower bound of 32, upper bound of 96, and 100 million trapezoids for a varying number of cores

Threads Used	2	4	8	16
Serial Time [s]	2.594087	2.597734	2.597690	2.595043
MT-Time [s]	1.372502	0.734493	0.416622	0.215862
Speedup [s]	1.221585	1.863241	2.181068	2.379181