Scott Williams
11356176

Explain the Results

To make it worth while to use multiple threads for matrix operations the amount of threads used has to correspond to the size of the matrices. This is due to the overhead of creating threads and, when complete, joining the threads created. The graph below shows how there is no noticeable advantage to using multiple threads on a matrix of less than 128 * 128 dimension. In fact it is slower to do so. It also shows that, even on a matrix with a large size such as 2048*2048, you can use too many threads so that the performance actually decreases. This is shown again by the graph, with a program utilising 8 threads, which greatly improves on a sequential program, performing very similar to 16 threads and outperforming one using 32 threads on a 2048*2048 matrix. Again, this is due to the overhead of creating threads, with the task performed by each thread to small to gain a benefit when a large number of threads are used.

The fact that I am also multiplying the matrices and finding the norm separately is also slows down the performance of the multi-threaded programs as the overhead is effectively doubled given that each thread has to be created twice. If the program computed the norm of each slice of the matrix at the same time of multiplication I believe that the multi-threaded programs would perform better, as I could then use 32 threads with the same overhead as there currently is with 16.