Multithreading

Mentee: Bradley Morrell
Mentor: Amanda Bienz

When running very large processes on a computer, it eventually becomes horribly inefficient to run each command one after another. Why should each part of the task be computed in order if the computer could split the task up into multiple parts and complete the parts of the task at the same time? This is where multithreading comes in, which allows computers to multitask in order to complete operations.

**Objective:** To understand more about multithreading implementations and test the most efficient combination of the number of OpenMP threads and MPI processes on a machine for a certain operation. Our specific tests use a machine with 2 cores and the operation we are testing is matrix vector multiplication with very large matrices.

We can utilize both of these implementations in order to maximize efficiency.

**Problem:** There is not a known most efficient way to split up simple linear operations using multithreading. This is because there is a startup cost associated with creating more threads or processes on a computer.

**Importance:** This objective is important because if the most efficient combination of OpenMP threads and MPI processes is found, then very large simple linear operations become much easier to compute very fast.

**Skills and Knowledge Gained:**
- Learned more about how processes are carried out in computers.
- Learned more about multithreading implementations in computers.
- Learned more about matrices and the process of multiplying through them.
- Learned more about how to deal with bugs and complications in code.
- Learned more about how matrices are stored in computers.
- How to use OpenMP in C++ code and how to use MPI.
- How to properly display the data in a graph.

**Progress:**
- Completed the matrix vector multiplication tests.

**Future Work:**
- Learn more about improving other operations besides matrix vector multiplication.