Lab 9 Report
Ian Taulli
Computational Physics

Part 2

I modified the work I did for Lab 7 so that I could use the pre-existing memory allocation and buffer rotation schemes of your CPInitialValuesSolver class. I wrote the new MultiLeap class to implement the multi-threaded solver. The functor getNext allows the thread to calculate the next values of the array in their sections. The sections are created by manipulating the allowed values of the iterator j in the getNext function, the thread's number from the iterator i in propagate's for-loop is passed to getNext for this purpose. The desired number of threads is passed as an argument when constructing the MultiLeap class and then used to determine the number of points to assign to each thread. Each time step is insulated from the next because they are calculated one at a time when solver.step is called in the python file. And each thread only cares about the current and last time steps which are determined before the thread is executed, so the threads sectioned off in this way can run independently of one another. Plus, not much modification is needed to the python files to be able to execute and animate the waveform, and setting the initial values is already handled.

I had trouble merging the makefiles from lab 7 and your multi-thread examples. The compiler doesn't like something about the way I'm using emplace_back from what I can tell.