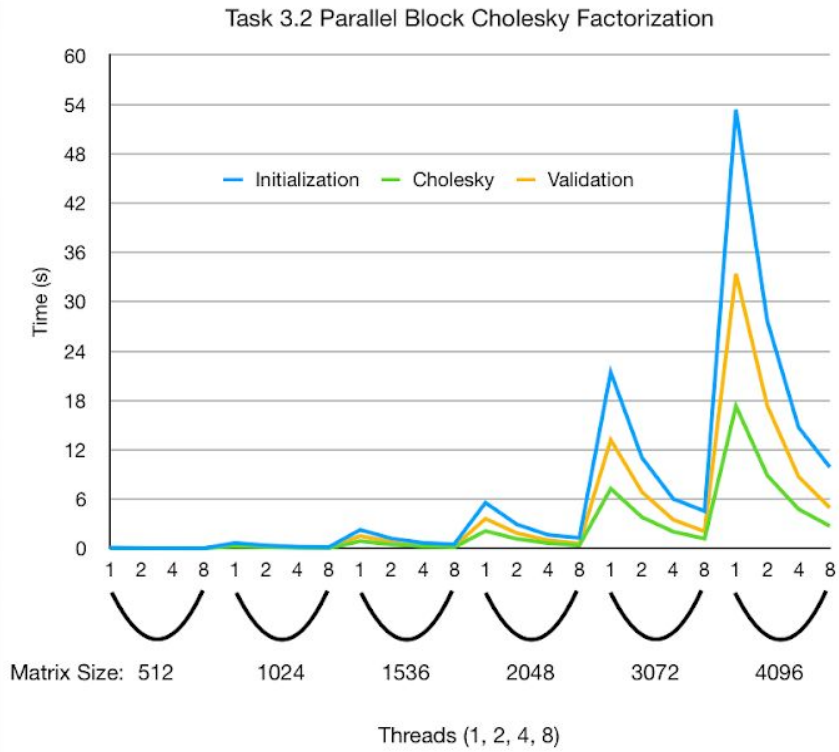
Task 3 Writeup



Speedup And Efficiency When Serial and Parallel Block Sizes Match For Best Speed

Matrix Size	Blocksize	Threads	Serial Time	Parallel Time	Speedup	Efficiency
2048	128	2	2.131786	1.164619	1.83	0.915
3072	128	2	3.800243	2.124533	1.79	0.895
4096	128	2	17.32218	8.887285	1.95	0.975
4096	128	4	17.32218	4.785665	3.62	0.905

openMP Parallelization Strategies:

For parallelizing blockCheolesky, we tried a number of approaches. When beginning to parallelize, we did not run into any bumps. We were able to parallelize the initialization with no problem. This was because we implemented perfectly nested loops in our initialization.However it was not so obvious how to implement parallelization for our blockCholeksy factorization. This was because in factorization we did not have perfectly nested loops. OpenMP does not support parallelization for not perfectly nested loops. To get around this and parallelize our factorization, we had to parallelize each loop on its own. To calculate L11, there wasn't any problem parallelizing that case. For L21 and L22, we tried different approaches. First we tried to parallelize with shared variables. This did not work because the iteration variables will have

different values, causing threads to execute on the same blocks. After this, we had to learn how the iteration indexes were being used by each thread. After learning this we were able to make the correct variables private to each thread. This gave the threads the correct variables that other threads could not change. We also needed to test and how much we should give each thread. After testing and determining amount of work each block should get, we were able to get great speedup. After completing L21, We used the same approach for L21.