# Lab2 report

How many floating operations are being performed in each of your vector add kernels? EXPLAIN.

One Integer operation to find the index of the vector, One floating operation to add the values at a given index in the vectors

How many global memory reads are being performed by each of your kernels? EXPLAIN.

Two per kernel, one read per input array.

How many global memory writes are being performed by each of your kernels? EXPLAIN.

One per kernel, writing one value to the output array.

Describe what possible optimizations can be implemented to your kernel in vectorAdd.cu to achieve a performance speedup.

Speedup could be achieved by making sure no more than half of a warp (16 threads) is allocated to a block so that all the threads can execute at the same time on separate multiprocessors without multiple threads trying to read from the same data block.

In what ways did Thrust make developing a functional vector addition code easier or harder?

I didn’t have to manually allocate and free global memory, this was handled by the device\_vector allocation. I didn’t have to write configure the grid and block dimensions. I didn’t have to write a vector addition kernel.

Name three applications of vector addition.

Mechanical simulation

Particle Physics Analysis

Game engine physics simulations