John S. Burke
EC 527
Lab 7

Part 1: Ran the tutorial as requested.

Part 2: Simplified SOR

Code in zip file

Tested two cases, one where GPU had syncthreads, and another where it was removed. Both cases met the tolerance imposed by comparison to GPU execution.

With syncthreads the GPU took `437022 ms` and the CPU took `107750 ms.` This first of all shows the cost of syncthreads, and second of all shows that the parallelism the GPU can leveraged wasn't fully used and that the cost of data transfer and other overheads made the CPU a faster execution.

Without syncthreads the GPU took `234675 ms` and the CPU took `107750 ms.` This exemplifies the overhead of syncthreads in general in comparison to the last and shows again the overhead of using the GPU in general.

Part 3: Multiple Blocks

Code in zip file

Since we were given the constraint of blocks with 16x16 threads, but each thread was now working on a single output element, a grid of various blocks had to be used. Since the array is 2048x2048 floats, 128x128 blocks were needed for the grid.

Furthermore, we had the host controlling the 2000 iterations rather than the kernel. This is better because when depedencies in data are presented in problems like SOR, we want to syncthreads. However, as seen in the last example, this has extreme overhead. Also, we cannot synchronize threads across blocks since they are independent, so having the kernel do a single iteration is the natural result.

Again, the GPU code met the tolerance requirements in comparison to the CPU. This time the GPU took `19699 ms` and the CPU took `107479 ms`, much like in part 2.