Homework #5

ME 471/571

1. Run the Jupyter notebook `sm_config.ipynb` on the course website, and experiment with different kernel configurations. Illustrate how you can ”expose parallelism” with different configuration parameters (number of blocks, threads per block and shared memory).

   You can also refer to the slides presented in class.

2. (**Using shared memory**). Modify `heat2d_cuda.cu` to fully take advantage of shared memory. Compare the timing results you get for the GPU, a non-shared memory approach (using only global memory) and an approach that uses shared memory.

3. (**Fractals.**) For this problem, you will generate a Julia fractal (or any other fractal you find interesting) using CUDA to accelerate the actual computational kernel.

   (a) Start by writing a serial CPU code that computes the fractal. Write out the results using your serial binary code and plot using a Jupyter notebook. You can start with the note book provided on the course repo.

   (b) Convert the piece that generates the fractal to a CUDA kernel.

   (c) Compare timing results between the CPU and GPU codes. Use `nvprof` to profile the code :

   ```
   $nvprof julia_gpu
   ```

   This will report the timing for the kernel alone. Can you improve this time? Adjust the block dimensions to optimize the use of threads/warps in the kernel.