Student Instructions

Read and review the heat1.cu code.

1. In the main routine, what is the effect of changing the variable "method" in the code? What are the two values to which it can be set?
2. Note the use of allocated memory for both the device and host, as well as copy statements before and after the main device iterative loop. Why are two copies of the arrays required when parallelizing code using CUDA?
3. CUDA devices have limitations on both the number of threads per block, the total number of blocks, the total number of threads requested, and other resources. How does the behaviour of your code change if you exceed these resources? What warnings are provided? [Hint, set threads per block to 20000, well above typical allowed threads per block. How does this break the code? Does the broken code run faster or slower than the none broken code? Is it obvious that it is broken when you first run it?]
4. Using the heat2_starter.cu, provide a kernel that will mimic the behaviour of the CPU based kernel provided. You can use heat1.cu as a model. Please use the provided memory structure of a flattened 1-D array for your temperature grid, where the rapidly changing index is column and [i*ncol+j] corresponds to the ith row and jth column. Use a 2-D breakdown of your blocks and threads per block, so that changing in the x direction in thread-block space corresponds to changing the column, and changing in the y direction in thread-block space corresponds to changing the row.
5. Run your 2D code for different choices of grid dimensions and block dimensions, as well as different problem sizes, and discuss the speedup that results.