ECE 4122/6122 **Advanced Programming Techniques**

Project #2 2D and 3D Heat Diffusion

Assigned: Thursday, November 8, 2018 *Due: Thursday, November 30, 2018*

In this exercise you will be simulating the diffusion of heat in two or three dimensions using CUDA. As in SE2, we will be using simplified heat diffusion equations. For 2 dimensions assume that there is a rectangular room that is divided into a grid. Inside the grid will be "heaters" with various fixed temperatures.

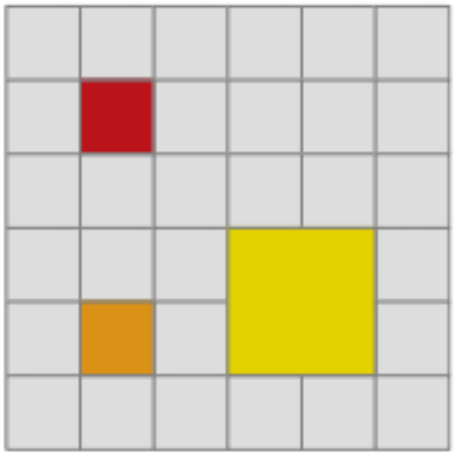

FIGURE 1 – 2 dimensional room with fixed "heaters"

Given the rectangular room and configuration of heaters, you need to simulate what happens to the temperature in every grid cell as time progresses. Cells with heaters in them always remain constant for simplicity. At each timestemp we assume heat flows between a cell and its neighbors. If a cell's neighbor is warmer that it is, it will tend to heat up. Conversely, if a cell's neighbor is colder than it is, it will tend to cool down. We will use the following simplified equation to model this behavior

$$T_{new} = T_{old} + \sum_{neighbors} k(T_{neighbor} - T_{old}) \tag{0.1}$$

In the previous equation, k represents the rate at which heat will flow through the system. A large value of k will drive the system to a constant temperature quickly. For the 2 dimensional case, we will consider only four neighbors (top, left, bottom, right) so we can simplify the equation to

$$T_{new} = T_{old} + k * (T_{top} + T_{bottom} + T_{left} + T_{right} - 4 * T_{old}) \tag{0.2}$$

Similarly the equation we will be using for 3D will be

$$T_{new} = T_{old} + k * (T_{front} + T_{back} + T_{top} + T_{bottom} + T_{left} + T_{right} - 6 * T_{old}) \tag{0.3}$$

k will remain constant during execution but will be a parameter provided to your program. For boundary conditions, use the current value of T_{old}. For example for the grid point x = 0, y = 1, there is not a T_{top}, since it is off the grid, therefore use T_{old} for the current cell as T_{top}.

WHAT YOU NEED TO DO You need to implement a CUDA program that will output the temperature at each grid point after running the equation for a specified number of timesteps. Example execution

./heat2D3D sample.conf

sample.conf is a configuration file which will specify the parameters for your simulation. Below are example configuration files for 2 and 3 dimensions.

```
##### 2D CONFIGURATION EXAMPLE #####
#your code should ignore lines starting with '#'
#you can assume arguments will always follow this ordering however whitespace may vary

#2D or 3D
2D

#the value for k
4

#number of timestep to run
200

#width (x-axis) and height (y-axis) of grid
800,800

#default starting temperature for nodes
5

#list of fixed temperature blocks (squares for 2D)
#can be 0, 1 or more
#assume blocks won't overlap
#location_x, location_y, width, height, fixed temperature
5, 5, 20, 20, 200

500, 500, 10, 10, 300
```

```
##### 3D CONFIGURATION EXAMPLE #####
#your code should ignore lines starting with '#'
#you can assume arguments will always follow this ordering however whitespace may vary

#2D or 3D
3D

#the value for k
4

#number of timestep to run
200

#width (x-axis) height (y-axis), depth (z-axis) of grid
800,800,800

#default starting temperature for nodes
5

#list of fixed temperature blocks (cubes for 3D)
#can be 0, 1 or more
#assume cubes won't overlap
#location_x, location_y, location_z, width, height, depth, fixed temperature
5, 5, 5, 20, 20, 20, 200

500, 500, 500, 10, 10, 10, 300
```

You may design your program however you wish as long as it is reasonably efficient. We will discuss possible implementations in class. At the end of execution, your program should write an output file named heatOutput.csv listing the temperatures at each grid point in comma separated format. Use **float**s for all values.

You MUST use CMake for configuring your project. I must be able to build your project using cmake .. from a build folder within your submission. You will need to use online resources to determine how to correctly refer to needed header files. Do NOT write your own Makefile directly. For submission, compress your solution into p2.zip (NOT .TAR.GZ, or .RAR or anything else). zip is installed on the login node. Submit via canvas.

Example Output 2D grid with width = 3, height = 2
334.2, 342.2, 300.2
342.1, 343.1, 32.0

Note the newlines and commas for each row.

Example Output 3D grid with width = 3, height = 2, depth = 4
334.2, 342.2, 300.2
342.1, 343.1, 32.0
blank line
323.2, 302.2, 230.2
142.1, 323.1, 332.0
blank line
323.2, 302.2, 230.2
142.1, 323.1, 332.0
blank line
323.2, 302.2, 230.2
142.1, 323.1, 332.0

Insert a blank line between each x,y slice