TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #2, Module: MA5615

Gustavo Ramirez

April 5, 2017

1 Part 1: serial

As can be seen in the file ./part1/cyl_rad.c, the serial implementation is a symple straighforward composition of **for** loops, representing the propagation of energy (i.e. heat) tangentially on the surface of the cylinder. Alghough the implementation of each step in the evolution of the energy through the surface of the cylinder is, in this serial case, performed by an external (from the main scop) function, this brings up an overhead in the execution of the evolution of the system; for this reason, in the serial part on the file for the parallel implementation, all the code in that external function is moved within the main code.

2 Part 2: parallel

As can be seen at ./part2/, a Makefile has been created for the automated build (and delete) of the executable and object files in the project.

The core of the parallel implementation lies within the kernel function __global__ void system_evolution_gpu(...). For the specific problem at hand, a combination of global and shared memory was used. Due to the restriction in size for the shared memory, a series of **if** statements were used to select between using either shared or global memory. The criteria used was based on a roughly size of 16 kb for the shared memory, restricting to a max size of roughly 2048 on the component m of the matrices.

The previous design criteria is based on the general scheme used on the creation of the grid as a whole: the number of blocks used were one for each row in the matrix M, and for each block (i.e. each row), shared memory was used in case $m < 2048$; if m goes beyond that value by say one order of magnitude, then shared memory can't be properly dynamically allocated.

After running executions with 1000 iterations, one important fact is that the error propagates out, and after the 1000 iterations the parallel and serial results are virtually the same.

2.1 Execution examples

In general, to display the thermostat, timings and a simple execution:

```
./cyl_rad -n N -m M -t -a -p NR_ITERATIONS
```

and specifically, for the reference case given at the description of this assignment:

```
./cyl_rad -n 15360 -m 15360 -t -a -p 1000 » file.txt 2>&1 &
```

becase the execution time for the serial case was of around 69 minutes.

3 Part 3: performance improvement

Different combinations of n, m and number of threads per block were used. All the resulting performances for each case of study can be seen at the file ./part3/timing.dat.
As can be seen in that file, speedups ranged from 6 to an interesting and rare case of 55; this latter case might be a mistake, but sounds plausible from the symmetry of the constructed grid and the matrix M of energies under consideration. In particular, for the case given as reference in the description of the assignment, a speedup of around 14 was obtained.
On the precision side, no errors greater than 1.E-5 were encountered.

4 Part 4: double precision testing

A few more tests were executed for the case of double precision; the code for this case can be found at ./part4/. The resulting extra runs for this double case can be found in the file ./timing_double.dat.
As can be seen from that file, the speedups were better in the case of float precision, than in the case of double precision.