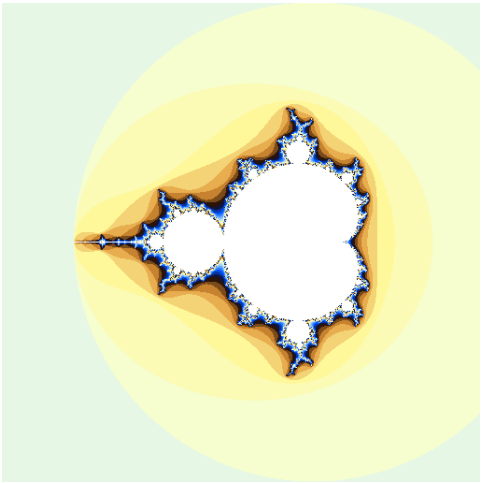
First Assignment
GPGPU & Accelerator Programming
COMP10065

Issue Date: Monday, January 30th, 2017
Due Date: **5pm, Friday, February 24th, 2017**

The Mandelbrot Set



"The" Mandelbrot set is the set obtained from the quadratic recurrence equation[1]:

$$z_{n+1} = z_n^2 + C$$

with $z_0 = C$, where points C in the complex plane for which the orbit of z_n does not tend to infinity are in the set. You are provided with serial C++ source code which generates a single image by sampling the complex numbers and determining, for each sample point c, whether the result of iterating the above

[1] Weisstein, Eric W. "Mandelbrot Set." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/MandelbrotSet.html`

recurrence goes to infinity. Your task is to use the NVIDIA CUDA C/C++ API to produce the same behaviour as this code, ideally with an improvement in performance.

Resources

You are given a serial program (mandelbrot.cpp) which visualises a section of the mandelbrot set and writes it to the hard disc as a PPM image. The program accepts two optional parameters which specify the x and y extent of the generated image. There is also a CMakeLists.txt file; with comments that may help configure your CUDA C/C++ program.

Submission

Your submission should include source code, along with a short report (circa 2000 words) to introduce the problem, before explaining the steps you have taken to realise your CUDA C/C++ version. Your report should include performance timings, and indicate whether any tools have helped you. Please make use of graphs, figures, and code excerpts. Work individually. Though you can share ideas, please do not share code.

Marking Scheme

40 marks are available for this assignment, corresponding to 40% of the marks awarded for the entire COMP09041 module. The following provides a breakdown of the marking scheme for this assignment:

Relative performance of the CUDA C/C++ program	10 marks
Quality and readability of the CUDA code	10 marks
Presentation of the written report	5 marks
Explanation of decisions and methodology	15 marks

Methodology

The following suggestions may help you to get started:

- Maintain the untouched serial version.
- Time the cost of each section of the serial code.
- Consider which sections should run on GPU, and which on the CPU.
- Which sections exhibit arithmetic intensity?
- How will you measure the similarity of your image to the original?
- Reproduce the serial results using CUDA.
- Consider varying parameters, such as:

  - program data: i.e. generated image size.
  - CUDA arguments: e.g. grid size; block size; shared memory size.
  - compiler switches: e.g. optimisation; GPU compute capability.
  - graph the timing results for subsets of such variations.
- Might the CUDA Occupancy Calculator be useful?