

PHYS 511: Computational Modeling and Simulation - Fall 2017
Assignment #5, due Monday November 27, by 5:00 pm

Plotting the Mandelbrot set; basics of parallel programming with MPI

The Mandelbrot set is the simplest and most commonly used example of a fractal structure. It is defined as a set of complex numbers $\{c\}$ for which the sequence

$$z_{n+1} = z_n^2 + c \quad (1)$$

does not diverge when iterated starting from $z_0 = c$. That is, the sequence of complex numbers z_0, z_1, z_2, \dots remains finite by magnitude when $n \rightarrow \infty$.

As a fractal structure the Mandelbrot set shows self-similarity, i.e. zooming in to a fractal yields pictures that are similar (whether exactly or somehow approximately) across widely varying scales.

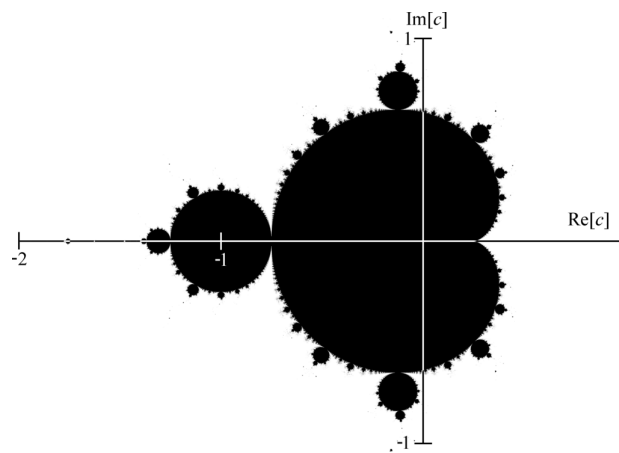


Image credit: Wikipedia

In this assignment we will plot the Mandelbrot set in the complex plane $-2 < x < 1$, $-1.5 < y < 1.5$ (where $x = \text{Re}[c]$ and $y = \text{Im}[c]$) by scanning this domain along x - and y -direction with some step $\Delta x = \Delta y = h$.

1. Write a Fortran program (`as5.f90`) that scans points in the above domain with a step h . Compute after how many iterations the sequence diverges (e.g. when $|z|$ becomes larger than 2). Set the maximum number of iterations $n_{\max} = 255$ for each point. If the maximum number of iteration is reached, we will assume that $n = 255$ and the point survived the test, i.e. it belongs to the Mandelbrot set.
2. Facilitate parallelism in your program by means of Message Passing Interface (MPI). You should divide work as uniformly as possible between N_p processors (the number of processors is not predefined, it is specified by the user at run time). Take into account that the computational effort (i.e. the number of iterations) needed for testing a single point may vary quite significantly depending on its position x, y . Think about statistical averages in the context of the above figure.
3. Output your map into a file (`map.dat`) in the following format:

```
 $x_1$   $y_1$   $n_1$   
 $x_2$   $y_2$   $n_2$   
:  
 $x_m$   $y_m$   $n_m$ 
```

Keep in mind that the output into the file should be performed by a single MPI process only (master). Otherwise conflicts, run-time errors, or data overwriting and loss will be unavoidable when different MPI processes compete for writing into the file. Therefore all MPI processes should communicate and send (or reduce) the chunks of results they generated to the master process.

4. When you are sure that your program works when executed in parallel, make test runs for the case $h = 0.001$ (i.e. 3001 points along x and y directions) involving a different number of processors/cores. For example, to run four MPI processes you would need to type the following command

```
mpirun -np 4 ./as5
```

Make sure that your executable (`as5`) is compiled with the `-O3` optimization flag. Measure the execution time for different number of MPI processes. Compare and explain differences and interpret any performance gain or lack thereof. Present your analysis in the report file (`report.txt`). One thing to keep in mind here is that the data file writing operation is not parallel, so if it is slow it may be a bottleneck. Therefore you may want to measure only the execution time of the part of the program that actually does calculations (not writing into the file). Alternatively, you can switch off data file writing when doing time measurements. Also, make sure your computer is not running any heavy tasks in the background as it may influence your time measurements in a unpredictable way.

5. Use any plotting software of your choice (e.g. GNUPLOT) to read the data file and to make a *nice looking* plot of the Mandelbrot set. Store it in high resolution (at least 1024×1024 pixels, but can be even higher as your data is very dense) as a png file (`map.png`). Be creative in choosing the color palette. The color should represent the number of iterations (e.g. the maximum n , 255, would correspond to the black color in the above figure).
6. Do you notice any self-similarity of the features that appear on the plot?
7. Submit files `as5.f90`, `map.png`, and `report.txt` by uploading them to your Google Drive directory named `as5` shared with the instructor. Do not submit data file `map.dat` as it may be too large.