SCIENTIFIC COMPUTING USING PYTHON

HIGH PERFORMANCE COMPUTING

PROJECT

Torben Larsen, Thomas Arildsen, Tobias L. Jensen

June 17, 2022

1 | The project

This roject relates to the Mandelbrot set (after the mathematician Benoit Mandelbrot), which leads to compelling two-dimensional fractal patterns. A fractal contains elements of self similarity when plotted. The Mandelbrot set is a quadratic complex mapping of the type:

$$z_{i+1} = z_i^2 + c, \quad i = 0, 1, \ldots, I-1 \tag{1.1}$$

where $c \in \mathbb{C}$ is a point in the complex plane, and $z_i \in \mathbb{C}$ for $i = 0, 1, \ldots, I$. This provides us with $z_0, z_1, \ldots, z_I$ where the initial condition is $z_0 = 0 + j \cdot 0$ and $z_1, \ldots, z_I$ are referred to as iteratively achieved outputs. In Equation (1.1) the iteration number is $i + 1 \in \{1, 2, \ldots, I\}$ and the total number of iterations is I. For each observed complex point c we then compute I iterations of Equation (1.1). For each complex point c we then determine:

$$\imath(c) = \min \mathbb{T}, \quad \mathbb{T} = \big\{ i \,\big|\, |z_i| > T, i = 1, 2, \ldots, I \big\} \cup \{I\} \tag{1.2}$$

where T is a threshold value and the initial condition is always chosen such that $|z_0| \leq T$. From Equation (1.2) we see that $1 \leq \imath(c) \leq I$. So, if $|z_{i+1}| \leq T$, $\forall i = 0, \ldots, I-1$ we obtain $\mathbb{T} = \emptyset \cup \{I\} = \{I\}$ meaning that $\imath(c) = I$. Obviously, in a computational implementation we need not compute all I iterations if an $i + 1 < I$ leads to $|z_{i+1}| > T$. We then just set $\imath(c)$ to the smallest $i + 1$ that leads to $|z_{i+1}| > T$. For plotting purposes we then form a simple linear mapping as:

$$\mathcal{M}(c) = \frac{\imath(c)}{I}, \quad 0 < \mathcal{M}(c) \leq 1 \tag{1.3}$$

The smaller $\mathcal{M}(c)$ value we have, the faster that specific complex point c makes $|z_{i+1}|$ in Equation (1.1) increase. In a case with the number of iterations $I = 100$ a value of $0 < \mathcal{M}(c) \lesssim 0.1$ indicates an extremely 'active' (unstable) point whereas a value of $\mathcal{M}(c) = 1$ indicates a stable point. A point c is said to belong to the Mandelbrot set if $|z_{n+1}|$ remains bounded (finite) for $n \to \infty$.

A pseudo-code example of the above mathematical description for a known/given c is

```
function M(c)
    z_0 = 0
    for i = 0, ..., I-1 do
        z_{i+1} = z_i^2 + c
        if |z_{i+1}| > T then
            return (i+1)/I
        end if
    end for
    return 1
end function
```

In this project, the computational task is then to determine $\mathcal{M}(c)$ for a c-mesh, which we limit c-wise to $-2 \leq \Re\{c\} \leq 1$ and $-1.5 \leq \Im\{c\} \leq 1.5$. We then select a certain number of points for each of $\Re\{c\}$ and $\Im\{c\}$ as p_{re} and p_{im}, respectively. The complex plane limited by the mesh is described by a complex matrix $\mathbf{C}$ as:

$$\mathbf{C} = \begin{bmatrix} -2.0 & \cdots & 1.0 \\ \vdots & & \vdots \\ -2.0 & \cdots & 1.0 \end{bmatrix} + j \cdot \begin{bmatrix} 1.5 & \cdots & 1.5 \\ \vdots & & \vdots \\ -1.5 & \cdots & -1.5 \end{bmatrix} \in \mathbb{C}^{p_{\text{re}} \times p_{\text{im}}} \tag{1.4}$$

You should select p_{re} and p_{im} according to the computational resources you have available and the desired resolution. Likely number could be something like $p_{\text{re}} = 5000$ and $p_{\text{im}} = 5000$. We use a threshold of $T = 2$. An example is shown in Figure 1.1.

1.1 Tasks

This section aims to clarify some of the expectations we have for the project and the documentation you deliver.
Output:

1. **Implementation:**
   Implement as minimum three different versions of the computational code being a 1) naïve version; 2) a vectorised, numba, cython or f2py version ; and 3) a multiprocessing version accepting a user selectable number of processing units. These implementations must be validated for correctness (as much as you find necessary to convince us).

2. **Output:**
   You must deliver plots for the Mandelbrot sets – you can use a colormap such as `matplotlib.pyplot.cm.hot`. Also, show the execution time for the different implementations – and for the parallel version also show execution time and speed-up versus number of computational units. Save all the figures in PDF format. Also save the relevant simulation data including the output $\mathbf{z}$.

3. **Software design:**
   Explain briefly about your considerations for the overall design of the modules and functions you wish to use for the project. You should provide a name for all functionalities, input, output and core task. In particular your considerations in

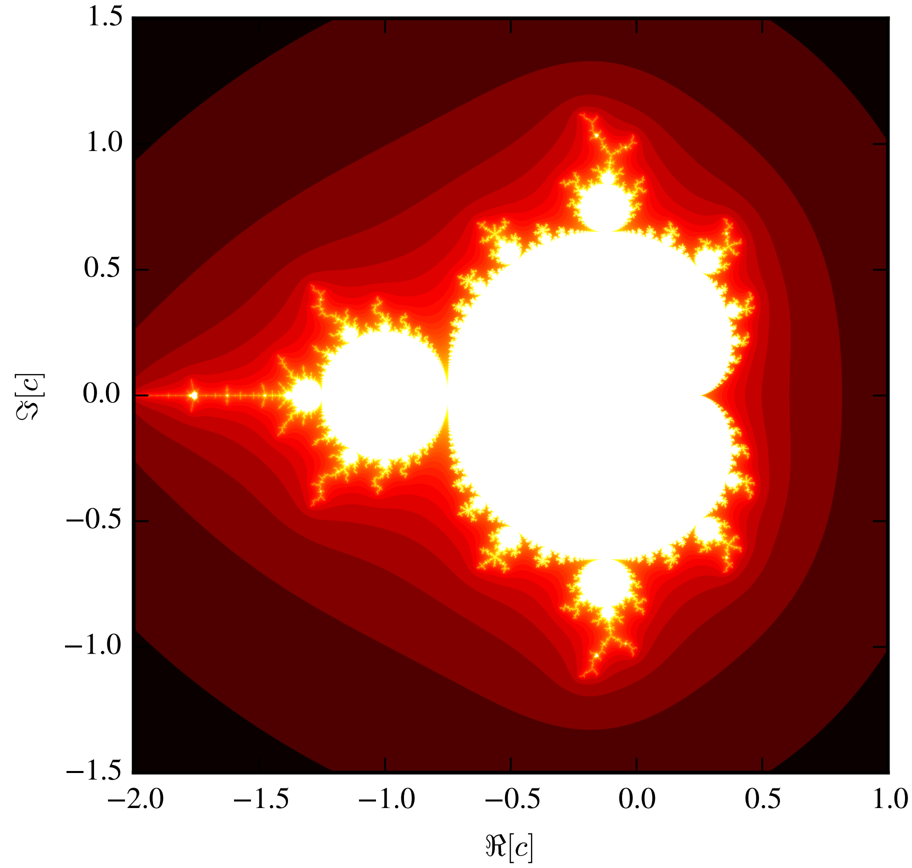

Figure 1.1:: Mandelbrot set plotting $\mathcal{M}(c)$ with $p_{\mathrm{re}} = 5000$ and $p_{\mathrm{im}} = 5000$ using the `matplotlib.pyplot.cm.hot` colour mapping in Pythons Matplotlib.

relation to optimisation should be included. We also expect that you deliver an algorithm for the computational part of the project. Considerations for data types, parameter passing between functions (avoiding global variables) etc. should be included.

4. **Test plan:**
   Briefly explain about the test plan you intend to use. You do not need to make a huge test scenario but include something to demonstrate that you have tried to implement testing.

5. **Profiling and benchmarking:**
   Show the key execution time/profiling for at least some of the implementations. Benchmarking (using e.g. `time.time`) must be provided such that we for example see speed-up versus number of computational cores. Show what you find relevant to demonstrate the quality of your optimisation.

You must deliver a zip-file including the code and documentation. Feel free to include the documentation as docstrings, LaTeX or whatever. Include a `README` file to explain what files you have included and how we should use the code.

1.2 Peer Evaluation

We use peer evaulation for this project. This means that each of you will be assigned two other projects to check through and provide feedback on. The Moodle course page will provide a system for managing the peer evaluation process of the submitted projects.
You will all be provided a list of clear criteria to check for in the projects. Peer evaluation will take place in an anonymous fashion so that you cannot see who has evaluated your project.
The course organiser will check the project evaluations and has the final say regarding whether a each project passes the described criteria. Students whose projects do not pass the criteria will be given an opportunity to re-submit the project.