Peer-review of assignment 4 for *INF3331-linuse*

Reviewer 1, steinrol, steinrol@uio.no
Reviewer 2, marikoll, marikoll@uio.no
Reviewer 3, henrisho, henrisho@uio.no

October 8, 2018

1 Review

Python version: 3.5.2, 3.6
OS: MacOs Mojave 10.14, High Sierra 10.13.6, Linux Ubuntu 16.04

Assignment 4.1

The code works as expected, in an expected time. Should maybe be able to run the function with external parameters.

generate_image would benefit from being enclosed inside the "if __name__ == "__main__" statement. This makes the functions "importable" from other scripts without being forced to invoke generate_image first.

No documentation, although clarifies that the CLI is documented.

The code is easy to read, and the functions, as well as the variable names are informative. the code is divided into logical functions.

Assignment 4.2

The code works as expected, in a timely manner. No documentation either as docstrings nor regular comments . It is divided into the same logical functions as the previous task, and it is written with good usage of the numpy libraries, and good vectorization, achieving a speedup of around 10 times, however this is not documented in the required report.

No report.

Assignment 4.3

Works as expected, pretty much the same solution as the plane-python implementation, but avoiding complex calculations such as squareroots(GOOD) this allows further speedups using the jit-compiler. Still no report on the runtime and implementation this script...

Still no documentation of this script...

Assignment 4.4

Not done

Assignment 4.5

This code mostly works, although the help output is not very explanatory, having to look at the code to understand what is going on is generally not a good idea. Especially with the regards to the file-names, where you seem to generate a report when a file-name is given but not a picture? this report though does not answer the questions listed in the assignment. And having to create the files yourself is not really part of the exercise. There is a bug where if a try to run your code without any commandline-arguments, it just gives me a bunch of errors, this generally is easy to fix with a:

```
1 if len(sys.argv)==0:
2     print(help())
```

In this part of the exercise you have created docstrings, these however lack an explanation of the parameters the functions takes in, and the return values of the function.

You have chosen to just copy everything from mandelbrot_n.py to the mandelbrot-file. You could have saved a lot of lines of code by using the code you have already made, and imported it as you needed them.

As for the picture generation i find it cool that you tried to generate your own picture! and it mostly works, it seems that as you are using the product of sin() and cos() in the generation of the blue coordinate in your RGB-array. This is problematic as it can easily become negative, thus going outside of the definition of a floating-point RGB-image [0.0-1.0]. this means that the plt.show() statement only yields a white picture for the full mandelbrot, and the plt.figsave() only gives errors and not a working image. At least on my macbook. However on linux machines it seems to work, but with warnings, warnings are generally bad for the portability of your code.

Nice check on intersection with the mandelbrot set!

Assignment 4.6

The setup script installs the module, containing the function compute-mandelbrot. As required in the task
No test file provided
Also no README provided, making it even more difficult to run the scripts.

Assignment 4.7

Not done

Assignment 4.8

Not done

General feedback

In summary:

- Document your code more, you should document all functions, in all files!, also remember that it helps very much to know what goes into and what comes out of your function
- Try to reuse more of your own code, both in terms of import-statements, and trying to copy everything and making longer than necessary if-else trees.
- Maybe try to allow the user not to read your code to understand how the CLI works