

Lab: Improve the runtime of the mandelbrot boost::python program

take the code in the lab9-mandelboot.zip file, compile the program (using the setup.py facility), use this command:

```
python setup.py build_ext --inplace
```

Then run the code and record the time it took. This lab will give 100 points and will award 100 points for the fastest code, 50 for the second fastest code and 20 for the third fastest code. There is only one rule: Only the `calcz()` function is in C++ the rest has to stay in python. You can change the loop function or the `calcz` function or both to make your code faster. Inspect the code that happens between the time measurements.

The code is already fast, but could improve potentially:

- the C++ `calcz` function: is it most efficient? what could lead to fewer calculations? The `numpy` limit of 255 should not be reduced! Check also the `setup()` facility does it uses the fastest code generation, if not change it (there are C++ compile flags that are `-O3` that is usually faster than `-O`)
- the loop function is calculating 1D and then reshaping, is this the fastest way to handle this? Check `numpy` whether there are faster ways to handle this.

Your report must contain your timings of the first (unchanged) code, for example you could report the ratio of the time of the new code over the time of the old code.

And the complete directory as a zip file.

Due date: Sunday April 21, 11:59pm