

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

Reviewer 1, UiO-username: laurahol, laura.holewa@sfe.uio.no

October 15, 2017

1 Introduction

1.1 Goal

The review should provide feedback on the solution to the student. The main goal is to *give constructive feedback and advice* on how to improve the solution. You, the peer-review team, can decide how you organise the peer-review work between you.

1.2 Guidelines

For each (coding) exercise, one should review the following points:

- Is the code **working as expected**? For non-internal functions (in particular for scripts that are run from the command-line), does the program handle invalid inputs sensibly?
- Is the code **well documented**? Are there docstrings and are they useful?
- Is the code written in **Pythonic way**¹? Is the code easy to read? Are the variable/class/function names sensible? Do you find overuse of classes or not sufficient use of functions or classes? Are there parts of the program that are hard to understand?
- Can you find **unnecessarily complicated parts** of the program? If so, suggest an improved implementation.
- List the programming parts that are not answered.

Use (shortened) code snippets where appropriate to show how to improve the solution.

1.3 Points

The review is completed by pushing the review Latex source file (.tex files) and the PDF files to each of the reviewed repositories. The name of the files should be *feedback.tex* and *feedback.pdf* in the students assignment4 directory.

You will get up to 10 points for delivering the peer-reviews. Each of you should contribute to the review roughly equivalently - your team will get the same number of points².

2 Review - *to be filled out*

Specify the system (Python version, operating system, ...) that was used for the review.

This review was done with Python version 3.5.2. The operating system was Linux Ubuntu.

General feedback

Use this section to give general feedback about the solution such as advice for improved programming or documentation style.

In general, the code was well written and functioned as expected. It would be beneficial to add more comments in the future and to ensure that all parts of assignment sections are answered.

¹<https://www.python.org/dev/peps/pep-0020/>

²In case a team-member does not contribute, please email simon@simula.no

Assignment 4.1

Add a review based on section 1.2. Do the tests have a meaningful name?

The tests do have meaningful names. The tests worked as expected. There were no docstrings, but it is not clear that any were needed as the test names and print statements made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. It appears that there is no test of integrating $2*x$ from 0 to 1 numerically, with a verification that the error is about $1/N$. That part appears to be missing.

Assignment 4.2

Add a review based on section 1.2.

The names are meaningful. The functions work as expected. There were no docstrings, but it is not clear that any were needed as the variable names and print statements made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. It would be helpful if the axes of the quadratic error plot were labeled and there was a plot title. I don't think that the error is plotted as a function of N because the y -values do not decrease as the x -values increase.

Assignment 4.3

Add a review based on section 1.2. In addition, review the following assignment specific items:

- Is numpy being used effectively (e.i. vectorization where possible)?

Numpy was used effectively. Vectorization was done when possible.

The names are meaningful. The functions work as expected. There were no docstrings, but it is not clear that any were needed as the variable names made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. Assignment 4.3 says to add "tests" to the test file, but only one test of the numpy integrator was added. Adding more tests would be beneficial.

Assignment 4.4

Add a review based on section 1.2.

The names are meaningful. Numba did not decrease the ammount of time to perform the integration, but I don't know whether numba should necessarily decrease the ammount of time to perform the integration in this particular case. There were no docstrings, but it is not clear that any were needed as the variable names made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. The question of how much speed was gained when switching to numba was not answered. Also, the question of whether there were advantages to using numba over numpy was not answered.

Assignment 4.5

Add a review based on section 1.2. If you are reviewing a INF3331 student, you can skip this review.

This section was not done.

Assignment 4.6

Add a review based on section 1.2.

The names are meaningful. The functions work as expected. There were some comments and the variable names made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. The question of finding the required N to get within 10 to the -10 th power of the actual answer was not answered.

Assignment 4.7

Add a review based on section 1.2.

It appears that the package was set up correctly.

Assignment 4.8

Add a review based on section 1.2.

The names are meaningful. The functions appear to work as expected. The variable names made it obvious what the code was doing. The code was written in a Pythonic way and was not not unnecessarily complicated. I am not sure what the actual answers are for the integrals of the weird functions, but the processes used in this case seemed to be correct.

2.1 Useful Latex snippets

Here are some sample usage of Latex.

2.1.1 Sample code

```
1 import sys
2 print "This is a sample code"
3 sys.exit(0)
```

2.1.2 Mathematical equation

$$2\pi > 6 \tag{1}$$