

Student Names: Zhang Chi,
Student ID: 5740568
Student Names: Shen Qiwei
Student ID: 5687500
Link: <https://github.com/zhangchi233/nobody.git>
Words count: Part 1: 183 words
Part 2: 370 words

Reports of Assignment 4

1. The difference between C++ and python

Firstly, Python is an interpreted programming language, and C++ is a compiled programming language. The data type needed to be instructed in C++, but in python, they just automatically identified and assigned by the computer. And C++ generates an executable file to instruct what the computer to do while Python just interprets line by line from the source code to tell what the computer should do.

In this N-body simulation, the coding way to calculate motion and store the data are different. In C++, there is a new class defined for three dimension data x , y , z , and operation for a different operator. Then define a new class to define body composed of coordinates, speed, and energy and use a self-defined class method to do the calculation in function. But in python, no new classes have been created, only use dictionary to store body information in different lists. Applying the deep copy and shallow copy in python, modifying the lists, and conveying the modification to the dictionary store the position and speed information in python.

2. The processing of how we solve the task

We checked the code of Python and C++ to understand the meaning of variables and functions and what each block wants to do. Then we read the input port setting of program and choose a proper region to add the i/o function so that we can store the motion trace of planet bodies defined by the programmer.

We wrote another Python script, *test.py* to test programs (C++ debugger, C++ release, and Python). Used `os.system()` to simulate shell command in the Python script, and took `time.counter_per` to calculate the beginning and end of each call. Substation of two `time.counter_per` is the runtime of each section. Then we stored the runtime into a dictionary and convert that dictionary into DataFrame through the pandas.

We discussed how to modify the code first, and the basic frame of program and test is decided before the coding job begins. After that we divide the job, one writes C++, and one writes python. The test file is written by the one who writes CPP because he is more familiar with matplotlib, while another guy writes the report. Lastly, we establish a repository and push our job to GitHub.

The difficulties we faced and how we solve them are a below:

First, the long testing time. When the number of iterations goes high, it takes too long to run. To save time, in the Python section we disable the i/o function and use the multiprocessing library to do multiprocessing save and plot runtime data (in *test.py*). After we made sure the result is successful, we used the traditional single thread method to output the result and enable the i/o function to output the trace of planet body movement. For insurance, in the last version, we all use the single thread for runtime plot and output movement trace. Noticing the long time of i/o manipulation, to remove the effect of memory read and write. We compared the runtime between three modes without any file output.

Another issue is that we need to learn, something hasn't been taught, we just directly google those issues and find out how to solve them such as how to write CSV file in C++.

3. Conclusion

The expected resource we got is four CSV files and three graphs: a CSV file record runtime, three CSV files record the movement trace when iteration time is equal to 5000, one graph plot the runtime curve of python and C++ (Fig. 1), and four screenshots of QGIS with loaded the CSV files record the movement trace, created by Python programs (Fig. 2-3), and C++ programs (Fig. 4-5), with one overview and one close-up respectively.

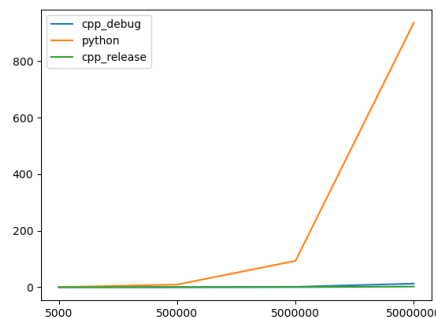


Figure 1. The runtime curve of python and C++

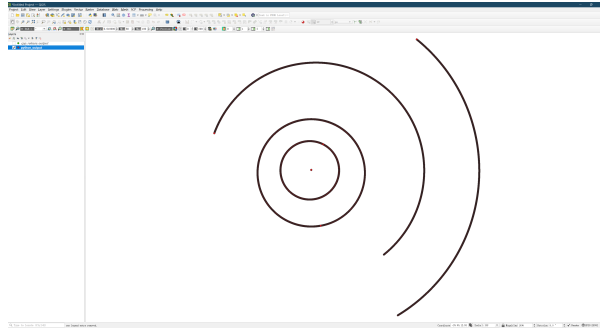


Figure 2. Overview of the trace of N-body movements (Created by Python Programs)

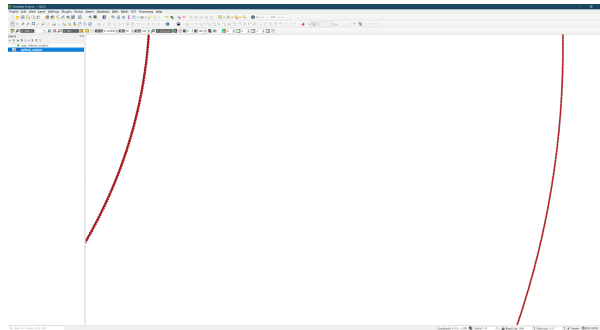


Figure 3. Close-up view of the trace of N-body movements (Created by Python Programs)

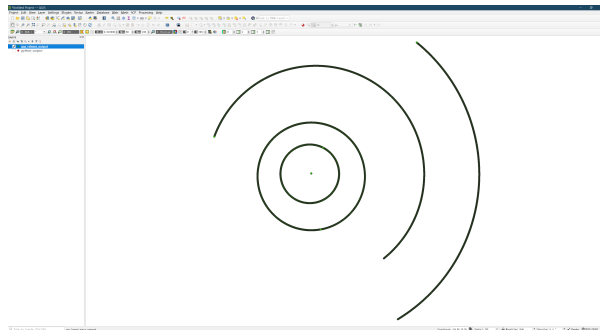


Figure 4. Overview of the trace of N-body movements (Created by C++ Programs)

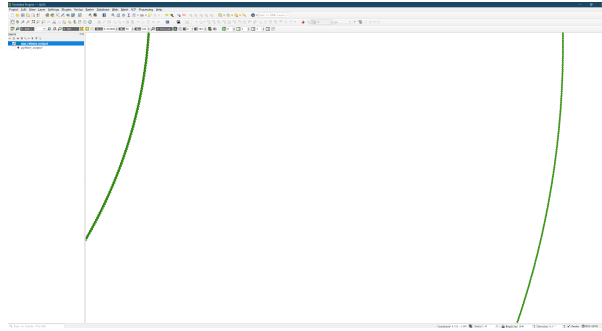


Figure 5. Close-up view of the trance of N-body movements (Created by C++ Programs)