Master in Particle Physics and Cosmology (UC-UIMP)

# A brief comment on Best Practices for Scientific Computing

Laura Trujillo Taborda.

*"The only thing that reliably increases programmer productivity is better working practices"* - Gregory V. Wilson.

Since modern scientific research requires managing and analyzing large amounts of data, software development becomes an important tool for research and thus, its correct practices are necessary to learn in order to improve efficiency, productivity and reliability of the software. On the other hand, software is often used for more than a single project and reused as well for other researchers and as a result, if there is no control on changes made by other peers, this will be likely lead to negative and several impacts on the scientific process due to computing errors. For the reasons presented above and many others, the adaptation of practices like easiness to read code, well documented code, optimization and debuggers are becoming not only necessary but vital for our daily research projects.

Despite lack of basic software development fundamentals, nowadays scientists and researchers are more and more self-aware of how simple their work can become if they start to apply such tools. More common and useful practices such as keeping track of changes using *Version Control Systems* (VCS) and *adding assertions* and *test cases* allow us to revert previous changes if something goes wrong, to highlight differences and to ensure the code to run properly with outputs being consistent. Hence, the motivation to acquire those fundamentals and making them through our path to become a young researcher is essential.

Some difficulties, whatsoever, could arise since some of those practices are not so straightforward to follow such as determining what the most productive way to enhance the code is and identifying possibles bottlenecks (*i.e* a part of the code which is making slower the performance of the program). The reason for so is that it requires an empirical knowledge rather than a theoretical one, even though it is matter of time and effort to achieve such experience. Consequently, once one get familiarized with the fundamentals of software development, acquiring the experience needed to develop some sensitivity when it comes to optimize the code, one could observe improvements in efficiency and the project could start to be more trustworthy and reliable for other peers to use.

Finally, as for myself I find all of those implementations enlighten the path to be a careful researcher as well as a path to achieve a level of rigor increasingly professional and, of course, rewarding. On account of that, I try to keep in practice the use of VCS, proper documentation of the code and code recycling (creating functions to avoid *copy/paste* of the code and so on with the proper use of test cases) and evidently, to train myself and learn new tools as there is still so much left to grasp. This not only improves the quality of the code but allows us eventually to be worried about science rather than about tech issues.

## References

[1] Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, et al. (2014) *Best Practices for Scientific Computing*. PLOS Biology 12(1): e1001745. https://doi.org/10.1371/journal.pbio.1001745