



Contribution ID: 26

Type: **Talk**

## Testing and validation approaches for scientific software

*Tuesday, 25 September 2018 12:00 (20 minutes)*

Nowadays, even though software has a fundamental role in scientific research, the wide majority of scientists is primarily self-taught and received no formal training in software engineering, with often leads to quality and reproducibility problems[1]. The space industry is in a similar situation, with many incident reports describing “various aspects of complacency and a discounting or misunderstanding of the risks associated with software”[2][3].

One of the most useful engineering techniques, software testing, is also the one that presents the biggest gap between its perceived importance and the skill level of scientists in it[4]. Testing, as well as other good practices such as version control and code reviews, not only make code more reusable but also increase the productivity of the developer[5]. However, the special nature of scientific or algorithmic software makes it difficult to apply commonplace testing practices, since the challenges lie in “separating software bugs from model errors and approximation error”[4].

In this talk we will discuss some testing approaches (or lack thereof) present in scientific software that fall short in helping the developers find errors or increase their productivity, and propose some other strategies based on our experience with poliastro, an open source Python library for Astrodynamics[6]. These strategies make use of automated testing frameworks, help covering test cases in an exhaustive way, take advantage of analytical solutions of the problems at hand or public data when available, and guarantee self consistency when there is nothing to compare against. Finally, we will analyze the limitations of these approaches and discuss possible solutions.

### References

- [1]: Wilson, Greg, Dhavide A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven HD Haddock et al. “Best practices for scientific computing.” PLoS biology 12, no. 1 (2014): e1001745.
- [2]: Leveson, Nancy G. “Role of software in spacecraft accidents.” Journal of spacecraft and Rockets 41, no. 4 (2004): 564-575.
- [3]: Lutz, Robyn R. “Analyzing software requirements errors in safety-critical, embedded systems.” In Requirements Engineering, 1993., Proceedings of IEEE International Symposium on, pp. 126-133. IEEE, 1993.
- [4]: Hannay, Jo Erskine, Carolyn MacLeod, Janice Singer, Hans Petter Langtangen, Dietmar Pfahl, and Greg Wilson. “How do scientists develop and use scientific software?.” In Software Engineering for Computational Science and Engineering, 2009. SECSE’09. ICSE Workshop on, pp. 1-8. Ieee, 2009.
- [5]: Wilson, Greg, Dhavide A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven HD Haddock et al. “Best practices for scientific computing.” PLoS biology 12, no. 1 (2014): e1001745.
- [6]: Juan Luis Cano Rodríguez, Antonio Hidalgo, Shreyas Bapat, Nikita Astrakhantsev, Alberto Lorenzo Márquez, Pablo Rodríguez Robles, Ritiek Malhotra, et al. 2018. “Poliastro/poliastro: Poliastro 0.9.1 (pickle Edition)”. Zenodo. doi:10.5281/zenodo.1245650.

**Primary author:** Mr CANO RODRÍGUEZ, Juan Luis

**Co-author:** Mr ASTRAKHANTSEV, Nikita

**Presenter:** Mr CANO RODRÍGUEZ, Juan Luis

**Session Classification:** Talks

**Track Classification:** Space Science Data and Software