
OOPAO: the Python legacy of OOMAO

Cedric Taissir Heritier^{*1,2,3}, Christophe Verinaud³, and Carlos Correia

¹DOTA, ONERA [F-13661 Salon cedex Air - France] – ONERA – France

²Laboratoire d'Astrophysique de Marseille – Aix Marseille Université, Institut National des Sciences de l'Univers, Centre National d'Études Spatiales [Toulouse], Centre National de la Recherche Scientifique – France

³European Southern Observatory – Germany

Abstract

The list of Adaptive Optics (AO) simulators in the community has constantly been growing, guided by different needs and purposes (Compass, HCIPY, OOMAO, SOAPY, YAO...). In this paper, we present OOPAO (Object Oriented Python Adaptive Optics), a simulation tool based on the Matlab distribution OOMAO to adapt its philosophy to the Python language. This code was initially intended for internal use but the choice was made to make it public as it can benefit the community since it is fully developed in Python. The OOPAO repository is available in free access on GitHub (<https://github.com/cheritier/OOPAO>) with several tutorials.

The tool consists of a full end-to-end simulator designed for AO analysis purposes. The principle is that the light from a given light source can be propagated through multiple objects (Atmosphere, Telescope, Deformable Mirror, Wave-Front Sensors...) among which experimental features can be input, in the spirit of OOMAO.

This code was designed to model an AO system as close as possible from reality to generate pseudo-synthetic interaction matrices for AO instruments and includes the SPRINT algorithm to calibrate the DM/WFS registration. The primary goal of this tool is then to offer modularity to the end user and has been parallelized to perform multi-threaded operations.

The goal of this communication is to introduce the code to the AO community, presenting its main functionalities and comparing its outputs with the theory. As well, an illustration of its application on the GHOST test-bench and on the Papyrus instrument will be presented.

Keywords: AO simulator, Python, End to End

^{*}Speaker