iactsim: a CUDA-accelerated Python simulation framework for Imaging Atmospheric Cherenkov Telescopes

Davide Mollica[a], Ciro Bigongiari[b,c], Antonino D'Aì[a], Fabio Pintore[a]

[a]INAF-IASF Palermo Ugo la Malfa 153, I-90146 Palermo, Italy
[b]INAF-OA Roma, Via Frascati 33, 00078 Monte Porzio Catone (Roma), Italy
[c]ASI – Space Science Data Center, Via del Politecnico s.n.c., 00133 Roma, Italy

ABSTRACT

In this contribution we present `iactsim`: a CUDA-accelerated Imaging Atmospheric Cherenkov Telescopes (IACTs) Python simulation framework. The framework is designed to accelerate ray-tracing and camera response simulation, with the additional aim of providing a flexible and user- and developer-friendly collection of tools for IACT performance studies. These tools are intended to facilitate and accelerate instrument design studies and data analysis. Although the framework is still in its early development phase, we present a real-world case study simulating the ASTRI dual-mirror optical system and its Silicon Photo-Multiplier based Cherenkov camera. This is done to demonstrate the framework capabilities and validate its performance.

iactsim

The framework is written in Python, C++ and CUDA, and utilizes `cupy` for GPU offloading. It is aimed to offer a fast, flexible and user/developer-friendly toolkit for IACT performance studies. It allows users to develop their own IACT telescope simulator by providing a set of functions and classes to define custom telescope optical systems and custom Cherenkov cameras, to visualize simulation results, such as photon hits and signal waveforms, and to read data from CORSIKA files.

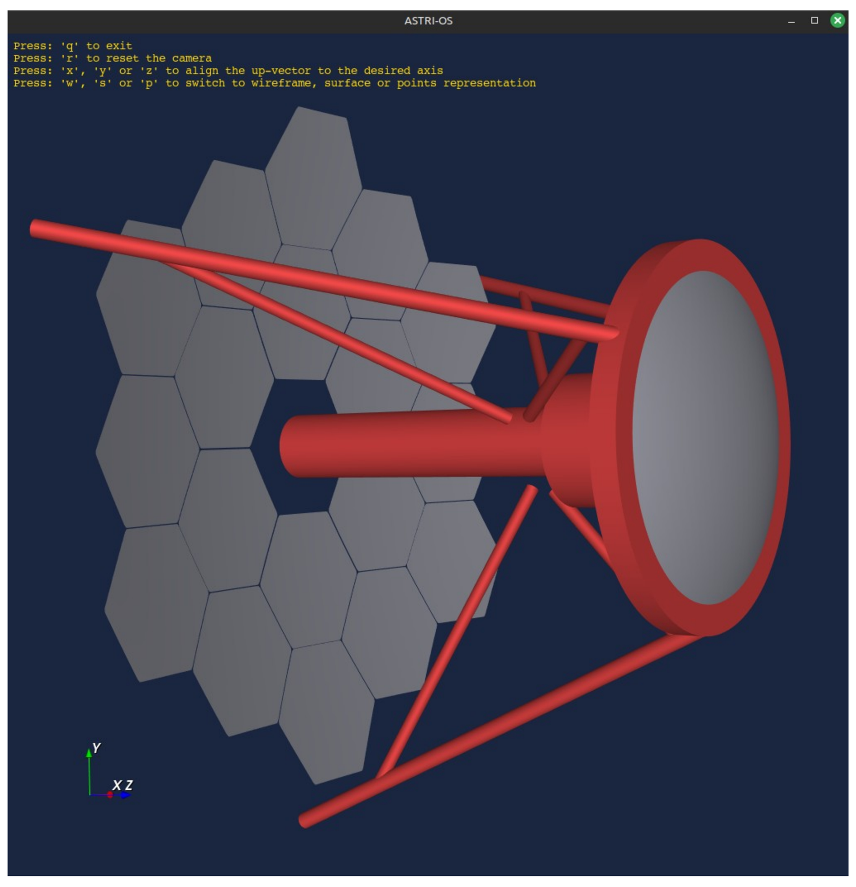

Fig. 1: 3D VTK visualization of the implemented ASTRI optical system geometry.

Optics

The core of the optical part of the framework are CUDA kernels which implement a non-sequential ray-tracing algorithm. These kernels are wrapped by a `IACT` class, which manages kernel calls, preparing and structuring data to match the kernels arguments. The optical system of a telescope can be built with an arbitrary number of surfaces. Each surface is represented by a Python class that handles the surfaces properties, such as shape, type, position, tilt angles and surface Gaussian scattering. Refractive object can also be simulated using a surface as an interface between two materials.

Cherenkov Cameras

The behaviour of a camera electronics is defined in the abstract class `CherenkovCamera`. A camera is composed of a certain number of pixels and channels, which can be *trigger*, *sampling* or *auxiliary* channels. Each channel is defined by its single-photoeletron waveform response. The signal of each channel is computed within a specific time window with custom time resolution. A custom Cherenkov camera can be created implementing the abstract methods that compute pixel signals, and define the trigger logic and the sampling logic to be performed after a camera trigger.

A derived abstract class `CherenkovSiPMCamera` has been implemented to serve as a base for Silicon Photomultiplier (SiPM) cameras. This class defines signal computation for SiPM, simulating optical prompt cross-talk and variation in pixel micro-cells gain. A custom SiPM camera can be defined providing a specific trigger logic and a signal sampling method. To facilitate this, additional CUDA kernels are provided, for instance, kernels that simulate ideal discriminators and ideal analog-to-digital converters.

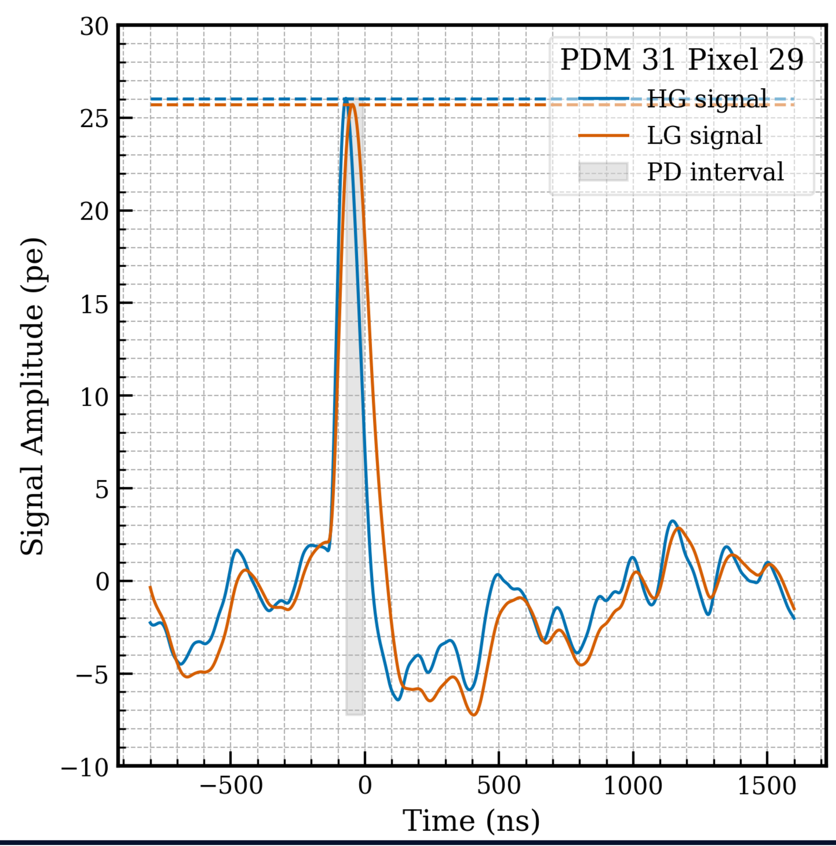

Fig. 2: ASTRI sampling channels simulation of a Cherenkov pulse.

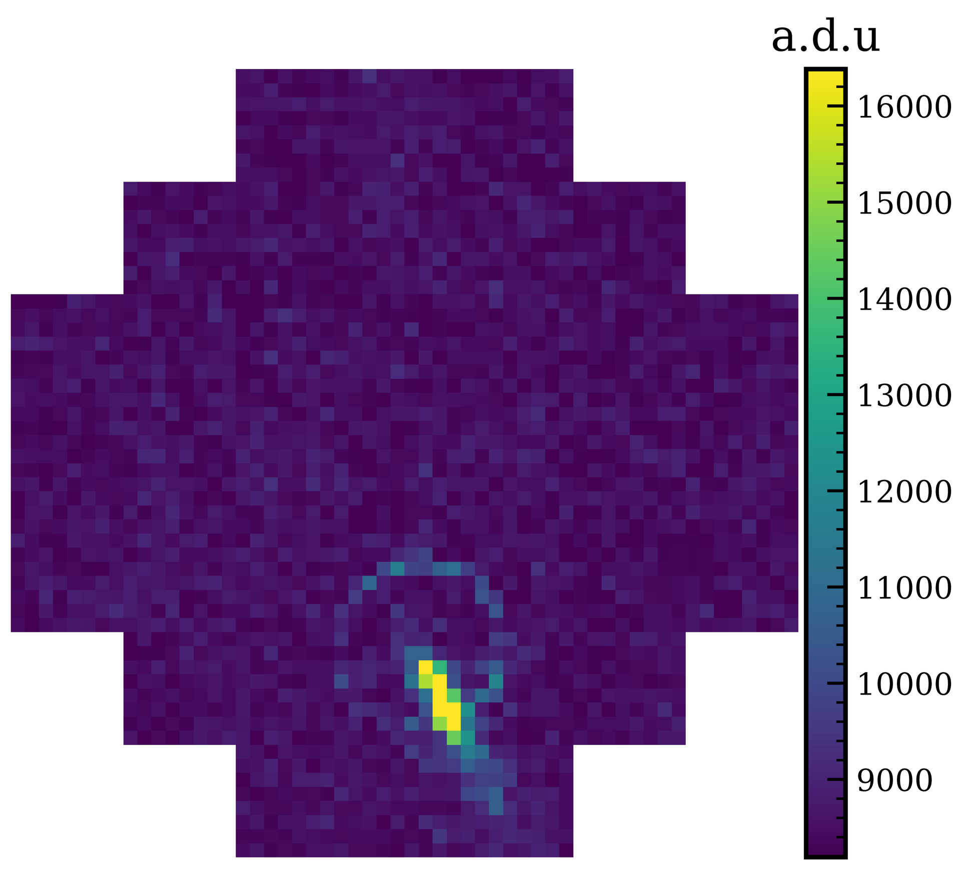

Fig. 3: Simulated ASTRI telescope detection of a proton shower.

ASTRI use-case

The ASTRI[1] telescope use-case has been implemented. The preliminary results here presented have been obtained configuring optical system and camera following measurements performed with the ASTRI-1 telescope. The simulated optical system includes the 18 primary mirror segments, the secondary mirror, the camera protective window and all main shadowing elements. The camera topological trigger logic has been implemented, as well as the signal sampling method based on a peak-detection circuit.

We have validated the framework flexibility by simulating a variety of ASTRI calibration procedures. The preliminary results are promising and the simulator is currently being configured with field measurements in order to match the actual ASTRI-1 camera [2] and optics response. In addition, we are also carrying out a performance comparison with the currently implemented `sim_telarray`-based ASTRI simulator [3]. Preliminary results suggest a speed-up of two orders of magnitude, with the actual value depending on the underlying hardware. However, we found that the reading of CORSIKA output files can be a significant performance bottleneck.

ACKNOWLEDGEMENTS
This work was conducted in the context of the ASTRI Project. We gratefully acknowledge support from the people, agencies, and organisations listed here: http://www.astri.inaf.it/en/library/

References:
[1] S. Scuderi et al., The ASTRI Mini-Array of Cherenkov telescopes at the Observatorio del Teide, Journal of High Energy Astrophysics, 2022
[2] D. Mollica et al., Calibration of the ASTRI-1 telescope camera, in 39th International Cosmic Ray Conference, 2025
[3] F.G. Saturni et al., Advances in the Simulation System of the ASTRI Project, in 39th International Cosmic Ray Conference, 2025

Progetto PNRR CTA+ Missione 4 Istruzione e Ricerca
Proposta IR0000012 Componente 2 Dalla ricerca all'impresa
CUP: C53C22000430006 Linea di investimento 3.1