

High Contrast Imaging for Python (HCIPy)

an open-source adaptive optics and coronagraph simulator



Emiel H. Por, Sebastiaan Y. Haffert, Vikram M. Radhakrishnan,
David S. Doelman, Maaïke A.M. van Kooten, Steven P. Bos

Leiden Observatory, The Netherlands

Contact: por@strw.leidenuniv.nl



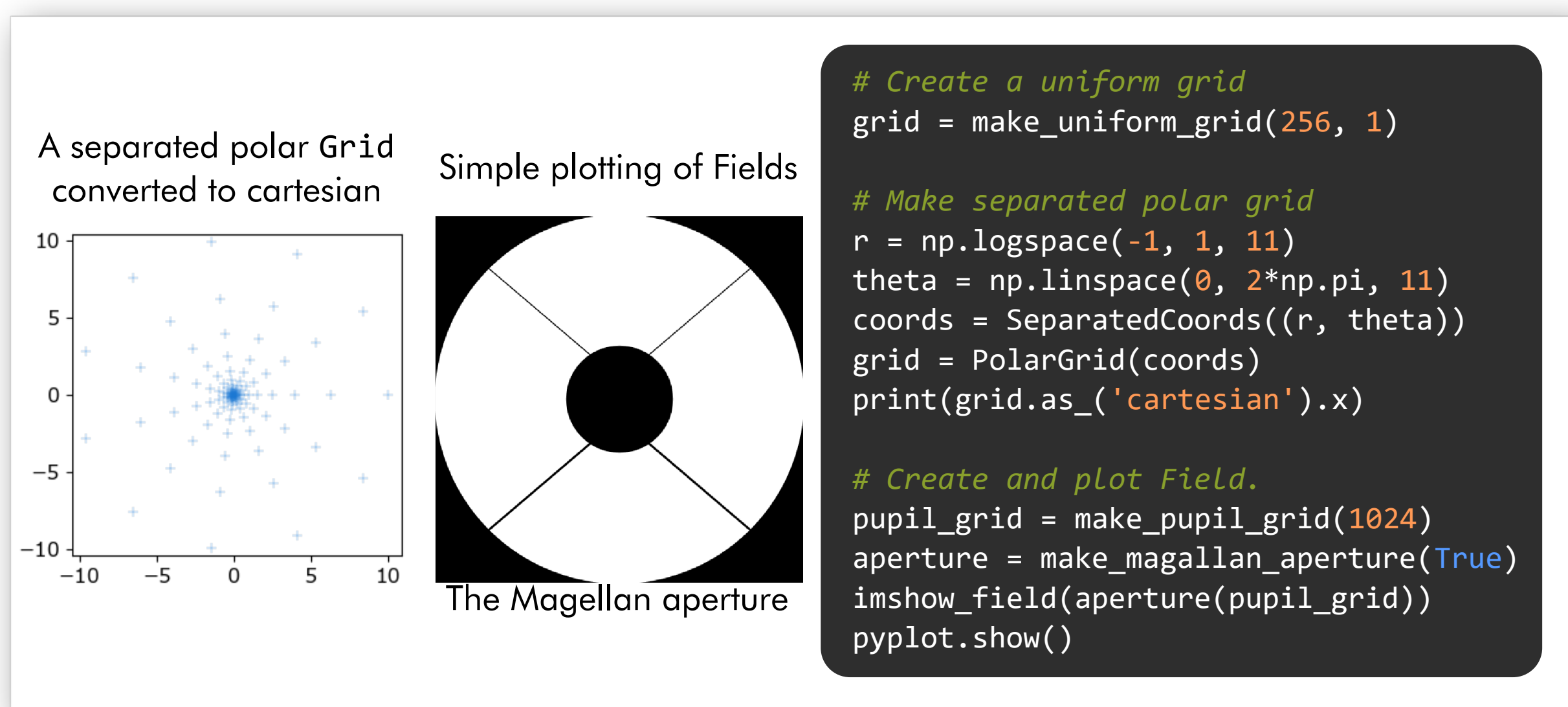
Try HCIPy out for yourself at:
<https://github.com/ehpor/hcipy>

Abstract

HCIPy is designed to facilitate rapid prototyping and end-to-end simulations of high-contrast imaging systems. Written in Python, it provides a modular framework of optical propagations, wavefront sensors and controllers, coronagraphs and more.

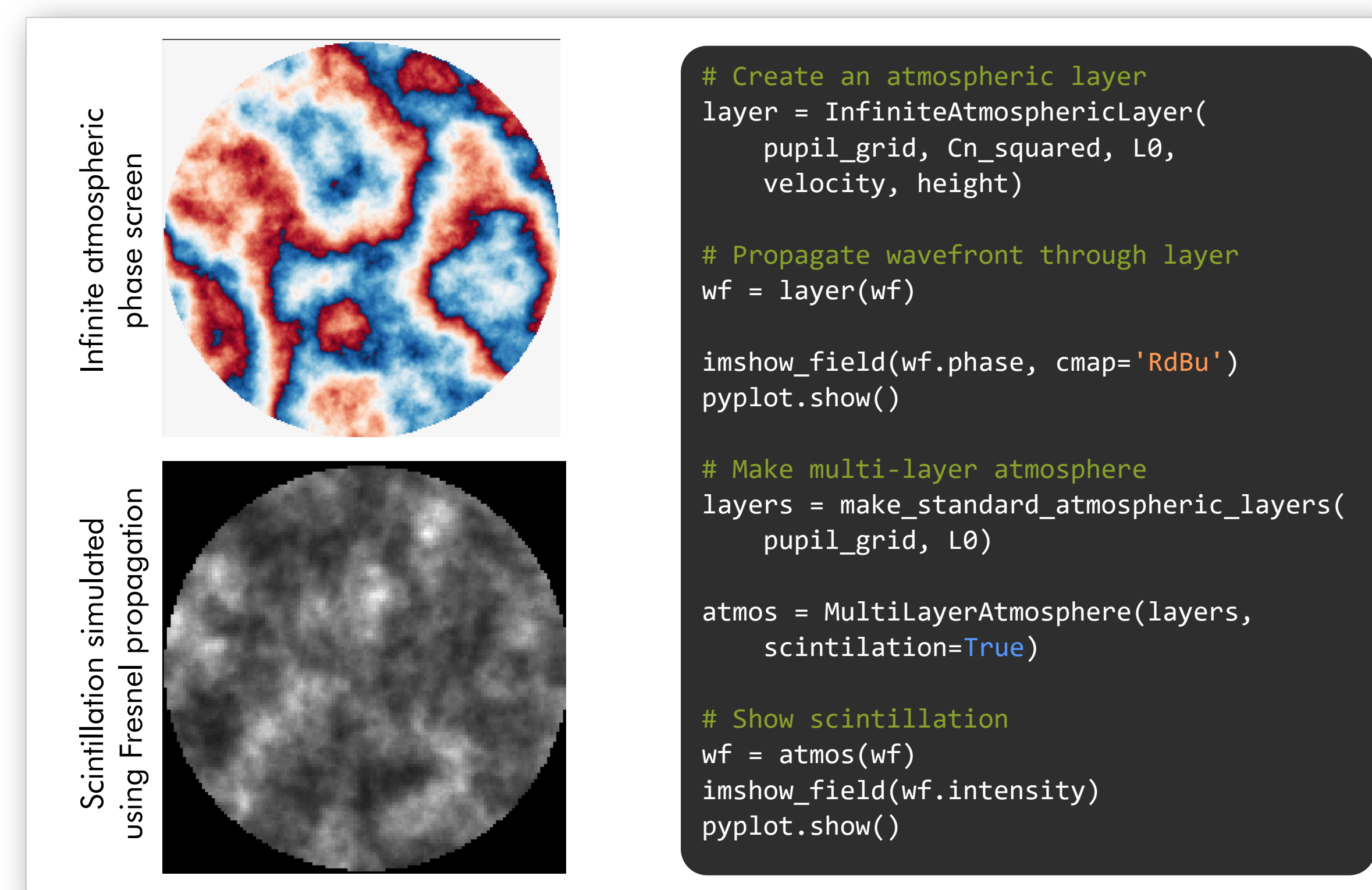
Grids and Fields

Fields in HCIPy unite the value of a quantity with the sampling, the Grid, on which it is calculated. HCIPy uses Fields extensively, to avoid sampling errors by the user.



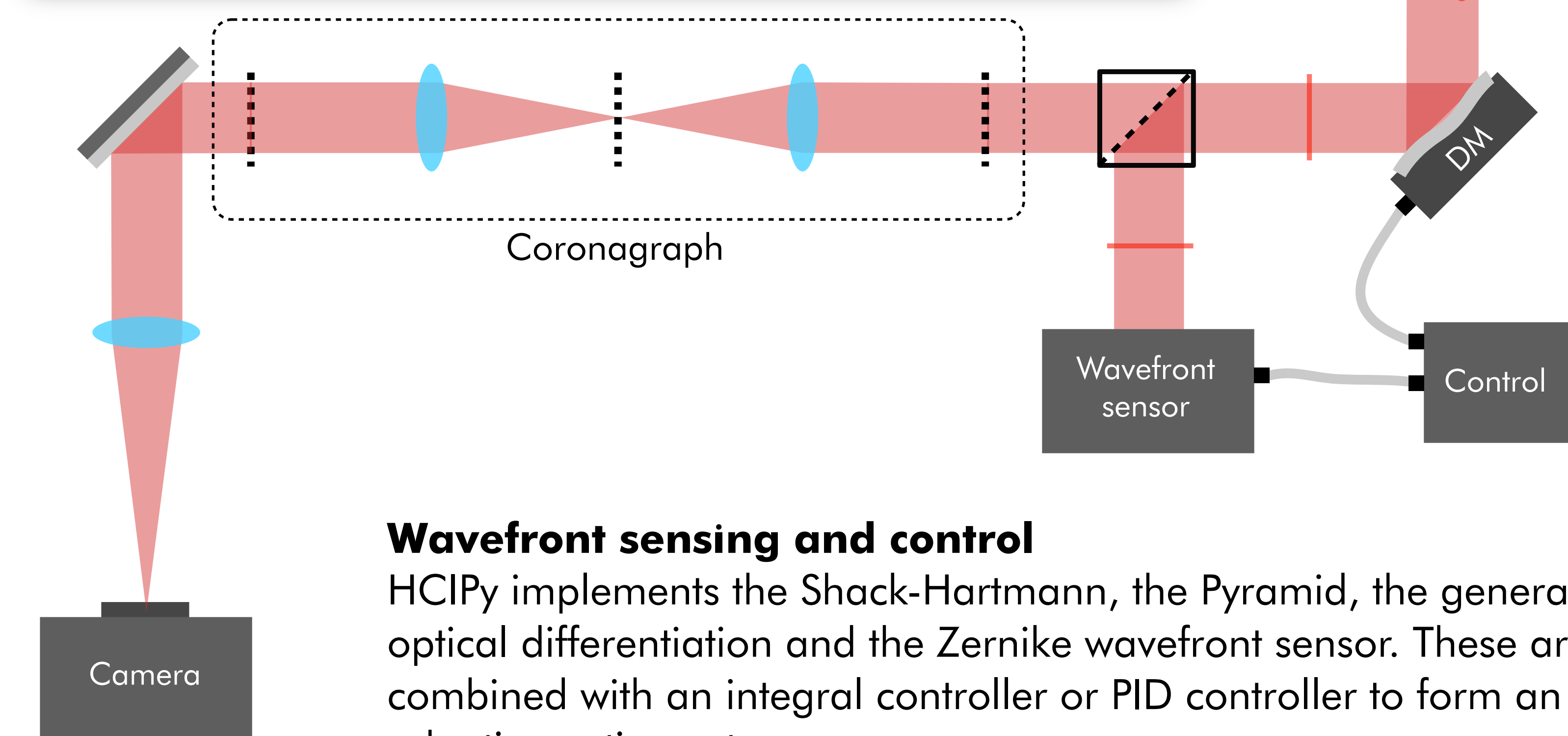
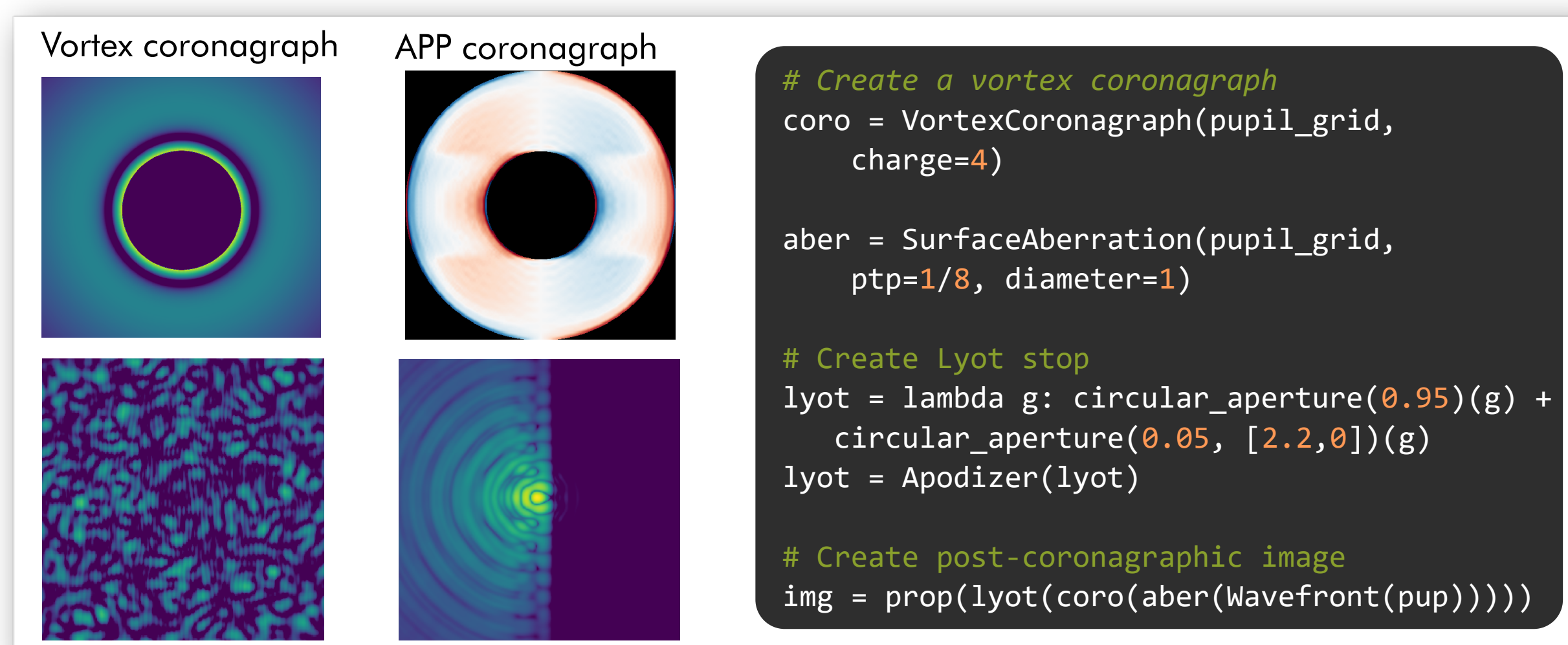
Atmospheric model

HCIPy uses a multi-layer atmospheric model, with each layer moving across the pupil using the "frozen flow" approximation. Both finite (using subharmonics) and infinite (Assémat et al. 2006) phase screens are available. Scintillation is simulated by Fresnel propagating the light between layers.



Coronagraphy

HCIPy implements the apodizing phase plate, the shaped pupil, the (ring-apodized) vortex, the (apodized) Lyot, and the theoretically perfect coronagraph. It also includes globally-optimal optimization methods for some of these.



Wavefront sensing and control

HCIPy implements the Shack-Hartmann, the Pyramid, the generalized optical differentiation and the Zernike wavefront sensor. These are combined with an integral controller or PID controller to form an adaptive optics system.

Polarization

HCIPy uses full Jones vector propagation of polarized light. This allows for simulation of polarization differential imaging and polarization aberrations.

