

pyhector: A Python interface for the simple climate model Hector

Sven N Willner^{1, 2}, Corinne Hartin³, and Robert Gieseke¹

¹ Potsdam Institute for Climate Impact Research ² University of Potsdam ³ Joint Global Change Research Institute

DOI: [10.21105/joss.00248](https://doi.org/10.21105/joss.00248)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Pyhector is a Python interface for the simple climate model Hector (Hartin et al. 2015) developed in C++. Simple climate models like Hector can, for instance, be used in the analysis of scenarios within integrated assessment models like GCAM¹, in the emulation of complex climate models, and in uncertainty analyses.

Hector is an open-source, object oriented, simple global climate carbon cycle model. Its carbon cycle consists of a one pool atmosphere, three terrestrial pools which can be broken down into finer biomes or regions, and four carbon pools in the ocean component. The terrestrial carbon cycle includes primary production and respiration fluxes. The ocean carbon cycle circulates carbon via a simplified thermohaline circulation, calculating air-sea fluxes as well as the marine carbonate system (Hartin et al. 2016).

The model input is time series of greenhouse gas emissions; as example scenarios for these the Pyhector package contains the Representative Concentration Pathways (RCPs)². These were developed to cover the range of baseline and mitigation emissions scenarios and are widely used in climate change research and model intercomparison projects. Using DataFrames from the Python library Pandas (McKinney 2010) as a data structure for the scenarios simplifies generating and adapting scenarios. Other parameters of the Hector model can easily be modified when running the model.

Pyhector can be installed using `pip` from the Python Package Index.³ Source code and issue tracker are available in Pyhector's GitHub repository⁴. Documentation is provided through Readthedocs⁵. Usage examples are also contained in the repository as a Jupyter Notebook (Pérez and Granger 2007; Kluyver et al. 2016). Courtesy of the Mybinder project⁶, the example Notebook can also be executed and modified without installing Pyhector locally.

References

Hartin, C. A., B. Bond-Lamberty, P. Patel, and A. Mundra. 2016. "Ocean Acidification over the Next Three Centuries Using a Simple Global Climate Carbon-Cycle Model:

¹<http://jgcri.github.io/gcam-doc/hector.html>

²<http://tntcat.iiasa.ac.at:8787/RcpDb/dsd?Action=htmlpage&page=welcome>

³<https://pypi.python.org/pypi/pyhector>

⁴<https://github.com/openclimatedata/pyhector>

⁵<http://pyhector.readthedocs.io/en/latest/>

⁶<http://mybinder.org/>

Projections and Sensitivities.” *Biogeosciences* 13 (15): 4329–42. doi:10.5194/bg-13-4329-2016.

Hartin, C. A., P. Patel, A. Schwarber, R. P. Link, and B. P. Bond-Lamberty. 2015. “A Simple Object-Oriented and Open-Source Model for Scientific and Policy Analyses of the Global Climate System – Hector V1.0.” *Geoscientific Model Development* 8 (4): 939–55. doi:10.5194/gmd-8-939-2015.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. “Jupyter Notebooks—a Publishing Format for Reproducible Computational Workflows.” *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90. doi:10.3233/978-1-61499-649-1-87.

McKinney, Wes. 2010. “Data Structures for Statistical Computing in Python.” In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 51–56.

Pérez, Fernando, and Brian E. Granger. 2007. “IPython: A System for Interactive Scientific Computing.” *Computing in Science and Engineering* 9 (3). IEEE Computer Society: 21–29. doi:10.1109/MCSE.2007.53.