

CLIMLAB: a Python toolkit for interactive, process-oriented climate modeling

Brian E. J. Rose¹

¹ Department of Atmospheric and Environmental Sciences, University at Albany (State University of New York)

DOI: [10.21105/joss.00659](https://doi.org/10.21105/joss.00659)

Software

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

Submitted: 23 March 2018

Published: 20 April 2018

Licence

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

Summary

CLIMLAB is an open-ended engine for interactive, process-oriented climate modeling for use in education and research. It is motivated by the need for simpler tools and more reproducible workflows with which to “fill in the gaps” between blackboard-level theory and the results of comprehensive climate models. With CLIMLAB you can interactively mix and match physical model components, or combine simpler process models together into a more comprehensive model. CLIMLAB is used in the classroom (undergraduate and graduate) to put models in the hands of students, and emphasize a hierarchical, process-oriented approach to understanding the key emergent properties of the climate system. CLIMLAB is equally a tool for climate research, where the same needs exist for more robust, process-based understanding and reproducible computational results (Held 2005; Jeevanjee et al. 2017).

CLIMLAB defines a base Python class called `Process`. This generalized model operator contains, at a minimum:

- a dictionary of state variables, each with a well-defined spatial domain
- a list of required input fields
- methods to compute tendencies (rates of change) of its state variables given inputs and current state.

A `Process` object can also contain an arbitrarily complex tree of subprocesses (each also some sub-class of `Process`). Tendencies are then computed by iterating through the subprocess tree and summing up contributions from each member. Using this object-oriented approach, every climate process (radiative, dynamical, physical, turbulent, convective, chemical, etc.) can be simulated as a stand-alone model given appropriate input, or as a component of a more complex model.

CLIMLAB has out-of-the-box support and documented examples for:

- Several atmospheric radiation codes including the widely used [RRTMG model](#) (E. J. Mlawer et al. 1997; S. Clough et al. 2005)
- Convection models including the [Emanuel moist convection scheme](#) (Emanuel 1991; Emanuel and Živković-Rothman 1999)
- Diffusion solvers for energy balance models
- Insolation for arbitrary orbital parameters (e.g. as used by Rose, Cronin, and Bitz 2017)
- Boundary layer turbulence

CLIMLAB allows the user to assemble arbitrary combinations of the above on 1D, 2D, or 3D spatial domains. It is also an open framework allowing arbitrary pieces of useful climate model code to be wrapped into new `Process` classes.

The user-facing parts of CLIMLAB are written in Python and well-suited to interactive command line use. CLIMLAB is agnostic about the underlying numerics; several process modules use compiled Fortran code. Automated build services are used to provide binaries for most common platforms, vastly simplifying the deployment of scientific software to non-specialist students and researchers.

In addition to the documentation, a large collection of example classroom usage can be found in the Jupyter notebook collection at https://github.com/brian-rose/ClimateModeling_courseware.

Links

- Source repository: <https://github.com/brian-rose/climlab>
- Documentation: <http://climlab.readthedocs.io/en/latest/>

Acknowledgement

Development of CLIMLAB is partially supported by the National Science Foundation under award AGS-1455071 to Brian Rose. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- Clough, S.A., M.W. Shepard, E.J. Mlawer, J.S. Delamere, M.J. Iacono, K. Cady-Pereira, S. Boukabara, and P.D. Brown. 2005. “Atmospheric Radiative Transfer Modeling: A Summary of the AER Codes.” *J. Quant. Spectroscopy Radiative Transfer* 91:233–44. <https://doi.org/10.1016/j.jqsrt.2004.05.058>.
- Emanuel, Kerry A. 1991. “A Scheme for Representing Cumulus Convection in Large-Scale Models.” *J. Atmos. Sci.* 48:2313–4. [https://doi.org/10.1175/1520-0469\(1991\)048<2313:ASFRCC>2.0.CO;2](https://doi.org/10.1175/1520-0469(1991)048<2313:ASFRCC>2.0.CO;2).
- Emanuel, Kerry A., and Marina Živković-Rothman. 1999. “Development and Evaluation of a Convection Scheme for Use in Climate Models.” *J. Atmos. Sci.* 56:1766–82. [https://doi.org/10.1175/1520-0469\(1999\)056<1766:DAEOAC>2.0.CO;2](https://doi.org/10.1175/1520-0469(1999)056<1766:DAEOAC>2.0.CO;2).
- Held, Isaac M. 2005. “The Gap Between Simulation and Understanding in Climate Modeling.” *Bull. Amer. Meteor. Soc.* 86 (11):1609–14. <https://doi.org/10.1175/BAMS-86-11-1609>.
- Jeevanjee, Nadir, Pedram Hassanzadeh, Spencer Hill, and Aditi Sheshadri. 2017. “A Perspective on Climate Model Hierarchies.” *J. Adv. Model. Earth Syst.* 9:1760–71. <https://doi.org/10.1002/2017MS001038>.
- Mlawer, Eli J., Steven J. Taubman, Patrick D. Brown, Michael J. Iacono, and Shepard A. Clough. 1997. “Radiative Transfer for Inhomogeneous Atmospheres: RRTM, a Validated Correlated-K Model for the Longwave.” *J. Geophys. Res.* 102:16663–82. <https://doi.org/10.1029/97JD00237>.
- Rose, Brian E. J., Timothy W. Cronin, and Cecilia M. Bitz. 2017. “Ice Caps and Ice Belts: The Effects of Obliquity on Ice–albedo Feedback.” *Astrophys. J.* 846:28. <https://doi.org/10.3847/1538-4357/aa8306>.