pysdm 2.109

Latest version

`pip install pysdm`

Released: Mar 20, 2025

Pythonic particle-based (super-droplet) warm-rain/aqueous-chemistry cloud microphysics package with box, parcel & 1D/2D prescribed-flow examples in Python, Julia and Matlab

Navigation

- Project description
- Release history
- Download files

Verified details

These details have been verified by PyPI

Project links

- Homepage
- Documentation
- Source
- Tracker

GitHub Statistics

- **Repository**
- **Stars:** 67
- **Forks:** 39
- **Open issues:** 146
- **Open PRs:** 30

Maintainers

- AgnieszkaZaba
- prbartman
- slayoo

Unverified details

*These details have **not** been verified by PyPI*

Meta

Project description

PySDM

Python 3 | LLVM Numba | CUDA ThrustRTC | Linux ✓ | macOS ✓ | Windows ✓
Jupyter ✓ | Maintained? yes | Open Hub 44.7K Lines | JOSS 10.21105/joss.03219
DOI 10.5281/zenodo.15056822
EU Funding by FNP | PL Funding by NCN | US DOE Funding by ASR

License GPL v3

tests no status | build passing | codecov 85%
pypi package 2.109 | docs pdoc.dev

PySDM is a package for simulating the dynamics of population of particles. It is intended to serve as a building block for simulation systems modelling fluid flows involving a dispersed phase, with PySDM being responsible for representation of the dispersed phase. Currently, the development is focused on atmospheric cloud physics applications, in particular on modelling the dynamics of particles immersed in moist air using the particle-based (a.k.a. super-droplet) approach to represent aerosol/cloud/rain microphysics. The package features a Pythonic high-performance implementation of the Super-Droplet Method (SDM) Monte-Carlo algorithm for representing collisional growth (Shima et al. 2009), hence the name.

PySDM documentation is maintained at: https://open-atmos.github.io/PySDM

There is a growing set of example Jupyter notebooks exemplifying how to perform various types of calculations and simulations using PySDM. Most of the example notebooks reproduce results and plot from literature, see below for a list of examples and links to the notebooks (which can be either executed or viewed "in the cloud").

There are also a growing set of tutorials, also in the form of Jupyter notebooks. These tutorials are intended for teaching purposes and include short explanations of cloud microphysical concepts paired with widgets for running interactive simulations using PySDM. Each tutorial also comes with a set of questions at the end that can be used as homework problems. Like the examples, these tutorials can be executed or viewed "in the cloud" making it an especially easy way for students to get started.

PySDM has two alternative parallel number-crunching backends available: multi-threaded CPU backend based on Numba and GPU-resident backend built on top of ThrustRTC. The `Numba` backend (aliased `CPU`) features multi-threaded parallelism for multi-core CPUs, it uses the just-in-time compilation technique based on the LLVM infrastructure. The `ThrustRTC` backend