

PyParticle: modular tool to build and analyze aerosol particle populations

Laura Fierce

Payton Beeler

Summary

PyParticle is a lightweight Python library for describing and analyzing aerosol particle populations. The package includes modular builders for aerosol species, particle populations, and particle morphologies, which interface with existing models for derived aerosol properties, such as cloud condensation nuclei (CCN) activity, ice-nucleating particle (INP) potential, and optical properties. Its **builder/registry** design allows new aerosol species, population types, and morphologies to be added by adding small modules to the appropriate **factory/** folders without modifying code API. This modular, reproducible framework facilitates process-level investigations, sensitivity analyses, and intercomparison studies across diverse model and observational datasets.

The core components include:

- **AerosolSpecies**, **AerosolParticle**, **ParticlePopulation** classes that provide a standardized representation of aerosol particles from diverse data sources.
- **Species builder** that supplies physical properties (e.g., density, hygroscopicity parameter, refractive index) for aerosol species, with optional per-species overrides.
- **Population builders** for parametric and model-derived populations (e.g., *binned lognormal*, *monodisperse*, and loaders for *MAM4* and *PartMC* simulation output).
- **Optical particle builders** that compute wavelength- and RH-dependent optical properties for different particle morphologies (e.g., *homogeneous*, *core-shell*, *fractal*) using existing libraries
- **Freezing particle builders** that estimate INP-relevant metrics from particle composition and size.
- An **analysis module** that calculates particle- and population-level variables from PyParticle populations.
- A **viz** package to generate figures from PyParticle populations.

Example scripts demonstrate (i) optical properties for lognormal mixtures, (ii) optical properties for lognormal mixtures with different morphologies (iii) comparisons of CCN activity between MAM4- and PartMC-derived populations, and (iv) freezing-oriented calculations on common temperature/RH grids.

Statement of need

The physical properties of aerosols must be well quantified for a variety of atmospheric, air quality, and industrial applications. A wide range of tools have been developed to simulate (e.g., Riemer et al. 2009; Liu et al. 2016; Bauer et al. 2008; Zaveri et al. 2008) and observe (e.g., Jayne et al. 2000; DeCarlo et al. 2006; Knutson and Whitby 1975; Wang and Flagan 1990; Schwarz et al. 2006, 2008; Zelenyuk et al. 2015) aerosol particle populations, producing varied aerosol data that is often difficult to compare directly. **PyParticle** provides a standardized description of aerosol particle populations and facilitates evaluation of derived aerosol properties.

Leveraging existing models. PyParticle is designed to interoperate with established aerosol-property models rather than reimplementing them. The current implementation focuses on aerosol effects relevant for the atmosphere, with many functions adapted from previous studies of CCN activity (Fierce et al. 2013, 2024; Fierce and McGraw 2017), optical properties (Fierce et al. 2016, 2020), mixing state approximations (Fierce et al. 2015, 2017, 2024), and heterogeneous ice nucleation (Alpert and Knopf 2016). For optical calculations, PyParticle can call external packages such as PyMieScatt (Sumlin et al. 2018) and pyBCabs (Beeler and Chakrabarty 2022). For hygroscopicity and CCN-relevant calculations it follows the kappa-Köhler framework (Petters and Kreidenweis 2007), treating kappa as a per-species property that can be supplied by the species registry or overridden at runtime. Immersion freezing is represented using the stochastic immersion freezing model (Knopf and Alpert 2013; Alpert and Knopf 2016). Model loaders (e.g., MAM4, PartMC) convert model outputs to the PyParticle internal representation so downstream analyses (CCN, optics, freezing) can use the same utilities. Where third-party packages are optional (e.g., `netCDF4` for NetCDF I/O or PyMieScatt for optics calculations), PyParticle raises explicit errors with clear remediation so analyses are deterministic and reproducible.

Modular structure. The codebase follows a strict builder/registry pattern so new capabilities are added by dropping a single module into a `factory/` folder. Modules for particle populations (`population/factory/`), optics morphologies (`optics/factory/`), freezing models (`freezing/factory/`), and aerosol species (`species/`) expose a small, well-documented `build(...)` function (or use a decorator-based registry). At runtime, discovery maps the config `type` string to the appropriate builder. This keeps the public API small while enabling application-specific extensions without changing core code.

Implication for practice. The same downstream computations (e.g., CCN spectra, optical coefficients, freezing propensity) can be applied to many descriptions of aerosol particle populations. The current implementation includes modules for representing populations from PartMC, MAM4, or parameterized distributions, each defined with identical configurations. Because species properties and morphologies are provided through modular factories, sensitivity studies (e.g., refractive indices, mixing rules, or kappa values) become simple configuration changes rather than code modifications. This encourages transparent, process-level benchmarking across diverse datasets.

Software description

Design & architecture

The repository is organized around clear extension points:

- `species/` — The species registry provides canonical physical properties (e.g., density [kg m^{-3}], kappa [-], molar mass [kg mol^{-1}], surface tension [N m^{-1}]) and a file/registry fallback. Public helpers include `register_species(...)`, `get_species(name, **mods)`, `list_species()`, and `retrieve_one_species(...)`. Resolution order is (1) the environment override `PYPARTICLE_DATA_PATH/species_data` (2) packaged data in `PyParticle/datasets/species_data/aero_data.dat`, then (3) a user-specified `specdata_path`. Per-species overrides (e.g., `{"SO4": {"kappa": 0.6}}`) apply at load time.

```
from PyParticle.species.registry import get_species
so4 = get_species("SO4", kappa=0.6)
```

- `aerosol_particle` — Defines the `Particle` class and helpers to build particles from species names, mass fractions, and diameters. A `Particle` stores per-species masses, dry/wet diameters, effective kappa, and basic metadata. Helpers provide kappa-Köhler growth and CCN activity (Petters and Kreidenweis 2007). By default, CCN is treated with the homogeneous-sphere assumption and water surface tension.

```
from PyParticle.aerosol_particle import make_particle
p = make_particle(D=100e-9, aero_spec_names=["SO4"], aero_spec_frac=[1.0], D_is_wet=True)
print(p.get_Ddry(), p.get_tkappa())
```

- **population/** — Exposes `build_population(config)` and a discovery system mapping `config["type"]` to a module in `population/factory/`. The config settings are specific to the population type. For example, the `binned_lognormals` builder requires "N", "GMD" (m), "GSD", "aero_spec_names", "aero_spec_fracs", and binning parameters. The `partmc` builder, on the other hand, requires specification of the directory where simulation data is located and the simulation timestep. Each population-builder produces a standardized particle population, defined by a list of constituent `aerosol_species`, an array of `species_masses` that provide the mass [kg] of each species in each particle, an array of `num_conc` that describe the number concentration [m^{-3}] associated with each computational particle, and a list of ids for each particle. An individual `Particle` is added to or extracted from the `ParticlePopulation` using the attached `add_particle` and `get_particle` functions, respectively.

```
from PyParticle.population import build_population
pop = build_population({"type": "binned_lognormals", "N": [1e7], "GMD": [100e-9],
                       "GSD": [1.6], "aero_spec_names": [[["SO4"]]],
                       "aero_spec_fracs": [[1.0]], "N_bins": 120})
```

- **optics/** — `build_optical_population(pop, config)` attaches per-particle optical properties over `wvl_grid` (m) and `rh_grid` (default [0.0]). Morphologies (`homogeneous`, `core_shell`, `fractal`) compute per-particle scattering, absorption, and extinction cross-sections and asymmetry parameter g . The resulting `OpticalPopulation` aggregates to population-level coefficients (m^{-1}).

```
from PyParticle.optics import build_optical_population
opt_pop = build_optical_population(pop, {"type": "homogeneous", "wvl_grid": [550e-9], "rh_grid": [0.0]}
print(opt_pop.get_optical_coeff("b_scat", rh=0.0, wvl=550e-9))
```

- **freezing/** — Contains routines for assessing the potential for immersion freezing. The freezing module uses particle composition and surface area to calculate freezing proxies and exposes a builder pattern so new parameterizations can be added. Accepts `Particle` or `ParticlePopulation` inputs and returns particle-level and population metrics (e.g., activated fraction vs. T for a given cooling rate).

```
from PyParticle.freezing import build_freezing_population
freezing_pop = build_freezing_population(pop, {"morphology": "homogeneous", "T_grid": [-100, -80, -60, -40]}
freezing_pop.get_frozen_fraction(-1.0) # specified cooling rate in K/s or C/s
```

- **analysis/** — Provides utilities for size distributions ($dN/d\ln D$), moments, mass/volume fractions, hygroscopic growth factors, and CCN spectra. Returns NumPy arrays or lightweight dataclasses for plotting and statistics.
- **viz/** — Provides plotter builders, style management, and grid helpers for consistent visualization of population outputs.

Implementation notes. The codebase uses SI units internally (meters for diameters/wavelengths) and defaults `rh_grid` to [0.0]. Optional dependencies such as `netCDF4` or `PyMieScatt` are imported only where needed; in their absence the code raises `ModuleNotFoundError` with an actionable message rather than silently substituting mock data.

Acknowledgements

The PyParticle package was originally developed under the Integrated Cloud, Land-surface, and Aerosol System Study (ICLASS), a Science Focus Area of the U.S. Department of Energy's Atmospheric System Research program at Pacific Northwest National Laboratory (PNNL). Optics modules and links to the PartMC and MAM4 modules was supported by PNNL's Laboratory Direct Research and Development program. PNNL is a multi-program national laboratory operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

References

- Alpert, Peter A., and Daniel A. Knopf. 2016. "Analysis of Isothermal and Cooling-Rate-Dependent Immersion Freezing by a Unifying Stochastic Ice Nucleation Model." *Atmospheric Chemistry and Physics* 16: 2083–107. <https://doi.org/10.5194/acp-16-2083-2016>.
- Bauer, Susanne E., Douglas L. Wright, Dorothy Koch, et al. 2008. "MATRIX (Multiconfiguration Aerosol TRacker of mIXing State): An Aerosol Microphysical Module for Global Atmospheric Models." *Atmospheric Chemistry and Physics* 8: 6003–35. <https://doi.org/10.5194/acp-8-6003-2008>.
- Beeler, Payton, and Rajan K Chakrabarty. 2022. "Constraining the Particle-Scale Diversity of Black Carbon Light Absorption Using a Unified Framework." *Atmospheric Chemistry and Physics* 22 (22): 14825–36.
- DeCarlo, Peter F., Jay R. Kimmel, Andrew Trimborn, et al. 2006. "Field-Deployable, High-Resolution, Time-of-Flight Aerosol Mass Spectrometer." *Analytical Chemistry* 78 (24): 8281–89. <https://doi.org/10.1021/ac061249n>.
- Fierce, Laura, Tami C. Bond, Susanne E. Bauer, Francisco Mena, and Nicole Riemer. 2016. "Black Carbon Absorption at the Global Scale Is Affected by Particle-Scale Diversity in Composition." *Nature Communications* 7: 12361. <https://doi.org/10.1038/ncomms12361>.
- Fierce, Laura, and Robert L. McGraw. 2017. "Multivariate Quadrature for Representing Cloud Condensation Nuclei Activity of Aerosol Populations." *Journal of Geophysical Research: Atmospheres* 122 (18): 9867–78. <https://doi.org/10.1002/2016JD026335>.
- Fierce, Laura, Timothy B. Onasch, Christopher D. Cappa, et al. 2020. "Radiative Absorption Enhancements by Black Carbon Controlled by Particle-to-Particle Heterogeneity in Composition." *Proceedings of the National Academy of Sciences* 117 (10): 5196–203. <https://doi.org/10.1073/pnas.1919723117>.
- Fierce, Laura, Nicole Riemer, and Tami C Bond. 2017. "Toward Reduced Representation of Mixing State for Simulating Aerosol Effects on Climate." *Bulletin of the American Meteorological Society* 98 (5): 971–80.
- Fierce, Laura, Nicole Riemer, and Tami C. Bond. 2013. "When Is Cloud Condensation Nuclei Activity Sensitive to Particle Mixing State?" *Journal of Geophysical Research: Atmospheres* 118 (24): 13476–88. <https://doi.org/10.1002/2013JD020608>.
- Fierce, Laura, Nicole Riemer, Tami C. Bond, Susanne E. Bauer, Francisco Mena, and Matthew West. 2015. "Explaining Variance in Black Carbon's Aging Timescale and Its Implications for Absorption Enhancement: A Particle-Resolved Modeling Study." *Atmospheric Chemistry and Physics* 15: 3173–91. <https://doi.org/10.5194/acp-15-3173-2015>.
- Fierce, Laura, Yu Yao, Richard C. Easter, et al. 2024. "Quantifying Structural Errors in Cloud Condensation Nuclei Activity from Reduced Representation of Aerosol Size Distributions." *Journal of Aerosol Science* 181: 106388. <https://doi.org/10.1016/j.jaerosci.2024.106388>.
- Jayne, John T., Danna C. Leard, Xuefeng Zhang, et al. 2000. "Development of an Aerosol Mass Spectrometer for Size and Composition Analysis of Submicron Particles." *Aerosol Science and Technology* 33 (1-2): 49–70. <https://doi.org/10.1080/027868200410840>.
- Knopf, Daniel A., and Peter A. Alpert. 2013. "A Water Activity Based Model of Heterogeneous Ice Nucleation Kinetics for Freezing of Water and Aqueous Solution Droplets." *Faraday Discussions* 165: 513–34. <https://doi.org/10.1039/C3FD00035D>.
- Knutson, E. O., and K. T. Whitby. 1975. "Aerosol Classification by Electric Mobility: Apparatus, Theory,

and Applications.” *Journal of Aerosol Science* 6 (6): 443–51. [https://doi.org/10.1016/0021-8502\(75\)90060-9](https://doi.org/10.1016/0021-8502(75)90060-9).

Liu, X., P.-L. Ma, H. Wang, et al. 2016. “Description and Evaluation of a New Four-Mode Version of the Modal Aerosol Module (MAM4) Within Version 5.3 of the Community Atmosphere Model.” *Geoscientific Model Development* 9: 505–22. <https://doi.org/10.5194/gmd-9-505-2016>.

McGraw, Robert. 1997. “Description of Aerosol Dynamics by the Quadrature Method of Moments.” *Aerosol Science and Technology* 27 (2): 255–65. <https://doi.org/10.1080/02786829708965471>.

Petters, Markus D., and Sonia M. Kreidenweis. 2007. “A Single Parameter Representation of Hygroscopic Growth and Cloud Condensation Nucleus Activity.” *Atmospheric Chemistry and Physics* 7 (8): 1961–71. <https://doi.org/10.5194/acp-7-1961-2007>.

Riemer, Nicole, Matthew West, Rahul A. Zaveri, and Richard C. Easter. 2009. “Simulating the Evolution of Soot Mixing State with a Particle-Resolved Aerosol Model.” *Journal of Geophysical Research: Atmospheres* 114: D09202. <https://doi.org/10.1029/2008JD011073>.

Schwarz, Joshua P., R.-S. Gao, David W. Fahey, et al. 2006. “Single-Particle Measurements of Midlatitude Black Carbon and Light-Scattering Aerosols from the Boundary Layer to the Lower Stratosphere.” *Journal of Geophysical Research: Atmospheres* 111: D16207. <https://doi.org/10.1029/2006JD007076>.

Schwarz, Joshua P., Jamie R. Spackman, David W. Fahey, et al. 2008. “Coatings and Their Enhancement of Black Carbon Light Absorption in the Tropical Atmosphere.” *Journal of Geophysical Research: Atmospheres* 113: D03203. <https://doi.org/10.1029/2007JD009042>.

Sumlin, Benjamin J., William R. Heinsohn, and Rajan K. Chakrabarty. 2018. “Retrieving the Aerosol Complex Refractive Index Using PyMieScatt: A Mie Computational Package with Visualization Capabilities.” *Journal of Quantitative Spectroscopy and Radiative Transfer* 205: 127–34. <https://doi.org/10.1016/j.jqsrt.2017.10.012>.

Tilmes, Simone, Michael J. Mills, Yunqian Zhu, et al. 2023. “Description and Performance of a Sectional Aerosol Microphysical Model in the Community Earth System Model (CESM2).” *Geoscientific Model Development* 16: 6087–117. <https://doi.org/10.5194/gmd-16-6087-2023>.

Wang, S.-C., and Richard C. Flagan. 1990. “Scanning Electrical Mobility Spectrometer.” *Aerosol Science and Technology* 13 (2): 230–40. <https://doi.org/10.1080/02786829008959441>.

Yao, Yixuan, Jeffrey H. Curtis, Jason Ching, Zhonghua Zheng, and Nicole Riemer. 2022. “Quantifying the Effects of Mixing State on Aerosol Optical Properties.” *Atmospheric Chemistry and Physics* 22: 9265–82. <https://doi.org/10.5194/acp-22-9265-2022>.

Zaveri, Rahul A., Richard C. Easter, Jerome D. Fast, and Leonid K. Peters. 2008. “Model for Simulating Aerosol Interactions and Chemistry (MOSAIC).” *Journal of Geophysical Research: Atmospheres* 113: D13204. <https://doi.org/10.1029/2007JD008782>.

Zelenyuk, Alla, Dan Imre, Jacqueline M. Wilson, Zhiyuan Zhang, Jun Wang, and Klaus Mueller. 2015. “Airborne Single Particle Mass Spectrometers (SPLAT II & miniSPLAT) and New Software for Data Visualization and Analysis in a Geo-Spatial Context.” *Journal of the American Society for Mass Spectrometry* 26 (2): 257–70. <https://doi.org/10.1007/s13361-014-1043-4>.