




ARTmie: a fast python package for mie scattering and backscattering calculations of single particles, coated particles and log-normal distributed particles

Enrico P. Metzner  

† Karlsruhe Institute of Technology (KIT), Germany ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

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

Summary

In atmospheric science, astronomy and similar areas, the optical properties of particles such as dust, ash and cloud droplets are required when calculating interactions with radiation. The simplest assumption for these particles is that they are spherical, which leads to the fully derived mathematical theory of Mie. The corresponding formulae and first computational algorithms were improved and published by Bohren & Huffman (1983).

ARTmie uses these formulae to calculate extinction, scattering, absorption and backscattering efficiencies as well as cross sections by implementing and porting of the relevant functions (Mätzler, 2002) from Matlab and the Bessel functions (Amos, 1986) from Fortran 77 to C++.

ARTmie was developed with a focus on speed, particularly with regard on particle size distributions and scattering angle weighted backscattering efficiency. ARTmie mainly gains its speed not only by using a C++ backend but also by optimizing the amount of necessary calculations when computing combined information, e.g. for log-normal particle size distributions.

Statement of need

A lot of research in physics and atmospheric and astronomical science depends on the optical properties of aerosols. The greater the variability in the constituents and the larger the number of particles, the more calculations are required. ARTmie is capable to do these calculations at a speed that pure Python code can never achieve.

In addition to the standard Mie efficiencies of extinction, scattering and absorption, ARTmie has the ability to calculate the backscatter efficiency by weighted averaging over the total backward scattering angles, as used in lidar-based measurement systems. Nevertheless, ARTmie is designed to be user-friendly with minimal preparation code required.

Furthermore, ARTmie has recently been used to improve the ICON-ART module of the numerical weather and climate prediction model ICON (Rieger et al., 2015; ?) by recalculating all optical properties of aerosols regularly used by ICON-ART and incorporating additional aerosols with different coatings (Hoshyaripour et al., 2025).

State of the field

There are many Mie scattering libraries available in a variety of programming languages. Python is probably the most accessible programming language for quickly and easily analyzing and testing data within the scientific community. ARTmie is not the first Python module for Mie scattering; therefore, it will be compared to three other commonly used Python libraries:

37 1. miepython (Prah, 2025) :

38 miepython is a minimal, pure Python implementation of classical Mie theory that provides
39 efficient scattering, absorption and extinction calculations for homogeneous spheres only.
40 While it is ideal for simple, educational or small-scale use cases, it lacks support for
41 coated particles or size distributions. Its performance is limited due to the absence of a
42 compiled backend accelerator.

43 2. PyMieScatt (Sumlin et al., 2018) :

44 PyMieScatt is a feature-rich Python library that supports homogeneous and single-coated
45 spheres. It also offers additional tools for data analysis and angular scattering, as well
46 as inversion methods. However, like miepython, it is implemented entirely in Python,
47 making it significantly slower than ARTmie for coated particles and polydisperse systems,
48 where computational demands are higher.

49 3. PyMieSim (Sivry, 2022; Sivry-Houle et al., 2023) :

50 Designed for simulating optical systems, PyMieSim provides a modular interface for
51 defining sources, detectors and particle configurations. It supports both homogeneous
52 and single-coated spheres. Although it uses a C++ backend to optimize performance,
53 the modular design introduces additional setup overhead. PyMieSim is best suited to
54 integrating Mie scattering into broader optical modelling workflows but does not yet
55 support polydisperse particles.

56 **Acknowledgement**

57 This work was done during the funding by the Deutsche Forschungsgemeinschaft (DFG) under
58 the project FE.5250.0010.7772.

59 **References**

- 60 Amos, D. E. (1986). Algorithm 644: A portable package for bessel functions of a complex
61 argument and nonnegative order. *ACM Trans. Math. Softw.*, 12(3), 265–273. <https://doi.org/10.1145/7921.214331>
62
- 63 Bohren, C. F., & Huffman, D. R. (1983). Absorption and scattering of light by small particles.
64 *Wiley-VCH*.
- 65 Hoshyaripour, G. A., Baer, A., Bierbauer, S., Bruckert, J., Brunner, D., Diekmann, C., Förstner,
66 J., Hamzehloo, A., Hanft, V., Keller, C., Klose, M., Ludwig, P., Metzner, E. P., Muth,
67 L., Pauling, A., Ramezani, M., Ziarani, Reddmann, T., Reißig, L., ... Hoose, C. (2025).
68 The atmospheric composition component of the ICON modeling framework: ICON-ART
69 version 2025.04. *Geoscientific Model Development*.
- 70 Mätzler, C. (2002). MATLAB functions for mie scattering and absorption, version 2. *IAP*
71 *Research Report*, 2002-11.
- 72 Prah, S. (2025). *Miepython: Pure python calculation of mie scattering* (Version 3.0.0).
73 Zenodo. <https://doi.org/10.5281/zenodo.15035350>
- 74 Rieger, D., Bangert, M., Bischoff-Gauss, I., Förstner, J., Lundgren, K., Reinert, D., Schröter,
75 J., Vogel, H., Zängl, G., Ruhnke, R., & Vogel, B. (2015). ICON-ART 1.0 – a new online-
76 coupled model system from the global to regional scale. *Geoscientific Model Development*,
77 8(6), 1659–1676. <https://doi.org/10.5194/gmd-8-1659-2015>
- 78 Sivry, M. P. de. (2022). *MartinPdeS/PyMieSim: PyMieSim beta (2.0.0)* (Version v2.0.0).
79 Zenodo. <https://doi.org/10.5281/zenodo.7407114>

- 80 Sivry-Houle, M. P. de, Godbout, N., & Boudoux, C. (2023). PyMieSim: An open-source
81 library for fast and flexible far-field mie scattering simulations. *Opt. Continuum*, 2(3),
82 520–534. <https://doi.org/10.1364/OPTCON.473102>
- 83 Sumlin, B. J., Heinson, W. R., & Chakrabarty, R. K. (2018). Retrieving the aerosol complex
84 refractive index using PyMieScatt: A mie computational package with visualization ca-
85 pabilities. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 205, 127–134.
86 <https://doi.org/https://doi.org/10.1016/j.jqsrt.2017.10.012>

DRAFT