NASA/TM–2022–104606/Vol. 63

Technical Report Series on Global Modeling and Data Assimilation, Volume 63

Randal D. Koster, Editor

GEOS aerosol optical table calculation package `GEOSmie`

Osku Kemppinen, Peter R. Colarco and Patricia Castellanos

October 2022

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Phone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Information Desk
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM–2022–104606/Vol. 63

Technical Report Series on Global Modeling and Data Assimilation, Volume 63

Randal D. Koster, Editor

GEOS aerosol optical table calculation package `GEOSmie`

Osku Kemppinen
University of Maryland, College Park, MD

Peter R. Colarco
NASA Goddard Space Flight Center, Greenbelt, MD

Patricia Castellanos
NASA Goddard Space Flight Center, Greenbelt, MD

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

October 2022

Author Note:
Osku Kemppinen authored this work while employed at the University of Maryland, College Park, MD. The author is now with Meta Platforms, Inc., Menlo Park, CA

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA’s Langley Research Center
Hampton, VA 23681-2199

National Technical Information
Service 5285 Port Royal Road
Springfield, VA 22161
703-605-6000

GEOS aerosol optical table calculation package `GEOSmie`

Osku Kemppinen, Peter R. Colarco and Patricia Castellanos

Contents

1 Introduction **3**
1.1 Description . . . 3
1.2 Output products . . . 3

2 Technical **3**
2.1 Overview . . . 3
2.1.1 Approach to size parameter . . . 4
2.1.2 Structure and dimensions of the optical tables . . . 4
2.1.3 Installation requirements . . . 4
2.2 Mie backend . . . 5
2.2.1 `numba` based inner functions . . . 5
2.2.2 Reused internal Mie variables . . . 5
2.2.3 Core-shell capability . . . 6
2.3 Integration and mixing . . . 6
2.3.1 Multiple subbin distributions . . . 7
2.4 Using pre-calculated kernels . . . 8
2.5 Postprocessing . . . 8
2.5.1 Band averaging . . . 8
2.5.2 Phase matrix moment expansions with generalized spherical functions . . . 8

3 Scientific design and options available **9**
3.1 PSD options and multiple subbin distributions . . . 9
3.1.1 Major size bins . . . 9
3.1.2 Multi-distribution PSD . . . 9
3.1.3 Particle size distribution types . . . 9
3.2 Hydration . . . 10
3.2.1 SS . . . 11
3.2.2 Hydrophobicness . . . 11
3.3 Refractive index files . . . 11
3.3.1 Example refractive index file for GEOS nitrate (first few wavelengths shown) . . . 11

4 User's guide **12**
4.1 How to get and install the code . . . 12
4.2 Particle definition files . . . 12
4.2.1 rhop0 . . . 12
4.2.2 rh . . . 12
4.2.3 rhDep . . . 13
4.2.4 psd . . . 13
4.2.5 ri . . . 14
4.2.6 hydrophobic [OPTIONAL] . . . 14
4.2.7 mode [OPTIONAL] . . . 14

4.2.8 kernel_params [OPTIONAL] 14
4.3 Command-line interface runscripts 15
4.3.1 Base optical tables: runoptics.py 15
4.3.2 Band-averaged optical tables: runbands.py 15
4.3.3 GSF-expanded tables: rungsf.py 16
4.3.4 GRASP-format kernel conversion: runkernelconversion.py 17

A Current GOCART particles 20
A.1 Summary of relevant parameters 20
A.2 Combined plots of fine-mode particles 22
A.3 Black Carbon 26
A.3.1 Microphysical parameter source 26
A.3.2 Particle definition JSON 26
A.3.3 Plots of optical properties 27
A.4 Organic Carbon 30
A.4.1 Microphysical parameter source 30
A.4.2 Particle definition JSON 30
A.4.3 Plots of optical properties 31
A.5 Brown Carbon 34
A.5.1 Microphysical parameter source 34
A.5.2 Particle definition JSON 34
A.5.3 Plots of optical properties 35
A.6 Sulfate 38
A.6.1 Microphysical parameter source 38
A.6.2 Particle definition JSON 38
A.6.3 Plots of optical properties 39
A.7 Nitrate 42
A.7.1 Microphysical parameter source 42
A.7.2 Particle definition JSON 42
A.7.3 Plots of optical properties 43
A.8 Sea salt 50
A.8.1 Microphysical parameter source 50
A.8.2 Particle definition JSON 50
A.8.3 Plots of optical properties 51
A.9 Dust 59
A.9.1 Microphysical parameter source 59
A.9.2 Particle definition JSON 59
A.9.3 Plots of optical properties 60

B Pre-calculated kernel format 63
B.1 Overview 63
B.2 Example: `GEOSmie`-compatible GRASP spheroidal kernel file 64
B.3 Example: Shape distribution file for GRASP kernels (`data/spheroid_fixed.txt`) 65

C NetCDF4 file stucture 66

1 Introduction

1.1 Description

This is a technical guide to the `GEOSmie` package for calculating optical properties of aerosol particles. The package has been developed in the context of the Goddard Earth Observation System (GEOS) Earth System Model (ESM) but is sufficiently general to suit a variety of other applications. The current version of the package utilizes the GOCART approach to aerosol representation [1] in which some aerosol species, namely sulfate, carbonaceous aerosols, and nitrate, are represented by one or more prescribed aerosol particle size distributions (PSD), e.g. both monomodal and multimodal distributions. The package also supports sectional approaches, i.e. multiple size bins, to describing optical properties, as is done in GOCART for dust and sea salt. The package supports a sectional approach with multiple size bins that have unique sub-bin distributions, such as bimodal lognormal distributions. In all cases, size distribution parameters can be different at different relative humidity values due to hygroscopic growth, but otherwise the size distribution within each bin is fixed. While the current version of the package does not support aerosol models with variable internal mixtures, such as CARMA [2, 3] or MAM [4], these are being considered for future development.

1.2 Output products

The package creates three different types of output files for each aerosol species. The first type is the monochromatic optical table, which contains optical values at a selected set of wavelengths. The second type is the band-averaged optical table, which draws from the monochromatic optical table to compute the band-averaged values for specific wavelength bands used in the GEOS-internal radiative transfer codes (e.g., RRTMG). Both of these tables are used online as the GEOS ESM is run, the monochromatic tables for the calculation of diagnostic Aerosol Optical Depth (AOD) values, and the band-averaged tables for radiation calculations within the model. The third table type is a variation on the monochromatic table that can be used for forward radiative transfer models such as VLIDORT for offline radiance calculations generated from GEOS simulated atmospheric profiles. All three types are described in more detail in Section 2.1.2.

2 Technical

2.1 Overview

`GEOSmie` is a tool written in Python to calculate aerosol optical properties for GEOS. It is a re-write of a previous IDL-based tool with improvements to code maintainability and performance. On a high level the package works by calculating optical properties for homogeneous spheres, or interpolating from pre-calculated kernel files, at various parameter values, integrating over the size parameter space, and saving the calculated optical parameters into multidimensional tables presented as NetCDF-format files. See 2.1.2 for details on the dimensions.

Rather than use hardcoded or otherwise *ad hoc* particle definitions, `GEOSmie` strives to be as flexible as possible and allow particle types beyond the specific requirements of GEOS. The way to accomplish that has been to use input data files to define all of the particle microphysical parameters. Particle names have no meaning in the context of the package other than to identify which input data file to use. This also means that although certain GEOS aerosol particles have unique behaviors, for example the size distribution or the relative humidity dependence of sea salt, there is nothing preventing one from using those behaviors with any other particle species. As far as the package is concerned these behaviors are decoupled from each other and can be mixed and matched at will. In other words, any existing or new particle size distribution or relative humidity dependence can be used to describe an arbitrary particle species. Note that implementing completely new size distributions or relative humidity behaviors beyond changing the parameters does require changing the code files.

Following the data-driven approach the package is agnostic to the values in the data files. The user is free to specify as many size bins as they want, covering any size ranges they want (within one's computational constraints, naturally). The same is true for using multiple modes within size bins. The package simply uses all of the modes defined in the data file with no restrictions or special behaviors. Likewise, the set of relative humidity points and wavelengths the values are calculated at are solely determined by the data files. A given application (such as GEOS) may require specific values for these, but for other purposes there are no restrictions from the point of view of `GEOSmie`.

To use the package one can choose from one of the pre-made particle definition JSON files under the `geosparticles/` directory or write one of their own (see Section 4.2 for details). Next, the user runs one or more of the provided command line interface scripts, depending on the types of optical tables needed. Each run will produce optical tables for the particle type specified, and as such, the package must be run once for each particle type desired. Naturally, one can use batch scripts or similar methods to accomplish this with minimal user interaction, and the code is thread-safe for reading the input files so that multiple tables may be generated simultaneously in parallel-executed calls. For more details on which scripts to use and how to use them, see Section 4.3.

Rather than use the command line interface scripts, for certain specific applications it is entirely possible to utilize the Python functions directly. Instructions on how to do that are not included in this document but one can look at the various command line interface scripts to see how the functions are used.

2.1.1 Approach to size parameter

In `GEOSmie` all calculations are done in the size parameter space (i.e., $x = 2\pi r/\lambda$) rather than in the physical radius space. In other words, the size parameters at which individual optical parameters are calculated for a given particle are fixed regardless of the wavelength or the relative humidity. Thus, the particle size distributions (which are defined in the radius space) are converted into size parameter space. This approach is a deviation from how the calculations were done in the original IDL-based version. The reason behind this choice is that having the size parameters be fixed allows the package to utilize an effective form of precomputed internal Mie variables. Many of the variables used in Mie calculations only depend on the size parameter and not on the refractive index, which naturally changes depending on the wavelength and the relative humidity. This approach cuts the computation time of the Mie simulations by approximately half. For more details on the precomputed Mie variables, see Section 2.2.2.

The main downside of this approach is that some derived physical parameters that should stay constant regardless of the wavelength, such as the effective radius or total particle mass, vary slightly due to rounding errors. The reason is that with finite precision numbers, converting values from radius space to size parameter size and back do not result in the same exact number when the wavelength changes. These derived parameters are not saved at individual wavelengths in the optical tables. Instead, the value that is saved is calculated at the smallest wavelength used. This is unlikely to have significant impacts on any results but is something to be aware of given that the choice of the wavelength at which to calculate the value has minor implications via rounding errors.

2.1.2 Structure and dimensions of the optical tables

The optical tables have three dimensions; these correspond to the size bins, relative humidities, and wavelengths stipulated in the input files. Additionally, the six independent, nonzero scattering matrix elements $P_{11}, P_{12}, P_{22}, P_{33}, P_{34}, P_{44}$ are calculated and saved at 371 scattering angles as an additional dimension. There are 100 angles between 0 and 1 degrees (0 included, 1 excluded), 100 angles between 1 and 10 degrees (1 included, 10 excluded), and 171 for the rest of the angle space (10 included, 180 included). It is currently not possible to change the output angle resolution without editing the source code. Within each size bin the optical parameters are calculated at individual size parameter values that span the desired size distribution range at a user-specified sub-bin resolution, and optical parameters are integrated over the desired size distribution. Thus, only the size-integrated values are saved. A sample output of the file structure from `ncdump -c` is shown in Appendix C.

2.1.3 Installation requirements

`GEOSmie` is quite portable and can run on most default Python 3 installations. A few key Python packages that are needed are `netCDF4`, `numpy`, `scipy` and `numba`, the last of which may not be part of some Python installations but can be installed via `pip`, `conda` or similar common tools.

2.2 Mie backend

GEOSmie runs on a Python-based Mie calculator that is forked from the pymiecoated GitHub repository maintained by Jussi Leinonen [5], which itself is a Python version of a MATLAB code developed by C. Matzler [6], which was an implementation of Bohren and Huffman's Mie algorithm [7]. Finally, Bohren and Huffman's Mie algorithm is based on many core algorithms developed by Wiscombe [8].

The modified pymiecoated is called pymiecoated-mm and has been heavily rewritten for improved performance. The two main modifications were moving from exclusively numpy-based calculations to numba-based calculations for many of the inner functions, and re-structuring the objects such that various internal Mie variables can be pre-calculated and re-used for particles of different sizes and/or different refractive indices.

2.2.1 numba based inner functions

While numpy is generally fast for vector and matrix operations, the Mie algorithm used here uses recursion to calculate various quantities. This turns out to be a major bottleneck for numpy based calculations, and a major performance enhancement was achieved by rewriting all functions where the main workload can not be effectively performed by numpy to use numba. This change improved the Mie calculation speed by a factor of 10 or more compared to the numpy-only version, making the code competitive with most compiled languages even without considering the reuse of internal Mie variables. The performance improvement from numpy-only version to numpy/numba hybrid implementation is 12x-18x, as shown in Figure 1.

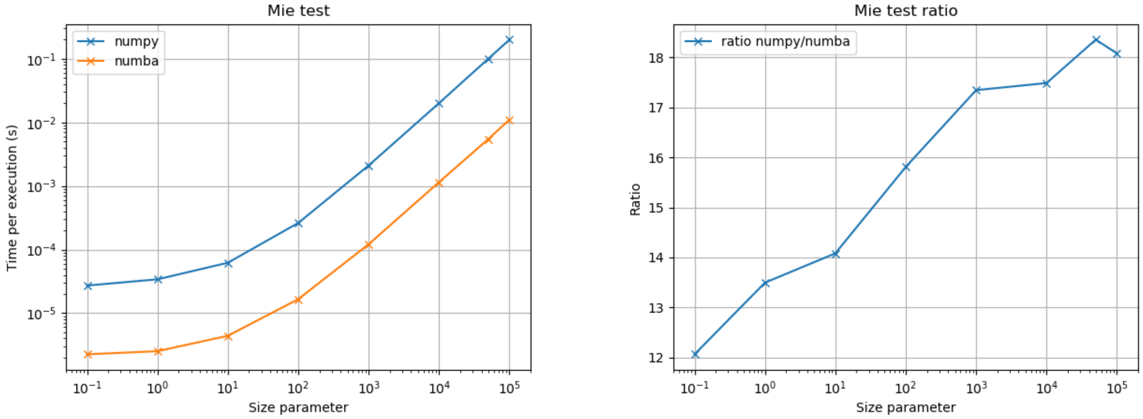


Figure 1: Mie simulation performance with numpy-only Mie code compared to the numpy/numba hybrid Mie code. The left plot shows the average time of one Mie simulation when calculating the optical properties for 1000 spheres at various size parameters. As the size parameter increases the computation takes longer due to more terms needing to be calculated. The right plot shows the ratio of the numpy-only code simulation time and the hybrid code simulation time as a function of the size of the spheres used. That is, the hybrid code is 12x-18x faster than the pure numpy code, depending on the size parameter.

2.2.2 Reused internal Mie variables

For a thorough explanation of the Mie algorithms, the reader is directed to Chapter 4 in [7]. Alternatively, [8] provides a clear, conscise summary of the key equations.

Thare are two main internal Mie variables that are pre-computed and re-used across simulations of particles with different refractive indices. The first are the special Bessel functions of the first and the second kind, J_v and Y_v [9]. When calculating the Mie coefficients a_n and b_n, the Bessel functions are functions of the size of the sphere only and independent of the particle refractive index. As we calculate Mie spheres at fixed size parameter points regardless of the wavelength (rather than fixed physical size points), this allows us to re-use the precomputed Bessel function values across all wavelengths. This corresponds to pre-computing the functions $j_n(x)$ and $h_n(x)$ for all x in Eq. 4.53 in [7].

The second set of internal variables that are reused are the angular eigenfunctions π and τ that are used when calculating the amplitude scattering matrix S_{12}. π and τ are related to associated Legendre polynomials and are calculated with a recursive algorithm that uses only the maximum number of polynomial expansion terms, which depends on the size and the angles at which the values are needed. Since the angles for the amplitude scattering matrix are fixed across all sizes and refractive indices in is possible to precompute π and τ. This is especially important since the recursive algorithm used to calculate them [8] is a significant computational burden. For reference, the calculation of S_{12} elements S_1 and S_2 are shown in Eqs. 1d-1e in [8]. The calculation of π and τ is shown in Eqs. 3-4 and Section III in [8].

The net effect of using pre-computed variables like this is to cut the computational cost of these variables by a very large factor, making the calculation of these variables essentially free. Based on manual accounting of floating-point operations in a single Mie calculation, the calculation of the Bessel functions and the angular eigenfunctions account for roughly half of the all floating-point operations needed. Indeed, as shown in Figure 2, the performance improvement due to this pre-calculation is about 50%, as expected. With a very large number of angles resolved the angular eigenfunction calculation can dominate the computation, in which case the relative performance would improve further. Likewise, in cases where the angular scattering matrix is not needed at all, the computational burden of the angular eigenfunctions would be lower, leading to only a 25% improvement from using pre-calculated values.

It should be noted that in Fig 2 the performance improvement of using `numba`-based inner functions is less than what is shown in Figure 1. This is due to the former being a more realistic benchmark, including a wider size distribution with a significant number of small sizes where `numba`-based performance improvements are less significant.

2.2.3 Core-shell capability

Our default application is for homogeneous spheres, but `pymiecoated-mm` allows one to use a so-called core-shell sphere as well. The core and the shell each have an individual radius and an individual complex refractive index. This capability is currently available via a separate branch in the GEOSmie GitHub repository and is not documented here. Sample JSON files for a coated black carbon particle, as well as an effective medium approximation version of the same particle, are provided in the core-shell branch of the repository, and interested users are encouraged to use them as templates. The output files of the core-shell branch of GEOSmie include the shell size as an additional dimension, rendering the files incompatible with current GEOS interface. As such, the core-shell branch has not been merged with the main GEOSmie repository.

2.3 Integration and mixing

Integration of the scattering properties is straightforward for a case where each size bin is represented by a single size distribution. The weighting used in the integration for any optical parameter is simply the cross-sectional area of the sphere (Eq. 1) multiplied by the number density. However, for the asymmetry parameter g, the integration is also weighted by the scattering efficiency Q_{sca} (Eq. 3).

In practical terms the size distribution is divided into a finite number of subbins. Optical properties are calculated for each of the midpoints of the subbins, and these are integrated together. The distribution of subbins is not uniform. Instead, the size range of a given bin is divided into size decades (e.g. [0.01, 0.1], [0.1, 1.0], [1.0, 10.0], ...), and each decade has a fixed number of subbin points. This helps ensure that both the smaller end as well as the larger end of the size distribution have a reasonable resolution. The number of subbins per decade parameter is set in the particle JSON file and can be adjusted to achieve a desired level of accuracy in the calculation (see Section 4.2.4).

The cross-sectional area of the particle is calculated as

$$A(r) = \pi r^2. \tag{1}$$

For all quantities $q \neq g$:

$$W_{q \neq g}(r) = A(r). \tag{2}$$

For the asymmetry parameter g:

$$W_{q=g}(r) = A(r) * Q_{sca}(r). \tag{3}$$

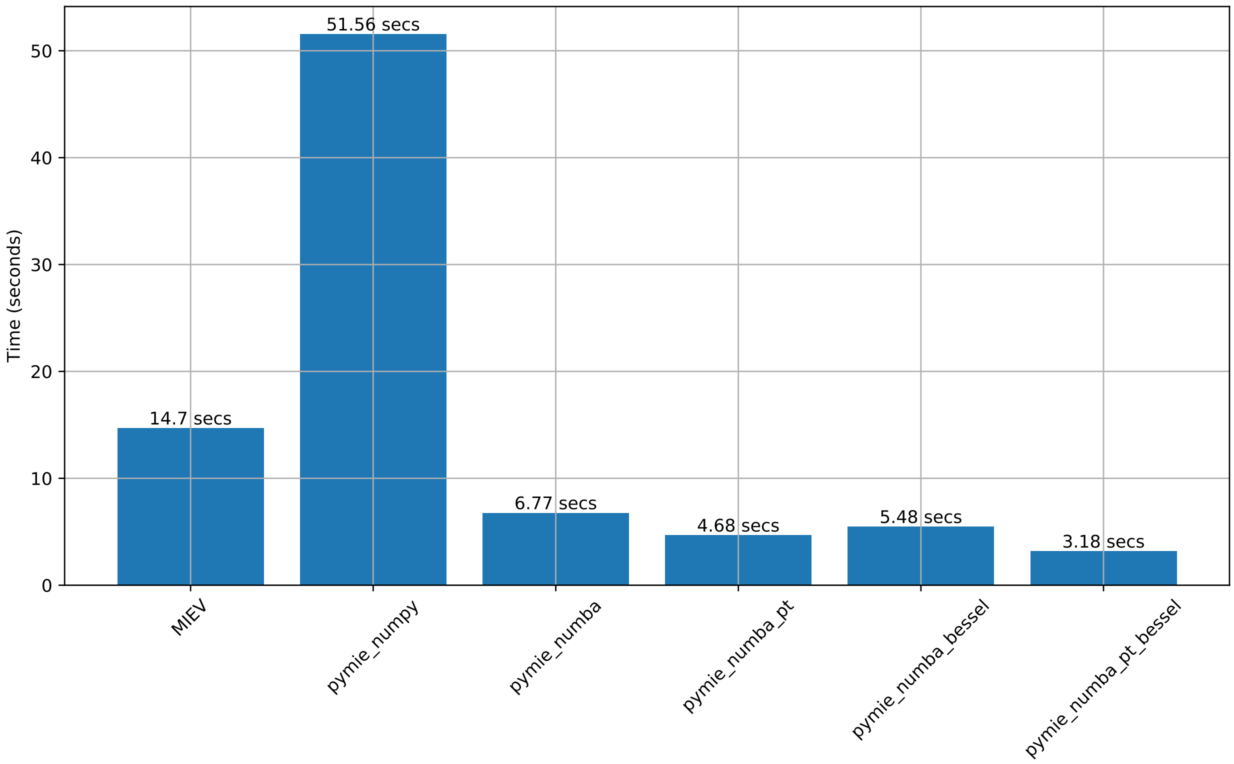


Figure 2: Comparison of the performance improvements due to using pre-computed internal Mie variables for a benchmark test. MIEV is the Fortran Mie routine used in the legacy GEOS optical table generation package. pymie_numpy is the numpy-based `pymiecoated`, i.e. without any of the improvements shown in this work. pymie_numba utilized numba but has no precomputation of Mie parameters, pymie_numba_pt uses numba and precomputed π and τ, pymie_numba_bessel uses numba and precomputed Bessel functions, and pymie_numba_pt_bessel uses numba and precomputed π, τ, and Bessel functions. The time improvement between pymie_numba and pymie_numba_pt_bessel version that used both sets of precomputed Mie parameters is from 6.77s to 3.18s, or a reduction of 53%.

The equation for the integrated values can then be acquired via

$$q_{integrated} = \frac{\sum_r q(r) * dndr * W_q(r)}{\sum_r dndr * W_q(r)}, \tag{4}$$

where q is the quantity to be integrated.

2.3.1 Multiple subbin distributions

When the size bin is represented by multiple subbin size distributions (e.g. multimodal lognormal distribution), the individual distributions need to be combined properly. Specifically, the optical properties of each subbin size distribution are externally mixed by taking the sum of the weighted optical properties for each size distribution and dividing by the sum of the weights.

Other mixing rules have been implemented, such as normalizing the PSD product in Eq. 4 only after different distributions have been summed together, but only the above external mixing rule, which is used for the first size bin of GEOS dust, is available via the default interface. User specification of the mixing rule for multi-distribution size bins is a planned future feature. It should be noted that the multi-distribution sub-bin mixing rule is a general way to mix more than one subbin distribution within a size bin and does not refer exclusively to bimodal lognormal distributions. Fundamentally it is simply a system to calculate optical properties of multiple size distributions within a given size bin and average them together via a given mixing rule. It can be used with any size distribution provided in GEOSmie, including any future ones due to it not being hard-coded for each distribution but a feature of how the integration step is implemented within the code.

For multi-distribution integration the Equation 4 has to be generalized as follows. The additional *subbin* parameter corresponds to the multiple subbin distribution:

$$q_{integrated} = \frac{\sum_{subbin}\sum_r q(r, subbin) * dndr(subbin) * W_q(r, subbin)}{\sum_{subbin}\sum_r dndr(subbin) * W_q(r, subbin)}. \tag{5}$$

2.4 Using pre-calculated kernels

For non-spherical particles GEOSmie offers an option to use pre-calculated kernels in place of direct Mie simulations for a given particle type. For the current GEOS version the main application is using GRASP spheroidal kernels [10, 11, 12] for dust, but GEOSmie can, at least in principle, use any set of pre-calculated kernels as long as they abide by the file format requirements defined in Appendix B. The user simply supplies a file with the kernels and a file that specifies the shape distribution, i.e. the weights for each individual kernel. Using a pre-integrated shape distribution is often desired for faster computations, given that the shape integration can take some time, and this can be achieved by calculating a pre-integrated kernel file with only one shape, and setting the shape weight to 1.0 for that single shape.

Given that the refractive index nodes provided by the kernels rarely match the values that are needed for a given particle, bilinear interpolation to the desired values is used. Kernels are not interpolated along the size parameter dimension. Instead, the size parameter nodes used are simply the ones the kernel tables provide. This leads to unavoidable inaccuracies in the particle size distribution integration, in particular when narrow distributions are used.

2.5 Postprocessing

2.5.1 Band averaging

Alongside the single-scattering optical tables that are calculated at individual wavelengths, band-averaged tables are calculated for use in radiative transfer calculations. This is done as a postprocessing step. At the moment the user can specify a given pre-programmed band scheme and supply it as a keyword argument along with the optical table to be averaged. The averaging algorithm then calculates band-averaged optical values for each size bin and relative humidity value using a specified set of bands, by finding linearly interpolated optical values for a number of individual evenly spaced wavenumbers, and then averaging over those optical properties.

2.5.2 Phase matrix moment expansions with generalized spherical functions

While the optical tables have the scattering matrix elements calculated as a function of the scattering angle, some applications need other formats. Most expansions are relatively easy for the user to calculate from the angle-dependent data given the abundance of such functions in most programming languages.

However, a particular expansion, called Generalized Spherical Function (GSF) expansion, is not widely available with e.g. standard Python libraries. Instead, separate, specialized software is often needed. At the same time the GSF format is required by the VLIDORT radiative transfer code. As such, conversion to the GSF expansion is provided as an optional postprocessing step for convenience.

For this purpose Michael Mishchenko's GSF expansion code [13] is used for the actual calculations via a Python wrapper. The wrapper adds a new multi-dimensional variable pmom to the existing optical tables without changing any of the existing variables and as such the GSF expansion can be considered fully backwards compatible.

The details of the expansion are not explained here; instead, the reader is directed to [13]. It should also be noted that the GSF expansion is similar to, but different from, the more common Legendre expansion. Care must be taken to not inadvertently mix the two.

3 Scientific design and options available

Sample output values and plots for current GEOS aerosol particles are shown in Appendix A. The same Appendix also includes the full input options used for each aerosol particle.

3.1 PSD options and multiple subbin distributions

For the particle size distribution (PSD) three keyword options are currently provided: `lognorm`, `ss` and `dust`, corresponding respectively to the lognormal distribution, the sea salt distribution, and the power law dust distribution.

3.1.1 Major size bins

Each PSD type can have multiple size bins. For example, one size bin could denote the fine mode of a given aerosol while another size bin could denote the coarse mode. The shape and refractive index assumptions are the same for the aerosols in the different size bins. In the GEOS model the aerosol particles in each size bin are transported independently.

3.1.2 Multi-distribution PSD

The user can define multiple size distributions for each size bin. The GEOS model does not see these distributions separately and will treat the size bin with multiple distributions the same as if it were a size bin described by a single size distribution.

Currently, the only GEOS particle that uses multiple subbin distributions is dust, where the first size bin is defined as a mixture of four non-overlapping subbin distributions. While this is different than e.g. a typical bimodal lognormal distribution, technically this special first bin is implemented as multiple distributions. `GEOSmie` supports traditional multimodal lognormal distributions via this same system, but none of the current GEOS aerosol particles require that. For more details on how multi-distribution size integration works see Section 2.3.1.

3.1.3 Particle size distribution types

`lognorm` chooses a lognormal PSD, i.e. a distribution where the logarithm of the size is normally distributed. Theoretically the lognormal PSD can span from arbitrarily small (non-zero) positive sizes to infinitely large, but for technical reasons the minimum and maximum radii have to be defined.

$$dndr(r) = \frac{1}{\sqrt{2\pi} r \log \sigma} exp(\frac{-\log(r/r_{mode})^2}{2\log(\sigma)^2}) \tag{6}$$

$$dndr(r > r_{max}) = 0 \tag{7}$$

$$dndr(r < r_{min}) = 0 \tag{8}$$

The PSD `ss` is a pre-defined PSD specific to sea salt, following [14]. The size distribution is defined as having hard cutoffs for the minimum and maximum size of each bin, and within that range the distribution is defined as:

$$dndr80 = 1.373 \cdot r_{80}^{-A}(1 + 0.057 \cdot r_{80}^{3.45}) \cdot 10^{(1.607 \cdot e^{-B^2})}, \tag{9}$$

where r_{80} is specified in microns and

$$A = 4.7 \cdot (1. + 30 \cdot r_{80})^{-0.017 \cdot r_{80}^{-1.44}} \tag{10}$$

and

$$B = \frac{0.433 - log_{10}(r_{80})}{0.433} \tag{11}$$

In addition,

$$r_{80} = r * ratio_{80}, \tag{12}$$

and

$$ratio_{80} = 1.65 \cdot \frac{r_{dry}}{r}, \tag{13}$$

where 1.65 is a constant value that is being used as an estimate for the ratio of the radius at 80% relative humidity and the dry radius. In other words, $r_{80} \approx 1.65 \cdot r_{dry}$, which can be derived from [15].

Finally,

$$dndr = dndr80 \cdot ratio_{80} \tag{14}$$

The PSD `du` is a PSD specific to dust particles. This PSD uses a hard-edged distribution, similar to the sea salt PSD. Within each bin the PSD is defined as:

$$dndr = r^{-4}. \tag{15}$$

The formulation in equation 15 comes from [16] and is based on the assumption that

$$dM/d(\log r) = constant, \tag{16}$$

from which equation 15 can be derived. Do note that the special treatment for the first size bin of GEOS dust is not implicit in the `du` distribution and is instead handled by manually defining the subbins (see Appendix A.9.2).

3.2 Hydration

`GEOSmie` currently offers three types of particle hydration schemes. The most common one, `gf`, uses precalculated growth factors (GFs) that are supplied in the particle definition files. For each relative humidity point requested, a GF has to be given. The main limitation of the static GFs is that the particle will grow by the same factor regardless of its size.

For hydrophobic particles, such as the current GEOS dust, a `trivial` hydration scheme is provided. This is simply a convenience scheme that calculates the optical properties once for the first relative humidity given, and copies all properties to the other humidity levels provided. The same effect could be achieved by setting GF equal to 1.0 for all relative humidities, but currently `GEOSmie` would still calculate the properties every time, even if the results would be identical. As such, `trivial` both simplifies the particle definition file as well as makes the calculations faster.

Finally, the third hydration scheme is called `ss` and is used for sea salt. A main functional difference to the static GF-based system is that the GFs in the `ss` system depend not only on the relative humidity, but also on the particle size (see Sec. 3.2.1). As such, `ss` behavior can not currently be mimicked by simply supplying `gf` values. It should also be noted that the `gf` system cannot be used to specify different growth factors for different size bins. This capacity may be added in a future `GEOSmie` update depending on user feedback.

Regardless of the scheme, for a given particle size and relative humidity, a GF is obtained. The GF influences both the particle radius (hydrated radius is called the wet radius as opposed to the unhydrated dry radius) and the refractive index.

The hydrated refractive index is calculated as follows

$$m = m_{water} + (m_{dry} - m_{water}) \cdot \frac{1}{gf^3} \tag{17}$$

where m_{water} is the refractive index of water, m_{dry} is the refractive index of the dry particle, and gf is the prescribed growth factor.

For all particles following the `simple` relative humidity dependence (or `trivial`, which is a special case of `simple`), the hydrated radius is calculated as follows:

$$r(rh) = r_{dry} \cdot gf(rh) \tag{18}$$

3.2.1 SS

Rather than use the `simple` relative humidity dependence, sea salt follows an empirical sea salt growth factor calculation by [15], which is denoted `ss` in the `GEOSmie` particle definitions.

$$r_{cm}(rh) = \sqrt[3]{\frac{c_1 * r_{dry,cm}^{c_2}}{c_3 \cdot r_{dry,cm}^{c_4} - log_{10}(rh)} + r_{dry,cm}^3} \tag{19}$$

and

$$r(rh) = \frac{r_{cm}(rh)}{100} \tag{20}$$

where $c_1 - c_4$ are defined in the particle definition file (see Section A.8.2 for the present sea salt values). Note that the equation expects the dry r to be in centimeters (as denoted by the subscript cm). Also note that given that the c_* parameters are exposed via the JSON interface, the `ss` hydration scheme can be used to replicate other particle growth schemes by setting the c_* values to the appropriate values from [15].

3.2.2 Hydrophobicness

There is a special flag `hydrophobic` that can be set in the particle definition file. As a post-processing step, the flag creates a new major size bin (bin 0), which is a copy of the original optical properties (now in major size bin 1) with the exception that only values from the first relative humidity point (typically 0%) are copied. Hence, this makes the new bin 0 a fully nonhydrated copy of the original optical properties.

This behavior is at present only used for carbonaceous aerosols, which have a single hydrophilic and hydrophobic mode, the latter of which is not subject to wet removal processes in the model. Technically, the behavior can be added to any particles that have a single size bin.

3.3 Refractive index files

For each aerosol particle (and optionally for each subbin distribution in a multi-distribution PSD, see Sect. 3.1.2) a file describing the spectral complex refractive indices needs to be supplied. This can be in the legacy format (not documented here), or a simple comma- or whitespace-separated file (recommended for any new refractive index files). The first column, wavelength, needs to be in units of micrometers. The second column, the real part of the refractive index, is defined as a positive value. The third column, the imaginary part of the refractive index, should be supplied as a negative number. The package accepts nonphysical (i.e. positive) refractive indices, in which case nonphysical optical properties will be produced. A warning will be included in a future version when positive refractive indices are used. Lines starting with a # are comments and ignored during the file read. An excerpt from a whitespace-separated refractive index file for GEOS nitrate particles is shown in Section 3.3.1 below.

The refractive index file supplied also implicitly defines the wavelengths at which the optical values are calculated. As such it needs to match the number of wavelengths desired as well as the wavelengths themselves. In case of a multi-distribution PSD different distributions can have different refractive indices, designated by providing a list of refractive index files rather than just a single one. In this case, each refractive index file needs to provide the refractive indices at the same wavelengths. None of the current GEOS particles use different refractive indices for different distributions but the capacity exists for research purposes and for providing future capabilities. At the moment there is no capability for different major size bins to have different refractive indices.

3.3.1 Example refractive index file for GEOS nitrate (first few wavelengths shown)

```
# lambda[um] m_real m_imaginary
0.250  1.588 -0.33E-07
0.300  1.573 -0.16E-07
0.350  1.566 -0.65E-08
0.400  1.561 -0.19E-08
0.450  1.559 -0.10E-08
```

```
0.500  1.557 -0.10E-08
0.550  1.556 -0.11E-08
```

4 User's guide

4.1 How to get and install the code

For up-to-date installation instructions refer to the README at `https://github.com/GEOS-ESM/GEOSmie/blob/feature/okemppin/first_commit/README.md` (to be updated to the main branch README after merge).

4.2 Particle definition files

`GEOSmie` is driven almost entirely by particle definition JSON files. Aside from adding entirely new capabilities (e.g. a new particle size distribution type) there should be little need for a user to edit any of the code files. For most users, editing the JSON file should be enough to add or customize aerosol particle types.

The default GEOS particle JSON files are included in the Appendix A. The appendix shows the structure and different options available, as well as plots of the optical properties that result in using these input parameters. All of the options are described below.

The JSON file structure uses nested structures both for objects ({}-enclosed blocks) as well as arrays ([]-enclosed lists) and care must be taken to use the proper nesting. There is currently only a very limited format validity checking in `GEOSmie` and as such if the format is incorrect, one will likely just have the program crash with a potentially confusing message. Using existing files as templates is highly encouraged.

The definition files have the following top-level keys:

- rhop0: Dry particle density
- rh: Relative humidity values
- rhDep: Particle hydration scheme type and parameters
- psd: Particle size distribution type and parameters
- ri: Refractive index file
- hydrophobic [OPTIONAL]: Create a special hydrophobic copy of the particle
- mode [OPTIONAL]: Optical calculation mode, if other than Mie
- kernel_params [OPTIONAL]: Kernel and shape distribution file paths

Some of these are simple while others are more complicated. Each is explained in a section below. Additionally, most of the parameters are mandatory with a few being optional. The optional ones are noted below.

4.2.1 rhop0

The dry density of the particle species. For particles with a single major size bin this value can be a single number. For particles with multiple size bins it must be an array. While particle density is not used for any intrinsic optical properties, the output tables contain values for mass efficiencies, for which the density value is required.

4.2.2 rh

Array of relative humidity values at which optical properties are to be calculated.

4.2.3 rhDep

This is a nested object with two levels. The first level uses only two keys: `type` and `params`. type determines the type of relative humidity dependence and will inform the package on how to calculate the hydrated radii and refractive indices. `params` is an object that defines the parameters for that given rhDep type.

- For `type` of `simple`, `params` is expected to have a single key, `gf`, with a value of an array that defines the individual growth factors at individual relative humidity points. The length of the `gf` array must match the length of the `rh` array defined above.
- For `type` of `trivial`, `params` is expected to have a single key, `gf`, which will be an array with a single element that specifies the growth factor across all RH values. The value is commonly set to 1.0 (relative humidity has no effect whatsoever) but in principle it could be set to any value.
- For `type` of `ss`, four keys are expected, with a single number for each. These are `c1`, `c2`, `c3`, `c4`. See Section 3.2.1 for more details on these parameters. It should be noted that since these parameters can be changed in the JSON file one can use the `ss` relative humidity dependence to replicate the behavior of the other particle types defined in [15].

4.2.4 psd

Like `rhDep`, `psd` is a nested structure. The top-level structure is the same as in `rhDep`, with `type` specifying the particle size distribution type, and `params` specifying the individual parameters used by that distribution.

- For `type` of `lognorm`, `params` is expected to have the following keys: `r0`, `rmin0`, `rmax0`, `sigma`, `numperdec` and `fracs`. All apart from `numperdec` need to be nested arrays, with the outer array denoting major size bins and the inner arrays denoting multiple distributions in that particular size bin. The length of `numperdec` must match the outer array lengths of the other parameters. That is, each distribution within a given major bin uses the same `numperdec` value.
  - `r0` is the lognormal distribution number median dry radius of the particle.
  - `rmin0` and `rmax0` are the minimum and maximum dry radii to which the particle size distribution is evaluated.
  - `sigma` is the standard deviation of the lognormal distribution.
  - `numperdec` refers to the number of individual size points per one decade of radius range at which the individual Mie simulations are run. That is, if the particle size distribution were to span a range from 10^{-9} to 10^{-6} (as defined by `rmin0` and `rmax0`) and `numperdec` were set to 500, then a total of $3 \cdot 500$ individual Mie simulations would be run to cover the size distribution. The procedure of how size integration is done is explained in Section 2.3.
  - `fracs` defines the number distribution fractions used for each of the subbin distributions for multimodal distributions. For the single-distribution size bins, used for most of the current GEOS particles, the inner arrays should simply have one element with the value of 1.0.
- For `type` of `ss`, `params` is expected to have the following keys: `rMinMaj`, `rMaxMaj`, `fracs` and `numperdec`.
  - `rMinMaj` and `rMaxMaj` are both arrays that list the minimum and maximum radii, respectively, for a given major bin.
  - The PSD type `ss` does not currently have multi-distribution capabilities so `fracs` should be set to an array of arrays with a single element, 1.0, as explained above for monomodal lognormal distributions.
  - Finally, `numperdec` is exactly the same as for the `lognorm` type.
- For `type` of `du`, the parameters are `rMinMaj`, `rMaxMaj` and `fracs`.

  - rMinMaj and rMaxMaj define the minimum and maximum radii for size bins as for ss, but crucially, for du both need to be nested arrays. This is largely due to GEOS dust making use of multi-distribution capabilities. As such, exactly like for multimodal lognorm PSD, the outer array refers to individual major size bins, and the inner array lists the minimum and maximum radii of the individual subdistributions.
  - Finally, fracs is similar to how it's used for the other PSD types. It is of note that here, as with the other two parameters, fracs is a nested array. Again, this deviates from ss due to du making use of subdistribution capabilities, which necessitates this structure.

4.2.5 ri

ri is an object that defines the refractive indices of the particle. It requires two key-value pairs: format expects one of three options: gads, csv, or wsv. gads refers to a legacy refractive index format that is not described here. Users are encouraged to look at existing examples in the data/ directory for examples. csv and wsv refer to comma-separated and whitespace-separated file formats. These two are equivalent apart from the separator type. Both csv and wsv expect a file with three columns: the wavelength in μm at which the refractive index is applicable, the real refractive index, and the imaginary refractive index (as a negative number). The list of wavelengths provided in this file also determines the wavelengths at which the optical properties are calculated.

The path parameter is an array of paths to the refractive index files. The array format allows multi-distribution size bins where each distribution has a different refractive index. To reiterate, the array should not be used to have different refractive indices for different major size bins. Rather, each major size bin is expected to have the same refractive indices. The array can only be used for mixing refractive indices in particles with multiple subbin distributions. This mixing exists for research purposes and is not currently used by any GEOS particle. For all current GEOS particles the path value should be an array with a single path in it.

4.2.6 hydrophobic [OPTIONAL]

hydrophobic is an optional parameter, defaulting to false. Setting it to true enacts a creation of a special hydrophobic size bin wherein the particle is assumed to not have any interactions with relative humidity (equivalent to growth factor 1.0 at all humidities). The special hydrophobic bin will be the size bin 0, with the original non-hydrophobic optical properties moved to the size bin 1. This behavior is only defined for particles with a single size bin.

4.2.7 mode [OPTIONAL]

mode is an optional parameter, defaulting to mie. Using mode mie means that all calculations are done with Mie simulations. The other valid option, kernel, means no simulations will be performed but instead values are interpolated from an external kernel file in lieu of calculating them. This is the only difference between the modes; integration, output format, and so on, function identically regardless of the mode.

4.2.8 kernel_params [OPTIONAL]

With mode kernel, kernel_params is mandatory. Otherwise, it has no effect.

The value for this key is a dictionary with two key-value pairs. An example can be seen in geosparticles/du.json. The two required values are:

- path is the path for the kernel NetCDF file. This can either be a file containing each individual kernel shape separately, or pre-integrated. In case the kernels are pre-integrated, GEOSmie still expects the ratio dimension, and a shape distribution file (with a single line for the only shape included) needs to be supplied. The file data/kernel_shape_dist/preintegrated.txt can be used for all preintegrated kernel files, that is, the ones where the ratio dimension size is equal to one.

- shape_dist is the path for the shape distribution to be used with the kernels. In other words, it is possible to use the same kernel file with multiple different shape distributions by using different shape_dist files.

No kernel files are included with GEOSmie. For users interested in using GRASP kernels, the raw kernels can be downloaded from the website https://www.grasp-open.com/products/spheroid-package-release/, and then converted to a GEOSmie-compatible format described in Appendix B using the included utility script runkernelconversion.py. Limited documentation is provided for the usage of the kernel conversion code, but basic usage can be seen in Section 4.3.4 and in the GitHub repository README file.

4.3 Command-line interface runscripts

The entirety of GEOSmie is intended to be run with command-line scripts. There are three, corresponding to three levels of processing.

- runoptics.py runs the package that generates the primary monochromatic optical tables that the majority of this document is describing. The existence of these monochromatic optical tables is a prerequisite for either of the other two scripts to be used.
- runbands.py averages the monochromatic optical properties over wavelength bands. These band-averaged tables are used in various radiation calculations.
- rungsf.py converts the scattering matrix elements of the regular monochromatic optical tables into a special Generalized Spherical Function base (see Section 2.5.2). This GSF expansion is required by VLIDORT radiative transfer code.
- runkernelconversion.py converts raw, text-based GRASP kernel files into a NetCDF file that can be used in GEOSmie.

Below all of the command-line parameters are explained for each of the three scripts.

4.3.1 Base optical tables: runoptics.py

Simple usage example:

```
python runoptics.py --name geosparticles/bc.json
```

All options:

```
Usage: runoptics.py

Options:
  --version            show program's version number and exit
  -h, --help           show this help message and exit
  --name=NAME          Particle file to use (default=)
  --datatype=DATATYPE  Particle data type to use ['json'] (default=json)
  --dest=DEST          Output directory to use (default=.)
```

4.3.2 Band-averaged optical tables: runbands.py

Simple usage example:

```
python runbands.py --filename integ-bc-raw.nc
```

All options:

```
Usage: runbands.py

Options:
  --version            show program’s version number and exit
  -h, --help           show this help message and exit
  --filename=FILENAME  Optical table file to use (default=)
  --partname=PARTNAME  Particle name to use in the qname variable of the
                       output table (default=)
  --dest=DEST          Output directory (default=.)
  --bandmode=BANDMODE  Band averaging type to use [’GEOS5’, ’RRTMG’, ’RRTMGP’,
                       ’PURDUE’] (default=RRTMG)
  --noIR=NOIR          Use the noIR option for GEOS5 band type (default=False)
  --useSolar=USESOLAR  Use the useSolar option (default=False)
```

4.3.3 GSF-expanded tables: rungsf.py

Simple usage example:

```
python rungsf.py --filename integ-bc-raw.nc
```

All options:

```
Usage: rungsf.py

Options:
  --version            show program’s version number and exit
  -h, --help           show this help message and exit
  --filename=FILENAME  Optical table file to use (default=)
  --dest=DEST          Output directory (default=.)
  --mode=MODE          Input file format (default=pygeos)
  --rhop=RHOP          Particle density (not needed/used for modes pygeos,
                       legendre) (default=1000.0)
```

4.3.4 GRASP-format kernel conversion: runkernelconversion.py

Simple usage example:

```
python runkernelconversion.py --filename data/kernelconversion/grasp.json
```

All options:

```
Usage: runkernelconversion.py

Options:
  --version            show program’s version number and exit
  -h, --help           show this help message and exit
  --filename=FILENAME  Kernel conversion JSON file path (default=)
  --dest=DEST          Output directory (default=.)
```

References

[1] Mian Chin, Paul Ginoux, Stefan Kinne, Omar Torres, Brent N Holben, Bryan N Duncan, Randall V Martin, Jennifer A Logan, Akiko Higurashi, and Teruyuki Nakajima. Tropospheric aerosol optical thickness from the gocart model and comparisons with satellite and sun photometer measurements. *Journal of the atmospheric sciences*, 59(3):461–483, 2002.

[2] Owen B Toon, RP Turco, D Westphal, R Malone, and M Liu. A multidimensional model for aerosols: Description of computational analogs. *Journal of Atmospheric Sciences*, 45(15):2123–2144, 1988.

[3] Peter R Colarco, Owen B Toon, Omar Torres, and Philip J Rasch. Determining the uv imaginary index of refraction of saharan dust particles from total ozone mapping spectrometer data using a three-dimensional model of dust transport. *Journal of Geophysical Research: Atmospheres*, 107(D16):AAC–4, 2002.

[4] X. Liu, R. C. Easter, S. J. Ghan, R. Zaveri, P. Rasch, X. Shi, J.-F. Lamarque, A. Gettelman, H. Morrison, F. Vitt, A. Conley, S. Park, R. Neale, C. Hannay, A. M. L. Ekman, P. Hess, N. Mahowald, W. Collins, M. J. Iacono, C. S. Bretherton, M. G. Flanner, and D. Mitchell. Toward a minimal representation of aerosols in climate models: description and evaluation in the community atmosphere model cam5. *Geoscientific Model Development*, 5(3):709–739, 2012.

[5] Jussi Leinonen. Python code for calculating mie scattering from single- and dual-layered spheres. Available at `https://github.com/jleinonen/pymiecoated/`, 2016.

[6] Christian Mätzler. Matlab functions for mie scattering and absorption, version 2. Available at `http://www.atmo.arizona.edu/students/courselinks/spring09/atmo656b/maetzler_mie_v2.pdf`, 2011.

[7] Craig F. Bohren and Donald R. Huffman. *Absorption and Scattering of Light by Small Particles.* John Wiley & Sons, New York, 1983. 530 pp.

[8] Warren J Wiscombe. Improved mie scattering algorithms. *Applied optics*, 19(9):1505–1509, 1980.

[9] George Arfken. *Mathematical Methods for Physicists.* Academic Press, Inc., San Diego, third edition, 1985.

[10] Oleg Dubovik, Alexander Sinyuk, Tatyana Lapyonok, Brent N Holben, Michael Mishchenko, Ping Yang, Tom F Eck, Hester Volten, Olga Munoz, Ben Veihelmann, et al. Application of spheroid models to account for aerosol particle nonsphericity in remote sensing of desert dust. *Journal of Geophysical Research: Atmospheres*, 111(D11), 2006.

[11] O Dubovik, BN Holben, T Lapyonok, A Sinyuk, MI Mishchenko, P Yang, and I Slutsker. Non-spherical aerosol retrieval method employing light scattering by spheroids. *Geophysical Research Letters*, 29(10):54–1, 2002.

[12] O Dubovik, M Herman, A Holdak, T Lapyonok, D Tanré, JL Deuzé, F Ducos, A Sinyuk, and A Lopatin. Statistically optimized inversion algorithm for enhanced retrieval of aerosol properties from spectral multi-angle polarimetric satellite observations. *Atmospheric Measurement Techniques*, 4(5):975–1018, 2011.

[13] Michael I Mishchenko, Igor V Geogdzhayev, and Ping Yang. Expansion of tabulated scattering matrices in generalized spherical functions. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 183:78–84, 2016.

[14] SL Gong. A parameterization of sea-salt aerosol source function for sub-and super-micron particles. *Global biogeochemical cycles*, 17(4), 2003.

[15] Hermann E Gerber. Relative-humidity parameterization of the navy aerosol model (nam). Technical report, Naval Research Lab Washington DC, 1985.

[16] Ina Tegen and Andrew A Lacis. Modeling of particle size distribution and its influence on the radiative properties of mineral dust aerosol. *Journal of Geophysical Research: Atmospheres*, 101(D14):19237–19244, 1996.

[17] Peter Colarco. personal communication.

[18] Peter Koepke, M Hess, I Schult, and EP Shettle. Global aerosol data set, 1997.

[19] Peter R Colarco, Santiago Gassó, Changwoo Ahn, Virginie Buchard, Arlindo M da Silva, and Omar Torres. Simulation of the ozone monitoring instrument aerosol index using the nasa goddard earth observing system aerosol reanalysis products. *Atmospheric Measurement Techniques*, 10(11):4121–4134, 2017.

[20] Melanie S Hammer, Randall V Martin, Aaron van Donkelaar, Virginie Buchard, Omar Torres, David A Ridley, and Robert JD Spurr. Interpreting the ultraviolet aerosol index observed with the omi satellite instrument to understand absorption by organic aerosols: implications for atmospheric oxidation and direct radiative effects. *Atmospheric Chemistry and Physics*, 16(4):2507–2523, 2016.

[21] Yan Feng and Joyce E Penner. Global modeling of nitrate and ammonium: Interaction of aerosols and tropospheric chemistry. *Journal of Geophysical Research: Atmospheres*, 112(D1), 2007.

[22] Huisheng Bian, Mian Chin, Didier A Hauglustaine, Michael Schulz, Gunnar Myhre, Susanne E Bauer, Marianne T Lund, Vlassis A Karydis, Tom L Kucsera, Xiaohua Pan, et al. Investigation of global particulate nitrate from the aerocom phase iii experiment. *Atmospheric Chemistry and Physics*, 17(21):12911–12940, 2017.

[23] SL Gong, LA Barrie, and J-P Blanchet. Modeling sea-salt aerosols in the atmosphere: 1. model development. *Journal of Geophysical Research: Atmospheres*, 102(D3):3805–3818, 1997.

[24] Peter Colarco, Arlindo da Silva, Mian Chin, and Thomas Diehl. Online simulations of global aerosol distributions in the nasa geos-4 model and comparisons to satellite and ground-based aerosol optical depth. *Journal of Geophysical Research: Atmospheres*, 115(D14), 2010.

[25] Paul Ginoux, Mian Chin, Ina Tegen, Joseph M Prospero, Brent Holben, Oleg Dubovik, and Shian-Jiann Lin. Sources and distributions of dust aerosols simulated with the gocart model. *Journal of Geophysical Research: Atmospheres*, 106(D17):20255–20273, 2001.

A Current GOCART particles

A.1 Summary of relevant parameters

The table below showcases the key input parameters of each particle type as well as a few of the optical values.

Table 1: Summary of a few key microphysical and optical properties of GEOS aerosol particle species. All values are provided at 0% RH unless otherwise noted. GF refers to the growth factor, BEXT refers to the mass extinction efficiency, and SSA refers to the single-scattering albedo.

Species	Bin	Density	Number median r	σ	Min r	Max r	Refreal	Refimag	GF	BEXT	SSA
		kg m-3	m		m	m				m2 kg-1	
							550nm	550nm	80%RH	550nm	550nm
bc	1	1000.0	1.18e-08	2.0	1e-10	3e-07	1.75	-0.44	1.19	9283.38	0.21
oc	1	1800.0	2.12e-08	2.2	1e-10	3e-07	1.53	-0.006	1.44	2667.44	0.96
brc	1	1800.0	2.12e-08	2.2	1e-10	3e-07	1.53	-0.006	1.44	2667.44	0.96
su	1	1700.0	6.95e-08	2.03	5e-09	3e-07	1.43	-1e-08	1.64	3143.54	1.00
ni	1	1725.0	4.45e-08	2.03	1e-09	1e-05	1.556	-1.1e-09	1.74	3829.37	1.00
ni	2	2200.0	6.00e-07	2.03	1e-09	0.0001	1.556	-1.1e-09	1.74	378.67	1.00
ni	3	2650.0	2.33e-06	2.0	1e-09	0.0001	1.556	-1.1e-09	1.74	77.83	1.00
ss	1	2200.0	N/A	N/A	3e-08	1e-07	1.5	-1e-08	1.57	726.79	1.00
ss	2	2200.0	N/A	N/A	1e-07	5e-07	1.5	-1e-08	1.61	3482.06	1.00
ss	3	2200.0	N/A	N/A	5e-07	1.5e-06	1.5	-1e-08	1.67	750.97	1.00
ss	4	2200.0	N/A	N/A	1.5e-06	5e-06	1.5	-1e-08	1.70	297.62	1.00
ss	5	2200.0	N/A	N/A	5e-06	1e-05	1.5	-1e-08	1.75	97.71	1.00
du	1	2500.0	N/A	N/A	1e-07	1e-06	1.53	-0.0055	1.00	1653.76	0.93
du	2	2650.0	N/A	N/A	1e-06	1.8e-06	1.53	-0.0055	1.00	540.50	0.87
du	3	2650.0	N/A	N/A	1.8e-06	3e-06	1.53	-0.0055	1.00	297.86	0.81
du	4	2650.0	N/A	N/A	3e-06	6e-06	1.53	-0.0055	1.00	168.11	0.74
du	5	2650.0	N/A	N/A	6e-06	1e-05	1.53	-0.0055	1.00	94.92	0.67

A.2 Combined plots of fine-mode particles

This section contains figures for all of the fine-mode particles: black carbon, organic carbon, brown carbon, sulfate, and nitrate. Many of the optical properties of organic carbon and brown carbon are identical, but given that some shown here do differ, the figures contain both even when the two overlap. All of the figures shown here have relative humidity as the x-axis and show the values at 550 nm wavelength. The particle-specific sections later in this appendix show additional figures for each particle with the wavelength as the x-axis.

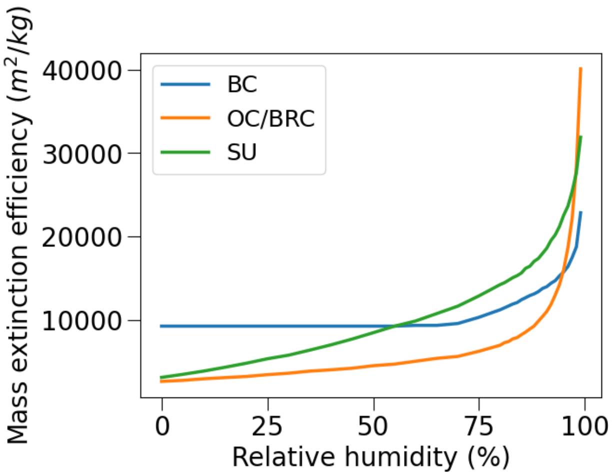


Figure 3: Fine-mode particles mass extinction coefficient at 550 nm

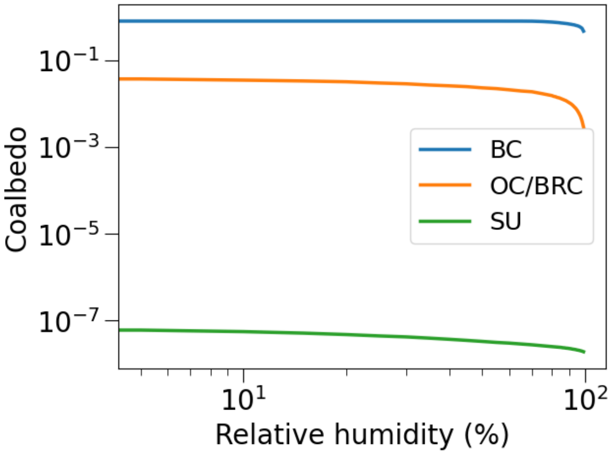


Figure 4: Fine-mode particles coalbedo at 550 nm

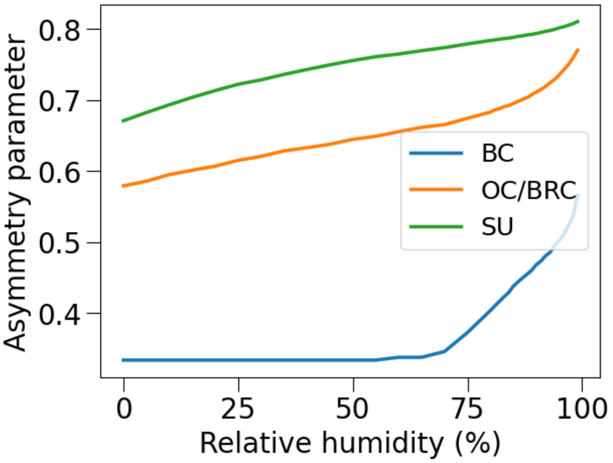


Figure 5: Fine-mode particles asymmetry factor at 550 nm

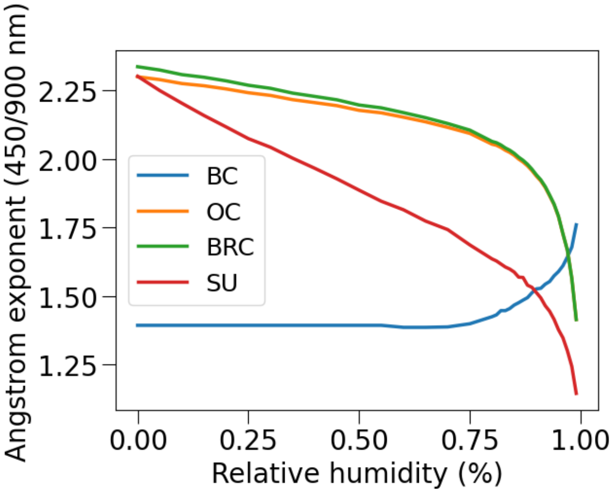


Figure 6: Fine-mode particles angstrom exponent (450/900 nm)

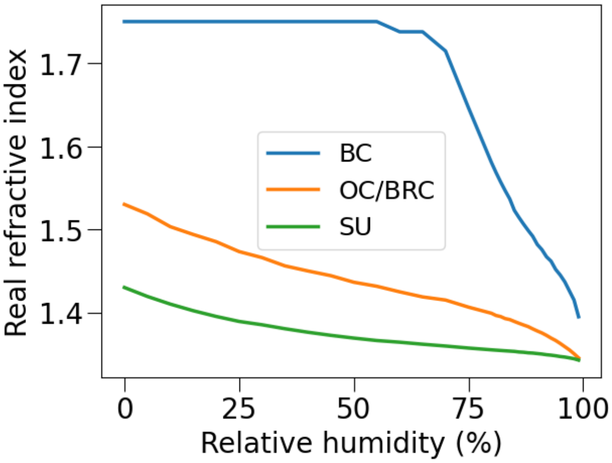


Figure 7: Fine-mode particles real refractive index at 550 nm

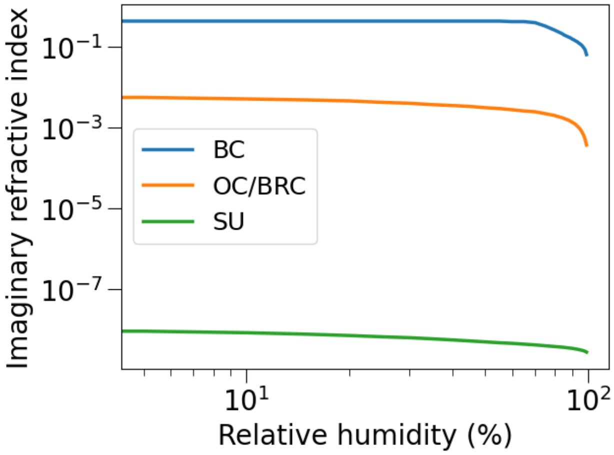


Figure 8: Fine-mode particles imaginary refractive index at 550 nm

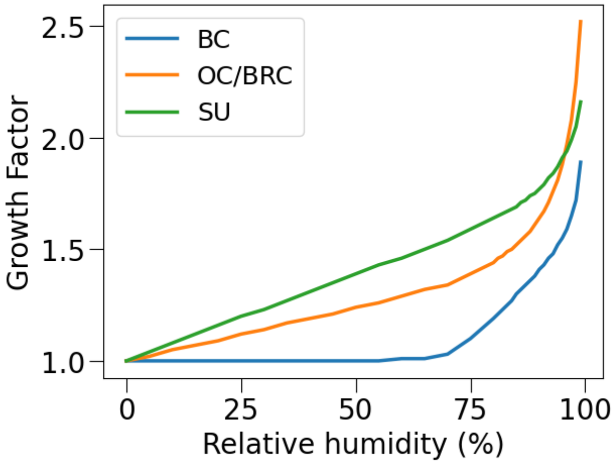


Figure 9: Fine-mode particles growth factor

A.3 Black Carbon

A.3.1 Microphysical parameter source

Parameter	Source	Notes
r0	[1]	
sigma	[1]	
rmax0	[1]	Source has a typo, it states 0.5µm while it should be 0.3 µm [17]
rmin0	This work	Chosen here to be sufficiently small
numperdec	This work	Chosen here to be sufficiently large
gf	[1]	
RI	[18]	
rho	[1]	

A.3.2 Particle definition JSON

```
{
"rhop0": 1000.0,

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
 0.97, 0.98, 0.99],

"rhDep": {"type": "simple", "params":{
  "gf": [1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00,
       1.00, 1.01, 1.01, 1.03, 1.10, 1.19, 1.21, 1.23, 1.25, 1.27, 1.30,
       1.32, 1.34, 1.36, 1.38, 1.41, 1.43, 1.46, 1.48, 1.52, 1.55, 1.59,
       1.65, 1.72, 1.89]}},

"psd": {"type": "lognorm", "params":{
  "r0": [[0.0118e-6]],
  "rmax0": [[0.3e-6]],
  "rmin0": [[1e-10]],
  "sigma": [[2.0]],
  "numperdec": [100],
  "fracs": [[1.0]]}},

"ri": {"format": "gads", "path": ["data/soot00"]},

"hydrophobic": true
}
```

A.3.3 Plots of optical properties

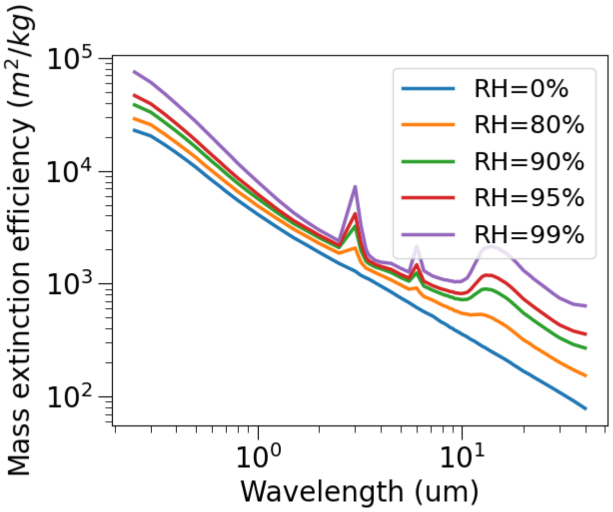


Figure 10: Black carbon mass extinction coefficient

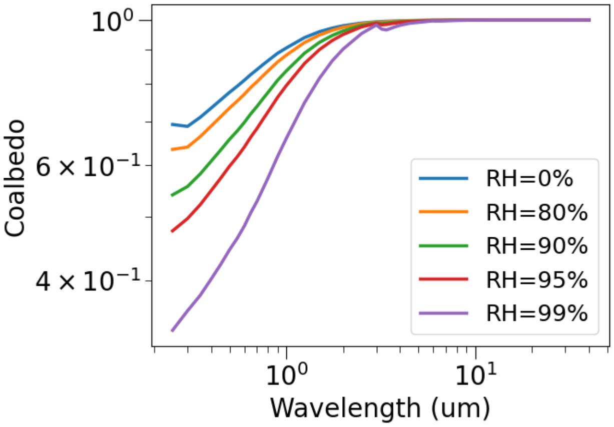


Figure 11: Black carbon coalbedo

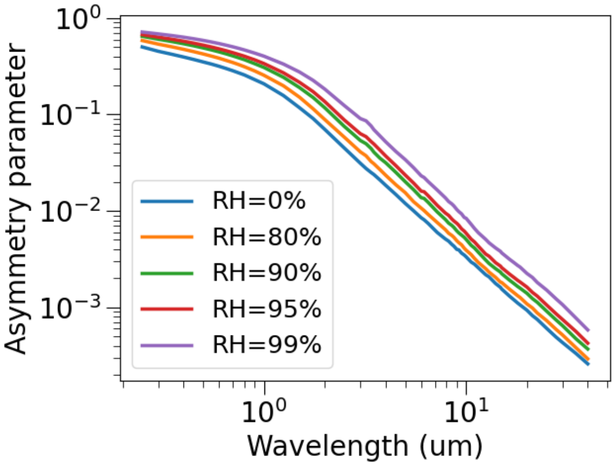


Figure 12: Black carbon asymmetry factor

A.4 Organic Carbon

A.4.1 Microphysical parameter source

Parameter	Source	Notes
r0	[1]	
sigma	[1]	
rmax0	[1]	See Sec. A.3
rmin0	This work	Chosen here to be sufficiently small
numperdec	This work	Chosen here to be sufficiently large
gf	[1]	
RI	[18]	
rho	[1]	

A.4.2 Particle definition JSON

```
{
"rhop0": 1800.0,

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
     0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
     0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
     0.97, 0.98, 0.99],

"rhDep": {"type": "simple", "params":{
"gf": [1.00, 1.02, 1.05, 1.07, 1.09, 1.12, 1.14, 1.17, 1.19, 1.21, 1.24,
       1.26, 1.29, 1.32, 1.34, 1.39, 1.44, 1.46, 1.47, 1.49, 1.50, 1.52,
       1.54, 1.56, 1.58, 1.61, 1.64, 1.67, 1.71, 1.76, 1.81, 1.88, 1.97,
       2.08, 2.25, 2.52]}},

"psd": {"type": "lognorm", "params":{
"r0":    [[0.0212e-6]],
"rmax0": [[0.3e-6]],
"rmin0": [[1e-10]],
"sigma": [[2.20]],
"numperdec": [800],
"fracs": [[1.0]]}},

"ri": {"format": "gads", "path": ["data/waso00"]},

"hydrophobic": true
}
```

A.4.3 Plots of optical properties

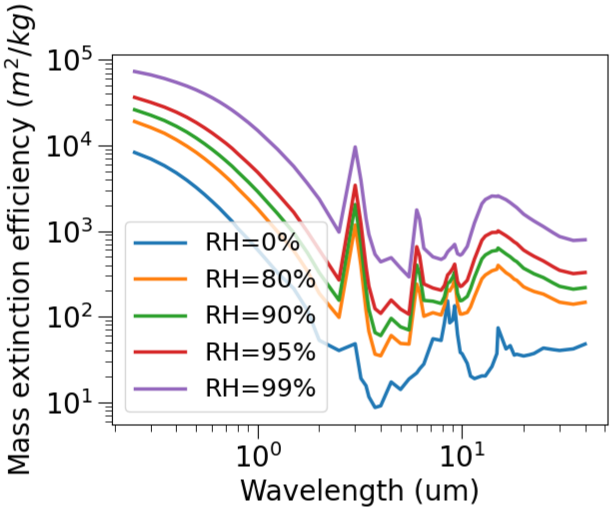


Figure 13: Organic carbon mass extinction coefficient

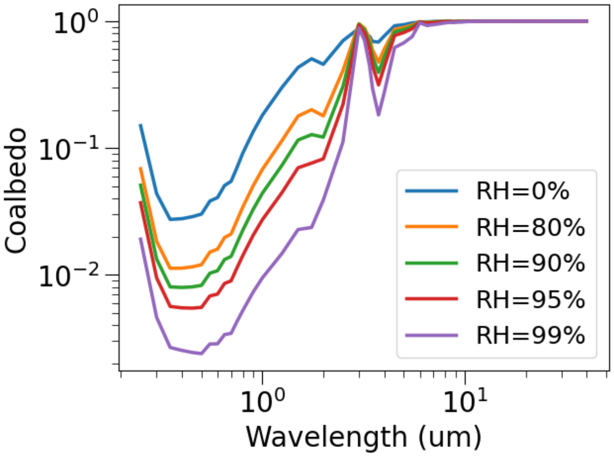


Figure 14: Organic carbon coalbedo

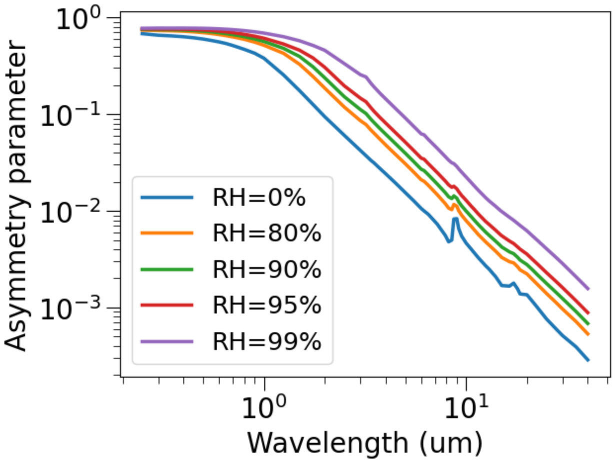


Figure 15: Organic carbon asymmetry factor

A.5 Brown Carbon

A.5.1 Microphysical parameter source

Parameter	Source	Notes
r0	[19, 20]	Same as Organic Carbon
sigma	[19, 20]	Same as Organic Carbon
rmax0	[19, 20]	Same as Organic Carbon
rmin0	This work	Chosen here to be sufficiently small
numperdec	This work	Chosen here to be sufficiently large
gf	[19, 20]	Same as Organic Carbon
RI	[19, 20]	
rho	[19, 20]	Same as Organic Carbon

A.5.2 Particle definition JSON

```
{
"rhop0": 1800.0,

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
     0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
     0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
     0.97, 0.98, 0.99],

"rhDep": {"type": "simple", "params":{
"gf": [1.00, 1.02, 1.05, 1.07, 1.09, 1.12, 1.14, 1.17, 1.19, 1.21, 1.24,
       1.26, 1.29, 1.32, 1.34, 1.39, 1.44, 1.46, 1.47, 1.49, 1.50, 1.52,
       1.54, 1.56, 1.58, 1.61, 1.64, 1.67, 1.71, 1.76, 1.81, 1.88, 1.97,
       2.08, 2.25, 2.52]}},

"psd": {"type": "lognorm", "params":{
"r0":    [[0.0212e-6]],
"rmax0": [[0.3e-6]],
"rmin0": [[1e-10]],
"sigma": [[2.20]],
"numperdec": [800],
"fracs": [[1.0]]}},

"ri": {"format": "wsv", "path": ["data/ri-brc.wsv"]},

"hydrophobic": true
}
```

A.5.3 Plots of optical properties

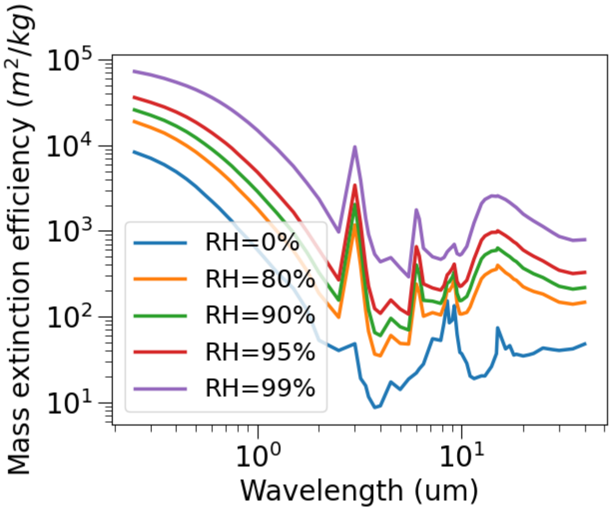


Figure 16: Brown carbon mass extinction coefficient

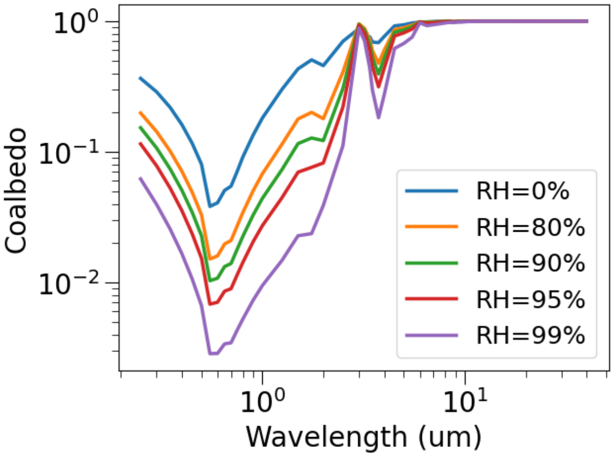


Figure 17: Brown carbon coalbedo

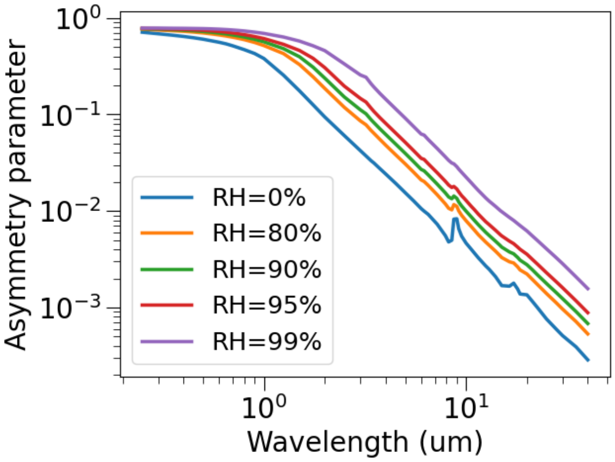


Figure 18: Brown carbon asymmetry factor

A.6 Sulfate

A.6.1 Microphysical parameter source

Parameter	Source	Notes
r0	[1]	
sigma	[1]	
rmax0	[1]	
rmin0	This work	Chosen here to be sufficiently small
numperdec	This work	Chosen here to be sufficiently large
gf	[1]	
RI	[18]	
rho	[1]	

A.6.2 Particle definition JSON

```
{

"rhop0": 1700.0,

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
 0.97, 0.98, 0.99],

"rhDep": {"type": "simple", "params":{
"gf": [1.00, 1.04, 1.08, 1.12, 1.16, 1.20, 1.23, 1.27, 1.31, 1.35, 1.39,
       1.43, 1.46, 1.50, 1.54, 1.59, 1.64, 1.65, 1.66, 1.67, 1.68, 1.69,
       1.71, 1.72, 1.74, 1.75, 1.77, 1.79, 1.82, 1.84, 1.87, 1.91, 1.94,
       1.99, 2.05, 2.16]}},

"psd": {"type": "lognorm", "params":{
"r0"   : [[0.0695e-6]],
"rmin0": [[0.005e-6]],
"rmax0": [[0.3e-6]],
"sigma": [[2.03]],
"numperdec": [1000],
"fracs": [[1.0]]}},

"ri": {"format": "gads", "path": ["data/suso00"]}
}
```

A.6.3 Plots of optical properties

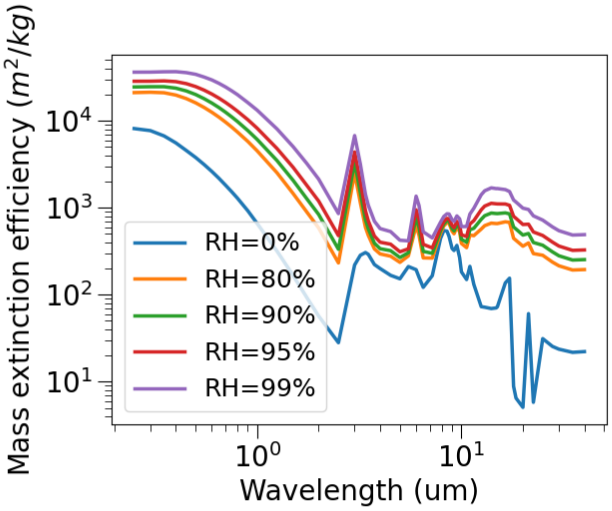


Figure 19: Sulfate mass extinction coefficient

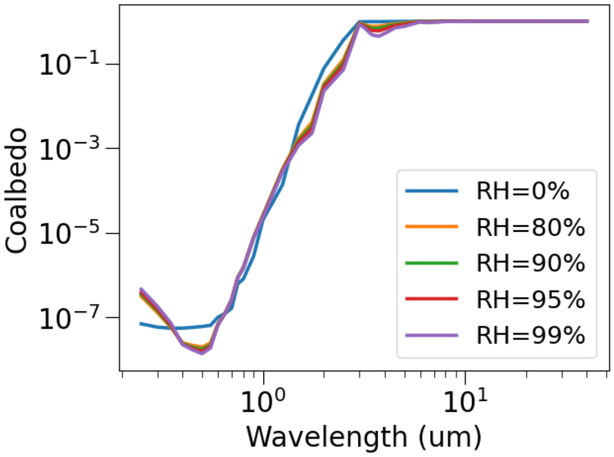


Figure 20: Sulfate coalbedo

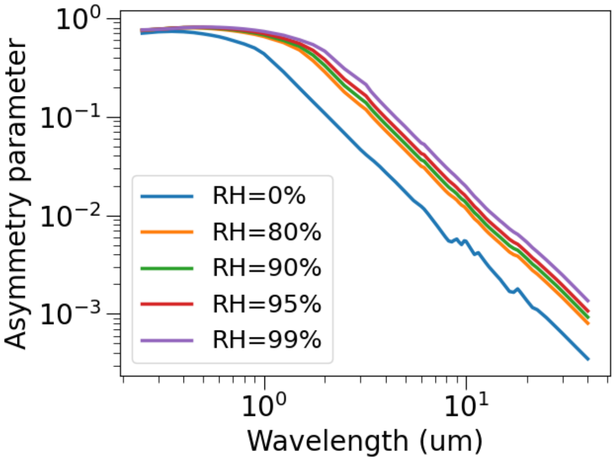


Figure 21: Sulfate asymmetry factor

A.7 Nitrate

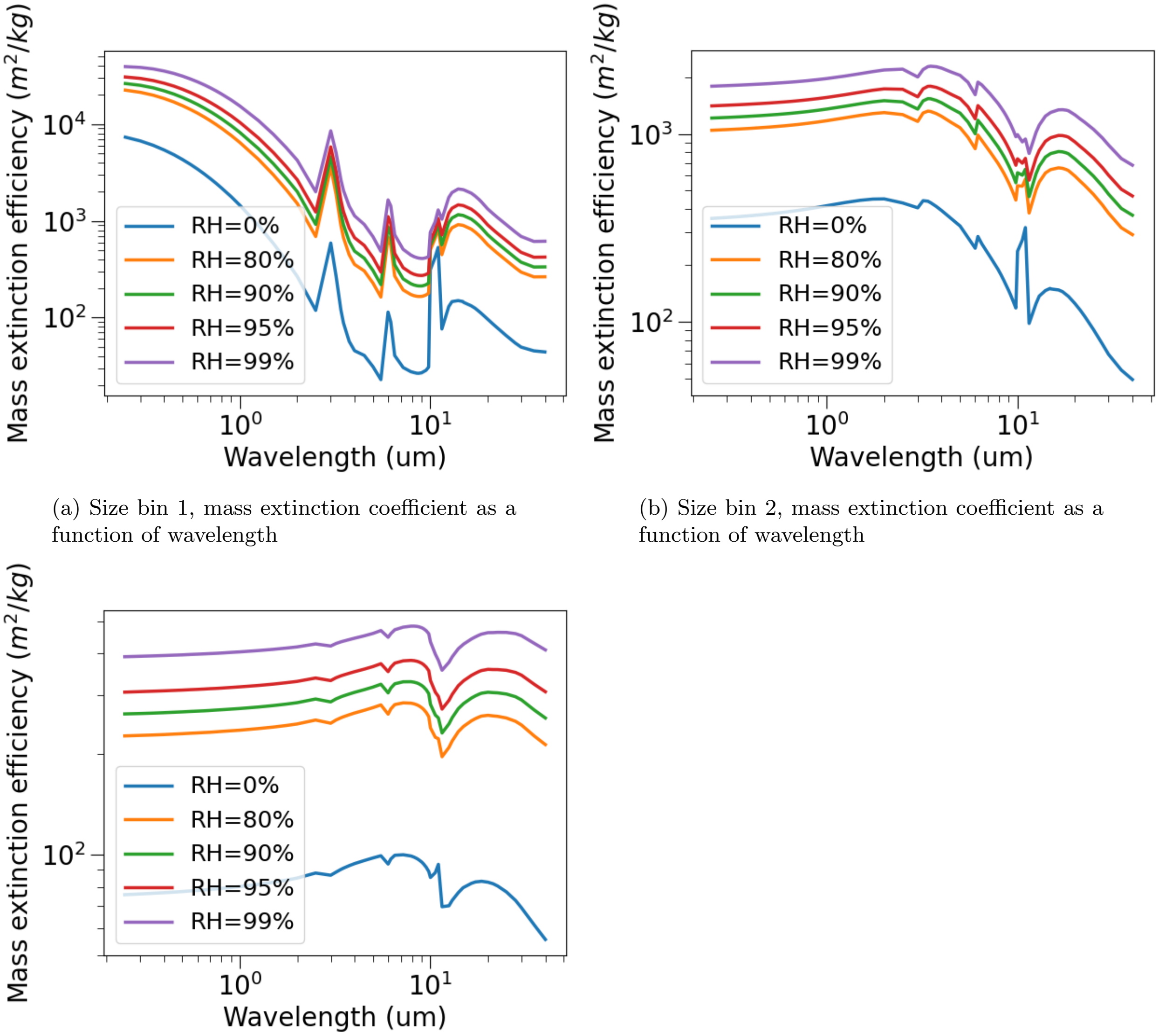
A.7.1 Microphysical parameter source

Parameter	Source	Notes
r0	[21, 22]	
sigma	[21, 22]	
rmax0	[21, 22]	
rmin0	This work	Chosen here to be sufficiently small
numperdec	This work	Chosen here to be sufficiently large
gf		
RI	[21, 22]	
rho	[21, 22]	

A.7.2 Particle definition JSON

```
{
"rhop0": [1725.0, 2200.0, 2650.0],

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
     0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
     0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
     0.97, 0.98, 0.99],

"rhDep": {"type": "simple", "params":{
"gf": [1.0  , 1.1024, 1.1448, 1.1872, 1.2296, 1.272 , 1.3038, 1.3462,
       1.3886, 1.431 , 1.4734, 1.5158, 1.5476, 1.59  , 1.6324, 1.6854,
       1.7384, 1.749 , 1.7596, 1.7702, 1.7808, 1.7914, 1.8126, 1.8232,
       1.8444, 1.855 , 1.8762, 1.8974, 1.9292, 1.9504, 1.9822, 2.0246,
       2.0564, 2.1094, 2.173 , 2.2896]}},

"psd": {"type": "lognorm", "params":{
"r0": [[4.45480907e-08], [5.99685837e-07], [2.33161301e-06]],
"rmin0": [[1e-9], [1e-9], [1e-9]],
"rmax0": [[10.0e-6], [1e-4], [1.0e-4]],
"sigma": [[2.03], [2.03], [2.0]],
"numperdec": [800, 800, 800],
"fracs": [[1.0], [1.0], [1.0]]}},

"ri": {"format": "wsv", "path": ["data/ri-nitrate.wsv"]}
}
```

A.7.3 Plots of optical properties



(a) Size bin 1, mass extinction coefficient as a function of wavelength

(b) Size bin 2, mass extinction coefficient as a function of wavelength

(c) Size bin 3, mass extinction coefficient as a function of wavelength

Figure 22: Fine-mode nitrate mass extinction coefficient

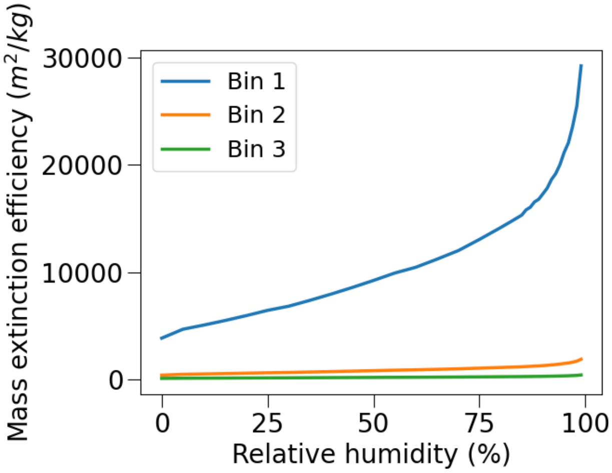
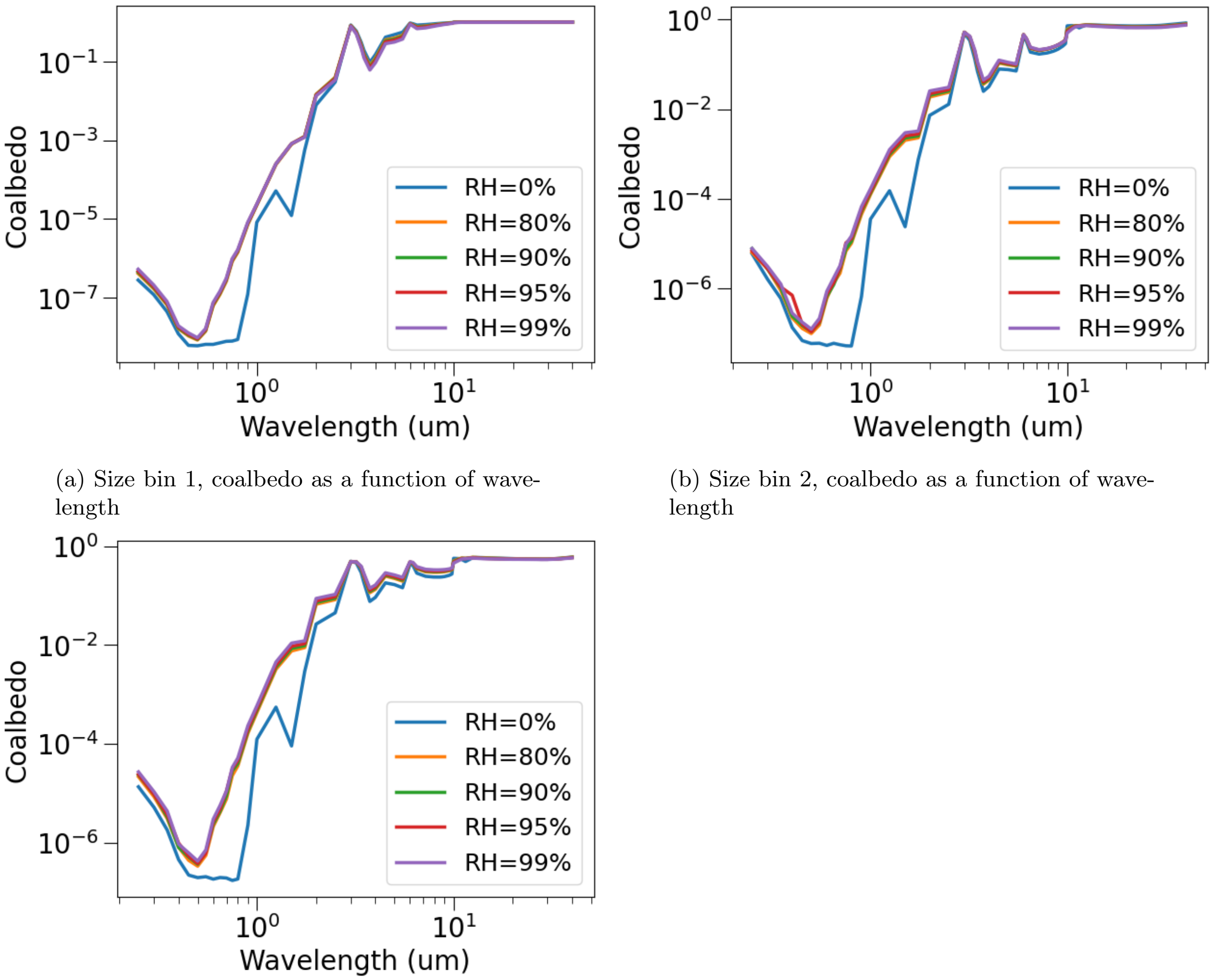


Figure 23: Fine-mode nitrate mass extinction coefficient at 550 nm



(a) Size bin 1, coalbedo as a function of wavelength

(b) Size bin 2, coalbedo as a function of wavelength

(c) Size bin 3, coalbedo as a function of wavelength

Figure 24: Fine-mode nitrate coalbedo

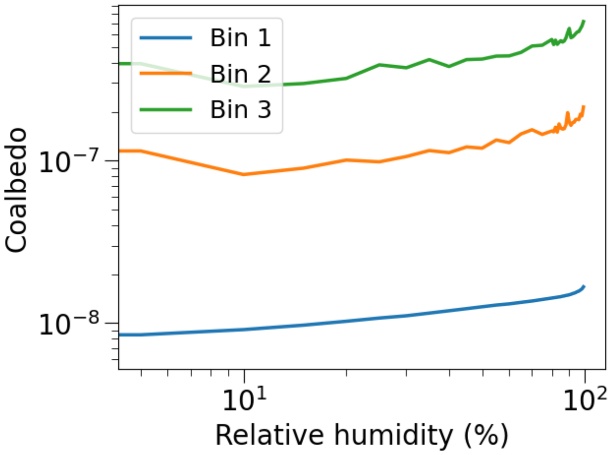
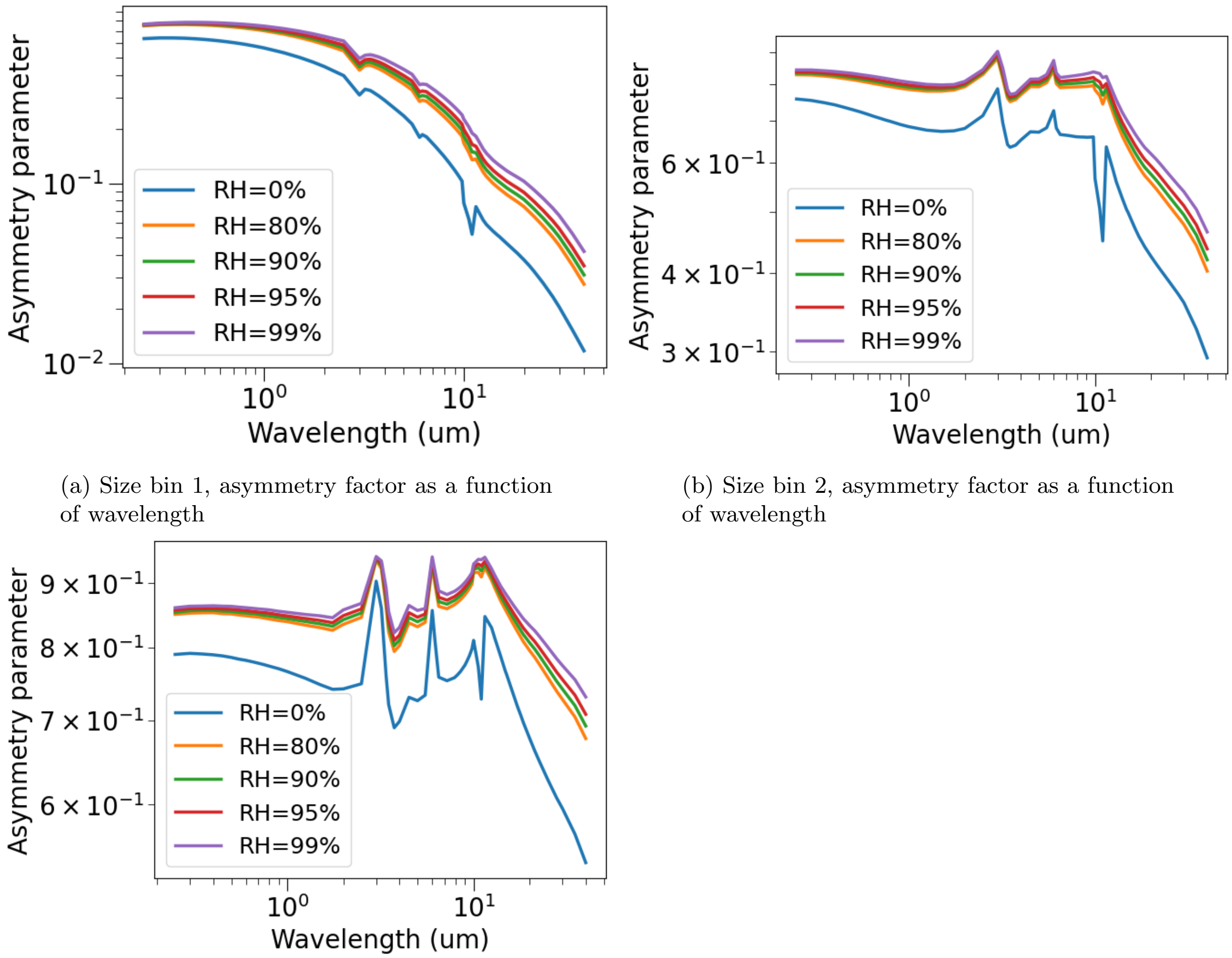


Figure 25: Fine-mode nitrate coalbedo at 550 nm



(a) Size bin 1, asymmetry factor as a function of wavelength

(b) Size bin 2, asymmetry factor as a function of wavelength

(c) Size bin 3, asymmetry factor as a function of wavelength

Figure 26: Fine-mode nitrate asymmetry factor

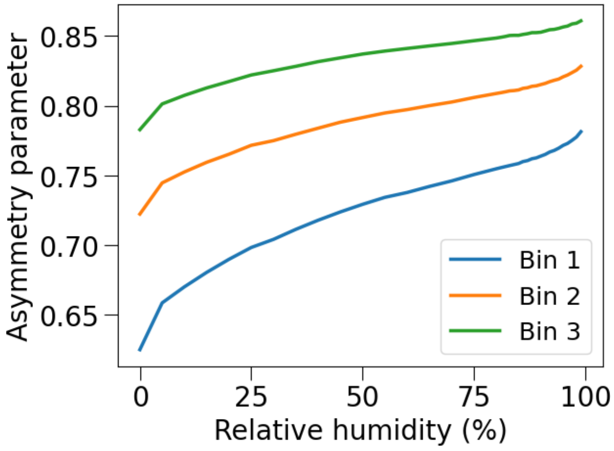


Figure 27: Fine-mode nitrate asymmetry factor at 550 nm

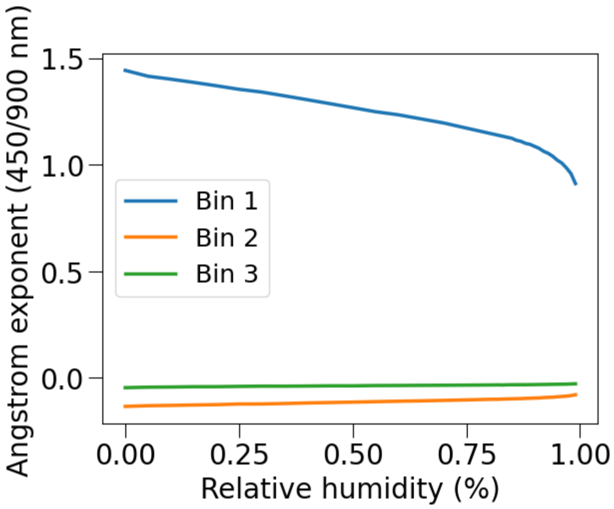


Figure 28: Fine-mode nitrate angstrom exponent (450/900 nm)

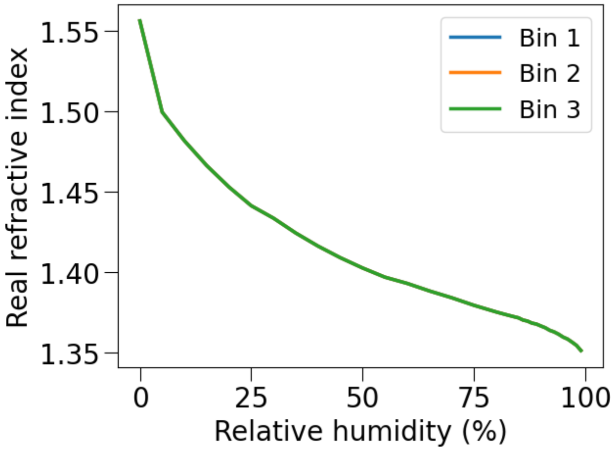


Figure 29: Fine-mode nitrate real refractive index at 550 nm

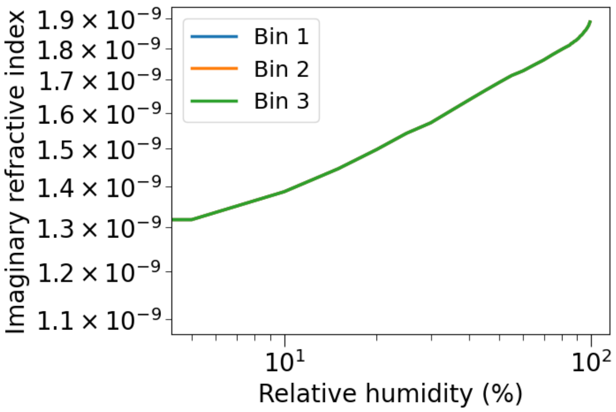


Figure 30: Fine-mode nitrate imaginary refractive index at 550 nm

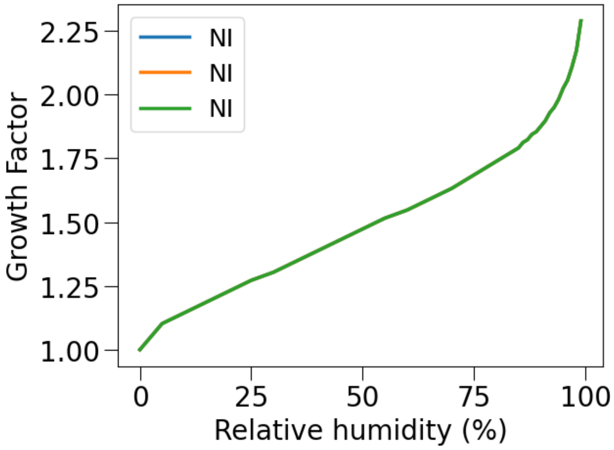


Figure 31: Fine-mode nitrate growth factor

A.8 Sea salt

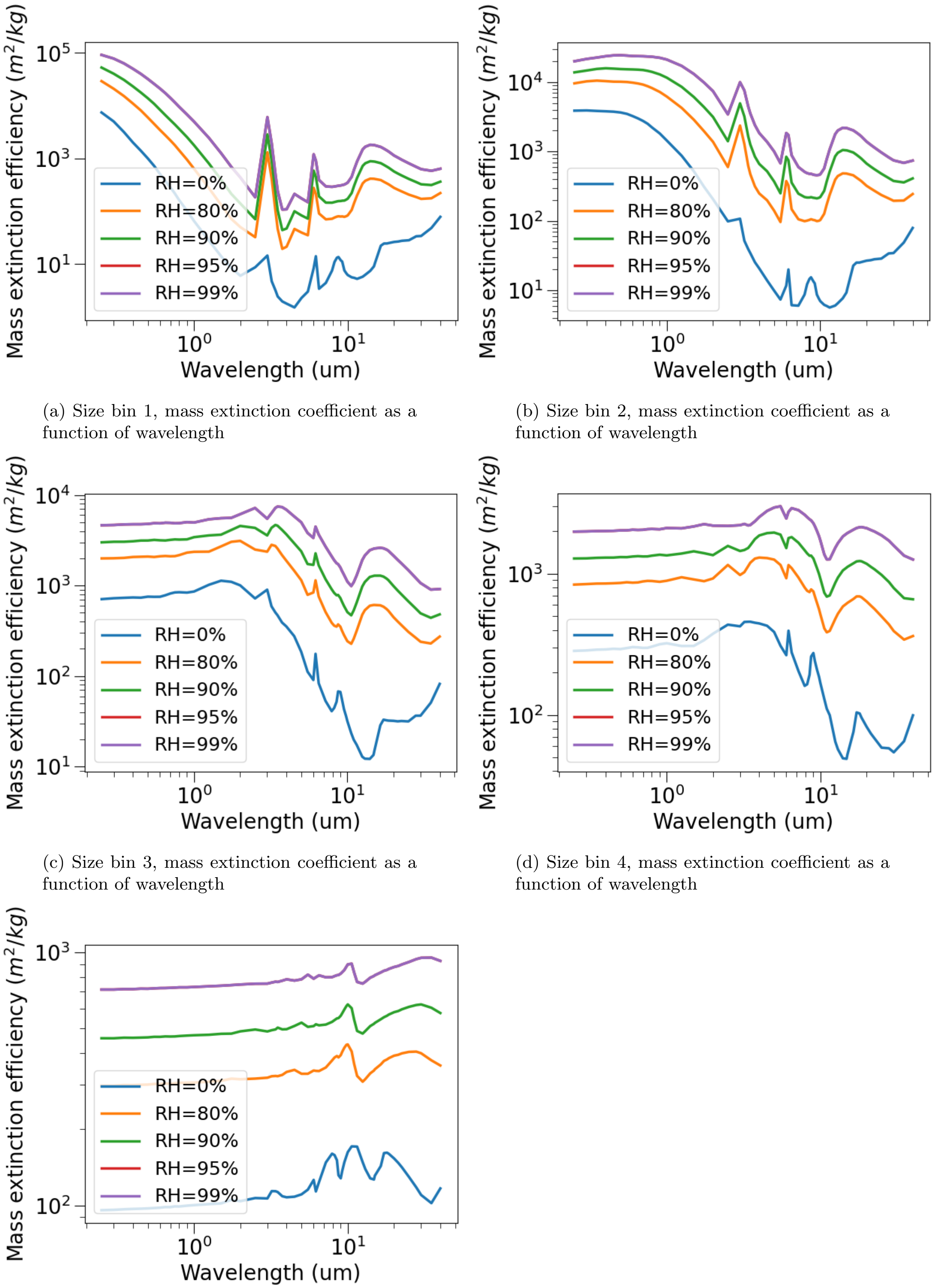
A.8.1 Microphysical parameter source

Parameter	Source	Notes
PSD	[14]	
numperdec	This work	Chosen here to be sufficiently large
Humidity growth (c1–c4)	[23]	
RI	[18]	
rho	[1]	

A.8.2 Particle definition JSON

```
{
"rhop0": 2200.0,

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
 0.97, 0.98, 0.99],

"maxrh": 0.95,

"rhDep": {"type": "ss", "params":{
    "c1": 0.7674,
    "c2": 3.079,
    "c3": 2.573e-11,
    "c4": -1.424}},

"psd": {"type": "ss", "params":{
    "rMinMaj": [0.03e-6,0.1e-6,0.5e-6,1.5e-6,5.0e-6],
    "rMaxMaj": [0.1e-6,0.5e-6,1.5e-6,5.0e-6,10.0e-6],
    "fracs": [[1.0], [1.0], [1.0], [1.0], [1.0]],
    "numperdec": [1600, 1600, 1600, 1600, 1600]}},

"ri": {"format": "gads", "path": ["data/sscm00"]}
}
```

A.8.3 Plots of optical properties



(a) Size bin 1, mass extinction coefficient as a function of wavelength

(b) Size bin 2, mass extinction coefficient as a function of wavelength

(c) Size bin 3, mass extinction coefficient as a function of wavelength

(d) Size bin 4, mass extinction coefficient as a function of wavelength

(e) Size bin 5, mass extinction coefficient as a function of wavelength

Figure 32: Sea salt mass extinction coefficient

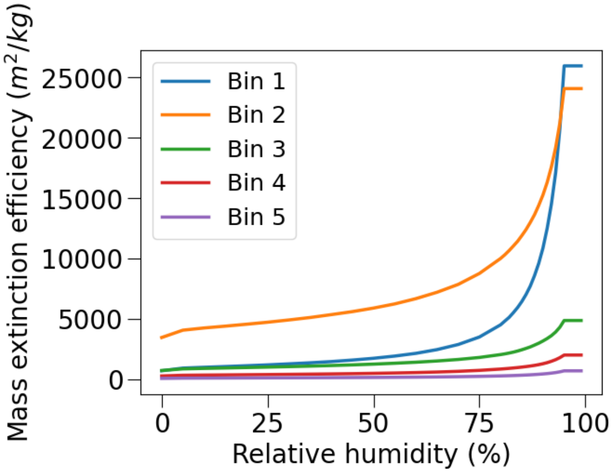
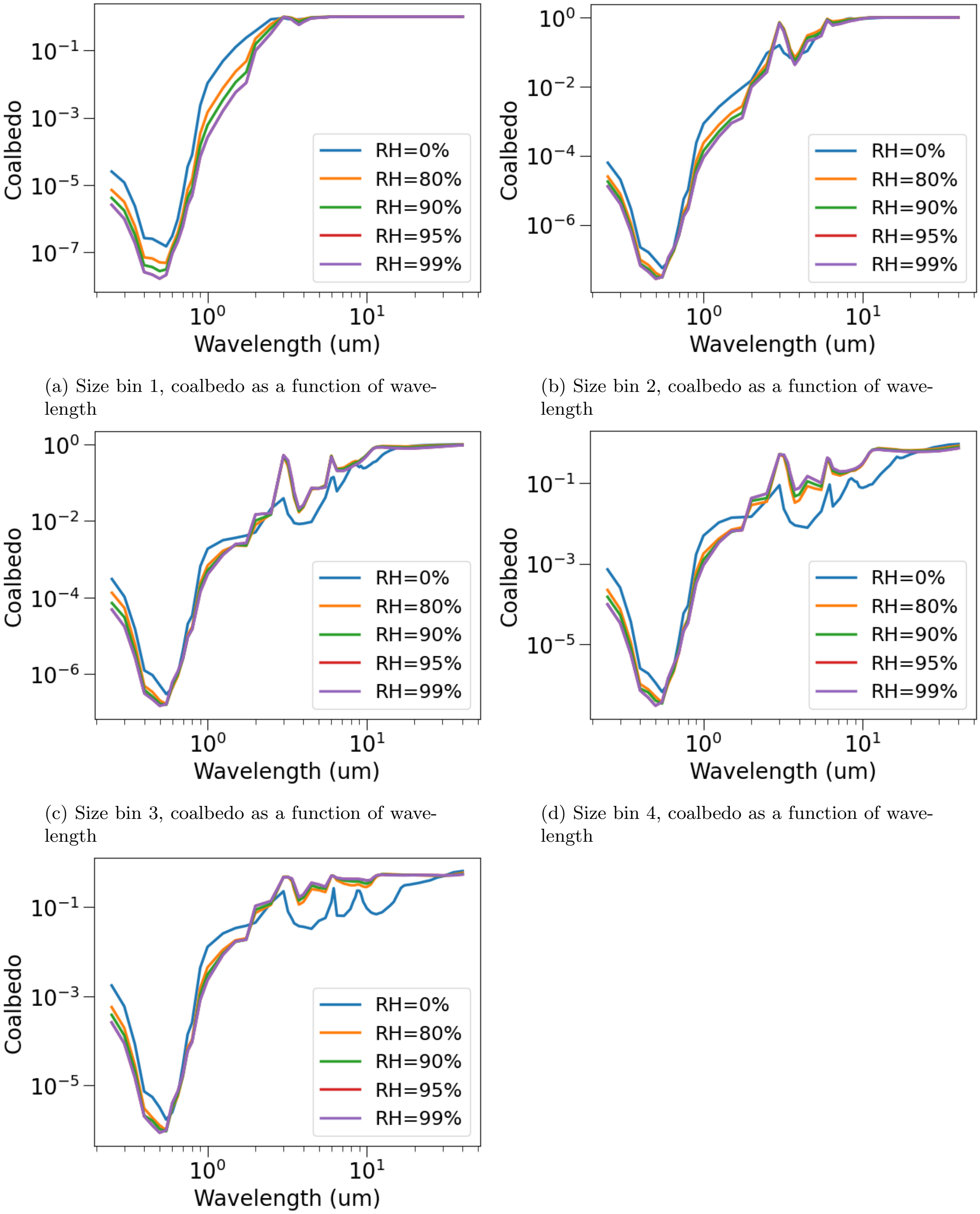


Figure 33: Sea salt mass extinction coefficient at 550 nm



(a) Size bin 1, coalbedo as a function of wavelength

(b) Size bin 2, coalbedo as a function of wavelength

(c) Size bin 3, coalbedo as a function of wavelength

(d) Size bin 4, coalbedo as a function of wavelength

(e) Size bin 5, coalbedo as a function of wavelength

Figure 34: Sea salt coalbedo

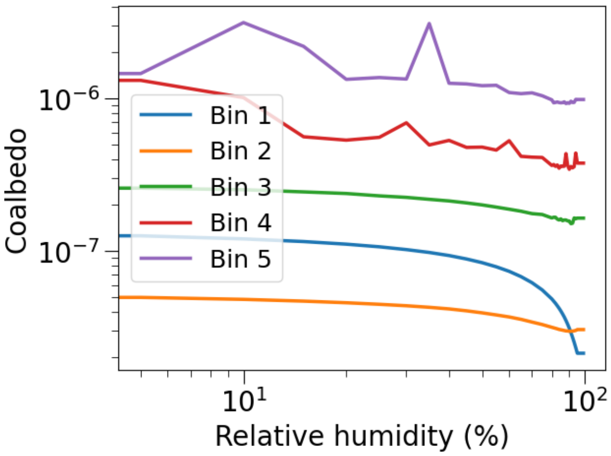
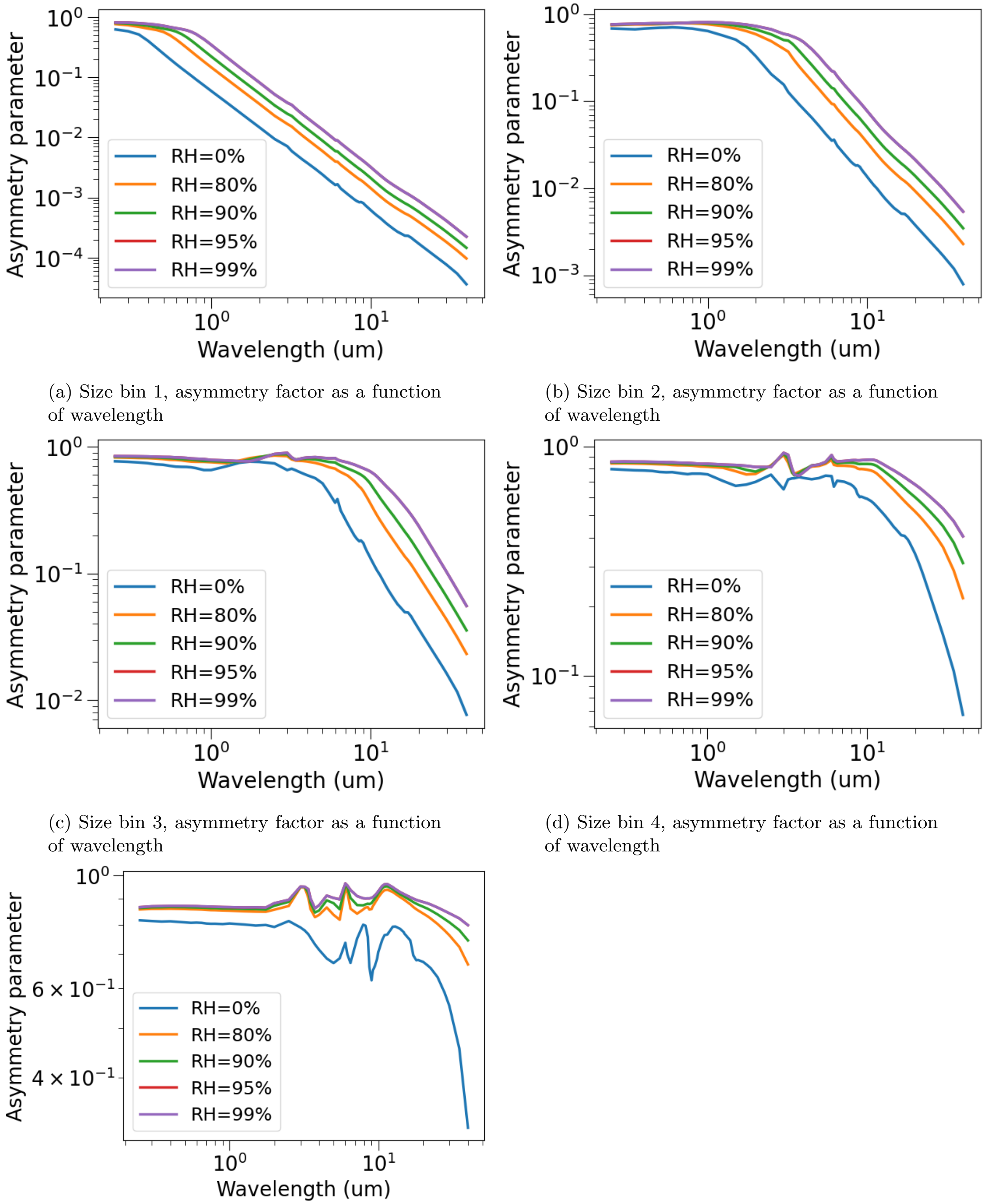


Figure 35: Sea salt coalbedo at 550 nm



(a) Size bin 1, asymmetry factor as a function of wavelength

(b) Size bin 2, asymmetry factor as a function of wavelength

(c) Size bin 3, asymmetry factor as a function of wavelength

(d) Size bin 4, asymmetry factor as a function of wavelength

(e) Size bin 5, asymmetry factor as a function of wavelength

Figure 36: Sea salt asymmetry factor

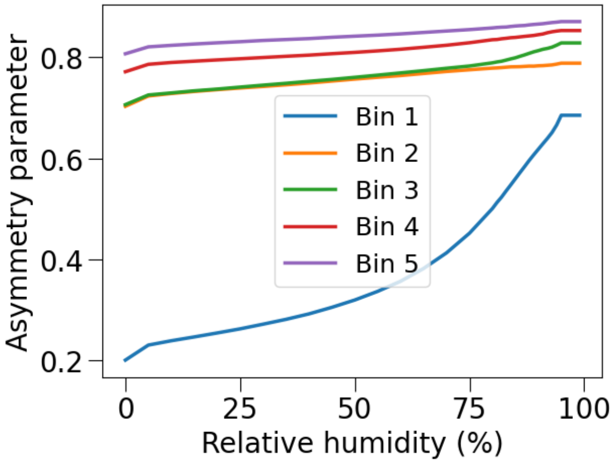


Figure 37: Sea salt asymmetry factor at 550 nm

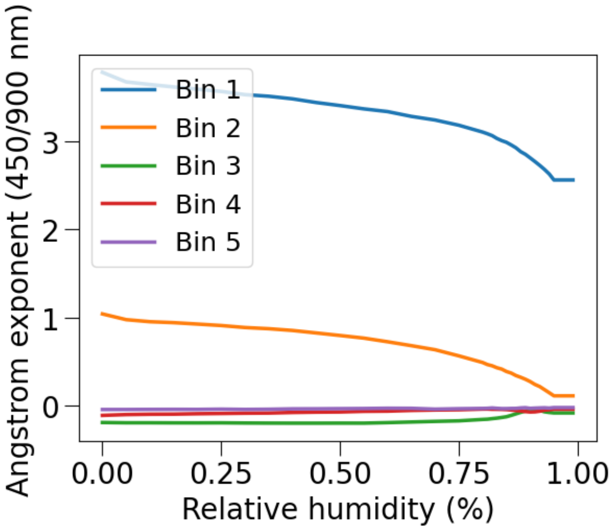


Figure 38: Sea salt angstrom exponent (450/900 nm)

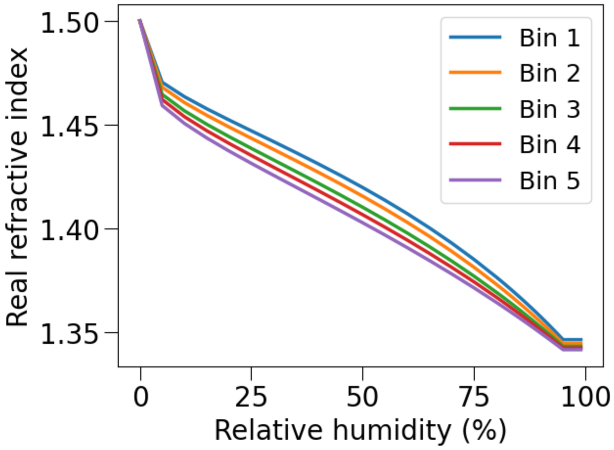


Figure 39: Sea salt real refractive index at 550 nm

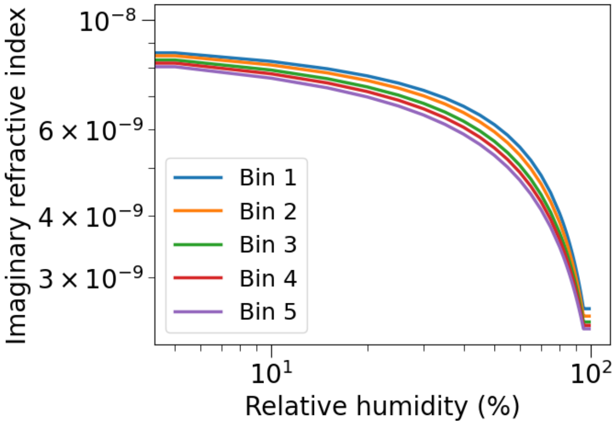


Figure 40: Sea salt imaginary refractive index at 550 nm

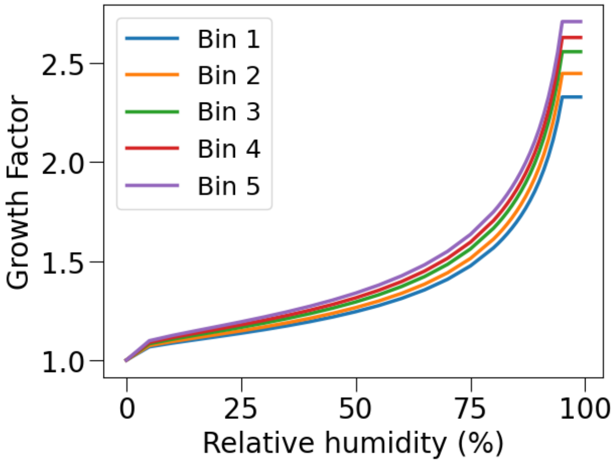


Figure 41: Sea salt growth factor

A.9 Dust

A.9.1 Microphysical parameter source

Parameter	Source	Notes
PSD	[24]	Assume dV/(d ln r) is constant
Bin edges	[24]	
fracs	[24]	
RI	[18]	
rho	[25]	
Spheroid kernels	[10]	

A.9.2 Particle definition JSON

```
{
"rhop0": [2500.0, 2650.0, 2650.0, 2650.0, 2650.0],

"rh": [0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50,
 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.81, 0.82, 0.83, 0.84, 0.85,
 0.86, 0.87, 0.88, 0.89, 0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96,
 0.97, 0.98, 0.99],

"rhDep": {"type": "trivial", "params":{
  "gf": [1.0]}},

"psd": {"type": "du", "params":{
      "rMinMaj": [[0.1e-6, 0.18e-6, 0.3e-6, 0.6e-6], [1.0e-6], [1.8e-6], [3.0e-6], [6.0e-6]],
      "rMaxMaj": [[0.18e-6, 0.3e-6, 0.6e-6, 1.0e-6], [1.8e-6], [3.0e-6], [6.0e-6], [10.0e-6]],
      "fracs": [[0.009, 0.081, 0.234, 0.676], [1.0], [1.0], [1.0], [1.0]]}},

"ri": {"format": "gads", "path": ["data/miam00"]},

"mode": "kernel",
"kernel_params": {
  "path": "/path/to/kernel",
  "shape_dist": "data/kernel_shape_dist/spheroid_fixed.txt"
  }
}
```

A.9.3 Plots of optical properties

Dust is defined as not changing its optical properties in any way as a function of relative humidity. As such, figures which have relative humidity as the x-axis have been omitted for dust.

The figures shown below illustrate the capability of `GEOSmie` to use external kernels. The data shown below has not yet been implemented in GEOS. We continue to develop the dust capabilities.

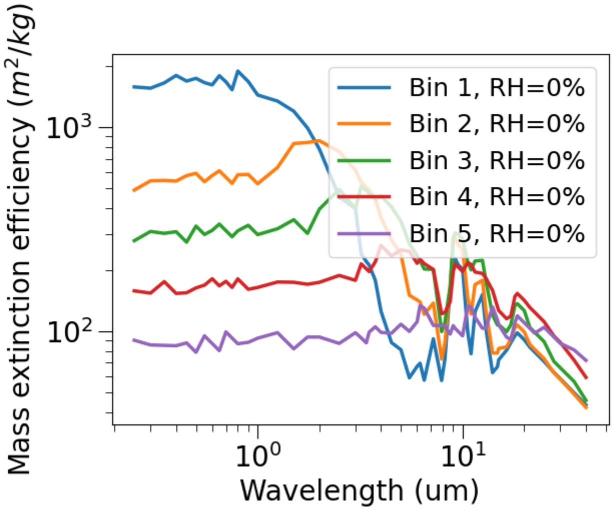


Figure 42: Dust mass extinction coefficient

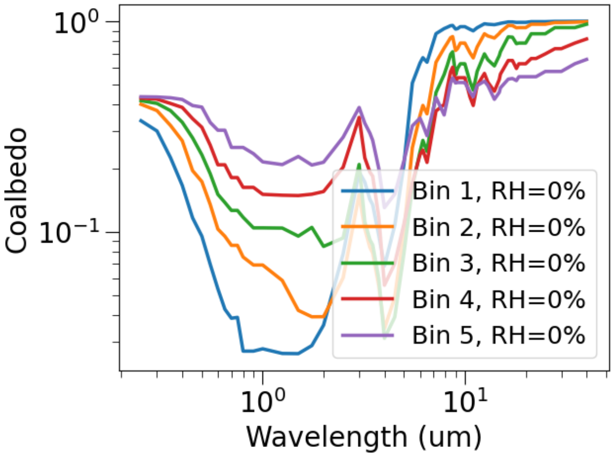


Figure 43: Dust coalbedo

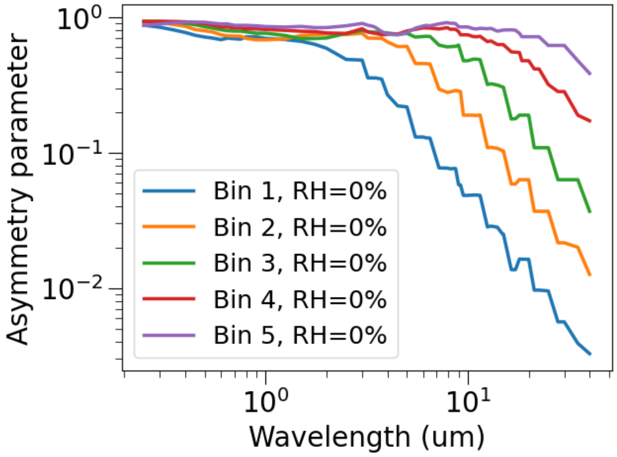


Figure 44: Dust asymmetry factor

B Pre-calculated kernel format

B.1 Overview

Below is a description of the current spheroid kernel format. The dimension `ratio` is the kernel identifier. For spheroidal particles it refers to the axis ratio, but `GEOSmie` does not use the values of these ratios in calculations. Rather, it is only used as dimension for individual spheroidal kernels within the kernel collection file, and will be used as an index in the shape integration step. As such, any non-spheroidal kernel should still use the `ratio` dimension and understand it is merely a shape identifier.

The precise dimension lengths shown below for any of the dimensions other than `scattering_element` can vary depending on the kernel file and do not need to be the same as in the spheroid kernel example below. `GEOSmie` will simply use the values provided in its calculations.

For the shape distribution any header lines that begin with `#` are ignored. Further, only the first column (titled `fraction` in the example below, but again, the header is optional) is needed and used. For example, the example below shows the aspect ratios for each of the individual shapes in the distribution, but this exists only for user convenience. All columns after the first are simply ignored.

Using a particular kernel database does not limit one to use a particular shape distribution. For example, with spheroidal kernels, the default GRASP shape distribution is provided, but the user can freely supply another shape distribution by creating a new shape distribution file and editing the corresponding file path in the JSON file.

B.2 Example: GEOSmie-compatible GRASP spheroidal kernel file

```
netcdf kernel_file {
dimensions:
        ratio = 25 ;
        mr = 22 ;
        mi = 16 ;
        x = 41 ;
        angle = 181 ;
        scattering_element = 6 ;
variables:
        double ratio(ratio) ;
        double mr(mr) ;
        double mi(mi) ;
        double x(x) ;
        double angle(angle) ;
        double ext(ratio, mr, mi, x) ;
        double abs(ratio, mr, mi, x) ;
        double sca(ratio, mr, mi, x) ;
        double qext(ratio, mr, mi, x) ;
        double qabs(ratio, mr, mi, x) ;
        double qsca(ratio, mr, mi, x) ;
        double qb(ratio, mr, mi, x) ;
        double g(ratio, mr, mi, x) ;
        double scama(ratio, mr, mi, x, scattering_element, angle) ;
}
```

B.3 Example: Shape distribution file for GRASP kernels (`data/spheroid_fixed.txt`)

```
# fraction     aspect_ratio
0.6618500E-01  0.3349000E+00
0.6502500E-01  0.3669000E+00
0.6363500E-01  0.4019000E+00
0.6205000E-01  0.4403000E+00
0.5872000E-01  0.4823000E+00
0.5335000E-01  0.5283000E+00
0.4776250E-01  0.5787000E+00
0.4295300E-01  0.6339000E+00
0.4032050E-01  0.6944000E+00
0.0000000E+00  0.7607000E+00
0.0000000E+00  0.8333000E+00
0.0000000E+00  0.9129000E+00
0.0000000E+00  0.1000000E+01
0.0000000E+00  0.1095400E+01
0.0000000E+00  0.1200000E+01
0.0000000E+00  0.1314500E+01
0.4032050E-01  0.1440000E+01
0.4295300E-01  0.1577400E+01
0.4776250E-01  0.1728000E+01
0.5335000E-01  0.1892900E+01
0.5872000E-01  0.2073600E+01
0.6205000E-01  0.2271500E+01
0.6363500E-01  0.2488320E+01
0.6502500E-01  0.2725800E+01
0.6618500E-01  0.2986000E+01
```

C NetCDF4 file stucture

```
netcdf integ-bc-raw {
dimensions:
radius = 2 ;
rh = 36 ;
lambda = 61 ;
ang = 371 ;
nPol = 6 ;
variables:
double radius(radius) ;
radius:long_name = "dry particle effective radius" ;
radius:units = "m" ;
double rh(rh) ;
rh:long_name = "relative humidity" ;
rh:units = "fraction" ;
double lambda(lambda) ;
lambda:long_name = "wavelength" ;
lambda:units = "m" ;
double ang(ang) ;
ang:long_name = "scattering angle" ;
ang:units = "degrees" ;
double nPol(nPol) ;
nPol:long_name = "Scattering matrix element index, ordered as P11, P12, P33, P34, P22, P44" ;
nPol:units = "dimensionless" ;
double s11(radius, rh, lambda, ang) ;
s11:long_name = "S11 element of the scattering matrix" ;
s11:units = "dimensionless" ;
double s12(radius, rh, lambda, ang) ;
s12:long_name = "S12 element of the scattering matrix" ;
s12:units = "dimensionless" ;
double s22(radius, rh, lambda, ang) ;
s22:long_name = "S22 element of the scattering matrix" ;
s22:units = "dimensionless" ;
double s33(radius, rh, lambda, ang) ;
s33:long_name = "S33 element of the scattering matrix" ;
s33:units = "dimensionless" ;
double s34(radius, rh, lambda, ang) ;
s34:long_name = "S34 element of the scattering matrix" ;
s34:units = "dimensionless" ;
double s44(radius, rh, lambda, ang) ;
s44:long_name = "S44 element of the scattering matrix" ;
s44:units = "dimensionless" ;
double qsca(radius, rh, lambda) ;
qsca:long_name = "scattering efficiency" ;
qsca:units = "dimensionless" ;
double qabs(radius, rh, lambda) ;
qabs:long_name = "absorption efficiency" ;
qabs:units = "dimensionless" ;
double qext(radius, rh, lambda) ;
qext:long_name = "extinction efficiency" ;
qext:units = "dimensionless" ;
double g(radius, rh, lambda) ;
g:long_name = "asymmetry factor" ;
```

```
g:units = "dimensionless" ;
double ssa(radius, rh, lambda) ;
ssa:long_name = "single-scattering albedo" ;
ssa:units = "dimensionless" ;
double qb(radius, rh, lambda) ;
qb:long_name = "backscattering efficiency" ;
qb:units = "dimensionless" ;
double bsca(radius, rh, lambda) ;
bsca:long_name = "mass scattering efficiency" ;
bsca:units = "m2 (kg dry mass)-1" ;
double bext(radius, rh, lambda) ;
bext:long_name = "mass extinction efficiency" ;
bext:units = "m2 (kg dry mass)-1" ;
double bbck(radius, rh, lambda) ;
bbck:long_name = "mass backscatter efficiency" ;
bbck:units = "m2 (kg dry mass)-1" ;
double mass(radius, rh) ;
mass:long_name = "particle mass" ;
mass:units = "kg" ;
double volume(radius, rh) ;
volume:long_name = "particle volume per kg dry mass" ;
volume:units = "m3 kg-1" ;
double area(radius, rh) ;
area:long_name = "particle cross sectional area per kg dry mass" ;
area:units = "m2 kg-1" ;
double rEff(radius, rh) ;
rEff:long_name = "effective radius of bin" ;
rEff:units = "m" ;
double rMass(radius, rh) ;
rMass:long_name = "effective mass of wet particle" ;
rMass:units = "kg" ;
double rUp(radius, rh) ;
rUp:long_name = "upper edge of radius bin" ;
rUp:units = "m" ;
double rLow(radius, rh) ;
rLow:long_name = "lower edge of radius bin" ;
rLow:units = "m" ;
double pback(nPol, radius, rh, lambda) ;
pback:long_name = "phase function in backscatter direction, ordered as P11, P12, P33, P34, P22, P44" ;
pback:units = "dimensionless" ;
double rhop(radius, rh) ;
rhop:long_name = "wet particle density" ;
rhop:units = "kg m-3" ;
double growth_factor(radius, rh) ;
growth_factor:long_name = "growth factor = ratio of wet to dry particle radius" ;
growth_factor:units = "fraction" ;
double refreal(radius, rh, lambda) ;
refreal:long_name = "real refractive index of wet particle" ;
refreal:units = "dimensionless" ;
double refimag(radius, rh, lambda) ;
refimag:long_name = "imaginary refractive index of wet particle" ;
refimag:units = "dimensionless" ;
data:
 radius = 1.18e-08, _ ;

 rh = 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6,
    0.65, 0.7, 0.75, 0.8, 0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88,
    0.89, 0.9, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99 ;

 lambda = 2.5e-07, 3e-07, 3.5e-07, 4e-07, 4.5e-07, 5e-07, 5.5e-07, 6e-07,
    6.5e-07, 7e-07, 7.5e-07, 8e-07, 9e-07, 1e-06, 1.25e-06, 1.5e-06,
    1.75e-06, 2e-06, 2.5e-06, 3e-06, 3.2e-06, 3.39e-06, 3.5e-06, 3.75e-06,
    4e-06, 4.5e-06, 5e-06, 5.5e-06, 6e-06, 6.2e-06, 6.5e-06, 7.2e-06,
    7.9e-06, 8.2e-06, 8.5e-06, 8.7e-06, 9e-06, 9.2e-06, 9.5e-06, 9.8e-06,
    1e-05, 1.06e-05, 1.1e-05, 1.15e-05, 1.25e-05, 1.3e-05, 1.4e-05, 1.48e-05,
    1.5e-05, 1.64e-05, 1.72e-05, 1.8e-05, 1.85e-05, 2e-05, 2.13e-05,
    2.25e-05, 2.5e-05, 2.79e-05, 3e-05, 3.5e-05, 4e-05 ;

 ang = 0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11,
    0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2, 0.21, 0.22, 0.23,
    0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3, 0.31, 0.32, 0.33, 0.34, 0.35,
    0.36, 0.37, 0.38, 0.39, 0.4, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47,
    0.48, 0.49, 0.5, 0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59,
    0.6, 0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7, 0.71,
    0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8, 0.81, 0.82, 0.83,
    0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9, 0.91, 0.92, 0.93, 0.94, 0.95,
    0.96, 0.97, 0.98, 0.99, 1, 1.09, 1.18, 1.27, 1.36, 1.45, 1.54, 1.63,
    1.72, 1.81, 1.9, 1.99, 2.08, 2.17, 2.26, 2.35, 2.44, 2.53, 2.62, 2.71,
    2.8, 2.89, 2.98, 3.07, 3.16, 3.25, 3.34, 3.43, 3.52, 3.61, 3.7, 3.79,
    3.88, 3.97, 4.06, 4.15, 4.24, 4.33, 4.42, 4.51, 4.6, 4.69, 4.78, 4.87,
    4.96, 5.05, 5.14, 5.23, 5.32, 5.41, 5.5, 5.59, 5.68, 5.77, 5.86, 5.95,
    6.04, 6.13, 6.22, 6.31, 6.4, 6.49, 6.58, 6.67, 6.76, 6.85, 6.94, 7.03,
    7.12, 7.21, 7.3, 7.39, 7.48, 7.57, 7.66, 7.75, 7.84, 7.93, 8.02, 8.11,
    8.2, 8.29, 8.38, 8.47, 8.56, 8.65, 8.74, 8.83, 8.92, 9.01, 9.1, 9.19,
    9.28, 9.37, 9.46, 9.55, 9.64, 9.73, 9.82, 9.91, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
    34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
    52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
    70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87,
    88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104,
    105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,
    119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
    133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146,
    147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160,
    161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174,
    175, 176, 177, 178, 179, 180 ;

 nPol = 0, 1, 2, 3, 4, 5 ;
}
```

Previous Volumes in This Series

Volume 1
September 1994

Documentation of the Goddard Earth Observing System (GEOS) general circulation model - Version 1

L.L. Takacs, A. Molod, and T. Wang

Volume 2
October 1994

Direct solution of the implicit formulation of fourth order horizontal diffusion for gridpoint models on the sphere

Y. Li, S. Moorthi, and J.R. Bates

Volume 3
December 1994

An efficient thermal infrared radiation parameterization for use in general circulation models

M.-D. Chou and M.J. Suarez

Volume 4
January 1995

Documentation of the Goddard Earth Observing System (GEOS) Data Assimilation System - Version 1

James Pfaendtner, Stephen Bloom, David Lamich, Michael Seablom, Meta Sienkiewicz, James Stobie, and Arlindo da Silva

Volume 5
April 1995

Documentation of the Aries-GEOS dynamical core: Version 2

Max J. Suarez and Lawrence L. Takacs

Volume 6
April 1995

A Multiyear Assimilation with the GEOS-1 System: Overview and Results

Siegfried Schubert, Chung-Kyu Park, Chung-Yu Wu, Wayne Higgins, Yelena Kondratyeva, Andrea Molod, Lawrence Takacs, Michael Seablom, and Richard Rood

Volume 7
September 1995

Proceedings of the Workshop on the GEOS-1 Five-Year Assimilation

Siegfried D. Schubert and Richard B. Rood

Volume 8
March 1996

Documentation of the Tangent Linear Model and Its Adjoint of the Adiabatic Version of the NASA GEOS-1 C-Grid GCM: Version 5.2

Weiyu Yang and I. Michael Navon

Volume 9	*Energy and Water Balance Calculations in the Mosaic LSM*
March 1996	Randal D. Koster and Max J. Suarez
Volume 10	*Dynamical Aspects of Climate Simulations Using the GEOS General Circulation Model*
April 1996	Lawrence L. Takacs and Max J. Suarez
Volume 11	*Documentation of the Tangent Linear and Adjoint Models of the Relaxed Arakawa-Schubert Moisture Parameterization Package of the NASA GEOS-1 GCM (Version 5.2)*
May 1997	Weiyu Yang, I. Michael Navon, and Ricardo Todling
Volume 12	*Comparison of Satellite Global Rainfall Algorithms*
August 1997	Alfred T.C. Chang and Long S. Chiu
Volume 13	*Interannual Variability and Potential Predictability in Reanalysis Products*
December 1997	Wie Ming and Siegfried D. Schubert
Volume 14	*A Comparison of GEOS Assimilated Data with FIFE Observations*
August 1998	Michael G. Bosilovich and Siegfried D. Schubert
Volume 15	*A Solar Radiation Parameterization for Atmospheric Studies*
June 1999	Ming-Dah Chou and Max J. Suarez
Volume 16	*Filtering Techniques on a Stretched Grid General Circulation Model*
November 1999	Lawrence Takacs, William Sawyer, Max J. Suarez, and Michael S. Fox-Rabinowitz
Volume 17	*Atlas of Seasonal Means Simulated by the NSIPP-1 Atmospheric GCM*
July 2000	Julio T. Bacmeister, Philip J. Pegion, Siegfried D. Schubert, and Max J. Suarez
Volume 18	*An Assessment of the Predictability of Northern Winter Seasonal Means with the NSIPP1 AGCM*

December 2000

Philip J. Pegion, Siegfried D. Schubert, and Max J. Suarez

Volume 19
July 2001

A Thermal Infrared Radiation Parameterization for Atmospheric Studies

Ming-Dah Chou, Max J. Suarez, Xin-Zhong Liang, and Michael M.-H. Yan

Volume 20
August 2001

The Climate of the FVCCM-3 Model

Yehui Chang, Siegfried D. Schubert, Shian-Jiann Lin, Sharon Nebuda, and Bo-Wen Shen

Volume 21
September 2001

Design and Implementation of a Parallel Multivariate Ensemble Kalman Filter for the Poseidon Ocean General Circulation Model

Christian L. Keppenne and Michele M. Rienecker

Volume 22
August 2002

A Coupled Ocean-Atmosphere Radiative Model for Global Ocean Biogeochemical Models

Watson W. Gregg

Volume 23
November 2002

Prospects for Improved Forecasts of Weather and Short-term Climate Variability on Subseasonal (2-Week to 2-Month) Time Scales

Siegfried D. Schubert, Randall Dole, Huang van den Dool, Max J. Suarez, and Duane Waliser

Volume 24
July 2003

Temperature Data Assimilation with Salinity Corrections: Validation for the NSIPP Ocean Data Assimilation System in the Tropical Pacific Ocean, 1993–1998

Alberto Troccoli, Michele M. Rienecker, Christian L. Keppenne, and Gregory C. Johnson

Volume 25
December 2003

Modeling, Simulation, and Forecasting of Subseasonal Variability

Duane Waliser, Siegfried D. Schubert, Arun Kumar, Klaus Weickmann, and Randall Dole

Volume 26

April 2005

Documentation and Validation of the Goddard Earth Observing System (GEOS) Data Assimilation System – Version 4

Senior Authors: S. Bloom, A. da Silva and D. Dee

Contributing Authors: M. Bosilovich, J-D. Chern, S. Pawson, S. Schubert, M. Sienkiewicz, I. Stajner, W-W. Tan, and M-L. Wu

Volume 27

December 2008

The GEOS-5 Data Assimilation System - Documentation of Versions 5.0.1, 5.1.0, and 5.2.0.

M.M. Rienecker, M.J. Suarez, R. Todling, J. Bacmeister, L. Takacs, H.-C. Liu, W. Gu, M. Sienkiewicz, R.D. Koster, R. Gelaro, I. Stajner, and J.E. Nielsen

Volume 28

April 2012

The GEOS-5 Atmospheric General Circulation Model: Mean Climate and Development from MERRA to Fortuna

Andrea Molod, Lawrence Takacs, Max Suarez, Julio Bacmeister, In-Sun Song, and Andrew Eichmann

Volume 29

June 2012

Atmospheric Reanalyses – Recent Progress and Prospects for the Future.

A Report from a Technical Workshop, April 2010

Michele M. Rienecker, Dick Dee, Jack Woollen, Gilbert P. Compo, Kazutoshi Onogi, Ron Gelaro, Michael G. Bosilovich, Arlindo da Silva, Steven Pawson, Siegfried Schubert, Max Suarez, Dale Barker, Hirotaka Kamahori, Robert Kistler, and Suranjana Saha

Volume 30

December 2012

The GEOS-iODAS: Description and Evaluation

Guillaume Vernieres, Michele M. Rienecker, Robin Kovach and Christian L. Keppenne

Volume 31

March 2013

Global Surface Ocean Carbon Estimates in a Model Forced by MERRA

Watson W. Gregg, Nancy W. Casey and Cécile S. Rousseaux

Volume 32

March 2014

Estimates of AOD Trends (2002-2012) over the World’s Major Cities based on the MERRA Aerosol Reanalysis

Simon Provencal, Pavel Kishcha, Emily Elhacham, Arlindo M. da Silva, and Pinhas Alpert

Volume 33
August 2014

The Effects of Chlorophyll Assimilation on Carbon Fluxes in a Global Biogeochemical Model

Cécile S. Rousseaux and Watson W. Gregg

Volume 34
September 2014

Background Error Covariance Estimation using Information from a Single Model Trajectory with Application to Ocean Data Assimilation into the GEOS-5 Coupled Model

Christian L. Keppenne, Michele M. Rienecker, Robin M. Kovach, and Guillaume Vernieres

Volume 35
December 2014

Observation-Corrected Precipitation Estimates in GEOS-5

Rolf H. Reichle and Qing Liu

Volume 36
March 2015

Evaluation of the 7-km GEOS-5 Nature Run

Ronald Gelaro, William M. Putman, Steven Pawson, Clara Draper, Andrea Molod, Peter M. Norris, Lesley Ott, Nikki Prive, Oreste Reale, Deepthi Achuthavarier, Michael Bosilovich, Virginie Buchard, Winston Chao, Lawrence Coy, Richard Cullather, Arlindo da Silva, Anton Darmenov, Ronald M. Errico, Marangelly Fuentes, Min-Jeong Kim, Randal Koster, Will McCarty, Jyothi Nattala, Gary Partyka, Siegfried Schubert, Guillaume Vernieres, Yuri Vikhliaev, and Krzysztof Wargan

Volume 37
March 2015

Maintaining Atmospheric Mass and Water Balance within Reanalysis

Lawrence L. Takacs, Max Suarez, and Ricardo Todling

Volume 38
September 2015

The Quick Fire Emissions Dataset (QFED) – Documentation of versions 2.1, 2.2 and 2.4

Anton S. Darmenov and Arlindo da Silva

Volume 39
September 2015

Land Boundary Conditions for the Goddard Earth Observing System Model Version 5 (GEOS-5) Climate Modeling System - Recent Updates and Data File Descriptions

Sarith Mahanama, Randal Koster, Gregory Walker, Lawrence Takacs, Rolf Reichle, Gabrielle De Lannoy, Qing Liu, Bin Zhao, and Max Suarez

Volume 40
October 2015

Soil Moisture Active Passive (SMAP) Project Assessment Report for the Beta-Release L4_SM Data Product

Rolf H. Reichle, Gabrielle J. M. De Lannoy, Qing Liu, Andreas Colliander, Austin Conaty, Thomas Jackson, John Kimball, and Randal D. Koster

Volume 41
October 2015

GDIS Workshop Report

Siegfried Schubert, Will Pozzi, Kingtse Mo, Eric Wood, Kerstin Stahl, Mike Hayes, Juergen Vogt, Sonia Seneviratne, Ron Stewart, Roger Pulwarty, and Robert Stefanski

Volume 42
November 2015

Soil Moisture Active Passive (SMAP) Project Calibration and Validation for the L4_C Beta-Release Data Product

John Kimball, Lucas Jones, Joseph Glassy, E. Natasha Stavros, Nima Madani, Rolf Reichle, Thomas Jackson, and Andreas Colliander

Volume 43
September 2015

MERRA-2: Initial Evaluation of the Climate

Michael G. Bosilovich, Santha Akella, Lawrence Coy, Richard Cullather, Clara Draper, Ronald Gelaro, Robin Kovach, Qing Liu, Andrea Molod, Peter Norris, Krzysztof Wargan, Winston Chao, Rolf Reichle, Lawrence Takacs, Yury Vikhliaev, Steve Bloom, Allison Collow, Stacey Firth, Gordon Labow, Gary Partyka, Steven Pawson, Oreste Reale, Siegfried Schubert, and Max Suarez

Volume 44
February 2016

Estimation of the Ocean Skin Temperature using the NASA GEOS Atmospheric Data Assimilation System

Santha Akella, Ricardo Todling, Max Suarez

Volume 45
October 2016

The MERRA-2 Aerosol Assimilation

C. A. Randles, A. M. da Silva, V. Buchard, A. Darmenov, P. R. Colarco, V. Aquila, H. Bian, E. P. Nowottnick, X. Pan, A. Smirnov, H. Yu, and R. Govindaraju

Volume 46
October 2016

The MERRA-2 Input Observations: Summary and Assessment

Will McCarty, Lawrence Coy, Ronald Gelaro, Albert Huang, Dagmar Merkova, Edmond B. Smith, Meta Sienkiewicz, and Krzysztof Wargan

Volume 47
May 2017

An Evaluation of Teleconnections Over the United States in an Ensemble of AMIP Simulations with the MERRA-2 Configuration of the GEOS Atmospheric Model.

Allison B. Marquardt Collow, Sarith P. Mahanama, Michael G. Bosilovich, Randal D. Koster, and Siegfried D. Schubert

Volume 48
July 2017

Description of the GMAO OSSE for Weather Analysis Software Package: Version 3

Ronald M. Errico, Nikki C. Prive, David Carvalho, Meta Sienkiewicz, Amal El Akkraoui, Jing Guo, Ricardo Todling, Will McCarty, William M. Putman, Arlindo da Silva, Ronald Gelaro, and Isaac Moradi

Volume 49
March 2018

Preliminary Evaluation of Influence of Aerosols on the Simulation of Brightness Temperature in the NASA Goddard Earth Observing System Atmospheric Data Assimilation System

Jong Kim, Santha Akella, Will McCarty, Ricardo Todling, and Arlindo M. da Silva

Volume 50
March 2018

The GMAO Hybrid Ensemble-Variational Atmospheric Data Assimilation System: Version 2.0

Ricardo Todling and Amal El Akkraoui

Volume 51
July 2018

The Atmosphere-Ocean Interface Layer of the NASA Goddard Earth Observing System Model and Data Assimilation System

Santha Akella and Max Suarez

Volume 52
July 2018

Soil Moisture Active Passive (SMAP) Project Assessment Report for Version 4 of the L4_SM Data Product

Rolf H. Reichle, Qing Liu, Randal D. Koster, Joe Ardizzone, Andreas Colliander, Wade Crow, Gabrielle J. M. De Lannoy, and John Kimball

Volume 53
October 2019

Ensemble Generation Strategies Employed in the GMAO GEOS-S2S Forecast System

Siegfried Schubert, Anna Borovikov, Young-Kwon Lim, and Andrea Molod

Volume 54
August 2020

Position Estimation of Atmospheric Motion Vectors for Observation System Simulation Experiments

David Carvalho and Will McCarty

Volume 55
February 2021

A Phenomenon-Based Decomposition of Model-Based Estimates of Boreal Winter ENSO Variability

Schubert, Siegfried, Young-Kwon Lim, Andrea Molod, and Alllison Collow

Volume 56
June 2021

Validation Assessment for the Soil Moisture Active Passive (SMAP) Level 4 Carbon (L4_C) Data Product Version 5

John S. Kimball, K. Arthur Endsley, Tobias Kundig, Joseph Glassy, Rolf H. Reichle, and Joseph V. Ardizzone

Volume 57
July 2021

Tendency Bias Correction in the GEOS AGCM

Yehui Chang, Siegfried Schubert, Randal Koster, and Andrea Molod

Volume 58
August 2021

Soil Moisture Active Passive (SMAP) Project Assessment Report for Version 5 of the L4_SM Data Product

Rolf H. Reichle, Qing Liu, Randal D. Koster, Joseph V. Ardizzone, Andreas Colliander, Wade Crow, Gabrielle J. M. De Lannoy, and John S. Kimball

Volume 59
November 2021

Observation-Corrected Land Surface Precipitation for the SMAP Level 4 Soil Moisture (Version 6) Product and the GEOS R21C Reanalysis

Rolf H. Reichle and Qing Liu

Volume 60
January 2022

Soil Moisture Active Passive (SMAP) Project Assessment Report for Version 6 of the L4_SM Data Product

Rolf H. Reichle, Qing Liu, Randal D. Koster, Joseph V. Ardizzone, Andreas Colliander, Wade Crow, Gabrielle J. M. De Lannoy, and John S. Kimball

Volume 61
April 2022

Validation Assessment for the Soil Moisture Active Passive (SMAP) Level 4 Carbon (L4_C) Data Product Version 6

John S. Kimball, K. Arthur Endsley, Tobias Kundig, Joseph Glassy, Rolf H. Reichle, Joseph V. Ardizzone

Volume 62

September 2022

Climate Characteristics of the Atmospheric Analysis Increments from the GEOS S2S V3 AOGCM Replayed to MERRA-2

Yehui Chang, Siegfried Schubert, Randal Koster, Andrea Molod, and Young-Kwon Lim