MASTER THESIS

Development of a real-time tool for deriving the optical properties of aerosol particles in PANGEA observatory, using ground-based passive remote sensing measurements.

Author: Georgiou Anastasia

Master of Science in Space Science Technologies and Applications

Department of Informatics and Telecommunications, University of Peloponnese
National Observatory of Athens, Greece

Thesis Supervisor

Dr. V. Amiridis – Research Director IAASARS National Observatory of Athens

Development of a real-time tool for deriving the optical properties of aerosol particles in PANGEA observatory, using ground-based passive remote sensing measurements.

Athens, 31 August 2020

Author

Name:	Anastasia Georgiou
Programme:	MSc in Space Science Technologies and Applications
Institute:	University of Peloponnese Department of Informatics and Telecommunications National Observatory of Athens, Greece
Student number:	2022201703002
Email:	anastasia.georgiou.anthi@gmail.com

Advisory Committee

Dr. Vassilis Amiridis – Research Director
Institute for Astronomy, Astrophysics, Space, Applications & Remote Sensing (IAASARS) National Observatory of Athens

Dr. Iphigenia Keramitsoglou – Research Director
Institute for Astronomy, Astrophysics, Space, Applications & Remote Sensing (IAASARS) National Observatory of Athens

Dr. Olga Sykioti – Senior Researcher
Institute for Astronomy, Astrophysics, Space, Applications & Remote Sensing (IAASARS) National Observatory of Athens

Abstract

The goal of this work is to create an automated tool that calculates in near real-time the optical properties of aerosol particles observed in the PANhellenic GEophysical observatory of Antikythera (PANGEA), when the particles have spherical shapes. For this scope we use the microphysical properties of the particles provided from the sun-photometer measurements at the station by the Aerosol Robotic Network (AERONET; Holben et al., 1998), and the Mie scattering code (Mie, 1908). First, the microphysical properties of the particles are automatically downloaded and displayed for the user. Then, we use them as input in the Mie code and we calculate the extinction, scattering, absorption and backscatter coefficients of the particles.

The constructed tool focuses mainly on the lidar-related optical properties. These properties can be used in the future step of this work, that will aim to combine the sun-photometer with lidar measurements for better characterization of aerosol vertical properties.

In order to fulfill the objectives of this work we need to:

1. Create a tool that automatically downloads from the AERONET website the retrieved microphysical properties of the particles at the Antikythera AERONET station.
2. Check if the particles have spherical shapes, by requiring the percentage of spherical particles in the atmospheric volume to be >98%. This parameter is provided in the AERONET aerosol product.
3. In case (2) is true, use the Mie code to calculate the particle optical properties: the extinction, absorption, scattering and backscatter coefficients.

Keywords: Aerosols, PANGEA, AERONET, Remote Sensing, Mie code

Περίληψη

Στόχος αυτής της εργασίας είναι η δημιουργία ενός αυτοματοποιημένου εργαλείου, που υπολογίζει σε σχεδόν πραγματικό χρόνο, τις οπτικές ιδιότητες των σφαιρικών αιωρούμενων σωματιδίων (αερολύματα), που παρατηρούνται στο Παρατηρητήριο Γεωεπιστημών και Κλιματικής αλλαγής ΠΑΓΓΑΙΑ στα Αντικύθηρα. Γι' αυτή τη μεταπτυχιακή διατριβή, θα χρησιμοποιήσουμε τις μετρήσεις του φωτόμετρου CIMEL που είναι εγκατεστημένο στο παρατηρητήριο, καθώς και τα προϊόντα αναστροφής του παγκόσμιου δικτύου AERONET της NASA, σε συνδυασμό με τον κώδικα σκέδασης Mie (Mie, 1908). Αρχικά, οι μικροφυσικές ιδιότητες των σωματιδίων (όπως αυτές παρέχονται στο προϊόν AERONET), θα λαμβάνονται αυτόματα στον υπολογιστή του χρήστη όπου και θα εμφανίζονται στην οθόνη του. Στη συνέχεια, χρησιμοποιώντας τον κώδικα σκέδασης Mie, θα υπολογίζονται οι ακόλουθοι συντελεστές: α) εξασθένισης (extinction coefficients), β) σκέδασης (scattering coefficients), γ) απορρόφησης (absorption coefficients) και δ) οπισθοσκέδασης (backscatter coefficients) των σωματιδίων.

Το εργαλείο το οποίο υλοποιήθηκε στα πλαίσια της παρούσας μελέτης, εστιάζει στον υπολογισμό των οπτικών ιδιοτήτων των αερολυμάτων, που σχετίζονται με τα συστήματα ενεργούς τηλεπισκόπησης lidar. Στα πλαίσια μια μελλοντικής εργασίας, αυτές οι ιδιότητες θα μπορούσαν να χρησιμοποιηθούν συνδυάζοντας τις μετρήσεις από το φωτόμετρο CIMEL και τη διάταξη lidar PollyXT, με απώτερο στόχο τον καλύτερο χαρακτηρισμό των κατακόρυφων ιδιοτήτων των αερολυμάτων.

Για να εκπληρώσουμε τους στόχους αυτού του έργου πρέπει:

1. Να δημιουργηθεί ένα εργαλείο που λαμβάνει αυτόματα τις μετρήσεις του φωτόμετρου από τον σταθμό Antikythera AERONET.
2. Να γίνει έλεγχος εάν τα σωματίδια έχουν σφαιρικά σχήματα, απαιτώντας το ποσοστό σφαιρικών σωματιδίων στον ατμοσφαιρικό όγκο να είναι > 98%. Αυτή η παράμετρος παρέχεται από το προϊόν αερολύματος AERONET.
3. Στην περίπτωση που το (2) ισχύει, θα χρησιμοποιηθεί ο κώδικας σκέδασης Mie για να υπολογιστούν οι οπτικές ιδιότητες των σωματιδίων: οι συντελεστές εξασθένισης, οι συντελεστές απορρόφησης, οι συντελεστές σκέδασης και οι συντελεστές οπισθοσκέδασης.

Λέξεις-κλειδιά: Αερολύματα, ΠΑΓΓΑΙΑ, AERONET, Τηλεπισκόπηση, Mie code

Acknowledgments

I would like to express my sincere gratitude and appreciation to my supervisor Dr. Vassilis Amiridis and to the postdoctoral researcher of the Institute of Astronomy, Astrophysics, Space Applications and Remote Sensing (IAASARS) of National Observatory of Athens (NOA), Alexandra Tsekeri. I sincerely appreciate their assistance and their support in my thesis with title "Development of a real-time tool for deriving the optical properties of aerosol particles in PANGEA observatory, using ground-based passive remote sensing measurements.", in the framework of the "Master of Science in Space Science Technologies and Applications" of the Department of Informatics and Telecommunications of University of Peloponnese, in collaboration with NOA.

It would be an omission not to mention the many hours that Dr. Tsekeri spent with me to help me to this thesis project and the interesting discussions that we had in order to to understand better the research project. With her feedback and precious guidance, the exciting journey in the magic world of atmospheric physics was started and finally, this research was completed. Last but not least, I would like to thank, the Phd Candidate, Vasiliki Daskalopoulou.

This thesis was realized in the period of my pregnancy and the period that humanity was infected from the coronavirus COVID-19. In this difficult period, I would like to express my thankfulness to my parents, my sister and my husband Andrea Petrucci for their support.

Anastasia

Dedicated to my family and
more specifically to
my husband and to my son

Nomenclature

Symbol	Description
c	Speed of light
λ	Wavelength of incident radiation in meters
v	Frequency in Hertz
x	Size parameter of particle
π	Mathematical constant 3.14
r	Radius of particle
I	Scattering intensity
I_o	Initial scattering intensity
N_{sca}	Number of scatterers
α	Polarizability of the scatterer
R	Distance from scatterers
θ_{sca}	Scattering angle
ε_{ext}	Extinction coefficient
N	Number of particles (concentration) with diameter D
σ_{ext}	Extinction cross-section
Q_{ext}	Extinction efficiency
D	Diameter of particle
ε_{abs}	Absorption coefficient
σ_{abs}	Absorption cross-section
Q_{abs}	Absorption efficiency
ε_{sca}	Scattering coefficient
σ_{sca}	Scattering cross-section
Q_{sca}	Scattering efficiency
A	Geometric cross-section of the particle [cm^2]
n_1, n_2	Refractive index of the medium1 and medium2
θ_1, θ_2	Angle of incidence and refraction
ω	Single scattering albedo
g	Asymmetry parameter
θ	Angle between the direction of the incident and scattered light
$P(\theta)$	Normalizing of the phase function
τ	Aerosol optical thickness
α	Ångström exponent
I_{sca}	Stokes vector of scattering light
$F(\theta)$	Scattering matrix
I_{inc}	Stokes vector of incoming light
Q	Intensity of electromagnetic radiation linearly polarized (0^o, 90^o)
U	Intensity of electromagnetic radiation linearly polarized (45^o, -45^o)
V	Intensity of electromagnetic radiation circularly polarized
$[]^T$	Transpose matrix
$F_{11}(\theta) \ldots F_{44}(\theta)$	Independent matrix elements
$dV/d \ln r$	Volume size distribution

V_o	Column volume of particles per cross section of atmospheric column
σ	Standard deviation of the natural logarithm of the radius
$\tilde{n}$	Complex refractive index
n	Real part of complex refractive index
k	Imaginary part of complex refractive index

Acronyms

SSA	Single Scattering Albedo
AOD	Aerosol Optical Depth
AOT	Aerosol Optical Thickness
EMS	ElectroMagnetic Spectrum
EMR	ElectroMagnetic Radiation
AAI	Absorbing Aerosol Index
Radar	RAdio Detection and Ranging
Lidar	Light Detection and Ranging
RS	Remote Sensing
AERONET	AErosol Robotic NETwork
NASA	National Aeronautics and Space Administration
GUI	Graphical User Interface
VSD	Volume Size Distribution
WMO	World Meteorological Organization
WHO	World Health Organization
NOA	National Observatory of Athens
SSA	Single Scattering Albedo
EARLINET	European Aerosol Research LIdar NETwork
PANGEA	PANhellenic GEophysical observatory of Antikythera
GAW	Global Atmosphere Watch Programme
ESA	European Space Agency
CWI	Centrum Wiskunde and Informatica
FSF	Free Software Foundation
UML	Unified Modeling Language
IDE	Integrated Development Environment

List of figures

Figure 1. 1: Examples of the direct and the indirect effect of aerosols in the atmosphere 14
Figure 1. 2: Radiative forcing of climate between 1750 and 2011 14

Figure 2. 1: Electromagnetic spectrum (EMS) 17
Figure 2. 2: Earth's energy budget.....18
Figure 2. 3: Graph of size parameter 19
Figure 2. 4: Classification of polarization ((a) linear, (b) circular and (c) elliptical polarised light 20

Figure 3. 1: Refractive index 24

Figure 4. 1: AERONET Site Information Map Interface 28
Figure 4. 2: CIMEL sun-photometer 28
Figure 4. 3: Graphical User Interface of AERONET 29
Figure 4. 4: Volume size distribution of the particles in PANGEA on October 30th, 2019 30
Figure 4. 5: Real part of the refractive index of the particles in PANGEA on October 30th, 2019.....31
Figure 4. 6: Imaginary part of the refractive index 31

Figure 5. 1: Anaconda Navigator.....33
Figure 5. 2: Python IDE 34
Figure 5. 3: UML Activity Diagram of the system,.....35
Figure 5. 4: Run configuration per file – command line of Anaconda-Spyder environment ..39
Figure 5. 5: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.44) 42
Figure 5. 6: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.44) 43

Figure 6. 1: Size distribution on 25/06/2020 46
Figure 6. 2: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.675) 48
Figure 6. 3: Plot of Volume Size Distribution (25-06-2020, 14:37:53, wavelenght:0.675) 48
Figure 6. 4: Plot of Volume Size Distribution (25-06-2020, 15:07:07, wavelenght:0.675) 49
Figure 6. 5: Plot of Volume Size Distribution (25-06-2020, 16:01:05, wavelenght:0.675) 50
Figure 6. 6: Plot of Volume Size Distribution (25-06-2020, 16:26:17, wavelenght:0.675) 50
Figure 6. 7: Plot of Volume Size Distribution (26-06-2020, 04:37:15, wavelenght:0.675) 53
Figure 6. 8: Plot of Volume Size Distribution (26-06-2020, 05:04:04, wavelenght:0.675) 54
Figure 6. 9: Plot of Volume Size Distribution (26-06-2020, 05:55:58, wavelenght:0.675) 54
Figure 6. 10: Plot of Volume Size Distribution (26-06-2020, 13:31:35, wavelenght:0.675) ..55
Figure 6. 11: Plot of Volume Size Distribution (26-06-2020, 14:37:59, wavelenght:0.675) ..55
Figure 6. 12: Plot of Volume Size Distribution (26-06-2020, 15:07:12, wavelenght:0.675) ..56
Figure 6. 13: Plot of Volume Size Distribution (26-06-2020, 16:01:10, wavelenght:0.675) ..56
Figure 6. 14: Plot of Volume Size Distribution (26-06-2020, 16:27:12, wavelenght:0.675) ..57

List of TABLES

Table 3. 1: Complex refractive index of aerosols 25

Table 5. 1: Python version list and released year.....33
Table 5. 2: Symbols in the Activity Diagram of Fig. 5.3.....36

Table of Contents

Chapter 1 Thesis introduction and outline 14

Chapter 2 Electromagnetic light scattering in Earth's atmosphere 17

2.1 Electromagnetic light in Earth's atmosphere 17

2.2 Light scattering 18

Chapter 3 Atmospheric aerosol particles 23

3.1 Microphysical properties of aerosol particles 23

3.2 Optical properties of aerosol particles 25

Chapter 4 AERONET measurements at PANGEA observatory 28

4.1 CIMEL sun-photometer 28

4.2 AERONET aerosol product 29

Chapter 5 Algorithm 32

5.1 Python 32

5.2 Anaconda- Navigator 33

5.3 Algorithm 34

5.4 Code analytic description 37

Chapter 6 Results 45

6.1 Results of 27/04/2020 45

6.2 Results of 25/06/2020 45

6.3 Results of 25/06/2020 – 26/06/2020 51

6.4 Discussion - Future perspectives 57

Bibliography 58

Appendices 61

Appendix A: Mie_code.py 61

Appendix B: My_mie.py 67

Appendix C: Bhmie.py 72

Chapter 1 Thesis introduction and outline

Aerosols can affect Earth's energy budget and therefore its climate, by directly scattering, absorbing and emitting the incoming solar radiation, and by indirectly influencing the cloud properties (Salby, 1996; Spyrou et al., 2013) (Fig. 1.1).

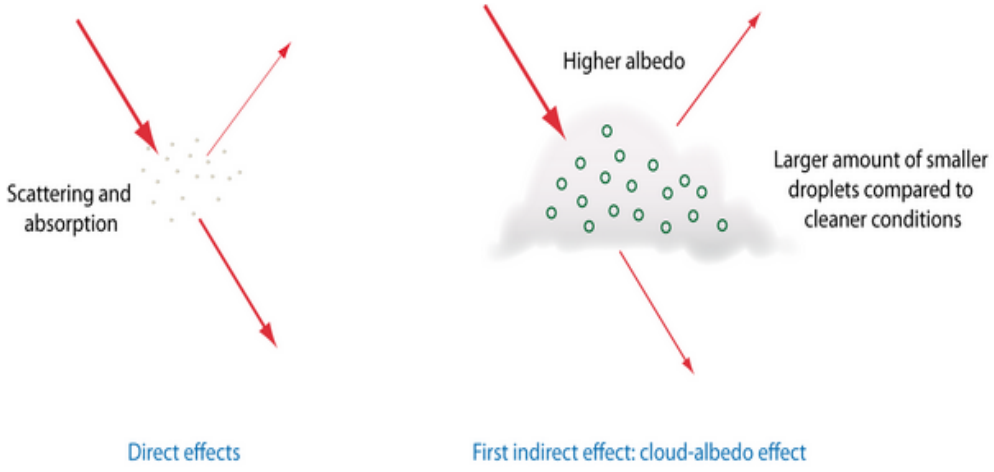


Figure 1. 1: Examples of the direct and the indirect effect of aerosols in the atmosphere(Source: IPCC, Fifth Assessment Report, 2013)

Figure 1.2 shows the radiative forcing of natural and anthropogenic agents for the period of 1750 (industrial revolution) to 2011 (urbanization). As shown in the figure, during the industrial era the aerosol radiative forcing is associated with large uncertainty. This uncertainty further propagates to climate predictions (IPCC, 2013).

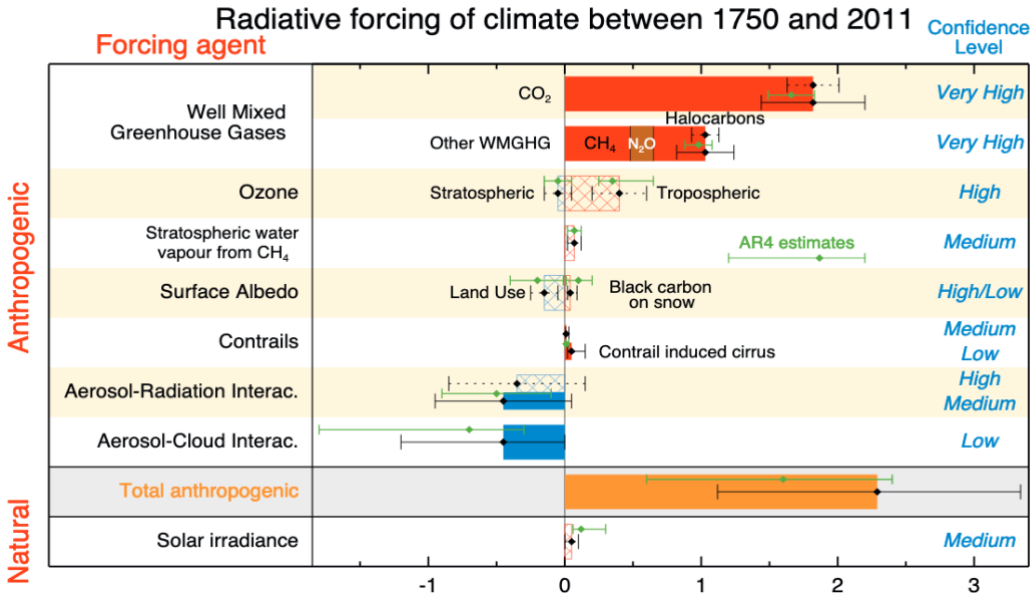


Figure 1. 2: Radiative forcing of climate between 1750 and 2011*(Source: IPCC, Fifth Assessment Report, 2013)*

To respond to the scientific need for accurate aerosol observations in the Mediterranean, the National Observatory of Athens (NOA; http://www.noa.gr) has installed the PANGEA observatory in Antikythera island. The observatory is created based on the standards of the World Meteorological Organization (GAW/WMO). The site is representative of the broader region, in the crossroad of Europe, Africa and Asia. In Antikythera the anthropogenic activities are very few, due to its population of approximately twenty (20) habitants. This constitutes the island as an ideal place for monitoring natural aerosols (dust, marine, volcanic and smoke particles) and the transported anthropogenic aerosols from big cities in Greece and the surrounding Mediterranean countries. Currently, NOA operates in PANGEA a PolyXT lidar, part of the European Aerosol Research LIdar NETwork (EARLINET, Bosenberg et al., 2000), and a CIMEL sun-photometer, part of the AERONET network.

The main goal of this thesis is to implement a new automated tool for downloading the AERONET product derived from the sun-photometer measurements in PANGEA, that provide the microphysical characteristics of the aerosol particles at the station. The downloaded microphysical parameters (e.g., the particle size distribution and refractive index) are then displayed in graphical plots in near real-time, and further used to calculate the optical properties of the particles, when the latter have spherical shapes.

The algorithm utilizes the programming language of Python (version 3.7) and the Anaconda graphical user interface. The program starts with a request to the AERONET server in order to download the AERONET product of the particle volume size distribution, the refractive index and the percentage of spherical particles, derived from sun-photometer measurements at Antikythera station. More specifically, it downloads the total volume concentration, the geometric mean volume radius and the geometric standard deviation for both fine and coarse modes, as well as the real part of refractive index at 440, 675, 870 and 1020 nm, the imaginary part of refractive index at 440, 675, 870 and 1020 nm and the percentage of spherical particles. Using these parameters as input in the Mie code, we calculate the extinction, scattering, absorption and backscatter coefficients, separately for the ensemble, the fine and coarse modes.

Thesis Outline

The thesis is composed by six chapters. The *first chapter* begins with a short introduction about the thesis, the research goal and the thesis outline. The *second chapter* focuses on the interaction of light with the aerosol particles in the Earth's atmosphere which defines the remote sensing measurements used herein. In the *third chapter*, we provide an overview of the optical and the microphysical properties of the aerosols in the atmosphere. The *fourth chapter* introduces the AERONET ground-based aerosol monitoring network and its aerosol product. The *fifth chapter* presents the algorithm used to download the AERONET product for the particles observed at PANGEA, and calculate the aerosol optical properties using the Mie code. It also provides a brief overview about the programming language of Python and the graphical

user interface of Anaconda Navigator. The *sixth chapter* includes the results of the study and the future work, as well as the code used.

Chapter 2 Electromagnetic light scattering in Earth's atmosphere

2.1 Electromagnetic light in Earth's atmosphere

Sun is the main power source of Earth and its atmosphere. The photons that are emitted by the Sun reach the Earth after eight minutes and the Earth's atmosphere acts as a filter of this radiation (Fleagle, 1980). The energy from the Sun is propagated on Earth as an electromagnetic wave, characterized by wavelength (λ), frequency (v) and velocity (c). The velocity is given by the following formula (Eq. 2.1):

$$c = v\,\lambda \tag{2.1}$$

As shown in Fig. 2.1, there are seven classifications of the electromagnetic radiation (EMR) based on wavelength (Halliday, Krane & Resnick, 2009):

- Gamma rays (γ)
- X-rays (Hard & Soft)
- Ultraviolet (UV)
- Visible spectrum
- Infrared (IR)
- Microwaves
- Radio waves

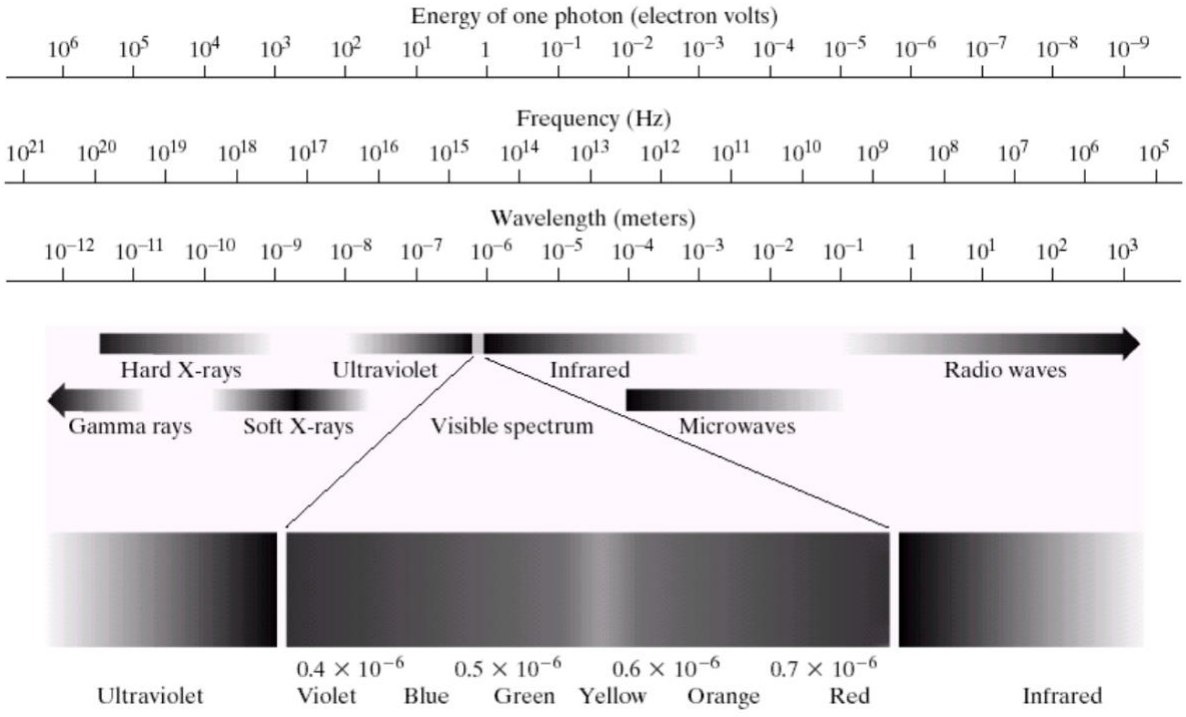


Figure 2. 1: Electromagnetic spectrum (EMS) (Source: Digital Image Processing, Gonzalez - Woods)

The Solar energy per unit area (W/m^2) is called **solar irradiance**. According to the "Earth observatory of NASA" (https://earthobservatory.nasa.gov), the average annual solar irradiance entering the Earth's system is approximately 1,360 W/m^2 (Fig 2.2). Approximately, 70% of the incoming solar energy is absorbed whereas 30% is reflected back to space (Haldoupis, 2015). This proportion of incoming and outgoing energy is called the **Earth's energy balance**, and it is used by the scientific community to understand better the global warming of the planet (Kant et al., 2017; Loeb et al., 2018).

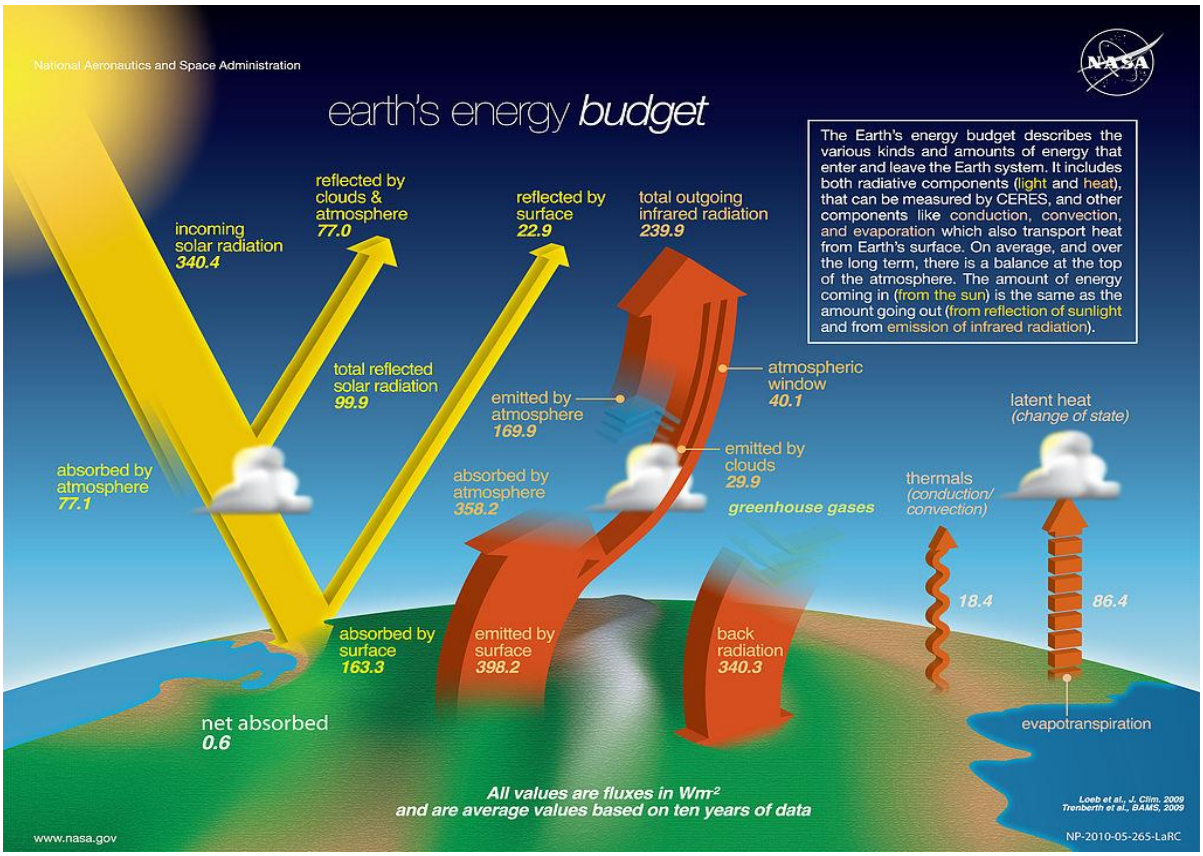


Figure 2. 2: Earth's energy budget (Source: https://www.nasa.gov)

2.2 Light scattering

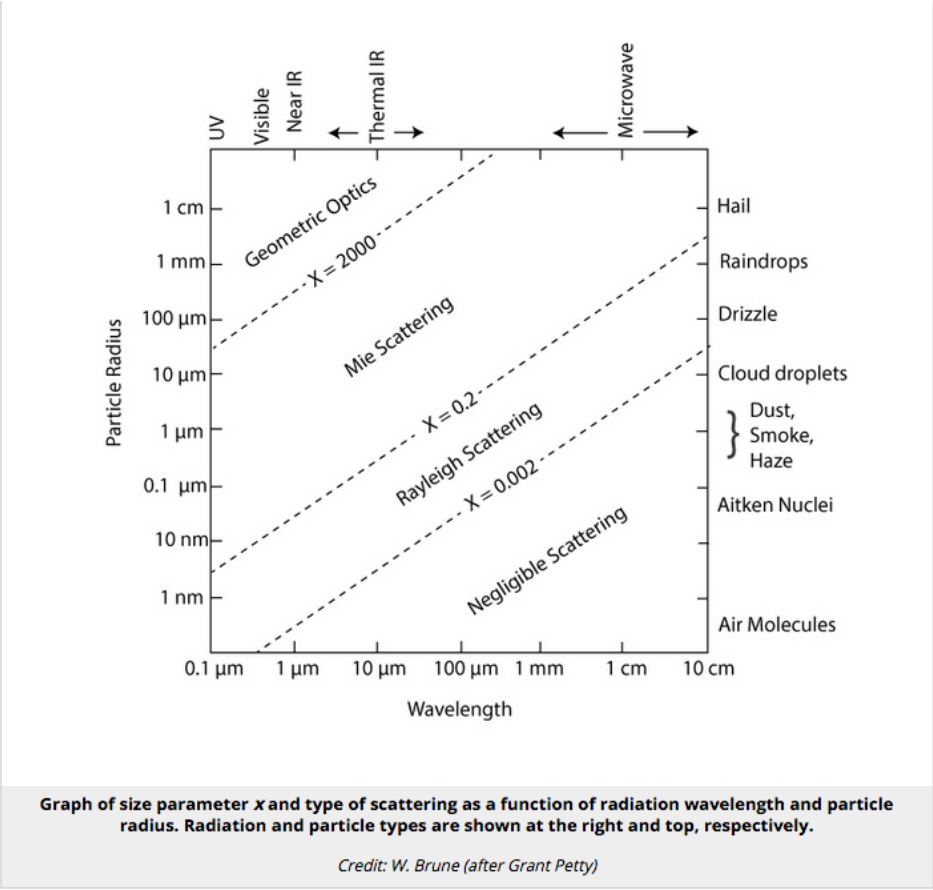
Light scattering is the process where the light is absorbed by the matter and reemitted in different directions (Haldoupis, 2015). The scattering is **elastic** when the energy and the wavelength of the scattered light do not change, and it is **inelastic** when they do.

Rayleigh scattering happens when the particles are much smaller than the wavelength (λ) of the incident light (Hansen & Travis, 1974), e.g. as gas molecules are for visible light. The intensity of the scattered light is proportional to wavelength (λ^{-4}), as shown in Eq. 2.2 (Salby, 1997).

$$I = I_0 \left(\frac{8\pi^4 N\, a^2}{\lambda^4 R^2}\right)(1 + \cos^2\theta) \qquad \textbf{(2.2)}$$

where α: polarizability
I_o: initial scattering intensity
θ_{sca}: scattering angle
R: distance from the scatterer
π: mathematical constant 3.14
N:number of scatterer

Mie scattering (Mie, 1908) happens when the particles have similar sizes to the wavelength of the incident light. More specifically, the size parameter ($x=2\pi r/\lambda$, r=particle radius) is close to 1 (Fig. 2.3). The Mie theory applies only to spherical particles.



Graph of size parameter *x* and type of scattering as a function of radiation wavelength and particle radius. Radiation and particle types are shown at the right and top, respectively.

Credit: W. Brune (after Grant Petty)

Figure 2. 3: Graph of size parameter

(from Sustainable Hydraulics in the Era of Global Change, CRC Press)

Geometric scattering is the scattering of light for particles much bigger than the wavelength (x»1).

The light from the Sun is ***unpolarized*** (Salby, 1997). Its propagation through the Earth's atmosphere changes its state of polarization. There are three types of polarization (Fig. 2.4):

- Linear polarization: The electric field of light oscillates along one plane in the same direction with the light propagation (Halliday, Krane & Resnick, 2009).
- Circular polarization: The electric field of the light creates a circular motion along the propagation direction.
- Elliptical polarization: The electric field of the light creates an elliptical motion along the propagation direction.

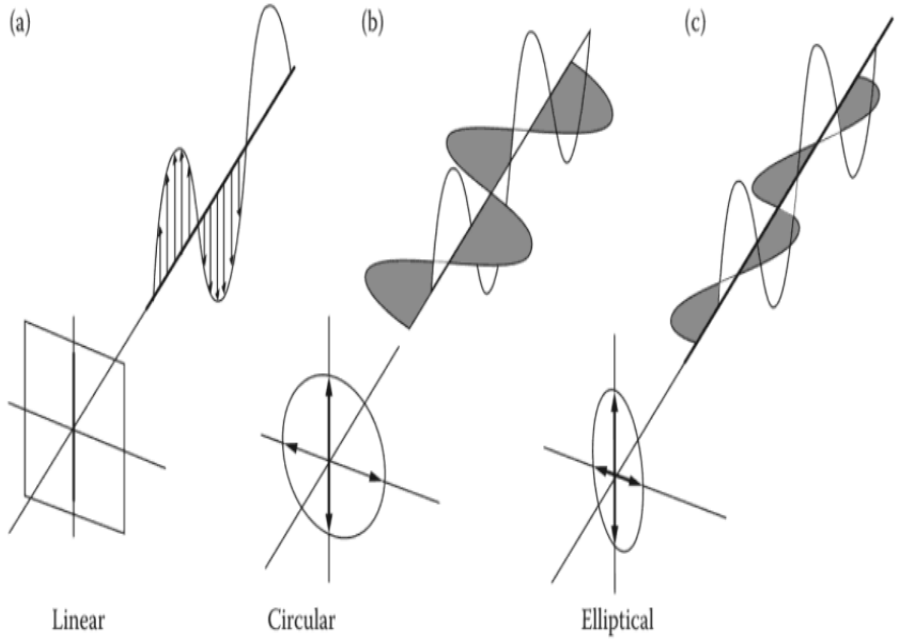


Figure 2. 4*: Classification of polarization ((a) linear, (b) circular and (c) elliptical polarised light (Source: Design Engineer's Sourcebook, CRC Press)*

The **Stokes vector** (Eq. 2.3) shows the polarization state of the light. In Eq. 2.3 the I, Q, U, V are known as *Stokes parameters* (Hansen and Travis, 1974):

$$I = \begin{bmatrix} I \\ Q \\ U \\ V \end{bmatrix} = [I\ Q\ U\ V]^T \qquad \textbf{(2.3)}$$

where I: total intensity of light (polarized and unpolarized)
Q: the linearly-polarized intensity along 0^o, 90^o

U: the linearly-polarized intensity along 45^o, -45^o
V: the left- or right-circularly-polarized intensity

Some examples of Stokes vectors for different types of polarized light are presented below:

- Vertically-polarized (linear) $\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$
- Horizontally-polarized (linear) $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$
- - 45^o polarized (linear) $\begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$
- 45^o polarized (linear) $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$
- Unpolarized $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- Left-hand-polarized (circular) $\begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$
- Right-hand-polarized (circular) $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

The total intensity is given by the sum of the intensity of polarized light and the intensity of the unpolarized light (Eq. 2.4) (Hansen & Travis, 1974):

$$I = I_{pol} + I_{unpol} \quad \textbf{(2.4)}$$

The intensity of the polarized light is calculated as shown in Eq. 2.5 (Hansen & Travis, 1974):

$$I_{pol}^2 = Q^2 + U^2 + V^2$$

$$\Rightarrow \quad I_{pol} = (Q^2 + U^2 + V^2)^{1/2} \quad \textbf{(2.5)}$$

The **scattering matrix** *F(θ)* describes the scattering of light from a particle or an ensemble of particles. As shown in Eq. 2.6, it is used to calculate the *Stokes vector* of the incident light (I^{inc}), from the Stokes vector of the scattered light (I^{sca}) (Hansen and Travis, 1974).

$$\mathrm{I}^{sca}(\theta) = \frac{C_{sca}}{4\pi r^2} F(\theta) I^{inc} \quad \textbf{(2.6)}$$

where θ: the scattering angle

C_{sca}: scattering cross-section of the particle

π: mathematical constant 3.14

r: radius of the particle

The scattering matrix for particles with random orientation has the following block-diagonal form (Eq. 2.7) (Mishchenko & Travis, 1994):

$$F(\theta) = \begin{bmatrix} F_{11}(\theta) & F_{12}(\theta) & 0 & 0 \\ F_{12}(\theta) & F_{22}(\theta) & 0 & 0 \\ 0 & 0 & F_{33}(\theta) & F_{34}(\theta) \\ 0 & 0 & -F_{34}(\theta) & F_{44}(\theta) \end{bmatrix} \quad \textbf{(2.7)}$$

Chapter 3 Atmospheric aerosol particles

In this chapter we characterize the atmospheric aerosol particles based on their microphysical and optical properties.

Aerosols are solid or liquid microscopic particles suspended in the air, from natural or human-activity sources (Haldoupis, 2015). The atmospheric aerosols play an important role to:

- Weather (Kant et al., 2017)
- Air quality
- Climate (Stocker et al., 2013; Nabat et al., 2015)
- Human health (WHO, 2013)
- Energy sector
- Ecosystems (Guieu et al., 2010)
- Biogeochemistry
- Transportation

Aerosols are usually composed from (Giles et al., 2012):

- Black Carbon: Absorbs the solar energy and it is a key element in the warming of the troposphere. The major source of black carbon is the biomass and fossil fuel combustion.
- Mineral Dust. It comes from deserts, roads and agriculture.
- Sea Salt. It comes from marine sources (evaporation of water).
- Sulfate/ Nitrate. Volcanos are the main source of sulfate.

Aerosols are classified based on their size in the following categories:

- Ultrafine. These particles have diameters from 0.1 nm to 100 nm.
- Fine. The particles have diameters of <2.5 μm.
- Coarse. The particles have diameters of >2.5 μm.

There are two different sources of aerosols (Haldoupis, 2015):

- Natural sources, as volcanoes, fires (forest, grassland, bushfire), sandstorms, sea spray and vegetation. The aerosols from natural (primary) sources are emitted directly from the source.
- Anthropogenic sources from human activities such as exhaust emissions, power plants and road traffic.

3.1 Microphysical properties of aerosol particles

Aerosol particles are characterized based on their **microphysical properties**:

- Size and the number/surface/mass/volume size distribution
- Chemical composition and corresponding refractive index

- Shape
- Mass density

The **volume size distribution** of atmospheric aerosols is given by Eq. 3.1 (Remer et al., 1998):

$$\frac{dV}{dlnr} = \frac{V_0}{\sigma(2\pi)^{1/2}} \exp\left(-\frac{[\ln(r/r_m)]^2}{2\sigma^2}\right) \quad \textbf{(3.1)}$$

where dV/ dln r: volume size distribution [$\mu m^3/\mu m^2$]
V_o: total volume of the particles
r: radius
r_m: modal radius
σ: standard deviation

The **complex refractive index** *(n + i k)* describes the speed of light in a medium as shown in Eq. 3.2 (Snell's - Descartes law) (Kovalensko, 2001). It is determined by the chemical composition of the particle.

$$\sin(\theta_1)\, n_1 = \sin(\theta_2)\, n_2 \quad \textbf{(3.2)}$$

where n_1, n_2: refractive index of the medium1 and medium2 (Fig. 3.1)
θ_1, θ_2: angle of incidence and refraction

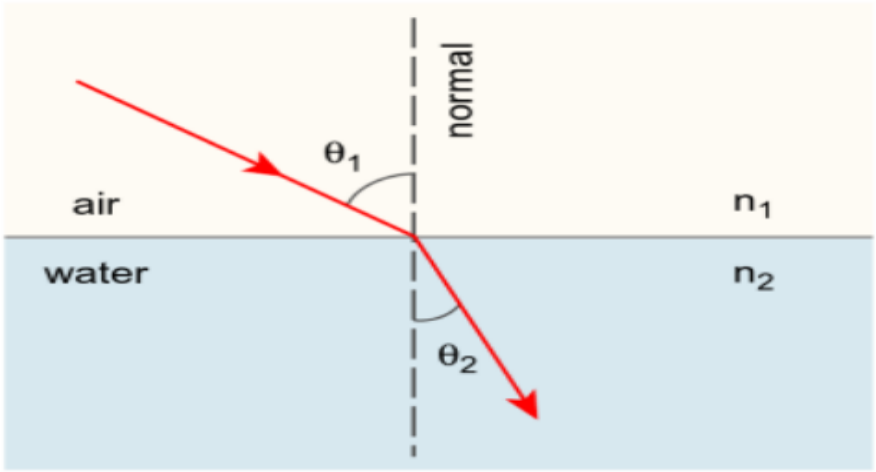


Figure 3. 1: Refractive index (Source: Computer and Information Security, Vacca)

Table 3.1 shows the complex refractive index for some atmospheric aerosol particles at 550 nm:

Atmospheric aerosol	**Complex refractive index**
Water	1.34 + i 0.00067

Sodium Chloride (NaCl)	1.54
Ammonium sulfate $(NH_4)_2SO_4$)	1.53
Atmospheric aerosol	**Complex refractive index**
Organic carbon (OC_a)	1.46
Elemental carbon (EC_a)	1.90 + i 0.20

Table 3. 1: Complex refractive index of aerosols

3.2 Optical properties of aerosol particles

The **optical properties** of the aerosol particles describe their interaction with the light:

- Extinction coefficient
- Scattering coefficient
- Absorption coefficient
- Backscatter coefficient
- Aerosol Optical Depth
- Ångström exponent
- Single Scattering Albedo (SSA or ω), is the ratio of the scattering to the extinction coefficient.
- Asymmetry parameter *(g)* (Eq. 3.3) (Zhao et al., 2017). Values of ~1 indicate predominance of forward scattering of the light from the particle, ~0 indicate isotropic scattering and ~-1 indicate predominance of backward scattering.

$$g = \frac{1}{2}\int_0^{\pi} cos\theta P(\theta)\sin(\theta)d(\theta) \quad \textbf{(3.3)}$$

where θ: the angle between the direction of the incident and scattered light
P(θ): the normalized phase function

Extinction is the sum of the scattering and absorption of the light. For an ensemble of particles, the **extinction coefficient** *(ε_{ext})* is proportional to the extinction cross-section (σ_{ext}) and the extinction efficiency (Q_{ext}) of the individual particles as shown in the following equations (Eq. 3.4) (Hansen and Travis, 1974):

$$\varepsilon_{ext} = \int N(r)\sigma_{ext}\,dr = \int N(r)Q_{ext}\,\pi r^2\,dr \quad \textbf{(3.4)}$$

where ε_{ext}: extinction coefficient
N(r): number size distribution of particles

σ_{ext}: extinction cross-section of particle with radius r
Q_{ext}: extinction efficiency of particle with radius r

The **absorption coefficient** *(ε_{abs})* is proportional to absorption cross-section (σ_{abs}) and to absorption efficiency (Q_{abs}) as shown in the following equation (Eq. 3.5) (Hansen and Travis, 1974):

$$\varepsilon_{abs} = \int N(r)\sigma_{abs}\, dr = \int N(r)Q_{abs}\, \pi r^2\, dr \qquad \textbf{(3.5)}$$

where ε_{abs}: absorption coefficient
σ_{abs}: absorption cross-section
Q_{abs}: absorption efficiency

The **scattering coefficient** *(ε_{sca})* is proportional to scattering cross-section (σ_{sca}) and to scattering efficiency (Q_{sca}) as shown the follow equation (Eq. 3.6) (Hansen and Travis, 1974):

$$\varepsilon_{sca} = \int N(r)\sigma_{sca}\, dr = \int N(r)Q_{sca}\, \pi r^2\, dr \qquad \textbf{(3.6)}$$

where ε_{sca}: scattering coefficient
σ_{sca}: scattering cross-section
Q_{sca}: scattering efficiency

The **backscatter coefficient** *(ε_{bck})* is proportional to the backscatter efficiency (Q_{bck}) as shown the follow equation (Eq. 3.7):

$$\varepsilon_{bck} = \int N(r)Q_{bck} \left(\frac{\lambda}{\pi r}\right)^2 dr \qquad \textbf{(3.7)}$$

where ε_{bck}: backscatter coefficient
Q_{bck}: backscatter efficiency

The **aerosol optical depth (AOD)** *(τ)* of the atmospheric column is calculated as shown in Eq. 3.8.

$$\tau = \int \varepsilon_{ext}(z)dz \qquad \textbf{(3.8)}$$

The Ångström exponent *(α)* at wavelengths λ_1 and λ_2 is calculated from the AOD at these wavelengths (Eq. 3.9). The Ångström exponent of particles with constant absorption at λ_1 and λ_2 is inversely related to the aerosol particle size (Toledano et al., 2007). Fine aerosols have Ångström exponent with values greater than one (α >1) and coarse aerosols have Ångström exponent with values smaller than one (α <1).

$$\alpha = -\frac{\ln\left(\frac{\tau_1}{\tau_2}\right)}{\ln\left(\frac{\lambda_1}{\lambda_2}\right)} \qquad \textbf{(3.9)}$$

where α: Ångström exponent

τ_1, τ_2: aerosol optical depth for wavelengths λ_1, λ_2

Chapter 4 AERONET measurements at PANGEA observatory

The AERONET network (https://aeronet.gsfc.nasa.gov/; Fig. 4.1) is a ground-based monitoring network which provides precise, columnar aerosol properties, using the direct-sun and diffused light measurements from the automatic, multispectral CIMEL sun-photometer (Fig. 4.2; https://www.cimel.fr) [Gregory, 2011; Barreto et al., 2015; Dubovic et al., 2000]. Specifically, the AERONET aerosol product provides the columnar optical, microphysical and radiative properties of the atmospheric aerosols (e.g. AOD, Angstrom exponent, phase function, asymmetry factor, volume size distribution, refractive index, etc.) (Holben, 1998). The worldwide spatial distribution of the AERONET stations is shown in Fig. 4.1.

Figure 4. 1: AERONET Site Information Map Interface (Source: https://aeronet.gsfc.nasa.gov)

4.1 CIMEL sun-photometer

Figure 4. 2: CIMEL sun-photometer (Source: https://aeronet.gsfc.nasa.gov)

The sun and sky measurements of CIMEL sun-photometer take place every day between sunlight and dawn. The CIMEL sun-photometer acquires measurements in nine spectral bands (340, 380, 440, 500, 670, 870, 940, 1020 and 1640 nm). The direct-sun measurements provide the AOD at the corresponding wavelengths. The direct-sun and diffuse-sky measurements are used in the inversion scheme of Dubovik and King (2000) and Dubovik et al. (2006), and provide retrievals of the optical and microphysical properties of the aerosol particles.

4.2 AERONET aerosol product

There are three versions available for the AERONET aerosol product: Version 1.0, 2.0 and 3.0. In the latest version (Version 3.0), the AOD is calculated based on the direct sun algorithm and there are three different data quality levels: Unscreened (Level 1.0), cloud-screened and quality controlled (Level 1.5) and quality-assured (Level 2.0). In Inversion Level 1.5, all the data are cleared from clouds and all the necessary quality controls are applied. The difference between the Inversion Level 1.5 and the Inversion Level 2.0 is that in the Inversion Level 2.0 the data have final calibration (pre-field, post-field) applied. Calibration is the procedure that converts the instrument measurement to a desired or known measurement. The instruments are calibrated before any measurements (Pre-field calibration) and also after the measurements (Post-field calibration) in order to ensure the reliability of the data.

The AERONET data products can be accessed through the AERONET web download tool in AERONET site (*https://aeronet.gsfc.nasa.gov*). AERONET data display uses a user-friendly interface that includes graphical representations to display the optical and microphysical properties of the aerosol particles. With the AERONET data download tool, the data can be obtained (Fig. 4.3).

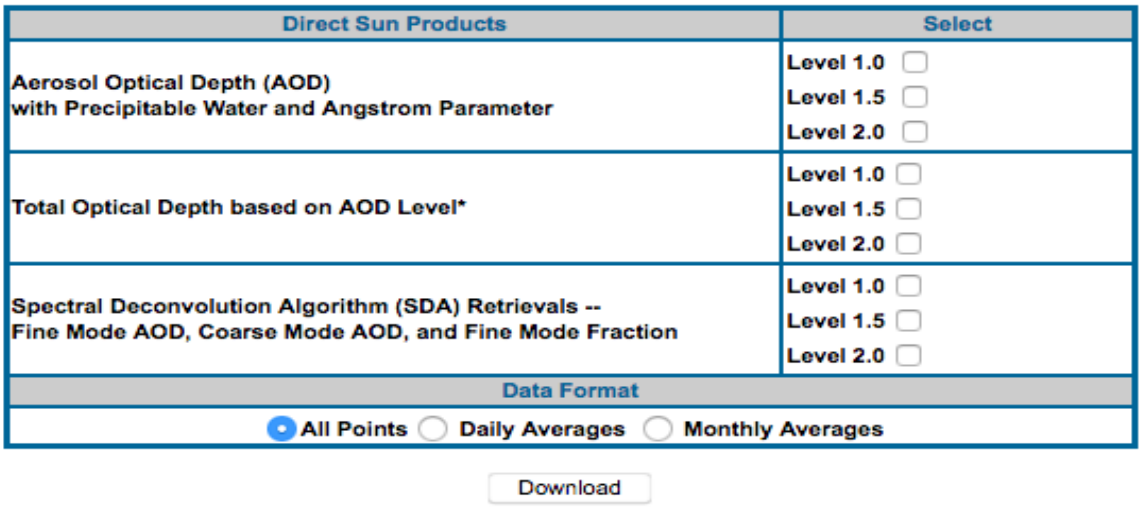


Figure 4. 3: Graphical User Interface of AERONET (Source: https://aeronet.gsfc.nasa.gov)

For this thesis, we download the AERONET aerosol product from the 'Antikythera_NOA' station. We choose the following parameters for the downloaded data:

- Inversion Level: Level 1.5
- Version 3 Inversion

The volume size distribution dV(r)/dlnr ($\mu m^3/\mu m^2$) is provided at twenty-two (22) different logarithmically points at equal distances with particle radii between 0.05 μm and 15 μm (Holben, 1998). Figure 4.4 displays the volume size distribution in three different time measurements at 07:12:53, 08:12:36 and 09:12:34, on October 30, 2019. The fitting error of the measurements for the three retrievals ("skyerr") is 1.0%, 1.1% and 1.2% respectively. The x-axis is the radius (logarithmic points) and the y-axis is the volume particle size distribution (values range from 0.00 to 0.03 $\mu m^3/\mu m^2$).

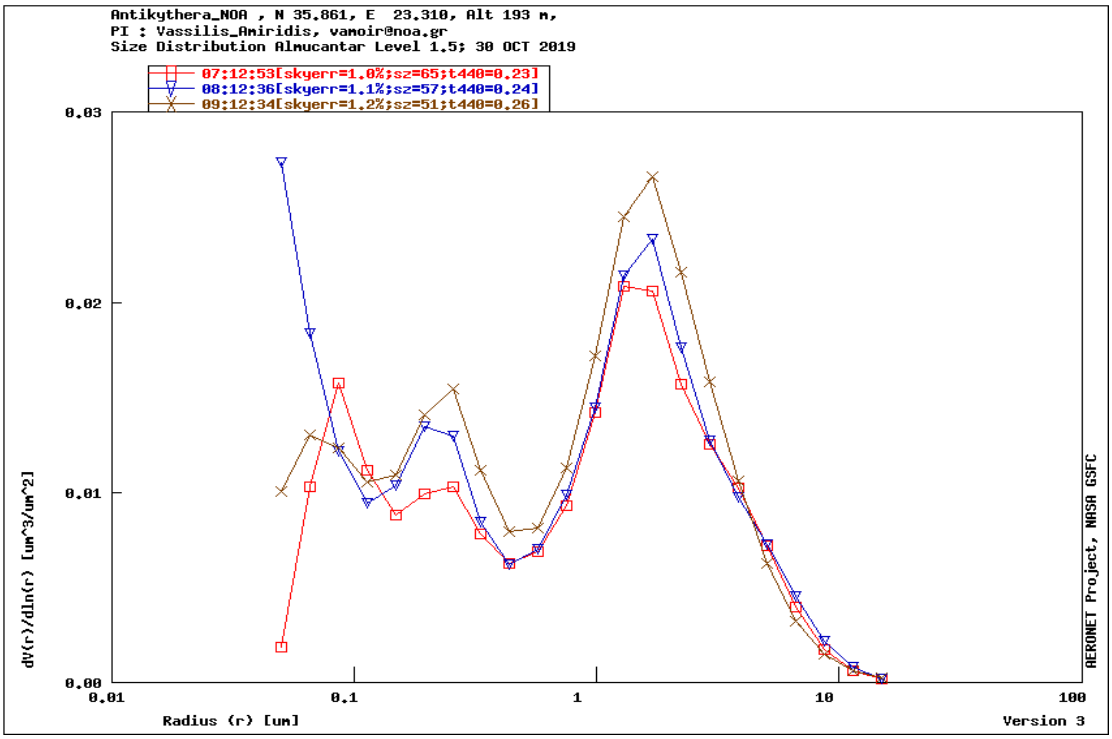


Figure 4. 4: Volume size distribution of the particles in PANGEA on October 30th, 2019 (Source: https://aeronet.gsfc.nasa.gov)

Figures 4.5 and 4.6 show the real and the imaginary part of the refractive index in three different time measurements at 07:12:53, 08:12:36 and 09:12:34, on October 30, 2019. The x-axis is the wavelength (nm) and the y-axis is the real and the imaginary part, respectively (values range from 1.40 to 1.60 and 0.00 to 0.03).

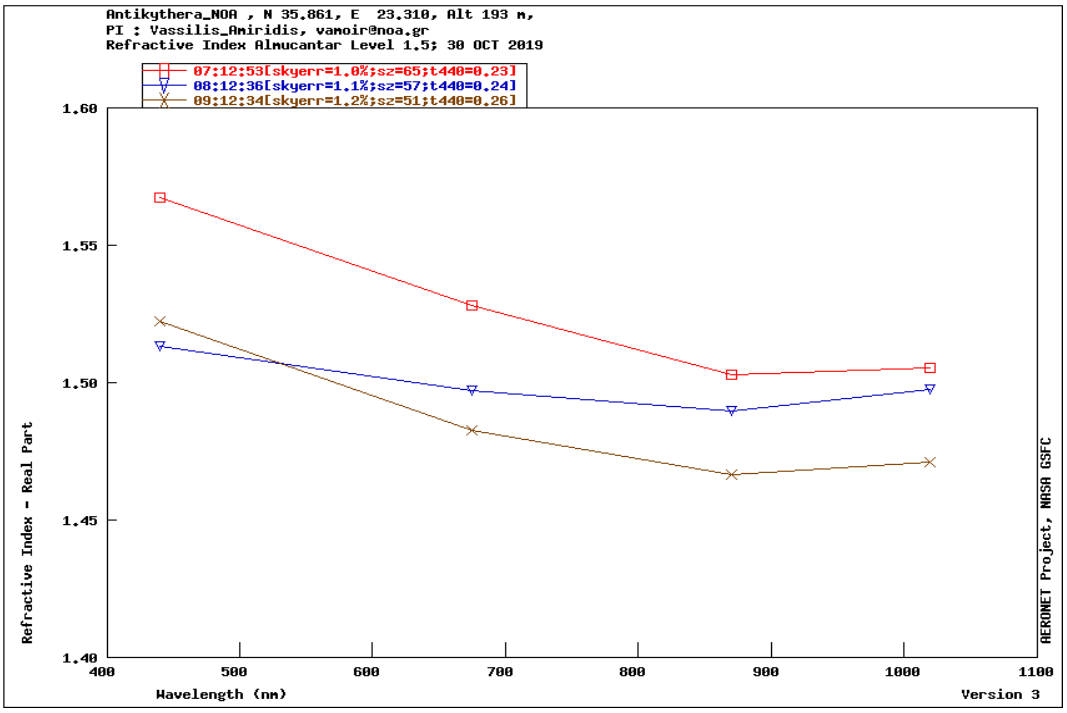


Figure 4. 5: Real part of the refractive index of the particles in PANGEA on October 30th, 2019 (Source: https://aeronet.gsfc.nasa.gov)

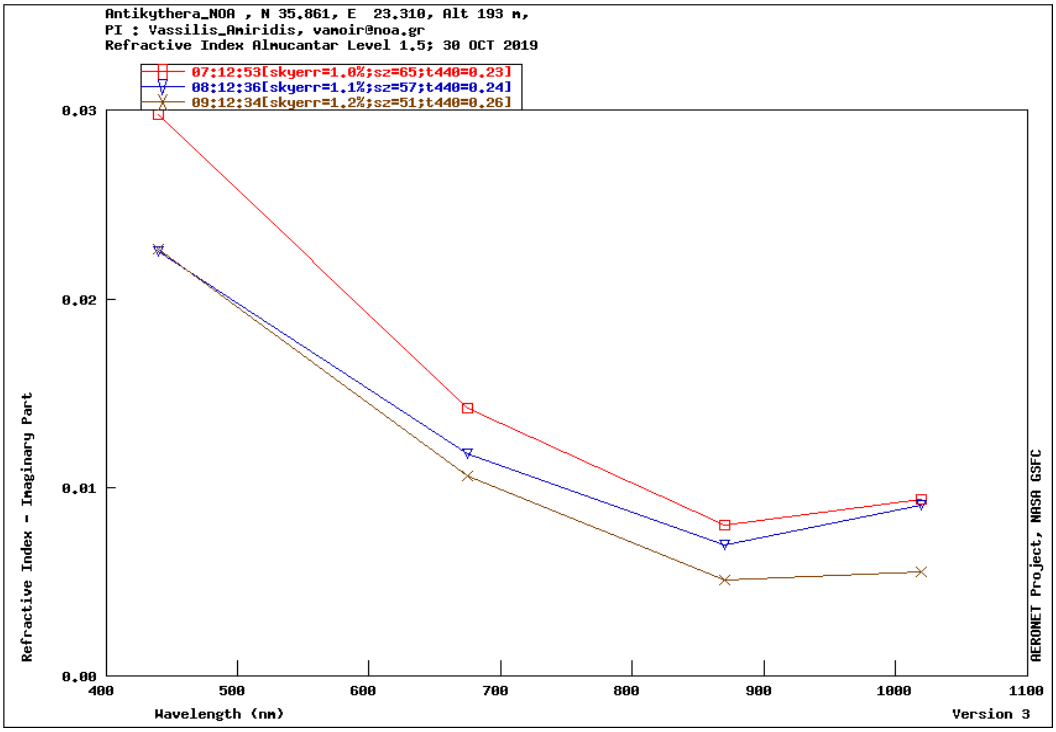


Figure 4. 6: Imaginary part of the refractive index of the particles in PANGEA on October 30th, 2019 (Source: https://aeronet.gsfc.nasa.gov*)*

Chapter 5 Algorithm

In this chapter we provide the algorithm for the automated download of the AERONET product from the PANGEA observatory and the calculation of the particle optical properties, using the Mie scattering code. A brief history of the powerful programming language Python, and a description of the graphical user interface (GUI) of Anaconda Navigator are also presented.

5.1 Python

The author of the programming language Python is the computer programmer from Netherlands, Guido Van Rossum. In 1989 he designed and created Python. The idea of the language name was inspired by the BBC's TV comedy "Monty Python's Flying Circus". Guido Van Rossum wanted an enigmatic, short and uncommon name for his project, so "Python" was born.
Python is a powerful tool because it simplifies complex programming computations making it a simple scripting language. "For someone who is not yet a programmer, who wants to become a programmer, for those people Python is particularly easy to get" says Guido Van Rossum. In 2001 Guido Van Rossum was awarded with the price of "Advancement of Free software" for his contribution of Python from the Free Software Foundation (FSF).

Python is a popular high-level programming language which is supported by many people around the world due to its reliability. Companies like IBM, Google and Cisco have developed a lot of their applications using Python as coding language. Table 5.1 shows chronological the different versions of Python (https://www.python.org). Python Version 1.0 was published in January 1994. The characteristics in this version were the programming tools such as the functions: map(), filter(), reduce() and lambda. In October 2000, Version 2.0 was released which supported also Unicode (standard computing encoding and text representation). The latest version of Python is Version 3.7.3.

Released Year	**Version of Python**
1994	Python 1.0
1997	Python 1.5
2000	Python 1.6
2000	Python 2.0
2001	Python 2.2
2003	Python 2.3
2004	Python 2.4
2006	Python 2.5
2008	Python 2.6
2009	Python 3.1

2011	Python 3.2
2014	Python 3.4
2015	Python 3.5
2016	Python 3.6
2018	Python 3.7
2019	Python 3.7.3

Table 5. 1: Python version list and released year

5.2 Anaconda- Navigator

Anaconda Navigator is a free open-source GUI of Conda distribution (https://docs.conda.io; https://docs.anaconda.com). Conda is a package manager which can distribute software from different languages such as Python, R, Ruby, Javascript, Java, C and C++. Initially, Anaconda Navigator was designed for scientific programs in Python. It is an environment that can be installed and run easily in the Operating Systems (Linux, Windows, macOS).

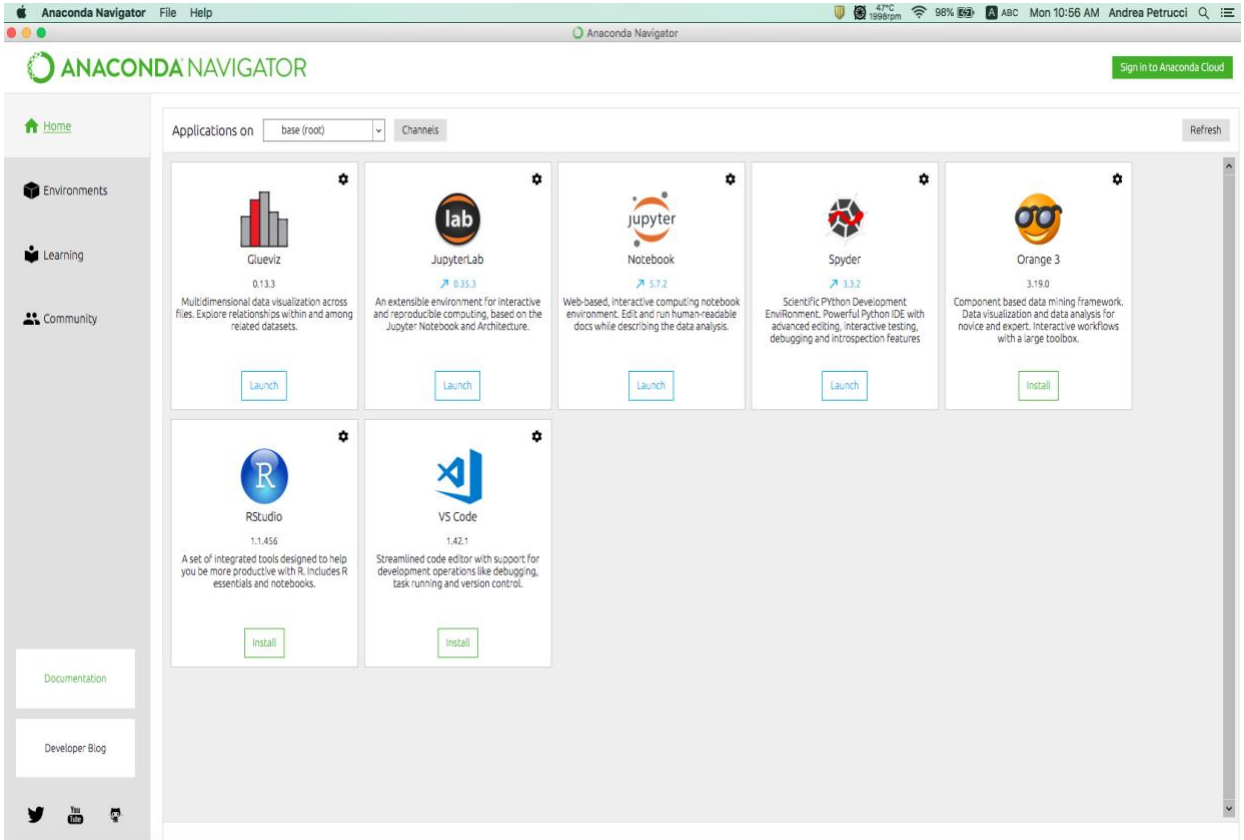


Figure 5. 1: Anaconda Navigator

The Anaconda Navigator can access the following applications by default (Fig. 5.1):

- Glueviz (Python library)

- JupiterLab (web-based user interface in order to develop open-source software (Project Jupiter))
- Notebook (open-source application which can be used to share documents that include code)
- Spyder (the development Environment of Python)
- Orange 3 (environment for data analysis. With Orange 3 the data can be visualized with an interactive way)
- RStudio (professional tool for statistical data analysis)
- VS Code (code editor for JavaScript, Node.js and TypeScript)

In this thesis, the code was written using the programming environment of Spyder and the Version 3.7 of Python. Spyder is a powerful Python Integrated Development Environment (IDE) for editing, testing and debugging.

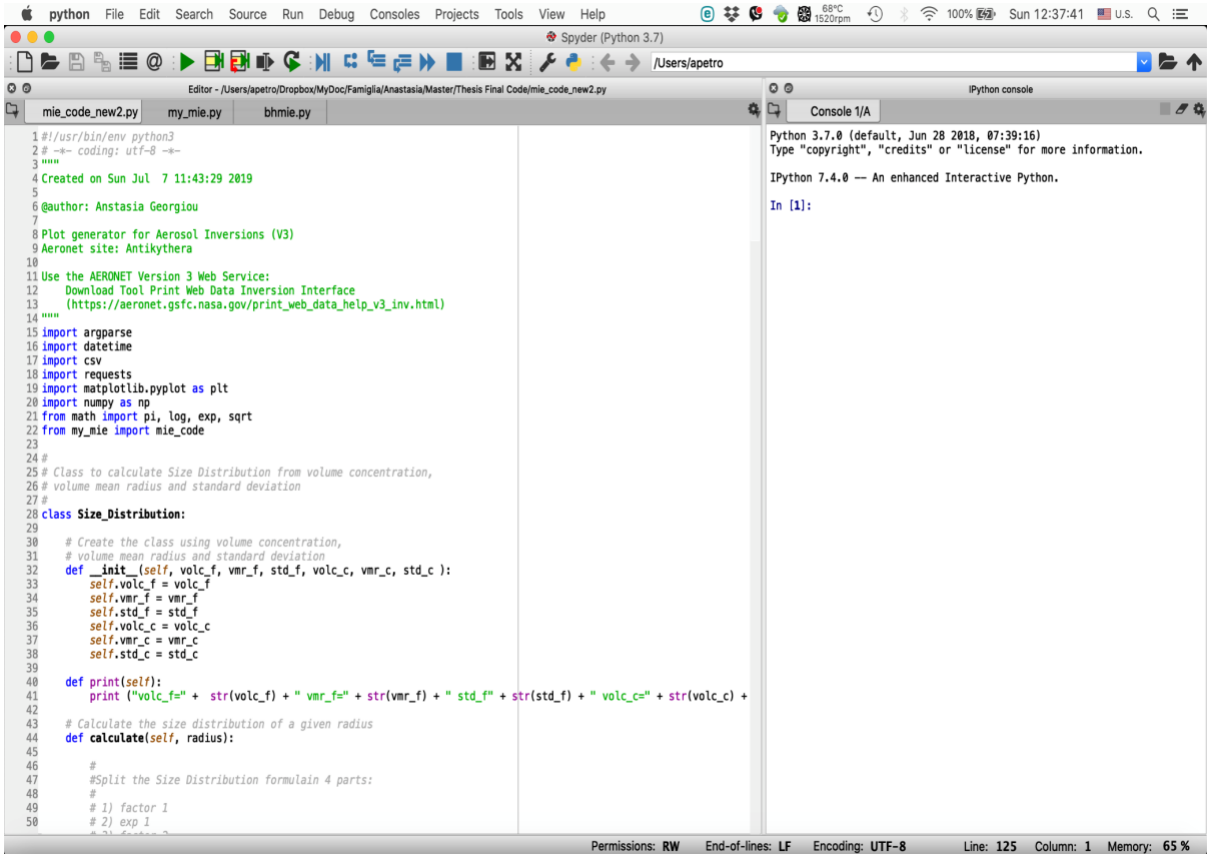


Figure 5. 2: Python IDE

5.3 Algorithm

The aim of this work is to develop a new automated tool that calculates the optical properties of aerosols, using the microphysical properties derived from the CIMEL sun-photometer measurements in the Antikythera AERONET station. Considering that the particles are spherical, the tool uses the Mie scattering code (bhmie.py) of Craig F.,

Bohren and Donald R. Fuffman (http://scatterlib.wdfiles.com/local--files/codes/bhmie.py), for the calculation of the optical properties of the particles. Figure 5.3 presents a Unified Modeling Language (UML) Activity Diagram that depicts the stages and the logic of the algorithm.

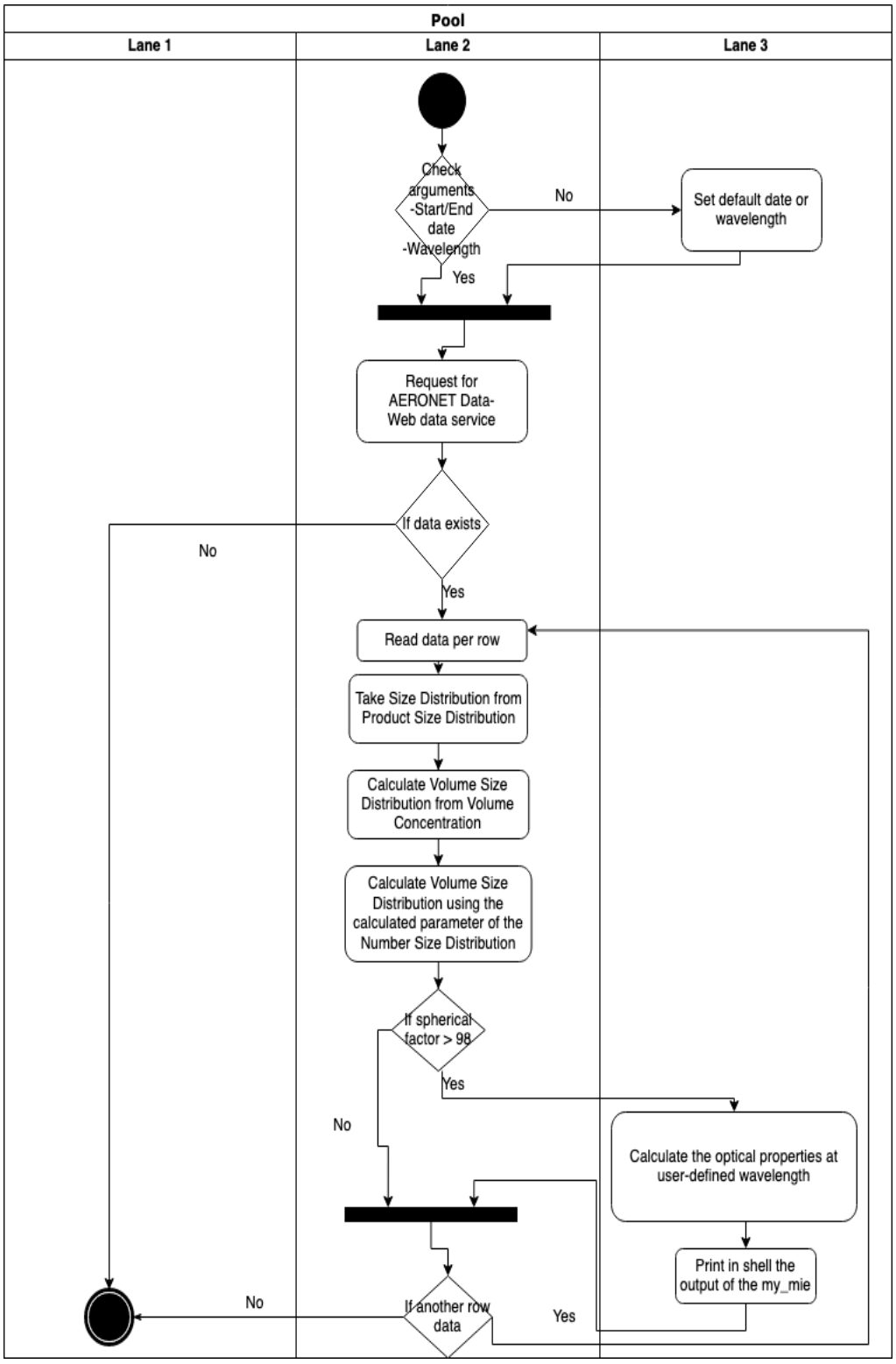


Figure 5. 3: UML Activity Diagram of the system, created from https://app.diagrams.net

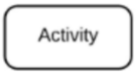
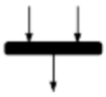
Symbol	Name of symbol	Description of symbol
	Start	It shows the start of a process.
	Connector	It represents the direction of the flow of the activities.
	Activity	It represents the activity.
	Note - Comment	It represents the comments and the notes (explanation) in the UML activity diagram.
	Joint	It represents the combination of two activities.
	Decision	It shows a decision.
[Condition]	Conditional text	It shows a conditional statement under which an activity can do something.
	End	It shows the end of a process.

Table 5. 2: Symbols in the Activity Diagram of Fig. 5.3

At the first step the user specifies the wavelength and the date or the period that wants to study. By default, the wavelength is equal to 0.44 μm and the period is the current day. Then a request to Antikythera AERONET server is being done for the AERONET data and the AERONET Web data service. If there are data for these dates, the program downloads automatically the relevant data. Otherwise, the system exits by informing the user that for these dates there are no data.

In the case that there are data, there is an iteration, because the same date there are measurements at different times. For a certain day, a certain time, the algorithm acquires the volume size distribution from the AERONET product and then it uses it to calculate the number size distribution, which is the input in the Mie code.

The next step involves checking for the spherical shape of the particles, based on the percentage of spherical particles in the atmospheric volume. If the percentage is more than 98% the particles are considered spheres and the algorithm continues to the next step. In the opposite case, the algorithm continues to the next iteration, to another row data with the same date in a different time or in a different day.

If the particles are spherical, the last step is the execution of the Mie code that calculates the particle extinction, scattering, absorption and backscatter coefficients. Specifically, the algorithm calculates these optical properties at the user defined wavelength, or at the default wavelength at 440 nm. The calculated optical properties (extinction, scattering, absorption and backscatter coefficient of the ensemble, fine mode and coarse mode) are saved in the shell of the system.

In the end, when all the data are processed, the program exits.

5.4 Code analytic description

The code starts with the import statements. Python looks for these specific imported modules and creates the relevant objects. It is important to mention that Python is an object-oriented programming language.

In the next step, a class is created with the name "Size_Distribution" and three functions are declared. With this class the volume size distribution is calculated, using the parameters provided from the AERONET product: the total volume concentration, the volume mean radius and the standard deviation for both fine and coarse modes (Eq. 5.1).

- With the function *_init_()* values are assigned to the parameters of the volume size distribution. The Python keyword 'self' shows the instance of the class of Size distribution.

- With the function *print ()* the values for fine and coarse modes of the total volume concentration ("volc_f" and "volc_c"), volume mean radius ("vmr_f" and "vmr_c"), standard deviation ("std_f" and "std_c"), real part ("rirp440", "rirp675", "rirp870" and "rirp1020") and imaginary part of refractive index ("riip440", "riip675", "riip870" and "rip1020) in four different wavelengths (440, 675, 870, 1020 nm) respectively, are printed, as shown below:

```
volc_f=0.013000 vmr_f=0.154000 std_f=0.450000 volc_c=0.023000
vmr_c=3.280000 std_c=0.601000 rirp440=1.600000 rirp675=1.534000
rirp870=1.512200 rirp1020=1.512100 riip440=0.001564 riip675=0.001537
riip870=0.001519 riip1020=0.001515
WVL 0.44
```

Also, the value of the wavelength is imported and printed, with default value 0.44 μm, as shown above.

- With the function *calculate ()* the values of the volume size distribution dV/dln(r) are calculated, using Eq. 5.1.

$$\frac{dV}{dlnr} = \frac{volc\_f}{\sqrt{2\pi}\ln(std\_f)} exp\left(-\frac{(\ln(r) - \ln(vmr\_f))^2}{2\ln(std\_f)^2}\right) + \frac{volc\_c}{\sqrt{2\pi}\ln(std\_c)} exp\left(-\frac{(\ln(r) - \ln(vmr\_c))^2}{2\ln(std\_c)^2}\right) \quad \textbf{(5.1)}$$

where $\frac{dV}{dlnr}$: the particle volume size distribution
$volc\_f$: the total volume concentration of fine mode
$volc\_c$: the total volume concentration of coarse mode
vmr_f: volume geometric mean radius of fine mode
vmr_c: volume geometric mean radius of coarse mode
std_f: geometric standard deviation of fine mode
std_c: geometric standard deviation of coarse mode

With the function web_service_request() a request is made to the URL of AERONET as shown below:

```
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=ASY&year=2019&month=7&da
y=18&year2=2019&month2=7&day2=18
```

The session objects allow the following parameters to pass onto the request:

- URL: The URL for the tool of Web data Inversion interface
- Product: There are four different products:
  - Siz: Volume Size Distribution file
    From the file product.siz, the program reads the radii of the size bins of the AERONET size distribution.
  - Vol: Volume Size Distribution Parameters file
    From the file product.vol, the program reads the geometric standard deviation, the total volume concentration and the geometric mean volume radius for both fine and coarse modes.
  - Rin: Refractive Index file (real and imaginary part)
    From the file product.rin, the program reads the refractive index the real and the imaginary part in different wavelengths (440 nm, 678 nm, 870 nm, 1020 nm).
  - Asy: Asymmetry Factor
    From the file product.asy, the program reads the sphericity factor (%).
- Start date (dd/mm/yyyy)

- End date (dd/mm/yyyy)

After the request to AERONET network, the corresponding files are downloaded. Through the interface of the parser module, the parse tree is created based on the downloaded files, with arguments, the start/end date and the wavelength. By default, the start and end date are the current day and the wavelength is 440 nm. If the user wishes to choose a different period or a different wavelength, he/she can use the following command:

-s *start_date(yyyy-mm-dd)* **-e** *end_date(yyyy-mm-dd)* **-w**l *wavelength_choice*
*For example: -**s** 2020-07-18 -**e** 2020-07-18 -**w**l 1.02*

The wavelength can take the following values: 0.44, 0.675, 0.87 and 1.02 μm. In the Anaconda-Spyder environment the command needs to be executed on 'run configuration per file' command line (Fig. 5.4).

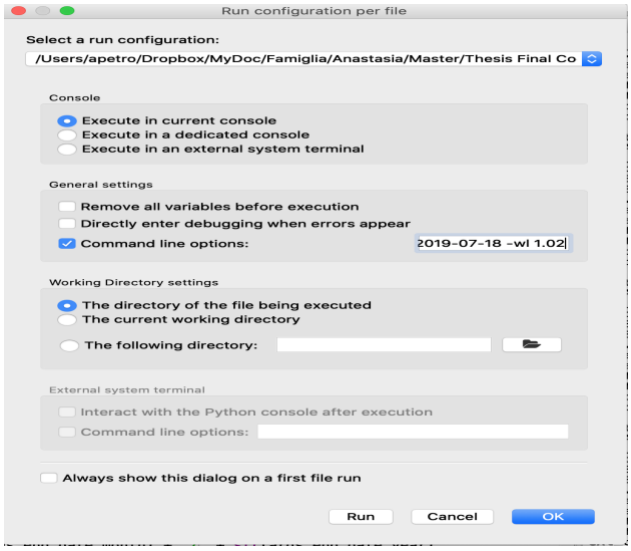


Figure 5. 4: Run configuration per file – command line of Anaconda-Spyder environment

If the data are not available for the user-defined days (or the current day by default) a relevant message appears to the user.

If the data are available, the radii of the size bins of the AERONET size distribution is read from the ".siz" file. From the ".vol" file the following parameters are read: the total volume concentration, volume mean radius and standard deviation for fine and coarse modes. The instance of the class Size_Distribution is called where the volume concentration (dv/dlnr) is calculated, and the relevant object of the class is created.

Then, the parameters of the number size distribution are calculated (i.e. for fine and coarse modes the total number concentration ("Nc_f" and "Nc_c") (Eq. 5.2 & Eq. 5.3) and the number mean radius ("nmr_f" and "nmr_c")) (Eq. 5.4 & Eq. 5.5). The geometric standard deviation ("std_f" and "std_c") does not change for number and volume size distributions.

$$Nc\_f = \sum \left(radius\_AER\_dif \cdot \left(\frac{1}{radius\_AER\_mean} \frac{volc\_f}{\sqrt{2\pi} std\_f} \frac{\exp\left(-\frac{((\ln(radius\_AER\_mean) - \ln(vmr\_f))^2)}{2std\_f^2}\right)}{\frac{4\pi}{3} radius\_AER\_mean^3} \right) \right) \quad \textbf{(5.2)}$$

Where vmr_f: volume geometric mean radius of fine mode
std_f: geometric standard deviation of fine mode
$radius\_AER\_mean$: the mean radius of the size bins of the AERONET size distribution
$radius\_AER\_dif$: width of the size bins of the AERONET size distribution

$$Nc\_c = \sum \left(radius\_AER\_dif \cdot \left(\frac{1}{radius\_AER\_mean} \frac{volc\_c}{\sqrt{2\pi} std\_c} \frac{\exp\left(-\frac{((\ln(radius\_AER\_mean) - \ln(vmr\_c))^2)}{2std\_c^2}\right)}{\frac{4\pi}{3} radius\_AER\_mean^3} \right) \right) \quad \textbf{(5.3)}$$

Where vmr_c: volume geometric mean radius of coarse mode
std_c: geometric standard deviation of coarse mode

$$nmr\_f = \frac{\exp\left(\ln(2 - vmr\_f) - 3std\_f^2\right)}{2} \quad \textbf{(5.4)}$$

Where nmr_f: number geometric mean radius of fine mode
vmr_f: volume geometric mean radius of fine mode
std_f: geometric standard deviation of fine mode

$$nmr\_c = \frac{\exp\left(\ln(2 - vmr\_c) - 3std\_c^2\right)}{2} \quad \textbf{(5.5)}$$

Where nmr_c: number geometric mean radius of coarse mode
vmr_c: volume geometric mean radius of coarse mode
std_c: geometric standard deviation of coarse mode

Using these parameters, the number size distribution is calculated, as shown in Eq. 5.6.

$$\frac{dN}{dlnr} = \frac{Nc\_f}{\sqrt{2\pi}\ln(std\_f)} exp\left(-\frac{(\ln(r) - \ln(nmr\_f))^2}{2\ln(std\_f)^2}\right) + \frac{Nc\_c}{\sqrt{2\pi}\ln(std\_c)} exp\left(-\frac{(\ln(r) - \ln(nmr\_c))^2}{2\ln(std\_c)^2}\right) \quad \textbf{(5.6)}$$

where $\frac{dN}{dlnr}$: the particle number size distribution
Nc_f: the total number concentration of fine mode
Nc_c: the total number concentration of coarse mode
nmr_f: number geometric mean radius of fine mode
nmr_c: number geometric mean radius of coarse mode
std_f: geometric standard deviation of fine mode
std_c: geometric standard deviation of coarse mode

The next step is to create the plots of the volume size distribution (from AERONET) for the specific time and date. In each plot, there are four different lines: The blue line denotes the volume size distribution values for each of the 22 size bins, as acquired from the AERONET product (*product.siz*). The orange line is the bimodal volume size distribution used to represent the data of the blue line using only six parameters: volc_f, vmr_f, std_f, volc_c, vmr_c, std_c. Then, the values of the volume size distribution are calculated as shown in Eq. 5.1. This is acquired from the AERONET product (*product.vol*) as well. The green line is a reproduction of the volume size distribution, using the formula $\frac{dV}{dlnr} = \frac{dN}{dlnr}\frac{4\pi radius\_AER\_mean^3}{3}$, using the calculated number size distribution. This line is used to verify our calculations for $\frac{dN}{dlnr}$ in Eq. 5.6, since in the ideal case it should coincide with the orange line. As shown in Fig. 5.5 below, this does not happen, mainly due to the limitation in Eq. 5.2 in using a representative value for the radius of the size bin (in Eq. 5.2 we used $radius\_AER\_mean$).

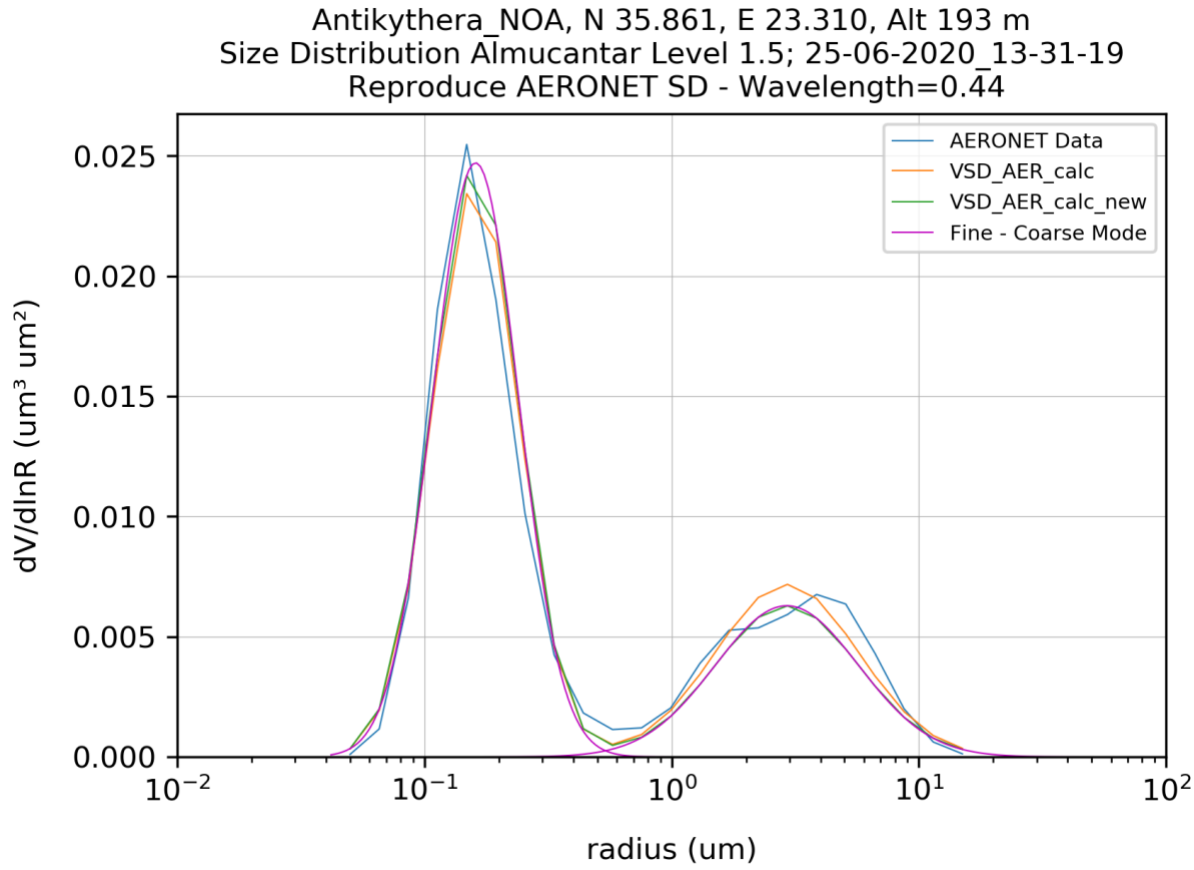


Figure 5. 5: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.44)

At the next step, if the sphericity factor is greater than 98%, the mie_code() function is called to calculate the optical properties at the user-defined wavelength. The inputs of the function are the user-defined wavelength, the parameters of the number size distribution (i.e., Nc_f, Nc_c, nmr_f, nmr_c, std_f and std_c) and the real and imaginary part of the refractive index (considered to be the same for fine and coarse modes).

Initially, the mie_code() function calculates new size bins (for the size distribution) for fine and coarse mode. More specifically, it finds the max and the min radii using the distribution parameters we provide in the input and it defines the representative radius of the bin. Then, it calculates the number concentration in each bin for fine and coarse mode respectively (Eq. 5.7 & Eq. 5.8).

$$dcon\_f = Nc\_f \exp\left(-1\,(ln\frac{r}{nmr\_f})^2/(2\,std\_f^{\,2})\right)\frac{1}{r\,std\_f\,\sqrt{2\pi}}$$

(Eq. 5.7)

where dcon_f: the number concentration in each bin for fine mode
Nc_f: the total number concentration of fine mode
nmr_f: number geometric mean radius of fine mode
r: bin center radii

std_f: geometric standard deviation of fine mode

$$dcon\_c = Nc\_c \exp\left(-1\ (ln\frac{r}{nmr\_c})^2/(2\ std\_c^2)\right)\frac{1}{r\ std\_c\ \sqrt{2\pi}} \quad \textbf{(Eq. 5.8)}$$

where dcon_c: the number concentration in each bin for coarse mode
Nc_c: the total number concentration of coarse mode
nmr_c: number geometric mean radius of coarse mode
r: bin center radii
std_c: geometric standard deviation of coarse mode

In the next step, the volume size distribution is calculated (SD_dv_dlnr) for fine and coarse mode (Eq. 5.9) and the pink line is created in the plot. This is a verification point in the algorithm (it should coincide with the green line), showing that the size distribution we provide in the input is indeed the one that is used in the Mie calculations.

$$SD\_dv\_dlnr = r\ \left(\frac{4\pi}{3}\right) r^3 dcon \quad \textbf{(Eq. 5.9)}$$

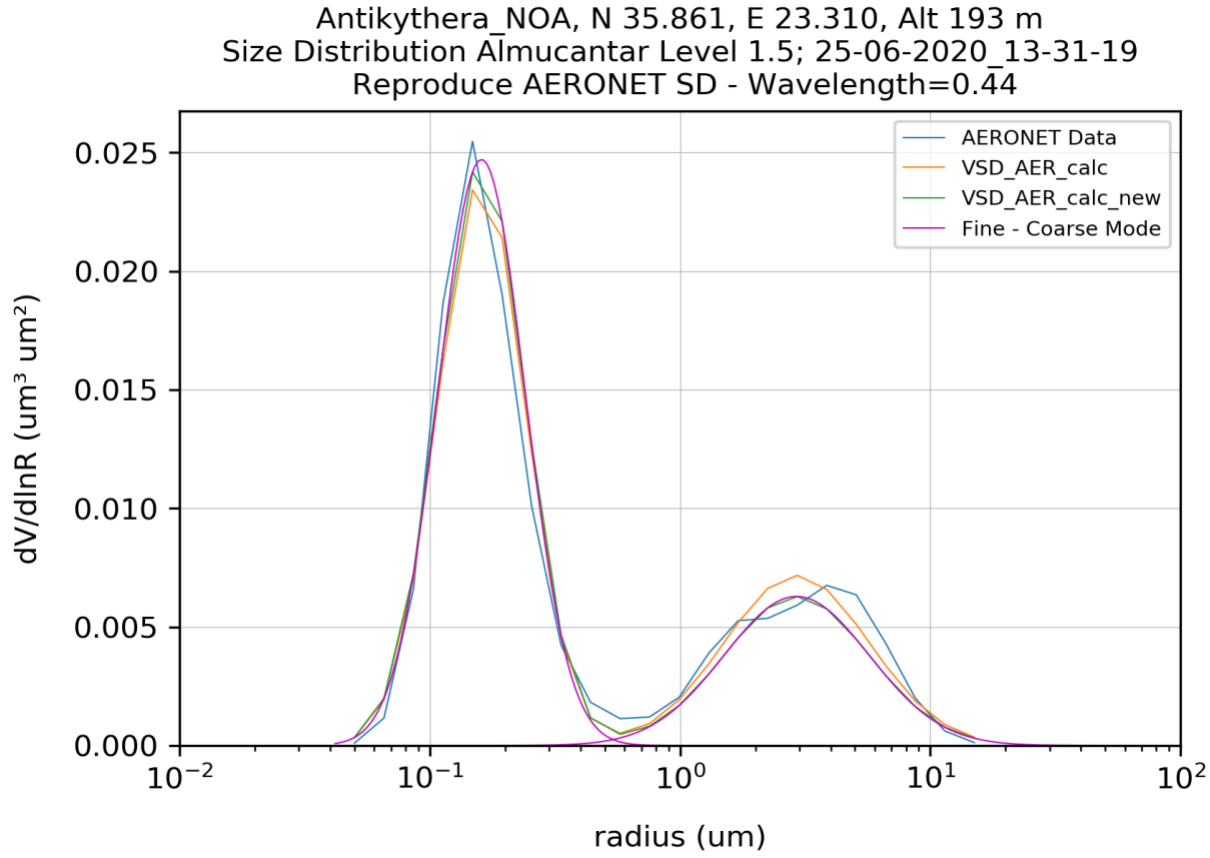


Figure 5. 6: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.44)

Furthermore, the mie_code() function uses the bmie() function that calculates the optical properties of the individual particle, with specific size and refractive index based on Mie scattering theory. The bhmie() function that was used in this thesis, was written in 1983 from Craig F. Bohren and Donald R. Huffman.

The bhmie() function considers a particle with the size of the representative radius of the bin, defined in the previous step (see Eq. 5.7 & Eq. 5.8). The inputs of the function are the size parameter ($x=2\pi r/\lambda$), the refractive index and the number of scattering angles in the range from 0 to $\pi/2$. The outputs are the extinction efficiency (q_{ext}), the scattering efficiency (q_{sca}), the backscatter efficiency (q_{back}), the asymmetry parameter (g_{sca}).

At the final step, the optical properties of the particle ensemble are calculated, using Eq. 3.4 – 3.7 for fine and coarse mode separately:

- The extinction coefficient
- The scattering coefficient
- The absorption coefficient
- The backscatter coefficient

Chapter 6 Results

6.1 Results of 27/04/2020

On the date of 27 of April on 2020, there are no available data from the Antikythera AERONET station, and the user is informed with the following message:

```
Start date is: year=2020&month=4&day=27
End date is: year2=2020&month2=4&day2=27&hour2=23
Wavelength is: 0.44
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=SIZ&year=202
0&month=4&day=27&year2=2020&month2=4&day2=27&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=VOL&year=202
0&month=4&day=27&year2=2020&month2=4&day2=27&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=ASY&year=202
0&month=4&day=27&year2=2020&month2=4&day2=27&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=RIN&year=202
0&month=4&day=27&year2=2020&month2=4&day2=27&hour2=23
Site=Antikythera_NO no data is available in this period: starting
date=27/4/2020 and ending date=27/4/2020
```

6.2 Results of 25/06/2020

The command for downloading the AERONET data from the Antikythera station for a specific date (e.g. 25/06/2020), is the following:

-s 2020-06-25 -e 2020-06-25 -wl 0.675

In this case, the start and end date are the same. The day is the 25th of June on 2020 and the wavelength of refractive index is equal to 0.675 μm. The data for this date is available as shows the message below:

```
Start date is: year=2020&month=6&day=25
End date is: year2=2020&month2=6&day2=25&hour2=23
Wavelength is: 0.675
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=SIZ&year=202
0&month=6&day=25&year2=2020&month2=6&day2=25&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=VOL&year=202
0&month=6&day=25&year2=2020&month2=6&day2=25&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=ASY&year=202
0&month=6&day=25&year2=2020&month2=6&day2=25&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=RIN&year=202
0&month=6&day=25&year2=2020&month2=6&day2=25&hour2=23
```

This certain day, based on the AERONET site, there are five different retrievals of the volume size distribution for different times during the day, as shown in the figure below: 13:31:19, 14:37:53, 15:07:07, 16:01:05, 16:27:17.

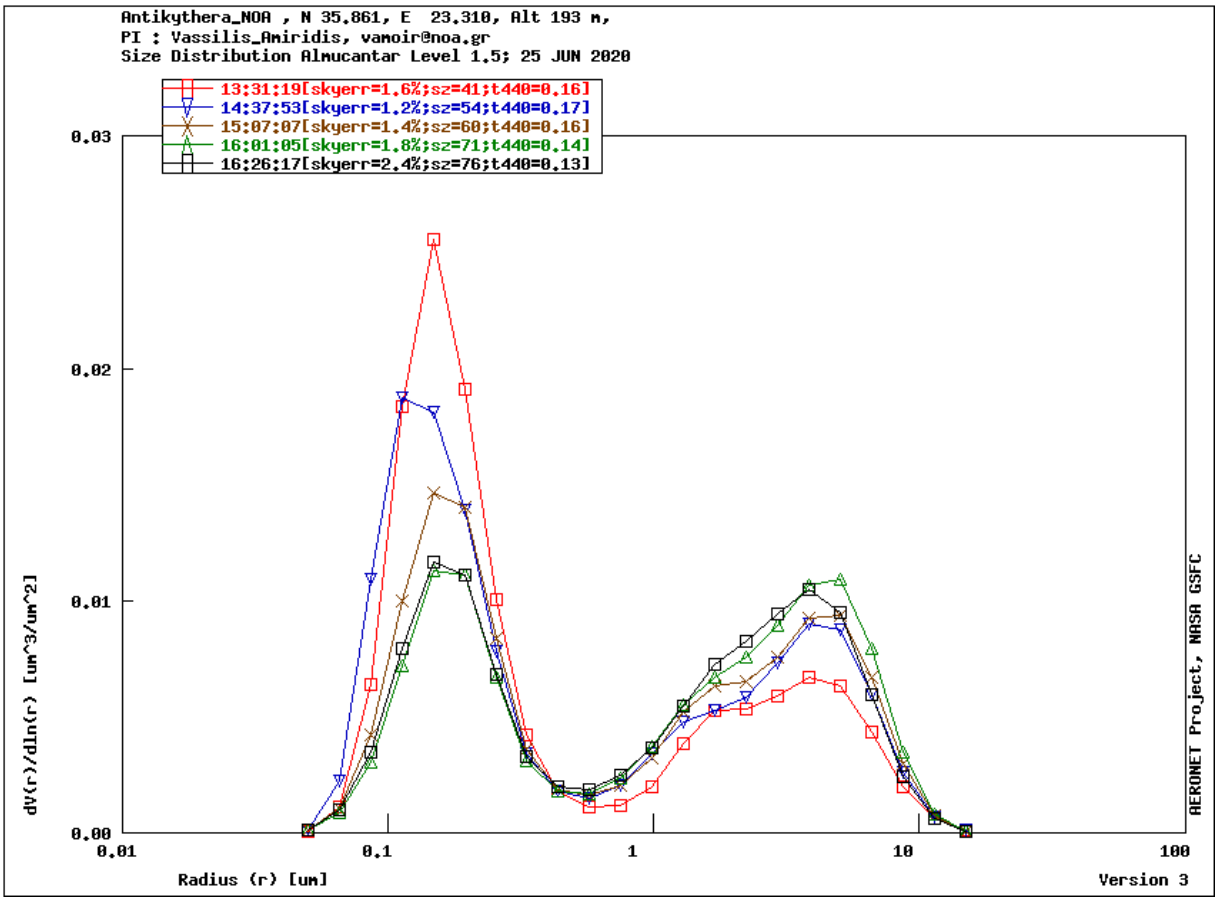


Figure 6. 1: Size distribution on 25/06/2020

(from https://aeronet.gsfc.nasa.gov)

- ***Time: 13:31:19***

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_13-31-19
SIZ_Antikythera_NO_25-06-2020_13-31-19 Sphericity_Factor 98.32 %
volc_f=0.024000 vmr_f=0.161000 std_f=0.399000 volc_c=0.012000
vmr_c=2.931000 std_c=0.668000 rirp440=1.401000 rirp675=1.420200
rirp870=1.426900 rirp1020=1.421100 riip440=0.000529 riip675=0.000535
riip870=0.000540 riip1020=0.000542
WVL 0.675
```

The AERONET product for this case provides the following aerosol properties:

- Total volume concentration of fine mode (volc_f) : 0.024000 cm^3 cm^{-3}
- Geometric mean volume radius of fine mode (vmr_f): 0.161000 μm
- Geometric standard deviation of fine mode (std_f): 0.399000

- Total volume concentration of coarse mode (volc_c): 0.012000 cm^3 cm^{-3}
- Geometric mean volume radius of coarse mode (vmr_c): 2.931000 μm
- Geometric standard deviation of coarse mode (std_c): 0.668000 μm
- Real part of the refractive index at 440 nm (rirp440): 1401000
- Real part of the refractive index at 675 nm (rirp675): 1.420200
- Real part of the refractive index at 870 nm (rirp870): 1.426900
- Real part of the refractive index at 1020 nm (rirp1020): 1.421100
- Imaginary part of the refractive index at 440 nm(riip440): 0.000529
- Imaginary part of the refractive index at 675 nm(riip675): 0.000535
- Imaginary part of the refractive index at 870 nm(riip870): 0.000540
- Imaginary part of the refractive index at 1020 nm(riip1020): 0.000542

The calculated outputs for this case are the following:

- Extinction coefficient of the ensemble (ext_all): 7.776747 Km^{-1}
- Scattering coefficient of the ensemble (sca_all): 7.727140 Km^{-1}
- Absorption coefficient of the ensemble (abs_all): 4.960720 Km^{-1}
- Backscatter coefficient of the ensemble (bac_all): 1.804243 Km^{-1} sr^{-1}
- Extinction coefficient of the fine mode (ext_f): 6.979561 Km^{-1}
- Scattering coefficient of the fine mode(sca_f): 6.946928 Km^{-1}
- Absorption coefficient of the fine mode (abs_f): 3.263332 Km^{-1}
- Backscatter coefficient of the fine mode (bac_f): 1.3275595 Km^{-1} sr^{-1}
- Extinction coefficient of the coarse mode (ext_c): 7.971864 Km^{-1}
- Scattering coefficient of the coarse mode (sca_c): 7.802126 Km^{-1}
- Absorption coefficient of the coarse mode (abs_c): 1.697387 Km^{-1}
- Backscatter coefficient of the coarse mode (bac_c): 4.286483 Km^{-1} sr^{-1}

```
ext_all=7.776747843882811e-05 sca_all=7.7271406364289e-05
abs_all=4.960720745391012e-07 bac_all=1.8042436765502466e-06
ext_f=6.979561354539787e-05 sca_f=6.946928026646682e-05
abs_f=3.2633327893104714e-07 bac_f=1.3755952770461018e-06
ext_c=7.971864893430241e-06 sca_c=7.802126097822187e-06
abs_c=1.697387956080541e-07 bac_c=4.2864839950414494e-07
```

Figure 6.2 shows the volume size distributions acquired from the AERONET product and calculated as an intermediate step in the code. The blue line ("AERONET Data") shows the volume size distribution acquired from the AERONET product. The orange line ("VSD_AER_calc") is the volume size distribution that is calculated as shown in Eq. 5.1, using volc_f, vmr_f, std_f, volc_c, vmr_c, std_c, provided by the AERONET product. The green line ("VSD_AER_calc_new"), reproduces the AERONET volume size distribution, using the calculated parameters of the number size distribution. The pink line ("Fine - Coarse Mode"), presents the volume size distribution for fine and coarse mode used in the Mie code calculations.

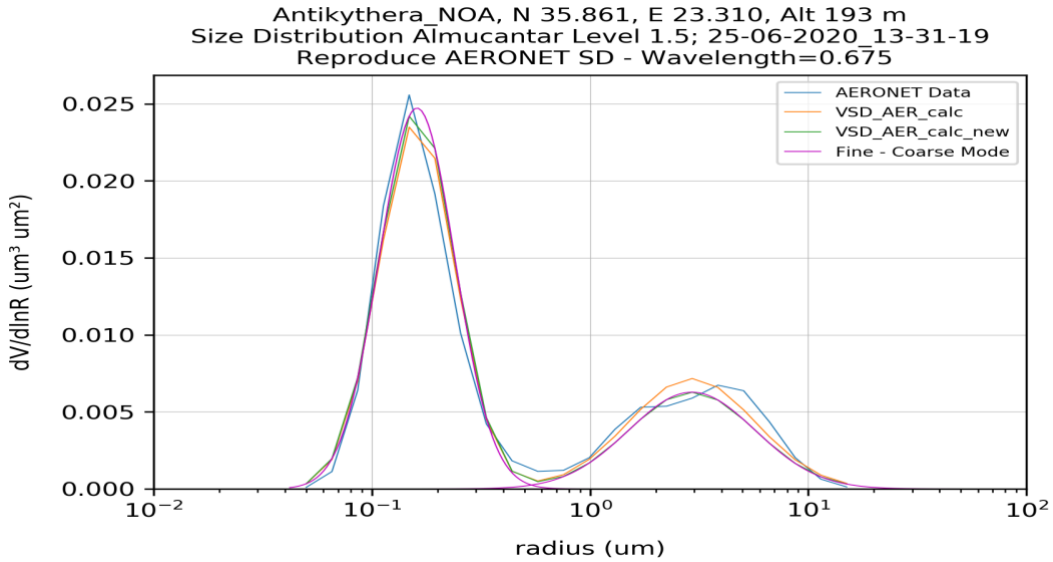


Figure 6. 2: Plot of Volume Size Distribution (25-06-2020, 13:31:19, wavelenght:0.675)

- ***Time: 14:37:53***

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_14-37-53
SIZ_Antikythera_NO_25-06-2020_14-37-53 Sphericity_Factor 97.90 %
Sphericity_Factor lower than 98 % therefore no spherical particle
```

Because the sphericity factor is lower than 98%, there aren't spherical particles and the Mie code calculations cannot be executed and for this reason, the fourth line is missing.

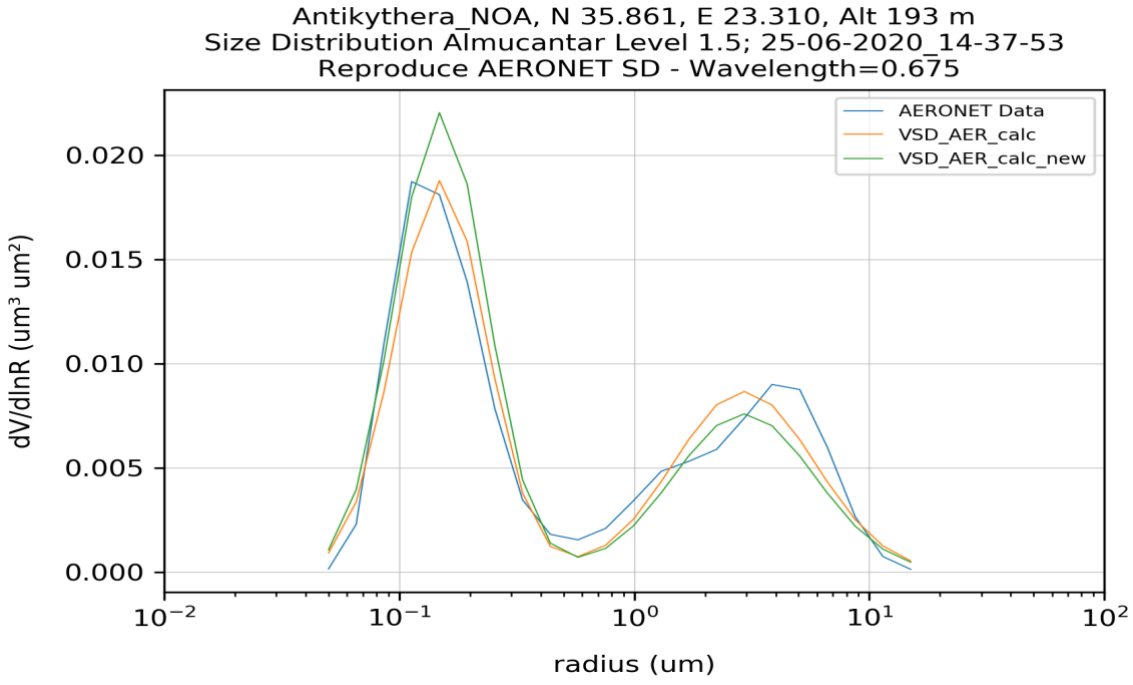


Figure 6. 3: Plot of Volume Size Distribution (25-06-2020, 14:37:53, wavelenght:0.675)

- ***Time: 15:07:07***

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_15-07-07
SIZ_Antikythera_NO_25-06-2020_15-07-07 Sphericity_Factor 99.00 %
volc_f=0.016000 vmr_f=0.171000 std_f=0.439000 volc_c=0.017000
vmr_c=2.956000 std_c=0.688000 rirp440=1.503200 rirp675=1.469000
rirp870=1.459000 rirp1020=1.447400 riip440=0.004791 riip675=0.005529
riip870=0.006318 riip1020=0.006569
WVL 0.675
ext_all=7.758613196101334e-05 sca_all=7.328363774885767e-05
abs_all=4.302494212155673e-06 bac_all=1.5221666095259921e-06
ext_f=6.622304173260436e-05 sca_f=6.374939318173908e-05
abs_f=2.473648550865276e-06 bac_f=1.1573436156250591e-06
ext_c=1.1363090228408979e-05 sca_c=9.534244567118582e-06
abs_c=1.8288456612903973e-06 bac_c=3.6482299390093305e-07
```

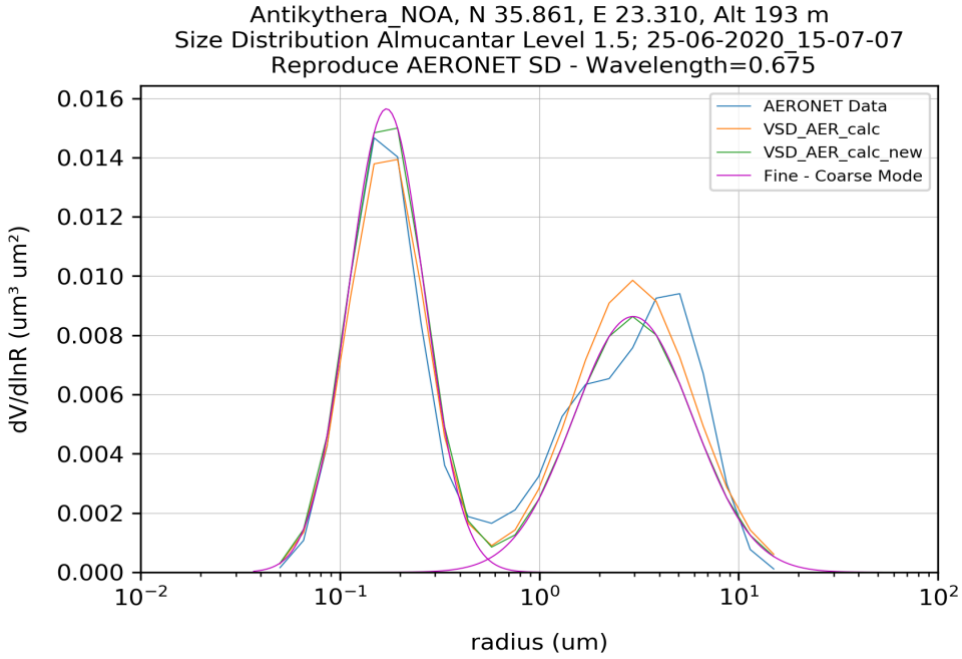


Figure 6. 4: Plot of Volume Size Distribution (25-06-2020, 15:07:07, wavelenght:0.675)

- ***Time: 16:01:05***

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_16-01-05
SIZ_Antikythera_NO_25-06-2020_16-01-05 Sphericity_Factor 98.99 %
volc_f=0.013000 vmr_f=0.175000 std_f=0.454000 volc_c=0.019000
vmr_c=3.007000 std_c=0.682000 rirp440=1.538400 rirp675=1.485000
rirp870=1.470400 rirp1020=1.453400 riip440=0.002781 riip675=0.003060
riip870=0.003475 riip1020=0.003578
WVL 0.675
ext_all=7.120117610678439e-05 sca_all=6.874877171877394e-05
abs_all=2.452404388010443e-06 bac_all=1.5724532356237454e-06
ext_f=5.888891097349746e-05 sca_f=5.7726741880013384e-05
abs_f=1.1621690934840743e-06 bac_f=1.0496386591736407e-06
ext_c=1.2312265133286928e-05 sca_c=1.1022029838760559e-05
abs_c=1.2902352945263686e-06 bac_c=5.228145764501048e-07
```

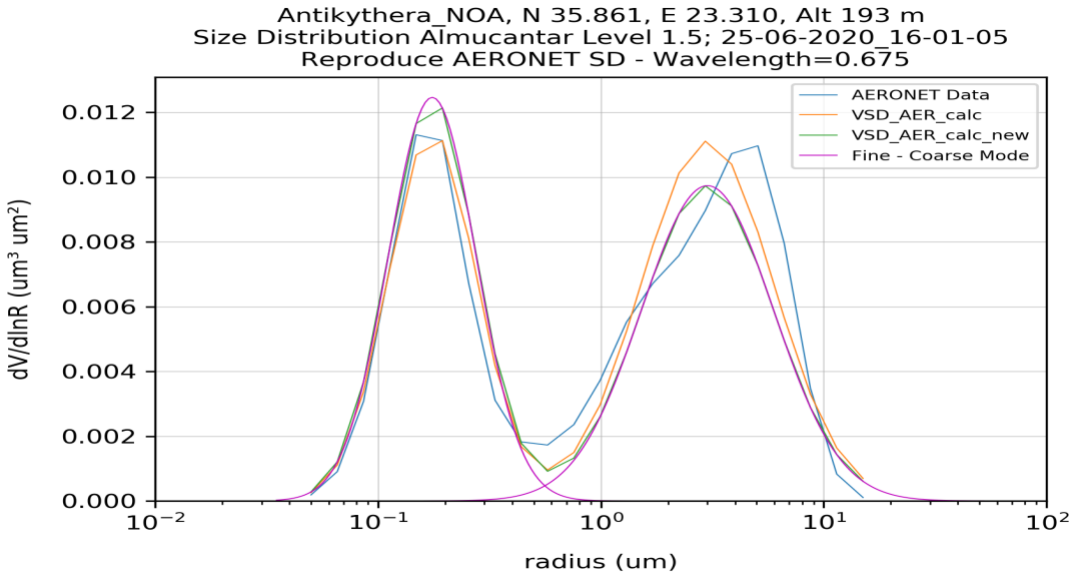


Figure 6. 5: Plot of Volume Size Distribution (25-06-2020, 16:01:05, wavelenght:0.675)

- ***Time: 16:26:17***

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_16-26-17
SIZ_Antikythera_NO_25-06-2020_16-26-17 Sphericity_Factor 99.00 %
volc_f=0.013000 vmr_f=0.174000 std_f=0.463000 volc_c=0.018000
vmr_c=2.809000 std_c=0.667000 rirp440=1.514600 rirp675=1.463200
rirp870=1.448700 rirp1020=1.440500 riip440=0.004401 riip675=0.004805
riip870=0.005423 riip1020=0.005587
WVL 0.675
ext_all=6.763934233292304e-05 sca_all=6.406750552704024e-05
abs_all=3.5718368058828013e-06 bac_all=1.3430572357717549e-06
ext_f=5.5193727998351284e-05 sca_f=5.338587021757837e-05
abs_f=1.8078577807729165e-06 bac_f=9.605253352005362e-07
ext_c=1.2445614334571765e-05 sca_c=1.068163530946188e-05
abs_c=1.7639790251098847e-06 bac_c=3.8253190057121 86e-07
```

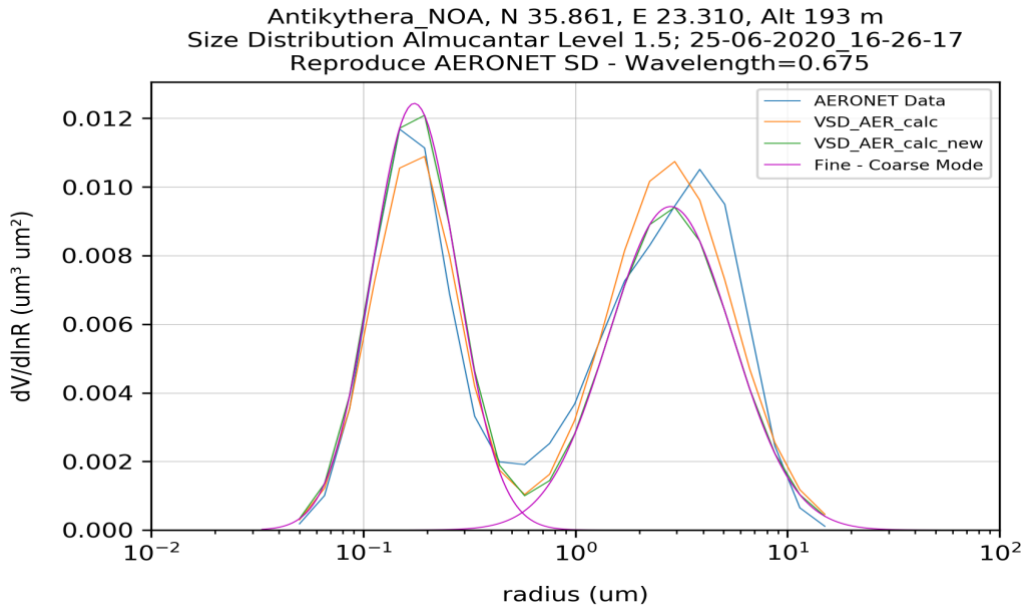


Figure 6. 6: Plot of Volume Size Distribution (25-06-2020, 16:26:17, wavelenght:0.675)

6.3 Results of 25/06/2020 – 26/06/2020

The user can choose more than one day. An example for the command for the period of 25/06/2020 – 26/06/2020 and refractive index of 0.675 μm is given below:

-s 2020-06-25 -e 2020-06-26 -wl 0.675

The data for these two days is available, as shown in the message below:

```
Start date is: year=2020&month=6&day=25
End date is: year2=2020&month2=6&day2=26&hour2=23
Wavelength is: 0.675
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=SIZ&year=2020&month=6&day=25&year2=2020&month2=6&da
y2=26&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=VOL&year=2020&month=6&day=25&year2=2020&month2=6&da
y2=26&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=ASY&year=2020&month=6&day=25&year2=2020&month2=6&da
y2=26&hour2=23
https://aeronet.gsfc.nasa.gov/cgi-bin/print_web_data_inv_v3?
site=Antikythera_NOA&AVG=10&ALM15=1&if_no_html=1&product=RIN&year=2020&month=6&day=25&year2=2020&month2=6&da
y2=26&hour2=23

Creating plot for SIZ_Antikythera_NO_25-06-2020_13-31-19
SIZ_Antikythera_NO_25-06-2020_13-31-19 Sphericity_Factor 98.32 %
volc_f=0.024000 vmr_f=0.161000 std_f=0.399000 volc_c=0.012000 vmr_c=2.931000 std_c=0.668000 rirp440=1.401000
rirp675=1.420200 rirp870=1.426900 rirp1020=1.421100 riip440=0.000529 riip675=0.000535 riip870=0.000540
riip1020=0.000542
WVL 0.675
ext_all=7.776747843882811e-05 sca_all=7.7271406364289e-05 abs_all=4.960720745391012e-07
bac_all=1.8042436765502466e-06 ext_f=6.979561354539787e-05 sca_f=6.946928026646682e-05
abs_f=3.2633327893104714e-07 bac_f=1.3755952770461018e-06 ext_c=7.971864893430241e-06
sca_c=7.802126097822187e-06 abs_c=1.697387956080541e-07 bac_c=4.2864839950414494e-07

Creating plot for SIZ_Antikythera_NO_25-06-2020_14-37-53
SIZ_Antikythera_NO_25-06-2020_14-37-53 Sphericity_Factor 97.90 %
Sphericity_Factor lower than 98 % therefore no spherical particle

Creating plot for SIZ_Antikythera_NO_25-06-2020_15-07-07
SIZ_Antikythera_NO_25-06-2020_15-07-07 Sphericity_Factor 99.00 %
volc_f=0.016000 vmr_f=0.171000 std_f=0.439000 volc_c=0.017000 vmr_c=2.956000 std_c=0.688000 rirp440=1.503200
rirp675=1.469000 rirp870=1.459000 rirp1020=1.447400 riip440=0.004791 riip675=0.005529 riip870=0.006318
riip1020=0.006569
WVL 0.675
ext_all=7.758613196101334e-05 sca_all=7.32836774885767e-05 abs_all=4.302494212155673e-06
bac_all=1.5221666095259921e-06 ext_f=6.622304173260436e-05 sca_f=6.374939318173908e-05
abs_f=2.473648550865276e-06 bac_f=1.1573436156250591e-06 ext_c=1.1363090228408979e-05
sca_c=9.534244567118582e-06 abs_c=1.8288456612903973e-06 bac_c=3.6482299390093305e-07
```

```
Creating plot for SIZ_Antikythera_NO_25-06-2020_16-01-05
SIZ_Antikythera_NO_25-06-2020_16-01-05 Sphericity_Factor 98.99 %
volc_f=0.013000 vmr_f=0.175000 std_f=0.454000 volc_c=0.019000 vmr_c=3.007000 std_c=0.682000 rirp440=1.538400
rirp675=1.485000 rirp870=1.470400 rirp1020=1.453400 riip440=0.002781 riip675=0.003060 riip870=0.003475
riip1020=0.003578
WVL 0.675
ext_all=7.120117610678439e-05 sca_all=6.874877171877394e-05 abs_all=2.452404388010443e-06
bac_all=1.5724532356237454e-06 ext_f=5.888891097349746e-05 sca_f=5.7726741880013384e-05
abs_f=1.1621690934840743e-06 bac_f=1.0496386591736407e-06 ext_c=1.2312265133286928e-05
sca_c=1.1022029838760559e-05 abs_c=1.2902352945263686e-06 bac_c=5.228145764501048e-07

Creating plot for SIZ_Antikythera_NO_25-06-2020_16-26-17
SIZ_Antikythera_NO_25-06-2020_16-26-17 Sphericity_Factor 99.00 %
volc_f=0.013000 vmr_f=0.174000 std_f=0.463000 volc_c=0.018000 vmr_c=2.809000 std_c=0.667000 rirp440=1.514600
rirp675=1.463200 rirp870=1.448700 rirp1020=1.440500 riip440=0.004401 riip675=0.004805 riip870=0.005423
riip1020=0.005587
WVL 0.675
ext_all=6.763934233292304e-05 sca_all=6.406750552704024e-05 abs_all=3.5718368058828013e-06
bac_all=1.3430572357717549e-06 ext_f=5.5193727998351284e-05 sca_f=5.338587021757837e-05
abs_f=1.8078577807729165e-06 bac_f=9.605253352005362e-07 ext_c=1.2445614334571765e-05
sca_c=1.068163530946188e-05 abs_c=1.7639790251098847e-06 bac_c=3.825319005712186e-07

Creating plot for SIZ_Antikythera_NO_26-06-2020_04-37-15
SIZ_Antikythera_NO_26-06-2020_04-37-15 Sphericity_Factor 98.96 %
volc_f=0.016000 vmr_f=0.158000 std_f=0.390000 volc_c=0.024000 vmr_c=2.778000 std_c=0.697000 rirp440=1.504800
rirp675=1.475900 rirp870=1.468000 rirp1020=1.459800 riip440=0.002464 riip675=0.002683 riip870=0.002971
riip1020=0.003059
WVL 0.675
ext_all=7.484414303539415e-05 sca_all=7.223542359983577e-05 abs_all=2.6087194355583747e-06
bac_all=1.916354542033806e-06 ext_f=5.755909181606792e-05 sca_f=5.644217535692827e-05
abs_f=1.1169164591396503e-06 bac_f=1.1359050762644795e-06 ext_c=1.728505121932622e-05
sca_c=1.5793248242907495e-05 abs_c=1.4918029764187244e-06 bac_c=7.804494657693266e-07

Creating plot for SIZ_Antikythera_NO_26-06-2020_05-04-04
SIZ_Antikythera_NO_26-06-2020_05-04-04 Sphericity_Factor 98.99 %
volc_f=0.017000 vmr_f=0.151000 std_f=0.390000 volc_c=0.024000 vmr_c=2.889000 std_c=0.704000 rirp440=1.486000
rirp675=1.472000 rirp870=1.464200 rirp1020=1.459400 riip440=0.003523 riip675=0.003825 riip870=0.004149
riip1020=0.004220
WVL 0.675
ext_all=7.507593505225148e-05 sca_all=7.141131552228819e-05 abs_all=3.6646195299633014e-06
bac_all=1.8541108767245855e-06 ext_f=5.837692413088416e-05 sca_f=5.6680525077634854e-05
abs_f=1.6963990532493058e-06 bac_f=1.2183076259536622e-06 ext_c=1.6699010921367323e-05
sca_c=1.4730790444653328e-05 abs_c=1.9682204767139956e-06 bac_c=6.358032507709233e-07

Creating plot for SIZ_Antikythera_NO_26-06-2020_05-55-58
SIZ_Antikythera_NO_26-06-2020_05-55-58 Sphericity_Factor 98.43 %
volc_f=0.019000 vmr_f=0.147000 std_f=0.383000 volc_c=0.022000 vmr_c=2.828000 std_c=0.703000 rirp440=1.461500
rirp675=1.460500 rirp870=1.461300 rirp1020=1.459400 riip440=0.003163 riip675=0.003730 riip870=0.004130
riip1020=0.004234
WVL 0.675
ext_all=7.544934264573564e-05 sca_all=7.186076177897954e-05 abs_all=3.5885808667561028e-06
bac_all=1.8066144893982054e-06 ext_f=5.977675510763713e-05 sca_f=5.7961100133644855e-05
abs_f=1.815654973992274e-06 bac_f=1.3087634752407626e-06 ext_c=1.5672587538098514e-05
sca_c=1.3899661645334685e-05 abs_c=1.7729258927638288e-06 bac_c=4.978510141574428e-07

Creating plot for SIZ_Antikythera_NO_26-06-2020_13-31-35
SIZ_Antikythera_NO_26-06-2020_13-31-35 Sphericity_Factor 98.98 %
volc_f=0.024000 vmr_f=0.167000 std_f=0.345000 volc_c=0.034000 vmr_c=2.692000 std_c=0.680000 rirp440=1.422100
rirp675=1.429800 rirp870=1.433700 rirp1020=1.427900 riip440=0.000500 riip675=0.000500 riip870=0.000501
riip1020=0.000503
WVL 0.675
ext_all=9.27935415744438e-05 sca_all=9.204201884326012e-05 abs_all=7.515227311836765e-07
bac_all=2.4826519996124103e-06 ext_f=6.754448272580867e-05 sca_f=6.726660948745701e-05
abs_f=2.7787323835166185e-07 bac_f=1.2524796505222351e-06 ext_c=2.5249058848635127e-05
sca_c=2.4775409355803113e-05 abs_c=4.736494928320146e-07 bac_c=1.2301723490901752e-06

Creating plot for SIZ_Antikythera_NO_26-06-2020_14-37-59
SIZ_Antikythera_NO_26-06-2020_14-37-59 Sphericity_Factor 99.00 %
volc_f=0.029000 vmr_f=0.188000 std_f=0.406000 volc_c=0.044000 vmr_c=2.718000 std_c=0.608000 rirp440=1.427800
rirp675=1.420900 rirp870=1.427300 rirp1020=1.423900 riip440=0.000500 riip675=0.000500 riip870=0.000500
riip1020=0.000500
WVL 0.675
ext_all=0.00012746925109318504 sca_all=0.0001265045985726131 abs_all=9.64652520571948e-07
bac_all=3.1564934758494135e-06 ext_f=9.719129332813671e-05 sca_f=9.682507280057934e-05
abs_f=3.662205275573727e-07 bac_f=1.5745106965451538e-06 ext_c=3.0277957765048343e-05
sca_c=2.9679525772033768e-05 abs_c=5.984319930145753e-07 bac_c=1.5819827793042595e-06

Creating plot for SIZ_Antikythera_NO_26-06-2020_15-07-12
SIZ_Antikythera_NO_26-06-2020_15-07-12 Sphericity_Factor 99.00 %
volc_f=0.029000 vmr_f=0.164000 std_f=0.365000 volc_c=0.039000 vmr_c=2.796000 std_c=0.709000 rirp440=1.424400
rirp675=1.423900 rirp870=1.426500 rirp1020=1.422500 riip440=0.002423 riip675=0.002829 riip870=0.003447
riip1020=0.003591
WVL 0.675
ext_all=0.00011029371486353277 sca_all=0.00010584194786515256 abs_all=4.451766998380212e-06
bac_all=2.478160504874499e-06 ext_f=8.172379959128289e-05 sca_f=7.978599349164851e-05
abs_f=1.9378060996343816e-06 bac_f=1.5250179143845985e-06 ext_c=2.8569915272249877e-05
sca_c=2.6055954373504047e-05 abs_c=2.5139608987458305e-06 bac_c=9.531425904899007e-07
```

```
Creating plot for SIZ_Antikythera_NO_26-06-2020_16-01-10
SIZ_Antikythera_NO_26-06-2020_16-01-10 Sphericity_Factor 98.99 %
volc_f=0.024000 vmr_f=0.178000 std_f=0.363000 volc_c=0.037000 vmr_c=2.624000 std_c=0.685000 rirp440=1.474600
rirp675=1.448700 rirp870=1.447200 rirp1020=1.432200 riip440=0.001059 riip675=0.001136 riip870=0.001275
riip1020=0.001310
WVL 0.675
ext_all=0.0001093170717184571 sca_all=0.00010756601149391038 abs_all=1.7510602245467214e-06
bac_all=2.734304563259277e-06 ext_f=8.107341114930796e-05 sca_f=8.041242647623265e-05
abs_f=6.609846730753062e-07 bac_f=1.369782444006112e-06 ext_c=2.8243660569149138e-05
sca_c=2.7153585017677723e-05 abs_c=1.0900755514714152e-06 bac_c=1.3645221192531651e-06

Creating plot for SIZ_Antikythera_NO_26-06-2020_16-27-12
SIZ_Antikythera_NO_26-06-2020_16-27-12 Sphericity_Factor 98.96 %
volc_f=0.024000 vmr_f=0.185000 std_f=0.379000 volc_c=0.041000 vmr_c=2.624000 std_c=0.676000 rirp440=1.478300
rirp675=1.440200 rirp870=1.434700 rirp1020=1.421300 riip440=0.001664 riip675=0.001784 riip870=0.001934
riip1020=0.001959
WVL 0.675
ext_all=0.0001141179281089906 8 sca_all=0.00011126049356203058 abs_all=2.857434546960115e-06
bac_all=2.4932970485117576e-06 ext_f=8.312639074396885e-05 sca_f=8.207317937472975e-05
abs_f=1.053211369239106e-06 bac_f=1.3396367493216068e-06 ext_c=3.099153736502183e-05
sca_c=2.9187314187300824e-05 abs_c=1.804223177721009e-06 bac_c=1.153660299190151e-06
```

The results for the day of 25/06/2020 were presented in section 6.2.

- ***Day: 26/06/2020, Time: 04:37:15***

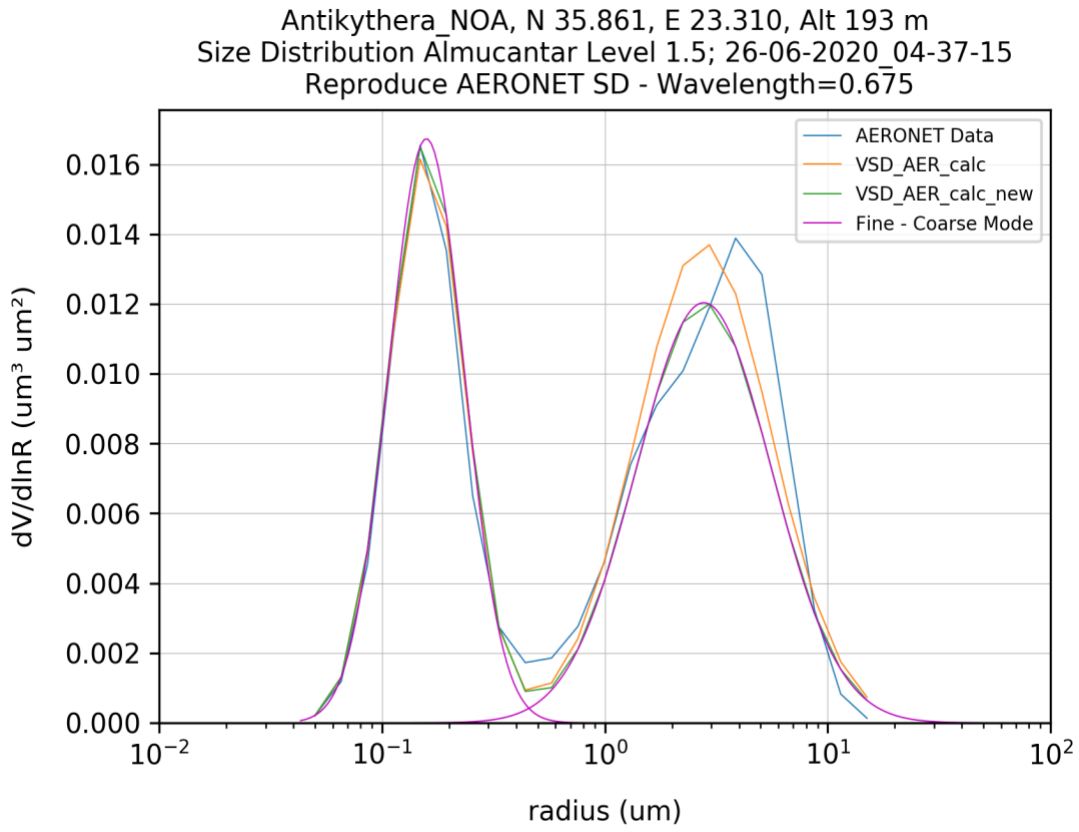


Figure 6. 7: Plot of Volume Size Distribution (26-06-2020, 04:37:15, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 05:04:04***

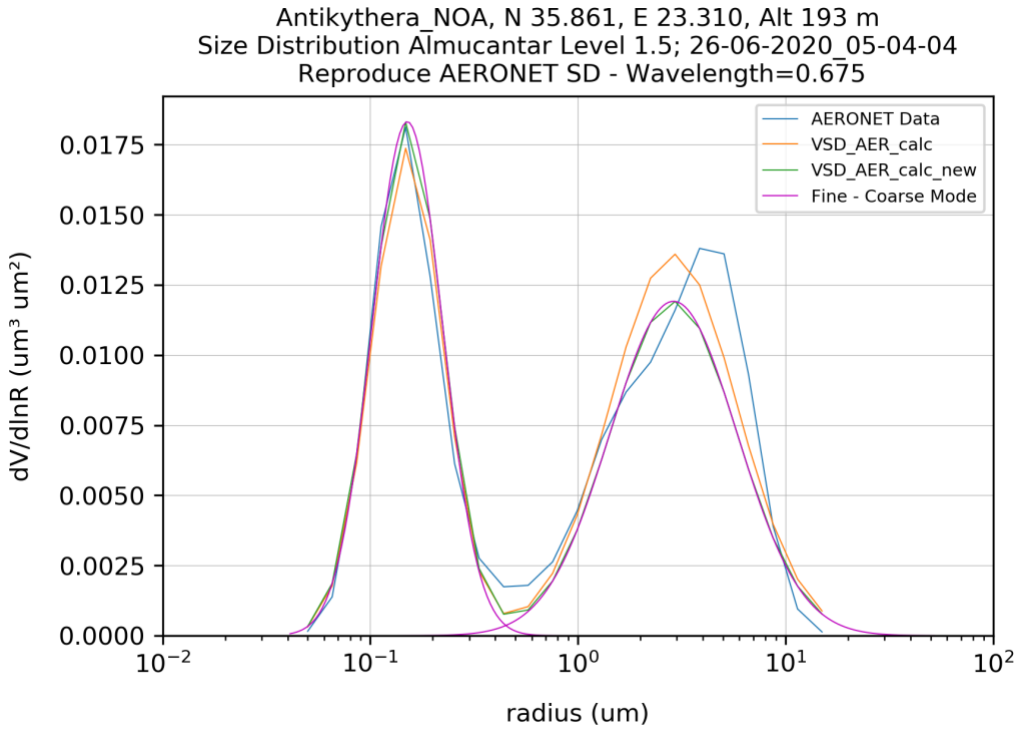


Figure 6. 8: Plot of Volume Size Distribution (26-06-2020, 05:04:04, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 05:55:58***

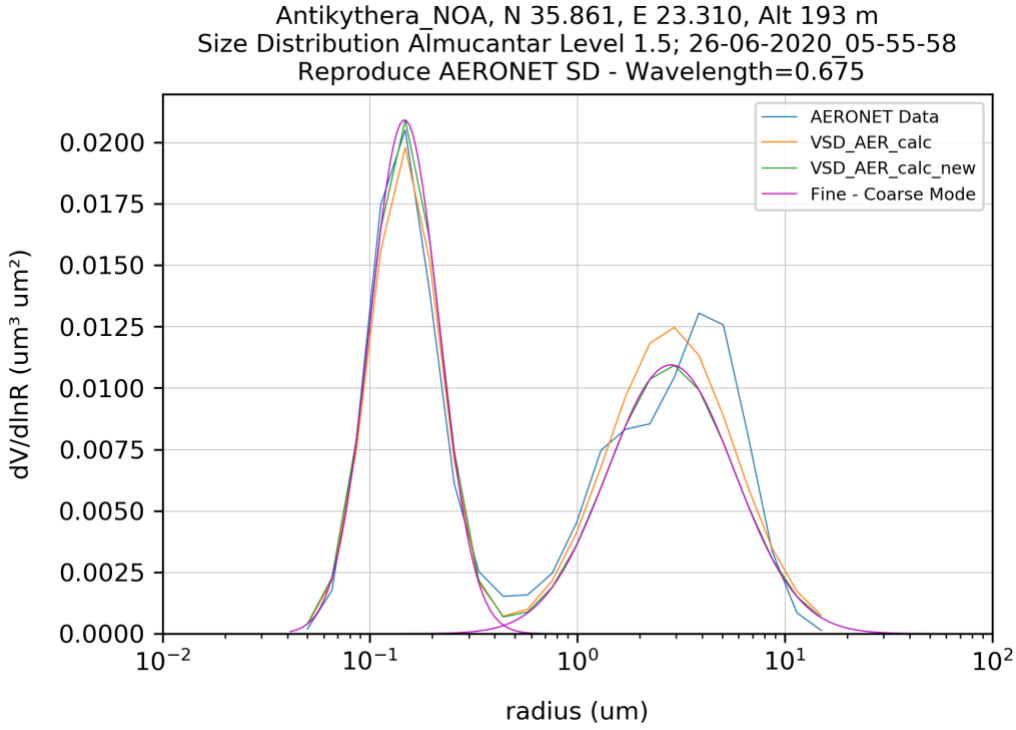


Figure 6. 9: Plot of Volume Size Distribution (26-06-2020, 05:55:58, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 13:31:35***

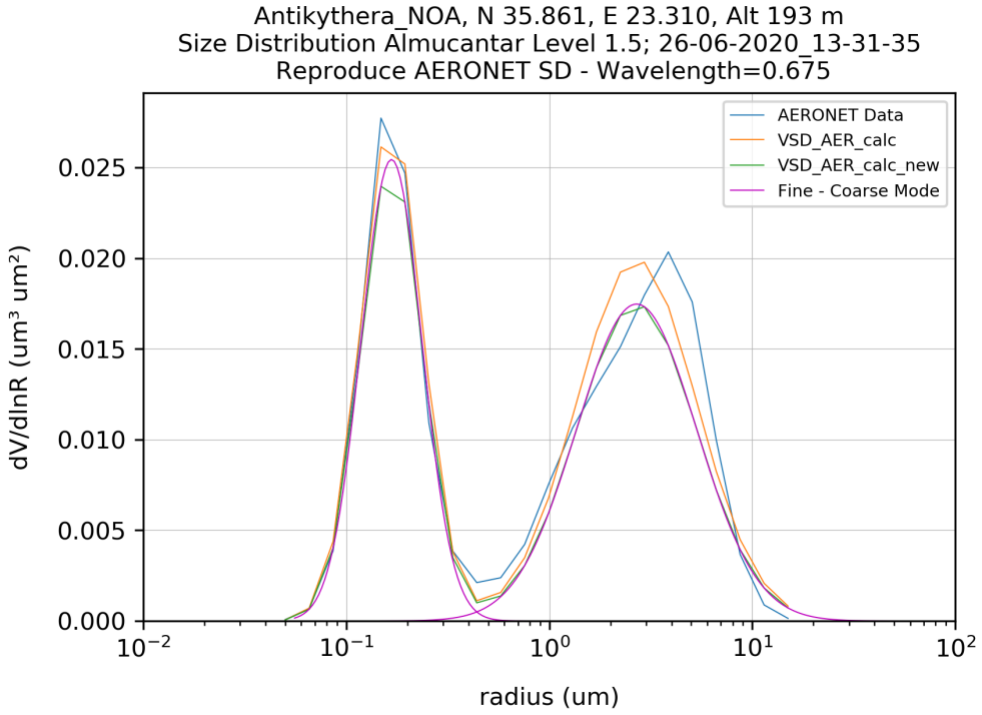


Figure 6. 10: Plot of Volume Size Distribution (26-06-2020, 13:31:35, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 14:37:59***

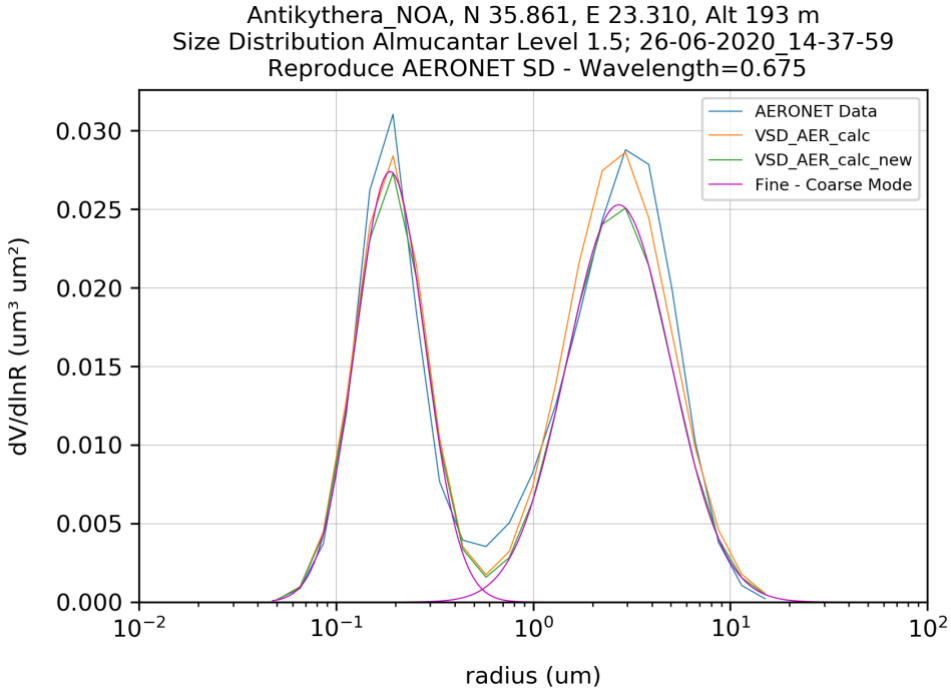


Figure 6. 11: Plot of Volume Size Distribution (26-06-2020, 14:37:59, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 15:07:12***

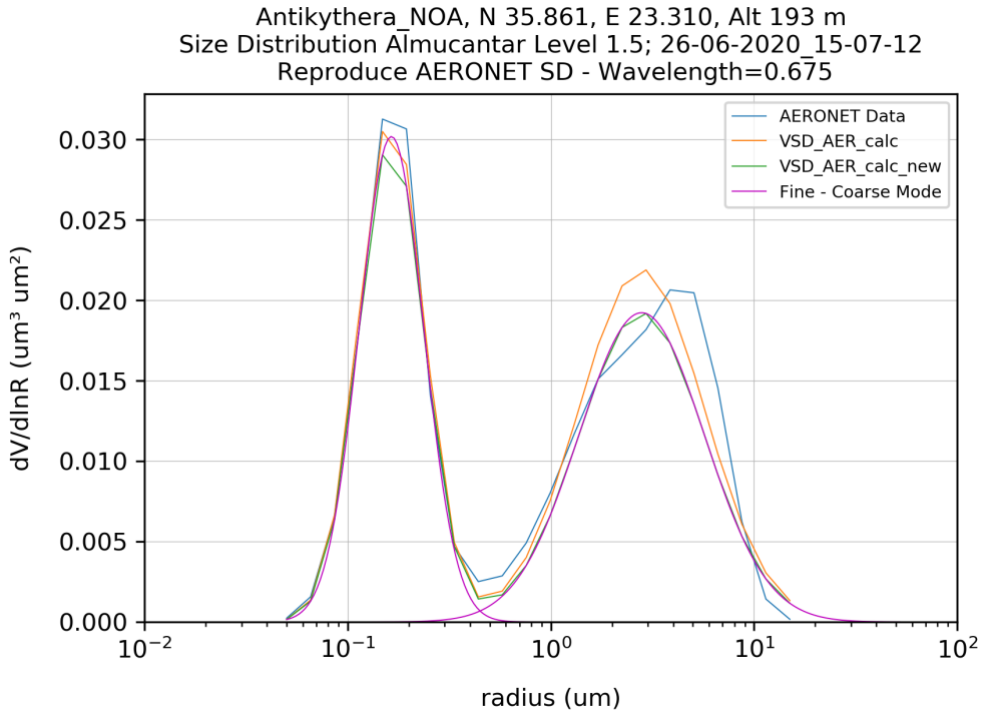


Figure 6. 12: Plot of Volume Size Distribution (26-06-2020, 15:07:12, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 16:01:10***

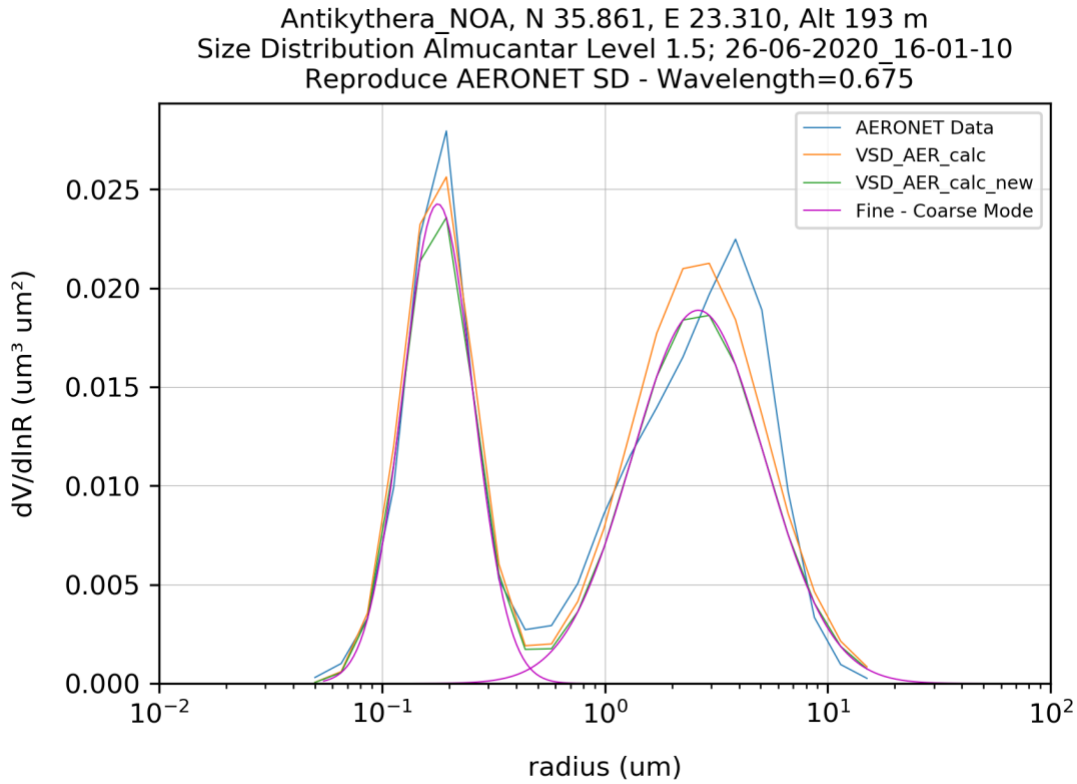


Figure 6. 13: Plot of Volume Size Distribution (26-06-2020, 16:01:10, wavelenght:0.675)

- ***Day: 26/06/2020, Time: 16:27:12***

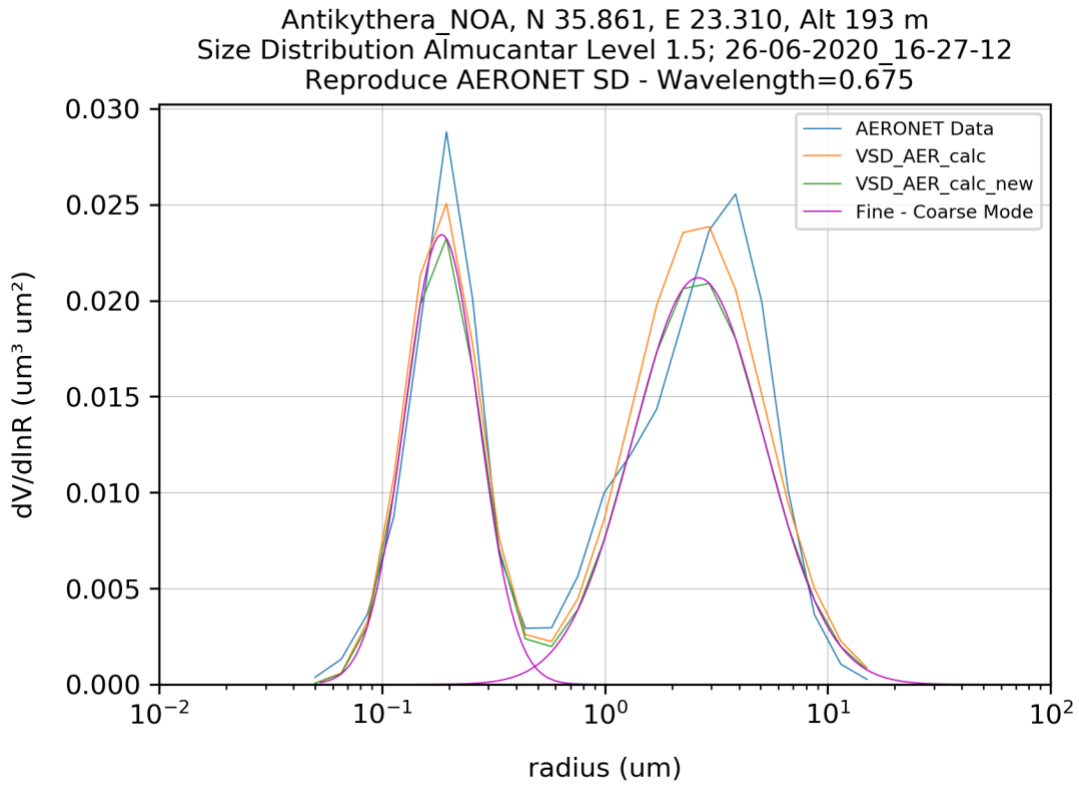


Figure 6. 14: Plot of Volume Size Distribution (26-06-2020, 16:27:12, wavelenght:0.675)

6.4 Discussion - Future perspectives

The results of this study can be used in the future in synergistic retrieval algorithms with the measurements of PollyXT lidar. The sun-photometer measurements provide information of the columnar properties of atmospheric aerosols, whereas the lidar measurements provide information about the vertical profiles of the aerosols. The combination of these two methods is a powerful tool in the hands of the scientific community.

Bibliography

Bosenberg J, Ansmann A, Baldasano J, Balis D, B ckmann ö C, Calpini B, et al. EARLINET: A European Aerosol Research Lidar Network. Advances in Laser Remote sensing, Selected papers 20th Int. Laser Radar Conference (ILRC), Vichi, France. 2000;155-158.

Dubovik, O. and King, M.: A flexible inversion algorithm for retrieval of aerosol optical properties from Sun and sky radiance measurements, J. Geophys. Res., 105, 20673–20696, doi:10.1029/2000JD900282, 2000.

Dubovik, O., Sinyuk, A., Lapyonok, T., Holben, B. N., Mishchenko, M., Yang, P., Eck, T. F., Volten, H., Munoz, O., Veihelmann, B., van der Zande, W. J., Leon, J.-F., Sorokin, M., and Slutsker, I.:Application of spheroid models to account for aerosol particlenonsphericity in remote sensing of desert dust, J. Geophys. Res.,111, D11208, doi:10.1029/2005JD006619, 2006.

Holben BN, Eck TF, Slutsker I, Tanre D, Buis JP, Setzer A, et al. AERONET-A federated instrument network and data archive for aerosol characterization. Remote Sens. Environ.1998;66:1-16.

IPCC 2013: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P. M.: Cambridge University Press, Cambridge, UK and New York, NY, USA, 1535 pp., available at: http://www.ipcc.ch/report/ar5/wg1/ (last access: 5 August 2017), 2013.

Mie, G.: Beiträge zur Optik trüber Medien, speziell kolloidaler Met-allösungen, Ann. Phys., 25, 377–445, 1908.

Haldoupis, C.: Introduction to Atmospheric Physics, Greek Academic Electronic Books and Aids, 2015.

Gonzalez, R. C., Woods, R. E.:Digital Image Processing, Tziolas publications, 2010.

Mishchenko, M., and Travis, L.: Light scattering by polydispersions of randomly oriented spheroids with sizes comparable to wavelengths of observation. Applied Optics, Vol 33, No 30, 7206, 1994.

Spyrou, C., Kallos, G., Mitsakou, C., Athanasiadis, P., Kalogeri, C., and Iacono, M. J.: Modeling the radiative effects of desert duston weather and regional climate, Atmos. Chem. Phys., 13, 5489–5504, doi:10.5194/acp-13-5489-2013, 2013.

Kokhanovsky, A.: Light Scattering Media Optics- problems and Solutions (Third Edition), Springer, 2004.

Hansen, J., and Travis, L.: Light Scattering in Planetary Atmospheres, 1974.

Gasteiger, J. and Wiegner, M.: MOPSMAP v1.0: a versatile tool for the modeling of aerosol optical properties, Geosci. Model Dev, 2739–2762, 2018.

Halliday, D., Krane, S. K., and Resnick, R.: Physics. Vol I, Pneumatikos, 2009.

Toledano, C., Cachorro, V. E., Berjon, A., Frutos, A. M., Sorribas, M., Morena, B. A., and Goloub, P.: Aerosol optical depth and Angstrom exponent climatology at El Arenosillo AERONET site (Huelva, Spain), Quarterly Journal of the Royal Meteorological Society, 2007.

Gregory, L.: CIMEL Sun Photometer (CSPHOT) Handbook, U.S.: Climate Research Facility, 2011.

Barreto, A., Cuevas, E., Granados-Munoz, M., Alados-Arboledas, L., Romero, P., Grobner, J., Kouremeti, N., Almansa, A., Stone, T., Toledano, C., Roman, R., Sorokin, M., Holben, B., Canini, M. and Yela, M. : The new sun-sky-lunar CIMEL CE318-T multiband photometer- a comprehensive performance evaluation, Atmos. Meas. Tech, 2016.

Remer, L. A., Kaufman, Y. J., Holben, B. N., Thompson, A. M., and McNamara, D.: Biomass burning aerosol size distribution and modeled optical properties, Journal of Geophysical Research, Vol 103, No D24, 31,879-31,891, 1998.
Epricum, S., Dewals, B., Archambeau, P., and Pirotton, M.: Sustainable Hydraulics in the Era of Global Change, CRC Press, 2016.
Richards, K. L.: Design Engineer's Sourcebook, CRC Press, 2018.
Vacca, J. R.: Computer and Information Security Handbook (Second edition), Morgan Kaufmann, 2013.
Guieu, C., Dulac, F., Desboeufs, K., Wagener, T., Pulido-Villena,E., Grisoni, J.-M., Louis, F., Ridame, C., Blain, S., Brunet, C.,Bon Nguyen, E., Tran, S., Labiadh, M., and Dominici, J.-M.:Large clean mesocosms and simulated dust deposition: a newmethodology to investigate responses of marine oligotrophicecosystems to atmospheric inputs, Biogeosciences, 7, 2765–2784, doi:10.5194/bg-7-2765-2010, 2010.
Myers, R. L.: The basics of Physics, Greenwood Press, 2006.
Dubovik, O., Holben, B., Eck, Th. F., Smirnov, A., Kaufman, Y. J., King, M. D., Tanre, D., and Slutsker, I.: Variability of Absorption and Optical Properties of Key Aerosol Types Observed in Worldwide Locations, Journal of the atmospheric sciences, Vol 59, 2002.
Kovalenko, S. A.: Descartes- Snell lao of refraction with absorption, Semiconductor, Physics, Quantum Electronics & Optoelectronics, Vol 4 No 3, pp 214 – 218, 2001.
Tsekeri, A., Lopatin, A., Amiridis, V., Marinou, E., Igloffstein, J., Siomos, N., Solomos, S., Kokkalis, P., Engelmann, R., Baars, H., Gratsea, M., Raptis, P., Binietoglou, I., Mihalopoulos, N., Kalivitis, N., Kouvarakis, G., Bartsotas, N., Kallos, G., Basart, S., Schuettemeyer, D., Wandinger, U., Ansmann, A., Chaikovsky, A., and Dubovic O.: GARRLiC and LIRIC: strengths and limitations for the characterization of dust and marine particles along with their mixtures, Atmos. Meas. Tech., 4995–5016, 2017.
Dubovik, O., Smirnov, A., Holben, B. N., King, M. D., Kaufman, Y. J., and Slutsker, I.: Accuracy assessmentso f aerosol optical properties retrieved from Aerosol Robotic Network (AERONET) Sun and sky radiance measurem easurements, Journal of Geophysical Research, Vol 105 No. D8, 9791-9806, 2000.
Fleagle, R. , and Businger, J.: An introduction to atmospheric Physics - Second edition, Academic Press, 1980.
Loeb, N. G., Thorsen, T. J., Norris, J. R., Wang, H., and Su, W.: Changes in Earth's Energy Budget during and after the "Pause" in Global Warming: An Observational Perspective, Climate, 6, 62, doi:10.3390/cli6030062, 2018.
WHO: Review of evidence on health aspects of air pollution -REVIHAAP Project, Technical Report, World Health Organization, Denmark, 2268-3798, 2013.
Kudo, R., Nishizawa, T., and Aoyagi, T.: Vertical profiles of aerosol optical properties and the solar heating rate estimated by combining sky radiometer and lidar measurements, Atmos. Meas. Tech, 3223–3243, 2016.
Zhao, G., Zhao, C., Kuang, Y., Bian, Y., Tao, J., Shen, C., and Yu, Y.: Calculating the aerosol asymmetry factor based on measurements from the humidified nephelometer system, Atmos. Chem. Phys, 2018.
Guth, J., Marecal, V., Josse, B., Arteta, J., and Hamer, P.: Primary aerosol and secondary inorganic aerosol budget over the Mediterranean Basin during 2012 and 2013, Atmos. Chem. Phys., 18, 4911-4934, 2018.
Hergert, W., and Wriedt, T.: Mie Theory 1908-2008, Present developments and Interdisciplinary aspects of light scattering, Universität Bremen, 2008.
Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K.,Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P. M.:Climate change 2013: The physical science basis, Intergovern-

mental Panel on Climate Change, Working Group I Contributionto the IPCC Fifth Assessment Report (AR5), Cambridge Univer-sity Press, New York, 2013.
Nabat, P., Somot, S., Mallet, M., Michou, M., Sevault, F., Dri-ouech, F., Meloni, D., di Sarra, A., Di Biagio, C., Formenti, P.,Sicard, M., Léon, J.-F., and Bouin, M.-N.: Dust aerosol radiativeeffects during summer 2012 simulated with a coupled regionalaerosol–atmosphere–ocean model over the Mediterranean, At-mos. Chem. Phys., 15, 3303–3326, https://doi.org/10.5194/acp-15-3303-2015, 2015.
Kant, S., Panda, J., Gautam, R., Wang, P. K., and Singh, S. P.: Significance of Aerosols influencing Weather and Climate over Indian Region, International Journal of Earth and Atmospheric Science, Volume 04, Issue 01, pp 01 – 20, January- March, 2017.
Giles, D. M., Holben, B. N., Eck, T. F., Sinyuk, A., Smirnov, A., Slutsker, I., Dickerson, R. R., Thompson, A. M., and Schafer, J. S.: An analysis of AERONET aerosol absorption properties and classifications representative of aerosol source regions, Journal of Geophysical Research, Vol. 117, D17203, doi: 10.1029/2012JD018127, 2012.
Burton, S. P., Ferrare, R. A., Hostetler, C. A., Hair, J. W., Rogers, R. R.., Obland, M. D., Butler, C. F., Cook, A. L., Harper, D. B., and Froyd, K. D.: Aerosol classification using airborne High Spectral Resolution Lidar measurements – methodology and examples, Atmos. Meas. Tech, 73–98, 2011.
Tsekeri, A., Amiridis, V., Marenco, F., Nenes, A., Marinou, E., Solomos, S., Rosenberg, P., Trembath, J., Nott, G. J., Allan, J., Le Breton, M., Bacak, A., Coe, H., Percival, C., and Mihalopoulos, N.: Profiling aerosol optical, microphysical and hygroscopic properties in ambient conditions by combining in situ and remote sensing, Atmos. Meas. Tech., 83–107, 2016.

Appendices

Appendix A: Mie_code.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jul  7 11:43:29 2019

@author: Anastasia Georgiou

Plot generator for Aerosol Inversions (V3)
Aeronet site: Antikythera

Use the AERONET Version 3 Web Service:
    Download Tool Print Web Data Inversion Interface
    (https://aeronet.gsfc.nasa.gov/print_web_data_help_v3_inv.html)
"""
import argparse
import datetime
import csv
import requests
import matplotlib.pyplot as plt
import numpy as np
from math import pi, log, exp, sqrt
from my_mie import mie_code

#
# Class to calculate Size Distribution from volume concentration,
# volume mean radius and standard deviation
#
class Size_Distribution:

    # Create the class using volume concentration,
    # volume mean radius and standard deviation
    def __init__(self, volc_f, vmr_f, std_f, volc_c, vmr_c, std_c ):
        self.volc_f = volc_f
        self.vmr_f = vmr_f
        self.std_f = std_f
        self.volc_c = volc_c
        self.vmr_c = vmr_c
        self.std_c = std_c

    def print(self):
        print ("volc_f=" +  str(volc_f) + " vmr_f=" + str(vmr_f) + " std_f" + str(s
td_f) + " volc_c=" + str(volc_c) + " vmr_c=" + str(vmr_c) + " std_c=" + str(std_c))


    # Calculate the size distribution of a given radius
    def calculate(self, radius):

        #
        #Split the Size Distribution formulain 4 parts:
        #
        # 1) factor 1
        # 2) exp 1
        # 3) factor 2
        # 4) exp 2

        factor1 =(volc_f/(sqrt(2*pi)*std_f))
        #print("factor1=" + str(factor1))
```

```
56.
57.         #exp1 = -((log(radius)-log(vmr_f)) * exp(2.0))/(2*(log(std_f)*exp(2.0)))
58.         exp1 = -(((log(radius)-log(vmr_f)) **2)/(2*((std_f)**2)))
59.         #print("exp1=" + str(exp1))
60.
61.         f1=factor1 * exp(exp1)
62.         #print("f1=" + str(f1))
63.
64.         factor2 = (volc_c/(sqrt(2*pi)*std_c))
65.         #print("factor2=" + str(factor2))
66.
67.         #exp2 = -((log(radius)-log(vmr_c)) * exp(2.0))/(2*(log(std_c)*exp(2.0)))
68.         exp2 = -(((log(radius)-log(vmr_c)) **2)/(2*((std_c)**2)))
69.         #print("exp2=" + str(exp2))
70.
71.         f2 = factor2 * exp(exp2)
72.         #print("f2=" + str(f2))
73.
74.         ## the calculated Distribution size is negative
75.         ## therefore add minus
76.         result = (f1 + f2)
77.
78.         return result
79.
80. # Webservice call to get the SIZ and VOL data from AERONET Web Data Service
81. def web_service_request(url,product,start_date,end_date,remove_headers_lines):
82.     data_list=[]
83.     with requests.Session() as s:
84.         url_request=url+"&"+product+"&"+start_date+"&"+end_date
85.         print(url_request)
86.         download = s.get(url_request)
87.
88.         decoded_content = download.content.decode('utf-8')
89.
90.         cr = csv.reader(decoded_content.splitlines(), delimiter=',')
91.         data_list = list(cr)
92.
93.         if len(data_list) > 1:
94.             # remove header headers
95.             del data_list[0:remove_headers_lines]
96.         else:
97.             del data_list[0]
98.     return data_list
99.
100.
101.        now = datetime.datetime.now()
102.
103.
104.        default_date=str(now.year) + "-" + str(now.month) +"-"+str(now.day);
105.
106.
107.        parser = argparse.ArgumentParser()
108.
109.        parser.add_argument("-s","--
    start_date", type=lambda s: datetime.datetime.strptime(s, '%Y-%m-
    %d'), default=default_date,
110.                            required=False,  help="Start date period of the selectio
    n data to download. Default value: " + default_date)
111.        parser.add_argument("-e","--
    end_date", type=lambda s: datetime.datetime.strptime(s, '%Y-%m-
    %d'), default=default_date,
112.                            required=False, help="Start date period of the selection
     data to download. Default value: " + default_date)
113.        parser.add_argument("-wl","--
    wave_lenght", type=float, choices=[0.44,0.675,0.87,1.02],default=0.44, help="Wavele
    nght value. Default value: 0.44")
```

```
114.
115.       args = parser.parse_args()
116.
117.       start_date="year="+str(args.start_date.year)+"&month="+str(args.start_date.m
    onth)+"&day="+str(args.start_date.day)
118.       end_date="year2="+str(+args.end_date.year)+"&month2="+str(args.end_date.mont
    h)+"&day2="+str(args.end_date.day)+"&hour2=23"
119.       wavelength=args.wave_lenght
120.
121.       print("Start date is: " + start_date)
122.       print("End date is: " + end_date)
123.       print("Wavelength is: " + str(wavelength))
124.
125.
126.       # Site
127.       site= "Antikythera_NO"
128.       site_lable="Antikythera_NOA, N 35.861, E 23.310, Alt 193 m\nSize Distributio
    n Almucantar Level 1.5; "
129.       # URL for Tool Print Web Data Inversion Interface
130.       URL_Data_Inversion="https://aeronet.gsfc.nasa.gov/cgi-
    bin/print_web_data_inv_v3?site="+ site +"A&AVG=10&ALM15=1&if_no_html=1"
131.
132.       # Product -    -    Size Distribution (SIZ)
133.       Product_SIZ="product=SIZ"
134.
135.       # Product -    Volume concentration, volume mean radius, effective radius an
    d standard deviation
136.       Product_VOL="product=VOL"
137.
138.       # Product - Asymmetry factor
139.       Product_ASY="product=ASY"
140.
141.       # Product - Refractive indicies (real and imaginary)
142.       Product_RIN="product=RIN"
143.
144.
145.       # Data SIZ,VOL,ASY
146.       data_siz_list=web_service_request(URL_Data_Inversion,Product_SIZ,start_date,
    end_date,6)
147.       data_vol_list=web_service_request(URL_Data_Inversion,Product_VOL,start_date,
    end_date,7)
148.       data_asy_list=web_service_request(URL_Data_Inversion,Product_ASY,start_date,
    end_date,7)
149.       data_rin_list=web_service_request(URL_Data_Inversion,Product_RIN,start_date,
    end_date,7)
150.
151.
152.
153.
154.       if len(data_siz_list) > 0:
155.           data_leggenda=data_siz_list[0]
156.           #Radii of the size bins of the AERONET size distribution
157.           x_np = np.array(data_leggenda[5:27])
158.           radius_AER = x_np.astype(np.float)
159.
160.           # radius_AER_dif:  Width of the size bins of the AERONET size distributi
    on
161.           radius_AER_dif = []
162.           radius_AER_dif.append(radius_AER[0] - 0)
163.           for index in range(0,len(radius_AER)-1):
164.               radius_AER_dif.append(radius_AER[index+1] - radius_AER[index])
165.
166.           # radius_AER__mul
167.           radius_AER__mul = (4*pi/3)*np.power(radius_AER,3)
168.
169.           # radius_AER__div2
```

```
170.            radius_AER__div2 = np.divide(1,radius_AER)
171.
172.            del data_siz_list[0:1]
173.            index=0
174.            for row in data_siz_list:
175.
176.                # Take the VOL of the same time and date
177.                vol=data_vol_list[index]
178.
179.                # Create the date and file name
180.                time=row[1].replace(":","-") + "_" + row[2].replace(":","-")
181.                file_name="SIZ_" + site + "_" + time
182.                print("\nCreating plot for " + file_name)
183.
184.                y_np = np.array(row[5:27])
185.                y = y_np.astype(np.float)
186.
187.                # Calculate the Size distribution from VOL
188.                volc_f = float(vol[9])
189.                vmr_f = float(vol[11])
190.                std_f = float(vol[12])
191.                volc_c = float(vol[13])
192.                vmr_c = float(vol[15])
193.                std_c = float(vol[16])
194.
195.                # Create the object to calculate the size distribution
196.                params = Size_Distribution(volc_f,vmr_f,std_f,volc_c,vmr_c,std_c)
197.
198.                # Calculate the size distribution for each radius
199.                z = []
200.                for i in range(len(radius_AER)):
201.                    z.append(params.calculate(radius_AER[i]))
202.
203.                # ------------------------------------------------------------------
    -----
204.                # Conversion of the parameters of the volume size distribution (AERO
    NET) to
205.                # the parameters of the number size distribution (used in Mie code)

206.                #
207.                # The geometric standard deviation does not change
208.                st_f=std_f
209.                st_c=std_c
210.                #
211.                # The geometric mean radius of the number size distribution is calcu
    lated
212.                # as shown in Eq. 5.2
213.                r_f=np.exp(np.log(2*vmr_f)-3*(st_f**2))/2
214.                r_c=np.exp(np.log(2*vmr_c)-3*(st_c**2))/2
215.
216.                # The total number concentration
217.                n_f0= ((radius_AER__div2)*((volc_f/(sqrt(2*pi)*std_f))*np.exp(-
    ((np.log(radius_AER)-np.log(vmr_f))**2)/(2*(std_f**2))))) #right
218.                n_f = sum(np.divide(np.multiply(radius_AER_dif, n_f0),radius_AER__mu
    l))
219.
220.
221.                n_c0= ((radius_AER__div2)*((volc_c/(sqrt(2*pi)*std_c))*np.exp(-
    ((np.log(radius_AER)-np.log(vmr_c))**2)/(2*(std_c**2)))))
222.                n_c = sum(np.divide(np.multiply(radius_AER_dif, n_c0),radius_AER__mu
    l))
223.
224.                #
225.                # CHECK THE CONVERSION RESULTS
226.                #
```

```
227.                # Reproduce the AERONET volume size distribution, using the calculat
    ed
228.                # parameters of the number size distribution
229.                VSD_AER_calc_new = np.add ( \
230.                        n_f*np.exp(-1*((np.log(radius_AER)-
    np.log(r_f))**2)/(2* (st_f**2)))*(1/(st_f * sqrt(2*pi)))*(radius_AER__mul), \
231.                        (n_c * np.exp(-1*((np.log(radius_AER)-
    np.log(r_c))**2)/(2* (st_c**2)))*(1 /(st_c *sqrt(2*pi))))*(radius_AER__mul))
232.
233.                # Drawing and store the plot
234.                plt.semilogx(radius_AER,y, linewidth=0.5)
235.                plt.semilogx(radius_AER, z, linewidth=0.5)
236.                plt.semilogx(radius_AER, VSD_AER_calc_new, linewidth=0.5)
237.
238.                asy=data_asy_list[index]
239.                sFactor=float(asy[17])
240.                print(file_name + " Sphericity_Factor %.2f %% " % (sFactor))
241.
242.                if(sFactor > float("98")):
243.                    rin=data_rin_list[index]
244.                    rirp440=float(rin[5])
245.                    rirp675=float(rin[6])
246.                    rirp870=float(rin[7])
247.                    rirp1020=float(rin[8])
248.                    riip440=float(rin[9])
249.                    riip675=float(rin[10])
250.                    riip870=float(rin[11])
251.                    riip1020=float(rin[12])
252.                    print("volc_f=%.6f vmr_f=%.6f std_f=%.6f volc_c=%.6f vmr_c=%.6f
    std_c=%.6f rirp440=%.6f rirp675=%.6f rirp870=%.6f rirp1020=%.6f riip440=%.6f riip67
    5=%.6f riip870=%.6f riip1020=%.6f" \
253.                          % (volc_f,vmr_f,std_f,volc_c,vmr_c,std_c,rirp440,rirp675,r
    irp870,rirp1020,riip440,riip675,riip870,riip1020))
254.
255.                    #wavelength=0.44
256.
257.                    # ----------------------------------------------------------------
    ---------
258.                    #
259.                    #
260.                    # ----------------------------------------------------------------
    ---------
261.                    # Choose the real and imaginary part of the refractive index, ac
    cording to
262.                    # the wavelength
263.                    #
264.                    if (wavelength ==0.44):
265.                        print("WVL 0.44")
266.                        RI_rl_f= rirp440
267.                        RI_rl_c= rirp440
268.                        RI_im_f= riip440
269.                        RI_im_c= riip440
270.                    elif (wavelength==0.675):
271.                        print("WVL 0.675")
272.                        RI_rl_f= rirp675
273.                        RI_rl_c= rirp675
274.                        RI_im_f= riip675
275.                        RI_im_c= riip675
276.                    elif (wavelength==0.87):
277.                        print("WVL 0.87")
278.                        RI_rl_f= rirp870
279.                        RI_rl_c= rirp870
280.                        RI_im_f= riip870
281.                        RI_im_c= riip870
282.                    elif (wavelength==1.02):
283.                        print("WVL 1.02")
```

```
                        RI_rl_f= rirp1020
                        RI_rl_c= rirp1020
                        RI_im_f= riip1020
                        RI_im_c= riip1020

                    mie_code(n_f, r_f, st_f, n_c, r_c, st_c, RI_rl_f, RI_im_f, RI_rl_c, RI_im_c, wavelength)


                else:
                    print("Sphericity_Factor lower than 98 % therefore no spherical particle")

                plt.legend(['AERONET Data','VSD_AER_calc','VSD_AER_calc_new','Fine - Coarse Mode'], loc='upper right', fontsize = 'x-small')
                plt.title(site_lable + time + "\n Reproduce AERONET SD - Wavelength= "+str(wavelength), fontsize = 'medium')
                plt.xlabel('radius (um)',labelpad=10)
                plt.xlim(right=100)
                plt.xlim(left=0.01)
                plt.ylabel('dV/dlnR (um\u00b3 um\u00b2)',labelpad=10)
                plt.grid(True, linewidth=0.3)
                plt.savefig(file_name+".png", dpi=300, bbox_inches="tight")
                plt.close()

                index= index+1
        else:
            period_start_date=str(args.start_date.day)+ "/" + str(args.start_date.month) + "/" + str(args.start_date.year)
            period_end_date=str(args.end_date.day)+ "/" + str(args.end_date.month) + "/" + str(args.end_date.year)
            print ("Site=" + site + " no data is available in this period: starting date=" + period_start_date + " and ending date=" + period_end_date )
```

Appendix B: My_mie.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Nov  8 17:35:50 2019

@author: Anastasia Georgiou
"""

from bhmie import bhmie

import numpy as np

import matplotlib.pyplot as plt

from math import pi, log,log10

def mie_code(n_f, r_f, st_f, n_c, r_c, st_c, RI_rl_f, RI_im_f, RI_rl_c, RI_im_c, wvl):

#    % INPUT
#% The parameters of the NUMBER size distribution:
#% n_f:  Total number concentration of fine mode
#% r_f:  Geometric mean radius of fine mode
#% st_f: Geometric standard deviation of fine mode
#% n_c:  Total number concentration of coarse mode
#% r_c:  Geometric mean radius of coarse mode
#% st_c: Geometric standard deviation of coarse mode
#% RI_rl_f:  Real part of the refractive index of fine mode
#% RI_im_f:  Imaginary part of the refractive index of fine mode
#% RI_rl_c:  Real part of the refractive index of coarse mode
#% RI_im_c:  Imaginary part of the refractive index of coarse mode
#% wvl:      Wavelength in microns
#
#% OUTPUT
#% The calculated optical properties are provided in the mie_optical vector
#% for the ensemble of the particles, but also for fine and coarse mode
#% separately:
#% ext_all: extinction coefficient of the ensemble
#% sca_all: scattering coefficient of the ensemble
#% abs_all: absorption coefficient of the ensemble
#% bac_all: backscatter coefficient of the ensemble
#% ext_f: extinction coefficient of the fine mode
#% sca_f: scattering coefficient of the fine mode
#% abs_f: absorption coefficient of the fine mode
#% bac_f: backscatter coefficient of the fine mode
#% ext_c: extinction coefficient of the coarse mode
#% sca_c: scattering coefficient of the coarse mode
#% abs_c: absorption coefficient of the coarse mode
#% bac_c: backscatter coefficient of the coarse mode
#
#% The function "mie_code" calculates the scattering properties at
#% user-defined wavelength, with user-defined size distribution and
#% refractive index as provided by the user in the input.
#% The program uses the "bmie" function that calculates the scattering
#% properties of an individual particle, with specific size and refractive
#% index.
#% The calculations are performed for all individual particles with sizes
#% provided by the user-provided size distribution (and the user-provided
#% refractive index)
#% The calculations for specific size (corresponding to specific size bin)
```

```
#% are multiplied with the number of the particles in the corresponding
#% size bin. Then, the sum provides the optical properties of the ensemble.
#% The calculations are performed seperately for fine and coarse mode and
#% they are summed at the end of the program. This is due to the
#% (capability of using) different refractive index for fine and coarse
#% modes
#
#
#% EXAMPLE
#% Input:

#    mie_code(n_f, r_f, st_f, n_c, r_c, st_c, RI_rl_f, RI_im_f, RI_rl_c, RI_im_c, wvl):
#% n_f=778; r_f=0.1; st_f=1.5; n_c=0.7; r_c=0.7; st_c=1.6;
#% RI_rl_f=1.45; RI_im_f=0.05; RI_rl_c=1.5; RI_im_c=0.001; wvl=0.532;
#% Output:
#% mie_optical =[0.0459, 0.0354, 0.0105, 0.0006, 0.0419, 0.0315,...
#% 0.0103, 0.0004, 0.0040, 0.0039, 0.0001, 0.0003]


    wavelength=wvl
    for i_mode in (1,2): #1:fine mode, 2: coarse mode

        #------------------------------------------------------------------------
        #---------- Calculate size bin (of the size distribution) parameters
        #------------------------------------------------------------------------

        if (i_mode==1): # FINE MODE
            n1,r1,s1,refr_real,refr_im = n_f,r_f,st_f,RI_rl_f,RI_im_f

        if (i_mode==2): # COARSE MODE
            n1,r1,s1,refr_real,refr_im = n_c,r_c,st_c,RI_rl_c,RI_im_c

        n2,r2,s2=0,0,0 # These are dummy variables

        nb   = 200     # max # of bins in size distribution
        nang = 180     # # of angles for the calculations
        scon = n1 + n2 # total concentration
        cmin = 1.e-9   # concentration cutoff, skip bins with n < cmin*scon

        #---- find the max and min radii using the distribution parameters
                esr = log10(r1)-s1      # exponent of smallest r (r = 10^esr)

                #---- define the radius bins
                elr= log10(50)                                      # exponent of large st r
                nd   = elr - esr                                    # number of decades, i.e., log(r)
                ra = 10 ** (esr + nd * np.arange(start=1, stop=nb+2, step=1) / nb) # bin upper radii, microns
                rb = 10 ** (esr + nd * np.arange(start=2, stop=nb+3, step=1) / nb) # bin lower radii
                r = (ra + rb) / 2.0                             # bin center radii, microns
                wid  = rb - ra                                    # bin widths, microns

                #---- calculate number concentrations in each bin
                #s1l  = log(s1)
                # last changed in mie_code_new2.m
                s1l  = s1

                dcon = np.multiply( \
                    np.multiply(n1,np.exp(np.divide(np.multiply(-1.0,np.power(np.log(np.divide(r, r1)), 2.0)), np.multiply(2.0,np.power(s1l,2.0))))) \
```

```
                    ,np.divide(1.0, np.multiply(np.multiply(r,s1l),np.sqrt(np.multiply(2.0,pi)))))

                #---- calculate and plot volume size distribution  ----------------
                SD_dV_dlnr = np.multiply(np.multiply(np.multiply(r,(4 * pi / 3)),np.power(r,3)), dcon)
                # distribution from the number
                # size distribution
                plt.figure(1)
                plt.ion()

                plt.interactive(True)

                plt.plot(r,SD_dV_dlnr,'m', linewidth=0.5)
                plt.gca()
                plt.ylim(0)
                #plt.semilogx()
                #plt.xlabel('radius  (um)')
                #plt.ylabel('dV/dlnR   (um^3 um^-^2)')

                #---- throw out bins with conc less than total conc * cmin

                conc   = np.multiply(dcon, wid)
                okWhere = np.where(conc > (scon * cmin))
                if (len(okWhere)>0):
                    ok=(np.asarray(list(okWhere)))[0]
                    conc   = conc[ok]
                    radi   = r[ok]
                    dconc  = dcon[ok]
                    width  = wid[ok]
                    vol = np.sum(np.multiply(np.multiply(np.multiply(1.33333, pi), radi ** 3), conc))
                    sfc = np.sum(np.multiply(np.multiply(np.multiply(4, pi), radi ** 2), conc))
                    con  = np.sum(conc)
                    if (con<(0.95*(n1+n2))):
                        print('size_part')
                        #print(size_part) - does "size_part" not exist in the code ?

                        print('integral concentration ')
                        percent = 100 * con / (s1+s2)
                        print("  %.2f %% of expected in lognorm.pro" % (percent))

                conc  = np.multiply(dconc, width)

                #---- zero the sums

                sext  = 0.0
                ssca  = 0.0
                sbac  = 0.0
                sabs  = 0.0

                #     intmie, wavl(i),index(i),radi,conc, $ ; input
                #     sext,ssca,sabs,sbac           ; output

                nang = 20
                nbin  = radi.size

                #---- loop over radius bins, do Mie calcs
                for j in range (0, nbin):
                    x_par = 2.0 * pi * radi[j] / wavelength
                    refrel=complex(refr_real,refr_im)

                    #_____________________________________________________________
    ________
```

```
                    #-------------------------------------------------------------
    --------
                    #_____________________________________________________________
    ________
                    #_____________________________________________________________
    ________
                    #-------------------------------------------------------------
    --------
                    #_____________________________________________________________
    ________

                    s1,s2,qext,qsca,qback2,gsca = bhmie(x_par,refrel,nang)
                    #   bhmie(x,refrel,nang)
                    # function [S1,S2,qext,qsca,qback,gsca]=bhmie(x_par,refrel,nang)

                    # Calculated based on Mie scattering theory
                    # input:
                    #      x_par - size parameter =2pi*radius/wavelength
                    #      refrel - refractive index in complext form for example:
    1.5+0.02*i;
                    #      nang - namber of angle for S1 and S2 function in range fr
    om 0 to pi/2
                    # output:
                    #        S1, S2 - funtion which coresponts to phase function
                    #        Qext - extinction efficiency
                    #        Qsca - scattering efficiency
                    #        Qback -backscatter efficiency
                    #        gsca- asymmetry parameter

                    # S1, S2, gsca are not used herein
                    # EXAMPLE: bhmie(10,complex(1.5,0.01),20)
                    #_____________________________________________________________
    ________
                    #-------------------------------------------------------------
    --------
                    #_____________________________________________________________
    ________

                    ext = pi * (radi[j]**2) * conc[j] * qext * 1.0e-3
                    sca = pi * (radi[j]**2) * conc[j] * qsca * 1.0e-3
                    bac = ((wavelength / (2 * pi))**2) * conc[j] * qback2 * 1.0e-3
                    sext = sext + ext
                    ssca = ssca + sca
                    sbac = sbac + bac

                if (i_mode==1):
                    ext_f=sext
                    sca_f=ssca
                    bac_f=sbac
                if (i_mode==2):
                    ext_c=sext
                    sca_c=ssca
                    bac_c=sbac

            abs_f = ext_f - sca_f
            abs_c= ext_c - sca_c

            ext_all = ext_f + ext_c
            sca_all=sca_f+sca_c
            abs_all=abs_f+abs_c
            bac_all=bac_f+bac_c

            print ("ext_all=" +  str(ext_all) + " sca_all=" + str(sca_all) + " abs_a
    ll=" + str(abs_all) + " bac_all=" + str(bac_all) + " ext_f=" + str(ext_f) + " sca_f
    =" + str(sca_f))
```

```
228.            print ("abs_f=" +  str(abs_f) + " bac_f=" + str(bac_f) + " ext_c=" + str
    (ext_c) + " sca_c=" + str(sca_c) + " abs_c=" + str(abs_c) + " bac_c=" + str(bac_c))
229.
230.            return ext_all,sca_all,abs_all,bac_all,ext_f,sca_f,abs_f,bac_f,ext_c,sca
    _c,abs_c,bac_c
231.            #% mie_optical =[0.0459, 0.0354, 0.0105, 0.0006, 0.0419, 0.0315,...
232.            #% 0.0103, 0.0004, 0.0040, 0.0039, 0.0001, 0.0003]
```

Appendix C: Bhmie.py

The following program bmie.py, it was written in 1983 from Craig F. Bohren and Donald R. Huffman.

```
from numpy import *

def bhmie(x,refrel,nang):
# This file is converted from mie.m, see http://atol.ucsd.edu/scatlib/index.htm
# Bohren and Huffman originally published the code in their book on light scatterin
g

# Calculation based on Mie scattering theory
# input:
#       x       - size parameter = k*radius = 2pi/lambda * radius
#                       (lambda is the wavelength in the medium around the scatterers)

#       refrel - refraction index (n in complex form for example:  1.5+0.02*i;
#       nang   - number of angles for S1 and S2 function in range from 0 to pi/2
# output:
#         S1, S2 - funtion which correspond to the (complex) phase functions
#         Qext   - extinction efficiency
#         Qsca   - scattering efficiency
#         Qback  - backscatter efficiency
#         gsca   - asymmetry parameter


    nmxx=150000

    s1_1=zeros(nang,dtype=complex128)
    s1_2=zeros(nang,dtype=complex128)
    s2_1=zeros(nang,dtype=complex128)
    s2_2=zeros(nang,dtype=complex128)
    pi=zeros(nang)
    tau=zeros(nang)

    if (nang > 1000):
        print ('error: nang > mxnang=1000 in bhmie')
        return

    # Require NANG>1 in order to calculate scattering intensities
    if (nang < 2):
        nang = 2

    pii = 4.*arctan(1.)
    dx = x

    drefrl = refrel
    y = x*drefrl
    ymod = abs(y)


    #    Series expansion terminated after NSTOP terms
    #    Logarithmic derivatives calculated from NMX on down

    xstop = x + 4.*x**0.3333 + 2.0
    nmx = max(xstop,ymod) + 15.0
    nmx=fix(nmx)

    # BTD experiment 91/1/15: add one more term to series and compare resu<s
    #      NMX=AMAX1(XSTOP,YMOD)+16
    # test: compute 7001 wavelen>hs between .0001 and 1000 micron
    # for a=1.0micron SiC grain.  When NMX increased by 1, only a single
    # computed number changed (out of 4*7001) and it only changed by 1/8387
```

```
    # conclusion: we are indeed retaining enough terms in series!

    nstop = int(xstop)

    if (nmx > nmxx):
        print ( "error: nmx > nmxx=%f for |m|x=%f" % ( nmxx, ymod) )
        return

    dang = .5*pii/ (nang-1)


    amu=arange(0.0,nang,1)
    amu=cos(amu*dang)

    pi0=zeros(nang)
    pi1=ones(nang)

    # Logarithmic derivative D(J) calculated by downward recurrence
    # beginning with initial value (0.,0.) at J=NMX

    nn = int(nmx)-1
    d=zeros(nn+1,dtype=complex128)
    for n in range(0,nn):
        en = nmx - n
        d[nn-n-1] = (en/y) - (1./ (d[nn-n]+en/y))

    #*** Riccati-Bessel functions with real argument X
    #    calculated by upward recurrence

    psi0 = cos(dx)
    psi1 = sin(dx)
    chi0 = -sin(dx)
    chi1 = cos(dx)
    xi1 = psi1-chi1*1j
    qsca = 0.
    gsca = 0.
    p = -1

    for n in range(0,nstop):
        en = n+1.0
        fn = (2.*en+1.)/(en* (en+1.))

           # for given N, PSI  = psi_n        CHI  = chi_n
           #              PSI1 = psi_{n-1}    CHI1 = chi_{n-1}
           #              PSI0 = psi_{n-2}    CHI0 = chi_{n-2}
           # Calculate psi_n and chi_n
               psi = (2.*en-1.)*psi1/dx - psi0
               chi = (2.*en-1.)*chi1/dx - chi0
               xi = psi-chi*1j

           #*** Store previous values of AN and BN for use
           #    in computation of g=<cos(theta)>
               if (n > 0):
                   an1 = an
                   bn1 = bn

           #*** Compute AN and BN:
               an = (d[n]/drefrl+en/dx)*psi - psi1
               an = an/ ((d[n]/drefrl+en/dx)*xi-xi1)
               bn = (drefrl*d[n]+en/dx)*psi - psi1
               bn = bn/ ((drefrl*d[n]+en/dx)*xi-xi1)

           #*** Augment sums for Qsca and g=<cos(theta)>
               qsca += (2.*en+1.)* (abs(an)**2+abs(bn)**2)
               gsca += ((2.*en+1.)/ (en* (en+1.)))*( real(an)* real(bn)+imag(an)*im
   ag(bn))
```

```
123.
124.                 if (n > 0):
125.                     gsca += ((en-
    1.)* (en+1.)/en)*( real(an1)* real(an)+imag(an1)*imag(an)+real(bn1)* real(bn)+imag(
    bn1)*imag(bn))
126.
127.
128.             #*** Now calculate scattering intensity pattern
129.             #    First do angles from 0 to 90
130.                 pi=0+pi1    # 0+pi1 because we want a hard copy of the values
131.                 tau=en*amu*pi-(en+1.)*pi0
132.                 s1_1 += fn* (an*pi+bn*tau)
133.                 s2_1 += fn* (an*tau+bn*pi)
134.
135.             #*** Now do angles greater than 90 using PI and TAU from
136.             #    angles less than 90.
137.             #    P=1 for N=1,3,...% P=-1 for N=2,4,...
138.             #   remember that we have to reverse the order of the elements
139.             #   of the second part of s1 and s2 after the calculation
140.                 p = -p
141.                 s1_2+= fn*p* (an*pi-bn*tau)
142.                 s2_2+= fn*p* (bn*pi-an*tau)
143.
144.                 psi0 = psi1
145.                 psi1 = psi
146.                 chi0 = chi1
147.                 chi1 = chi
148.                 xi1 = psi1-chi1*1j
149.
150.             #*** Compute pi_n for next value of n
151.             #    For each angle J, compute pi_n+1
152.             #    from PI = pi_n , PI0 = pi_n-1
153.                 pi1 = ((2.*en+1.)*amu*pi- (en+1.)*pi0)/ en
154.                 pi0 = 0+pi   # 0+pi because we want a hard copy of the values
155.
156.             #*** Have summed sufficient terms.
157.             #    Now compute QSCA,QEXT,QBACK,and GSCA
158.
159.             #   we have to reverse the order of the elements of the second part of s
    1 and s2
160.             s1=concatenate((s1_1,s1_2[-2::-1]))
161.             s2=concatenate((s2_1,s2_2[-2::-1]))
162.             gsca = 2.*gsca/qsca
163.             qsca = (2./ (dx*dx))*qsca
164.             qext = (4./ (dx*dx))* real(s1[0])
165.
166.             # more common definition of the backscattering efficiency,
167.             # so that the backscattering cross section really
168.             # has dimension of length squared
169.             #qback = 4*(abs(s1[2*nang-2])/dx)**2
170.             #qback = ((abs(s1[2*nang-2])/dx)**2 )/pii  #old form
171.             qback = ((abs(s1[2*nang-2]))**2 ) # Alex
172.
173.             return s1,s2,qext,qsca,qback,gsca
```