

**Tikhonov Advanced Regularization Algorithm (TiARA) for  
lidar retrievals of aerosol optical and microphysical parameters.**

**Algorithm Theoretical Basis Document**

**Primary Authors:**

Eduard V. Chemyakin, Science Systems and Applications, Inc. (SSAI), Hampton, VA, USA  
Detlef G. Müller, Science Systems and Applications, Inc. (SSAI), Hampton, VA, USA  
Alexei V. Kolgotin, Physics Instrumentation Center, Troitsk, Moscow Region, Russia

**??-??-??**

**Release 1.0 – DRAFT**

**Last Update: 30 May 2018**

**TiARA for lidar retrievals of aerosol optical and microphysical  
parameters. Algorithm Theoretical Basis Document**

**Document No: ??-???-???**

Prepared By:

---

Eduard V. Chemyakin  
Senior Research Scientist

Date

---

Detlef G. Müller  
Chief Research Scientist

Date

---

Alexei V. Kolgotin  
Research Scientist

Date

Approved By:

---

Chris A. Hostetler  
HSRL Principal Investigator

Date

---

Richard A. Ferrare  
HSRL Principal Investigator

Date

## Table of Contents

1. Introduction .....	5
1.1. Purpose and Scope.....	5
1.2. Revision History .....	5
2. Inputs and Outputs .....	5
3. User Input File .....	9
4. Inversion Methodology Overview.....	9
4.1. Definition of the Search Space.....	11
4.2. Radius Grid Bins .....	13
4.3. Base Functions .....	14
4.4. Tikhonov Regularization.....	15
4.4.1. Regularization Parameter $\gamma$ .....	19
4.4.2. Smoothing Matrix $\mathbf{H}$ .....	20
4.5. Selection and Averaging of Solutions.....	21
4.6. Extreme Error Computation Scheme .....	25
4.7. Summary of Inversion Variables .....	28
5. Details of Implementation .....	29
5.1. Optimized Look-Up Table.....	29
5.2. Calculation of Microphysical Size Parameters from Pre-Integration of Triangular Base Functions	39
5.2.1. Total Number Concentration .....	39
5.2.2. Total Surface Area Concentration.....	39
5.2.3. Total Volume Concentration.....	40
5.2.4. Total Effective Variance .....	40
5.3. Parallelization of Computations.....	41
5.4. Automated Unsupervised Inversion .....	42
6. Simulations.....	49
6.1. Simulation Mode.....	49
6.2. Logarithmic-Normal Particle Size Distribution.....	52
6.3. Preliminary Uncertainty Analysis .....	53
6.4. The cross-talk between effective radius and total number concentration .....	55

6.5. Simulation Results.....	56
6.5.1. Case Study.....	59
6.5.2. Statistics: Histograms.....	65
Appendix A. Inversion Parameters ASCII File.....	67
Appendix B. TiARA’s Software Routines .....	74
Acronyms and Abbreviations.....	78
References .....	79

## 1. Introduction

### 1.1. Purpose and Scope

This document describes the Tikhonov Advanced Regularization Algorithm (TiARA) that is used for the analysis of data acquired by the NASA Langley Research Center (LaRC) High-Spectral Resolution Lidar (HSRL-2). The purpose of TiARA is to retrieve the aerosol microphysical parameters using experimental HSRL-2 data (inversion), and thus to offer a possibility of obtaining the quantitative measure of the spatial and temporal variation of aerosol properties. The outputs of TiARA are the numerous microphysical parameters that can be used for the comparisons with the results of measurements of *in situ* and remote sensing instruments, and for other related purposes. The major advantage of TiARA is that all the delivered aerosol products are vertically-resolved. The vertical resolution of aerosol parameters is critical for some atmospheric applications like, for instance, climate modelling.

In the following sections we describe the theory of inversion that is used in TiARA software. We summarize simulation studies which show the performance, i.e., quality of inversion results, and discuss the capabilities and limitations of the retrieval of aerosol microphysical parameters. We also present the overall architecture of the TiARA software design and provide the details of choosing the data processing settings.

### 1.2. Revision History

The algorithm is described in this document the way it has been implemented in Version 1.0 of the HSRL production data processing system. Subsequent changes, if any, will be documented in Table 1.

Table 1: Revision history of the Tikhonov Advanced Regularization Algorithm (TiARA)

Version	Release Date	Comments
1.0	30 May 2018	Initial version

## 2. Inputs and Outputs

The retrieval of aerosol microphysical parameters requires the solution of a system of Fredholm integral equations of the first kind. The input optical data provided by an advanced lidar system consist of the backscatter ( $\beta$ ) and extinction ( $\alpha$ ) coefficients:

$$\beta_{\lambda} = \int_0^{\infty} K_{\beta}(m, r, \lambda) f(r) dr = \int_0^{\infty} \pi r^2 Q_{\beta}(m, r, \lambda) f(r) dr, \quad (1)$$

$$\alpha_\lambda = \int_0^\infty K_\alpha(m, r, \lambda) f(r) dr = \int_0^\infty \pi r^2 Q_\alpha(m, r, \lambda) f(r) dr. \quad (2)$$

The term  $\lambda$  denotes the wavelength of interacting light and  $m = m_R - i \cdot m_I$  is the complex refractive index (CRI) representing the material / chemical composition of particles. The expression  $m_R$  denotes the real part that describes the refraction of electromagnetic radiation by the particles, and the expression  $m_I$  denotes the imaginary part that describes the absorption of electro-magnetic radiation by the particles. The function  $f(r)$  describes the particle size distribution (PSD) expressed as the number of particles per unit volume between particle radius  $r$  and  $r + dr$ . The terms  $K_\beta(m, r, \lambda) = \pi r^2 Q_\beta(m, r, \lambda)$  and  $K_\alpha(m, r, \lambda) = \pi r^2 Q_\alpha(m, r, \lambda)$  describe, respectively, the backscatter and extinction number kernel functions. The multiplier  $\pi r^2$  is the geometric cross section of a homogeneous spherical particle whose interaction with the light can be characterized by a corresponding backscatter ( $Q_\beta$ ), extinction ( $Q_\alpha$ ), absorption ( $Q_{\text{abs}}$ ), and scattering ( $Q_{\text{sca}}$ ) efficiencies [2]. The surface area kernel functions can be expressed by the corresponding number kernel functions as:

$$K_{\beta,S}(m, r, \lambda) = \frac{1}{4\pi r^2} K_\beta(m, r, \lambda) = \frac{1}{4} Q_\beta(m, r, \lambda), \quad (3)$$

$$K_{\alpha,S}(m, r, \lambda) = \frac{1}{4\pi r^2} K_\alpha(m, r, \lambda) = \frac{1}{4} Q_\alpha(m, r, \lambda). \quad (4)$$

The volume kernel functions can be expressed by the corresponding number kernel functions as:

$$K_{\beta,V}(m, r, \lambda) = \frac{1}{\frac{4\pi}{3} r^3} K_\beta(m, r, \lambda) = \frac{3}{4r} Q_\beta(m, r, \lambda), \quad (5)$$

$$K_{\alpha,V}(m, r, \lambda) = \frac{1}{\frac{4\pi}{3} r^3} K_\alpha(m, r, \lambda) = \frac{3}{4r} Q_\alpha(m, r, \lambda). \quad (6)$$

We recommend using the volume kernel functions for the microphysical retrievals because the optical effects of ambient aerosols are more closely related to their volume rather than their number or surface area [24, 32]. Nevertheless, the number and surface area kernel functions can also be used in the inversion software. If the surface area kernel functions are selected then the retrieved PSD  $f_S(r) = 4\pi r^2 f(r)$  represents the surface area, and if the volume kernel functions are used then the retrieved PSD  $f_V(r) = \frac{4\pi}{3} r^3 f(r)$  represents the volume of particles per unit volume between particle radius  $r$  and  $r + dr$ .

Here we consider only the NASA LaRC High-Spectral Resolution Lidar (HSRL-2) as a source of input  $3\beta+2\alpha$  information. In this case, the set of optical data  $\{\beta_{355}, \beta_{532}, \beta_{1064}, \alpha_{355}, \alpha_{532}\}$  consisting of the backscatter coefficients at 355, 532, and 1064 nm, and extinction coefficients at 355 and 532 nm is known as the result of measurements. The unknown quantities to be retrieved from the system of Eqs. (1)–(2) are the real ( $m_R$ ) and imaginary ( $m_I$ ) parts of the CRI that are assumed to be particle size- and wavelength-independent, and the PSD  $f(r)$ . We solve the inverse problem in order to estimate the PSD and CRI from which we can compute the microphysical parameters listed below. Among others, the list of retrieved microphysical parameters includes absorption (abs) and scattering (sca) coefficients:

$$\text{abs}_\lambda = \int_0^\infty K_{\text{abs}}(m, r, \lambda) f(r) dr = \int_0^\infty \pi r^2 Q_{\text{abs}}(m, r, \lambda) f(r) dr, \quad (7)$$

$$\text{sca}_\lambda = \int_0^\infty K_{\text{sca}}(m, r, \lambda) f(r) dr = \int_0^\infty \pi r^2 Q_{\text{sca}}(m, r, \lambda) f(r) dr, \quad (8)$$

where the terms  $K_{\text{abs}}(m, r, \lambda) = \pi r^2 Q_{\text{abs}}(m, r, \lambda)$  and  $K_{\text{sca}}(m, r, \lambda) = \pi r^2 Q_{\text{sca}}(m, r, \lambda)$  describe, respectively, the absorption and scattering number kernel functions.

Let us provide the full list of microphysical parameters that can be retrieved with TiARA. All outputs are available for the total, fine, and coarse modes. The fine/total mode ratios are also provided. For the total mode case, the integration is done for the whole selected range  $[r_{\text{min}}, r_{\text{max}}]$  (see Section “4.1. Definition of the Search Space” for more details) of radii of optically active particles. For the fine mode case, the integration is done for the range  $[r_{\text{min}}, r_{\text{fine}}]$ , where  $r_{\text{fine}} (\leq r_{\text{max}})$  is the chosen threshold of the fine mode aerosols (see note at the end of this paragraph). For the coarse mode case, the integration is done for the range  $[r_{\text{fine}}, r_{\text{max}}]$ . All the equations in this section are given for the total mode parameters and their fine/total mode ratios. To compute the value of parameters for the fine and coarse modes, it is necessary to change the limits of integration in the following equations. The numerical integration is done using Simpson’s rule [7]. Without loss of generality, the term  $f(r)$  in this section means the retrieved PSD, and  $m$  is the retrieved CRI.

*Note.* In the inversion parameters ASCII file the radius threshold  $r_{\text{fine}}$  between the fine and coarse mode aerosols can be selected using the parameter “BorderOfFineMode”:  
// Border between fine and coarse modes, num.  
BorderOfFineMode=0.5

Total number concentration and its fine/total mode ratio:

$$n = \int_{r_{\text{min}}}^{r_{\text{max}}} f(r) dr, \quad n^{\text{ratio}} = \frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} f(r) dr}. \quad (9)$$

Total surface area concentration and its fine/total mode ratio:

$$s = 4\pi \int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr, \quad s^{\text{ratio}} = \frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} r^2 f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr}. \quad (10)$$

Total volume concentration and its fine/total mode ratio:

$$v = \frac{4\pi}{3} \int_{r_{\text{min}}}^{r_{\text{max}}} r^3 f(r) dr, \quad v^{\text{ratio}} = \frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} r^3 f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} r^3 f(r) dr}. \quad (11)$$

Total effective radius and its fine/total mode ratio:

$$r_{\text{eff}} = \frac{\int_{r_{\text{min}}}^{r_{\text{max}}} r^3 f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr}, \quad r_{\text{eff}}^{\text{ratio}} = \frac{\frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} r^3 f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{fine}}} r^2 f(r) dr}}{\frac{\int_{r_{\text{min}}}^{r_{\text{max}}} r^3 f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr}}. \quad (12)$$

Total effective variance and its fine/total mode ratio:

$$v_{\text{eff}} = \frac{\int_{r_{\text{min}}}^{r_{\text{max}}} (r - r_{\text{eff}})^2 r^2 f(r) dr}{r_{\text{eff}}^2 \int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr}, \quad v_{\text{eff}}^{\text{ratio}} = \frac{\frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} (r - r_{\text{eff},\text{fine}})^2 r^2 f(r) dr}{r_{\text{eff},\text{fine}}^2 \int_{r_{\text{min}}}^{r_{\text{fine}}} r^2 f(r) dr}}{\frac{\int_{r_{\text{min}}}^{r_{\text{max}}} (r - r_{\text{eff}})^2 r^2 f(r) dr}{r_{\text{eff}}^2 \int_{r_{\text{min}}}^{r_{\text{max}}} r^2 f(r) dr}}. \quad (13)$$

Total optical coefficients and their fine/total mode ratios:

$$l_{\lambda} = \int_{r_{\text{min}}}^{r_{\text{max}}} K_l(m, r, \lambda) f(r) dr, \quad l_{\lambda}^{\text{ratio}} = \frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} K_l(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} K_l(m, r, \lambda) f(r) dr}, \quad (14)$$

where  $l$  represents the type of data ( $\beta$  is the backscatter,  $\alpha$  is the extinction, “abs” is the absorption, and “sca” is the scattering coefficient, respectively), and the supported wavelengths  $\lambda$  are 355, 532, and 1064 nm. We have to note that the retrieved backscatter and extinction coefficients are comparable to the measured values but not necessarily identical.

Total single-scattering albedos (SSA) and their fine/total mode ratios:

$$\text{SSA}_{\lambda} = \frac{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\text{sca}}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\alpha}(m, r, \lambda) f(r) dr}, \quad \text{SSA}_{\lambda}^{\text{ratio}} = \frac{\frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} K_{\text{sca}}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{fine}}} K_{\alpha}(m, r, \lambda) f(r) dr}}{\frac{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\text{sca}}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\alpha}(m, r, \lambda) f(r) dr}}. \quad (15)$$

Total lidar ratios and their fine/total mode ratios:

$$S_{a,\lambda} = \frac{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\alpha}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\beta}(m, r, \lambda) f(r) dr}, \quad S_{a,\lambda}^{\text{ratio}} = \frac{\frac{\int_{r_{\text{min}}}^{r_{\text{fine}}} K_{\alpha}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{fine}}} K_{\beta}(m, r, \lambda) f(r) dr}}{\frac{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\alpha}(m, r, \lambda) f(r) dr}{\int_{r_{\text{min}}}^{r_{\text{max}}} K_{\beta}(m, r, \lambda) f(r) dr}}. \quad (16)$$

Total Ångström coefficients for the retrieved data:

$$\text{Å}_{355/532}^{l^*} = \frac{\ln\left(\frac{l_{355}^*}{l_{532}^*}\right)}{\ln\left(\frac{532}{355}\right)}, \quad \text{Å}_{532/1064}^{l^*} = \frac{\ln\left(\frac{l_{532}^*}{l_{1064}^*}\right)}{\ln\left(\frac{1064}{532}\right)}. \quad (17)$$

where the symbol  $l^*$  represents the type of data ( $\beta$  is the backscatter,  $\alpha$  is the extinction, “abs” is the absorption, “sca” is the scattering, “SSA” is the single-scattering albedo, and  $S_a$  is the lidar ratio).

### 3. User Input File

The microphysical retrievals can be performed in numerical simulation mode or by using the experimental lidar data stored in a HDF-file. In simulation mode, the inputs are computed on the basis of PSD and CRI set by the user or directly taken from the user input file (see Section “Appendix A. Inversion Parameters ASCII File”) and the retrieved information is stored on the hard drive of the computer as multiple ASCII solutions-set files (\*.ssf files). These \*.ssf files can be used for further detailed analysis using the manual expert version of TiARA. In experimental mode, the inputs are taken from standard HSRL-2 HDF-file and outputs are stored in the same HDF-file. The experimental mode is primarily considered in this ATBD. Additional information about the simulation mode can be found in Section 6.

One has to use the command line to run the TiARA microphysical retrievals. The user input file (see Section “Appendix A. Inversion Parameters ASCII File”) is needed as a part of input information. This ASCII file allows one choosing the TiARA settings. In order to make it easier to understand how to choose these settings, we provide the relevant fragments of configuration file throughout the document as the “*Note*” paragraphs in places where each input parameter is discussed. In experimental mode, the most important user settings are stored in the “Process\_Log” record of the HDF-file for the purpose of quality control.

### 4. Inversion Methodology Overview

In the microphysical retrievals we assume that particles are of spherical shape and that their inner structure is homogeneous, i.e., the chemical components of each particle are internally mixed. In that case we can compute the backscatter ( $Q_\beta$ ), extinction ( $Q_\alpha$ ), absorption ( $Q_{\text{abs}}$ ), and scattering ( $Q_{\text{sca}}$ ) efficiencies using Mie-scattering theory [2], and the verified code [3]:

$$Q_\beta(m, r, \lambda) = \frac{1}{x^2} \left| \sum_{n=1}^{\infty} (2n+1)(-1)^n (a_n - b_n) \right|^2, \quad (18)$$

$$Q_\alpha(m, r, \lambda) = \frac{2}{x^2} \sum_{n=1}^{\infty} (2n+1) \Re(a_n + b_n), \quad (19)$$

$$Q_{\text{sca}}(m, r, \lambda) = \frac{2}{x^2} \sum_{n=1}^{\infty} (2n+1) (|a_n|^2 + |b_n|^2), \quad (20)$$

$$Q_{\text{abs}}(m, r, \lambda) = Q_\alpha(m, r, \lambda) - Q_{\text{sca}}(m, r, \lambda), \quad (21)$$

where  $x = 2\pi r/\lambda$  is the size parameter. The coefficients  $a_n$  and  $b_n$  are computed as [2, 3]:

$$a_n = \frac{m\psi_n(mx)\psi'_n(x) - \psi_n(x)\psi'_n(mx)}{m\psi_n(mx)\xi'_n(x) - \xi_n(x)\psi'_n(mx)}, \quad (22)$$

$$b_n = \frac{\psi_n(mx)\psi'_n(x) - m\psi_n(x)\psi'_n(mx)}{\psi_n(mx)\xi'_n(x) - m\xi_n(x)\psi'_n(mx)}, \quad (23)$$

$$\psi_n(\rho) = \rho j_n(\rho), \quad \xi_n(\rho) = \rho j_n(\rho) + i\rho y_n(\rho), \quad (24)$$

where  $j_n$  and  $y_n$  are the spherical Bessel functions of the first (see Fig. 1.a) and second (see Fig. 1.b) kind:

$$j_0(\rho) = \frac{\sin \rho}{\rho}, \quad j_1(\rho) = \frac{\sin \rho}{\rho^2} - \frac{\cos \rho}{\rho}, \quad j_n(\rho) = \frac{2n-1}{\rho} j_{n-1}(\rho) - j_{n-2}(\rho), \quad (25)$$

$$y_0(\rho) = -\frac{\cos \rho}{\rho}, \quad y_1(\rho) = -\frac{\cos \rho}{\rho^2} - \frac{\sin \rho}{\rho}, \quad y_n(\rho) = \frac{2n-1}{\rho} y_{n-1}(\rho) - y_{n-2}(\rho), \quad (26)$$

and

$$\psi'_n(\rho) = \psi_{n-1}(\rho) - \frac{n\psi_n(\rho)}{\rho}, \quad \xi'_n(\rho) = \xi_{n-1}(\rho) - \frac{n\xi_n(\rho)}{\rho}. \quad (27)$$

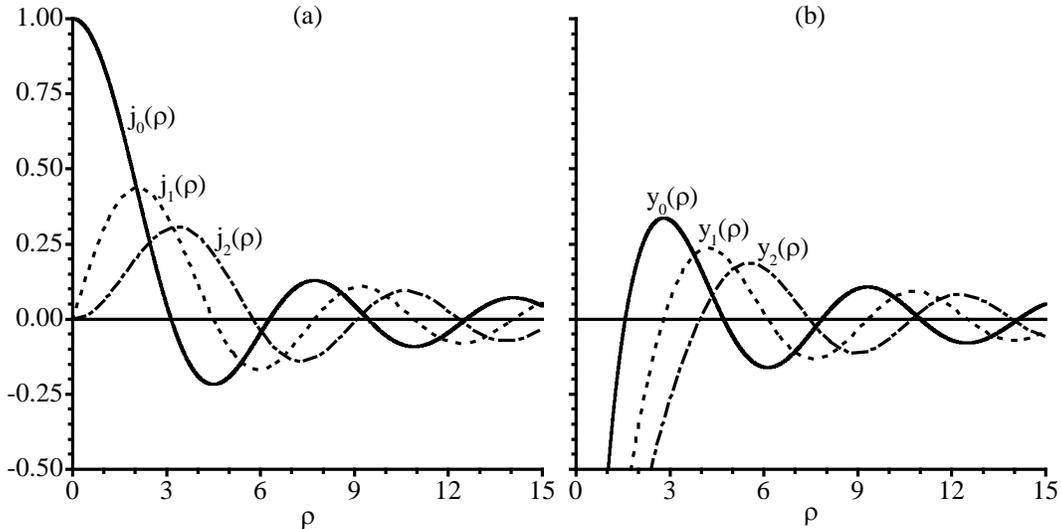


Figure 1. Spherical Bessel functions of the first (a) and second (b) kind.

It is well known that the maximum sensitivity of particle light-scattering occurs if the particle size (diameter) is in the range of the sounding wavelength [2]. Therefore the range of retrievable PSDs is limited. For instance, we can predict theoretically that any inversion scheme will experience difficulties for atmospheric particles that are much smaller compared to the HSRL-2 measurement wavelengths. The spherical Bessel functions of the second kind  $y_n$  [see Eq. (26)] for all orders ( $n = 0, 1, 2, \dots$ ) become infinite as the argument approaches the origin [see Fig. 1.b], restricting the capacity of microphysical retrievals for small particles if only lidar wavelengths 355, 532 and 1064 nm are used.

If we compute backscatter and extinction coefficients which serve as input information for the simulations then the numerical integration in Eqs. (1)–(2) is done using Simpson’s rule [7]. The range of integration is selected in an adaptive manner that allows us to cover the non-zero part of a given input PSD, but at least the radius range from 1 nm to 20  $\mu\text{m}$  is covered.

*Note.* In the inversion parameters ASCII file the step of numerical integration of input  $3\beta+2\alpha$  optical data can be selected using the parameter “OpticalStep”:  
// Integration step during generation of optical data, mum.  
OpticalStep=0.001

## 4.1. Definition of the Search Space

Realistically, the integration in Eqs. (1)–(2) should be done not from zero to infinity, but for the range  $[r_{\min}, r_{\max}]$  of radii of aerosols that are optically active at the measurement wavelengths. In order to estimate the range  $[r_{\min}, r_{\max}]$  for every individual  $3\beta+2\alpha$  optical data set that is processed by TiARA, we introduce a *gliding inversion window* and the specific feature that the inversion window has a variable width. We predefine two sets consisting of several minimum and maximum particle radii. These two sets might have an insignificant overlap. Figure 2.a shows a qualitative example of three entries in both minimum and maximum radii sets. The given example results in nine inversion windows because each of the three minimum radii  $r_{\min}$  forms a separate inversion window  $[r_{\min}^{(k)}, r_{\max}^{(k)}]$  if combined with each of the three maximum radii  $r_{\max}$  [see Fig. 2.a]. We recommend avoiding the inversion windows narrower than 0.38  $\mu\text{m}$  because we consider them to be nonphysical. The minimum radius  $r_{\min}^{(k)}$  of inversion window also should not be higher than 0.3  $\mu\text{m}$ . If the optimized look-up table (see Section “5.1. Optimized Look-Up Table” for more details) is used for the microphysical retrievals then these two limitations are applied automatically. There is an option to ignore the limitations and have more freedom in selecting inversion windows if the direct Mie-scattering calculations are used instead of the optimized look-up table.

The idea behind the gliding inversion window is the following. By moving the window like a caterpillar, we will at one point cover the active radius range of the sought PSD by a set of base functions that are defined in the Section “4.3. Base Functions”. At that point we will be able to reconstruct the sought PSD. When the active radius range matches the sought PSD, the agreement should improve between the reconstructed optical coefficients and the measured optical coefficients  $\{\beta_{355}, \beta_{532}, \beta_{1064}, \alpha_{355}, \alpha_{532}\}$ .

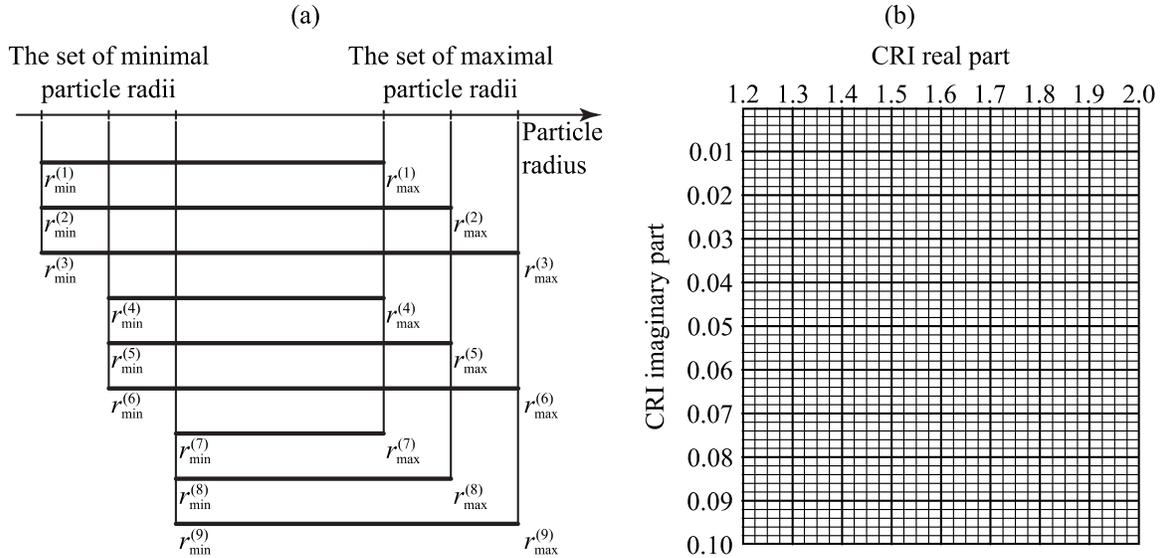


Figure 2. a) Gliding inversion window. The formation of the set of nine inversion windows.  
b) Grid of complex refractive indices.

The set of inversion windows is accompanied by a set of CRIs. We are looking for the real part of CRI ( $m_R$ ) somewhere in the range from 1.2 to 2, which is slightly wider than it is expected to be for the ambient aerosols [8]. For the imaginary part ( $m_I$ ) the range is from 0 to 0.1 [8]. We introduce dense grid to evenly cover the search space of CRI [see Fig. 2.b]. We recommend to use the step of grid for the imaginary part of CRI about ten times smaller compared with the step of grid for the real part. Sensitivity study based on  $3\beta+2\alpha$  data shows that the achievable precision of  $m_R$  estimation is  $\pm 0.05$ , and for the  $m_I$  estimation is  $\pm 0.01$  [29]. The corresponding grid steps for the real and imaginary part of CRI should be at least two times smaller than the achievable precisions.

*Note.* In the inversion parameters ASCII file the sets consisting of several minimum ( $r_{\min}$ ) and maximum ( $r_{\max}$ ) particle radii can be selected using the parameters “RminMin”, “RminMax”, “RminStep”, “RmaxMin”, “RmaxMax”, and “RmaxStep”:

```
// Minimum value of lower limit of particle radii, mum.
```

```
RminMin=0.05
```

```
// Maximum value of lower limit of particle radii, mum.
```

```
RminMax=0.2
```

```
// Step of lower limit of particle radii, mum.
```

```
RminStep=0.01
```

```
// Minimum value of upper limit of particle radii, mum.
```

```
RmaxMin=0.5
```

```
// Maximum value of upper limit of particle radii, mum.
```

```
RmaxMax=8
```

```
// Step of upper limit of particle radii, mum.
```

```
RmaxStep=0.25
```

The grid of complex refractive indexes can be defined using the parameters “CRIRealMin”, “CRIRealMax”, “CRIRealStep”, “CRIImagMin”, “CRIImagMax”, and “CRIImagStep”:

```
// Minimum value of real part of complex refractive index.
CRIRealMin=1.325
// Maximum value of real part of complex refractive index.
CRIRealMax=1.8
// Step of real part of complex refractive index.
CRIRealStep=0.025

// Minimum value of imaginary part of complex refractive index.
CRIImagMin=0
// Maximum value of imaginary part of complex refractive index.
CRIImagMax=0.1
// Step of imaginary part of complex refractive index.
CRIImagStep=0.002
```

## 4.2. Radius Grid Bins

The software supports two options with regard to how the  $N_B$  triangular base functions are distributed along the radius grid bins of the inversion window  $[r_{\min}, r_{\max}]$ . The first option is the equidistant distribution:

$$r_j = r_{\min} + j \frac{r_{\max} - r_{\min}}{N_B + 1}, \quad j = 0, \dots, N_B + 1. \quad (28)$$

The second option is the logarithmic-equidistant distribution:

$$r_j = r_{\min}^{1 - \frac{j}{N_B + 1}} r_{\max}^{\frac{j}{N_B + 1}}, \quad j = 0, \dots, N_B + 1. \quad (29)$$

The number of grid bins in both cases is equal to  $N_B + 2$ . We recommend to use the logarithmic-equidistant distribution for experimental  $3\beta + 2\alpha$  lidar data processing because the shape of ambient PSDs is commonly assumed to be multimodal logarithmic-normal [10, 24]. The equidistant distribution might be used though for numerical simulations.

We would like to point out that the logarithmic-equidistant distribution of radii grid bins may be more appropriate for the case of resolving PSDs that mainly consist of small particles [see Table 3 as an example of 71 grid bins in the range from 10 nm to 10  $\mu$ m]. Thus, we expect that fine mode particles will be represented better compared to coarse mode in the microphysical retrievals if the logarithmic-equidistant distribution is used. Fine mode particles are optically the most active in view of the sounding wavelength of the HSRL-2 instrument (355, 532, and 1064 nm) and the fact that the maximum sensitivity of particle light-scattering occurs if the particle size (diameter) is in the range of the sounding wavelength [2, 3].

*Note.* In the inversion parameters ASCII file the type of distribution of radius grid bins can be selected by using the parameter “GridBinsDistr”:

```
// Type of distribution of particle radii grid bins.
// If "E" then equidistant distribution.
// If "L" then logarithmic-equidistant distribution.
GridBinsDistr=L
```

### 4.3. Base Functions

We avoid strict *a priori* assumptions regarding the shape of the sought PSD  $f(r)$ . Instead, we use a set of base functions  $\{B_j(r)\}$  that approximate the shape of the PSD:

$$f(r) = \sum_{j=1}^{N_B} f_j B_j(r) + \varepsilon^{\text{appr}}(r) = \sum_{j=1}^{N_B} f_{j,S} \frac{B_j(r)}{4\pi r^2} + \varepsilon^{\text{appr}}(r) = \sum_{j=1}^{N_B} f_{j,V} \frac{B_j(r)}{\frac{4\pi}{3} r^3} + \varepsilon^{\text{appr}}(r), \quad (30)$$

where  $f_j$  are the weight coefficients if the number kernel functions are used,  $f_{j,S}$  are the weight coefficients if the surface area kernel functions are used,  $f_{j,V}$  are the weight coefficients if the volume kernel functions are used in the retrievals. A mathematical error  $\varepsilon^{\text{appr}}(r)$  result from this approximation [for example, see crosshatched areas in Fig. 3].

Triangular base functions  $B_j(r)$  are the only supported type of base functions because they are simple and can adequately reproduce the logarithmic-normal shape of PSDs, which is believed to be typical for the ambient aerosols [10, 24]. We also implicitly introduce an assumption into inversion that the sought PSD is equal to zero at its left ( $r_{\min}$ ) and right ( $r_{\max}$ ) borders, i.e.,  $f(r_{\min}) = f(r_{\max}) = 0$ .

Each triangular function is given by the expression:

$$B_j(r) = \begin{cases} 0; & r \leq r_{j-1} \\ \frac{r - r_{j-1}}{r_j - r_{j-1}}; & r_{j-1} \leq r \leq r_j \\ \frac{r_{j+1} - r}{r_{j+1} - r_j}; & r_j \leq r \leq r_{j+1} \\ 0; & r \geq r_{j+1} \end{cases} \quad j = 1, \dots, N_B, \quad (31)$$

where  $r_0 = r_{\min}$  and  $r_{N_B+1} = r_{\max}$ . The recommended number of base functions  $N_B$  to be used for the processing of data from  $3\beta+2\alpha$  lidar systems is equal to eight [see Fig. 3]. In this case, the PSD is retrieved for ten radius grid bins with a zero assumptions at the borders of inversion window  $r_{\min}$  and  $r_{\max}$ . Assumption of zero for the value of the PSD allows us to increase the number of pieces of input information from five to seven. We do not recommend increasing parameter  $N_B$  more than double number of input optical data because the inversion problem becomes extremely unstable if too many base functions are used.

Figure 3 illustrates the arrangement of the base functions. The thick solid curve shows a logarithmic-normal distribution function (see Section “6.2. Logarithmic-Normal Particle Size Distribution”) that represents in an appropriate manner the typical shape of monomodal atmospheric particle size distributions. One can see that eight triangular functions have the capacity to take account of this shape, successfully compensating for the fact that each single base function separately contains one discontinuity (first derivative) at its peak value. Selected shape of base functions implicitly contains also the smoothing constraint. Triangles suppress oscillations along the upward and downward slope of each triangle. Oscillations are only introduced by the combination of weight factors  $f_j$ .

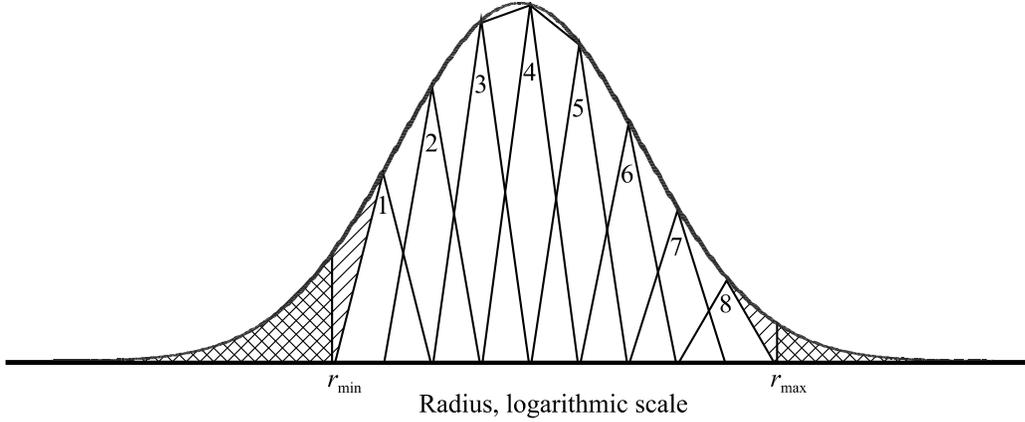


Figure 3. Example of the qualitative reconstruction of a monomodal logarithmic-normal PSD with eight base functions of triangular shape defined on a logarithmic-equidistant scale in the radius interval  $[r_{\min}, r_{\max}]$ . This radius interval is denoted as *inversion window* in the text.

*Note.* In the inversion parameters ASCII file the number  $N_B$  of triangular base functions can be selected using the parameters “DefineNumberOfGridBins” and “NumberOfInternalGridBins”:  
// Define manually number of internal (between lower and upper limits) particle radii grid bins.  
// If "0" then number of internal grid bins will coincide with number of used optical coefficients.  
// If "1" then manual definition will be used.

DefineNumberOfGridBins=1

// Number of internal (between lower and upper limits) particle radii grid bins.

// Used for manual definition.

NumberOfInternalGridBins=8

#### 4.4. Tikhonov Regularization

We approximate the sought PSD  $f(r)$  by using a set of triangular base functions [see Eq. (30)] which discretizes the system of integral equations [see Eqs. (1)–(2)] because these equations cannot be solved analytically. The substitution of Eq. (30) into Eqs. (1)–(2) gives us for the case of volume PSD:

$$g_p = l_\lambda(m) = \sum_{j=1}^{N_B} A_{pj,v}(m, \lambda) f_{j,v} + \varepsilon_p, \quad (32)$$

where the symbol  $l$  represents the type of data ( $\beta$  is the backscatter and  $\alpha$  is the extinction coefficient, respectively),

$$A_{pj,v}(m, \lambda) = \int_{r_{j-1}}^{r_{j+1}} \frac{K_l(m, r, \lambda)}{\frac{4\pi}{3} r^3} B_j(r) dr, \quad (33)$$

and  $\varepsilon_p$  is the error from a variety of different sources, including measurement error and approximation error. One source of approximation error is the use of only a few base functions

$B_j(r)$  that are defined on a finite radius range  $[r_{\min}, r_{\max}]$  [see Eq. (30) and crosshatched areas in Fig. 3]. The index  $p = 1, \dots, N_O$  states the sequential number of input optical coefficient of type  $l$  at wavelength  $\lambda$  in the vector of experimental data  $\mathbf{g} = [g_p]$  and in the vector of uncertainty  $\boldsymbol{\varepsilon} = [\varepsilon_p]$ . The amount of available inputs  $N_O$  is equal to five for the HSRL-2 instrument. We use the volume kernel functions in Eq. (33) as we recommended before because the optical effects of ambient aerosols are more closely related to their volume rather than their number or surface area [24, 32]. The numerical integration in Eq. (33) is done using Simpson's rule [7].

*Note.* In the inversion parameters ASCII file the type of used kernel functions can be selected using the parameter “KernelType”:

// Type of kernel functions to be used during inversion.

// N - number kernel functions.

// S - surface area kernel functions.

// V - volume kernel functions.

KernelType=V

The step of numerical integration of the matrix  $\mathbf{A}$  elements can be selected using the parameter “KernelStep”:

// Integration step during generation of integral equation matrix, mum.

KernelStep=0.001

We can rewrite Eq. (32) as a vector–matrix equation:

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \boldsymbol{\varepsilon}, \quad (34)$$

where the elements of rectangular weight matrix  $\mathbf{A} = [A_{pj,V}]$  are calculated from Eq. (33), and the unknown vector  $\mathbf{f} = [f_{j,V}]$  needs to be found.

It is well known that the underlying inverse mathematical problem is ill-posed because of the relatively small number of measurement wavelengths compared to the total number of unknown quantities [4, 6, 29]. In general, the system has a null space where there is insufficient information content in the measurements to accurately determine the state. In that case, the solution will be non-unique as the different combinations of target parameters may lead to the similar optical spectra within the measurement uncertainty [6]. Equation (34) is inverted to produce the weight factors as:

$$\mathbf{f} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{g} + \boldsymbol{\varepsilon}', \quad (35)$$

wherein  $\boldsymbol{\varepsilon}'$  includes null space error as well as propagated measurement error and approximation error,  $\mathbf{A}^T$  is the transposed matrix of  $\mathbf{A}$  and  $(\mathbf{A}^T\mathbf{A})^{-1}$  denotes the inverse matrix of  $\mathbf{A}^T\mathbf{A}$ . The solution from Eq. (35) (ignoring the error term) potentially strongly oscillating since this is a solution that exactly fits the measured data, which in general are noisy. In other words, fitting the noise introduces oscillations [22]. The true solution is likely to be smoother than the exact fit of the noisy data, and yet fits the measurements within the measurement error, if the measurement errors are reported correctly. In general, even if the situation is favorable, such that the null space error is small, there are still many closely related solutions that fit the measurements to within

measurement error. Tikhonov regularization enables finding a smooth solution that is consistent with the measurements.

One physical reason for the instability is given by the small number of measurement wavelengths that are used to describe the highly complex structure of tropospheric aerosols, as well as the fact that only backscatter and extinction information is available. A related potential problem is numerical instability of the inversion, revealed by the analysis of the eigenvalues of  $\mathbf{A}^T\mathbf{A}$  and  $(\mathbf{A}^T\mathbf{A})^{-1}$  that show a high dynamic range across several orders of magnitude. The mechanism of inversion is the reconstruction of the investigated PSD by means of the eigenvectors of  $(\mathbf{A}^T\mathbf{A})^{-1}$ . The contribution of the different eigenvectors is determined by the respective eigenvalues of  $(\mathbf{A}^T\mathbf{A})^{-1}$ . Because small eigenvalues of  $\mathbf{A}^T\mathbf{A}$  correspond to large eigenvalues of  $(\mathbf{A}^T\mathbf{A})^{-1}$ , it follows that in the case of even small errors in optical data the error of the reconstructed distribution can be amplified by many orders of magnitude. Reason for this error amplification is that small eigenvalues are used for the reconstruction of PSD. At this stage Tikhonov regularization is introduced, which suppresses small eigenvalues and thus unstable solutions [25, 26]:

$$\mathbf{A}^T\mathbf{g} = (\mathbf{A}^T\mathbf{A} + \gamma\mathbf{H})\mathbf{f}, \quad (36)$$

where the smoothing matrix  $\mathbf{H}$  constrains the oscillations of retrieved PSD, and the regularization parameter  $\gamma$  defines the strength of smoothing [11, 14, 15, 16, 21, 25, 26, 27, 28]. In this section we excluded the uncertainty vector  $\boldsymbol{\varepsilon}$  from Eq. (36) to consider it separately in the following section. Finally, the vector  $\mathbf{f}$  of weight coefficients can be derived from the following relation:

$$\mathbf{f} = (\mathbf{A}^T\mathbf{A} + \gamma\mathbf{H})^{-1}\mathbf{A}^T\mathbf{g}. \quad (37)$$

In order to solve the vector–matrix Eq. (37) with the help of the Gauss-Jordan method [1], we discretize the search space for CRI and PSD. This discretization gives us  $N_S$  pairs of CRI and PSD, each of them resulting in a mathematical solution of Eq. (37). Every individual solution number  $k$  is characterized by an inversion window  $[r_{\min}^{(k)}, r_{\max}^{(k)}]$ , a CRI  $m^{(k)} = m_R^{(k)} - i \cdot m_I^{(k)}$ , a regularization parameter  $\gamma^{(k)}$ , and a corresponding vector  $\mathbf{f}^{(k)}$  of the results obtained with Tikhonov regularization. Let us note that following the selected mathematical approach we obtain the solution space of the CRI implicitly if we solve for the investigated PSD. The vector  $\mathbf{g}^{(k)} = [g_p^{(k)}]$  of back-calculated optical coefficients is obtained by using the weight matrix  $\mathbf{A}^{(k)}$ :

$$\mathbf{g}^{(k)} = \mathbf{A}^{(k)}|\mathbf{f}^{(k)}|, \quad (38)$$

where the symbol  $|\cdot|$  means that every element of the vector  $\mathbf{f}^{(k)}$  is converted into its absolute value. By using absolute values we take advantage of the physical constraint that the retrieved PSD has to be positive. We would like to point out that the solutions  $\mathbf{f}^{(k)}$  found in that way could contain some negative oscillations because the constraint of non-negativity is applied only to the back-calculated optical coefficients, but not to the solutions themselves. According to our experience, these negative oscillations mostly happen at the left and right tails of the inversion window.

The quality of each mathematical solution is evaluated using the normalized discrepancy because we naturally demand that the back-calculated optical data match the input optical data as closely as possible [11, 14, 15, 16, 27, 28]:

$$\rho^{(k)} = \frac{1}{N_0} \sum_{j=1}^{N_0} \frac{|g_j^{(k)} - g_j|}{g_j} 100\%. \quad (39)$$

For each mathematical solution we have to determine the regularization parameter  $\gamma^{(k)}$  for which the distance  $\rho^{(k)}$  between the vector of input optical data  $\mathbf{g}$  and the vector of back-calculated optical data  $\mathbf{g}^{(k)}$  reaches its minimum. This leads us to the minimization concept which allows us to find the regularization parameter  $\gamma^{(k)}$  that brings the minimum distance  $\rho^{(k)}$  for each individual mathematical solution. The constraint of smoothness is implemented in Eq. (37) through an additional penalty term  $\gamma^{(k)}\mathbf{H}$ . The relative difference between the magnitude of the elements  $\mathbf{A}^T\mathbf{A}$  and  $\gamma^{(k)}\mathbf{H}$  controls the smoothing effect. In the general case, if the regularization parameter is chosen too large then the solution PSD is oversmoothed, and for small  $\gamma^{(k)}$  the PSD oscillates, i.e., undersmoothed. Thus, the range of values of the regularization parameter has to be selected carefully in order to control the degree of regularization.

Figure 4.b shows a qualitative example of the typical shape of Tikhonov regularization curve that helps to find the ideal degree of smoothing. The optimum solution for the weight factors is defined by the global minimum of the regularization curve (case C). Figure 4.a illustrates in a qualitative manner the effect of regularization parameter  $\gamma$  in the inversion for a monomodal logarithmic-normal PSD function (see Section “6.2. Logarithmic-Normal Particle Size Distribution”). Curve D shows the given (correct) distribution from which the  $3\beta+2\alpha$  optical data were calculated. In the inversion, case A shows insufficient smoothing and thus oscillations because  $\gamma$  has been chosen too low. Case B leads to a strong oversmoothing because  $\gamma$  has been chosen too high. Case C results in an appropriate smoothing because of an optimally selected value of  $\gamma$ .

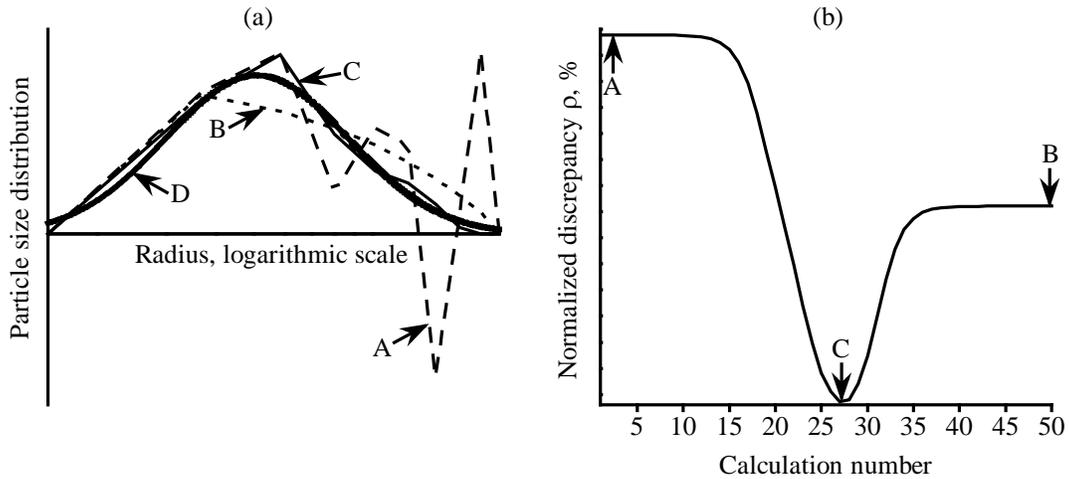


Figure 4. a) Effect of the value of regularization parameter  $\gamma$  on the quality of inversion. The undersmoothed (curve A), oversmoothed (curve B), and optimally smoothed (curve C) results are shown for a prescribed monomodal logarithmic-normal PSD (curve D).  
 b) Minimization of the normalized discrepancy  $\rho$  as a function of the regularization parameter  $\gamma$ . Calculation number “I” is related to the regularization parameter as  $\gamma = 10^{-28} \cdot 2^I$ .

#### 4.4.1. Regularization Parameter $\gamma$

Tikhonov regularization scheme requires using a small non-negative regularization parameter  $\gamma$  [25]. A penalty term, which in TiARA is equivalent to the smoothness of derivative of PSD in radius space, is multiplied by  $\gamma$ . In order to find the minimum of normalized discrepancy, the parameter  $\gamma$  is varied along several orders of magnitude. The only supported form of the regularization parameter  $\gamma$  is:

$$\gamma = b \cdot a^I, \quad (40)$$

where the recommended values of the constituents are:  $a = 2$ ,  $b = 10^{-28}$ , and the range of degree  $I = 1, \dots, 50$ . In that case, parameter  $\gamma$  varies from  $10^{-28}$  to about  $10^{-13}$  (see Fig. 5). It is assumed that  $\gamma = 0$  if  $I = 0$ .

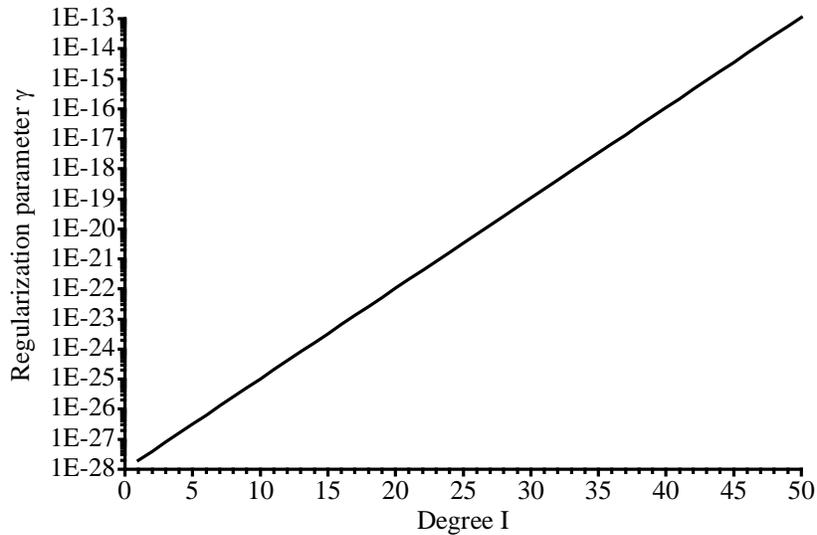


Figure 5. Dependence of the regularization parameter  $\gamma$  on the degree  $I$ .

*Note.* In the inversion parameters ASCII file the constituents of regularization parameter  $\gamma$  can be selected using the parameters “MinI”, “MaxI”, “ValueA”, and “ValueB”:

```

::::: Regularization parameter :::::
// gamma = 0          at I = 0.
// gamma = b*a^I     at I > 0.
// Minimum value of degree "I".
MinI=1

// Maximum value of degree "I".
MaxI=50

// Value of parameter "a".
ValueA=2

```

// Value of parameter "b".  
ValueB=1e-28

#### 4.4.2. Smoothing Matrix $\mathbf{H}$

We use smoothness of the retrieved PSD as a regularization factor in the inversion [26]. Smoothness can be computed in terms of a norm, which is defined by the distance in Euclidian space of successive base functions, weighted by the respective values of the weight factors [26]. More specifically, the smoothing matrix  $\mathbf{H}$  describes the mathematical constraint that the retrieved particle size distribution does not show large oscillations within an inversion window  $[r_{\min}, r_{\max}]$ . The different forms of  $\mathbf{H}$  influence the maximum difference between the weight factors of successive base functions [26]. Here we show the supported types of  $\mathbf{H}$  for the recommended case of eight base functions.

Form of the smoothing matrix  $\mathbf{H}$  that restricts the absolute values of the weight factors for base functions:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (41)$$

Form of the smoothing matrix  $\mathbf{H}$  that restricts the change of the weight factors of two successive base functions, i.e., smoothness of the PSD in the first derivative is required:

$$\mathbf{H} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}. \quad (42)$$

Form of the smoothing matrix  $\mathbf{H}$  that restricts the change of the weight factors of three successive base functions, i.e., smoothness of the PSD in the second derivative is required:

$$\mathbf{H} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}. \quad (43)$$

Form of the smoothing matrix  $\mathbf{H}$  that restricts the change of the weight factors of four successive base functions, i.e., smoothness of the PSD in the third derivative is required:

$$\mathbf{H} = \begin{bmatrix} 1 & -3 & 3 & -1 & 0 & 0 & 0 & 0 \\ -3 & 10 & -12 & 6 & -1 & 0 & 0 & 0 \\ 3 & -12 & 18 & -15 & 6 & -1 & 0 & 0 \\ -1 & 6 & -15 & 20 & -15 & 6 & -1 & 0 \\ 0 & -1 & 6 & -15 & 20 & -15 & 6 & -1 \\ 0 & 0 & -1 & 6 & -15 & 18 & -12 & 3 \\ 0 & 0 & 0 & -1 & 6 & -12 & 10 & -3 \\ 0 & 0 & 0 & 0 & -1 & 3 & -3 & 1 \end{bmatrix}. \quad (44)$$

For the purpose of microphysical parameters retrievals using experimental  $3\beta+2\alpha$  data, we recommend requiring smoothness of the PSD in the second derivative [see Eq. (43)].

*Note.* In the inversion parameters ASCII file the type of smoothing matrix  $\mathbf{H}$  can be selected using the parameter “SmoothingMatrixOrder”:

```
// Type of the smoothing matrix.
// 0 - restricting the absolute values of weight factors.
// 1 - smoothness in the first derivative.
// 2 - smoothness in the second derivative.
// 3 - smoothness in the third derivative.
SmoothingMatrixOrder=2
```

## 4.5. Selection and Averaging of Solutions

Due to the ill-posedness of the mathematical problem expressed by Eqs. (1)–(2), we average multiple solutions rather than selecting one single solution for the estimation of final microphysical products. Without additional constraints we could perfectly reproduce the optical input data from the derived weight coefficients  $\mathbf{f}$ , but these oscillate more than the true solution is believed to, due to overfitting of noise. The use of constraints on the one hand reduces the instability of the inversion problem and produces smoother solutions that are probably more realistic. On the other hand, the smoothed solutions do not perfectly reproduce the optical data. However, there are multiple solutions that satisfy the smoothness constraint and also still reproduce the measurements to within a certain range of uncertainty. We use these individual solutions from which we produce the mean value and the uncertainty bars.

Considering the aforementioned rationale, the final PSD  $\tilde{f}(r)$ , CRI  $\tilde{m} = \tilde{m}_R - i \cdot \tilde{m}_I$ , and all the microphysical parameters [see Eqs. (9)–(17)] are the result of averaging of limited number of mathematical solutions in a vicinity of the smallest value of normalized discrepancy among all of the solutions. We introduced the averaging because in ill-posed problems the solutions can be numerous and any quasi-solution  $\mathbf{f}^{(k)}$  with small normalized discrepancy  $\rho^{(k)}$  should be considered as an approximate solution of the inverse problem. The averaging procedure also takes advantage of the fact that different solutions can have oscillations of opposite sign and after averaging the mean oscillation might become much smaller compared with the oscillations of individual solutions. In some cases even the bimodality of structure of the final PSD becomes more obvious after averaging of many individual solutions compared with the appearance of individual solutions themselves.

In order to additionally stabilize the retrieval results, we apply constraints to the averaged solution space. The first constraint is the already mentioned selection of the vicinity around the global minimum of discrepancy as it defines what we denote as the discrepancy averaging interval  $\rho_{av}$  [27]. We use the discrepancy averaging interval in the sense that we take into account only a few percent of individual solutions of the total number of mathematical solutions that we obtain for all the inversion windows and the grid of CRIs tested in the inversion for a single optical data set.

The individual solutions of the whole solution space should fulfill a simple discrepancy test expressed as  $\rho^{(k)} < \rho_{av}$ . The exact value  $\rho_{av}$  of discrepancy that allows us to accept the corresponding individual solution as part of the final, averaged solution space depends on various factors. However, the use of only this constraint usually does not lead to a final solution, which is the average of individual solutions that fulfill  $\rho^{(k)} < \rho_{av}$ , with tolerable uncertainty. In fact, even a small value of the discrepancy  $\rho^{(k)}$  within this acceptable value of  $\rho_{av}$  can be linked to an individual solution with microphysical parameters that do not make sense from a physical point of view and thus has to be classified as outlier.

While data operators with specific long-term experience are able to identify such outliers, it is important for the retrieval to be automated and objective. Taking into consideration the huge amount of optical data sets that need to be processed, inversion software must run in an unsupervised mode, i.e., the averaging interval should be identified automatically. For that reason we introduced several modifications in order to allow for an automated unsupervised data analysis. These modifications simultaneously take into account the following constraints imposed on the solution space:

1. The number of solutions that are averaged is  $N_A$ . Usually  $N_A$  does not exceed a few percent of the total number of solutions for a given optical input data set. The recommended value of  $N_A$  is equal to 500. That value is approximately 1% of the total number of solutions of the whole solution space.
2. The discrepancy  $\rho_{av}$  is the number that defines all individual solutions with larger discrepancy and which are not considered in the averaging process. The discrepancy averaging interval is recommended to be less than 10% with regard to retrievals that use experimental lidar data.
3. We attempt to exclude outliers by limiting the effective radius and number concentration to solutions that are similar to each other. The strictness of the exclusion is controlled by approximate threshold percent differences  $\delta_{r_{eff}}$  and  $\delta_n$  that are used as described below. Usually we set these thresholds to 25% and 100%, respectively. Then, the averaging procedure involves the following steps:
  - 3.a. The initial step is to find the first individual solution of the final solution space. This first solution can be found from the minimum value of the discrepancy, defined as  $\rho^{(1)}$ , as it defines the values of effective radius  $r_{eff}^{(1)}$  and number concentration  $n^{(1)}$ .
  - 3.b. After that first step we select the number concentration  $n^{(2)}$  and effective radius  $r_{eff}^{(2)}$  that belong to  $\rho^{(2)}$ , which is the second smallest discrepancy value.
  - 3.c. We compare the values of  $r_{eff}^{(1)}$  and  $n^{(1)}$  and  $r_{eff}^{(2)}$  and  $n^{(2)}$ . If the relative deviations

$$\left| \frac{r_{\text{eff}}^{(1)} - r_{\text{eff}}^{(2)}}{r_{\text{eff}}^{(1)}} \right| 100\% < \delta_{r_{\text{eff}}} \quad \text{and} \quad \left| \frac{n^{(1)} - n^{(2)}}{n^{(1)}} \right| 100\% < \delta_n \quad (45)$$

do not exceed the respective thresholds  $\delta_{r_{\text{eff}}}$  and  $\delta_n$  then we average the effective radii of the 1<sup>st</sup> and 2<sup>nd</sup> individual solutions. The same is done for number concentration. If conditions (45) are not fulfilled simultaneously then the 2<sup>nd</sup> individual solution is ignored.

3.d. We continue with the selection procedure. We test the  $(k+1)^{\text{th}}$  individual solution that has the discrepancy  $\rho^{(k+1)}$ . This discrepancy value is the next smallest to the value of the discrepancy  $\rho^{(k)}$  of the  $k^{\text{th}}$  individual solution. We compare the parameters  $r_{\text{eff}}^{(k+1)}$  and  $n^{(k+1)}$  with the values  $\tilde{r}_{\text{eff}}$  and  $\tilde{n}$  of the previously averaged  $k_1$  individual solutions that passed through all the tests. The average values of effective radius and number concentration are defined as:

$$p = \frac{1}{k_1} \sum_{j=1}^{k_1} p^{(j)}, \quad p = r_{\text{eff}}, n, \quad k_1 \leq k. \quad (46)$$

If the relative deviations

$$\left| \frac{\tilde{r}_{\text{eff}} - r_{\text{eff}}^{(k+1)}}{\tilde{r}_{\text{eff}}} \right| 100\% < \delta_{r_{\text{eff}}} \quad \text{and} \quad \left| \frac{\tilde{n} - n^{(k+1)}}{\tilde{n}} \right| 100\% < \delta_n \quad (47)$$

do not exceed the respective thresholds  $\delta_{r_{\text{eff}}}$  and  $\delta_n$  then the  $(k+1)^{\text{th}}$  individual solution is used to update the average values of effective radius and number concentration. If conditions (47) are not fulfilled simultaneously then the  $(k+1)^{\text{th}}$  individual solution is ignored.

3.e. This selection process is continued as long as the number of averaged solutions is less than  $N_A$  and the discrepancy of the next individual solution is  $\rho^{(k)} \leq \rho_{\text{av}}$ .

Figure 6 shows a quantitative example of processing of one  $3\beta+2\alpha$  experimental optical data set. Blocks (a, c, e) show the statistical distribution of three microphysical size parameters for the first 500 of solutions with the smallest discrepancies if the  $\delta_{r_{\text{eff}}}$  |  $\delta_n$  constraints are not applied. One can see that the distributions for number and volume concentrations have very long tails consisting of outliers (see Figs. 6.a and 6.e). Blocks (b, d, f) show the statistical distribution of the same microphysical parameters if all the constraints are applied as described above. The tails of outliers are becoming much shorter (see Figs. 6.b, 6.d and 6.f), the tail-cut solution space is than used for the final averaging.

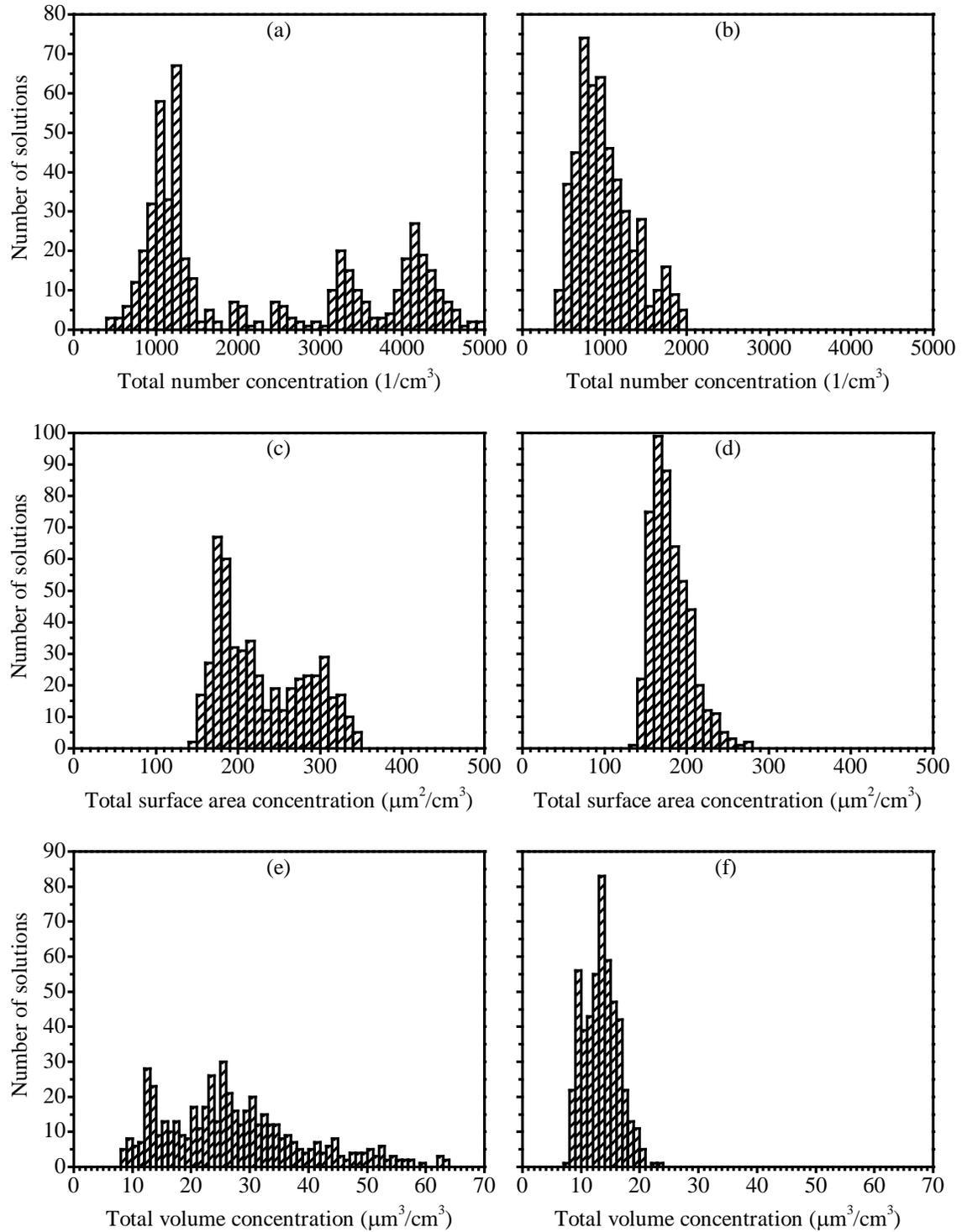


Figure 6. Effect of the  $\delta_{r_{\text{eff}}}|\delta_n$  constraints before (a, c, e) and after (b, d, f) being imposed on the averaged solution space.

After these post-processing routines we obtain a subset of the best individual solutions that is constrained by the number  $N_A$  and/or the maximum discrepancy  $\rho_{av}$ . The uncertainty of effective radius and number concentration, that arises from the averaged values, is limited by  $\delta_{r_{eff}}$  and  $\delta_n$ , respectively. The standard deviation of these best solutions provides a proxy measure of the retrieval uncertainty for any selected microphysical parameter.

*Note.* In the inversion parameters ASCII file the number  $N_A$  of solutions that are averaged can be selected using the parameter “SolutionsNumberPostProc”:

```
// Number of solutions to be included into post-processing.  
SolutionsNumberPostProc=500
```

The corresponding thresholds with regard to the deviation of individual solutions from the mean values can be set using the parameters “ReffUncertaintyPostProc”, “NumCUncertaintyPostProc”, and “ODUncertaintyPostProc”:

```
// Post-processing stage. Allowed uncertainty level for the effective radius, %.  
ReffUncertaintyPostProc=25
```

```
// Post-processing stage. Allowed uncertainty level for the number concentration, %.  
NumCUncertaintyPostProc=100
```

```
// Post-processing stage. Allowed uncertainty level for the optical data, %.  
ODUncertaintyPostProc=10
```

## 4.6. Extreme Error Computation Scheme

The retrieval uncertainties for microphysical parameters (see Section “2. Inputs and Outputs”) are the combination of uncertainties resulting from the measurement uncertainties of  $3\beta+2\alpha$  optical data and the mathematical uncertainties. A full account of the uncertainties introduced by the TiARA inversion algorithm itself cannot be made at the moment. This complex topic covers several challenges, as for example: non-uniqueness of inverse solution in the majority of cases, choice of grid size of the CRI, assumption that the CRI is wavelength- and size-independent in the retrievals, choice of number and position of inversion windows, choice of shape of base function, choice of regularization method, correct assessment of regularization parameter and so on, just to mention the most obvious sources of uncertainty. These effects are still under investigation.

It is also a challenge to process measurement errors in an ill-posed system in a realistic manner. Measurement uncertainties propagate during the inversion process in a way that does not easily allow for an analytical expression of the effect of measurement uncertainties on the final data products. Microphysical parameters change, but not necessarily in a way proportional to the change of input optical data  $\mathbf{g}$ . One of the characteristics of ill-posed inverse problems is that a small amount of uncertainties or even the absence of uncertainties in the input data can lead to large errors in the final data products. There remain the questions how to treat measurements uncertainties in uncertainties analysis, how the error distribution function affects the quality of the inversion results, and how big measurement uncertainties can be until the inversion results become meaningless. The common approach found in mathematical literature is unsuitable for our purposes because the input data uncertainties often do not exceed 1% in these sensitivity studies [9, 20, 30, 31], which is an unrealistic assumption for remote sensing applications.

Considering the aforementioned reasons, the error propagation in TiARA is currently estimated in a numerical fashion: we invert the measured optical data set  $\mathbf{g}$  and then distort the optical data [ $\mathbf{g} - \boldsymbol{\varepsilon} = \mathbf{g}_d = \mathbf{A}\mathbf{f}$  as follows from Eq. (34)] within the error bars  $\boldsymbol{\varepsilon}$  for a few more inversion runs. This procedure is time consuming but allows us to compute uncertainties of the data products on the basis of a very simple scheme that we denote as “extreme error computation model” (EEM). Table 2 shows the basic principle of this scheme. For example, run number one describes the situation in which backscatter and extinction optical coefficients at the wavelengths 355 and 532 nm are overestimated by the corresponding components of  $\boldsymbol{\varepsilon}$ , whereas the backscatter coefficient at 1064 nm is underestimated by  $\varepsilon_{\beta_{1064}}$ .

Table 2. Distortion of the experimental optical data in each measurement channel in the case of EEM. The + sign indicates that a data point in one of the five measurement channels is changed by +x. The – sign indicates that a data point is changed by –x. Run number denotes the eight different inversion runs that we carry out for each optical data set.

Run number	$\beta_{355}$	$\beta_{532}$	$\beta_{1064}$	$\alpha_{355}$	$\alpha_{532}$
1	$+\varepsilon_{\beta_{355}}$	$+\varepsilon_{\beta_{532}}$	$-\varepsilon_{\beta_{1064}}$	$+\varepsilon_{\alpha_{355}}$	$+\varepsilon_{\alpha_{532}}$
2	$+\varepsilon_{\beta_{355}}$	$+\varepsilon_{\beta_{532}}$	$-\varepsilon_{\beta_{1064}}$	$-\varepsilon_{\alpha_{355}}$	$-\varepsilon_{\alpha_{532}}$
3	$+\varepsilon_{\beta_{355}}$	$+\varepsilon_{\beta_{532}}$	$-\varepsilon_{\beta_{1064}}$	$+\varepsilon_{\alpha_{355}}$	$-\varepsilon_{\alpha_{532}}$
4	$+\varepsilon_{\beta_{355}}$	$+\varepsilon_{\beta_{532}}$	$-\varepsilon_{\beta_{1064}}$	$-\varepsilon_{\alpha_{355}}$	$+\varepsilon_{\alpha_{532}}$
5	$-\varepsilon_{\beta_{355}}$	$-\varepsilon_{\beta_{532}}$	$+\varepsilon_{\beta_{1064}}$	$+\varepsilon_{\alpha_{355}}$	$+\varepsilon_{\alpha_{532}}$
6	$-\varepsilon_{\beta_{355}}$	$-\varepsilon_{\beta_{532}}$	$+\varepsilon_{\beta_{1064}}$	$-\varepsilon_{\alpha_{355}}$	$-\varepsilon_{\alpha_{532}}$
7	$-\varepsilon_{\beta_{355}}$	$-\varepsilon_{\beta_{532}}$	$+\varepsilon_{\beta_{1064}}$	$+\varepsilon_{\alpha_{355}}$	$-\varepsilon_{\alpha_{532}}$
8	$-\varepsilon_{\beta_{355}}$	$-\varepsilon_{\beta_{532}}$	$+\varepsilon_{\beta_{1064}}$	$-\varepsilon_{\alpha_{355}}$	$+\varepsilon_{\alpha_{532}}$

Theoretically, except the case of 0% error, we find 242 error combinations that we would need to test. For practical reasons (computation time, unreasonable distortion of backscatter and extinction spectra that are not possible from a physical point of view) we carry out only nine inversion runs for each optical data set. First run is for the measured data  $\mathbf{g}$  and eight runs are for the backscatter and extinction spectra distorted according to Table 2. Based on our experience we assume that nine inversion runs is just about sufficient to give us statistically meaningful results. The rationale behind the EEM is that the true optical data are not known; only the measured lidar data  $\mathbf{g}$  are available. The true optical data can be anything around  $\mathbf{g}$  within the uncertainties  $\boldsymbol{\varepsilon}$  [see Eq. (34)]. The distorted data that are used as input parameters during any of the eight extra inversion runs might be closer to the true optical data compared with the measured  $\mathbf{g}$ . The mathematical solutions resulting from all nine inversion runs are treated equally and considered in the final averaging of the individual solutions.

One point should be mentioned. We do not consider correlations among the five measurement channels, i.e., we treat the measurement uncertainty of each channel independently of each other. The EEM type of error may introduce “distortions” in the spectra of the Ångström exponents and lidar ratios. For example, let us assume that the lidar ratio at 355 nm is larger than the lidar ratio at 532 nm in the case of the measured data. It may happen that the application of the EEM may

cause a change of the spectral dependence of the lidar ratio if measurement errors are introduced. It is assumed that from a measurement point of view this “distortion” of the spectrum of the lidar ratio is acceptable as it would be within the measurement uncertainties.

*Note.* In the inversion parameters ASCII file the extreme error scheme can be activated using the parameter “UseExtremeDistortion”:  
// Usage of "extreme" noise for the input optical data distortion.  
// If "0" then "extreme" distortion will not be used.  
// If "1" then "extreme" distortion will be used.  
UseExtremeDistortion=1

The error bars  $\epsilon$  for the HSRL-2 optical data are estimated on the basis of measured lidar signals, delivered as a standard product, and used by TiARA for the microphysical retrievals. For the purpose of simulations the components of error bars vector  $\epsilon$  can be set individually in the inversion parameters ASCII file for each measurement channels using corresponding parameters:

```
***** Extreme error levels *****
:::: Backscatter coefficients ::::
// Value of "extreme" distortion for backscatter coefficient No1, %.
BackscatterExtreme01=10
// Value of "extreme" distortion for backscatter coefficient No2, %.
BackscatterExtreme02=5
// Value of "extreme" distortion for backscatter coefficient No3, %.
BackscatterExtreme03=15
// Value of "extreme" distortion for backscatter coefficient No4, %.
BackscatterExtreme04=0
// Value of "extreme" distortion for backscatter coefficient No5, %.
BackscatterExtreme05=0
// Value of "extreme" distortion for backscatter coefficient No6, %.
BackscatterExtreme06=0
// Value of "extreme" distortion for backscatter coefficient No7, %.
BackscatterExtreme07=0
// Value of "extreme" distortion for backscatter coefficient No8, %.
BackscatterExtreme08=0
// Value of "extreme" distortion for backscatter coefficient No9, %.
BackscatterExtreme09=0
// Value of "extreme" distortion for backscatter coefficient No10, %.
BackscatterExtreme10=0

:::: Extinction coefficients ::::
// Value of "extreme" distortion for extinction coefficient No1, %.
ExtinctionExtreme01=10
// Value of "extreme" distortion for extinction coefficient No2, %.
ExtinctionExtreme02=10
// Value of "extreme" distortion for extinction coefficient No3, %.
ExtinctionExtreme03=0
// Value of "extreme" distortion for extinction coefficient No4, %.
ExtinctionExtreme04=0
// Value of "extreme" distortion for extinction coefficient No5, %.
ExtinctionExtreme05=0
// Value of "extreme" distortion for extinction coefficient No6, %.
```

```
ExtinctionExtreme06=0
// Value of "extreme" distortion for extinction coefficient No7, %.
ExtinctionExtreme07=0
// Value of "extreme" distortion for extinction coefficient No8, %.
ExtinctionExtreme08=0
// Value of "extreme" distortion for extinction coefficient No9, %.
ExtinctionExtreme09=0
// Value of "extreme" distortion for extinction coefficient No10, %.
ExtinctionExtreme10=0
```

## 4.7. Summary of Inversion Variables

We designed the inversion code in a way that user has control over the major aspects of TiARA. In the text we listed the recommended settings of algorithm in the “*Note*” paragraphs of each section. These settings might be changed according to the features of the measurement session or following the user preferences.

Generally, we recommend narrowing down the search space as much as possible. If it is known that the coarse mode particles are not present then the inversion windows should not cover large particles (see Section “4.1. Definition of the Search Space”). If the type of aerosols is known then the search space for CRI should be set to cover only the area of the CRI’s that are typical for these particular aerosols (see Section “4.1. Definition of the Search Space”).

We spent quite a lot of time running simulations in order to find the optimal range of values for the regularization parameter  $\gamma$  (see Section “4.4.1. Regularization Parameter  $\gamma$ ”) and the type of smoothing matrix  $\mathbf{H}$  (see Section “4.4.2. Smoothing Matrix  $\mathbf{H}$ ”). We do not expect that something can be improved in this part of TiARA, but user has an option to adjust these settings too. As a result of adjustments, the smoothness of PSD will be affected.

Our recommendations for the other adjustable parameters of algorithm like the number of triangular base functions  $N_B$  (see Section “4.3. Base Functions”), the type of distribution of radius grid bins (see Section “4.2. Radius Grid Bins”), the type of used kernel functions (see Section “4.4. Tikhonov Regularization”), and the post-processing thresholds (see Section “4.5. Selection and Averaging of Solutions”) are also based on our experience of running simulations and experimental data processing. We do not expect to see any positive effect if these settings will be changed by user.

It is known that the solution of inverse problem is not unique [4, 6, 29]. In the other words, even if the input optical data  $\{\beta_{355}, \beta_{532}, \beta_{1064}, \alpha_{355}, \alpha_{532}\}$  are well reproduced and the value of normalized discrepancy is small (see Section “4.4. Tikhonov Regularization”) then it does not necessarily mean that the found solution is the true solution. Despite this, the smaller values of normalized discrepancy are preferable because at least it guaranties that there is a way to fit the lidar measured data into the Mie-scattering theory [2]. Extreme error computation scheme (see Section “4.6. Extreme Error Computation Scheme”) often helps to slightly reduce the final values of normalized discrepancies and improve the overall quality of microphysical retrievals. This scheme uses the uncertainties of input optical data as a source of extra information. The quality of this information directly influences the efficiency of scheme.

## 5. Details of Implementation

### 5.1. Optimized Look-Up Table

The most time consuming procedure during the microphysical parameters retrieval is the calculation of the elements of matrix  $\mathbf{A}$  with the help of Eq. (33). Our estimations show that the direct computation of the light-scattering kernel functions  $K_l(m, r, \lambda)$  consumes more than 90% of the whole inversion time budget. In order to reduce the computation time, we improved the efficiency of our algorithm by using an optimized look-up table (LUT). The elements of this LUT are the elements of matrix  $\mathbf{A}$  pre-integrated for the set of triangular base functions and the grid of CRIs.

TiARA's LUT takes advantage of the fact that the Mie kernel functions are dependent on the size parameter  $x = 2\pi r/\lambda$  rather than directly on the particle radius  $r$ . That is, we can use the following properties of Eqs. (18)–(23):

$$Q_l(m, r, \lambda_0) = Q_l\left(m, \frac{\lambda}{\lambda_0}r, \lambda\right), \quad (48)$$

and Eqs. (1)–(2), (7)–(8):

$$\int_{r_{\min}}^{r_{\max}} K_l(m, r, \lambda_0)b(r)dr = \left(\frac{\lambda_0}{\lambda}\right)^3 \int_{\frac{\lambda}{\lambda_0}r_{\min}}^{\frac{\lambda}{\lambda_0}r_{\max}} K_l(m, r, \lambda)B(r)dr, \quad (49)$$

$$\int_{r_{\min}}^{r_{\max}} K_{l,S}(m, r, \lambda_0)b(r)dr = \frac{\lambda_0}{\lambda} \int_{\frac{\lambda}{\lambda_0}r_{\min}}^{\frac{\lambda}{\lambda_0}r_{\max}} K_{l,S}(m, r, \lambda)B(r)dr, \quad (50)$$

$$\int_{r_{\min}}^{r_{\max}} K_{l,V}(m, r, \lambda_0)b(r)dr = \int_{\frac{\lambda}{\lambda_0}r_{\min}}^{\frac{\lambda}{\lambda_0}r_{\max}} K_{l,V}(m, r, \lambda)B(r)dr, \quad (51)$$

where  $\lambda_0$  is the reference wavelength which is 355 nm for our LUT, and  $\lambda$ , theoretically, is any other wavelength, but equal to 532 or 1064 nm for the case of HSRL-2 instrument. The term  $K_{l,*}(m, r, \lambda)$  describes the backscatter ( $l = \beta$ ), extinction ( $l = \alpha$ ), absorption ( $l = \text{abs}$ ), and scattering ( $l = \text{sca}$ ) number (no subscript), surface area (S), and volume (V) kernel functions that are calculated using Mie-scattering theory [2, 3]. The triangular functions  $b(r)$  and  $B(r)$  are defined on the radii ranges  $[r_{\min}, r_{\max}]$  and  $[\frac{\lambda}{\lambda_0}r_{\min}, \frac{\lambda}{\lambda_0}r_{\max}]$  respectively.

We created three LUTs; we used Eq. (49) for the number, Eq. (50) for the surface area, and Eq. (51) for the volume kernel functions. Each LUT consists of four separate files and contains the results of the numerical integration of the left side of Eqs. (49)–(51) for the corresponding backscatter, extinction, absorption, and scattering kernel functions, respectively.

For all three LUTs the CRI  $m$  ranges from 1.2 to 2 with stepsize 0.005 for the real part ( $m_R$ ), and from 0 to 0.5 with stepsize 0.0005 for the imaginary part ( $m_I$ ). We split the particle radius range from 10 nm to 10  $\mu\text{m}$  into 71 logarithmic-equidistant grid bins  $r_0, \dots, r_{70}$  [see Eq. (29) and Table 3]. These 71 grid bins allow us to form 69 integration intervals  $[r_{j_1-1}, r_{j_1+1}]$ , where  $j_1 = 1, \dots, 69$ . Each interval has three grid bins  $r_{j_1-1}$ ,  $r_{j_1}$ , and  $r_{j_1+1}$  that are used to create a triangular base function  $b_{j_1}(r)$  [see Fig. 7]. Then we did the integration with corresponding kernel functions following the left-hand side of Eqs. (49)–(51). This procedure leads to the LUT elements in the form of:

$$[LUT_{j_1}(m_{j_2 j_3})]_l = \int_{r_{j_1-1}}^{r_{j_1+1}} K_l(m_{j_2 j_3}, r, \lambda_0) b_{j_1}(r) dr, \quad (52)$$

$$[LUT_{j_1}(m_{j_2 j_3})]_{l,S} = \int_{r_{j_1-1}}^{r_{j_1+1}} K_{l,S}(m_{j_2 j_3}, r, \lambda_0) b_{j_1}(r) dr, \quad (53)$$

$$[LUT_{j_1}(m_{j_2 j_3})]_{l,V} = \int_{r_{j_1-1}}^{r_{j_1+1}} K_{l,V}(m_{j_2 j_3}, r, \lambda_0) b_{j_1}(r) dr, \quad (54)$$

where  $m_{j_2 j_3} = m_{R,j_2} - i \cdot m_{I,j_3}$  is the discretized CRI with real part  $m_{R,j_2} = 1.2 + j_2 \cdot 0.005$  for  $j_2 = 0, \dots, 160$  and imaginary part  $m_{I,j_3} = j_3 \cdot 0.0005$  for  $j_3 = 0, \dots, 1000$ . Thus, each of the four files of each of the three LUTs contains  $69 \cdot 161 \cdot 1001 = 11,120,109$  elements. The integration was done with the radius step 1 nm using Simpson's rule [7].

Table 3: Radius grid bins of the LUT.

Bin #	Radius ( $\mu\text{m}$ )						
0	0.01	18	0.059078379115879445	36	0.34902548789595805	54	2.0619860095022204
1	0.011037155032027573	19	0.065205722934286142	37	0.38522484200367518	55	2.2758459260747883
2	0.012181879120101156	20	0.071968567300115208	38	0.42517863033828901	56	2.5118864315095801
3	0.013445328842997610	21	0.079432823472428152	39	0.46927624593488376	57	2.7724079967417743
4	0.014839817889675653	22	0.087671238729686823	40	0.51794746792312107	58	3.0599496872071956
5	0.016378937069540640	23	0.096764105370945347	41	0.57166665019136154	59	3.3773139087910096
6	0.018077686769634345	24	0.10680004325145757	42	0.63095734448019325	60	3.7275937203149403
7	0.019952623149688795	25	0.11787686347935873	43	0.69639740296243191	61	4.1142029787528420
8	0.022022019499873754	26	0.13010252169108313	44	0.76862461003977389	62	4.5409096109724763
9	0.024306044333844090	27	0.14359617019622148	45	0.84834289824407205	63	5.0118723362727229
10	0.026826957952797256	28	0.15848931924611134	46	0.93632920882394144	64	5.5316811976172264
11	0.029609329396270836	29	0.17492711874398423	47	1.0334410638805562	65	6.1054022965853285
12	0.032680275894101260	30	0.19306977288832503	48	1.1406249238513209	66	6.7386271680309458
13	0.036069727153262905	31	0.21309410153667971	49	1.2589254117941673	67	7.4375272756590460
14	0.039810717055349727	32	0.23519526350709588	50	1.3894954943731377	68	8.2089141596382564
15	0.043939705607607911	33	0.25958865861263941	51	1.5336077187700115	69	9.0603058224533761
16	0.048496934285281984	34	0.28651202696637806	52	1.6926666150378760	70	10
17	0.053526818228471051	35	0.31622776601683794	53	1.8682223847710371		

For practical purposes, we normally form each of the  $N_B$  triangular base functions  $B_j(r)$  [see Eq. (33)] by using an odd number of the triangular base functions  $b_{j_1}(r)$  of the LUT. The radius grid bins  $r_{j-1}$ ,  $r_j$ , and  $r_{j+1}$  of base function  $B_j(r)$  should coincide with the grid bins of LUT. Figure 7 shows a qualitative example when three LUT base functions  $b_{j_1}(r)$  are used to build one base function  $B_j(r)$  that mathematically can be expressed as:

$$B_j(r) = \frac{1}{2} b_{j_1}(r) + b_{j_1+1}(r) + \frac{1}{2} b_{j_1+2}(r). \quad (55)$$

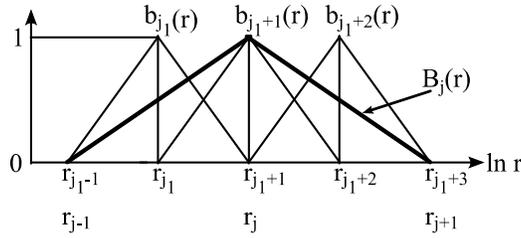


Figure 7. Example of the linear combination of three LUT triangular base functions  $b_{j_1}(r)$ ,  $b_{j_1+1}(r)$ , and  $b_{j_1+2}(r)$  into one inversion triangular base function  $B_j(r)$  [see, for instance, option #52 (inversion window #33) in Table 4. Part 2 of 3].

The term  $B_j(r)$  in Eqs. (33) and (55) should be treated as  $B(r)$  on the right-hand side of one of the Eqs. (49)–(51). Thus, the result of the right-hand side integration of any of the Eqs. (49)–(51) can be expressed in a linear fashion through the left-hand side values that were pre-calculated and stored in the LUT. The other desired triangular base functions  $B_j(r)$  can be formed using another linear combination of the LUT base functions  $b_1(r), \dots, b_{69}(r)$ , but the total number of such base functions is limited and is defined by the user-supplied value  $N_B$ . Table 4 lists 80 inversion windows that can be formed for the case of  $N_B=8$  and using only the LUT radius grid bins [see Table 3]. Each column of Table 4 starting from the second shows ten positions of the LUT grid bins that might be used as an inversion window  $[r_{\min}^{(k)}, r_{\max}^{(k)}]$  [see Eq. (38)]. The red color coding means that the difference  $r_{\max}^{(k)} - r_{\min}^{(k)}$  is narrower than  $0.38 \mu\text{m}$  or  $r_{\min}^{(k)} > 0.3 \mu\text{m}$ , and the inversion window is rejected as nonphysical [see Table 4, all three parts]. The blue color coding shows the inversion windows that have no restrictions and can be applied [see Table 4, all three parts]. The upper numbers in the first row of Table 4 show the sequential number of an inversion window that potentially might be used for the inversion. The lower numbers are shown in bold and track the sequential number of the inversion window that we accepted to be used.

Following the same principle as used in Eq. (55), five LUT base functions  $b_{j_1}(r)$  can be used to build one base function  $B_j(r)$  as [see, for instance, option #53 (inversion window #34) in Table 4. Part 2 of 3]:

$$B_j(r) = \frac{1}{3} b_{j_1-1}(r) + \frac{2}{3} b_{j_1}(r) + b_{j_1+1}(r) + \frac{2}{3} b_{j_1+2}(r) + \frac{1}{3} b_{j_1+3}(r). \quad (56)$$

Seven LUT base functions  $b_{j_1}(r)$  can be used to build one base function  $B_j(r)$  as [see, for instance, option #54 (inversion window #35) in Table 4. Part 2 of 3]:

$$B_j(r) = \frac{1}{4}b_{j_1-2}(r) + \frac{2}{4}b_{j_1-1}(r) + \frac{3}{4}b_{j_1}(r) + b_{j_1+1}(r) + \frac{3}{4}b_{j_1+2}(r) + \frac{2}{4}b_{j_1+3}(r) + \frac{1}{4}b_{j_1+4}(r). \quad (57)$$

Nine LUT base functions  $b_{j_1}(r)$  can be used to build one base function  $B_j(r)$  as [see, for instance, option #55 (inversion window #36) in Table 4. Part 2 of 3]:

$$B_j(r) = \frac{1}{5}b_{j_1-3}(r) + \frac{2}{5}b_{j_1-2}(r) + \frac{3}{5}b_{j_1-1}(r) + \frac{4}{5}b_{j_1}(r) + b_{j_1+1}(r) + \frac{4}{5}b_{j_1+2}(r) + \frac{3}{5}b_{j_1+3}(r) + \frac{2}{5}b_{j_1+4}(r) + \frac{1}{5}b_{j_1+5}(r). \quad (58)$$

Eleven LUT base functions  $b_{j_1}(r)$  can be used to build one base function  $B_j(r)$  as [see, for instance, option #6 (inversion window #4) in Table 4. Part 1 of 3]:

$$B_j(r) = \frac{1}{6}b_{j_1-4}(r) + \frac{2}{6}b_{j_1-3}(r) + \frac{3}{6}b_{j_1-2}(r) + \frac{4}{6}b_{j_1-1}(r) + \frac{5}{6}b_{j_1}(r) + b_{j_1+1}(r) + \frac{5}{6}b_{j_1+2}(r) + \frac{4}{6}b_{j_1+3}(r) + \frac{3}{6}b_{j_1+4}(r) + \frac{2}{6}b_{j_1+5}(r) + \frac{1}{6}b_{j_1+6}(r). \quad (59)$$

In order to increase the total number of available inversion windows and cover the larger particles, we introduced 68 intermediate inversion windows that are listed in the first row of Table 5 as lower numbers (shown in bold). We estimate these intermediate values  $I_v$  from the LUT's pre-integrated entries by linear interpolation:

$$[Iv_{j_1}(m_{j_2j_3})]'_l \approx [LUT_{j_1}(m_{j_2j_3})]_l + \frac{[LUT_{j_1}(m_{j_2j_3})]_{l+1} - [LUT_{j_1}(m_{j_2j_3})]_l}{\ln r_{l+1} - \ln r_l} (\ln r'_l - \ln r_l), \quad (60)$$

where  $r'_l$  is the intermediate radius located between the  $r_l$  and  $r_{l+1}$  LUT's radii [see Table 3].

The green color coding in Table 5 shows the range  $[r_{\min}, r_{\max}]$  that is covered by the intermediate inversion windows. The blue color coding refers to the original accepted inversion window with the sequential number shown in bold in Table 4, which corresponds to the upper numbers in the first row of Table 5. We keep the  $r_{\min}$  border of the intermediate inversion window such that it coincides with the host original inversion window. The  $r_{\max}$  border is shown as white number for each inversion window and can have the values 5.9, 6.9, 7.9, 8.9, or 9.9  $\mu\text{m}$  in our case [see Table 5].

We use the LUTs not only to compute the backscatter and extinction coefficients but also the absorption and scattering coefficients. Backscatter and extinction kernels are used for the inversion itself. We introduced absorption and scattering kernels in order to speed up the computations of the absorption and scattering coefficients, and the single-scattering albedo as retrieved microphysical parameters that are computed from the respective PSD and CRI. The LUT is uploaded to the operational memory of a computer, which allows us to further reduce the data processing time.

*Note.* In the inversion parameters ASCII file the use of optimized look-up table can be activated by using the parameters “UseOptimizedDataBank” and “OptimizedDataBankName”:

// Usage of optimized look-up table.

// If "0" then Mie-scattering code will be used.

// If "1" then optimized look-up table will be used.

UseOptimizedDataBank=1

// The file name of optimized look-up table.

OptimizedDataBankName=69V\_1.2-0.005-2\_i0-i0.0005-i0.5

Table 4: Inversion windows for the case of  $N_B$  equal to eight base functions. Part 1 of 3\*

Radius ( $\mu\text{m}$ )	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.01																																												
0.011037																																												
0.012182																																												
0.013445																																												
0.014840																																												
0.016379																																												
0.018078																																												
0.019953																																												
0.022022																																												
0.024306																																												
0.026827																																												
0.029609																																												
0.032680	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.036070	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.039811	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.043940	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.048497	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.053527	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.059078	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.065206	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.071969	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.079433	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.087671	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.096764	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.106800	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.117877	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.130103	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.143596	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.158489	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.174927	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.193070	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.213094	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.235195	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.259589	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.286512	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.316228	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.349026	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.385225	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.425179	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.469276	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.517948	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.571667	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.630957	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.696397	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.768625	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
0.848343	1	2	3	4	5	6	7	8	9	10																																		









## 5.2. Calculation of Microphysical Size Parameters from Pre-Integration of Triangular Base Functions

We approximate the PSD  $f(r)$  by a set of triangular base functions  $B_j(r)$  as [see Eq. (30)]:

$$f(r) = \sum_{j=1}^{N_B} f_j B_j(r) + \varepsilon^{\text{appr}}(r). \quad (61)$$

For simplicity, let us neglect the approximation error  $\varepsilon^{\text{appr}}(r)$ . The triangular base functions  $B_j(r)$  can be pre-integrated analytically and used for the purpose of fast calculations of microphysical size parameters.

### 5.2.1. Total Number Concentration

The total number concentration for the PSD  $f(r)$  is calculated as:

$$n = \int_{r_{\min}}^{r_{\max}} f(r) dr = \sum_{j=1}^{N_B} f_j \int_{r_{j-1}}^{r_{j+1}} B_j(r) dr. \quad (62)$$

If the PSD  $f(r)$  was retrieved using number kernel functions then the total number concentration is calculated as:

$$n = \frac{1}{2} \sum_{j=1}^{N_B} f_j (r_{j+1} - r_{j-1}). \quad (63)$$

For the case of surface area kernel functions the total number concentration is calculated as:

$$n = \frac{1}{4\pi} \sum_{j=1}^{N_B} f_{j,S} \left( \frac{1}{r_j - r_{j-1}} \ln \frac{r_j}{r_{j-1}} - \frac{1}{r_{j+1} - r_j} \ln \frac{r_{j+1}}{r_j} \right). \quad (64)$$

For the case of volume kernel functions the total number concentration is calculated as:

$$n = \frac{3}{8\pi} \sum_{j=1}^{N_B} f_{j,V} \frac{r_{j+1} - r_{j-1}}{r_{j-1} r_j r_{j+1}}. \quad (65)$$

### 5.2.2. Total Surface Area Concentration

The total surface area concentration for the PSD  $f(r)$  is calculated as:

$$s = 4\pi \int_{r_{\min}}^{r_{\max}} r^2 f(r) dr = 4\pi \sum_{j=1}^{N_B} f_j \int_{r_{j-1}}^{r_{j+1}} r^2 B_j(r) dr. \quad (66)$$

If the PSD  $f(r)$  was retrieved using number kernel functions then the total surface area concentration is calculated as:

$$s = \frac{\pi}{3} \sum_{j=1}^{N_B} f_j [-r_{j-1}(r_{j-1}^2 + r_{j-1}r_j + r_j^2) + r_{j+1}(r_j^2 + r_jr_{j+1} + r_{j+1}^2)]. \quad (67)$$

For the case of surface area kernel functions the total surface area concentration is calculated as:

$$s = \frac{1}{2} \sum_{j=1}^{N_B} f_{j,S}(r_{j+1} - r_{j-1}). \quad (68)$$

For the case of volume kernel functions the total surface area concentration is calculated as:

$$s = 3 \sum_{j=1}^{N_B} f_{j,V} \left( -\frac{r_{j-1}}{r_j - r_{j-1}} \ln \frac{r_j}{r_{j-1}} + \frac{r_{j+1}}{r_{j+1} - r_j} \ln \frac{r_{j+1}}{r_j} \right). \quad (69)$$

### 5.2.3. Total Volume Concentration

The total volume concentration for the PSD  $f(r)$  is calculated as:

$$v = \frac{4\pi}{3} \int_{r_{\min}}^{r_{\max}} r^3 f(r) dr = \frac{4\pi}{3} \sum_{j=1}^{N_B} f_j \int_{r_{j-1}}^{r_{j+1}} r^3 B_j(r) dr. \quad (70)$$

If the PSD  $f(r)$  was retrieved using number kernel functions then the total volume concentration is calculated as:

$$v = \frac{\pi}{15} \sum_{j=1}^{N_B} f_j \left[ 4 \frac{r_j^5 - r_{j-1}^5}{r_j - r_{j-1}} - 5r_{j-1}(r_j + r_{j-1})(r_j^2 + r_{j-1}^2) + 5r_{j+1}(r_{j+1} + r_j)(r_{j+1}^2 + r_j^2) - 4 \frac{r_{j+1}^5 - r_j^5}{r_{j+1} - r_j} \right]. \quad (71)$$

For the case of surface area kernel functions the total volume concentration is calculated as:

$$v = \frac{1}{18} \sum_{j=1}^{N_B} f_{j,S}(r_{j+1} - r_{j-1})(r_{j-1} + r_j + r_{j+1}). \quad (72)$$

For the case of volume kernel functions the total volume concentration is calculated as:

$$v = \frac{1}{2} \sum_{j=1}^{N_B} f_{j,V}(r_{j+1} - r_{j-1}). \quad (73)$$

### 5.2.4. Total Effective Variance

The total effective variance for the PSD  $f(r)$  is calculated as:

$$\begin{aligned} v_{\text{eff}} &= \frac{\int_{r_{\min}}^{r_{\max}} (r - r_{\text{eff}})^2 r^2 f(r) dr}{r_{\text{eff}}^2 \int_{r_{\min}}^{r_{\max}} r^2 f(r) dr} = \frac{4\pi s}{9 v^2} \int_{r_{\min}}^{r_{\max}} r^4 f(r) dr - 1 = \\ &= \frac{4\pi s}{9 v^2} \sum_{j=1}^{N_B} f_j \int_{r_{j-1}}^{r_{j+1}} r^4 B_j(r) dr - 1. \end{aligned} \quad (74)$$

If the PSD  $f(r)$  was retrieved using number kernel functions then the total effective variance is calculated as:

$$v_{\text{eff}} = \frac{2\pi s}{135 v^2} \sum_{j=1}^{N_B} f_j \left[ 5 \frac{r_j^6 - r_{j-1}^6}{r_j - r_{j-1}} - 6r_{j-1} \frac{r_j^5 - r_{j-1}^5}{r_j - r_{j-1}} + 6r_{j+1} \frac{r_{j+1}^5 - r_j^5}{r_{j+1} - r_j} - 5 \frac{r_{j+1}^6 - r_j^6}{r_{j+1} - r_j} \right] - 1. \quad (75)$$

For the case of surface area kernel functions the total effective variance is calculated as:

$$v_{\text{eff}} = \frac{1}{108} \frac{s}{v^2} \sum_{j=1}^{N_B} f_{j,S} \left[ -r_{j-1} (r_{j-1}^2 + r_{j-1} r_j + r_j^2) + r_{j+1} (r_j^2 + r_j r_{j+1} + r_{j+1}^2) \right] - 1. \quad (76)$$

For the case of volume kernel functions the total effective variance is calculated as:

$$v_{\text{eff}} = \frac{1}{18} \frac{s}{v^2} \sum_{j=1}^{N_B} f_{j,V} (r_{j+1} - r_{j-1}) (r_{j-1} + r_j + r_{j+1}) - 1. \quad (77)$$

### 5.3. Parallelization of Computations

The inversion of lidar data allows us for an efficient parallelization of computation steps. We discretize the search space in terms of CRI and inversion windows, which allows us to create a finite set of input parameters. This set of parameters forms the set of mathematical solutions.

The manipulations related to the matrix–vector equations, analysis of the Tikhonov regularization curve, computation of the weight coefficients for the base functions and estimation of the microphysical parameters can be done for each individual solution independently from each other. This feature allows us to split the inversion calculations into independent computational threads, each dealing with its own selected subset of mathematical solutions. Within one application each thread can be launched on a separate core of a multi-core shared memory system. According to our experience, the time of calculations for such a system decreases almost linearly with increasing number of the used cores. Further decreases of the data processing time are possible if the number of cores is increased. We also expect that the floating-point calculations speed of the microprocessors will increase in future.

*Note.* In the inversion parameters ASCII file the number of microprocessors to be used for the data processing can be set by using the parameter “NumOfProcessors”:

```
// Number of CPUs to be used for the parallel data processing.
```

```
NumOfProcessors=64
```

## 5.4. Automated Unsupervised Inversion

Figure 8 shows the flowchart of the TiARA software. The methods were described in the previous sections. Note: the inversion algorithm currently is restricted to the assumption of spherical particle shape.

The TiARA code that is discussed in this ATBD was developed using C++ programming language. The inversion code can be compiled in operating systems Windows and Linux, and executed using command line. Because of the ill-posedness of the mathematical problem expressed as Eqs. (1)–(2) and the differences in the compilers, the microphysical results obtained in Windows and Linux might be slightly different in some cases. It is an agreement to use as a reference the OS “Linux Red Hat 4.4.7-16” version of executables that were compiled with help of “gcc 4.4.7 20120313”. We provide the HDF-file 20160819\_F1\_1min\_oversample\_TiARA.h5 as a reference example of microphysical results retrieved by TiARA using HSRL-2 data and the settings listed in this ATBD.

Table 6 summarizes the data products that we infer from the automated unsupervised inversion algorithm. We use the retrieved PSD to compute particle effective radius, effective variance, number, surface area, and volume concentrations. The combination of retrieved PSD and CRI allows us to compute also all the optical coefficients, lidar ratios, single-scattering albedos at 355, 532, and 1064 nm, and corresponding Ångström coefficients. Statistical uncertainties are provided for all the retrieved parameters.

We define the 500 nm radius as the boundary between fine and coarse mode particles. We are aware that this threshold approach with fixed number predefined in the inversion parameters ASCII file (see Section “Appendix A. Inversion Parameters ASCII File”) may not always be the best choice and other options are possible. For instance, Aerosol Robotic Network (AERONET) uses the minimum of the bimodal size distribution as the fine mode separator. We use the fixed number mostly in order to be able to directly compare the HSRL-2 retrievals with *in situ* measurements. The *in situ* instruments normally have the fixed diameter of the sampling inlet. One has to consider the possible difference in definition of the aerosol fine mode fraction before the comparisons of HSRL-2 results with other instruments or inversion algorithms are carried out.

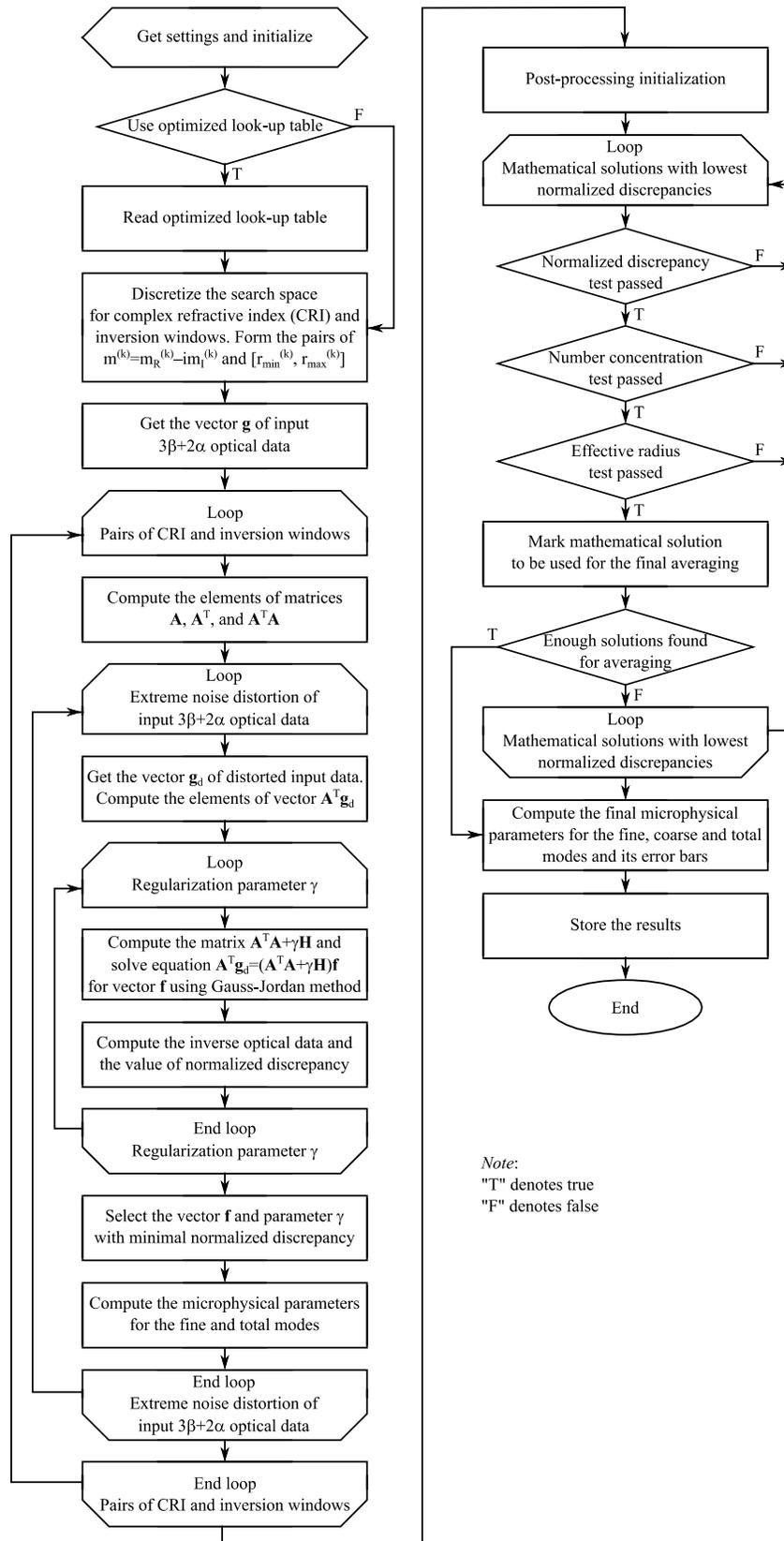


Figure 8. Generalized block scheme of TiARA.

Table 6: Retrieved microphysical parameters as found in a standard HSRL-2 HDF-file

Table name in the HDF-file	Interpretation
abs_coef_total_355	Total absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_total_355	Uncertainty for the total absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
abs_coef_fine_355	Fine mode absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_fine_355	Uncertainty for the fine mode absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
abs_coef_coarse_355	Coarse mode absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_coarse_355	Uncertainty for the coarse mode absorption coefficient at 355 nm ( $\text{km}^{-1}$ )
abs_355_fine_frac_abs_355_total	Fine/total mode absorption coefficient ratio at 355 nm
dstat_(abs_355_fine_frac_abs_355_total)	Uncertainty for the fine/total mode absorption coefficient ratio at 355 nm
abs_coef_total_532	Total absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_total_532	Uncertainty for the total absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
abs_coef_fine_532	Fine mode absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_fine_532	Uncertainty for the fine mode absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
abs_coef_coarse_532	Coarse mode absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_coarse_532	Uncertainty for the coarse mode absorption coefficient at 532 nm ( $\text{km}^{-1}$ )
abs_532_fine_frac_abs_532_total	Fine/total mode absorption coefficient ratio at 532 nm
dstat_(abs_532_fine_frac_abs_532_total)	Uncertainty for the fine/total mode absorption coefficient ratio at 532 nm
abs_coef_total_1064	Total absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_total_1064	Uncertainty for the total absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
abs_coef_fine_1064	Fine mode absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_fine_1064	Uncertainty for the fine mode absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
abs_coef_coarse_1064	Coarse mode absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
dstat_abs_coef_coarse_1064	Uncertainty for the coarse mode absorption coefficient at 1064 nm ( $\text{km}^{-1}$ )
abs_1064_fine_frac_abs_1064_total	Fine/total mode absorption coefficient ratio at 1064 nm
dstat_(abs_1064_fine_frac_abs_1064_total)	Uncertainty for the fine/total mode absorption coefficient ratio at 1064 nm
abs_Ang_total_355	Total Ångström coefficient for the absorption coefficient at 355/532 nm
dstat_abs_Ang_total_355	Uncertainty for the total Ångström coefficient for the absorption coefficient at 355/532 nm
abs_Ang_fine_355	Fine mode Ångström coefficient for the absorption coefficient at 355/532 nm
dstat_abs_Ang_fine_355	Uncertainty for the fine mode Ångström coefficient for the absorption coefficient at 355/532 nm
abs_Ang_coarse_355	Coarse mode Ångström coefficient for the absorption coefficient at 355/532 nm
dstat_abs_Ang_coarse_355	Uncertainty for the coarse mode Ångström coefficient for the absorption coefficient at 355/532 nm
abs_Ang_total_532	Total Ångström coefficient for the absorption coefficient at 532/1064 nm
dstat_abs_Ang_total_532	Uncertainty for the total Ångström coefficient for the absorption coefficient at 532/1064 nm
abs_Ang_fine_532	Fine mode Ångström coefficient for the absorption coefficient at 532/1064 nm
dstat_abs_Ang_fine_532	Uncertainty for the fine mode Ångström coefficient for the absorption coefficient at 532/1064 nm
abs_Ang_coarse_532	Coarse mode Ångström coefficient for the absorption coefficient at 532/1064 nm
dstat_abs_Ang_coarse_532	Uncertainty for the coarse mode Ångström coefficient for the absorption coefficient at 532/1064 nm
AverDiscr	Averaged discrepancy (%)
dstat_AverDiscr	Uncertainty for the averaged discrepancy (%)
bsc_coef_total_355	Total backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
dstat_bsc_coef_total_355	Uncertainty for the total backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
bsc_coef_fine_355	Fine mode backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
dstat_bsc_coef_fine_355	Uncertainty for the fine mode backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
bsc_coef_coarse_355	Coarse mode backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
dstat_bsc_coef_coarse_355	Uncertainty for the coarse mode backscatter coefficient at 355 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
bsc_355_fine_frac_bsc_355_total	Fine/total mode backscatter coefficient ratio at 355 nm
dstat_(bsc_355_fine_frac_bsc_355_total)	Uncertainty for the fine/total mode backscatter coefficient ratio at 355 nm
bsc_coef_total_532	Total backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
dstat_bsc_coef_total_532	Uncertainty for the total backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
bsc_coef_fine_532	Fine mode backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
dstat_bsc_coef_fine_532	Uncertainty for the fine mode backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )

<b>bsc_coef_coarse_532</b>	Coarse mode backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>dstat_bsc_coef_coarse_532</b>	Uncertainty for the coarse mode backscatter coefficient at 532 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>bsc_532_fine_frac_bsc_532_total</b>	Fine/total mode backscatter coefficient ratio at 532 nm
<b>dstat_(bsc_532_fine_frac_bsc_532_total)</b>	Uncertainty for the fine/total mode backscatter coefficient ratio at 532 nm
<b>bsc_coef_total_1064</b>	Total backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>dstat_bsc_coef_total_1064</b>	Uncertainty for the total backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>bsc_coef_fine_1064</b>	Fine mode backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>dstat_bsc_coef_fine_1064</b>	Uncertainty for the fine mode backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>bsc_coef_coarse_1064</b>	Coarse mode backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>dstat_bsc_coef_coarse_1064</b>	Uncertainty for the coarse mode backscatter coefficient at 1064 nm ( $\text{km}^{-1}\text{sr}^{-1}$ )
<b>bsc_1064_fine_frac_bsc_1064_total</b>	Fine/total mode backscatter coefficient ratio at 1064 nm
<b>dstat_(bsc_1064_fine_frac_bsc_1064_total)</b>	Uncertainty for the fine/total mode backscatter coefficient ratio at 1064 nm
<b>bsc_Ang_total_355</b>	Total Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>dstat_bsc_Ang_total_355</b>	Uncertainty for the total Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>bsc_Ang_fine_355</b>	Fine mode Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>dstat_bsc_Ang_fine_355</b>	Uncertainty for the fine mode Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>bsc_Ang_coarse_355</b>	Coarse mode Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>dstat_bsc_Ang_coarse_355</b>	Uncertainty for the coarse mode Ångström coefficient for the backscatter coefficient at 355/532 nm
<b>bsc_Ang_total_532</b>	Total Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>dstat_bsc_Ang_total_532</b>	Uncertainty for the total Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>bsc_Ang_fine_532</b>	Fine mode Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>dstat_bsc_Ang_fine_532</b>	Uncertainty for the fine mode Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>bsc_Ang_coarse_532</b>	Coarse mode Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>dstat_bsc_Ang_coarse_532</b>	Uncertainty for the coarse mode Ångström coefficient for the backscatter coefficient at 532/1064 nm
<b>effvar_total</b>	Total effective variance
<b>dstat_effvar_total</b>	Uncertainty for the total effective variance
<b>effvar_fine</b>	Fine mode effective variance
<b>dstat_effvar_fine</b>	Uncertainty for the fine mode effective variance
<b>effvar_coarse</b>	Coarse mode effective variance
<b>dstat_effvar_coarse</b>	Uncertainty for the coarse mode effective variance
<b>effvar_fine_frac_effvar_total</b>	Fine/total mode effective variance ratio
<b>dstat_(effvar_fine_frac_effvar_total)</b>	Uncertainty for the fine/total mode effective variance ratio
<b>ext_coef_total_355</b>	Total extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_total_355</b>	Uncertainty for the total extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>ext_coef_fine_355</b>	Fine mode extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_fine_355</b>	Uncertainty for the fine mode extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>ext_coef_coarse_355</b>	Coarse mode extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_coarse_355</b>	Uncertainty for the coarse mode extinction coefficient at 355 nm ( $\text{km}^{-1}$ )
<b>ext_355_fine_frac_ext_355_total</b>	Fine/total mode extinction coefficient ratio at 355 nm
<b>dstat_(ext_355_fine_frac_ext_355_total)</b>	Uncertainty for the fine/total mode extinction coefficient ratio at 355 nm
<b>ext_coef_total_532</b>	Total extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_total_532</b>	Uncertainty for the total extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>ext_coef_fine_532</b>	Fine mode extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_fine_532</b>	Uncertainty for the fine mode extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>ext_coef_coarse_532</b>	Coarse mode extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_coarse_532</b>	Uncertainty for the coarse mode extinction coefficient at 532 nm ( $\text{km}^{-1}$ )
<b>ext_532_fine_frac_ext_532_total</b>	Fine/total mode extinction coefficient ratio at 532 nm
<b>dstat_(ext_532_fine_frac_ext_532_total)</b>	Uncertainty for the fine/total mode extinction coefficient ratio at 532 nm
<b>ext_coef_total_1064</b>	Total extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_total_1064</b>	Uncertainty for the total extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )
<b>ext_coef_fine_1064</b>	Fine mode extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )
<b>dstat_ext_coef_fine_1064</b>	Uncertainty for the fine mode extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )
<b>ext_coef_coarse_1064</b>	Coarse mode extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )

dstat_ext_coef_coarse_1064	Uncertainty for the coarse mode extinction coefficient at 1064 nm ( $\text{km}^{-1}$ )
ext_1064_fine_frac_ext_1064_total	Fine/total mode extinction coefficient ratio at 1064 nm
dstat_(ext_1064_fine_frac_ext_1064_total)	Uncertainty for the fine/total mode extinction coefficient ratio at 1064 nm
ext_Ang_total_355	Total Ångström coefficient for the extinction coefficient at 355/532 nm
dstat_ext_Ang_total_355	Uncertainty for the total Ångström coefficient for the extinction coefficient at 355/532 nm
ext_Ang_fine_355	Fine mode Ångström coefficient for the extinction coefficient at 355/532 nm
dstat_ext_Ang_fine_355	Uncertainty for the fine mode Ångström coefficient for the extinction coefficient at 355/532 nm
ext_Ang_coarse_355	Coarse mode Ångström coefficient for the extinction coefficient at 355/532 nm
dstat_ext_Ang_coarse_355	Uncertainty for the coarse mode Ångström coefficient for the extinction coefficient at 355/532 nm
ext_Ang_total_532	Total Ångström coefficient for the extinction coefficient at 532/1064 nm
dstat_ext_Ang_total_532	Uncertainty for the total Ångström coefficient for the extinction coefficient at 532/1064 nm
ext_Ang_fine_532	Fine mode Ångström coefficient for the extinction coefficient at 532/1064 nm
dstat_ext_Ang_fine_532	Uncertainty for the fine mode Ångström coefficient for the extinction coefficient at 532/1064 nm
ext_Ang_coarse_532	Coarse mode Ångström coefficient for the extinction coefficient at 532/1064 nm
dstat_ext_Ang_coarse_532	Uncertainty for the coarse mode Ångström coefficient for the extinction coefficient at 532/1064 nm
mReal_total	Real part of the CRI
dstat_mReal_total	Uncertainty for the real part of the CRI
mImag_total	Imaginary part of the CRI
dstat_mImag_total	Uncertainty for the imaginary part of the CRI
N_total	Total number concentration ( $1/\text{cm}^3$ )
dstat_N_total	Uncertainty for the total number concentration ( $1/\text{cm}^3$ )
N_fine	Fine mode number concentration ( $1/\text{cm}^3$ )
dstat_N_fine	Uncertainty for the fine mode number concentration ( $1/\text{cm}^3$ )
N_coarse	Coarse mode number concentration ( $1/\text{cm}^3$ )
dstat_N_coarse	Uncertainty for the coarse mode number concentration ( $1/\text{cm}^3$ )
N_fine_frac_N_total	Fine/total mode number concentration ratio
dstat_(N_fine_frac_N_total)	Uncertainty for the fine/total mode number concentration ratio
reff_total	Total effective radius ( $\mu\text{m}$ )
dstat_reff_total	Uncertainty for the total effective radius ( $\mu\text{m}$ )
reff_fine	Fine mode effective radius ( $\mu\text{m}$ )
dstat_reff_fine	Uncertainty for the fine mode effective radius ( $\mu\text{m}$ )
reff_coarse	Coarse mode effective radius ( $\mu\text{m}$ )
dstat_reff_coarse	Uncertainty for the coarse mode effective radius ( $\mu\text{m}$ )
reff_fine_frac_reff_total	Fine/total mode effective radius ratio
dstat_(reff_fine_frac_reff_total)	Uncertainty for the fine/total mode effective radius ratio
rmin_total	Minimum particle radius ( $\mu\text{m}$ )
dstat_rmin_total	Uncertainty for the minimum particle radius ( $\mu\text{m}$ )
rmax_total	Maximum particle radius ( $\mu\text{m}$ )
dstat_rmax_total	Uncertainty for the maximum particle radius ( $\mu\text{m}$ )
S_total	Total surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
dstat_S_total	Uncertainty for the total surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
S_fine	Fine mode surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
dstat_S_fine	Uncertainty for the fine mode surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
S_coarse	Coarse mode surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
dstat_S_coarse	Uncertainty for the coarse mode surface area concentration ( $\mu\text{m}^2/\text{cm}^3$ )
S_fine_frac_S_total	Fine/total mode surface area concentration ratio
dstat_(S_fine_frac_S_total)	Uncertainty for the fine/total mode surface area concentration ratio
Sa_total_355	Total lidar ratio at 355 nm (sr)
dstat_Sa_total_355	Uncertainty for the total lidar ratio at 355 nm (sr)
Sa_fine_355	Fine mode lidar ratio at 355 nm (sr)
dstat_Sa_fine_355	Uncertainty for the fine mode lidar ratio at 355 nm (sr)
Sa_coarse_355	Coarse mode lidar ratio at 355 nm (sr)
dstat_Sa_coarse_355	Uncertainty for the coarse mode lidar ratio at 355 nm (sr)

Sa_355_fine_frac_Sa_355_total	Fine/total mode lidar ratio at 355 nm
dstat_(Sa_355_fine_frac_Sa_355_total)	Uncertainty for the fine/total mode lidar ratio at 355 nm
Sa_total_532	Total lidar ratio at 532 nm (sr)
dstat_Sa_total_532	Uncertainty for the total lidar ratio at 532 nm (sr)
Sa_fine_532	Fine mode lidar ratio at 532 nm (sr)
dstat_Sa_fine_532	Uncertainty for the fine mode lidar ratio at 532 nm (sr)
Sa_coarse_532	Coarse mode lidar ratio at 532 nm (sr)
dstat_Sa_coarse_532	Uncertainty for the coarse mode lidar ratio at 532 nm (sr)
Sa_532_fine_frac_Sa_532_total	Fine/total mode lidar ratio at 532 nm
dstat_(Sa_532_fine_frac_Sa_532_total)	Uncertainty for the fine/total mode lidar ratio at 532 nm
Sa_total_1064	Total lidar ratio at 1064 nm (sr)
dstat_Sa_total_1064	Uncertainty for the total lidar ratio at 1064 nm (sr)
Sa_fine_1064	Fine mode lidar ratio at 1064 nm (sr)
dstat_Sa_fine_1064	Uncertainty for the fine mode lidar ratio at 1064 nm (sr)
Sa_coarse_1064	Coarse mode lidar ratio at 1064 nm (sr)
dstat_Sa_coarse_1064	Uncertainty for the coarse mode lidar ratio at 1064 nm (sr)
Sa_1064_fine_frac_Sa_1064_total	Fine/total mode lidar ratio at 1064 nm
dstat_(Sa_1064_fine_frac_Sa_1064_total)	Uncertainty for the fine/total mode lidar ratio at 1064 nm
Sa_Ang_total_355	Total Ångström coefficient for the lidar ratio at 355/532 nm
dstat_Sa_Ang_total_355	Uncertainty for the total Ångström coefficient for the lidar ratio at 355/532 nm
Sa_Ang_fine_355	Fine mode Ångström coefficient for the lidar ratio at 355/532 nm
dstat_Sa_Ang_fine_355	Uncertainty for the fine mode Ångström coefficient for the lidar ratio at 355/532 nm
Sa_Ang_coarse_355	Coarse mode Ångström coefficient for the lidar ratio at 355/532 nm
dstat_Sa_Ang_coarse_355	Uncertainty for the coarse mode Ångström coefficient for the lidar ratio at 355/532 nm
Sa_Ang_total_532	Total Ångström coefficient for the lidar ratio at 532/1064 nm
dstat_Sa_Ang_total_532	Uncertainty for the total Ångström coefficient for the lidar ratio at 532/1064 nm
Sa_Ang_fine_532	Fine mode Ångström coefficient for the lidar ratio at 532/1064 nm
dstat_Sa_Ang_fine_532	Uncertainty for the fine mode Ångström coefficient for the lidar ratio at 532/1064nm
Sa_Ang_coarse_532	Coarse mode Ångström coefficient for the lidar ratio at 532/1064 nm
dstat_Sa_Ang_coarse_532	Uncertainty for the coarse mode Ångström coefficient for the lidar ratio at 532/1064 nm
scat_coef_total_355	Total scattering coefficient at 355 nm (km <sup>-1</sup> )
dstat_scat_coef_total_355	Uncertainty for the total scattering coefficient at 355 nm (km <sup>-1</sup> )
scat_coef_fine_355	Fine mode scattering coefficient at 355 nm (km <sup>-1</sup> )
dstat_scat_coef_fine_355	Uncertainty for the fine mode scattering coefficient at 355 nm (km <sup>-1</sup> )
scat_coef_coarse_355	Coarse mode scattering coefficient at 355 nm (km <sup>-1</sup> )
dstat_scat_coef_coarse_355	Uncertainty for the coarse mode scattering coefficient at 355 nm (km <sup>-1</sup> )
scat_355_fine_frac_scat_355_total	Fine/total mode scattering coefficient ratio at 355 nm
dstat_(scat_355_fine_frac_scat_355_total)	Uncertainty for the fine/total mode scattering coefficient ratio at 355 nm
scat_coef_total_532	Total scattering coefficient at 532 nm (km <sup>-1</sup> )
dstat_scat_coef_total_532	Uncertainty for the total scattering coefficient at 532 nm (km <sup>-1</sup> )
scat_coef_fine_532	Fine mode scattering coefficient at 532 nm (km <sup>-1</sup> )
dstat_scat_coef_fine_532	Uncertainty for the fine mode scattering coefficient at 532 nm (km <sup>-1</sup> )
scat_coef_coarse_532	Coarse mode scattering coefficient at 532 nm (km <sup>-1</sup> )
dstat_scat_coef_coarse_532	Uncertainty for the coarse mode scattering coefficient at 532 nm (km <sup>-1</sup> )
scat_532_fine_frac_scat_532_total	Fine/total mode scattering coefficient ratio at 532 nm
dstat_(scat_532_fine_frac_scat_532_total)	Uncertainty for the fine/total mode scattering coefficient ratio at 532 nm
scat_coef_total_1064	Total scattering coefficient at 1064 nm (km <sup>-1</sup> )
dstat_scat_coef_total_1064	Uncertainty for the total scattering coefficient at 1064 nm (km <sup>-1</sup> )
scat_coef_fine_1064	Fine mode scattering coefficient at 1064 nm (km <sup>-1</sup> )
dstat_scat_coef_fine_1064	Uncertainty for the fine mode scattering coefficient at 1064 nm (km <sup>-1</sup> )
scat_coef_coarse_1064	Coarse mode scattering coefficient at 1064 nm (km <sup>-1</sup> )
dstat_scat_coef_coarse_1064	Uncertainty for the coarse mode scattering coefficient at 1064 nm (km <sup>-1</sup> )
scat_1064_fine_frac_scat_1064_total	Fine/total mode scattering coefficient ratio at 1064 nm
dstat_(scat_1064_fine_frac_scat_1064_total)	Uncertainty for the fine/total mode scattering coefficient ratio at 1064 nm
scat_Ang_total_355	Total Ångström coefficient for the scattering coefficient at 355/532 nm
dstat_scat_Ang_total_355	Uncertainty for the total Ångström coefficient for the scattering coefficient at 355/532 nm

scat_Ang_fine_355	Fine mode Ångström coefficient for the scattering coefficient at 355/532 nm
dstat_scat_Ang_fine_355	Uncertainty for the fine mode Ångström coefficient for the scattering coefficient at 355/532 nm
scat_Ang_coarse_355	Coarse mode Ångström coefficient for the scattering coefficient at 355/532 nm
dstat_scat_Ang_coarse_355	Uncertainty for the coarse mode Ångström coefficient for the scattering coefficient at 355/532 nm
scat_Ang_total_532	Total Ångström coefficient for the scattering coefficient at 532/1064 nm
dstat_scat_Ang_total_532	Uncertainty for the total Ångström coefficient for the scattering coefficient at 532/1064 nm
scat_Ang_fine_532	Fine mode Ångström coefficient for the scattering coefficient at 532/1064 nm
dstat_scat_Ang_fine_532	Uncertainty for the fine mode Ångström coefficient for the scattering coefficient at 532/1064 nm
scat_Ang_coarse_532	Coarse mode Ångström coefficient for the scattering coefficient at 532/1064 nm
dstat_scat_Ang_coarse_532	Uncertainty for the coarse mode Ångström coefficient for the scattering coefficient at 532/1064 nm
SSA_total_355	Total SSA at 355 nm
dstat_SSA_total_355	Uncertainty for the total SSA at 355 nm
SSA_fine_355	Fine mode SSA at 355 nm
dstat_SSA_fine_355	Uncertainty for the fine mode SSA at 355 nm
SSA_coarse_355	Coarse mode SSA at 355 nm
dstat_SSA_coarse_355	Uncertainty for the coarse mode SSA at 355 nm
SSA_355_fine_frac_SSA_355_total	Fine/total mode SSA ratio at 355 nm
dstat_(SSA_355_fine_frac_SSA_355_total)	Uncertainty for the fine/total mode SSA ratio at 355 nm
SSA_total_532	Total SSA at 532 nm
dstat_SSA_total_532	Uncertainty for the total SSA at 532 nm
SSA_fine_532	Fine mode SSA at 532 nm
dstat_SSA_fine_532	Uncertainty for the fine mode SSA at 532 nm
SSA_coarse_532	Coarse mode SSA at 532 nm
dstat_SSA_coarse_532	Uncertainty for the coarse mode SSA at 532 nm
SSA_532_fine_frac_SSA_532_total	Fine/total mode SSA ratio at 532 nm
dstat_(SSA_532_fine_frac_SSA_532_total)	Uncertainty for the fine/total mode SSA ratio at 532 nm
SSA_total_1064	Total SSA at 1064 nm
dstat_SSA_total_1064	Uncertainty for the total SSA at 1064 nm
SSA_fine_1064	Fine mode SSA at 1064 nm
dstat_SSA_fine_1064	Uncertainty for the fine mode SSA at 1064 nm
SSA_coarse_1064	Coarse mode SSA at 1064 nm
dstat_SSA_coarse_1064	Uncertainty for the coarse mode SSA at 1064 nm
SSA_1064_fine_frac_SSA_1064_total	Fine/total mode SSA ratio at 1064 nm
dstat_(SSA_1064_fine_frac_SSA_1064_total)	Uncertainty for the fine/total mode SSA ratio at 1064 nm
SSA_Ang_total_355	Total Ångström coefficient for the SSA at 355/532 nm
dstat_SSA_Ang_total_355	Uncertainty for the total Ångström coefficient for the SSA at 355/532 nm
SSA_Ang_fine_355	Fine mode Ångström coefficient for the SSA at 355/532 nm
dstat_SSA_Ang_fine_355	Uncertainty for the fine mode Ångström coefficient for the SSA at 355/532 nm
SSA_Ang_coarse_355	Coarse mode Ångström coefficient for the SSA at 355/532 nm
dstat_SSA_Ang_coarse_355	Uncertainty for the coarse mode Ångstr. coefficient for the SSA at 355/532 nm
SSA_Ang_total_532	Total Ångström coefficient for the SSA at 532/1064 nm
dstat_SSA_Ang_total_532	Uncertainty for the total Ångström coefficient for the SSA at 532/1064 nm
SSA_Ang_fine_532	Fine mode Ångström coefficient for the SSA at 532/1064 nm
dstat_SSA_Ang_fine_532	Uncertainty for the fine mode Ångström coefficient for the SSA at 532/1064 nm
SSA_Ang_coarse_532	Coarse mode Ångström coefficient for the SSA at 532/1064 nm
dstat_SSA_Ang_coarse_532	Uncertainty for the coarse mode Ångstr. coefficient for the SSA at 532/1064 nm
V_total	Total volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
dstat_V_total	Uncertainty for the total volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
V_fine	Fine mode volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
dstat_V_fine	Uncertainty for the fine mode volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
V_coarse	Coarse mode volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
dstat_V_coarse	Uncertainty for the coarse mode volume concentration ( $\mu\text{m}^3/\text{cm}^3$ )
V_fine_frac_V_total	Fine/total mode volume concentration ratio
dstat_(V_fine_frac_V_total)	Uncertainty for the fine/total mode volume concentration ratio

## 6. Simulations

### 6.1. Simulation Mode

Historically, the automated unsupervised TiARA software grew up from the manual expert-mode version of the code that was and still is a testbed for the  $3\beta+2\alpha$  inversion development [11, 14, 15, 16, 27, 28]. Due to a lack of the reliable experimental data suitable for the inter-comparisons of microphysical retrievals, we were actively using numerical simulations on the early stages of the code development.

The simulation mode still is a natural part of TiARA software and there are two options how to run it. The first option is to simulate  $3\beta+2\alpha$  optical data using a logarithmic-normal PSD with up to three modes (see Section “6.2. Logarithmic-Normal Particle Size Distribution”). In this case, the  $3\beta+2\alpha$  input data will be computed using Eqs. (1)–(2), Mie-scattering theory [2, 3], and Simpson’s rule of numerical integration [7]. The second option is to manually fill in the  $3\beta+2\alpha$  pre-computed or experimental data into the user input file (see Section “Appendix A. Inversion Parameters ASCII File”). In both cases, the retrieved information is stored on the hard drive of the computer as multiple ASCII solutions-set files (\*.ssf files). These \*.ssf files can be used for further detailed analysis using the manual expert version of TiARA.

*Note.* In the inversion parameters ASCII file the parameters of simulation mode can be selected using the corresponding parameters:

```
***** General parameters of input data *****
// Type of input data. Simulation or experimental.
// If "0" then simulation of log-normal particle size distribution is used.
// If "1" then experimental data are used.
InputDataType=0

***** Experimental optical data *****
::::: Extinction coefficients :::::
// Use ("1") or not ("0") extinction coefficient No1 for inversion.
UseExtinction01=1
// Wavelength of extinction coefficient No1, nm.
ExtinctionWavelength01=355
// Value of extinction coefficient No1, 1/m.
ExtinctionCoef01=1.09026e-6
// Use ("1") or not ("0") extinction coefficient No2 for inversion.
UseExtinction02=1
// Wavelength of extinction coefficient No2, nm.
ExtinctionWavelength02=532
// Value of extinction coefficient No2, 1/m.
ExtinctionCoef02=7.43007e-7
// Use ("1") or not ("0") extinction coefficient No3 for inversion.
UseExtinction03=0
// Wavelength of extinction coefficient No3, nm.
ExtinctionWavelength03=0
// Value of extinction coefficient No3, 1/m.
ExtinctionCoef03=0
// Use ("1") or not ("0") extinction coefficient No4 for inversion.
```

```
UseExtinction04=0
// Wavelength of extinction coefficient No4, nm.
ExtinctionWavelength04=0
// Value of extinction coefficient No4, 1/m.
ExtinctionCoef04=0
// Use ("1") or not ("0") extinction coefficient No5 for inversion.
UseExtinction05=0
// Wavelength of extinction coefficient No5, nm.
ExtinctionWavelength05=0
// Value of extinction coefficient No5, 1/m.
ExtinctionCoef05=0
// Use ("1") or not ("0") extinction coefficient No6 for inversion.
UseExtinction06=0
// Wavelength of extinction coefficient No6, nm.
ExtinctionWavelength06=0
// Value of extinction coefficient No6, 1/m.
ExtinctionCoef06=0
// Use ("1") or not ("0") extinction coefficient No7 for inversion.
UseExtinction07=0
// Wavelength of extinction coefficient No7, nm.
ExtinctionWavelength07=0
// Value of extinction coefficient No7, 1/m.
ExtinctionCoef07=0
// Use ("1") or not ("0") extinction coefficient No8 for inversion.
UseExtinction08=0
// Wavelength of extinction coefficient No8, nm.
ExtinctionWavelength08=0
// Value of extinction coefficient No8, 1/m.
ExtinctionCoef08=0
// Use ("1") or not ("0") extinction coefficient No9 for inversion.
UseExtinction09=0
// Wavelength of extinction coefficient No9, nm.
ExtinctionWavelength09=0
// Value of extinction coefficient No9, 1/m.
ExtinctionCoef09=0
// Use ("1") or not ("0") extinction coefficient No10 for inversion.
UseExtinction10=0
// Wavelength of extinction coefficient No10, nm.
ExtinctionWavelength10=0
// Value of extinction coefficient No10, 1/m.
ExtinctionCoef10=0

:::: Backscatter coefficients ::::
// Use ("1") or not ("0") backscatter coefficient No1 for inversion.
UseBackscatter01=1
// Wavelength of backscatter coefficient No1, nm.
BackscatterWavelength01=355
// Value of backscatter coefficient No1, 1/(m•sr).
BackscatterCoef01=7.94049e-9
```

```
// Use ("1") or not ("0") backscatter coefficient No2 for inversion.
UseBackscatter02=1
// Wavelength of backscatter coefficient No2, nm.
BackscatterWavelength02=532
// Value of backscatter coefficient No2, 1/(m•sr).
BackscatterCoef02=6.9284e-9
// Use ("1") or not ("0") backscatter coefficient No3 for inversion.
UseBackscatter03=1
// Wavelength of backscatter coefficient No3, nm.
BackscatterWavelength03=1064
// Value of backscatter coefficient No3, 1/(m•sr).
BackscatterCoef03=1.67199e-9
// Use ("1") or not ("0") backscatter coefficient No4 for inversion.
UseBackscatter04=0
// Wavelength of backscatter coefficient No4, nm.
BackscatterWavelength04=0
// Value of backscatter coefficient No4, 1/(m•sr).
BackscatterCoef04=0
// Use ("1") or not ("0") backscatter coefficient No5 for inversion.
UseBackscatter05=0
// Wavelength of backscatter coefficient No5, nm.
BackscatterWavelength05=0
// Value of backscatter coefficient No5, 1/(m•sr).
BackscatterCoef05=0
// Use ("1") or not ("0") backscatter coefficient No6 for inversion.
UseBackscatter06=0
// Wavelength of backscatter coefficient No6, nm.
BackscatterWavelength06=0
// Value of backscatter coefficient No6, 1/(m•sr).
BackscatterCoef06=0
// Use ("1") or not ("0") backscatter coefficient No7 for inversion.
UseBackscatter07=0
// Wavelength of backscatter coefficient No7, nm.
BackscatterWavelength07=0
// Value of backscatter coefficient No7, 1/(m•sr).
BackscatterCoef07=0
// Use ("1") or not ("0") backscatter coefficient No8 for inversion.
UseBackscatter08=0
// Wavelength of backscatter coefficient No8, nm.
BackscatterWavelength08=0
// Value of backscatter coefficient No8, 1/(m•sr).
BackscatterCoef08=0
// Use ("1") or not ("0") backscatter coefficient No9 for inversion.
UseBackscatter09=0
// Wavelength of backscatter coefficient No9, nm.
BackscatterWavelength09=0
// Value of backscatter coefficient No9, 1/(m•sr).
BackscatterCoef09=0
// Use ("1") or not ("0") backscatter coefficient No10 for inversion.
```

```

UseBackscatter10=0
// Wavelength of backscatter coefficient No10, nm.
BackscatterWavelength10=0
// Value of backscatter coefficient No10, 1/(m*sr).
BackscatterCoef10=0

```

## 6.2. Logarithmic-Normal Particle Size Distribution

For the simulation purposes we use the logarithmic-normal PSDs. The logarithmic-normal shape is the only option supported in the software package because it is believed to be a shape that in an approximate manner describes the naturally occurring PSDs well enough [10, 24].

The monomodal logarithmic-normal PSD is described as:

$$f_{\text{mono}}(r) = \frac{n_0}{r\sqrt{2\pi}\ln\sigma} \exp\left[-\frac{(\ln r - \ln r_{\text{med}})^2}{2\ln^2\sigma}\right], \quad (78)$$

where the count median radius  $r_{\text{med}}$  is defined as the radius above which there are as many particles as there are particles with radii below  $r_{\text{med}}$ . The term  $\sigma$  denotes the geometric standard deviation. The scaling parameter  $n_0$  is equal to  $1 \text{ cm}^{-3}$ .

A multimodal logarithmic-normal PSD with up to three modes is described as:

$$f_{\text{multi}}(r) = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^3 \frac{n_j}{r\ln\sigma_j} \exp\left[-\frac{(\ln r - \ln r_{\text{med},j})^2}{2\ln^2\sigma_j}\right], \quad (79)$$

where the count median radii  $r_{\text{med},j}$  can be defined for each mode separately, as well as the geometric standard deviations  $\sigma_j$ . The scaling parameter  $n_1$  is predefined to be equal to  $1 \text{ cm}^{-3}$ , but the scaling parameters  $n_2$  and  $n_3$  can be selected individually.

*Note.* In the inversion parameters ASCII file the parameters of simulated multimodal (up to three modes) logarithmic-normal PSDs can be selected using the corresponding parameters:

```

::::: First mode for simulations :::::
// Value of mean radius for first simulation mode, mum.
MeanRadius1=0.3
// Geometrical standard deviation for first simulation mode.
ModeWidth1=1.4
// Concentration value for first simulation mode is predefined to be equal to 1 1/cm^3.

::::: Second mode for simulations :::::
// Use or not the second mode during simulations.
// If "0" then second mode is absent, if "1" then second mode is present.
UseMode2=0
// Value of mean radius for second simulation mode, mum.
MeanRadius2=0.5
// Geometrical standard deviation for second simulation mode.
ModeWidth2=1.3
// Concentration value for second simulation mode, 1/cm^3.
Concentration2=0

```

```

::::: Third mode for simulations :::::
// Use or not the third mode during simulations.
// If "0" then third mode is absent, if "1" then third mode is present.
UseMode3=0
// Value of mean radius for third simulation mode, num.
MeanRadius3=0.9
// Geometrical standard deviation for third simulation mode.
ModeWidth3=1.2
// Concentration value for third simulation mode, 1/cm^3.
Concentration3=0
    
```

### 6.3. Preliminary Uncertainty Analysis

It is known that the extinction number kernel functions for the wavelength 355 and 532 nm qualitatively are very similar and can be well approximated by parabolas [6]. Figure 9 shows as an example the extinction kernel functions for the CRI  $m = 1.44 - i \cdot 0.0025$ . As one can see in Fig. 9, the kernel functions are oscillating around the parabolic curve for this particular CRI, i.e.,  $K_\alpha(m, r, \lambda) \sim a_\lambda r^2$ , where  $a_\lambda$  is an approximation coefficient. For other ambient-like CRIs, the qualitative behavior of the extinction kernel functions is about the same. This feature allows us to conclude that the measurement of extinction coefficients [see Eq. (2)] is a direct proxy of the particle surface area concentration [see Eq. (10)] and that these two parameters are linearly correlated [6]. Thus, we can expect that theoretically the relative uncertainty of the surface area concentration retrieval will be a) the smallest compared to all other microphysical parameters and b) approximately twice the relative uncertainty of the extinction coefficients measurement [6]. Numerical simulations and processing of experimental lidar data confirm this result [6, 11, 16, 27, 28].

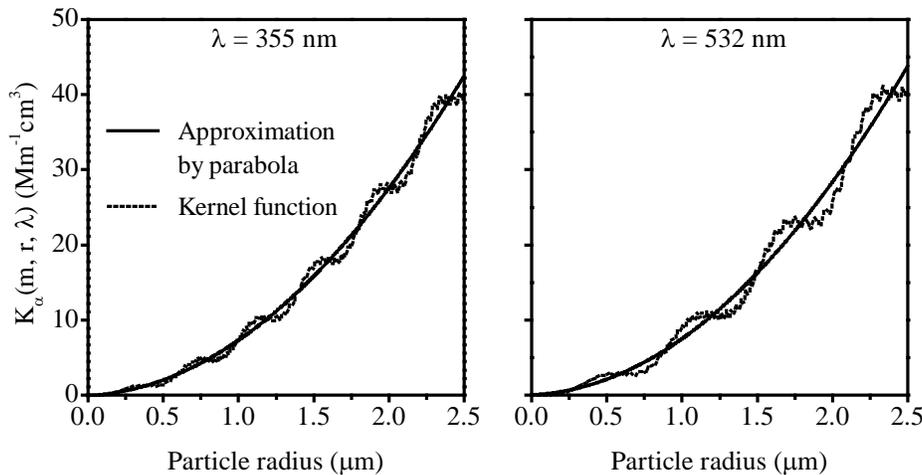


Figure 9. Extinction number kernel functions and their approximations by parabolas.

For the monomodal logarithmic-normal PSD [see Eq. (78)] we can express number, surface area and volume concentrations, effective radius and effective variance [see Eqs. (9)–(13)] directly

with the parameters of PSD  $\{n_0, r_{\text{med}}, \sigma\}$  [6]. We do not retrieve  $r_{\text{med}}$  and  $\sigma$  in TiARA explicitly but we can use them to qualitatively explain why some of the size microphysical parameters will be retrieved better or worse compared to the others. For the purpose of preliminary uncertainty analysis it would be more convenient to use the set  $\{s, r_{\text{med}}, \sigma\}$  because we know that the relative uncertainty  $\frac{\Delta s}{s}$  of surface area concentration retrieval is small [6].

Total surface area concentration can be expressed as:

$$s = 4\pi n_0 r_{\text{med}}^2 \exp(2\ln^2 \sigma). \quad (80)$$

Total number concentration can be expressed as:

$$n = n_0 = \frac{s}{4\pi r_{\text{med}}^2 \exp(2\ln^2 \sigma)}. \quad (81)$$

Total volume concentration can be expressed as:

$$v = \frac{4\pi}{3} n_0 r_{\text{med}}^3 \exp(4.5\ln^2 \sigma) = \frac{1}{3} s r_{\text{med}} \exp(2.5\ln^2 \sigma). \quad (82)$$

Total effective radius can be expressed as:

$$r_{\text{eff}} = r_{\text{med}} \exp(2.5\ln^2 \sigma). \quad (83)$$

Total effective variance can be expressed as:

$$v_{\text{eff}} = \exp(\ln^2 \sigma) - 1. \quad (84)$$

The relative uncertainty of the total number concentration retrieval can be estimated as:

$$\frac{\Delta n}{n} \approx \frac{1}{n} \left[ \left| \frac{\partial n}{\partial s} \right| \Delta s + \left| \frac{\partial n}{\partial r_{\text{med}}} \right| \Delta r_{\text{med}} + \left| \frac{\partial n}{\partial \sigma} \right| \Delta \sigma \right] = \frac{\Delta s}{s} + 2 \frac{\Delta r_{\text{med}}}{r_{\text{med}}} + 4 \ln \sigma \frac{\Delta \sigma}{\sigma}. \quad (85)$$

The relative uncertainty of the total volume concentration retrieval can be estimated as:

$$\frac{\Delta v}{v} \approx \frac{1}{v} \left[ \left| \frac{\partial v}{\partial s} \right| \Delta s + \left| \frac{\partial v}{\partial r_{\text{med}}} \right| \Delta r_{\text{med}} + \left| \frac{\partial v}{\partial \sigma} \right| \Delta \sigma \right] = \frac{\Delta s}{s} + \frac{\Delta r_{\text{med}}}{r_{\text{med}}} + 5 \ln \sigma \frac{\Delta \sigma}{\sigma}. \quad (86)$$

The relative uncertainty of the total effective radius retrieval can be estimated as:

$$\frac{\Delta r_{\text{eff}}}{r_{\text{eff}}} \approx \frac{1}{r_{\text{eff}}} \left[ \left| \frac{\partial r_{\text{eff}}}{\partial s} \right| \Delta s + \left| \frac{\partial r_{\text{eff}}}{\partial r_{\text{med}}} \right| \Delta r_{\text{med}} + \left| \frac{\partial r_{\text{eff}}}{\partial \sigma} \right| \Delta \sigma \right] = \frac{\Delta r_{\text{med}}}{r_{\text{med}}} + 5 \ln \sigma \frac{\Delta \sigma}{\sigma}. \quad (87)$$

The relative uncertainty of the total effective variance retrieval can be estimated as:

$$\frac{\Delta v_{\text{eff}}}{v_{\text{eff}}} \approx \frac{1}{v_{\text{eff}}} \left[ \left| \frac{\partial v_{\text{eff}}}{\partial s} \right| \Delta s + \left| \frac{\partial v_{\text{eff}}}{\partial r_{\text{med}}} \right| \Delta r_{\text{med}} + \left| \frac{\partial v_{\text{eff}}}{\partial \sigma} \right| \Delta \sigma \right] = 2 \left( 1 + \frac{1}{v_{\text{eff}}} \right) \ln \sigma \frac{\Delta \sigma}{\sigma}. \quad (88)$$

The terms  $\frac{\Delta s}{s}$ ,  $\frac{\Delta r_{\text{med}}}{r_{\text{med}}}$ , and  $\frac{\Delta \sigma}{\sigma}$  have noticeably different magnitudes in Eqs. (85)–(88). The component  $\frac{\Delta s}{s}$  is statistically fairly small and nearly twice as large as the relative uncertainty of the measured extinction coefficients [6]. The dynamic range of the geometric standard deviation  $\sigma$  is quite narrow and varies from approximately 1.5 to 2.5. The impact of  $\ln \sigma \frac{\Delta \sigma}{\sigma}$  on its relative uncertainty might reach few dozens of percent [6]. The most significant influence on the uncertainties has the  $\frac{\Delta r_{\text{med}}}{r_{\text{med}}}$  component because its relative uncertainty might be hundreds of

percent [6]. The influence of  $\frac{\Delta r_{\text{med}}}{r_{\text{med}}}$  on the uncertainty is doubled for the total number concentration compared to the volume concentration. In our opinion, this is the reason why the total number concentration will be retrieved statistically worse than the other microphysical size parameters considered in this ATBD. Compared to the other parameters,  $\frac{\Delta n}{n}$  is twice as much affected by the potentially large uncertainty of estimation of the count median radius ( $r_{\text{med}}$ ). The absence of the term  $\frac{\Delta s}{s}$  in Eq. (87) theoretically implies that effective radius will be retrieved slightly better than the total volume concentration but significantly worse than the surface area concentration. The missing single component  $\frac{\Delta s}{s}$  is definitely much smaller than  $\frac{\Delta r_{\text{med}}}{r_{\text{med}}} + 5 \ln \sigma \frac{\Delta \sigma}{\sigma}$ .

The uncertainty of the effective variance is a special case because it depends also on the value of effective variance itself [see Eq. (88)]. The smaller the effective variance, the larger the factor  $1 + \frac{1}{v_{\text{eff}}}$ , magnifying the impact of  $\ln \sigma \frac{\Delta \sigma}{\sigma}$  component, which becomes critical because the values of effective variance are normally  $< 1$ . Considering this, we can expect that the quality of the effective variance retrieval will be slightly better compared to the total number concentration, but significantly worse compared to surface area and volume concentrations, and effective radius.

## 6.4. The cross-talk between effective radius and total number concentration

In Section “6.3. Preliminary Uncertainty Analysis” we provided theoretical basis why the relative uncertainty of the surface area concentration retrieval will be the smallest compared to all other microphysical parameters. From this we can establish the link between the true parameters of PSD  $\{n_0, r_{\text{med},0}, \sigma_0\}$  and the virtual estimated parameters  $\{n_1, r_{\text{med},1}, \sigma_1\}$  as:

$$n_0 r_{\text{med},0}^2 \exp(2 \ln^2 \sigma_0) \approx n_1 r_{\text{med},1}^2 \exp(2 \ln^2 \sigma_1). \quad (89)$$

Equation (89) can help us to explain the cross-talk between effective radius and total number concentration [4]. Let us introduce the under/overestimation rates as:

$$k_n = \frac{n_1}{n_0}, \quad k_r = \frac{r_{\text{med},1}}{r_{\text{med},0}}, \quad k_\sigma = \frac{\sigma_1}{\sigma_0}, \quad k_{r_{\text{eff}}} = \frac{r_{\text{eff},1}}{r_{\text{eff},0}}. \quad (90)$$

where

$$r_{\text{eff},0} = r_{\text{med},0} \exp(2.5 \ln^2 \sigma_0), \quad r_{\text{eff},1} = r_{\text{med},1} \exp(2.5 \ln^2 \sigma_1). \quad (91)$$

The rates [see Eq. (90)] exceed one if the corresponding parameters are overestimated and stay below one if underestimated. Equation (89) can be rearranged to estimate the count median radius rate as:

$$k_r \approx \exp(\ln^2 k_\sigma + 2 \ln \sigma_0 \ln k_\sigma - 0.5 \ln k_n). \quad (92)$$

In its turn, the under/overestimation rate for the effective radius can be estimated as:

$$k_{r_{\text{eff}}} = k_r \exp(2.5 \ln^2 k_\sigma + 5 \ln \sigma_0 \ln k_\sigma) \approx \exp(3.5 \ln^2 k_\sigma + 7 \ln \sigma_0 \ln k_\sigma - 0.5 \ln k_n). \quad (93)$$

One can see that both Eqs. (92) and (93) in general case depend on the true value of the geometric standard deviation  $\sigma_0$ . Figure 10 shows five scenarios that cover the narrow ( $\sigma_0 = 1.5$ )

and the wide ( $\sigma_0 = 2.5$ ) monomodal logarithmic-normal PSDs [5], plus the 10% underestimation ( $k_\sigma = 0.9$ ) and the 10% overestimation ( $k_\sigma = 1.1$ ) of geometric standard deviation. We consider the realistic situation when the total number concentration rate varies from 0.2 ( $n_1$  is five times underestimated in the retrievals compared to the true value  $n_0$ ) to 5 ( $n_1$  is five times overestimated) [5].

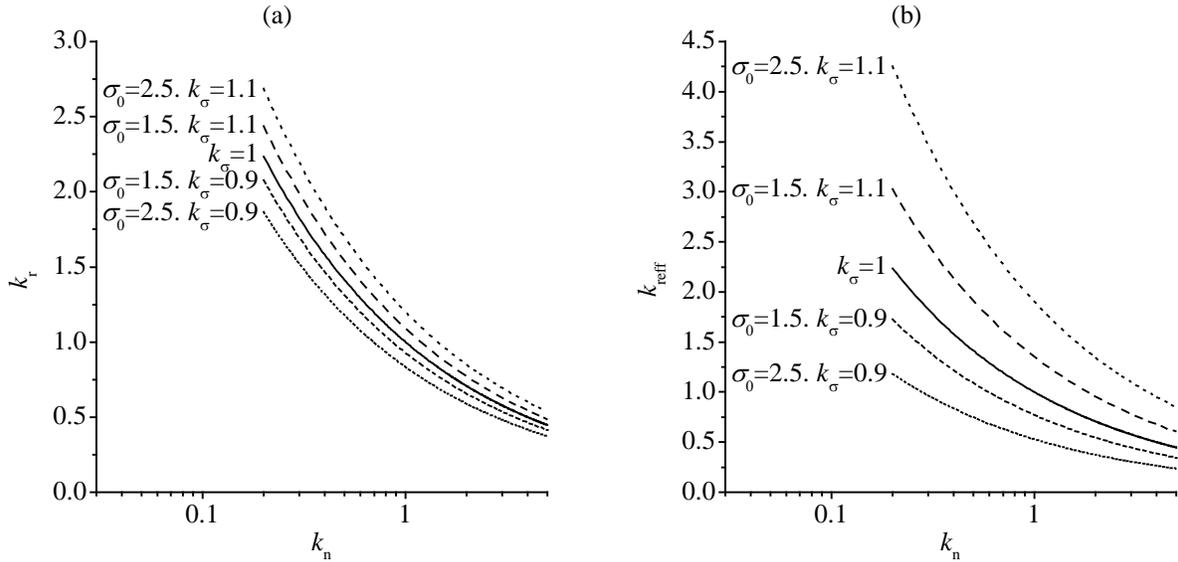


Figure 10. Dependence of the under/overestimation rates.

According to Fig. 10.b, the general rule can be stated as: if the total number concentration was overestimated ( $k_n > 1$ ) during the microphysical retrievals then the effective radius will be underestimated ( $k_{\text{r,eff}} < 1$ ). The underestimation ( $k_n < 1$ ) of total number concentration will lead to the overestimation of effective radius ( $k_{\text{r,eff}} > 1$ ).

## 6.5. Simulation Results

Figure 11 shows the normalized to one monomodal logarithmic-normal PSDs that were used in our simulations. Monomodal logarithmic-normal PSDs in terms of number/volume concentration (see Section “2. Inputs and Outputs”) can be characterized by three parameters (see Section “6.2. Logarithmic-Normal Particle Size Distribution”). One parameter is the count median radius  $r_{\text{med}}$  which defines the radius at which half of the particles in PSD have radii lower than this median radius and the other half of the particles have a radius larger than the median radius. The second parameter is the geometric standard deviation  $\sigma$  which defines the spread of the PSD, i.e., the number of particles per radius interval between the minimum-chosen particle radius and the maximum-chosen particle radius. The third parameter  $n_0$  is the number of particles per  $\text{cm}^3$ .

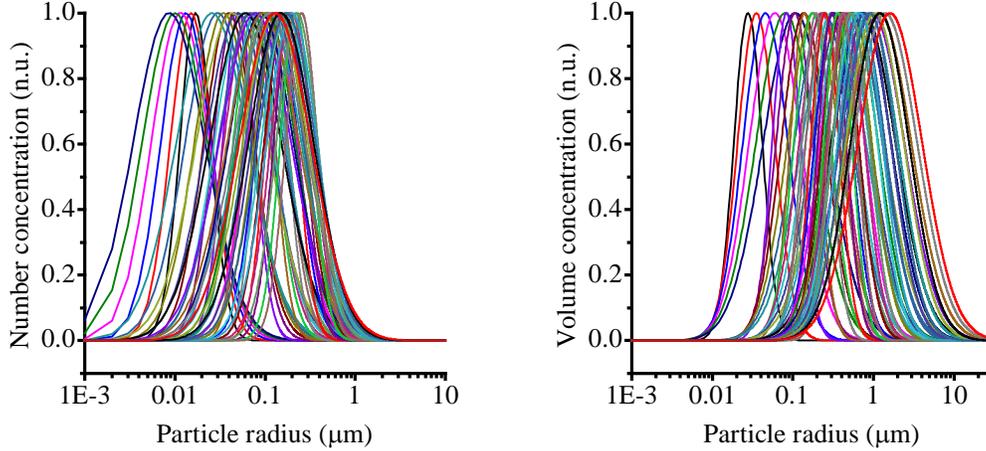


Figure 11. PSDs that were tested in our simulation studies.

We used PSDs in terms of volume concentration representation in our simulations. Figure 11 shows that selected PSDs cover not only the fine-mode range, but also the coarse-mode range. We define 500 nm particle radius as separation point between the fine-mode and coarse-mode particles.

Table 7 presents the numerical values of the parameters of the PSDs that we used in our simulations. The optical data were computed from these size distributions with the assumption of one particle per  $\text{cm}^3$ . We do not use other values of particle number concentration  $n_0$  as this parameter only serves as multiplication factor of the intensive properties of the investigated PSDs, i.e., number, surface area, and volume concentrations.

Table 7: Input parameters of the size distributions used in the simulation studies. We used PSDs normalized to one particle per  $\text{cm}^3$

Count median radius $r_{\text{med}}$ (nm)	Effective radius $r_{\text{eff}}$ ( $\mu\text{m}$ )	Surface area concentration $s$ ( $\mu\text{m}^2/\text{cm}^3$ )	Volume concentration $v$ ( $\mu\text{m}^3/\text{cm}^3$ )
20	0.03;0.04;0.06;0.08;0.11;0.16	$7\text{e-}3$ ; $8.8\text{e-}3$ ; $1.2\text{e-}2$ ; $1.5\text{e-}2$ ; $2\text{e-}2$ ; $2.7\text{e-}2$	$7\text{e-}5$ ; $1.2\text{e-}4$ ; $2.1\text{e-}4$ ; $4\text{e-}4$ ; $7.6\text{e-}4$ ; $1.5\text{e-}3$
60	0.09;0.12;0.17;0.24;0.34;0.49	$6.3\text{e-}2$ ; $0.08$ ; $0.10$ ; $0.14$ ; $0.18$ ; $0.24$	$2\text{e-}3$ ; $3.2\text{e-}3$ ; $5.8\text{e-}3$ ; $1.1\text{e-}2$ ; $2.1\text{e-}2$ ; $0.04$
100	0.15;0.20;0.28;0.40;0.57;0.81	$0.18$ ; $0.22$ ; $0.29$ ; $0.38$ ; $0.50$ ; $0.67$	$8.8\text{e-}3$ ; $1.5\text{e-}2$ ; $2.7\text{e-}2$ ; $5\text{e-}2$ ; $9.5\text{e-}2$ ; $0.18$
140	0.21;0.28;0.39;0.55;0.79;1.14	$0.34$ ; $0.43$ ; $0.56$ ; $0.74$ ; $0.99$ ; $1.32$	$2.4\text{e-}2$ ; $4.1\text{e-}2$ ; $7.3\text{e-}2$ ; $0.14$ ; $0.26$ ; $0.50$
180	0.27;0.36;0.50;0.71;1.02;1.47	$0.57$ ; $0.72$ ; $0.93$ ; $1.22$ ; $1.63$ ; $2.18$	$5.1\text{e-}2$ ; $8.7\text{e-}2$ ; $0.16$ ; $0.29$ ; $0.55$ ; $1.07$
220	0.33;0.45;0.62;0.87;1.25;1.79	$0.85$ ; $1.07$ ; $1.39$ ; $1.83$ ; $2.44$ ; $3.26$	$9.4\text{e-}2$ ; $0.16$ ; $0.29$ ; $0.53$ ; $1.01$ ; $1.95$
260	0.39;0.53;0.73;1.03;1.47;2.12	$1.18$ ; $1.49$ ; $1.94$ ; $2.55$ ; $3.40$ ; $4.55$	$0.15$ ; $0.26$ ; $0.47$ ; $0.88$ ; $1.67$ ; $3.22$
300	0.45;0.61;0.84;1.19;1.70;2.44	$1.57$ ; $1.99$ ; $2.58$ ; $3.40$ ; $4.53$ ; $6.06$	$0.24$ ; $0.40$ ; $0.72$ ; $1.35$ ; $2.56$ ; $4.94$
$\sigma$	1.5; 1.7; 1.9; 2.1; 2.3; 2.5		
$m_R$	1.4; 1.5; 1.6; 1.7		
$m_i$	0 <i>i</i> ; 0.0001 <i>i</i> ; 0.001 <i>i</i> ; 0.0025 <i>i</i> ; 0.005 <i>i</i> ; 0.0075 <i>i</i> ; 0.01 <i>i</i> ; 0.015 <i>i</i> ; 0.02 <i>i</i> ; 0.025 <i>i</i> ; 0.03 <i>i</i> ; 0.035 <i>i</i> ; 0.04 <i>i</i> ; 0.045 <i>i</i> ; 0.05 <i>i</i>		

The chosen values for median radius and geometric standard deviation cover a comparably wide range of particle scenarios. The values of effective radius are well within the range of numbers found for the fine-mode fraction of PSDs. We also cover to some extent particles in the Aitken

and coarse modes. However, we cannot provide reliable results of microphysical parameters if the investigated PSD contains a high share of particles below 50 nm radius. The smallest laser wavelength is 355 nm, which results in large uncertainties of the retrieved PSDs at small particle radii.

We also restricted the maximum value of effective radius in our simulations. One reason is the fact that the version of LUT which was developed for TiARA and used in this simulation study did not include particles with radii larger than 10  $\mu\text{m}$ . This value restricts the maximum retrievable effective radius to approximately 2.5  $\mu\text{m}$ .

Table 8 shows that we used measurement-channel-independent uncertainties of 0%, 5%, 10%, 15%, 20%, and 25%. The case of error free data was investigated too. We hypothesize that the uncertainty of the data products in that case is driven by the mathematically-induced uncertainty of TiARA.

Table 8: Error scenarios of the simulation studies. EEM denotes the extreme error computation model. GEM denotes the Gauss error computation model

	$\beta_{355}$ $\beta_{532}$ $\beta_{1064}$ $\alpha_{355}$ $\alpha_{532}$
EEM 5	5% for all five input data
EEM 10	10% for all five input data
EEM 15	15% for all five input data
EEM 20	20% for all five input data
EEM 25	25% for all five input data
GEM 5, one sigma percentile	5% and 1.7% for all five input data
GEM 10, one sigma percentile	10% and 3.3% for all five input data
GEM 15, one sigma percentile	15% and 5% for all five input data
GEM 20, one sigma percentile	20% and 6.6% for all five input data
GEM 25, one sigma percentile	25% and 8% for all five input data

We used two types of uncertainty analysis. Details can be found elsewhere [19]. Briefly, the uncertainty model we mainly used in the simulations is denoted as "extreme error computation model" (EEM) which has been used for the manual version of the inversion method in the past 20 years [16]. This method allows us to compute the uncertainty of the data products on the basis of a very simple assumption: if the slope of the extinction-related Ångström and/or backscatter-related Ångström exponent changes, the microphysical parameters also change, though not necessarily in a proportional way. We are aware that EEM comes with certain flaws. For example, it is an intuitive model in which we assume that extreme distortions of the spectra of backscatter and extinction coefficients result in proportional distortions of the solution space of microphysical parameters. This may not always be true and we lack in a theoretical model that could give us more insight into the properties of the final solution spaces. However, the model has been successfully applied [23], and is our preferred model for uncertainty analysis. We tested another error model that could be used in future uncertainty analysis. We call it "Gauss-distributed error computation model" (GEM). The uncertainty bars have a Gaussian-curve like shape. We started exploring if such a model could provide us with a mathematically sound uncertainty analysis and even replace EEM in future versions of TiARA.

### 6.5.1. Case Study

Figure 12 shows the results for effective radius (first and second row) and number (third and fourth row) concentration. For each microphysical parameter we show the results for the case of error free optical input data and the case of 15% measurement error in each channel. Each retrieval result is shown in terms of its mean value (symbols) and standard deviation from the mean value (uncertainty bars). Figure 13 shows the results for surface area (first and second row) and volume concentrations (third and fourth row). Figure 14 shows the results for complex refractive index. This example serves as illustration of the general features of TiARA.

The 1-1 lines indicate perfect reproduction of the theoretical values of the investigated parameters. The green lines show that the retrieved data products would need to be reduced, respectively increased by 20% in order to be on the 1-1 line. The red lines indicate the situation in which the retrieved values would need to be reduced by a factor of two, respectively doubled to be on the 1-1 line.

Effective radius usually remains within the red sector of the true results if the input optical data are error free. There are some outliers who seem to occur for the case of low real parts and increasing imaginary part. The reason for these comparably strong outliers is unknown. We also find that the retrieved effective radius remains within the green sector, at least up to an effective radius of 0.5  $\mu\text{m}$ , except for a few outliers.

50% is the upper value of the statistical uncertainty of the mean value of effective radius. In many cases this statistical noise is considerably less. The outliers which are mainly related to the lowest real part tested in this study, i.e. 1.4, also show the strongest statistical uncertainty.

15% uncertainty of the optical data reduces the retrieval accuracy. The mean values remain in the red sector if effective radius is less than 0.5  $\mu\text{m}$  and the imaginary part is  $\leq 0.01i$ . For higher absorbing particles, i.e. imaginary parts larger than  $0.01i$ , there is a possibility that effective radius is overestimated by more than a factor two. Statistical uncertainty also increases if measurement errors are 15%.

In summary, we find that PSDs with particle effective radii below approximately 0.5  $\mu\text{m}$  can be retrieved to at least 50% accuracy and better than 50% precision if particle absorption is moderate, i.e. imaginary parts are  $\leq 0.01i$ . If particles are highly light-absorbing, there is an increasing chance of overestimating particle effective radius.

The mean value of number concentration for the most part remains in the red sector. TiARA tends to underestimate number concentration if the input optical data are error free (see Fig. 12). The underestimation seems to grow with increasing imaginary part. The statistical error usually is less than 50%, in part it is significantly less than 50%. It seems that accuracy improves with increasing real part.

15% measurement uncertainty increases the pattern of underestimating number concentration. As in the case of error free data, number concentration tends to be stronger underestimated if the imaginary part increases. Statistical uncertainty seems to increase, too, compared to the case of error free optical data. This is particularly obvious for the cases for which number concentration is overestimated.

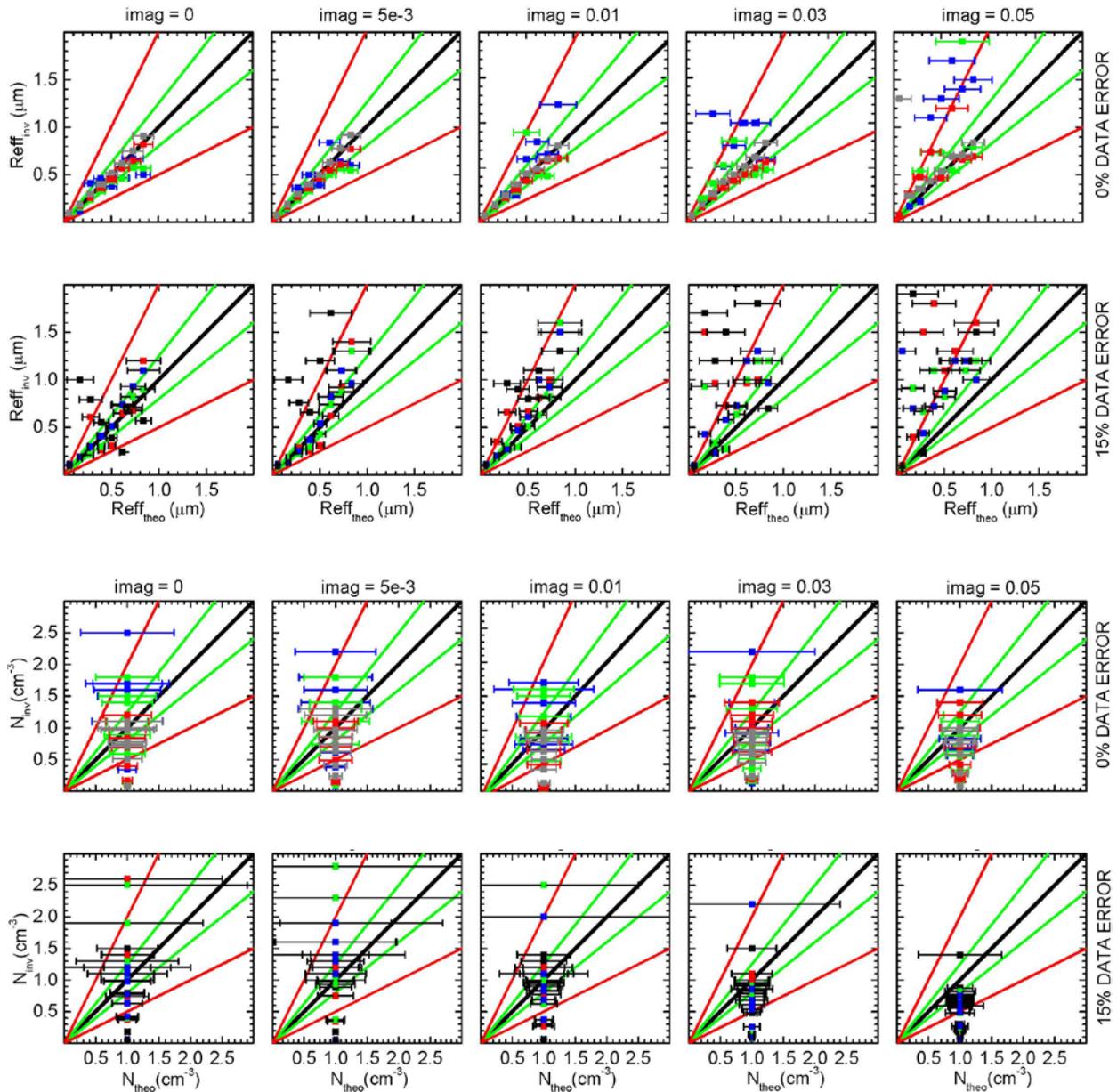


Figure 12. Example of retrieval accuracy (bias) and precision (noise) for the case of a size distribution with geometric standard deviation 1.9. Shown are the results for eight mean radii: 20, 60, 100, 140, 180, 220, 260, and 300 nm. The plots in each row show the result for the imaginary part  $0i$ ,  $0.005i$ ,  $0.01i$ ,  $0.03i$ , and  $0.05i$ . The first row shows the results for effective radius and 0% data uncertainty. The second row shows the results for 15% data uncertainty. The third and fourth rows show the results for number concentration, again separated according to 0% data uncertainty and 15% data uncertainty. We used the extreme error model. The colored lines describe by how much the inversion results need to be decreased/increased until they fall onto the 1-1 line. Green = reduction/increase by 20% (denoted in the text as green sector); red = reduction/increase by factor two (red sector). The squares describe the results of the four real parts tested in the simulations: 1.4 (blue), 1.5 (green), 1.6 (red), and 1.7 (grey).

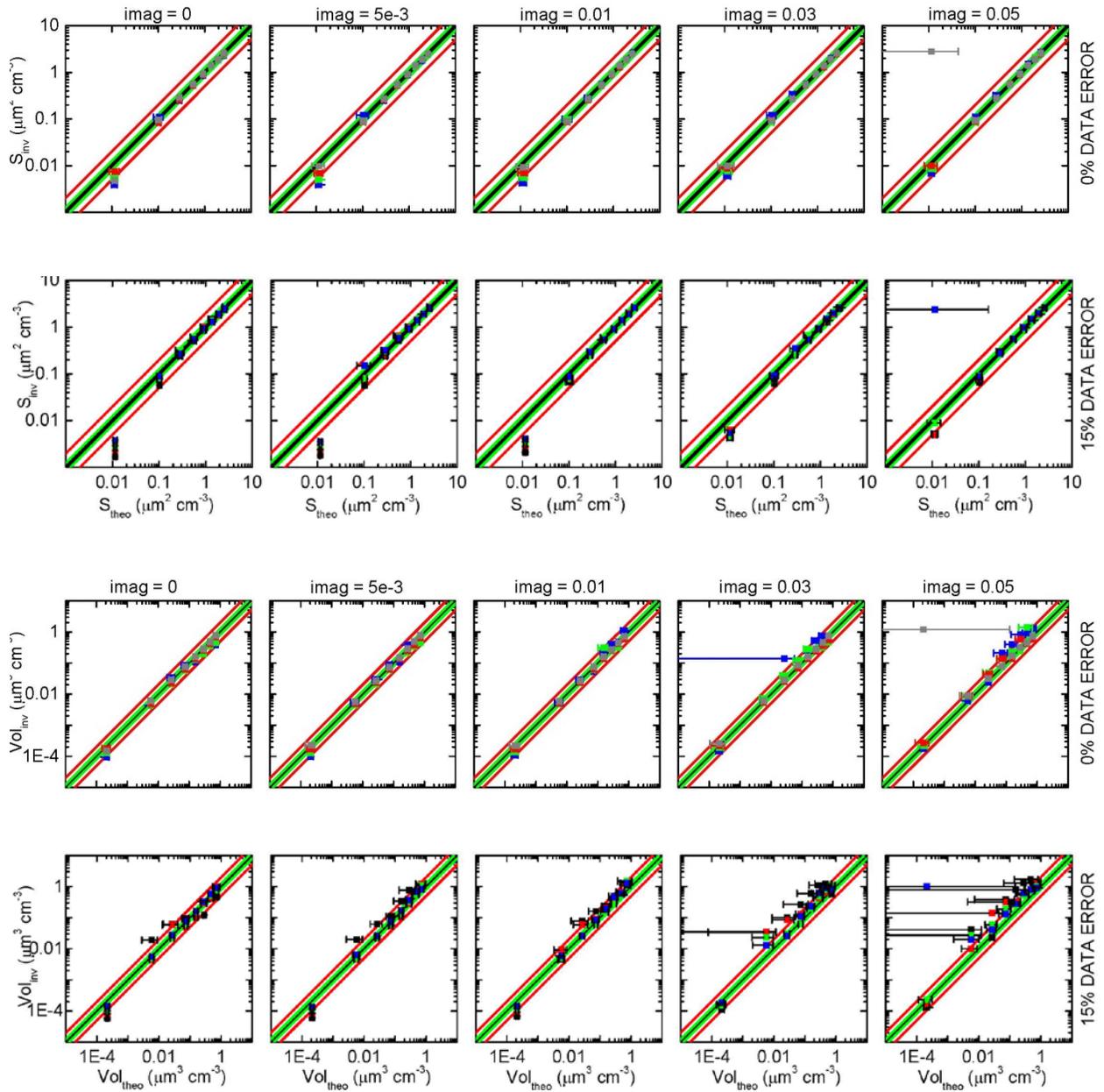


Figure 13. Same as in Fig. 12 but for the surface area concentration (rows one and two) and the volume concentration (rows three and four). Meaning of the symbols and colors for surface area and volume concentrations is the same as in Fig. 12.

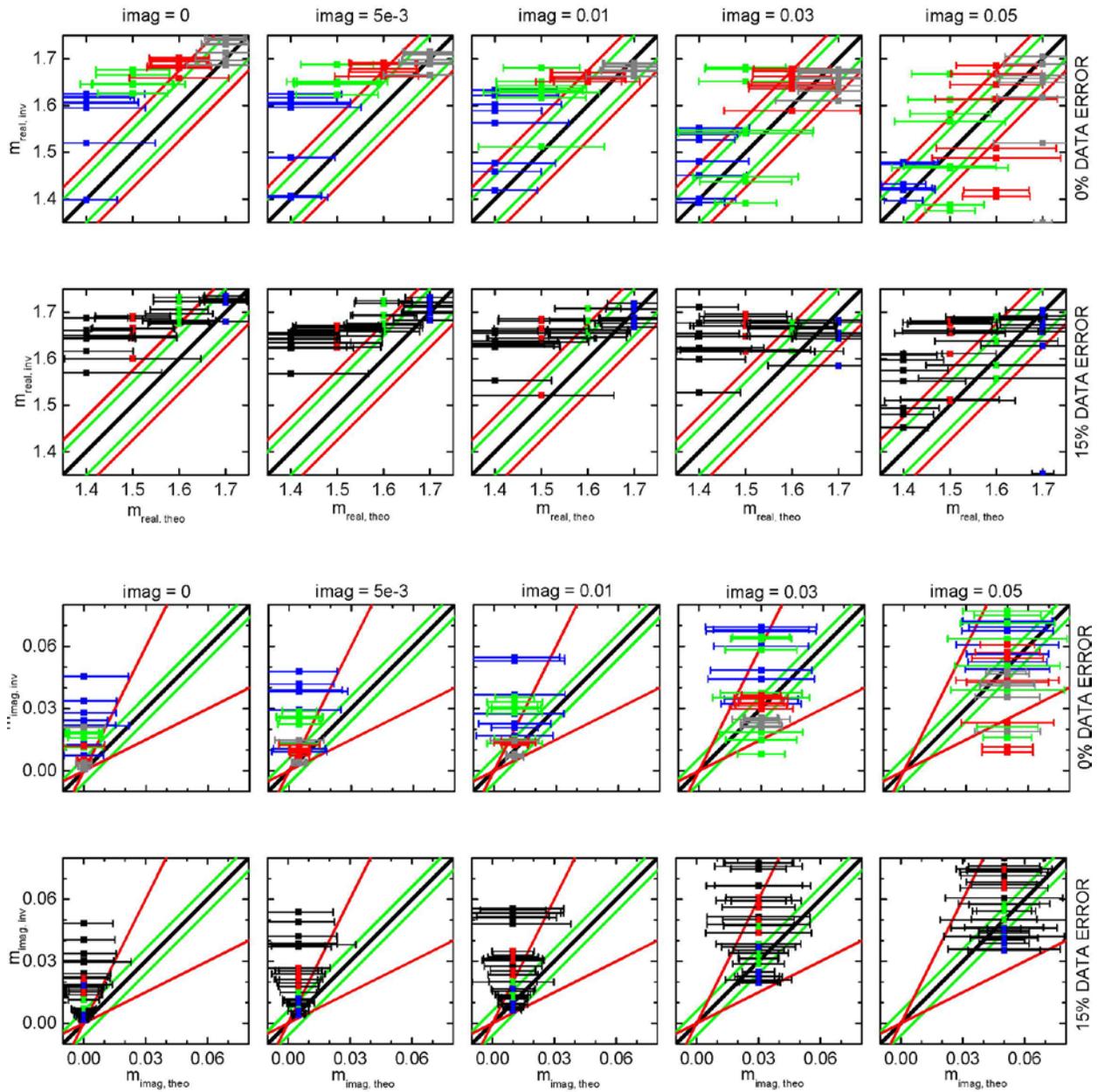


Figure 14. Same as in Fig. 12 but for the real part (rows one and two) and the imaginary part (rows three and four) of the complex refractive index. Black = 1-1 line. Meaning of the colored lines in the plots is as follows. Real part: green = reduction/increase by 0.05 and red = reduction/increase by 0.075 before results coincide with the true values. Imaginary part: green = deviation of  $\pm 0.006i$  from true value; red = imaginary part needs to be reduced/increased by factor two until it coincides with the 1-1 line.

As noted in previous studies [5, 11, 16, 17, 18, 27, 28] we find exceptionally accurate inversion results for surface area concentration. An explanation for this mechanism has recently been published [6, 12, 13]. Values of surface area concentration cover approximately three orders of magnitude in our simulation study. Regardless of the true imaginary part, the accuracy is high. The results for surface area concentration are inside the red sector, even if measurement errors are 15%. The only exception is very low surface area concentrations. In this case we significantly underestimate surface area concentration if imaginary parts are  $\leq 0.01i$ . In contrast, highly light-absorbing PSDs can be retrieved comparably accurately.

We are investigating these outliers that appear for low imaginary parts. One reason for this behavior could be that we cannot resolve in a trustworthy manner PSDs below 50 nm particle radius which would lead to an underestimation of surface area concentration. In fact, number concentration would be underestimated even more in such a case. We found this underestimation of number concentration for PSDs that consist of a comparably high share of small particles below 50 nm radius. We will address this issue in our next performance test of TiARA which will use an upgraded version of LUT tables and an improved solution-space post-processing scheme [12, 13].

We find accurate inversion results of volume concentration across four orders of magnitude if the optical data are error free. The quality of the inversion results seems to decrease for imaginary parts of  $0.05i$ , i.e., the case of highly light-absorbing particles. We do not know the reason for the two outliers seen in the plot for the imaginary part  $0.03i$  and  $0.05i$  (see Fig. 13). Statistical uncertainty of the retrieved values is always below 50% except for these two outliers. In most cases the statistical uncertainty is less than 20–30%.

The accuracy of the inversion results slightly worsens for 15% measurement uncertainty if the imaginary part is  $\leq 0.01i$ . TiARA tends to overestimate volume concentration. Mean values still remain within the red sector except for very small volume concentrations where we find a similar pattern to the one described for surface area concentration, i.e., volume concentration may be underestimated by more than 50%. However, as in the case of surface area concentration this underestimation seems to disappear for large imaginary parts, i.e.,  $>0.01i$ . At the same time, TiARA seems to start overestimating volume concentration for large imaginary parts of  $0.03i$  and  $0.05i$ . The statistical noise also increases significantly.

Figure 14 (first and second rows) shows the results for the retrieved real part. The true value was 1.6 in this simulation case. We used a search grid from 1.325–1.8 (real part) and  $0i$ – $0.05i$  (imaginary part) in our simulations (see Fig. 2). We tested four different real parts: 1.4, 1.5, 1.6, and 1.7. As final goal of our algorithm development we target a retrieval accuracy of 0.05 to 0.075 for the real part. With regards to the imaginary part we aim at 50% retrieval accuracy as minimum goal. The main challenge in that regard lies in retrieving low values of the imaginary part which we define as  $0.01i$ . We cannot use negative imaginary parts in the inversion. Based on the current way how we select the final solution space we naturally end up with an overestimation of the imaginary part. This overestimation is particularly strong the closer to  $0i$  the true value is.

TiARA tends to overestimate the real part by more than 0.075 if the true value is 1.4 and the input optical data are error free. This overestimation disappears for larger real part. The range of retrieved real parts for the different size distributions (effective radii) seems to vary in dependence of the underlying imaginary part. For example, in the case of the true value 1.6 (red

symbols) we see that all solutions are clustered between approximately 1.65 and 1.7 if imaginary parts are small. In contrast, if the imaginary part is  $0.05i$  the solutions are spread from 1.4 to 1.7. If the true value is 1.4 this pattern is nearly opposite, i.e., real parts vary across a wide range of numbers for low imaginary parts. In contrast retrieved real parts seem to accumulate between 1.4 and 1.5 if the imaginary part is  $0.05i$ .

In summary, we find the following pattern of retrieval accuracy: the real part is overestimated, particularly if real parts are 1.4 or 1.5. Real parts are underestimated if values are 1.6 and 1.7. Statistical uncertainty can be as high as  $\pm 0.1$ . Such values are larger than what we need to achieve with TiARA in view of the accuracy requirements for single-scattering albedo, see e.g. [18]. We are exploring methods that allow us to reduce the uncertainty, e.g. [5].

In the case of 15% measurement error TiARA seems to generally overestimate the real part. The majority of retrieved real parts is at/above 1.6. We currently have no explanation for this pattern, i.e., a general overestimation when we move from error free to erroneous optical input data. An insensitivity of TiARA with regards to the real part retrieval might explain this pattern only in part.

Another reason could be where the true value of effective radius, respectively the PSD is located with respect to the particle radius grid. The analysis of this pattern however goes beyond the main goal of this ATBD, which is about identifying typical patterns and the general applicability of TiARA. However, these patterns form the basis for a more refined analysis of our simulations. We hope that a better understanding of these patterns will help us improving TiARA in future. First results in that regard have been published in recent years [5, 12, 13, 18].

Figure 14 (rows three and four) shows the results for the retrieved imaginary parts. If the optical data are error free, TiARA overestimates the imaginary part. This pattern is particularly strong for low imaginary parts. If the imaginary part is high, i.e.,  $0.03i$  and  $0.05i$  accuracy improves. The mean values are within the red sector. Still, we find on average an overestimation of the imaginary part in these two cases.

The situation is similar in case of 15% measurement error. The imaginary part is largely overestimated if the true values are  $0i$ ,  $0.005i$ , and  $0.01i$ , respectively. However, we find some (minor) differences compared to the case of 0% measurement uncertainty. The mean value for the most part stays within  $\pm 0.006i$  if the real part is 1.4 or 1.5. A more detailed analysis needs to be carried out on the basis of a larger set of simulations. We can then decide if this is a result that can be generalized, i.e., if better retrieval results for small imaginary parts can be obtained if real parts are small. We find better retrieval results of the imaginary part in the case of 0% data error. Accuracy is better than  $\pm 0.006i$  if real parts are large, i.e., 1.6 and 1.7. If imaginary parts are  $0.03i$  or  $0.05i$  the underestimation of the retrieved values largely disappears. Most of the mean values remain in the red sector. Statistical uncertainty mainly is in the range of 50% if the true imaginary parts are  $0.03i$  or  $0.05i$ . If imaginary parts are smaller ( $0.005i$  or  $0.01i$ ) the statistical uncertainty is as high as 100%. If the true imaginary part is  $0i$  uncertainties can be as high as  $0.02i$ .

In summary: most of the results we obtained for the imaginary part in this study confirm the findings we obtained with the manual version of the inversion algorithm: a) retrieval results are within  $\pm 50\%$  for highly light-absorbing particles, and b) imaginary parts can be retrieved to  $\pm 100\%$  if particles are low light-absorbing. Imaginary parts of  $0i$  cannot be retrieved at the moment.

### 6.5.2. Statistics: Histograms

Figure 15 shows a statistical overview of the deviation between the reconstructed microphysical parameters and the true values for all size distributions tested in this study, including the six error levels, i.e., 0%, 5%, 10%, 15%, 20%, and 25%. The figures show how results differ if we use the EEM (grey-shaded columns and black numbers) and the GEM (green-shaded columns and green numbers). The histogram distributions show the percentage of simulated cases (8 PSDs, 4 real parts, 15 imaginary parts) that deviate by a given percentage from the true results of effective radius, number, surface area, and volume concentrations. We only show results for which the retrieval products deviate  $\leq 50\%$  from the true results. In the case of real and imaginary parts we show absolute deviations from the true results.

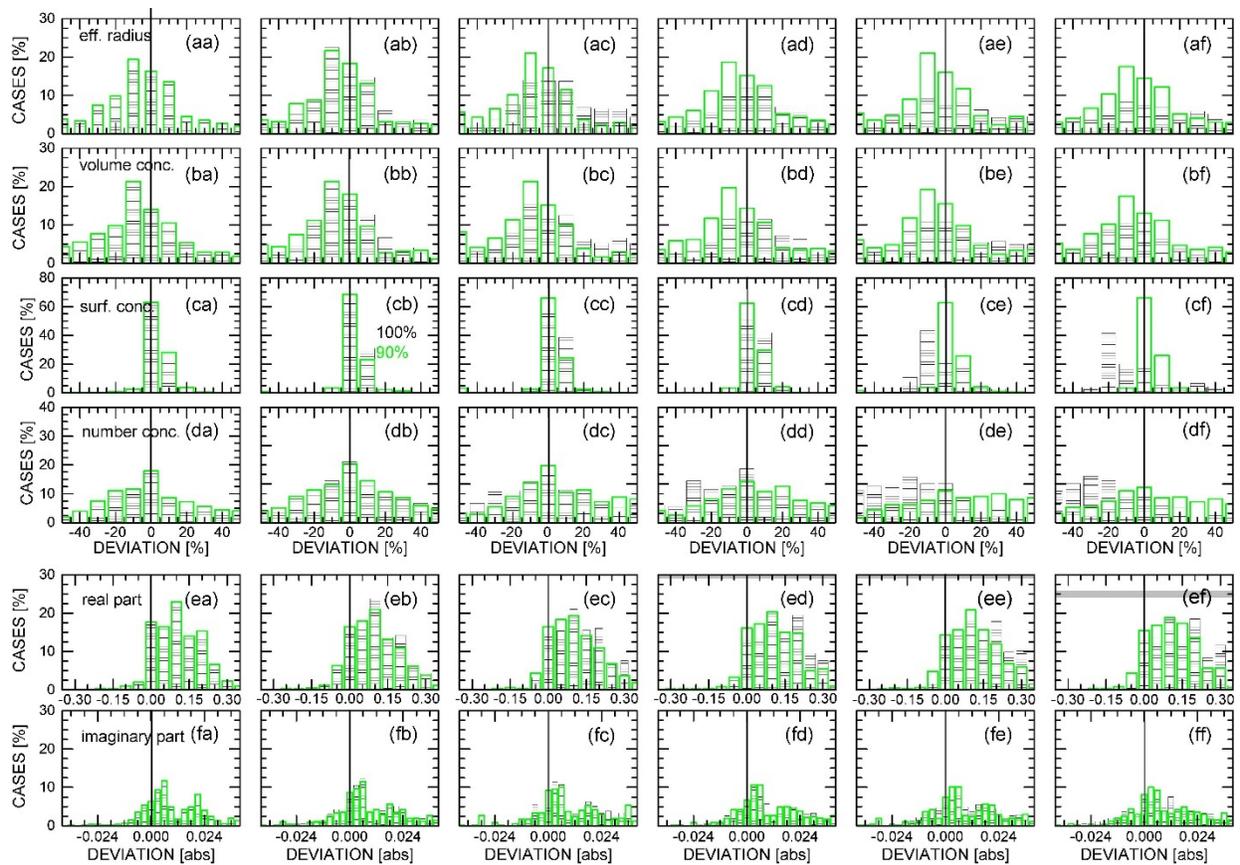


Figure 15. Inversion results for all values of sigma, all mean radii, and all imaginary parts (see Table 7). Each row shows the results for one of the six investigated data products. Each column shows the results for (from left to right): 0%, 5%, 10%, 15%, 20%, and 25% optical data error. The histograms show the percentage deviation from the true value for the cases (aa)–(df) and the absolute errors for the cases (ea)–(ff). We show the results for the EEM (grey-shaded columns) and the GEM (green-shaded columns).

From a qualitative point of view the distribution (spread) of histogram columns generally is wider for the EEM compared to the GEM, i.e., results on average tend toward larger systematic deviations from the true values for the extreme-error model compared to the Gauss-error model. On average the Gauss-error model delivers more results that are close to 0% deviation from the true results compared to the extreme-error model. The inversion results tend to be better, i.e., narrower around the 0%-deviation point for measurement errors of 10–15%. We again notice the exceptionally high retrieval quality of surface area concentration. With regard to the Gauss-error model we notice on average a slight underestimation of the size parameters. The exception is surface area concentration which is overestimated by up to 10%.

With regard to the real and imaginary part we find the following results. We systematically overestimate the real part in all cases considered in our study. Only for case of high imaginary parts we find situations in which we underestimate the real part for some cases. If imaginary parts are large, there is a rather clear systematic behavior with regard to retrieval quality in dependence of uncertainty of the optical input data. The retrieval accuracy drops with increasing uncertainty of the input data. We do not find a significant performance difference of TiARA whether we use the EEM or the GEM in our simulations.

With regard to the imaginary part we again notice an overestimation of the retrieved values compared to the theoretical values. The theoretical values are overestimated in the inversion, regardless of the used error model.

It seems that 15% data uncertainty is an upper threshold for which a reasonable retrieval performance can be achieved for the size parameters. The errors of the optical input data seem to have little influence on the retrieval quality of the complex refractive index. This pattern points to a possibly low sensitivity of TiARA with regard to retrieving the complex refractive index. In contrast, the algorithm is capable of deriving trustworthy values of particle size parameters, i.e., effective radius, number, surface area, and volume concentration.

## Appendix A. Inversion Parameters ASCII File

[Server]

```
***** General parameters of input data *****
// Type of input data. Simulation or experimental.
// If "0" then simulation of log-normal particle size distribution is used.
// If "1" then experimental data are used.
InputDataType=0

:::: First mode for simulations ::::
// Value of count median radius for first simulation mode, mum.
MeanRadius1=0.3
// Geometric standard deviation for first simulation mode.
ModeWidth1=1.4
// Real part of complex refractive index for first simulation mode.
CRReal1=1.33
// Imaginary part of complex refractive index for first simulation mode.
CRImag1=0
// Concentration value for first simulation mode is always equal to 1 1/cm^3.

:::: Second mode for simulations ::::
// Use ("1") or not ("0") the second mode during simulations.
UseMode2=0
// Value of count median radius for second simulation mode, mum.
MeanRadius2=0.5
// Geometric standard deviation for second simulation mode.
ModeWidth2=1.3
// Concentration value for second simulation mode, 1/cm^3.
Concentration2=0
// Real part of complex refractive index for second simulation mode.
CRReal2=1.33
// Imaginary part of complex refractive index for second simulation mode.
CRImag2=0

:::: Third mode for simulations ::::
// Use ("1") or not ("0") the third mode during simulations.
UseMode3=0
// Value of count median radius for third simulation mode, mum.
MeanRadius3=0.9
// Geometric standard deviation for third simulation mode.
ModeWidth3=1.2
// Concentration value for third simulation mode, 1/cm^3.
Concentration3=0
// Real part of complex refractive index for third simulation mode.
CRReal3=1.33
// Imaginary part of complex refractive index for third simulation mode.
CRImag3=0
```

```
***** Experimental optical data *****
::::: Extinction coefficients ::::::
// Use ("1") or not ("0") extinction coefficient No1 for inversion.
UseExtinction01=1
// Wavelength of extinction coefficient No1, nm.
ExtinctionWavelength01=355
// Value of extinction coefficient No1, 1/m.
ExtinctionCoef01=1.09026e-6
// Use ("1") or not ("0") extinction coefficient No2 for inversion.
UseExtinction02=1
// Wavelength of extinction coefficient No2, nm.
ExtinctionWavelength02=532
// Value of extinction coefficient No2, 1/m.
ExtinctionCoef02=7.43007e-7
// Use ("1") or not ("0") extinction coefficient No3 for inversion.
UseExtinction03=0
// Wavelength of extinction coefficient No3, nm.
ExtinctionWavelength03=0
// Value of extinction coefficient No3, 1/m.
ExtinctionCoef03=0
// Use ("1") or not ("0") extinction coefficient No4 for inversion.
UseExtinction04=0
// Wavelength of extinction coefficient No4, nm.
ExtinctionWavelength04=0
// Value of extinction coefficient No4, 1/m.
ExtinctionCoef04=0
// Use ("1") or not ("0") extinction coefficient No5 for inversion.
UseExtinction05=0
// Wavelength of extinction coefficient No5, nm.
ExtinctionWavelength05=0
// Value of extinction coefficient No5, 1/m.
ExtinctionCoef05=0
// Use ("1") or not ("0") extinction coefficient No6 for inversion.
UseExtinction06=0
// Wavelength of extinction coefficient No6, nm.
ExtinctionWavelength06=0
// Value of extinction coefficient No6, 1/m.
ExtinctionCoef06=0
// Use ("1") or not ("0") extinction coefficient No7 for inversion.
UseExtinction07=0
// Wavelength of extinction coefficient No7, nm.
ExtinctionWavelength07=0
// Value of extinction coefficient No7, 1/m.
ExtinctionCoef07=0
// Use ("1") or not ("0") extinction coefficient No8 for inversion.
UseExtinction08=0
// Wavelength of extinction coefficient No8, nm.
ExtinctionWavelength08=0
// Value of extinction coefficient No8, 1/m.
ExtinctionCoef08=0
// Use ("1") or not ("0") extinction coefficient No9 for inversion.
```

```
UseExtinction09=0
// Wavelength of extinction coefficient No9, nm.
ExtinctionWavelength09=0
// Value of extinction coefficient No9, 1/m.
ExtinctionCoef09=0
// Use ("1") or not ("0") extinction coefficient No10 for inversion.
UseExtinction10=0
// Wavelength of extinction coefficient No10, nm.
ExtinctionWavelength10=0
// Value of extinction coefficient No10, 1/m.
ExtinctionCoef10=0

:::: Backscatter coefficients ::::
// Use ("1") or not ("0") backscatter coefficient No1 for inversion.
UseBackscatter01=1
// Wavelength of backscatter coefficient No1, nm.
BackscatterWavelength01=355
// Value of backscatter coefficient No1, 1/(m•sr).
BackscatterCoef01=7.94049e-9
// Use ("1") or not ("0") backscatter coefficient No2 for inversion.
UseBackscatter02=1
// Wavelength of backscatter coefficient No2, nm.
BackscatterWavelength02=532
// Value of backscatter coefficient No2, 1/(m•sr).
BackscatterCoef02=6.9284e-9
// Use ("1") or not ("0") backscatter coefficient No3 for inversion.
UseBackscatter03=1
// Wavelength of backscatter coefficient No3, nm.
BackscatterWavelength03=1064
// Value of backscatter coefficient No3, 1/(m•sr).
BackscatterCoef03=1.67199e-9
// Use ("1") or not ("0") backscatter coefficient No4 for inversion.
UseBackscatter04=0
// Wavelength of backscatter coefficient No4, nm.
BackscatterWavelength04=0
// Value of backscatter coefficient No4, 1/(m•sr).
BackscatterCoef04=0
// Use ("1") or not ("0") backscatter coefficient No5 for inversion.
UseBackscatter05=0
// Wavelength of backscatter coefficient No5, nm.
BackscatterWavelength05=0
// Value of backscatter coefficient No5, 1/(m•sr).
BackscatterCoef05=0
// Use ("1") or not ("0") backscatter coefficient No6 for inversion.
UseBackscatter06=0
// Wavelength of backscatter coefficient No6, nm.
BackscatterWavelength06=0
// Value of backscatter coefficient No6, 1/(m•sr).
BackscatterCoef06=0
// Use ("1") or not ("0") backscatter coefficient No7 for inversion.
UseBackscatter07=0
```

```
// Wavelength of backscatter coefficient No7, nm.
BackscatterWavelength07=0
// Value of backscatter coefficient No7, 1/(m•sr).
BackscatterCoef07=0
// Use ("1") or not ("0") backscatter coefficient No8 for inversion.
UseBackscatter08=0
// Wavelength of backscatter coefficient No8, nm.
BackscatterWavelength08=0
// Value of backscatter coefficient No8, 1/(m•sr).
BackscatterCoef08=0
// Use ("1") or not ("0") backscatter coefficient No9 for inversion.
UseBackscatter09=0
// Wavelength of backscatter coefficient No9, nm.
BackscatterWavelength09=0
// Value of backscatter coefficient No9, 1/(m•sr).
BackscatterCoef09=0
// Use ("1") or not ("0") backscatter coefficient No10 for inversion.
UseBackscatter10=0
// Wavelength of backscatter coefficient No10, nm.
BackscatterWavelength10=0
// Value of backscatter coefficient No10, 1/(m•sr).
BackscatterCoef10=0

// Usage of "extreme" noise for the input optical data distortion.
// If "0" then "extreme" distortion will not be used.
// If "1" then "extreme" distortion will be used.
UseExtremeDistortion=1

***** Extreme error levels *****
:::: Backscatter coefficients ::::
// Value of "extreme" distortion for backscatter coefficient No1, %.
BackscatterExtreme01=10
// Value of "extreme" distortion for backscatter coefficient No2, %.
BackscatterExtreme02=5
// Value of "extreme" distortion for backscatter coefficient No3, %.
BackscatterExtreme03=15
// Value of "extreme" distortion for backscatter coefficient No4, %.
BackscatterExtreme04=0
// Value of "extreme" distortion for backscatter coefficient No5, %.
BackscatterExtreme05=0
// Value of "extreme" distortion for backscatter coefficient No6, %.
BackscatterExtreme06=0
// Value of "extreme" distortion for backscatter coefficient No7, %.
BackscatterExtreme07=0
// Value of "extreme" distortion for backscatter coefficient No8, %.
BackscatterExtreme08=0
// Value of "extreme" distortion for backscatter coefficient No9, %.
BackscatterExtreme09=0
// Value of "extreme" distortion for backscatter coefficient No10, %.
BackscatterExtreme10=0
```

```
::::: Extinction coefficients ::::
// Value of "extreme" distortion for extinction coefficient No1, %.
ExtinctionExtreme01=10
// Value of "extreme" distortion for extinction coefficient No2, %.
ExtinctionExtreme02=10
// Value of "extreme" distortion for extinction coefficient No3, %.
ExtinctionExtreme03=0
// Value of "extreme" distortion for extinction coefficient No4, %.
ExtinctionExtreme04=0
// Value of "extreme" distortion for extinction coefficient No5, %.
ExtinctionExtreme05=0
// Value of "extreme" distortion for extinction coefficient No6, %.
ExtinctionExtreme06=0
// Value of "extreme" distortion for extinction coefficient No7, %.
ExtinctionExtreme07=0
// Value of "extreme" distortion for extinction coefficient No8, %.
ExtinctionExtreme08=0
// Value of "extreme" distortion for extinction coefficient No9, %.
ExtinctionExtreme09=0
// Value of "extreme" distortion for extinction coefficient No10, %.
ExtinctionExtreme10=0

// Type of distribution of particle radii grid bins.
// If "E" then equidistant distribution.
// If "L" then logarithmic-equidistant distribution.
GridBinsDistr=L

// Type of kernel functions to be used during inversion.
// N - number kernel functions.
// S - surface area kernel functions.
// V - volume kernel functions.
KernelType=V

// Define manually number of internal (between lower and upper limits) particle radii grid bins.
// If "0" then number of internal grid bins will coincide with number of used optical coefficients.
// If "1" then manual definition will be used.
DefineNumberOfGridBins=1

// Number of internal (between lower and upper limits) particle radii grid bins.
// Used for manual definition.
NumberOfInternalGridBins=8

// Type of the smoothing matrix.
// 0 - restricting the absolute values of weight factors.
// 1 - smoothness in the first derivative.
// 2 - smoothness in the second derivative.
// 3 - smoothness in the third derivative.
SmoothingMatrixOrder=2

// Integration step during generation of integral equation matrix, mum.
KernelStep=0.001
```

```
// Integration step during generation of optical data, mum.
OpticalStep=0.001

// Minimum value of lower limit of particle radii, mum.
RminMin=0.01
// Maximum value of lower limit of particle radii, mum.
RminMax=0.2
// Step of lower limit of particle radii, mum.
RminStep=0

// Minimum value of upper limit of particle radii, mum.
RmaxMin=0.1
// Maximum value of upper limit of particle radii, mum.
RmaxMax=8
// Step of upper limit of particle radii, mum.
RmaxStep=0

// Minimum value of real part of complex refractive index.
CRRealMin=1.325
// Maximum value of real part of complex refractive index.
CRRealMax=1.8
// Step of real part of complex refractive index.
CRRealStep=0.025

// Minimum value of imaginary part of complex refractive index.
CRImagMin=0
// Maximum value of imaginary part of complex refractive index.
CRImagMax=0.1
// Step of imaginary part of complex refractive index.
CRImagStep=0.003

:::: Regularization parameter ::::
//  $\gamma = 0$  at  $I = 0$ .
//  $\gamma = b \cdot a^I$  at  $I > 0$ .
// Minimum value of degree "I".
MinI=1

// Maximum value of degree "I".
MaxI=50

// Value of parameter "a".
ValueA=2

// Value of parameter "b".
ValueB=1e-28

// Border between fine and coarse modes, mum.
BorderOfFineMode=0.5
```

```
// Usage of optimized look-up table.  
// If "0" then Mie-scattering code will be used.  
// If "1" then optimized look-up table will be used.  
UseOptimizedDataBank=1  
  
// The file name of optimized look-up table.  
OptimizedDataBankName=69V_1.2-0.005-2_i0-i0.0005-i0.5  
  
// Number of CPUs to be used for the parallel data processing.  
NumOfProcessors=64  
  
// Number of solutions to be included into post-processing.  
SolutionsNumberPostProc=500  
  
// Post-processing stage. Allowed uncertainty level for the effective radius, %.  
ReffUncertaintyPostProc=25  
  
// Post-processing stage. Allowed uncertainty level for the number concentration, %.  
NumCUncertaintyPostProc=100  
  
// Post-processing stage. Allowed uncertainty level for the optical data, %.  
ODUncertaintyPostProc=10
```

## Appendix B. TiARA's Software Routines

### Tree diagram of the functions/procedures calls HSRL-2 optical data processing using the optimized look-up table

```
main (InversionServer\InversionServer.cpp)
  FindFile (CommonFunc\Functions.cpp)
  ReadHDF (InversionServer\ReadWrite.cpp)
  KillFile (CommonFunc\LogFunctions.cpp)
  GeneralInit (InversionServer\ServerThread.cpp)
    FindFile (CommonFunc\Functions.cpp)
    GetConfigurationFileString (CommonFunc\Functions.cpp)
    ReadOptimizedDataBankDimentions (CommonFunc\Functions.cpp)
    OptimizedRecalculateParameters (InversionServer\ReadWrite.cpp)
  ReadOptimizedDataBankDimentions (CommonFunc\Functions.cpp)
  ReadOptimizedDataBank (CommonFunc\Functions.cpp)
  ProvideDataSharedMemory (InversionServer\ServerThread.cpp)
  OutputInfoOnScreen (InversionServer\ReadWrite.cpp)
  LogFile (CommonFunc\LogFunctions.cpp)
  ThreadFuncInversion (InversionServer\ServerThread.cpp)
    CTextProgressCtrl/SetRange/SetText/SetPos/FinishIt (CommonFunc\TextProgressCtrl.cpp)
    InversionAlgorithmClassic (Classic\Main.cpp)
      CMatrix/~CMatrix/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      CVector/~CVector/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      MakeH (Classic\Main.cpp)
      MatrAClassic (Classic\Main.cpp)
        OptimizedIntgrlKsp (Classic\Main.cpp)
          Optimized_K_spCRIIntrpltn (Classic\Main.cpp)
      MT (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      ExtremeNoiseDistortion (CommonFunc\Functions.cpp)
      MUV (Classic\Main.cpp)
      DiscrepancyFunctionClassic (Classic\Main.cpp)
        GammaFunction (Classic\Main.cpp)
        LUdecomposition (Classic\Main.cpp)
      Concentration (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      MeanR (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      SurFaseConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      VolumeConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      M_1 (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      SSAbedo (Classic\Main.cpp)
        Optimized_K_spCRIIntrpltn (Classic\Main.cpp)
  PostProcessingNumOfBestSolutions (InversionServer\PostProcessing.cpp)
    MakeRangDiscrIndArr2 (CommonFunc\Functions.cpp)
    FillStructuresWithDataPostProc (InversionServer\PostProcessing.cpp)
    FlGivenNumConcUncertainty (InversionServer\PostProcessing.cpp)
    FlGivenODUncertainty (InversionServer\PostProcessing.cpp)
    FlGivenReffUncertainty (InversionServer\PostProcessing.cpp)
    FlGivenSzRng (InversionServer\PostProcessing.cpp)
    ArrayAverage (CommonFunc\Functions.cpp)
    ArrayDeviation (CommonFunc\Functions.cpp)
    WriteResutsHDF (InversionServer\ReadWrite.cpp)
      WriteHDFStructure (InversionServer\ReadWrite.cpp)
  WriteHDFLog (InversionServer\ReadWrite.cpp)
```

**Tree diagram of the functions/procedures calls  
HSRL-2 optical data processing using direct Mie-scattering calculations**

```
main (InversionServer\InversionServer.cpp)
  FindFile (CommonFunc\Functions.cpp)
  ReadHDF (InversionServer\ReadWrite.cpp)
  KillFile (CommonFunc\LogFunctions.cpp)
  GeneralInit (InversionServer\ServerThread.cpp)
    FindFile (CommonFunc\Functions.cpp)
    GetConfigurationFileString (CommonFunc\Functions.cpp)
  ProvideDataSharedMemory (InversionServer\ServerThread.cpp)
  OutputInfoOnScreen (InversionServer\ReadWrite.cpp)
  LogFile (CommonFunc\LogFunctions.cpp)
  ThreadFuncInversion (InversionServer\ServerThread.cpp)
    CTextProgressCtrl/SetRange/SetText/SetPos/FinishIt (CommonFunc\TextProgressCtrl.cpp)
    InversionAlgorithmClassic (Classic\Main.cpp)
      CMatrix/~CMatrix/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      CVector/~CVector/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      MakeH (Classic\Main.cpp)
      MatrAClassic (Classic\Main.cpp)
        IntegralKiBjClassic (Classic\Main.cpp)
          MieCoefficients (CommonFunc\Functions.cpp)
            BHMieEmbedded (CommonFunc\MieSca.cpp)
              B (CommonFunc\Functions.cpp)
        MT (Classic\Main.cpp)
        MUM (Classic\Main.cpp)
        ExtremeNoiseDistortion (CommonFunc\Functions.cpp)
        MUV (Classic\Main.cpp)
        DiscrepancyFunctionClassic (Classic\Main.cpp)
          GammaFunction (Classic\Main.cpp)
          LUdecomposition (Classic\Main.cpp)
        Concentration (CommonFunc\Functions.cpp)
          NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
        MeanR (CommonFunc\Functions.cpp)
          NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
        SurFaseConcent (CommonFunc\Functions.cpp)
          NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
        VolumeConcent (CommonFunc\Functions.cpp)
          NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
        M_1 (Classic\Main.cpp)
        MUM (Classic\Main.cpp)
        SSAbedo (Classic\Main.cpp)
          MieCoefficients (CommonFunc\Functions.cpp)
            BHMieEmbedded (CommonFunc\MieSca.cpp)
          ChangeASDTypeToNumber (CommonFunc\Functions.cpp)
            f (CommonFunc\Functions.cpp)
              B (CommonFunc\Functions.cpp)
    PostProcessingNumOfBestSolutions (InversionServer\PostProcessing.cpp)
      MakeRangDiscrIndArr2 (CommonFunc\Functions.cpp)
      FillStructuresWithDataPostProc (InversionServer\PostProcessing.cpp)
      FIGivenNumConcUncertainty (InversionServer\PostProcessing.cpp)
      FIGivenODUncertainty (InversionServer\PostProcessing.cpp)
      FIGivenReffUncertainty (InversionServer\PostProcessing.cpp)
      FIGivenSzRng (InversionServer\PostProcessing.cpp)
      ArrayAverage (CommonFunc\Functions.cpp)
      ArrayDeviation (CommonFunc\Functions.cpp)
      WriteResutsHDF (InversionServer\ReadWrite.cpp)
        WriteHDFStructure (InversionServer\ReadWrite.cpp)
  WriteHDFLog (InversionServer\ReadWrite.cpp)
```

**Tree diagram of the functions/procedures calls  
Simulated optical data processing using the optimized look-up table**

```
main (InversionServer\InversionServer.cpp)
  FindFile (CommonFunc\Functions.cpp)
  KillFile (CommonFunc\LogFunctions.cpp)
  GeneralInit (InversionServer\ServerThread.cpp)
    FindFile (CommonFunc\Functions.cpp)
    GetConfigurationFileString (CommonFunc\Functions.cpp)
    ReadOptimizedDataBankDimentions (CommonFunc\Functions.cpp)
    OptimizedRecalculateParameters (InversionServer\ReadWrite.cpp)
    MakeLNDistrRminRmax (CommonFunc\Functions.cpp)
      Gaus (CommonFunc\Functions.cpp)
        LnSigma (CommonFunc\Functions.cpp)
        MeanRadius (CommonFunc\Functions.cpp)
    g (CommonFunc\Functions.cpp)
      MieCoefficients (CommonFunc\Functions.cpp)
        BHMieEmbedded (CommonFunc\MieSca.cpp)
      Gaus1 (CommonFunc\Functions.cpp)
        LnSigma (CommonFunc\Functions.cpp)
        MeanRadius (CommonFunc\Functions.cpp)
    VolumeConcentAn (CommonFunc\Functions.cpp)
    SurfaceConcentAn (CommonFunc\Functions.cpp)
  ReadOptimizedDataBankDimentions (CommonFunc\Functions.cpp)
  ReadOptimizedDataBank (CommonFunc\Functions.cpp)
  ProvideDataSharedMemory (InversionServer\ServerThread.cpp)
  OutputInfoOnScreen (InversionServer\ReadWrite.cpp)
  LogFile (CommonFunc\LogFunctions.cpp)
  ThreadFuncInversion (InversionServer\ServerThread.cpp)
    CTextProgressCtrl/SetRange/SetText/SetPos/FinishIt (CommonFunc\TextProgressCtrl.cpp)
    InversionAlgorithmClassic (Classic\Main.cpp)
      CMatrix/~CMatrix/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      CVector/~CVector/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      MakeH (Classic\Main.cpp)
      MatrAClassic (Classic\Main.cpp)
        OptimizedIntgrlKsp (Classic\Main.cpp)
          Optimized_K_spCRIIntrpltn (Classic\Main.cpp)
      MT (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      ExtremeNoiseDistortion (CommonFunc\Functions.cpp)
      MUV (Classic\Main.cpp)
      DiscrepancyFunctionClassic (Classic\Main.cpp)
        GammaFunction (Classic\Main.cpp)
        LUdecomposition (Classic\Main.cpp)
      Concentration (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      MeanR (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      SurFaseConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      VolumeConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      M_1 (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      SSAlbedo (Classic\Main.cpp)
        Optimized_K_spCRIIntrpltn (Classic\Main.cpp)
  MakeRangDiscrIndArr2 (CommonFunc\Functions.cpp)
  SaveResultsToSSFFile (InversionServer\ReadWrite.cpp)
    FormatDEMOSStyle (InversionServer\ReadWrite.cpp)
    SSFWrite (InversionServer\ReadWrite.cpp)
```

**Tree diagram of the functions/procedures calls  
Simulated optical data processing using direct Mie-scattering calculations**

```
main (InversionServer\InversionServer.cpp)
  FindFile (CommonFunc\Functions.cpp)
  KillFile (CommonFunc\LogFunctions.cpp)
  GeneralInit (InversionServer\ServerThread.cpp)
    FindFile (CommonFunc\Functions.cpp)
    GetConfigurationFileString (CommonFunc\Functions.cpp)
    MakeLNDistrRminRmax (CommonFunc\Functions.cpp)
      Gaus (CommonFunc\Functions.cpp)
        LnSigma (CommonFunc\Functions.cpp)
        MeanRadius (CommonFunc\Functions.cpp)
    g (CommonFunc\Functions.cpp)
      MieCoefficients (CommonFunc\Functions.cpp)
        BHMieEmbedded (CommonFunc\MieSca.cpp)
      GausI (CommonFunc\Functions.cpp)
        LnSigma (CommonFunc\Functions.cpp)
        MeanRadius (CommonFunc\Functions.cpp)
    VolumeConcentAn (CommonFunc\Functions.cpp)
    SurfaceConcentAn (CommonFunc\Functions.cpp)
  ProvideDataSharedMemory (InversionServer\ServerThread.cpp)
  OutputInfoOnScreen (InversionServer\ReadWrite.cpp)
  LogFile (CommonFunc\LogFunctions.cpp)
  ThreadFuncInversion (InversionServer\ServerThread.cpp)
    CTextProgressCtrl/SetRange/SetText/SetPos/FinishIt (CommonFunc\TextProgressCtrl.cpp)
    InversionAlgorithmClassic (Classic\Main.cpp)
      CMatrix/~CMatrix/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      CVector/~CVector/SetSize/ZeroData (CommonFunc\Matrix.cpp)
      MakeH (Classic\Main.cpp)
      MatrAClassic (Classic\Main.cpp)
        IntegralKiBjClassic (Classic\Main.cpp)
          MieCoefficients (CommonFunc\Functions.cpp)
            BHMieEmbedded (CommonFunc\MieSca.cpp)
          B (CommonFunc\Functions.cpp)
      MT (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      ExtremeNoiseDistortion (CommonFunc\Functions.cpp)
      MUV (Classic\Main.cpp)
      DiscrepancyFunctionClassic (Classic\Main.cpp)
        GammaFunction (Classic\Main.cpp)
        LUdecomposition (Classic\Main.cpp)
      Concentration (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      MeanR (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      SurFaseConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      VolumeConcent (CommonFunc\Functions.cpp)
        NumberBaseFunction/SFAreaBaseFunction/VolumeBaseFunction (...)
      M_1 (Classic\Main.cpp)
      MUM (Classic\Main.cpp)
      SSAbedo (Classic\Main.cpp)
        MieCoefficients (CommonFunc\Functions.cpp)
          BHMieEmbedded (CommonFunc\MieSca.cpp)
        ChangeASDTypeToNumber (CommonFunc\Functions.cpp)
          f (CommonFunc\Functions.cpp)
            B (CommonFunc\Functions.cpp)
  MakeRangDiscrIndArr2 (CommonFunc\Functions.cpp)
  SaveResultsToSSFFile (InversionServer\ReadWrite.cpp)
    FormatDEMOSyle (InversionServer\ReadWrite.cpp)
    SSFWrite (InversionServer\ReadWrite.cpp)
```

## Acronyms and Abbreviations

$3\beta+2\alpha$  – Configuration of a lidar instrument that provides three backscatter ( $\beta$ ) plus two extinction ( $\alpha$ ) coefficients as a result of measurements.

AERONET – Aerosol Robotic Network.

ASCII – American Standard Code for Information Interchange.

ATBD – Algorithm Theoretical Basis Document.

cm – centimeter.

CRI – Complex Refractive Index.

EEM – Extreme error computation model.

GEM – Gauss-distributed error computation model.

HDF – Hierarchical Data Format.

HSRL – High-Spectral Resolution Lidar.

LaRC – Langley Research Center.

Lidar – Light identification, detection, and ranging.

LUT – Optimized look-up table.

m – meter.

Mm – megameter.

μm, μm – micrometer.

NASA – National Aeronautics and Space Administration.

nm – nanometer.

n.u. – no units.

OS – Operating System.

PSD – Particle Size Distribution.

sr – steradian.

SSA – Single-Scattering Albedo.

TiARA – Tikhonov Advanced Regularization Algorithm.

## References

1. Atkinson, K. A., ed., *An Introduction to Numerical Analysis* (Wiley, 1989).
2. Bohren, C. F., and D. R. Huffman, eds., *Absorption and Scattering of Light by Small Particles* (Wiley, 1983).
3. Bohren and Huffman Mie-code: [code.google.com/archive/p/scatterlib/wikis/Spheres.wiki](http://code.google.com/archive/p/scatterlib/wikis/Spheres.wiki)
4. Burton, S. P., E. Chemyakin, X. Liu, K. Knobelspiesse, S. Stammes, P. Sawamura, R. H. Moore, C. A. Hostetler, and R. A. Ferrare, 2016: “Information content and sensitivity of the  $3\beta+2\alpha$  lidar measurement system for aerosol microphysical retrievals”, *Atmos. Meas. Tech.*, **9**, 5555–5574.
5. Chemyakin, E., D. Müller, S. P. Burton, A. Kolgotin, C. A. Hostetler, and R. A. Ferrare, 2014: “Arrange and average algorithm for the retrieval of aerosol parameters from multiwavelength high-spectral-resolution lidar/Raman lidar data”, *Appl. Opt.*, **53**, 7252–7266.
6. Chemyakin, E., S. P. Burton, A. Kolgotin, D. Müller, C. A. Hostetler, and R. A. Ferrare, 2016: “Retrieval of aerosol parameters from multiwavelength lidar: investigation of the underlying inverse mathematical problem”, *Appl. Opt.*, **55**, 2188–2202.
7. Courant, R., ed., *Vorlesungen über Differential- und Integralrechnung* (Springer, 1971).
8. Dubovik, O., B. Holben, T. F. Eck, A. Smirnov, Y. J. Kaufman, M. D. King, D. Tanré, and I. Slutsker, 2002: “Variability of absorption and optical properties of key aerosol types observed in worldwide locations”, *J. Atmos. Sci.*, **59**, 590–608.
9. Golub, G. H., Heath, M., and Wahba, G., 1979: “Generalized cross validation as a method for choosing a good ridge parameter”, *Technometrics*, **21**, 215–223.
10. Hinds, W. C., ed., *Aerosol Technology. Properties, Behavior and Measurement of Airborne Particles* (Wiley, 1999).
11. Kolgotin, A., and D. Müller, 2008: “Theory of inversion with two-dimensional regularization: profiles of microphysical particle properties derived from multiwavelength lidar measurements”, *Appl. Opt.*, **47**, 4472–4490.
12. Kolgotin, A., D. Müller, E. Chemyakin, and A. Romanov, 2016: “Improved identification of the solution space of aerosol microphysical properties derived from the inversion of profiles of lidar optical data, part 1: theory”, *Appl. Opt.*, **55**, 9839–9849.
13. Kolgotin, A., D. Müller, E. Chemyakin, and A. Romanov, 2016: “Improved identification of the solution space of aerosol microphysical properties derived from the inversion of profiles of lidar optical data, part 2: simulations with synthetic data”, *Appl. Opt.*, **55**, 9850–9865.
14. Müller, D., U. Wandinger, D. Althausen, I. Mattis, and A. Ansmann, 1998: “Retrieval of physical particle properties from lidar observations of extinction and backscatter at multiple wavelengths”, *Appl. Opt.*, **37**, 2260–2263.
15. Müller, D., U. Wandinger, and A. Ansmann, 1999: “Microphysical particle parameters from extinction and backscatter lidar data by inversion with regularization: theory”, *Appl. Opt.*, **38**, 2346–2357.

16. Müller, D., U. Wandinger, and A. Ansmann, 1999: “Microphysical particle parameters from extinction and backscatter lidar data by inversion with regularization: simulation”, *Appl. Opt.*, **38**, 2358–2368.
17. Müller, D., C. A. Hostetler, R. A. Ferrare, S. P. Burton, E. Chemyakin, A. Kolgotin, J. W. Hair, A. L. Cook, D. B. Harper, R. R. Rogers, R. W. Hare, C. S. Cleckner, M. D. Obland, J. Tomlinson, L. K. Berg, and B. Schmid, 2014: “Airborne Multiwavelength High Spectral Resolution Lidar (HSRL-2) observations during TCAP 2012: vertical profiles of optical and microphysical properties of a smoke/urban haze plume over the northeastern coast of the US”, *Atmos. Meas. Tech.*, **7**, 3487–3496.
18. Müller, D., C. Böckmann, A. Kolgotin, L. Schneidenbach, E. Chemyakin, J. Rosemann, P. Znak, and A. Romanov, 2016: “Microphysical particle properties derived from inversion algorithms developed in the framework of EARLINET”, *Atmos. Meas. Tech.*, **9**, 5007–5035.
19. Müller, D., A. Kolgotin, E. Chemyakin, S. P. Burton, C. A. Hostetler, R. A. Ferrare, A. Romanov, and M. Tesche, 2018: “Automated, unsupervised inversion of multiwavelength lidar data with TiARA: microphysical parameters and uncertainties derived from simulation studies”, *Appl. Opt.*, to be submitted.
20. Nychka, D., Wahba, G., Goldfarb, S., and Pugh, T., 1984: “Cross validated spline methods for the estimation of three-dimensional tumor size distributions from observations on two-dimensional cross sections”, *J. Am. Stat. Soc.*, **79**, 832–846.
21. Qing, P., H. Nakane, Y. Sasano, and S. Kitamura, 1989: “Numerical simulation of the retrieval of aerosol size distribution from multiwavelength laser radar measurements”, *Appl. Opt.*, **28**, 5259–5265.
22. Rodgers, C. D., *Inverse Methods for Atmospheric Sounding: Theory and Practice*. Series on Atmospheric, Oceanic and Planetary Physics, Vol. 2, edited by Taylor, F. W. (World Scientific, New Jersey, 2000).
23. Sawamura, P., D. Müller, R. M. Hoff, C. A. Hostetler, R. A. Ferrare, J. W. Hair, R. R. Rogers, B. E. Anderson, L. D. Ziemba, A. J. Beyersdorf, K. L. Thornhill, E. L. Winstead, and B. N. Holben, 2014: “Aerosol optical and microphysical retrievals from a hybrid multiwavelength lidar dataset – DISCOVER-AQ 2011”, *Atmos. Meas. Tech.*, **7**, 3095–3112.
24. Seinfeld, J. H., and S. N. Pandis, eds., *Atmospheric Chemistry and Physics. From Air Pollution to Climate Change* (Wiley, 2006).
25. Tikhonov, A. N., and V. Y. Arsenin, eds., *Solution of Ill-Posed Problems* (Wiley, 1977).
26. Twomey, S., ed., *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements* (Elsevier, 1977).
27. Veselovskii, I., A. Kolgotin, V. Griaznov, D. Müller, U. Wandinger, and D. N. Whiteman, 2002: “Inversion with regularization for the retrieval of tropospheric aerosol parameters from multiwavelength lidar sounding”, *Appl. Opt.*, **41**, 3685–3699.
28. Veselovskii, I., A. Kolgotin, V. Griaznov, D. Müller, K. Franke, and D. N. Whiteman, 2004: “Inversion of multiwavelength Raman lidar data for retrieval of bimodal aerosol size distribution”, *Appl. Opt.*, **43**, 1180–1195.

29. Veselovskii, I., Kolgotin, A., Müller, D., and Whiteman, D. N., 2005: “Information content of multiwavelength lidar data with respect to microphysical particle properties derived from eigenvalue analysis”, *Appl. Opt.*, **44**, 5292–5303.
30. Wolfenbarger, J. K., and Seinfeld, J. H., 1990: “Inversion of aerosol size distribution data”, *J. Aerosol Sci.*, **21**, 227–247.
31. Wolfenbarger, J. K., and Seinfeld, J. H., 1991: “Regularized solutions to the aerosol data inversion problem”, *SIAM J. Sci. Stat. Comput.*, **12**, 342–361.
32. Whitby, K. T., 1978: “The physical characteristics of sulfur aerosols”, *Atmos. Environ.*, **12**, 135–159.
33. Zuev, V. E., and I. E. Naats, eds., *Inverse Problems of Lidar Sensing of the Atmosphere* (Springer-Verlag, 1983).