SPARTA - Solver for Polarized Atmospheric Radiative Transfer Applications: Manual

THIS IS SPARTA

Corresponds to:

Vasileios Barlakas
Leibniz–Institute for Tropospheric Research, Permoserstr. 15, D–04318 Leipzig

email:
barlakas@tropos.de

Date: 10.12.2016

Contents

1 Introduction **1**
1.1 What is SPARTA . . . 1
1.2 Copyright and Disclaimer . . . 2
1.3 Features Overview . . . 2

2 Installation **4**
2.1 Compatibility . . . 4
2.2 Unpacking . . . 4
2.3 Directory Tree . . . 4
2.3.1 example/ . . . 4
2.3.2 doc/ . . . 5
2.3.3 solvers/ . . . 5
2.3.4 utils/ . . . 5
2.4 How to Compile and Run SPARTA . . . 6

3 Model Description **8**
3.1 SPARTA Overview . . . 8
3.2 Radiative Transfer Solvers . . . 10
3.3 Methods and Techniques . . . 10
3.4 Surface Treatment . . . 10
3.5 SPARTA Model Program Flow . . . 11
3.6 SPARTA Efficiency . . . 12

4 Input and Output Files **13**
4.1 Description of Files . . . 13
4.1.1 Description of the Input Files . . . 13
4.1.2 Description of the Output Files . . . 17

5 How to Create Input Files **20**
5.1 Description of make.f90 . . . 20
5.2 Description of mainfile.f90 . . . 21

Bibliography **22**

List of Symbols **24**

List of Abbreviations **28**

List of Figures **29**

List of Tables **30**

1 Introduction

1.1 What is SPARTA

The Solver for Polarized Atmospheric Radiative Transfer Applications (SPARTA) is a new three–dimensional (3D) vector radiative transfer model. The model employs the forward Monte Carlo technique for efficient column–response, pixel–based radiance calculations including polarization for 3D inhomogeneous cloudless and cloudy atmospheres. SPARTA is based on the scalar Monte Carlo model of the Leibniz Institute of Marine Sciences (now GEOMAR) at the UNIversity of Kiel (MC–UNIK, Macke et al. 1999), which participated in the international Intercomparison of 3D Radiation Codes (I3RC, Cahalan et al. 2005). MC–Unik has been extended to account for polarization effects due to multiple–scattering by non-spherical particles, i.e., coarse mode dust or ice particles. Note that SPARTA is a pure radiative transfer solver. There are no utilities for calculating monochromatic or spectral band gas absorption, phase matrices, or single scattering albedos. The model output is used to parameterize broadband solar upwelling and downwelling radiative fluxes (in both sclalar and vector modes) in terms of bulk cloud properties; thermal emission is not included. Although several one–dimensional (1D) and 3D vector models have been developed, the availability is usually restricted to the 1D codes (with only a few exceptions). Therefore, SPARTA has been developed in order to fill this gap. SPARTA is an open source, user–friendly model. It will help to compare and improve models (world–wide model assembly, model intercomparison).

SPARTA has been tested for different atmospheric conditions (e.g., Rayleigh scattering, aerosol particles and randomly oriented prolate spheroids) and comprehensively compared to benchmark results (de Haan et al., 1987; Natraj et al., 2009; Kokhanovsky et al., 2010). It took part in the model intercomparison project launched by the polarization working group of the International Radiation Commission (IPRT) (Emde et al., 2015). Overall, the comparisons demonstrated that the performance of SPARTA is excellent (Barlakas et al., 2014; Emde et al., 2015; Barlakas et al., 2016a; Barlakas, 2016b). All deviations found in case of highly asymmetric phase matrices can be explained by the noise of the Monte Carlo technique in radiance calculations. The noisiness of the signal is a function of the number of photons. Increasing the selected number of photons diminishes the noise. Note that the second phase of IPRT is on the way.

This report documents SPARTA and describes how the model can be used and run in different operating systems. For the basic concepts and definitions, the details of the implementation of polarization in MC–UNIK, the differences between MC–UNIK and SPARTA, as well as the validation report of the model, the reader is referred to Barlakas et al. (2014); Emde et al. (2015); Barlakas et al. (2016a); Barlakas (2016b).

1.2 Copyright and Disclaimer

SPARTA is an open source. The model is available for download at `http://tools.tropos.de` or by e–mail request. The author itends to provide continuous support. However, the author is not responsible for issues caused by any misuse of the code. Any publications involving the use of SPARTA must mention the name of the model and cite the corresponding paper (Barlakas et al., 2016a). Finally, in case you are willing to use SPARTA, you may want to notify the author.

1.3 Features Overview

Pros of SPARTA: Open source, user–friendly solver

Basic algorithm: Forward Monte Carlo technique

Radiative transfer solvers:

- Three–dimensional (3D) radiative transfer
- Two–dimensional (2D) radiative transfer
- One–dimensional (1D) radiative transfer
- Independent Column Approximation (ICA)

Programming language:

- Fortran90
- not modular, seperate code

Required libraries: None

Minimum set of physical quantities (input): Three–dimensionally inhomogeneous atmosphere

- Volume extinction or scattering coefficient
- Single scattering albedo
- Particle effective radius
- Phase matrix
- Full cartesian grid (no gaps)

Requirements:

- Linux/UNIX–based operating system
- Fortran90 compiler

Surface treatment:

- Black surface (non–scattering)

- Lambertian surface (isotropic reflection)
- Ocean reflection (anisotropic and polarizing reflection including shadow effects)

Numerical techniques:

- Local Estimate Method (LEM)
- Importance sampling method

Output quantities: Upward and downward radiative fluxes

Are parameters given by namelist:

- No namelist
- ASCII and binary files

2 Installation

2.1 Compatibility

SPARTA runs in Linux/UNIX–based operating systems. The model is written in FORTRAN 90 and conforms to the freely availble gfortran compiler and the commericial Intel Fortran compiler, i.e., GNU Compiler Collection (GCC) 4.8.1 (other compilers may work; not tested).

2.2 Unpacking

The model is available for download at `http://tools.tropos.de` or by e–mail request. SPARTA is packed in a zip archive, SPARTA.zip. In order to unzip it on a Linux/UNIX–based operating system, one should first move the zip file (current directory) to a desired working directory (working directory):

$ mv -v /path/current_directory/SPARTA.zip /path/working_directory/

then, change directory (go to the working_directory):

$ cd /path/working_directory/

and finally type:

$ unzip SPARTA.zip

Consequently, a directory named SPARTA is found with the following file structure:

$ cd SPARTA # enter first the directory

- example/
- doc/
- solvers/
- utils/

2.3 Directory Tree

2.3.1 example/

- example/
  SPARTA.f90

input/
output/

This directory contains the fortran90 source code of SPARTA (SPARTA.f90), as well as the example input (in input/) and the corresponding output files (in output/). This example corresponds to one of the test cases of the model intercomparison project form phase A (IPRT–A). The explanation of all the input and output files is given in Chapter 4.

2.3.2 doc/

It contains this manual as well as the two major related citations (Barlakas et al., 2016a; Barlakas, 2016b).

2.3.3 solvers/

- solvers/
  3D_mode/
    lambert/
    ocean/
  IPA_mode/
    lambert/
    ocean/

In this directory, two radiative transfer solvers are found, namely SPARTA in full 3D mode and IPA mode (see Chapter 4). In addition, two seperate versions of SPARTA are found, for the different surface treatment, namely isotropic (lambert/) and anisotropic (ocean/).

2.3.4 utils/

- utils/
  make.f90
  mainfile.f90
  info/
  input/

The fourth directory contains two example–programs for creating the main input and the input files, namely mainfile.f90 and make.f90, respectively. The latter program read the information contained in info/ and stores the input files in the desired format in input/ of the aforementioned example (example/). For more information, see Chapter 5.

2.4 How to Compile and Run SPARTA

- Create a test directory (test/) by copying and renaming the directory example/
  $ cp -r example test

- Enter the directory test/
  $ cd test

- Delete the example output files (1.*)
  $ rm -r output/1.*

- Compile SPARTA. SPARTA has been comprehensively tested; If the user employs the right compiler, there should not be any error message.
  $ gfortran -o SPARTA SPARTA.f90
  where -o <name> specifies the name for the output object (executable).

- Run SPARTA by typing
  $ ./SPARTA
  then the following message appear

  ---

  provide the name of the main input file =

- One should provide the main input file. For the current example, please type 1 and then press the enter key. Some information is then illustrated in the terminal, namely,

  ---

  SPARTA

- When the radiative transfer simulations are about to finish, one should see the number of scattering (N_scat) and reflecting (N_refl) events (on the terminal); the user might want to comment out this lines (it has been used for statistics reasons).
- Finally, if the model run succesfully, the following message will appear at the screen:

  SPARTA simulations terminated successfully

- All the output data (for the current example, 1.*) are stored in the directory output/.
- In case the user will modify the source code, it is suggested to debug the code. Preferably, one shoud use the following debugging commands (only once):

$ gfortran -g -fbounds-check -fcheck-array-temporaries -o SPARTA SPARTA.f90

where:
-g : generates debug information for use by a symbolic debugger.

-fbounds-check : enable generation of run-time checks for array subscripts and against the declared minimum and maximum values. It also checks array indices for assumed and deferred shape arrays against the actual allocated bounds and ensures that all string lengths are equal for character array constructors without an explicit typespec.

-fcheck-array-temporaries: Warns at run time when for passing an actual argument a temporary array had to be generated. The information generated by this warning is sometimes useful in optimization, in order to avoid such temporaries.

3 Model Description

A comprehensive description of SPARTA is found in Barlakas (see Chapter 3, 2016b). In that chapter, the basic concepts and definitions of SPARTA are introduced. The details of the implementation of polarization are also presented; the differences between MC–UNIK and SPARTA are outlined. Parts of that chapter, have been also published in Barlakas et al. (2014) and Barlakas et al. (2016a).

3.1 SPARTA Overview

SPARTA considers a 3D Cartesian domain, which is split into grid–boxes, defined by geometrical dimensions along x-, y-, and z-directions. A total volumetric extinction $\beta_{\text{ext}}^{\text{tot}}$ and scattering coefficient $\beta_{\text{sca}}^{\text{tot}}$, a total normalized scattering phase matrix $\tilde{\mathbf{P}}^{\text{tot}}(\vartheta)$ with the scattering angle ϑ, and a total single–scattering albedo $\tilde{\omega}^{\text{tot}}$ are allocated to each grid–box. The superscript "tot" indicates the combined optical properties of the different types of particles (e.g., molecules, aerosol, and cloud particles), which are contained in each grid–box. For example, in case of a grid–box consisting of molecules and dust particles, the total scattering phase matrix $\tilde{\mathbf{P}}^{\text{tot}}(\vartheta)$ at each grid box is derived by averaging the Rayleigh scattering phase matrix, $\tilde{\mathbf{P}}_{\text{r}}(\vartheta)$, and the scattering phase matrix of dust particles, $\tilde{\mathbf{P}}_{\text{d}}(\vartheta)$, weighted over their characteristic volumetric scattering coefficients ($\beta_{\text{sca}}^{\text{r}}$ and $\beta_{\text{sca}}^{\text{d}}$ for the individual grid box):

$$\tilde{\mathbf{P}}^{\text{tot}}(\vartheta) = \frac{\beta_{\text{sca}}^{\text{r}} \cdot \tilde{\mathbf{P}}_{\text{r}}(\vartheta) + \beta_{\text{sca}}^{\text{d}} \cdot \tilde{\mathbf{P}}_{\text{d}}(\vartheta)}{\beta_{\text{sca}}^{\text{r}} + \beta_{\text{sca}}^{\text{d}}}, \tag{3.1}$$

Likewise, the total single–scattering albedo (for this grid–box) is given by the sum of the Rayleigh scattering coefficient and the scattering coefficient of the aerosol particles divided by the total volumetric extinction coefficient.

A representation of the model domain is illustrated in Figure 3.1. Directions are specified by means of the angles θ and ϕ (Hovenier et al., 2004; Mishchenko et al., 2002). The azimuth angle ϕ is counted anti–clockwise when looking upwards (positive z-direction) and the zenith angle θ is the angle with respect to the upward normal direction (measured from the positive z-direction). As a consequence, the viewing zenith angle θ_{det} is measured from the upward normal, meaning it ranges between 0° and 90° for the reflected, and between 90° and 180° for the transmitted electromagnetic (EM) radiation.

Free path lengths are simulated as outlined by Marchuk et al. (1980) by random number processes with attenuation described by the law of Bouguer–Beer. Scattering directions are

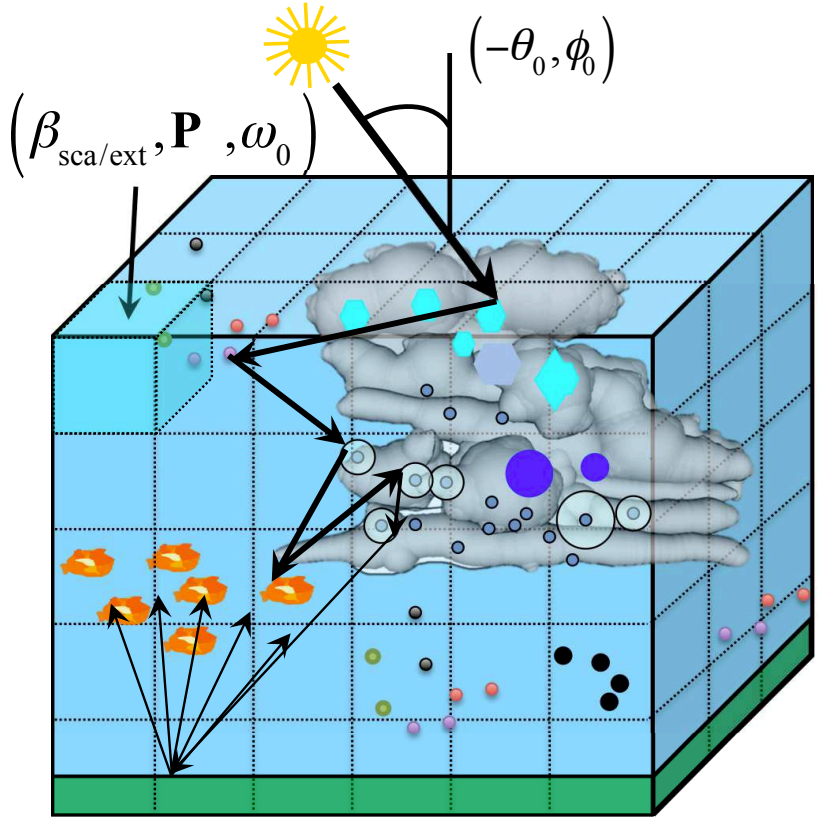

Figure 3.1: A representation of the scheme of the photon path within the 3D domain of the Monte Carlo radiative transfer model: $(-\mu_0, \phi_0)$ is the initial incident photon direction (downward solar direction) defined by the cosine of the solar zenith ($\mu_0 = \cos\theta_0$) and azimuth angles (ϕ_0), $\beta^{\text{tot}}_{\text{sca/ext}}$ stands for either the total volumetric scattering (subscript "sca") or extinction coefficient (subscript "ext"), $\tilde{\mathbf{P}}^{\text{tot}}$ for the total normalized scattering phase matrix, and $\tilde{\omega}^{\text{tot}}$ is the total single–scattering albedo. The superscript "tot" indicates the combined optical properties of the different types of particles which are contained in each grid–box.

calculated according to an importance sampling method, which seems to be the fastest approach of the different methods to the polarized radiative transfer problem (Collins et al., 1972; Marchuk et al., 1980; Emde et al., 2010). An alternative approach for polarization problems, which has been investigated during the development of SPARTA, is to employ the normalized scattered Stokes vector as the probability density function (PDF) to sample the new direction. However, this leads to a bi–variable PDF of the scattering zenith and azimuth angles introducing numerical problems (due to the trigonometric functions) and consequently to an increase in computational needs. Absorption is taken into account by decreasing the initial Stokes weight by the estimated total absorption coefficient, along the photon path, with the Bouguer–Beer Law. The surface contribution is calculated assuming isotropic reflection (Lambertian surface) or anisotropic ocean reflection as outlined in Mishchenko and Travis (1997). In order to obtain precise radiance calculations for each wavelength and to diminish the noise of simulations for highly asymmetric phase matrices the Local Estimate Method has been applied (Collins et al., 1972; Marchuk et al., 1980; Marshak and Davis, 2005). Other variance reduction methods have not been implemented yet. Future work will include ap-

plying various variance reduction techniques (truncation techniques, Rozanov and Lyapustin 2010; Buras and Mayer 2011).

3.2 Radiative Transfer Solvers

- **Three–dimensional (3D) radiative transfer.** A 3D solver is found in solvers/3D_mode/.
- **Two–dimensional (2D) radiative transfer.** In this mode, the variability in only one horizontal direction (x or y) is considered.
- **One–dimensional (1D) radiative transfer.** It refers to 1D calculations according to the plane–parallel assumption utilizing domain–averaged optical properties.
- **Independent Column/Pixel Approximation (ICA/IPA).** In this scheme, 1D radiative transfer simulations are independently conducted pixel by pixel (column by column) by simply fixing the x- and y- geometrical components of the photons to the initial values preventing the horizontal photon transport between the pixels; the horizontal inhomogeneity is considered. A IPA solver is found in solvers/IPA_mode/.

3.3 Methods and Techniques

As already mentioned, two methods have beem implemented in SPARTA (for better performance). A detailed description is found in Barlakas (see Sections 3.2.4 and 3.2.6, Chapter 3, 2016b).

- Local Estimate Method (LEM).
- Importance sampling method.

3.4 Surface Treatment

- Black surface (non–scattering). It fully absorbs the incident radiation. The surface albedo is set to zero ($a_{\mathrm{L}} = 0$).
- Lambertian surface (isotropic reflection). Lambertian reflection represents a surface, which reflects isotropically and completely depolarizes the incident radiation. The angular distribution of the reflected radiation is uniform and independent of the incident direction and state of polarization. The surface albedo values range: $0 < a_{\mathrm{L}} \leq 1$.
- Ocean reflection (including shadow effects). To consider anisotropic and polarizing ocean reflection, the code by Mishchenko and Travis (1997) has been implemented into SPARTA. It calculates the reflection matrix for rough water surfaces ($\mathbf{IR}_{\mathrm{ocean}}$) utilizing the Fresnel formulas.

More information is found in Barlakas (see Sections 3.2.5, Chapter 3, 2016b)

3.5 SPARTA Model Program Flow

SPARTA employs the statistical Monte Carlo technique (in the forward scheme). In Monte Carlo theory, photons refer to imaginary discrete "packets" or "units" of the incident energy rather than the physical photons (quantum electrodynamics) (Mishchenko, 2009).

This is just a brief introduction of SPARTA program flow. For an extended description, one should go through the source code of SPARTA and Barlakas (see Chapter 3, 2016b). In brief, the SPARTA program goes as follows:

- Initialization. Initialize general variables by reading the main input file (see subroutine INPUT).
- Construct the Look–up tables (see subroutine INIT).
- Photon initiation. Simulations begin with a photon entering randomly at the top layer of the model domain. To each photon a scalar and a vector weight (Stokes weight) to each photon is given; the direction is specified by the solar position (see subroutine init_photon).
- Photon tracing and absorption. Photons are traced throughout their propagation in a scattering and absorbing medium (atmosphere) until a scattering event takes place (see subroutine PhotonPath). In case a photon reaches the edge of the domain, periodic boundary conditions are applied, meaning the photon re–enters the domain from the exact opposite side preserving its directional vector and conserving energy.
- Anisotropic scattering. The importance sampling method is used to sampe the new direction of the photon. Consequently, the scalar and Stokes weights are updated (see subroutine polarization).
- Multiple–scattering. Photons are traced from scattering to scattering throught the atmosphere.
- Surface reflection. When the photons hit the surface, the surface contribution is considered. Consequently, the scalar and Stokes weights are updated (see subroutine reflection).
- Count the photon. Whenever a scattering or reflection event takes place, the Local Estimate method is applied (see subroutine PhotonPathDetector). It accounts for the probability that the photon is scattered into the direction of the sensor at each scattering process.
- Photon termination. The photon is terminated when it is absorbed (reaches a specific minimum) or exits the scattering media. Thereafter, a new photon is created.
- End of simulations. When the last photon is traced, the requested output will be stored in files (see subroutine OUTPUT).

Table 3.1: Computation times for 10 individual simulations. Rayleigh scattering at $\lambda = 325\,\text{nm}$ wavelength, Rayleigh scattering at $\lambda = 350\,\text{nm}$, and mixed atmosphere (Rayleigh + spheroids) at $\lambda = 350\,\text{nm}$, for one viewing direction. The errors correspond to standard deviation.

Settings	**Rayleigh (325 nm)**	**Rayleigh (350 nm)**	**Mixed (350 nm)**
10^4 photons	0.104 s (0.224 %)	0.098 s (0.209 %)	1.393 s (0.821 %)
10^5 photons	0.618 s (0.110 %)	0.550 s (0.106 %)	1.897 s (0.488 %)

3.6 SPARTA Efficiency

SPARTA is written in Fortran 90 and its performance has been tested on an Intel Core i7 processor with 2.6 GHz. The computation time (Central Processing Unit, CPU) is mostly dependent on the grid resolution of the model domain, the solar and viewing directions, the phase matrix, and more importantly, on the optical thickness. Three realistic atmospheric scenarios were considered. Simulations were performed for 1D standard atmospheres (Anderson et al., 1986) at 325 nm and 350 nm wavelengths above a black non–scattering surface. The model domain consists of 30 equidistant layers and is lifted to 30 km height. The Rayleigh optical thickness is approximately 0.96 at 325 nm wavelength and 0.62 at 350 nm wavelength. In case of aerosol scattering, prolate spheroidal particles with an optical thickness of 0.2 were added (Gasteiger et al., 2011). The corresponding CPU times for the diffuse upward radiation at TOA are listed in Table 3.1.

4 Input and Output Files

- example/
  SPARTA.f90
  input/
  output/

4.1 Description of Files

SPARTA.f90: Monte Carlo radiative transfer code with local estimate scheme for efficient radiance calculations including polarization. SPARTA is a pure radiative transfer solver. No utilities for monochromatic or spectral band gas absorption, phase matrices, and single scattering albedos calculations are included.

4.1.1 Description of the Input Files

- example/input/
  1
  angle
  cloud
  cloud.p11
  cloud.p12
  cloud.p22
  cloud.p33
  cloud.p34
  cloud.p44
  cube_files
  cube_files.abs
  cube_files.ext
  cube_files.sca
  cube_files.w0
  cube_files.pm-index
  x_grid
  y_grid
  z_grid

cube_files: Contains the name list of the cloud files. The latter files contain the cloud information for each grid box.

- **cube_files.ext:** 3D field of the total volume extinction coefficient ($\beta_{\text{ext}}^{\text{tot}}$) in m^{-1} or km^{-1} (depending on the units of the grid).
- **cube_files.w0:** 3D field of the total single–scattering albedo ($\tilde{\omega}^{\text{tot}}$).
- **cube_files.sca:** 3D field of volume scattering coefficient ($\beta_{\text{sca}}^{\text{tot}}$) in m^{-1} or km^{-1} (depending on the units of the grid).
- **cube_files.abs:** 3D field of volume absorption coefficient ($\beta_{\text{abs}}^{\text{tot}}$) in m^{-1} or km^{-1} (depending on the units of the grid).
- **cube_files.pm–index:** 3D field of the total scattering phase matrix indices.

Note: if no information is given for the volume scattering and absorption coeficients, they can be calculated before simulation (see subroutine INIT, lines 498 - 499).

cloud: Contains information for the number of the total normalized scattering phase matrices ($\tilde{\mathbf{P}}^{\text{tot}}$) and the name list of the total normalized phase matrix elements. The superscript "tot" indicates the combined optical properties of the different types of particles (e.g., molecules, aerosol, and cloud particles), which are contained in each grid–box (see Eq 3.1). For simplicity, the term "total" is omitted.

- **cloud.p11:** set of normalized scattering phase functions ($\tilde{P}_{11}^{\text{tot}}$).
- **cloud.p12:** set of $\tilde{P}_{12}^{\text{tot}}$ element of the normalized scattering phase matrices.
- **cloud.p22:** set of $\tilde{P}_{22}^{\text{tot}}$ element of the normalized scattering phase matrices.
- **cloud.p33:** set of $\tilde{P}_{33}^{\text{tot}}$ element of the normalized scattering phase matrices.
- **cloud.p34:** set of $\tilde{P}_{34}^{\text{tot}}$ element of the normalized scattering phase matrices.
- **cloud.p44:** set of $\tilde{P}_{44}^{\text{tot}}$ element of the normalized scattering phase matrices.

angle: scattering angles (ϑ) of the aforementioned set of scattering phase matrices.

x_grid: grid coordinates in x-direction.

y_grid: grid coordinates in y-direction.

z_grid: grid coordinates in z-direction.

1: example input file for Monte Carlo run; more details are given below.

Subroutine INPUT: Organization of Input Files

This subroutine opens one main input file **"mainfile"** which contains the information needed for the Monte Carlo run, including further filenames which contain input data (e.g., **"1"**). The list of input filenames and input values is in the same order as found inside INPUT.

1:

- **cube_files:** body of cloud filename (inside INPUT is read as cubefile).
- **cloud:** scattering phase matrix filename; see above (inside INPUT is read as pm_file).
- **angle:** table of scattering phase matrix scattering angles; see above (inside INPUT is read as anglefile).
- **x_grid:** see above (inside INPUT is read as xmeshfile).
- **y_grid:** see above (inside INPUT is read as ymeshfile).
- **z_grid:** see above (inside INPUT is read as zmeshfile).
- **ptoton_total:** number of incoming photons.
- **theta_0, phi_0:** solar zenith and azimuth angles in degrees.
- **albedo_lambert:** isotropic surface albedo. Note that in case of ocean reflection, the surface albedo is set to unity (for simplicity). The ocean reflection case has been implemented and tested for the needs of IPRT project.
- **n_rand:** number of bins in the lookup table for the phase matrix.
- **n_radiance:** number of selected radiances.
- **theta_choice(i), phi_choice(i):** list of the viewing zenithal and azimuth angles.

As it was already mentioned, the cube_file contains the name list of the cloud files. The format for each file, i.e., for each value is:

```
do i = xmin, xmax
   do j = ymin, ymax
      read(unit) (value(i, j, k), k = zmin, zmax)
   end do
end do
```

Value is a 3D real(8) array representing the total volume extinction, single–scattering albedo, volume scattering coefficient, and volume absorption coefficient. In case of the phase matrix index, value is a 3D integer(4) array. The format of all these files is always ”unformated” (binary files).

anglefile: contains the scattering angles (in degrees) for the scattering phase matrix:

```
read(unit) n_angle ! number of scattering angles
do i = 1, n_angle
   read(unit) angle(i) ! number of scattering angles
end do
```

pm_file: contains the set of the total scattering phase matrices:

```
read(unit) n_pf ! number of total scattering phase matrices
do i = 1, n_pf
```

read(unit1) (p11_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{11}^{\text{tot}}$
read(unit2) (p12_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{12}^{\text{tot}}$
read(unit3) (p22_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{22}^{\text{tot}}$
read(unit4) (p33_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{33}^{\text{tot}}$
read(unit5) (p34_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{34}^{\text{tot}}$
read(unit6) (p44_all(i,i_angle), i_angle = 1, n_angle) ! set of $\tilde{P}_{44}^{\text{tot}}$
end do
All the scattering phase matrix elements are a 2D read(8) array.

Note that the normalized scattering phase matrix $\tilde{\mathbf{P}}^{\text{tot}}$ is given in discrete steps with scattering angles ϑ_{i}. The latter represents the average scattering phase matrix along a finite scattering angle interval $[\vartheta_{\text{i,min}}, \vartheta_{\text{i,max}}]$ with center angle ϑ_{i}. For example, if the scattering angles are given as:

0.25
1.00
2.00
.
.
.
178
179.75
then, it corresponds to the angular bins

$[0.0, 0.5]$
$[0.0, 2.5]$
.
.
.
$[178.5, 179.5]$
$[179.5, 180.0]$

The Monte Carlo technique does a linear extrapolation to get the scattering phase matrix values at exactly 0° and 180° scattering angle. One can use discrete scattering phase matrix values but the corresponding scattering angles must be center angles of finite sized angular bins. If these angles are given, no interpolation is needed (see subroutine INPUT, lines 308 - 313).

xmeshfile: contains the grid coordinates in x-direction in units of 1 over volume scattering coefficient. In other words, if you specify the volume scattering coeffiecient in $[\text{m}^{-1}]$, then, the grid coordinates must me given in [m].

read(unit, *) xmax ! number of layers
do i = xmin, xmax + 1
read(unit, *) xmesh(i)
end do
This program reads the coordinates at xmax + 1 levels. The same stands for **ymeshfile**

and **zmeshfile**. Note that the formatted input is used; in contrast to the format of the aforementioned input files.

4.1.2 Description of the Output Files

- example/output/
  1.cloud
  1.status
  1.p11
  1.char_radiance
  1.I
  1.Q
  1.U
  1.V
  1.B3.dat
  1.DegreeOfPolarization
  1.H
  1.A
  1.R
  1.T

The subroutine OUTPUT stores the sampled data. The output files get filenames "**mainfile**".EXT (e.g., "**1**"). Where .EXT corresponds to the different extensions:

mainfile.cloud: optical thickness, pixel by pixel (for testing only).

mainfile.status: runtime information. Additionally, it contains the random number generator seed, the total number of photons, and some information with respect to the dimensions of the grid and cloud cover.

mainfile.p11: domain averaged scattering phase function. This is only for testing reasons; in order to check wether the sampling of the scattering angle is correctly calculated (if the total scattering phase function is reproduced). It might be commented out.

Radiance Output

The simulated Stokes vector (carrying the units of radiance; resulting from each pixel) is given by:

$$\vec{\mathbf{S}} = \frac{1}{N_{\mathrm{ph}}} \cdot \sum_{N_{\mathrm{ph}}} \sum_{M} \frac{\tilde{\mathbf{Z}}\left(\theta_{\mathrm{inc}}, \phi_{\mathrm{inc}}; \theta_{\mathrm{det}}, \phi_{\mathrm{det}}\right)}{4\pi} \cdot \frac{\exp\left(-\tau\right)}{\cos(\theta_{\mathrm{det}})}, \tag{4.1}$$

$\tilde{\mathbf{Z}}/4\pi$ represents the scattering probability density. Dividing by $\cos(\theta_{\mathrm{det}})$ accounts for the slant area in the radiance definition. Consequently, the contribution should be summed over all the scattering events M and the number of photons N_{ph}, and normalized over N_{ph}. Finally, the following normalized radiance is obtained:

$$\text{Normalized Radiance} = \frac{\pi \cdot \vec{\mathbf{S}}}{\mu_0 \cdot F_0 \cdot (1\,0\,0\,0)^{\mathrm{T}}}. \tag{4.2}$$

The normalized radiance holds for both the reflectivity and transmissivity (depending on the viewing direction). $F_0 \cdot (1\,0\,0\,0)^{\mathrm{T}}$ is the incident unpolarized extraterrestrial quasi–monochromatic electromagnetic radiation (solar irradiance) with μ_0 being the cosine of the solar zenith angle. In the scalar scheme, $\vec{\mathbf{S}}$ is replaced by I (scalar radiance) and $\tilde{\mathbf{Z}}$ by the scalar scattering phase finction ($\tilde{P}_{11} = \mathcal{P}$). In addition, the statistical Monte Carlo error for the radiances is given by $1/\sqrt{N_{\mathrm{ph}}}$.

mainfile.I: fist Stokes element (total radiance) and statistical error, pixel by pixel.

```
do ix = xmin, xmax
   do iy = ymin, ymax
      write(unit, '(f12.5)') stokes_select_radiance(ix,iy,1,i), &
                             error_stokes_select_radiance(ix,iy,1,i)
   end do
end do
```

mainfile.Q: second Stokes element and statistical error, pixel by pixel (same format as above).

mainfile.U: third Stokes element and statistical error, pixel by pixel (same format as above).

mainfile.V: forth Stokes element and statistical error, pixel by pixel (same format as above).

mainfile.dat: collective file for the full Stokes vector. Here one might chose to use only this instead the four aforementioned output files. In addition, the user might modify this according to his needs.

mainfile.DegreeOfPolarization: degree of polarization pixel by pixel (same format as above).

mainfile.char_radiance: scalar normalized radiance, pixel by pixel. In case you are not in need of the scalar radiance, the user might comment the following out.

```
do ix = xmin, xmax
   do iy = ymin, ymax
      write(unit, '(f12.5)') select_radiance(ix,iy,i)
   end do
end do
```

Net Horizontal Photon Transport

To illustrate horizontal fluxes in inhomogeneous atmospheres, the net horizontal photon transport (H, energy balance equation) is derived as follows (Marshak and Davis, 2005):

$$H = I - R - T(1 - a_{\mathrm{L}}) - A \tag{4.3}$$

Where R is the upward flux (photons that are reflected from the top), T corresponds to the downward flux (photons transmitted through the bottom), A defines the absorbed flux (photons absorbed from the atmosphere), and I denotes the incoming flux.

mainfile.R: albedo, pixel by pixel.

```
do ix = xmin, xmax
   do iy = ymin, ymax
      write(unit, '(f15.7)') albedo(ix, iy)
   end do
end do
```

mainfile.T: total transmission, pixel by pixel (same format as above).

mainfile.A: absorption, pixel by pixel (same format as above).

mainfile.H: net horizontal transport, pixel by pixel (same format as above).

5 How to Create Input Files

The input variables should be stored in files (input files) with the same format as they are read in SPARTA (see subroutine INPUT). In the directory utils/, two programs for storing the input variables are given:

- **make.f90**: it stores the input variables in the input files with the same as that they are read in SPARTA.
- **mainfile.f90**: It creates the main input file **"mainfile"**.

Note that both programs create the input files needed for the attached example: case B3 ("aerosol profile") of the IPRT–A (phase A). In brief, this test case includes a standard atmosphere with Rayleigh scattering and molecular absorption. Aerosol with a specified optical thickness profile is added; spheroidal aerosol particles are assumed. More information is found in Barlakas (see Section 4.2, 2016b) or in Emde et al. (2015).

The user should follow the instructions found within these two programs and modify them according to his needs. Note that for the compilation and run of both codes the rules intoduced in 2.4 should be applied.

5.1 Description of make.f90

In directory utils/, two subdirectories are found, info/ and input/. In the first subdirectory, all the information needed to create the setup is given:

- sizedistr_spheroid.dat: contains a brief description of the optical properties of the ensemble of particles investigated in the current example (prolate spheroids).
- tau_rayleigh_350.dat: scattering optical thicknesses (model atmosphere with 30 layers).
- tau_molabs_350.dat: absorption optical thicknesses (model atmosphere with 30 layers).
- tau_aerosol.dat: aerosol optical thickness profile (model atmosphere with 30 layers).
- angles: scattering angles of the scattering phase matrix; in the first line, the total number of the scattering angles is stored.
- spheroidal.p11: scattering phase function as a function of the scattering angle.
- spheroidal.p12: $\tilde{P}_{12}$ element of the scattering phase matrix as a function of the scattering angle.

- spheroidal.p22: $\tilde{P}_{22}$ element of the scattering phase matrix as a function of the scattering angle.
- spheroidal.p33: $\tilde{P}_{33}$ element of the scattering phase matrix as a function of the scattering angle.
- spheroidal.p34: $\tilde{P}_{34}$ element of the scattering phase matrix as a function of the scattering angle.
- spheroidal.p44: $\tilde{P}_{44}$ element of the scattering phase matrix as a function of the scattering angle.

The second subdirectory corresponds to the directory where the input files are stored. make.f90 reads the information contained in info/ and stores the info in the desired format in input/. A comprehensive explanation for all the input, output variables, and line by line procedure is found within the source code.

5.2 Description of mainfile.f90

mainfile.f90 creates the main input file **"mainfile"**. It is an "interactive" code. Once it is compiled and run all the instructions needed to create the main input file are illustrated in the terminal (according to the current example). A comprehensive explanation for the input, output, and line by line procedure is found within the source code. The user should slightly modify the source code according to his needs (only lines 107 - 118; explanation is incuded in the source code).

Finally, the main input file is stored in input/.

Bibliography

Anderson, G., Clough, S., Kneizys, F., Chetwynd, J., and Shettle, E.: AFGL Atmospheric Constituent Profiles (0–120 km), Tech. Rep. AFGL-TR-86-0110, AFGL (OPI), Hanscom AFB, MA 01736, 1986.

Barlakas, V.: A New Three–Dimensional Vector Radiative Transfer Model and Applications to Saharan Dust Fields, Ph.D. thesis, University of Leipzig University of Leipzig, Faculty of Physics and Earth Sciences, http://www.qucosa.de/recherche/frontdoor/?tx_slubopus4frontend[id]=20746, 2016b.

Barlakas, V., Macke, A., Wendisch, M., and Ehrlich, A.: Implementation of polarization in a 3D Monte Carlo Radiative Transfer Model, Wissenschaftliche Mitteilungen aus dem Institut für Meteorologie der Universität Leipzig, 52, 1–14, iSBN: 978-3-9814401-2-6, 2014.

Barlakas, V., Macke, A., and Wendisch, M.: SPARTA – Solver for Polarized Atmospheric Radiative Transfer Applications: Introduction and application to Saharan dust fields, J. Quant. Spectrosc. Radiat. Transfer, 178, 77 – 92, doi:10.1016/j.jqsrt.2016.02.019, 2016a.

Buras, R. and Mayer, B.: Efficient unbiased variance reduction techniques for Monte Carlo simulations of radiative transfer in cloudy atmospheres: The solution, J. Quant. Spectrosc. Radiat. Transfer, 112, 434 – 447, http://dx.doi.org/10.1016/j.jqsrt.2010.10.005, 2011.

Cahalan, R. F., Oreopoulos, L., Marshak, A., Evans, K. F., Davis, A. B., Pincus, R., Yetzer, K. H., Mayer, B., Davies, R., Ackerman, T. P., Barker, H. W., Clothiaux, E. E., Ellingson, R. G., Garay, M. J., Kassianov, E., Kinne, S., Macke, A., O'Hirok, W., Partain, P. T., Prigarin, S. M., Rublev, A. N., Stephens, G. L., Szczap, F., Takara, E. E., Várnai, T., Wen, G., and Zhuravleva, T. B.: The I3RC: Bringing together the most advanced radiative transfer tools for cloudy atmospheres, Bull. Am. Meteorol. Soc., 86, 1275–1293, doi:10.1175/BAMS-86-9-1275, 2005.

Collins, D. G., Blättner, W. G., Wells, M. B., and Horak, H. G.: Backward Monte Carlo calculations of the polarization characteristics of the radiation emerging from spherical-shell atmospheres, Appl. Opt., 11, 2684–2696, doi:10.1364/AO.11.002684, 1972.

de Haan, J. F., Bosma, P. B., and Hovenier, J. W.: The adding method for multiple scattering calculations of polarized light, Astron. Astr., 183, 371–391, http://adsabs.harvard.edu/abs/1987A%26A...183..371D, 1987.

Emde, C., Buras, R., Mayer, B., and M., B.: The impact of aerosols on polarized sky radiance: model development, validation, and applications, Atmos. Chem. Phys., 10, 383–396, doi:10.5194/acp-10-383-2010, 2010.

Emde, C., Barlakas, V., Cornet, C., Evans, F., Korkin, S., Ota, Y., Labonnote, L. C., Lyapustin, A., Macke, A., Mayer, B., and Wendisch, M.: IPRT polarized radiative transfer model intercomparison project – Phase A, J. Quant. Spectrosc. Radiat. Transfer, 164, 8 – 36, http://dx.doi.org/10.1016/j.jqsrt.2015.05.007, 2015.

Gasteiger, J., Groß, S., Freudenthaler, V., and Wiegner, M.: Volcanic ash from Iceland over Munich: mass concentration retrieved from ground-based remote sensing measurements, Atmos. Chem. Phys., 11, 2209–2223, doi:10.5194/acp-11-2209-2011, 2011.

Hovenier, J. W., van der Mee, C., and Domke, H.: Transfer of Polarized Light in Planetary Atmospheres, Kluwer Academic Publishers, Dordrecht, Netherlands, 2004.

Kokhanovsky, A. A., Budak, V. P., Cornet, C., Duan, M., Emde, C., Katsev, I. L., Klyukov, D. A., Korkin, S. V., C-Labonnote, L., Mayer, B., Min, Q., Nakajima, T., Ota, Y., Prikhach, A. S., Rozanov, V. V., Yokota, T., and Zege, E. P.: Benchmark results in vector atmospheric radiative transfer, J. Quant. Spectrosc. Radiat. Transfer, 111, 1931 – 1946, http://dx.doi.org/10.1016/j.jqsrt.2010.03.005, 2010.

Macke, A., Mitchell, D. L., and V., B. L.: Monte Carlo radiative transfer calculations for inhomogeneous mixed phase clouds, Phys. Chem. Earth, 24, 237–241, http://dx.doi.org/10.1016/S1464-1909(98)00044-6, 1999.

Marchuk, G. I., Mikhailov, G. A., and A., N. M.: The Monte Carlo Methods in Atmospheric Optics, Springer-Verlag, New York, 1980.

Marshak, A. and Davis, A.: 3D Radiative Transfer in Cloudy Atmospheres, Springer Berlin Heidelberg, http://dx.doi.org/10.1007/3-540-28519-9_4, 2005.

Mishchenko, M. I.: Gustav Mie and the fundamental concept of electromagnetic scattering by particles: A perspective, J. Quant. Spectrosc. Radiat. Transfer, 110, 1210 – 1222, http://dx.doi.org/10.1016/j.jqsrt.2009.02.002, 2009.

Mishchenko, M. I. and Travis, L. D.: Satellite retrieval of aerosol properties over the ocean using polarization as well as intensity of reflected sunlight, J. Geophys. Res., 102, 16 989–17 013, doi:10.1029/96JD02425, 1997.

Mishchenko, M. I., Travis, L. D., and Lacis, A. A.: Scattering, Absorption, and Emission of Light by Small Particles, Cambridge University Press, iSBN: 9780521782524, 2002.

Natraj, V., Li, K.-F., and Yung, Y. L.: Rayleigh scattering in planetary atmospheres: Corrected tables through accurate computation of X and Y functions, Astrophys. J., 691, 1909–2009, http://stacks.iop.org/0004-637X/691/i=2/a=1909, 2009.

Rozanov, V. V. and Lyapustin, A. I.: Similarity of radiative transfer equation: Error analysis of phase function truncation techniques, J. Quant. Spectrosc. Radiat. Transfer, 111, 1964–1979, 2010.

List of Symbols

Adopted by Barlakas (2016b).

α_{L}	-	Surface albedo
β_{abs}	m^{-1}	Spectral volumetric absorption coefficient
$\beta_{\mathrm{abs}}^{\mathrm{tot}}$	m^{-1}	Total volumetric absorption coefficient
β_{ext}	m^{-1}	Spectral volumetric extinction coefficient
$\beta_{\mathrm{ext}}^{\mathrm{tot}}$	m^{-1}	Total volumetric extinction coefficient
β_{sca}	m^{-1}	Spectral volumetric scattering coefficient
$\beta_{\mathrm{sca}}^{\mathrm{tot}}$	m^{-1}	Total volumetric scattering coefficient
δ	-	Depolarization factor
ϵ	$\mathrm{A\,s\,V^{-1}\,m^{-1}}$	Electric permeability
ζ	rad or °	Phase of the electric field vector
η	rad or °	Rotation angles
θ	rad or °	Atmospheric zenith angle
θ_0	rad or °	Solar zenith angle
θ_{det}	rad or °	Zenith angle, viewing (detector) direction
θ_{inc}	rad or °	Zenith angle, incident direction
θ_{refl}	rad or °	Zenith angle, reflection direction
θ_{sca}	rad or °	Zenith angle, scattering direction
ϑ	rad or °	Scattering zenith angle
κ	$\mathrm{V\,s\,A^{-1}\,m^{-1}}$	Magnetic permittivity
λ	m	Wavelength
μ	-	Cosine of the zenith angle
ρ_λ	-	Spectral albedo
τ	-	Cloud optical thickness
τ_{cum}	-	Cumulated optical thickness
τ_{rand}	-	Randomly chosen optical thickness
ϕ	rad or °	Atmospheric azimuth angle
ϕ_0	rad or °	Solar azimuth angle
ϕ_{det}	rad or °	Azimuth angle, viewing (detector) direction
ϕ_{inc}	rad or °	Azimuth angle, incident direction
ϕ_{refl}	rad or °	Azimuth angle, reflection direction
ϕ_{sca}	rad or °	Azimuth angle, scattering direction
φ	rad or °	Scattering azimuth angle
ξ	-	Random number
$\tilde{\omega}$	-	Single–scattering albedo

$\tilde{\omega}^{\mathrm{tot}}$	-	Total single–scattering albedo
ω_{c}	$\mathrm{s}^{-1} = \mathrm{Hz}$	Circular frequency
d^2A	m^2	Infinitesimal area element
$\mathrm{d}^2A_\perp$	m^2	Infinitesimal area element perpendicular to $\widehat{k}$
$\Theta_j\,(j=1,2)$	rad or $^\circ$	Incident and refracted angles with respect to the surface normal
Φ_λ	$\mathrm{J\,s^{-1}\,nm^{-1}}$	Spectral radiant energy flux
Φ_{abs}	$\mathrm{J\,s^{-1}\,nm^{-1}}$	Radiant energy flux absorbed by an individual particle
Φ_{sca}	$\mathrm{J\,s^{-1}\,nm^{-1}}$	Radiant energy flux scattered by an individual particle
X, Y	-	Solutions of the two nonlinear integral equations of radiative transfer equation
$\mathrm{d}^2\Omega$	sr	Infinitesimal solid angle
b	-	Hemispheric backscatter ratio
c	$\mathrm{m\,s^{-1}}$	Velocity of light in vacuum
$\hat{e}_1$, $\hat{e}_2$, $\hat{e}_3$	-	Euklidic base vectors
g	-	Asymmetry parameter
h	J s	Planck's constant
$\widehat{k}$	-	Direction of propagation
$\widehat{\mathrm{k}}_{\mathrm{inc}}$	-	Incident direction
$\widehat{\mathrm{k}}_{\mathrm{inc},0}$	-	Initital incident direction
$\widehat{\mathrm{k}}_{\mathrm{sca}}$	-	Scattering direction
$\widehat{\mathrm{k}}_{\mathrm{refl}}$	-	Reflection direction
k	m^{-1}	Wavenumber
k_{B}	$\mathrm{J\,deg^{-1}}$	Boltzmann constant
l_{back}	m	Backward step
$\widehat{n}$	-	Surface orientation of d^2A
$\widehat{n}_\perp$	-	Surface orientation of $\mathrm{d}^2A_\perp$
$\tilde{n}$	-	Complex index of refraction
n_j	m^{-3}	Particle number density
n_{re}	-	Real part of $\tilde{n}$
r	m	Radius of aerosol particle
s^2	$\mathrm{m\,s^{-1}}$	Mean square surface slope of waves
t	s	Time
w	$\mathrm{m\,s^{-1}}$	Near–surface wind velocity
x, y, z	-	Axes of Cartesian coordinate system (z vertical altitude above ground)
z_{base}	m	Medium base, altitude
z_{top}	m	Medium top, altitude
IA	m	Complex amplitude scattering matrix
$A_{ij}\,(i,j=1,2)$	m	Complex scattering amplitudes
B_λ	$\mathrm{W\,m^{-2}\,sr^{-1}\,\mu m^{-1}}$	Spectral Planck function
BRDF	-	Bidirectional reflection distribution function

C_{abs}	m^2	Absorption cross section
$\langle C_{\mathrm{abs}} \rangle$	m^2	Average absorption cross section per particle
C_{ext}	m^2	Extinction cross section
$\langle C_{\mathrm{ext}} \rangle$	m^2	Average extinction cross section per particle
C_{sca}	m^2	Scattering cross section
$\langle C_{\mathrm{sca}} \rangle$	m^2	Average scattering cross section per particle
D	m	Diameter of cloud particle
$\vec{\mathbf{E}}$	$\mathrm{V\,m^{-1}}$	Complex electric field vector
$\vec{\mathbf{E}}_0$	$\mathrm{V\,m^{-1}}$	Complex electric amplitude vector
E_{rad}	J	Radiant energy
ERF	$\mathrm{W\ m^{-2}}$	Effective radiative forcing
$\left\langle \vec{\mathbf{F}} \right\rangle$	$\mathrm{W\,m^{-2}}$	Poynting vector
F_0	$\mathrm{W\,m^{-2}\,nm^{-1}}$	Extraterrestrial irradiance
F_λ	$\mathrm{W\,m^{-2}\,nm^{-1}}$	Spectral irradiance
$F_\lambda^\downarrow$	$\mathrm{W\,m^{-2}\,nm^{-1}}$	Downward spectral irradiance
$F_\lambda^\uparrow$	$\mathrm{W\,m^{-2}\,nm^{-1}}$	Upward spectral irradiance
F_{inc}	$\mathrm{W\,m^{-2}\,nm^{-1}}$	Incident irradiance
$\vec{\mathbf{H}}$	$\mathrm{A\,m^{-1}}$	Complex magnetic field vector
I_λ	$\mathrm{W\,m^{-2}\,nm^{-1}\,sr^{-1}}$	Spectral radiance
$\mathbf{M}$	-	Mueller matrix
$\mathbf{M}_{\mathrm{refl}}$	-	Matrix based on Fresnel formulas
M		Number of scattering events
$M_{ij}\ (i,j=1,2)$	-	Mueller matrix elements
N	-	Number of particles
N_{ph}	-	Number of particles photons
$\mathbf{P}$	m^2	Scattering phase matrix
$\tilde{\mathbf{P}}^{\mathrm{tot}}$	m^2	Total normalized scattering phase matrix
$\tilde{\mathbf{P}}$	-	Normalized scattering phase matrix
$\langle \mathbf{P} \rangle$	m^2	Average scattering phase matrix
$\mathcal{P}$	sr	Phase function
P	-	Degree of polarization
$P_{ij}\ (i,j=1,2)$	m^2	Scattering phase matrix elements
P_{cir}	-	Degree of circular polarization
P_{lin}	-	Degree of linear polarization
Q	$\mathrm{W\,m^{-2}}$	Parallel minus perpendicular linear polarized irradiance
$\mathbf{R}$	-	Rotation matrix
$\mathbf{IR}_{\mathrm{L}}$	-	Lambertian (isotropic) reflection matrix
$\mathbf{IR}_{\mathrm{ocean}}$	-	Ocean reflection matrix
RD	%	Relative differences between the 2D and 1D modes
$\mathrm{RD}_{\mathrm{2D\,-\,IPA}}$	%	Relative differences between the 2D and IPA modes
$\mathrm{RD}_{\mathrm{IPA\,-\,1D}}$	%	Relative differences between the IPA and 1D modes
$\mathrm{RD}_{\mathrm{V\,-\,S}}$	%	Relative differences between the vector and scalar modes

RF	W m^{-2}	Radiative forcing
$\vec{\mathbf{S}}$	$W\,m^{-2}$	Stokes vector
$\vec{\mathbf{S}}_{inc}$	$W\,m^{-2}$	Incident Stokes vector
$\vec{\mathbf{S}}_{sca}$	$W\,m^{-2}$	Scattered Stokes vector
S	-	Bidirectional shadowing function
T	K	Absolute temperature
U	$W\,m^{-2}$	Linear polarized irradiance under 45°
V	$W\,m^{-2}$	Circularly polarized irradiance
W	-	The exponential term of W describes the Gaussian distribution of the surface slope
$\mathbf{Z}$	m^2	Transformation phase matrix
$\tilde{\mathbf{Z}}$	-	Normalized transformation phase matrix
$\langle\mathbf{Z}\rangle$	m^2	Ensemble average transformation phase matrix
$\langle\mathbf{Z}_j\rangle$	m^2	Ensemble average transformation phase matrix for one particle

List of Abbreviations

1D	One–dimensional
2D	Two–dimensional
3D	Three–dimensional
BOA	Bottom Of Atmosphere
CPU	Central Processing Unit
EM	ElectroMagnetic radiation
GCC	GNU Compiler Collection
I3RC	Intercomparison of 3–D Radiation Codes
ICA	Independent Column Approximation
IPA	Independent Pixel Approximation
IPRT	International Polarized Radiative Transfer
LEM	Local Estimate Method
MC–UNIK	Monte Carlo model of the Leibniz Institute of Marine Sciences (now GEOMAR) at the UNIversity of Kiel (GEOMAR - Helmholtz Centre for Ocean Research Kiel)
PDF	Probability Density Function
SPARTA	Solver for Polarized Atmospheric Radiative Transfer Applications
TOA	Top Of Atmosphere
TROPOS	Leibniz Institute for Tropospheric Research

List of Figures

3.1 A representation of the scheme of the photon path within the 3D domain of the Monte Carlo radiative transfer model. 9

List of Tables

3.1 SPARTA efficiency. 12