ARTDECO V1.2.0
Technical User Guide

distributed by :
AERIS/ICARE Data and Services Center
Université de Lille, 59655 Villeneuve d'Ascq, France
contact@icare.univ-lille1.fr

http://www.icare.univ-lille1.fr/projects/artdeco

Victor Winiarek
Philippe Dubuisson
Laurent Labonnote
Mathieu Compiègne
Landa Mcharek
Nicolas Henriot
Jacques Descloitres

Change records of version 1.2.0

Scientific update	–
Bug Fix	Minor modification to handle BRDF
Technical update	Add python3.7 compatibility , python2.7 is still usable

Table of contents

1. Introduction

ARTDECO (Atmospheric Radiative Transfer Database for Earth Climate Observation) is a numerical tool that gathers several models and data for the simulation of Earth atmosphere radiances and radiative fluxes as observed with passive sensors (hyperspectral excluded) in the UV to thermal IR range.

In ARTDECO, users can either access a library of predefined conditions (atmosphere profile, surface, aerosol and cloud description, filter transmission, etc.) or use their own description through ASCII input files. User-defined aerosol and cloud properties can be computed. Users can choose among available models (several methods for the truncation of the phase function, several 1D radiative transfer equation solver) to compute radiative quantities corresponding to the scene. This software package is especially powerful to study and optimize performances of different methodologies to model radiative quantities corresponding to a given scene (Compiègne et al., 2013).

The scientific numeric core of ARTDECO is written in Fortran90. The configuration files and the main driver are written in python and libraries are in ASCII format, leading to a very high flexibility and portability.

Main features of ARTDECO :

Radiative transfer (1D)
- DISORT2.1 discrete ordinate(I, thermal), Stamnes et al. (1988)
- adding-doubling (I Q U V), de Haan et al. (1987)
- Monte-Carlo (I Q U V) , 1D version of Cornet *et al.* (2010)
- Single scattering approximation (I Q U V)

Spectral range
- From UV to infrared: 0.2 to 50 µm
- Solar spectrum: Kurucz high resolution spectrum

Gaseous absorption
- k-distribution coefficients for absorption lines : H_2O, CO_2, O_3, O_2, N_2O, CO, CH_4, N_2
- Continuum absorption: H_2O, CO_2, O_3, N_2

Truncation of phase function
- δ–Potter, δ-M or δ–fit methods

Intensity correction for first order scattering (TMS)
- Nakajima & Tanaka (1988)

Instrumental spectral response
- For usual satellite sensors : MODIS, POLDER, IIR, SEVIRI, etc.

Radiative quantities

- Radiance ($W.m^{-2}.sr^{-1}$), flux ($W.m^{-2}$), warming rate (K/day) and net radiative heating rate ($W/m^2/m$)
- Top-of-Atmosphere, surface level or any atmospheric level
- For any viewing geometry typical for 1D atmospheric observations
- At a given wavelength or for a specified spectral resolution (from 10 to 400 cm^{-1})
- For the solar and infrared spectrum (0.2 to 50 µm)

Atmosphere definition (plane parallel approximation (1D))

- Pressure, temperature and gas concentration vertical profiles
- AFGL & Mc Clatchey profiles
- User-defined profiles

Particle optical properties:

- Refractive index for liquid and ice water and for aerosol material (OPAC database, $\lambda = 0.25 – 40$ µm)
- Microphysical properties definition for water clouds (Stephens, OPAC), for aerosols (OPAC) and for cirrus (Baum et al., POLDER/PARASOL model)
- Henyey-Greenstein (1941) approximation
- Mie spherical particles
- Ray tracing for hexagonal monocrystal particles : pristine(PHM), rough (RHM) or with inclusions (IHM)

Surface

- Lambertian spectral albedo for various surfaces
- Cox & Munk ocean surface BRDF/BPDF
- “Ross–Li”&“Rahman–Pinty–Verstraete” with hot-spot land surface BRDF
- “Maignan & Bréon” land surface BPDF

2. Compilation of the ARTDECO software

2.1 Download

The package Artdeco is distributed by AERIS/ICARE data and services center :

http://www.icare.univ-lille1.fr/projects/artdeco

2.2 Prerequisite softwares

ARTDECO should run on most Unix, Linux systems as long as the following softwares are available :

- GNU Fortran (tested on version 4.7.7) or Intel Fortran compilers,
- python 2.x and common modules (such as os, sys, shutil).
- netCDF4 module for python if netCDF4 output files will be written.
- h5py module for python if HDF5 output files will be written.

It has been succesfully compiled and run on MacOS as well.

2.3 Unpack and compile the distribution

To unpack the distribution with the *tar* tool and enter in the *src* directory :

```
tar -zxvf ARTDECO_V1.2.0.tar.gz
cd  ARTDECO_V1.2.0
cd src
```

If your computer is **32-bit**, you then need to open "**Makefile**" file and change :

```
#FFLAGS = -m64 -O2 -fbounds-check
FFLAGS = -m32 -O2 -fbounds-check
```

The Makefile (and all secondary Makefiles) is pre-configured for use with GNU Fortran and Intel Fortran. To switch between those, you have also to edit this file :

```
#####################
# Choose compiler
FC = gfortran
#FC = ifort
```

Finally, to compile ARTDECO and write the compilation log in a specific file **:**

```
make cleanall && make artdeco >& compil.log
```

An executable named 'artdeco' should now be in the directory.

2.4 Launch a test case

Once in the main directory, a test case can be executed with the command :

```
python Artdeco.py execute examples/kokhanovsky/rayleigh/disort/config
```

The output files written in *./out/kokhanovsky_rayleigh_disort* should be close to the reference ones in *./examples/kokhanovsky/rayleigh/disort*.

Other examples are available (configuration files and reference outputs) in the directory *./examples*. These examples are the same as in Kokhanovsky *et al.* (2010).

Another example can be found in ./examples/kdis. In this configuration, two particles are modelled using k–distributions :

- opac_fog : the Legendre coefficients are directly read in ./lib/betal.
- baum_ghm_cirrus_40 : the Legendre coefficients are computed from the optical properties in ./lib/baum/ghm.

Before launching the test case, the baum package must be uncompressed in ./lib/baum with the commands :

```
cd lib/baum
tar -xzvf ghm.tar.gz
cd ../..
```

then the test case can be launched with :

```
python Artdeco.py execute examples/kdis/config
```

The results are written in ./out/kdis_opac+cirrus and can be compared with the reference ones in ./examples/kdis.

3. Package content

Artdeco.py is in the main directory. It is the main driver to launch a run. The ARTDECO package contains 9 directories :

- *./src* : The main ARTDECO source files, particularly the main Makefile and artdeco.f90 that contains the PROGRAM statement are in *./src*. All other files in that directory begin with the prefix “artdeco_”. There are then 12 sub-directories :

    - *./src/disort* : contains a Makefile and the **DISORT2.1 RT model** source files (named with the prefix “od_”).

    - *./src/doad* : contains a Makefile and the **Vesperini adding-doubling RT model** source files (named with the prefix “doad_”).

    - *./src/mcrad1d* : contains a Makefile and **Monte Carlo RT model** source files (named with the prefix “mcrad1d_”).

    - *./src/cont* : contains a Makefile and routines for the **gas continuum absorption** (named with the prefix “cont_”).

    - *./src/hm* : contains a Makefile and the **Hexagonal Monocrystal optical properties model** source files (named with the prefix “hm_”).

    - *./src/mie3* : contains a Makefile and **Meerhoff Mie code** source file.

    - *./src/nm* : contains a Makefile and numerical methods source files (named with the prefix “nm_”).

    - *./src/sol* : contains a Makefile and the routines to compute the **solar position** in the sky and the variation of the solar constant over a year (named with the prefix “sol_”).

    - *./src/surface* : contains a Makefile and source files for routines that provide **BRDF/BPDF surface properties**.

    - *./src/trunc* : contains a Makefile and source files for routines that perform **phase matrices truncations** (named with the prefix “trunc_”)

- *./config* : The subdirectory *reference* contains **reference configuration files** that should not be modified. It is then advised to create new configuration directories in *./config* (see Section 4 for the ARTDECO instructions).
- *./lib* : contains data libraries. These files are used as input but are sometimes written as an output for further use by ARTDECO.
    - *./lib/atm* : The subdirectory *reference/* contains **atmospheric profiles** (Pressure, temperature, air density, absorbing gases density). Then the program will copy (and possibly modify) the required file in *atm/*.
    - *./lib/baum* : contains Look-Up Tables for optical properties of the **Baum cirrus**[1].
    - *./lib/filter* : contains **instrument spectral response functions**.
    - *./lib/ihm* : contains phase matrix of inclusions computed by **Inhomogeneous Hexagonal Monocrystal optical properties** code for further re-use.
    - *./lib/kdis* : The subdirectory *./lib/kdis/reference* contains coefficients for the **correlated k-distribution** treatment of gas absorption. Then the program will copy (and possibly modify) the required file in *./lib/kdis*.
    - *./lib/opt* : contains **optical properties** of particles (aerosol or cloud).
    - *./lib/prop* : contains files with **microphysical properties** definition for the computation of particles optical properties.
    - *./lib/refind* : contains **refractive indices** for material to be used to compute new optical properties.
    - *./lib/solrad* : contains **solar spectra**.
    - *./lib/surface* : contains **surface definition** files. *surfalb_* files contain albedo as a function of wavelength for lambertian surfaces while *surfbrdf_* contains parameters for BRDF/BPDF models.
- *./examples* : contains reference examples (configurations from Kokhanovsky et al., 2010).
- *./out* : contains ARTDECO outputs. For each run a directory containing the results is created in *./out* . Python configuration files will also be copied in this created sub-

1 http://www.ssec.wisc.edu/~baum/

directory.

- *./input* : The python driver will write some ASCII and python input files here when launching ARTDECO.
- *./mod* : contains Fortran modules generated when compiling the program.
- *./obj* : contains object files generated when compiling the program.
- *./doc* : contains user's guides and necessary documentation.

4. Run instructions

The main driver to execute ARTDECO is Artdeco.py.

The command :

```
python Artdeco.py usage
```

will indicate how to launch any functionality of Artdeco.py.

```
Python Artdeco.py new <configuration_directory>
```

will create <configuration_directory> and copy reference configuration files (all python files) in it. If <configuration_directory> already exists, it will not be erased and the command will raise an error to alert on the problem. It is advised to create the new configuration directories in *./config*.

The next step is to edit some main keywords in the file main.py in the <configuration_directory> (see Section 5.1).

Depending on the options chosen in main.py, several secondary configuration files will have to be edited in <configuration_directory>. Once the file main.py has been edited, the command :

```
python Artdeco.py help-configuration <configuration_directory>
```

will indicate which secondary configuration files have to be edited (see Section 5.2).

The command :

```
python Artdeco.py check-configuration <configuration_directory>
```

will check the configuration without launching the execution of the program in order to detect any error.

Finally, the program is launched with the command :

```
python Artdeco.py execute <configuration_directory>
```

5. Run configuration

5.1 Main configuration file : main.py

NB : For users that would not be familiar with python files, it must be mentioned first that any character following a '#' is considered a comment and is not interpreted by the python script. Any new comment line can be added, as well as erased, without consequence on the run. Secondly, the different keywords can be moved throughout the file, as long as the lines underneath the warning “No user changes should be necessary under this line.” are not modified. These remarks still hold for the secondary configuration files that users will have to modify later.

In main.py, the following keywords have to be edited :

- **Mode** : Mode of simulation for the radiative transfer. Two options are available :
    - **“mono”** for monochromatic mode. In this case the file *mono.py* will have to be edited (see Section 5.2 for the edition of the secondary files).
    - **“kdis”** for correlated k-distribution mode. In this case the file *kdis.py* will have to be edited.
- **With_particles** : Whether particles (cloud or aerosols) are taken into account.
    - **True** if this is the case, then the file *particles.py* has to be edited.
    - **False** otherwise.
- **With_atmosphere** :
    - **True** if an atmospheric profile is taken into account, then the file *atmosphere.py* has to be edited.
    - **False**. In this case, the program will run with a single layer void of gas.
- **RTE_solver** : Name of the radiative transfer equation solver ('RTE solver' from now on).
    - **“disort”** for DISORT2.1 RTE solver (Stamnes *et al.*, 1988). It is a scalar RTE solver based on the discrete ordinate method. It is widely used, very stable and well documented. The manual[2] that comes with it also gives a general introduction on 1D radiative transfer. It allows to account for thermal emission and can handle BRDF surface definition. The secondary configuration file *disort.py* will have to be edited.

2 http://www.icare.univ-lille1.fr/sites/default/files/files/disortreport1_1.pdf

- **“doad”** for the adding-doubling code developed at the LOA by M. Vesperini. It is based on de Haan *et al.* (1987). It accounts for polarization and can handle BRDF/BPDF surface definition.. That code is used for the “Radiation Budget and Clouds” inversion pipeline of PARASOL[3]. The secondary configuration file *doad.py* will have to be edited.

- **“mcrad1d”** for the Monte Carlo code developed at the LOA. It accounts for polarization and can handle BRDF/BPDF surface definition, but only in monochromatic mode. The secondary configuration file *mcrad1d.py* will have to be edited.

- **“sinsca”** for a single-scattering approximation code that accounts for polarization and can handle BRDF/BPDF surface definition. That code can be used for fast resolution in case of very low optical depth, to test the single scattering approximation against other RTE solvers. However, its main usage in ARTDECO is to apply the single scattering correction (TMS correction, see details for the edition of *particles.py*).

- **“none”** if no radiative transfer solver is to be used. In this case only optical properties and/or truncation and Legendre polynomial expansion are performed.

- **Legendre_and_truncation** : This keyword is read only if RTE_solver = “none”.

  - **False** : In this case the program will only compute the particles optical properties.

  - **True** : In this case, the program will possibly compute optical properties (if not provided as an input) and perform the truncation and Legendre polynomial expansion.

- **Rayleigh_scattering** :

  - **True** : if the Rayleigh molecular scattering is taken into account.

  - **False** otherwise.

- **Stokes_components** : Number of Stokes parameters to take into account.

  - **1** : scalar radiative transfer equation (no polarization).
  - **3** : three Stokes parameters approximation (I, Q, U).
  - **4** : all Stokes parameters (I, Q, U, V).

- **Prefix_of_input_files** : When launching ARTDECO through the python script *Artdeco.py,* the program will write some ASCII configuration files in *input/*. All these

3 https://parasol.cnes.fr/fr

files names will begin with the string specified with this keyword.

- **Verbose** : Should the program be verbose ? **True** of **False**.

- **Warnings** : Should the program display warnings ? **True** or **False**.

- **Log_file** : Name of the file in which logs are written.

- **Output_directory** : Name of the sub-directory in which results are written. If the output directory exists, files will be overwritten, otherwise the directory is created.

- **Output_format** :
  - **"ascii"** : ASCII files only.
  - **"netcdf"** : for NetCDF outputs.
  - **"hdf"** : for HDF outputs.
  - **"all"** : for ASCII, NetCDF and HDF outputs.

  ASCII files are nonetheless always written no matter the chosen option.

- **Print_betal** :
  - **True** : the Legendre expansion coefficients will be written in the output directory.
  - **False** otherwise.

- **Print_recomposed_phase_matrix** :
  - **True** : the recomposed phase matrix from the Legendre coefficients will be written in the output directory.
  - **False** otherwise.

Screenshot of the file main.py

```
## In which mode is the code running ?
##      - "mono" for monochromatic mode.
##      - "kdis" for k-distribution parameterization.
Mode = "kdis"

## Are there particles (or cloud) ?
With_particles = False

## Is there an atmosphere ?
With_atmosphere = True

## Which radiative transfer model will be used ?
##      - "none" : only optical properties will be computed (no RT model).
##      - "disort" : Discrete ordinate model DISORT 2.0.
##      - "doad" : Verperini Doubling-adding model.
##      - "sinsca" : Single scattering approximation.
##      - "mcrad1d" : MCRAD1D Monte Carlo model.
RTE_solver = "sinsca"

## If no RT model is used, should ARTDECO compute at least Legendre
## polynomial expansion and phase matrix truncation ? If no, ARTDECO
## will only compute optical properties.
Legendre_and_truncation = False

## Is the Rayleigh molecular scattering taken into account ?
Rayleigh_scattering = True

## Which level of polarization should be taken into account ?
##      - 1 : scalar.
##      - 3 : I, Q, U components of Stokes vector.
##      - 4 : All components of Stokes vector.
Stokes_components = 1

#############################
## Program  options.

## Prefix used for the configuration files which will be written.
## These configuration files will be written in the directory "./input".
Prefix_of_input_files = "test_final"

Verbose = True
Warnings = False
Log_file = "log_Test.txt"

#############################
## Output options.

## Name of the output subdirectory. It is located in "./out/".
## If it does not exist, it is created.
## If it exists, files in it will be erased.
Output_directory = "test_output"

## Format of output files :
##      - "ascii" for ascii files only.
##      - "netcdf" or "NetCDF" for NetCDF (not implemented yet).
##      - "hdf" or "HDF" for HDF (not implemented yet).
##      - "both" for both HDF and NetCDF (not implemented yet).
## In all cases, ascii files will be written.
Output_format = "ascii"

## Should the Legendre expansion coefficients be written in the output files ?
Print_beta1 = False

## Should the recomposed phase matrix from the Legendre
## coefficients be written in the output files ?
Print_recomposed_phase_matrix = True
```

5.2 Secondary configuration files

Once the main configuration file main.py has been modified, the secondary configuration files that must be modified (based on the configuration specified in main.py) are indicated by the command :

```
python Artdeco.py help-configuration <configuration_directory>
```

5.2.1 Run in monochromatic mode – Edition of *mono.py*

- **Wavelength_unit** : keywords to specify the unit of the specified wavelengths in input. Valid strings are :
  - **"cm-1"** or **"cm^-1"** for wavelengths in cm^{-1}.
  - **"microm"**, **"mum"** or **"micrometers"** for wavelengths in μm.

- **Wavelength_list** : list of *float* or *integer* specifying simulated wavelengths. The items in the list might be in ascendant or descendant order, but not unordered.

- **Depolarization_value** : Value (*float* or *int*) of the depolarization factor. It might also be the string **"default"**. In this case, the value is 0,0279 (following Young, 1980) in the V1.2.0 of ARTDECO.

Screenshot of the file mono.py :

```
################################################################################
##                                                                            ##
## Additional configuration file used to define specific parameters           ##
## for the monochromatic mode of ARTDECO.                                     ##
##                                                                            ##
################################################################################

## Wavelength unit might be :
##         - "cm^-1" or "cm-1".
##         - "microm", "micrometers", "mum".
## Wavelengths in the list must be in ascendant or descendant order,
##the program will deal with it.
Wavelength_unit = "cm-1"
Wavelength_list = [5000., 7500.]

## Depolarization mode :
##         - "default" : value = 0.0279 for every wavelength.
##         - user-defined value (float or integer)
Depolarization_value = "default"
```

5.2.2 Run in correlated k–distribution mode – Edition of kdis.py

The gas absorption is accounted for by applying the correlated-k technique (Lacis and Oinas, 1991) rather than a *line by line* computation which is very expensive in CPU time. This method is used to compute integrated radiative values (over the entire spectrum or an instrumental spectral response function).

In the ARTDECO package, we have three sets of k-distribution coefficients available, but any new parameterization can be added in the library (files are in ASCII format). See Section 6.2 for details about the kdis library files.

The k-distribution parameterization needs the edition of the following keywords in the secondary configuration file *kdis.py* :

- **K_distribution_parameterization** : Name of the k-distribution parameterization. The program will then read the file *kdis_<K_distribution_parameterization>_def.dat* in the library directory *./lib/kdis/reference/.*

- **Gas_list** : List of gas taken into account in the atmospheric absorption. The program will read the files *kdis_<K_distribution_parameterization>_<gas>.dat* for any gas of the Gas_list.

- **Gas_continuum_list** : List of gas whose continuum is taken into account in the run. Any gas of this list must be first in the Gas_list.

- **Instrumental_filters_list** : To specify instrumental spectral response functions (ISRF) to integrate the radiance and flux into. An empty list must appear if no such ISRF is to be used. Filters use is only possible in **kdis** mode. Corresponding files in *./lib/filter* will be needed. See in the Section 6.3 for the description of these files. Some default ISRF are already in the library :
    - parasol_1020
    - parasol_443
    - parasol_490
    - parasol_565
    - parasol_670
    - parasol_763
    - parasol_765
    - parasol_865
    - parasol_910

- **Wavelength_unit** : keywords to specify the unit of the specified wavelengths in input. Valid strings are :
    - **"cm-1"** or **"cm^-1"** for wavelengths in cm^{-1}.
    - **"microm"**, **"mum"** or **"micrometers"** for wavelengths in μm.

- **Wavelength_start :** *float* (recommended) or *int* specifying the lower limit of the spectral domain.

- **Wavelength_end :** *float* (recommended) or *int* specifying the upper limit of the spectral domain.

- **Depolarization_value** : Value (*float* or *int*) of the depolarization factor. It might also be the string **"default"**. In this case, the value is 0,0279 (following Young, 1980) in the V1.2.0 of ARTDECO.

Screenshot of the file kdis.py :

```
##############################################################################
##                                                                          ##
## Additional configuration file used to define specific parameters         ##
## for kdis mode of ARTDECO.                                                ##
##                                                                          ##
##############################################################################

## Name of the k-distribution parameterization. The program will
## then search for the file "./lib/kdis/kdis_<name>.dat".
K_distribution_parameterization = "gamesw_146"

## List of gas taken into account for the gaseous absorption.
## If no gas are taken into account, Gas_list = []
Gas_list = ["o2", "co2", "o3", "h2o"]
## List of gas continuum taken into account.
## If no continuum is taken into account, Gas_continuum_list = []
Gas_continuum_list = ["co2", "o3", "h2o"]

## Instrumental filters list. The program will search for the files
## "./lib/filter/filter_<name>.dat".
## example : Instrumental_filters_list = ["parasol565", "parasol670"]
Instrumental_filters_list = []

## Wavelength unit might be :
##         - "cm^-1" or "cm-1".
##         - "microm", "micrometers", "mum".
## Spectral domain limits "start" and "end" may be unordered, the program
## will deal with it.
Wavelength_unit = "cm-1"
Wavelength_start = 2500.
Wavelength_end = 15000.

## Depolarization value :
##         - "default" : value = 0.0279 for every wavelength.
##         - user-defined value (float or int).
Depolarization_value = "default"
```

5.2.3 Configuration of the discrete ordinate (Disort2.1) RTE solver – Edition of disort.py

- **Compute_radiance** :
  - **True** : Computes radiances and flux.
  - **False** : Computes only flux.

- **Print_downward_radiance** : The program will write the downward radiances in the output files. **True** or **False**.

- **Thermal_emission** : The solver takes into account thermal emission.

- **Thermal_only** : The incoming radiation at the top of the atmosphere is set to zero. **True** or **False**.

- **N_stream** : Number of Gauss points used for the zenithal integration (number of streams). Default value is 8.

- **Convergence_criterion** : Convergence criterion for azimuthal (Fourier cosine) series. DISORT will stop when the following occurs twice : largest term being added is less than Convergence_criterion multiplied by the total series sum. (Twice because there are cases where terms are anomalously small but azimuthal series has not converged.) Convergence_criterion should be between 0 and 0.01 to avoid risk of serious non-convergence. Default value is 0,0.

Screenshot or the file disort.py

```
##############################################################################
##                                                                          ##
## Additional configuration file used to define specific parameters         ##
## for DISORT model.                                                        ##
##                                                                          ##
##############################################################################

## Should radiances be computed ? Otherwise, only fluxes will be.
Compute_radiance = False

## Should the program write the downward radiances at the surface ?
Print_downward_radiance = False

## Is the thermal local emission taken into account ?
Thermal_emission = False

## Thermal local emission only ? In this case, the incoming radiation at
## the top of atmosphere is set to zero.
Thermal_only = False

##############################################################################
##                                                                          ##
## Advanced parameters.                                                     ##
##                                                                          ##
##############################################################################

## Number of streams (default: 8).
N_stream = 8

## Convergence criterion (default: 0.0).
## (it should be between 0.0 and 0.001).
Convergence_criterion = 0.000
```

5.2.4 Configuration of the 1D Monte Carlo RTE solver – Edition of mcrad1d.py

- **N_photons** : Total number of photons to be thrown in the Monte Carlo solver. Default value is 1e6.

- **Maximum_interactions** : Number of scattering order to account for. This option is reserved for testing runs. In order to keep an unlimited value, the user may indicate **-1** (default value).

- **Variance_reduction_methods** : The variance reduction optimal options method (Buras *et al.*, 2011) is activated ? **True** or **False** (default value).

- The following keywords are related to the use of the variance reduction optimal options method and the indicated default values are the ones given in Buras *et al.* (2011).

Screenshot of the file mcrad1d.py

```
################################################################################
##                                                                            ##
## Additional configuration file used to define specific parameters           ##
## for 1D Monte Carlo radiation model                                         ##
##                                                                            ##
################################################################################


################################################################################
##                                                                            ##
## Advanced parameters.                                                       ##
##                                                                            ##
################################################################################

## Number of photons (default: 1e6).
N_photons = 1.e6

## Maximum number of interactions (default: -1).
Maximum_interactions = -1

## Use of variance reduction methods (default: False).
Variance_reduction_methods = False

## Epsilon_ddis (default: 0.1).
Epsilon_ddis = 0.1

## Cloned photon specifications for N-tuple LE.
##     - mc-vrm-n-firstcp (default: 1).
##     - mc-vrm-n-sccp (default: 12).
##     - mc-vrm-n-lecp (default: 10).
Mc_vrm_n_firstcp = 1
Mc_vrm_n_sccp = 12
Mc_vrm_n_lecp = 10

## Critical weight for splitting (default: 3.0).
Critical_weight_splitting = 3.

## Maximum number of secondary photons / splitting (default: 5000)
Maximum_number_secondary_photons = 5000

## Critical weight for russian roulette (default: 0.3).
Critical_weight_russian_roulette = 0.3

## Minimum probability to kill a photon (default: 0.2).
Minimum_probability_photon = 0.2
```

5.2.5 Configuration of the Adding-doubling RTE solver – Edition of doad.py

- **Print_downward_radiance** : The program will write the computed downward radiances in output files. **True** or **False**.

- **N_stream** : Number of Gauss points used for the zenith integration. Default value is 24.

- **Adding_epsilon_accuracy** : Adding calculation accuracy in the zenith integration. Default value is 1e-5.

- **Maximum_fourier_terms** and **Minimum_fourier_terms** : Maximum and minimum number of Fourier terms used to achive the accuracy. Default values are 1000 and 10.

Screenshot of the file doad.py

```
##############################################################################
##                                                                          ##
## Additional configuration file used to define specific parameters         ##
## for Vesperini Adding-doubling model                                      ##
##                                                                          ##
##############################################################################

## Should the program write the downward radiances at the surface ?
Print_downward_radiance = False


##############################################################################
##                                                                          ##
## Advanced parameters.                                                     ##
##                                                                          ##
##############################################################################

## Number of Gauss points (default: 24).
N_stream = 24

## Accuracy of adding calculation epsilon (default: 1e-5).
Adding_epsilon_accuracy = 1e-5

## Absolute maximum number of Fourier terms (default: 1000).
Maximum_fourier_terms = 1000

## Minimum number of Fourier terms in the BDRF/BPDF Fourier expansion.
## (default: 10).
Minimum_fourier_terms = 10
```

5.2.6 Configuration of the atmospheric profile – Edition of atmosphere.py

- **Atmospheric_profile** : Name of the atmospheric profile. The program will then look for the file *atm_<Atmospheric_profile>.dat* in the directory *./lib/atm/reference/*.

- **Atmosphere_in_ppmv** : The gas vertical profiles are given in ppmv in the atmospheric profile file. **True** or **False**. If True, then the name of the atmospheric profile must end with the suffix "_ppmv".

- **Uniform_gas_list** : List of gas whose concentration is uniform throughout the entire profile. The value is then given in the same list (in ppmv). If a gas is indicated in the Uniform_gas_list, the program will ignore any value concerning the gas in the atmospheric profile file.

Screenshot of the file atmosphere.py

```
##############################################################################
##                                                                          ##
## Additional configuration file used to define specific parameters         ##
## for the atmospheric profile.                                             ##
##                                                                          ##
##############################################################################

## Name of the atmospheric profile. The program will then be read in
## "./lib/atm/atm_<name>.dat".
Atmospheric_profile = "tropic100_ppmv"

## Are the values of sampled concentration in ppmv ?
Atmosphere_in_ppmv = True

## List of species with a uniform concentration distribution
## which should not be read in the atmospheric profile file.
## Format : [[name1, value1], [name2, value2]] or [].
Uniform_gas_list = [["co2", 3.63e2],
                    ["o2", 2.09e5],
                    ]
```

5.2.7 Configuration of the particles properties – Edition of particles.py

- **Particle_option** : List of particles parameters. Each item of the list follows the same pattern :
  [particle name, use_betal flag, interp. flag, H-G flag, integrated opacity at 550 nm, vertical distribution type, parameters for the vertical distribution (0, 1 or 2 parameters)]

  - **particle name** : Name of the particle. It will give the name of the files that the program will read depending on the following flags.

  - use_betal flag :
    - **True** : The program will directly read the Legendre expansion coefficients. The corresponding file *betal_<particle_name>.dat* is required and must be stored in */lib/betal/*.
    - **False** : The program will compute these coefficients. If optical properties are available in *./lib/opt/opt_<particle_name>.dat*, it will read this file. Otherwise, it will compute them by reading microphysical properties file in *./lib/prop/prop_<particle_name>.dat* and refractive indices in *./lib/refind*.

  - Interp. flag : Optical properties read in *./lib/opt/opt_<particle_name>.dat* are interpolated between pre-existing ones (**True**) or only exact working wavelengths definition must be used (**False**). In the first case, optical properties will be read in the file *./lib/opt/opt_<particle_name>.dat*. In the

latter case, the program will compute exact values using microphysical properties files read in *./lib/prop/prop_<particle_name>.dat* and refractive indices read in *./lib/refind/refind_<particle_type>.dat*, where *<particle_type>* is defined in the file *prop_<particle_name>.dat* (for example, “lwat” for liquid water).

- H-G flag : The phase function can be defined with an Henyey-Greenstein approximation (only available when Stokes_components = 1 in main.py) using the asymmetry parameter read in *lib/opt/opt_<particle_name>.dat*. **True** or **False**.

- Integrated opacity at 550 nm.

- Vertical distribution type and related parameters :

  - **“layer”** : The particles are entirely located in a single atmospheric layer (defined between two altitude levels, z_{bottom} and z_{top}). Only one parameter is then expected : $\mathbf{z_{ptcle}}$ (km). The layer will be the one for which $z_{bottom} \leq z_{ptcle} < z_{top}$.

  - **“sh”** : “scale height” parameterization. This corresponds to an exponential decay of the particle density $\exp(-z/z_{sh})$. Only one parameter is then expected : $\mathbf{z_{sh}}$ (km).

  - **“gauss”** : The density of particles is normally distributed around an altitude z_0 with a standard deviation σ_z. Two parameters are then required : $\mathbf{z_0}$ and $\boldsymbol{\sigma_z}$ in km.

  - **“user”** : In that case the file *./lib/vdist/vdist_<particle name>.dat* describing the vertical distribution of the particle is read.

- **Truncation_method** : Method used to truncate the phase matrix.

  - **“none”** : no truncation.
  - **“potter”** : Potter truncation (Potter, 1970).
  - **“dm”** : δ-M method (Wiscombe, 1977).
  - **“dfit”** : δ-fit method (Hu *et al.*, 2000).

- **Number_of_betal** : Maximum number of expansion coefficients (*int*). In DOAD and DISORT RTE solvers, the phase matrix is described by Legendre coefficients while in SINSCA and MCRAD1D the tabulated phase matrix is required. In the latter case, if a truncation is performed, the phase matrix will then be recomposed to be used as an input of the RTE solver. If you set **Number_of_betal = -1** or **“optimized”** (default), ARTDECO will automatically set it to the optimized value regarding the number of quadrature points (so called number of streams) for the radiative transfer. If you call for TMS correction (see below), **Number_of_betal = -1** is required.

- **TMS_correction** : The intensity correction of Nakajima and Tanaka (1988) is applied. **True** or **False**.

- **Theta_min** and **Theta_max** : The two angles (*float* or *int*) used for the extrapolation of the phase function following Potter (1970). Used only if Truncation_method = “potter”.

- **Theta_cut** : When fitting the phase function, the weight of angles below Theta_cut (*float* or *int*) will be set to 0 so that the fit ignore them (see Hu et al., 2000). Used only if Truncation_method = “dfit”.

- **Fit_all** : After their renormalisation, the terms of the phase matrix other than the phase function can be either only expanded into Legendre polynomials (**True**) or fitted (**False**). Used only if Truncation_method = “dfit”.

Screenshot of the file particles.py

```
#############################################################################
##                                                                         ##
## Additional configuration file used to define specific parameters        ##
## for the definition of particles and clouds in ARTDECO.                  ##
##                                                                         ##
#############################################################################

## Definition of particles parameterization.
## Pattern:
## [particle name, use_betal flag, interp. flag, H-G flag,
##  integrated opacity at 550 nm, vertical distribution type,
##  parameters for the vertical distribution (0, 1 or 2 parameters)]
##
## particle name :
Particle_option = [
    #["aerosols_mono_10980-10990-cm-1", True, False, False, 0.5, "layer", 2.0],
    ["opac_soot", False, False, False, 0.05, "sh", 0.1],
    #["stephens_st2", False, False, True, 5.0, "layer", 0.05]
                  ]

## Truncation method for the phase matrix. Possibilities :
##     - "none"
##     - "potter"
##     - "dfit"
##     - "dm"
Truncation_method = "none"

## Number of expansion coefficients.
## If Number_of_betal = -1 or Number_of_betal = "optimized", ARTDECO will
## automatically set it to the optimized value regarding the number of
## quadrature points (so called number of streams) for the radiative transfer.
Number_of_betal = "optimized"

## Should the intensity correction (TMS) of Nakajima & Tanaka (1988)
## should be applied ?
## If so, the Number_of_betal should be set to "optimized".
TMS_correction = False


#############################################################################
##                                                                         ##
## Advanced parameters for the Potter truncation method.                   ##
##  (Only used if Truncation_method = "potter")                            ##
##                                                                         ##
#############################################################################

## Theta_min (default = 14.0)
Theta_min = 14.0

## Theta_max (default = 15.0)
Theta_max = 15.0


#############################################################################
##                                                                         ##
## Advanced parameters for the d-fit truncation method.                    ##
##  (Only used if Truncation_method = "dfit")                              ##
##                                                                         ##
#############################################################################

## Theta_cut (default = 0.0)
Theta_cut = 0.0

## Fit all (default = False) ?
Fit_all = False
```

5.2.8 Configuration of the surface parameters – Edition of surface.py

- **Surface_type** :
  - **"lambert"** : for a lambertian surface.
  - **"brdf"** : for a surface characterized with a bidirectional reflectance distribution function (BRDF) and eventually a bidirectional polarization distribution function (BPDF).

- **Surface_nature** :
  - **"cste"** : Only for lambertian surfaces. It allows to specify a constant albedo value with the following keyword **Surface_albedo**.
  - **"ocean"** : Only for a brdf surface type. Some additional keywords have to be edited in the file : Wind_speed, Ocean_salinity, Pigment_concentration and Shadowing.
  - A specific name (*string*) for example "forest". In this case the program will read the file *./lib/surface/surfalb_forest.dat* or *./lib/surface/surfbrdf_forest.dat*.

- **Surface_temperature** : Surface temperature (in K). It might be equal to the temperature in the lowest atmosepheric layer. In this case, the indicated value must be -1.

Screenshot of the file surface.py

```
##############################################################################
##                                                                          ##
## Additional configuration file used to define specific parameters         ##
## for the surface configuraiton in ARTDECO.                                ##
##                                                                          ##
##############################################################################

## Surface type ?
##      - "lambert".
##      - "brdf".
Surface_type = "brdf"

## Surface name ?
##      - "cste" for a constant lambertian albedo.
##      - "ocean" for an ocean brdf surface.
##      - "<name>", the properties will then be read in the file
##                "./lib/surface/surfalb_<name>.dat" for a lambertian surface or
##                "./lib/surface/surfbrdf_<name>.dat" for a brdf surface.
Surface_nature = "ocean"

## In the case of a lambertian surface with a constant albedo.
Surface_albedo = 0.2

## Surface temperature (K).
## -1 to take the same temperature as the lowest layer
##  in the atmospheric profile.
Surface_temperature = -1

###########################################
## Specific parameters for ocean surface.  ##
###########################################

## Wind speed in m/s.
Wind_speed = 5.0

## Salinity in ppt.
Ocean_salinity = 34.3

## Pigment concentration in mg/m^3.
Pigment_concentration = 0.0

## Is shadowing effect taken into account ? True or False.
Shadowing = False
```

5.2.9 Configuration of the geometric parameterization – Edition of geometrics.py

- **Incident_spectrum** : Only in kdis mode. Name of the incident spectrum parameterization. The program will search the file solrad_<Incident_spectrum>.dat in *./lib/solrad*.

- **Fbeam_user** : Only in mono mode. Whether the Fbeam value is specified by the user (**True** and the value is indicated with the keyword **Fbeam_value**). Otherwise (**False**), its default value is 1.

- **Solar_configuration_mode** : The way the solar configuration is specified :

    - **“angle”** : the solar configurations are directly defined by solar zenith angles defined in **Solar_zenith_angle_list** (list of *float* or *int*).

    - **“position”** : the solar configurations are defined by lat-lon positions and UT times defined in **Geographic_position** (list of items with the format [longitude1, latitude1, day1 of the year, time1 (decimal, UT)]). The program will then compute corresponding solar zenith angles.

- **View_zenith_angle_list** and **View_azimuthal_angle_list** define the observation geometry (both list of *int* or *float*).

Screenshot of the file geometrics.py

```
################################################################################
##                                                                            ##
## Additional configuration file used to define specific parameters           ##
## for the geometric configurations in ARTDECO.                               ##
##                                                                            ##
################################################################################

################################
## Incident radiation.        ##
################################

## In kdis mode : name of the incident spectrum characterization.
## The program will then read the file "./lib/solrad/solrad_<name>.dat"
Incident_spectrum = "kurudz_medium"

## In monochromatic mode, is the beam source specified by the user ?
## Otherwise its value will be set to 1.
Fbeam_user = False
## If True, which value ?
Fbeam_value = 3.141592653

################################
## Solar configuration.       ##
################################

## Solar configuration mode :
##      - "angle" : Solar zenith angles will be defined.
##      - "position" : lat-lon positions and time will be defined.
Solar_configuration_mode = "angle"

## if Solar_configuration_mode = "angle", list of solar
## zenith angles (in degree).
Solar_zenith_angle_list = [60.0,]

## if Solar_configuration_mode = "position", list of position-time
## with the format :
## Geographic_position = [
##              [longitude1, latitude1, day1 of the year, time1 (decimal, UT)],
##              [longitude2, latitude2, day2 of the year, time2 (decimal, UT)],
##              ...,
##              ]
Geographic_position = [
    [3.0, 50.6, 125, 12.0],
    [3.0, 50.6, 200, 12.0],
    ]

################################
## Observation geometry.      ##
################################

## List of view zenith angles (in degree).
View_zenith_angle_list = [30.0,]

## List of view azimuthal angles (in degree).
View_azimuthal_angle_list = [0.0, 15.0,]
```

6. Library files

6.1 Atmospheric profile files

These files are located in *.//lib/atm/reference/* with the name atm_<name>.dat or atm_<name>_ppmv.dat, <name> being the profile name defined in <configuration_directory>/atmosphere.py. They contain the atmosphere profile for pressure, temperature, absorbing gas mixing ratio if given in ppmv or air density and absorbing gas density if not. The altitude must be sorted in decreasing order. Note that for absorbing gas that are uniformly distributed in the atmosphere (for example CO_2), their mixing ratio in ppmv may be directly set in <configuration_directory>/atmosphere.py. In this case, if a value is also given in the atmospheric profile file, the program will ignore it to consider only the value given in <configuration_directory>/atmosphere.py.

In the atmospheric profile file, the number of levels, the number of absorbing gas and their name are specified. Then the profile themselves are given.

Important : level sampling in that files will constrain the layering of the atmosphere in the model : one layer between two sampled altitudes.

A set of files in .//*lib/atm/reference* is available for use:

- **McClathey :**
  - mcclatchey_tropic : For tropical profile.
  - mcclatchey_midsum : For mid latitude summer.
  - mcclatchey_midwin : For mid latitude winter.
  - mcclatchey_subsum : For sub arctic summer.
  - mcclatchey_subwin : For sub arctic winter.
- **AFGL :**
  - afgl_tropic : For tropical profile.
  - afgl_us62 : For US standard.
  - afgl_midsum : For mid latitude summer.
  - afgl_midwin : For mid latitude winter.
  - afgl_subsum : For sub arctic summer.
  - afgl_subwin : For sub arctic winter.

Screenshot of an atmospheric profile file :

```
# model: Midlatitude winter  mc clatchey
#
# number of altitude
 33
# number of sampled gas :
 2
# sampled gas :
 h2o
 o3
#   z (km)        P(z) (mb)        T(z) (K)      air (cm-3)     H2O (cm-3)      o3 (cm-3)
   100.00        3.000e-04         210.2       1.0394e+13      3.3456e+07      5.3948e+05
    70.00        4.670e-02         230.7       1.4657e+15      4.6839e+09      1.0790e+09
    50.00        6.820e-01         265.7       1.8613e+16      2.1077e+11      5.3948e+10
    45.00        1.290e+00         258.5       3.6191e+16      6.3567e+11      1.6310e+11
    40.00        2.530e+00         243.2       7.5355e+16      1.4386e+12      5.1439e+11
    35.00        5.180e+00         227.8       1.6472e+17      3.6802e+12      1.1542e+12
    30.00        1.110e+01         217.4       3.7064e+17      1.2044e+13      2.3838e+12
    25.00        2.430e+01         215.2       8.2111e+17      2.2416e+13      4.2657e+12
    24.00        2.860e+01         215.2       9.6121e+17      2.0074e+13      4.5166e+12
    23.00        3.340e+01         215.2       1.1256e+18      1.8066e+13      4.8930e+12
    22.00        3.910e+01         215.2       1.3175e+18      1.7063e+13      5.3948e+12
    21.00        4.580e+01         215.2       1.5426e+18      1.7063e+13      5.3948e+12
    20.00        5.370e+01         215.2       1.8064e+18      1.5055e+13      5.6458e+12
    19.00        6.280e+01         215.2       2.1141e+18      1.6394e+13      5.3948e+12
    18.00        7.350e+01         215.7       2.4696e+18      1.6728e+13      5.1439e+12
    17.00        8.610e+01         216.2       2.8853e+18      1.8736e+13      4.8930e+12
    16.00        1.007e+02         216.7       3.7833e+18      2.1412e+13      4.5166e+12
    15.00        1.178e+02         217.2       3.9288e+18      2.5427e+13      4.2657e+12
    14.00        1.378e+02         217.7       4.5857e+18      3.3456e+13      4.0148e+12
    13.00        1.610e+02         218.2       5.3465e+18      6.0221e+13      3.7638e+12
    12.00        1.882e+02         218.7       6.2342e+18      2.0074e+14      3.2620e+12
    11.00        2.199e+02         219.2       7.2673e+18      2.3085e+14      2.6347e+12
    10.00        2.568e+02         219.7       8.4647e+18      2.5092e+14      2.0074e+12
     9.00        2.992e+02         225.7       9.6018e+18      5.3530e+14      1.5055e+12
     8.00        3.473e+02         231.7       1.0855e+19      1.1710e+15      1.1292e+12
     7.00        4.016e+02         237.7       1.2236e+19      2.8438e+15      9.6605e+11
     6.00        4.627e+02         243.7       1.3749e+19      7.0258e+15      8.0295e+11
     5.00        5.313e+02         249.7       1.5406e+19      1.2713e+16      7.2768e+11
     4.00        6.081e+02         255.7       1.7216e+19      2.2081e+16      6.1476e+11
     3.00        6.938e+02         261.7       1.9187e+19      4.0148e+16      6.1476e+11
     2.00        7.897e+02         265.2       2.1557e+19      6.0221e+16      6.1476e+11
     1.00        8.973e+02         268.7       2.4155e+19      8.3641e+16      6.7749e+11
     0.00        1.018e+03         272.2       2.7045e+19      1.1710e+17      7.5277e+11
```

6.2 K-distribution files

In the ARTDECO package, we have three sets of k-distribution coefficients available, but any new parameterization can be added in the library (files are in ASCII format). The default k-distribution coefficients :

- the GAME coefficients from Dubuisson *et al*. (2004, 2005) for different resolutions. It is suited for radiances and flux to be integrated within instrumental spectral filters. Absorption coefficients are available for H_2O, CO_2, O_3, O_2, N_2O, CO, CH_4, N_2, NO_2, SO_2 and the continuum follows the CKD parameterization from Clough *et al.* (1989). Different resolutions are available in the library :

    - **gamesw_1** : Coefficients are given with a resolution of 10 cm^{-1} in the range [2500 – 50000 cm^{-1}] (e.g. 0.2 – 4.0 μm).

    - **gamesw_2** : Coefficients are given with a resolution of 20 cm^{-1} in the range [2500 – 50000 cm^{-1}] (e.g. 0.2 – 4.0 μm).

    - **gamesw_3** : Coefficients are given with a resolution of 10 cm^{-1} in the range [25000 – 50000 cm^{-1}] (e.g 0.4 – 4.0 μm), then with a resolution of 200 cm^{-1} in the range [2500 – 25000 cm^{-1}] (e.g 0.2 – 0.4 μm).

    - **gamesw_4** : Coefficients are given with a resolution of 100 cm^{-1} in the range [2500 – 14400 cm^{-1}] (e.g 0.694 – 4.0 μm), then with a resolution of 400 cm^{-1} in the range [14800 – 25200 cm^{-1}] (e.g 0.676 – 0.397 μm), since there is only continuum absorption in this range.

    - **gamesw_5** : Coefficients are given with a resolution of 100 cm^{-1} in the range [2500 – 14400 cm^{-1}] (e.g 0.694 – 4.0 μm), then with a resolution of 400 cm^{-1} in the range [14800 – 50000 cm^{-1}] (e.g 0.676 – 0.2 μm), since there is only continuum absorption in this range.

    - **gamelw_1** : Coefficients are given with a resolution of 20 cm^{-1} in the range [220 – 2500 cm^{-1}] (e.g. 4.0 – 45.45 μm).

    - **gamelw_2** : Coefficients are given with a resolution of 10 cm^{-1} in the range [20 – 2500 cm^{-1}] (e.g. 4.0 – 500.0 μm).

    - **gamelw_3** : Coefficients are given with a resolution of 20 cm^{-1} in the range [40 – 2500 cm^{-1}] (e.g. 4.0 – 250.0 μm).

    - **gamefull** : Coefficients are given with a resolution of 20 cm^{-1} in the range [220 – 2500 cm^{-1}] (4.0 – 45.45 μm), 10 cm^{-1} in the range [2500 – 25000 cm^{-1}] (0.4 – 4.0 μm) and 200 cm^{-1} in the range [25000 – 50000 cm^{-1}] (0.2 – 0.4 μm).

- **kato** : The coefficients from Kato *et al.* (1999) with a relatively low resolution. It is

well suited to compute flux integrated over the whole solar range [2500 – 41646 cm^{-1}] (0.24 – 4.0 μm).

- **metim** : Non-adjacent bands with low resolution in the range [758 – 23364 cm^{-1}] (0.428 – 13.19 μm).

- **3mi** : Non-adjacent bands with low resolution in the range [4651 – 25000 cm^{-1}] (0.4 – 2.15 μm).

6.3 Instrumental filter files

These files are located in *./lib/filter* and contain any instrumental spectral response function you want to integrate the radiative quantities into. The above example partly shows the *lib/filter/filter_parasol1020.dat* file:

Screenshot of parasol1020.dat

```
#
#  Transmission filter file
#  for use in ARTDECO
#
# nlambda
   120
#
# lamb(microns)    transmission
   0.98900     0.00000
   0.98950     0.00180
   0.99000     0.00240
   0.99050     0.00260
   0.99100     0.00230
   0.99150     0.00220
   0.99200     0.00240
   0.99250     0.00240
   0.99300     0.00230
   0.99350     0.00240
   0.99400     0.00290
   0.99450     0.00300
   0.99500     0.00270
   0.99550     0.00280
   0.99600     0.00350
   0.99650     0.00400
   0.99700     0.00410
   0.99750     0.00420
   0.99800     0.00460
   0.99850     0.00530
   0.99900     0.00660
   0.99950     0.00710
   1.00000     0.00640
```

The file contains the number of wavelengths sampled, the wavelengths and transmissions. Wavelengths must be sorted in increasing order. It is recommended to cut the *ISRF* range as close as possible to the filter bandwidth to lower the number of k-distribution band to be used and then lower the CPU demand.
A list of Instrumental spectral response function files are available in lib/filter :

- parasol1020
- parasol443
- parasol490
- parasol565
- parasol670
- parasol763
- parasol765
- parasol865
- parasol910

6.4 Particle microphysical properties files

These files are located in *./lib/prop* directory. They contain all information to compute new particle optical properties : bulk material, model to use and its parameters. If new optical properties are required for a particle named “example”, a file lib/prop/prop_example.dat is necessary. At the beginning of the file, you must first specify the bulk material for the particle:

```
#
# ARTDECO : File containig properties for new particles (aerosols or hygrosols)
# needed to obtain optical properties (file opt_*.dat)
#
# Material
lwat
```

For a material “lwat” (liquid water) to be used, a file lib/refind/refind_lwat.dat containing the complex refractive index is required.

Then five models are available:

mie3 for Mie (spherical) particles
phm for Pristine Hexagonal Monocrystal
rhm for Rough Hexagonal Monocrystal
ihm for Inhomogeneous Hexagonal Monocrystal
baum for the Baum & Co cirrus model. When using baum model, the material is ignored because the code will just interpolate in LUT of precomputed optical properties. Note that for Baum cirrus, no lib/opt/opt_type.dat will be written and interp. flag must be set to False in <configuration_directory>/particles.py. This is because the way we obtain this optical properties is already by interpolating into LUT.

Mie Model

If the **Mie** model is used, the following specifications are necessary :

```
# optical model
mie3
#######################################################
# INPUT FOR THE MEERHOFF MIE PROGRAM VERSION 3.0 size distribution
# truncation of Mie sum  (delta)
1.00E-12
# cutoff value of the size distribution (cutoff)
1.00E-06
# index size distribution        (idis)
0
# number of subintervals for r   (nsubr)
20
# number of Gauss points in sub  (ngaur)
500
# PAR1
20.000
# PAR2
0.0
# PAR3
0.0
#
#
#############################################################
#                       sixe distrib.
# idist                         PAR1        PAR2        PAR3
# ------------------------------------------------------------
#  0   no distribution          r           -           -
#  1   two parameter gamma      alpha       b           -
#  2   two parameter gamma      reff        veff        -
#  3   bimodal gamma            reff1       reff2       veff
#  4   log normal               rg          sigma       -
#  5   log normal               reff        veff        -
#  6   power law                alpha       rmin        rmax
#  7   modified gamma           alpha       rc          gamma
#  8   modified gamma           alpha       b           gamma
```

The Mie sum (for a given size bin, see Hansen and Travis (1974)) will stop when the difference between two successive terms is less than delta. The other parameters are related to the size distribution. Excepted for the power law size distribution, the boundary sizes r_{min} and r_{max} of n(r) will be computed so that $n(r_{min})$ < cutoff and $n(r_{max})$ < cutoff. The size distribution type is specified using the index idist, the number of subintervals for integration over r, “nsubr”, and the number of gauss points used per subintervals “ngaur” .

Hexagonal monocristal model

To use the model for inhomogeneous hexagonal monocrystal, **ihm**, the following parameters are needed after the bulk material definition :

```
# optical model
 ihm
# Number of "photons" for the simulation
 10000000
#####################################################
# Caracteristics of hexagonal monocrystal
#
# length of the particle (microns)
137.147
# radius of the circumscribed circle related to the hexagonal face (microns)
27.4290
# mean free path (microns)
15.0
# reff air bubbles (microns)
1.5
# veff air bubbles
0.05
```

with the length of the hexagonal monocrystal and radius of the circumscribed circle related to the hexagonal face (in µm). Then the inclusions (air bubbles) are specified by providing the mean free path between them (in µm) and the effective radius (in µm) and variance for their (gamma) size distribution.

Rough hexagonal monocrystal

To use the model for rough hexagonal monocrystal, **rhm** , the following parameters are needed after the bulk material definition:

```
# optical model
rhm
# Number of "photons" for the simulation
 10000000
#####################################################
# Caracteristics of hexagonal monocrystal
#
# length of the particle (microns)
137.147
# radius of the circumscribed circle related to the hexagonal face (microns)
27.4290
# Tilt angle
15.00
```

with the length of the hexagonal monocrystal and radius of the circumscribed circle related to the hexagonal face (in µm). The tilt angle for the roughness is then specified too.

Pristine hexagonal monocrystal

For the pristine hexagonal monocrystal, **phm** , only the size (with the length of the particle and the radius of the circumscribed circle) of the crystal is to be declared :

```
# optical model
 phm
# Number of "photons" for the simulation
 10000000
##################################################
# Caracteristics of hexagonal monocrystal
#
# length of the particle (microns)
 137.147
# radius of the circumscribed circle related to the hexagonal face (microns)
 27.4290
```

Baum & Co cirrus model

In order to use the Baum & Co cirrus model, the following parameters are needed after bulk material definition:

```
# optical model
  baum
#######################
#  type
 ghm
#  Effective diameter (microns)
 20.0
```

Three different types are available for use :

- ghm : for General Habit mixture
- asc : for aggregate solid column
- sc : for solid columns

Available optical properties files

A set of files lib/prop/prop_ptcle.dat is already available for use. The list of particles is:

- Liquid water clouds :
  - from Stephens , “Optical properties of eight water cloud types”, Melbourne : CSIRO , 1979 :
    - stephens_as Altostratus
    - stephens_cb Cumulominbus
    - stephens_cu Fair weather cumulus
    - stephens_ns Nimbostratus

        - stephens_sc1 Stratocumulus I
        - stephens_sc2 Stratocumulus II
        - stephens_st1 Stratus I
        - stephens_st2 Stratus II
    - from OPAC (Hess *et al.*, 1998) database:
        - opac_cumucoclean Continental cumulus clean
        - opac_cumucopollu Continental cumulus polluted
        - opac_cumuma Maritime cumulus
        - opac_fog Fog
        - opac_stratusco Continental stratus
        - opac_stratusma Maritime stratus

- Ice clouds
    - ihm_parasol Inhomogeneous Hexagonal Monocrystal, IHM, used for the "Radiation Budget and Clouds" inversion pipeline of PARASOL data.
    - rhm_parasol Rough Hexagonal Monocrystal , RHM, used for the "Radiation Budget and Clouds" inversion pipeline of PARASOL data.

- Aerosols
    - from OPAC (Hess et al., 1998) database :
        - opac_inso water-insoluble particles.
        - opac_waso## water-soluble particles. ## is the relative humidity. eight level of relative humidity are defined 00% (hence opac_waso00), 50%, 70%, 80%, 90%, 95%, 98% and 99%. For the different relative humidity either the size distribution and refractive index are different.
        - opac_soot Soot
        - opac_ssam## Sea salt in accumulation mode. ## is the relative humidity. eight level of relative humidity are defined 00% (hence opac_ssam00), 50%, 70%, 80%, 90%, 95%, 98% and 99%. For the different relative humidity either the size distribution and refractive index are different.
        - opac_sscm## Sea salt in coarse mode. ## is the relative humidity. eight level of relative humidity are defined 00% (opac_waso00), 50%, 70%, 80%, 90%, 95%, 98% and 99%. For the different relative humidity either the size distribution and refractive index are different.
        - opac_minm Mineral aerosols in nucleation mode.
        - opac_miam Mineral aerosols in accumulation mode.
        - opac_micm Mineral aerosols in coarse mode.
        - opac_mitr Mineral transported aerosols.
        - opac_suso## Sulfate droplets aerosols. ## is the relative humidity. eight level of relative humidity are defined 00% (opac_waso00), 50%, 70%, 80%, 90%, 95%, 98% and 99%. For the different relative humidity either the size distribution and refractive index are different.

  NB : Note that the optical properties computed with ARTDECO for OPAC particles may be slightly different regarding the one of Hess *et al*. (1998) since we do not

exactly apply the same size cut to the gamma distribution.

6.5 Refractive index files

It must be located in *./lib/refind* . It contains the real (n) and imaginary (k) parts of the complex refractive index (m=n-ik) of material as a function of the wavelength in micrometers. Wavelengths must be sorted in increasing order. For a material "example" the file must be ./lib/refind/refind_example.dat. Here we partly show lib/refind/refind_lwat.dat :

```
#
#     DEFINES WAVELENGTH DEPENDENT COMPLEX INDEX OF REFRACTION FOR WATER
#     ALLOWABLE WAVELENGTH RANGE EXTENDS FROM .2 MICRONS TO 1000.0 microns
#
#     ERIC A. SMITH
#     DEPT OF ATMOSPHERIC SCIENCE
#     COLORADO STATE UNIVERSITY
#     FORT COLLINS,CO  80523
#     TEL   303-491-8533
#
#     REFERENCES
#
#     0.2 UM - 0.69 UM
#
#     HALE,G., AND M. QUERRY,1972.
#     OPTICAL CONSTANTS OF WATER IN THE 200 NM TO 200 UM WAVELENGTH REGI
#     APPLIED OPTICS,12,3,555-563.
#
#     0.69 UM - 2.0 UM
#
#     PALMER,K.F., AND D. WILLIAMS,1974.
#     OPTICAL PROPERTIES OF WATER IN THE NEAR INFRARED.
#     JOURNAL OF THE OPTICAL SOCIETY OF AMERICA,64,8,1107-1110.
#
#     2.0 UM - 1000.0 UM
#
#     DOWNING,H.D., AND D. WILLIAMS,1975.
#     OPTICAL CONSTANTS OF WATER IN THE INFRARED.
#     JOURNAL OF GEOPHYSICAL REVIEW,80,12,1656-1661.
#
#  nlambda =
 518
#    lambda          real              im
#   (microns)
  0.20000000       1.3960000      1.10000002E-07
  0.22499999       1.3730000      4.90000005E-08
  0.25000000       1.3620000      3.39999993E-08
  0.27500001       1.3540000      2.40000002E-08
  0.30000001       1.3490000      1.59999995E-08
  0.32499999       1.3460000      1.10000000E-08
  0.35001001       1.3430001      6.50000009E-09
  0.37500000       1.3410000      3.50000007E-09
  0.40000001       1.3390000      1.89999994E-09
  0.42501000       1.3380001      1.30000000E-09
  0.44999999       1.3370000      9.99999972E-10
  0.47499001       1.3360000      9.39999967E-10
  0.50000000       1.3350000      9.99999972E-10
  0.52499002       1.3340000      1.30000000E-09
  0.54999000       1.3329999      1.99999994E-09
```

A set of lib/refind/refind_material.dat files is available. The list of material is:

- lwat liquid water ($0.2 \leq \lambda \leq 1000$ µm)
- iwat ice water ($0.045 \leq \lambda \leq 167$ µm)
- refractive index taken from OPAC ($0.25 \leq \lambda \leq 40$ µm, Hess *et al.*, 1998) :
    - opac_lwat for liquid water
    - opac_inso for insoluble. It is for soil particles with a certain amount of organic material)
    - opac_waso## for water soluble. It consists of various kinds of sulfates, nitrates, and other, also organic, water-soluble substances). ## is the value of relative humidity. Eight values are available: 0%, 50%, 70%, 80%, 90%, 95%, 98%, and 99%.
    - opac_soot for soot used to represent absorbing black carbon.
    - opac_seas## for sea salt. It consist of the various kinds of salt contained in seawater. ## is the value of relative humidity. Eight values are available: 0%, 50%, 70%, 80%, 90%, 95%, 98%, and 99%.
    - opac_mine for mineral. It consists of a mixture of quartz and clay minerals.
    - opac_sulf## for sulfates. ## is the value of relative humidity. Eight values are available: 0%, 50%, 70%, 80%, 90%, 95%, 98%, and 99%.

6.6 Optical properties

Optical properties files are located in *./lib/opt*. The material (corresponding to a refractive index lib/refind/refind_<material>.dat) and model used to compute it are recalled. In case you provide files computed without ARTDECO, you may indicate “unknown” for that two items. For atmospheric particles, only six elements are needed to describe the *Mueller* matrix. Indeed, the four upper right and the four lower left elements are zero, and $F^{21}=F^{12}$, $F^{34}=-F^{43}$. For spherical particles, $F^{11}=F^{22}$ and $F^{33}=F^{44}$, so that only four elements are needed. In optical property files, only the needed elements are stored and the number of these elements is written at the beginning of the file. You first find a list of the sampled wavelengths together with the corresponding extinction cross section (μm^2), single scattering albedo and asymmetry parameter. Wavelengths must be sorted in increasing order. Then, the phase matrix is given with the $\mu = \cos(\theta_{scat})$ sorted in increasing order.
When ARTDECO computes new optical properties for a given particle, it may access pre-existing file and add the new optical properties.

For example, we partially show below the file lib/opt/opt_opac_waso70.dat that contains one sampled wavelength:

```
#
# Optical properties to be used by ARTDECO
#
# Used model to obtain that properties is:
 mie3
# Used material is:
 opac_waso70
# Number of phase matrix elements :
           4
# number of wavelengths
           1
#  wlambda(microns)   nteta   Cext (microns^2)       SSA              g
     0.550000000E+00    901     0.790283684E-02     0.981127676E+00    -0.327680000E+05
# Phase matrix
# lambda =      0.5500
#         u                F11              F44               F21               F34
    -0.100000000E+01     0.206130131E+00    -0.206130131E+00     0.000000000E+00     0.000000000E+00
    -0.999996426E+00     0.206116496E+00    -0.206116493E+00     0.679575075E-05    -0.466031055E-05
    -0.999981170E+00     0.206058383E+00    -0.206058294E+00     0.357900084E-04    -0.244832825E-04
    -0.999953722E+00     0.205954221E+00    -0.205953687E+00     0.878860518E-04    -0.598547963E-04
    -0.999914078E+00     0.205804649E+00    -0.205802812E+00     0.162981109E-03    -0.110285844E-03
    -0.999862238E+00     0.205610614E+00    -0.205605906E+00     0.260911855E-03    -0.175065442E-03
    -0.999798200E+00     0.205373341E+00    -0.205363278E+00     0.381463992E-03    -0.253273209E-03
    -0.999721967E+00     0.205094324E+00    -0.205075309E+00     0.524373743E-03    -0.343788146E-03
    -0.999633539E+00     0.204775311E+00    -0.204742451E+00     0.689328792E-03    -0.445298675E-03
    -0.999532918E+00     0.204418290E+00    -0.204365223E+00     0.875969253E-03    -0.556314267E-03
    -0.999420104E+00     0.204025468E+00    -0.203944214E+00     0.108388875E-02    -0.675178569E-03
    -0.999295099E+00     0.203599255E+00    -0.203480078E+00     0.131263562E-02    -0.800083924E-03
    -0.999157904E+00     0.203142243E+00    -0.202973533E+00     0.156171426E-02    -0.929087136E-03
    -0.999008521E+00     0.202657184E+00    -0.202425358E+00     0.183058657E-02    -0.106012630E-02
    -0.998846952E+00     0.202146965E+00    -0.201836392E+00     0.211867358E-02    -0.119103858E-02
    -0.998673199E+00     0.201614584E+00    -0.201207531E+00     0.242535711E-02    -0.131957866E-02
    -0.998487263E+00     0.201063126E+00    -0.200539728E+00     0.274998159E-02    -0.144343778E-02
    -0.998289148E+00     0.200495740E+00    -0.199833985E+00     0.309185602E-02    -0.156026314E-02
```

Note that when the Henyey-Greenstein, **H-G** approximation is turned on in <configuration_directory>/particles.py, the code will skip the phase matrix reading. It will generate a phase function from the asymmetry parameter. This option is only available with Stokes_components = 1 (main.py).

6.7 Legendre expansion coefficients

They are located in *./lib/betal*.

Here we partly show *lib/betal/betal_example.dat* as an example:

```
#
# ARTDECO : Beta_l
#
# Used model for truncation was:
 dm
# number of wavelengths
         2
# wlambda(microns)     nBetal     Trunc. Coeff.     Cext (microns^2)       SSA             g
     0.251263000E+00      8     0.508488047E+00     0.653685697E+03     0.999985879E+00     0.868645573E+00
     0.550000000E+00      8     0.480565655E+00     0.637321376E+03     0.999999587E+00     0.863741726E+00
# Betal
# lambda =      0.2513
#      l     alpha1            alpha2            alpha3            alpha4            beta1             beta2
       0     0.999999774E+00     0.000000000E+00     0.000000000E+00     0.895168774E+00     0.000000000E+00     0.000000000E+00
       1     0.217741409E+01     0.000000000E+00     0.000000000E+00     0.227279772E+01     0.000000000E+00     0.000000000E+00
       2     0.289468074E+01     0.404311118E+01     0.377194314E+01     0.279290068E+01     0.272835438E-02    -0.301677351E-01
       3     0.231502459E+01     0.327297768E+01     0.331219990E+01     0.238841779E+01    -0.197163821E+00    -0.430759758E-02
       4     0.173617911E+01     0.238036882E+01     0.229228806E+01     0.177390621E+01    -0.457542813E-01    -0.466699991E-01
       5     0.122939731E+01     0.145375104E+01     0.132421215E+01     0.115806293E+01    -0.223803126E+00     0.432175663E-01
       6     0.622744466E+00     0.953182856E+00     0.993641902E+00     0.700683344E+00     0.384273178E-01     0.189338514E-01
       7     0.577007384E+00     0.476836799E+00     0.349381900E+00     0.472380411E+00    -0.789707452E-01     0.228171962E-01
       8     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00
# lambda =       0.550
#      l     alpha1            alpha2            alpha3            alpha4            beta1             beta2
       0     0.999999984E+00     0.000000000E+00     0.000000000E+00     0.903787135E+00     0.000000000E+00     0.000000000E+00
       1     0.220901328E+01     0.000000000E+00     0.000000000E+00     0.225951570E+01     0.000000000E+00     0.000000000E+00
       2     0.298855560E+01     0.404768899E+01     0.382830768E+01     0.288690832E+01    -0.306331414E-03    -0.944569317E-02
       3     0.254273330E+01     0.343372230E+01     0.345825293E+01     0.260386787E+01    -0.166450507E+00     0.313437498E-01
       4     0.198493415E+01     0.268174531E+01     0.258081162E+01     0.201087799E+01     0.227508382E-02    -0.262263231E-01
       5     0.138877811E+01     0.166780716E+01     0.154938140E+01     0.135561026E+01    -0.212893510E+00     0.191553956E-01
       6     0.678606356E+00     0.108459105E+01     0.105656962E+01     0.712493104E+00    -0.249297110E-01    -0.196222754E-01
       7     0.359726979E+00     0.333227561E+00     0.231338343E+00     0.299544858E+00    -0.292606854E-01    -0.275115126E-01
       8     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00     0.000000000E+00
```

Note that the Legendre coefficients of the phase matrix expansion at the reference wavelength of 0.550 µm must be provided.
If you want to set a new file containing the legendre coefficients of the phase matrix, you may follow the file format as in already existing file.

6.8 Land surface specification

Lambertian surfaces

Located in *./lib/surface*, these files contain the spectral albedo for lambertian surface. Their name must start with the prefix surfalb_. Wavelengths must be sorted in increasing order. Below we partly show the file containing the albedo for ice, lib/surface/surfalb_ice.dat as an example.

```
#
# Surface albedo : Savannah
#
# Number of sampled values :
 208
# wavelength        albedo
#  (microns)
    0.20000000    0.00900000
    0.20242915    0.00900000
    0.20408164    0.00900000
    0.20576131    0.00900000
    0.20746888    0.00900000
    0.20920502    0.00900000
    0.21097046    0.00900000
    0.21276596    0.00900000
    0.21459228    0.00900000
    0.21645021    0.00900000
    0.21834061    0.00900000
    0.22026432    0.00900000
    0.22222222    0.00900000
    0.22421524    0.00900000
    0.22624435    0.00900000
```

ARTDECO contains a library of surface albedo for the range λ ~ 0.2 - 2.5 µm :

- alb_black
- alb_bog
- alb_conifer
- alb_ice
- alb_pasture
- alb_sand
- alb_savannah
- alb_snow
- alb_veg : for generic vegetation surfaces
- alb_white

BRDF surfaces

Located in *./lib/surface*, these files contain the parameterization for land surfaces defined with a BRDF/BPDF. Their name must start with the prefix surfbrdf_ . For each sampled

wavelength, 3 parameters are needed for the Li-Ross BRDF model and the parameter C_v and the refractive index for the Maignan BPDF model. Wavelengths must be sorted in increasing order.

```
#
lisp_ross
#
2
#
1.0 0.04 0.045342 0.016312 5.0 1.5 0.0
2.0 0.005 0.098 0.02 5.0 1.5 0.0
#
```

6.9 Particle vertical distribution specification

When using the option “user” in Particle_option in particles.py, the program needs input files to specify the vertical distribution for the corresponding particles. These files should be in *./lib/vdist* and follow the pattern :

```
#
# The name of the file should be vdist_<name_of_the_particle>.dat
#  Altitude distribution of particles
#
# number of altitude sampled :
  5
#
#    Alt (km)    concentration (arbitrary unit)
1.2e+02 0.0e+00
2.001 0.0
1.50000e+00     1.00000e+00
0.999 0.0
0.0e+00 0.0e+00
```

Altitudes must be in decreasing order, and at least the top and bottom altitudes of the atmosphere are needed. Concentration may be given in arbitrary units since there is a normalization process in the program. Concentrations between two altitudes are interpolated as well.

6.10 Incoming radiation flux

Located in *./lib/solrad, they* contain the flux of the incoming radiation flux across a

surface perpendicular to the beam propagation. The wavelengths must be sorted in increasing order. The spectral flux files are named with the prefix solrad_. It should correspond to a mean over a year since it can further be weighted as a function of the day of the year you use it for. Below, we show as an example the file solrad_kurudz_medium.dat for "kurudz_medium" spectral flux:

```
#
# Monochromatic solar spectrum
# Integrated value (W m-2) :
1373.097
# Number of sampled values :
54154
#
# lambda (microns)         Spectr. (W m-2 microns-1)
1.84501845e-01  2.4997646e+00
1.84505249e-01  2.6424476e+00
1.84508653e-01  2.8278747e+00
1.84512058e-01  3.0837513e+00
1.84515462e-01  3.0540998e+00
1.84518867e-01  2.2190739e+00
1.84522272e-01  2.4802355e+00
1.84525677e-01  2.9392399e+00
1.84529082e-01  3.3050928e+00
1.84532487e-01  3.4320339e+00
1.84535892e-01  3.2835852e+00
1.84539298e-01  2.8868930e+00
1.84542703e-01  2.5499616e+00
1.84546109e-01  2.1908132e+00
1.84549515e-01  1.9062358e+00
1.84552921e-01  1.6330624e+00
1.84556327e-01  1.0570680e+00
1.84559733e-01  8.9094292e-01
1.84563139e-01  1.1644794e+00
1.84566545e-01  1.6664578e+00
```

This spectral flux will be integrated into k-distribution spectral bands. A file containing the integrated spectral flux will be written for further use. For example, if you were using the gamesw_4 k-distribution and the spectral flux kurudz_medium, the file will be named lib/solrad/solrad_kdis_gamesw_4_kurudz_medium.dat :

```
#
# Solar spectrum integrated over k-distribution bands
#
# Solar constant (W m-2) :
    1373.097000000
# Number of sampled values :
         146
#
# lambda (microns)         Spectr. (W m-2)
 0.400026      0.979191505E+01
 0.406531      0.109776023E+02
 0.413251      0.120025995E+02
 0.420198      0.121818905E+02
 0.427382      0.112839154E+02
 0.434815      0.130950441E+02
 0.442513      0.144918369E+02
 0.450487      0.164717707E+02
 0.458754      0.171401315E+02
 0.467331      0.174141135E+02
 0.476234      0.184233118E+02
 0.485483      0.181658549E+02
 0.495098      0.192055901E+02
 0.505102      0.194955299E+02
 0.515519      0.194470732E+02
 0.526374      0.209723544E+02
 0.537697      0.215446309E+02
 0.549517      0.227024474E+02
 0.561869      0.231927956E+02
 0.574789      0.243679823E+02
 0.588317      0.249491977E+02
 0.602497      0.257070396E+02
 0.617378      0.258819051E+02
```

There are two available files of the integrated spectral flux:

- kurudz_medium
- kurudz_full

6.11 User-defined library files

It is possible in ARTDECO to copy and modify all the library files previously described, as long as the format remains the same as the one described in this section.

7. Output files

7.1 Radiances

Upward radiances at the top of the atmosphere are written in output for each view configuration. The name of the output file is **ARTDECO_Upward_radiances.xxx** (xxx may be dat, nc and/or hdf5 depending on the output options specified in main.py). If polarization is taken into account (Stokes_components = 3 or 4 in main.py), the differents parameters are in the same ASCII file with the names I, Q, U, V. If netCDF or HDF5 output files are written, there is one variable by parameter : I, Q, U and V, depending on the number of Stokes parameters in the computation.

In mono mode, it contains radiances in $W.m^{-2}.sr^{-1}$ (multiplied by the value of F_{beam} specified in <configuration_directory>/geometrics.py) for any wavelength and any geometry.

In kdis mode, the output file contains the radiances (in $W.m^{-2}.sr^{-1}$) integrated in each k-distribution band for each geometry.

If ISRF are used, an additional file will be written for radiances integrated within the instrument filter. For example for parasol865, an additional output file named **ARTDECO_Upward_radiance_parasol865.xxx** will be written.

If Print_downward_radiance=True in main.py, the downward radiance at the bottom of atmosphere is also printed out, with the same characteristics as upward radiances. The name of the output file will be **ARTDECO_Downward_radiance.xxx**.

7.2 Fluxes

Fluxes are only printed out when ARTDECO is configured in kdis mode.

Integrated fluxes :

Fluxes can either be integrated within an instrumental spectral response function (ISRF) if one is given in the secondary configuration file kdis.py, or over the spectral domain specified in kdis.py if no instrumental filter is specified. It is always given for any altitude sampled in the input atmosphere definition file. The direct downward, the total (e.g. direct + diffuse) downward and the upward flux are given in $W.m^{-2}$ for each solar zenith angle.

After a run with ISRF, for example for parasol865, the output file is named **ARTDECO_Integrated_flux_parasol865.xxx**. After a run without ISRF and integration of flux over a spectral domain, the output file is named **ARTDECO_Integrated_flux.xxx**.

Flux over kdis bands :

For each "k-distribution" band, the direct downward, the total (e.g. direct + diffuse)

downward and the total upward flux are given in W.m $^{-2}$ for each solar zenith angle at each altitude sampled in the input atmosphere definition file.

Output file is named **ARTDECO_Kdis_bands_flux.xxx**.

7.3 Warming rate and net radiative heating rate

When the code is used in kdis mode without instrumental filter, it will return the warming rate (in K.day^{-1}) and the net radiative heating rate (in W.m^{-2}.m^{-1}) for each atmospheric layer. It corresponds to the radiation taken between the boundaries of the layer.

The output files are respectively named **ARTDECO_Warming_rate.xxx** and **ARTDECO_Net_radiative_heating_rate.xxx**.

7.4 Legendre expansion coefficients

If Print_betal = True is specified in main.py, output files containing the Legendre coefficients of the phase matrix expansion, after truncation if any is used, will be written for each particle. They contain information about the truncation method, the truncation factor, the extinction coefficient, the single scattering albedo and the asymmetry parameter, as well as the coefficients for every wavelengths.

For the particle named “example”, the output files will be named **ARTDECO_Betal_example.xxx**.

7.5 Recomposed phase matrix

If Print_recomposed_phase_matrix = True is specified in main.py, output files containing the recomposed phase matrix from the Legendre coefficients are written for each particle. As well as the Legendre coefficients output files, they contain information about the truncation method, the truncation factor, the extinction coefficient, the single scattering albedo and the asymmetry parameter.

For the particle named “example”, the output files will be named **ARTDECO_Recomposed_phase_matrix_example.xxx**.

5.6 Incoming radiation flux

In kdis mode, an additional output file recalls the incoming radiation flux at the Top-of-atmosphere that was used for the computation. It contains the flux at each working wavelengths and eventually the correcting factor of the solar constant that was computed depending on the day of the year (if specified).

The ASCII output files will be named **ARTDECO_Incoming_radiation_flux.xxx**.

A.1 References

Buras, R and Mayer, B. 2011. Efficient unbiased variance reduction techniques for Monte Carlo simulations of radiative transfer in cloudy atmospheres: The solution. *J Quant Spectrosc Radiat Transf*, 112(3), **434-447**.

Clough, SA, Kneizys, FX and Davies, RW. 1989. Line shape and the water vapor continuum. *Atmos Res*, 23(3-4), **229-241.**

Compiègne, M. and C-Labonnote, L. and Dubuisson, P., 2013. The phase matrix truncation impact on polarized radiance . *AIP Conference Proceedings*, 1531, **95-98**.

Cornet, C., C-Labonnote, L. and Szczap, F. 2010. Three-dimensional polarized Monte Carlo atmospheric radiative transfer model (3DMCPOL): 3D effects on polarized visible reflectances of a cirrus cloud. *J Quant Spectrosc Radiat Transf*, 111(1), **174-186**.

de Haan, JF, Bosma, PB and Hovenier, JW. 1987. The adding method for multiple scattering calculations of polarized light, *A&A*, 183, **371-391.**

Dubuisson, P., Dessailly, D, Vesperini, M and Frouin, R. 2004. Water vapor retrieval over ocean using near-infrared radiometry. *J. Geophys. Res. Atmos.*, 109, **D19**.

Dubuisson, P., Giraud, V., Chomette, O, Chepfer, H and Pelon, J. 2005. Fast radiative transfer modeling for infrared imaging radiometry. J *Quant Spectrosc Radiat Transf*, 95(2), **201-220**.

Hansen, JE and Travis, LD. 1974. Light scattering in planetary atmospheres. *Space Science reviews*. 16(4), **527-610**.

Henyey Greenstein. 1941. Diffuse radiation in the galaxy. *Astrophysical journal,* 93, **70-83**.

Hess, M., Koepke, P. and Schult, I. 1998. Optical Properties of Aerosols and Clouds: The Software Package OPAC. *J. Atmos. Sciences,* 79, **5**.

Hu, YX, Wielicki, B, Lin, B, Gibson, G, Tsay, SC, Stamnes, K and Wong, T. 2000. δ-Fit: A fast and accurate treatment of particle scattering phase functions with weighted singular-value decomposition least-squares fitting. *J Quant Spectrosc Radiat Transf*, 65(4), **681-690**.

Kato, S., Ackerman, TP, Mather, JH and Clothiaux, EE. 1999. The k-distribution method and correlated-k approximation for a shortwave radiative transfer model. *J Quant Spectrosc Radiat Transf*, 62(1), **109-121.**

Kokhanovsky AA, Budak VP, Cornet C, Duan M, Emde C, Katsev IL, et al. 2010. Benchmark results in vector atmospheric radiative transfer. *J Quant Spectrosc Radiat Transf* , 111(12-13), **1931-1946**.

Nakajima, T. and Tanaka, M. 1988. Algorithms for radiative intensity calculations in moderately thick atmospheres using a truncation approximation. *J Quant Spectrosc Radiat Transf*, 40(1), **51-69.**

Potter, JF. 1970. The Delta Function Approximation in Radiative Transfer Theory. *J. Atmos. Sciences*, 37, **6**.

Stamnes K, Tsay, SC, Wiscombe, W. and Jayaweera, K. 1988. Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media. *Applied optics*, 27(12), **2502-2509.**

Stephens, L. Optical properties of eight water cloud types. Melbourne : CSIRO, 1979.

Wiscombe, WJ. 1977. The Delta–M Method: Rapid Yet Accurate Radiative Flux Calculations for Strongly Asymmetric Phase Functions. *J. Atmos. Sciences*, 34, **9.**

Young AT. 1980. Revised depolarization corrections for atmospheric extinction. *Applied optics*, 19(20), **3427-3428.**