Manual

GAME & ARTDECO: A Comprehensive Guide to execute automatically these Radiative Transfer Models

Cristina Gil Díaz[1]

[1] *CommSensLab, Dept of Signal Theory and Communications, Universitat Politècncia de Catalunya (UPC), Barcelona, 08034, Spain*

Email: cristina.gil.diaz@upc.edu

Contents

1. Introduction

This manuscript will detail the algorithms I have developed during my thesis to automate the execution of the radiative transfer models: GAME and ARTDECO models. These models simulate the radiative fluxes in each layer of the atmosphere which can be composed of gases, aerosols and/or cirrus clouds. Where aerosols and clouds are treated equally in both models.

2. Global Atmospheric ModEl (GAME)

The GAME radiative transfer model is designed to simulate the propagation of electromagnetic radiation through the Earth's atmosphere. It provides accurate estimations of radiative fluxes, at various atmospheric levels, by accounting for key processes such as absorption, scattering, and emission by atmospheric gases, aerosols, and clouds. GAME is capable of handling a wide range of wavelengths, from ultraviolet to infrared, making it a versatile tool for various applications, including climate research, satellite data analysis, and environmental monitoring. The model's flexibility allows users to adjust input parameters to reflect specific atmospheric conditions, ensuring tailored and precise simulations. In this section, we will describe the inputs that the model needs to work, the Matlab scripts developed to create the input files, the automated execution algorithm and the outputs that the model provides. In summary, the main features are:

- Calculates spectrally integrated, upward and downward radiative fluxes in plane and homogeneous layers.

- Accounts for thermal emission, absorption and scattering, as well as their interactions, using the Discrete Ordinates Method (DISORT) (Stamnes et al., 1988).

- Gaseous absorption (H2O, CO2, O3, N2O, CO, CH4 and N2) is treated from the correlated K-distribution (Lacis and Oinas, 1991).

- Considers aerosol scattering and absorption.

- Has a relatively good spectral resolution.

In the GAME model, the atmosphere is represented like in the Figure 1.

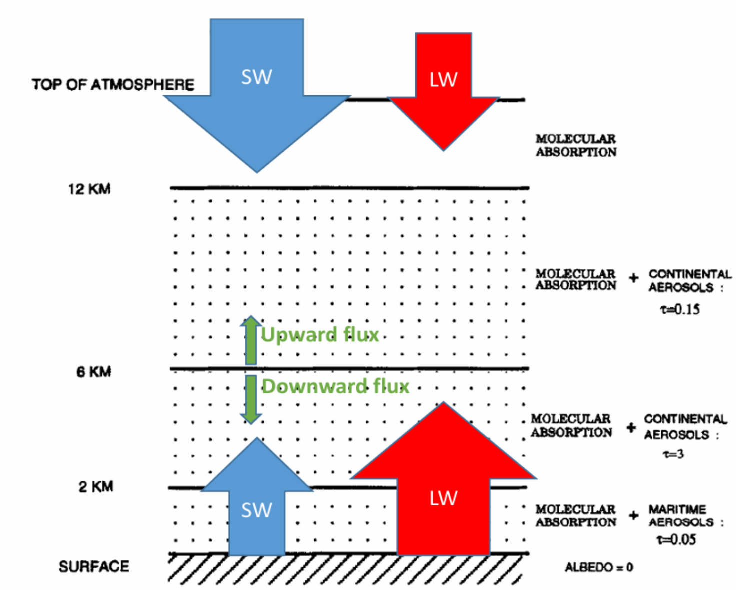


Figure 1. Scheme of the atmosphere in the GAME model.

From the upward and downward radiative fluxes in the shortwave (SW) and longwave (LW) spectra, the radiative direct effect of aerosols (only caused by absorption and scattering of radiation by aerosols an-s/or clouds) can be calculated at the bottom-of-the-atmosphere (BOA) and top-of-the-atmosphere (TOA), using the following equations:

$$BOA\ DRE = (F_{gas+aer}^{BOA}\downarrow - F_{gas+aer}^{BOA}\uparrow) - (F_{gas}^{BOA}\downarrow - F_{gas}^{BOA}\uparrow) \quad (1)$$

$$\begin{aligned} TOA\ DRE &= (F_{gas+aer}^{TOA} \downarrow - F_{gas+aer}^{TOA} \uparrow) - (F_{gas}^{TOA} \downarrow - F_{gas}^{TOA} \uparrow) \\ &= -(F_{gas+aer}^{TOA} \uparrow - F_{gas}^{TOA} \uparrow) \end{aligned} \quad (2)$$

If the direct radiative effect of the cirrus cloud is to be calculated, the radiative fluxes of the simulation that considers the cloud with gases and the simulation that considers only gases must be subtracted. With this sign criterion, a positive direct radiative effect indicates a warming of the atmosphere and a negative sign indicates a cooling of the atmosphere.

The characteristics of the simulations performed with the GAME model are shown in Table 1.

	Shortwave	Longwave
Wavelength range (μm)	Adjustable 0.2-4 (typical)	Adjustable 4-50 (typical)
Wavelength subbands	7	115 20 cm^{-1} (typical)
Vertical range (km)	0.005-20	0-100
Vertical layers	31	40

Table 1. Summary of the characteristics of the simulations performed with the GAME model.

The vertical and spectral distribution shown in Table 1 can be adjusted depending on the type of simulation being performed. For instance, if you are validating a pyranometer, you can adjust the spectral range of the simulation to the measuring range of that instrument. Alternatively, if you need to analyze an aerosol layer or cloud with a higher vertical resolution, additional vertical layers can be added into the model's vertical distribution.

2.1. Inputs

The inputs to the GAME model are:

- **Thermodynamic profiles of the atmosphere:** height (km), pressure (hPa), temperature (K), water vapor mixing ratio (ppmv) and ozone concentration (ppmv).
- **optical scattering properties of aerosols and/or clouds:** wavelength (μm), aerosol optical depth, extinction (m^{-1}), single scattering albedo and asymmetry factor.
- **Land surface properties:** albedo and temperature (K).

2.2. Matlab script for inputs

The Matlab script is called "Main_inputs_GAME.m" and has five default boolean configuration variables and two variables associated to the location of the atmospheric scene:

- **bool_cld:** it is true if clouds are considered in the simulation.
- **bool_aer:** it is true if aerosols are considered in the simulation.
- **bool_sw:** it is true if the simulation is made in the shortwave spectrum (0.2 - 4 μm).
- **bool_lw:** it is true if the simulation is made in the longwave spectrum (4 - 50 μm).
- **bool_mie:** it is true if the optical scattering properties for aerosols in the longwave spectrum are obtained from the MIE code.
- **lat_place:** latitude of the place where the atmospheric scene is located.
- **lon_place:** longitude of the place where the atmospheric scene is located.

In addition, an array is provided with the dates of the aerosol and/or cloud scenes to be simulated (v_date) and the geometrical properties of the clouds are read: base and top heights (v_cbh, v_cth). These properties can be read from data measured with the MPL or can be created manually. Remember that the properties of a cirrus clouds must follow the restrictions of (Gil-Díaz et al., 2024a). The cirrus clouds and aerosols must be vertically separated in the simulation, as it is not known how their optical scattering properties could be combined.

This script in turn is composed of six functions. The first function calculates the thermodynamic profiles of the atmosphere (profil_thermo_game.m). The second and third functions calculate the optical scattering properties of cirrus clouds and aerosols (prop_rad_cirrus_game.m and prop_rad_aer_game.m, respectively), the fourth function distributes vertically the optical scattering properties of the cirrus cloud and the aerosols (profil_rad_game.m), the fifth function calculates the surface properties (prop_surface_game.m) and the sixth function creates the main input file for the GAME model (files_in_game.m). A scheme of the structure of this script, with the database employed and input files created, is shown in Figure 2.

2.2.1. profil_thermo_game.m

First, the algorithm reads the vertical height distribution of the simulation spectrum, either shortwave or longwave. In GAME, the vertical distribution in the SW is usually given more resolution at low levels and reaches up to 20 km. In contrast, the resolution of the vertical distribution in the LW is lower and usually extends to 100 km. The files with the vertical distributions for the shortwave and longwave spectra are named: z_p_GAME_SW.txt and z_p_GAME_LW.txt, respectively. Second, the file containing the vertical distribution of ozone (CAMS_O3.nc) is read. For data download, see appendix D.1. The closest data to the place of the atmospheric scene is searched for and read. Subsequently, the ozone concentration units are converted from kg/kg to ppmv. Third, a file with the 1976 standard atmosphere (COESA, 1976) is read to fill in the atmospheric profiles at the heights where the radiosonde does not reach and where there is no observational ozone data. Fourth, the radiosonde for that date is read. Once all the data are read, the ozone concentration data are vertically interpolated with the radiosonde data. Then, the vertical profile of the water vapour mixing ratio is calculated and all variables are again interpolated with the heights of the vertical distribution of the model. Finally, the null values are replaced with values from the 1976 standard atmosphere and the thermodynamic input files of the model are written. All input files are located in the same folder named "GAME\INPUTS", except for the radiosondes which are located in a folder named "Dataset\Radiosoundings". Likewise, the output files are written to a folder named "GAME\OUTPUTS\SW" or "GAME\OUTPUTS\LW", for the shortwave and longwave spectrum, respectively. The location of the files can be changed by changing the paths.

The inputs of this function are: date (datenum), bool_sw, bool_lw, lat_place and lon_place. Where the last two variables indicate the location of the atmospheric scene. The files read by this function are: z_p_GAME_SW.txt, z_p_GAME_LW.txt, CAMS_O3.nc, thermo_profil_us67.txt and DORSBCNT0001_#. This function does not have any output, but creates the data file profil_thermo_case.dat, which is an input file of the GAME model. This input file has the structure shown in Figure 3.

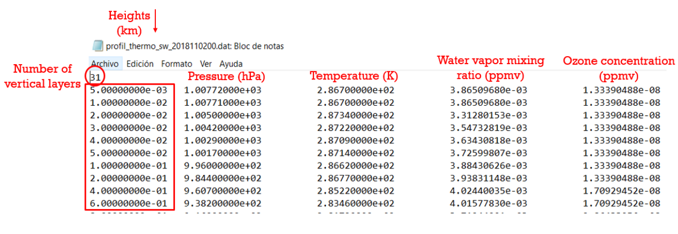


Figure 3. Example of structure of the data file: profil_thermo_case.dat, which is the GAME input file that contains the thermodynamical profiles.

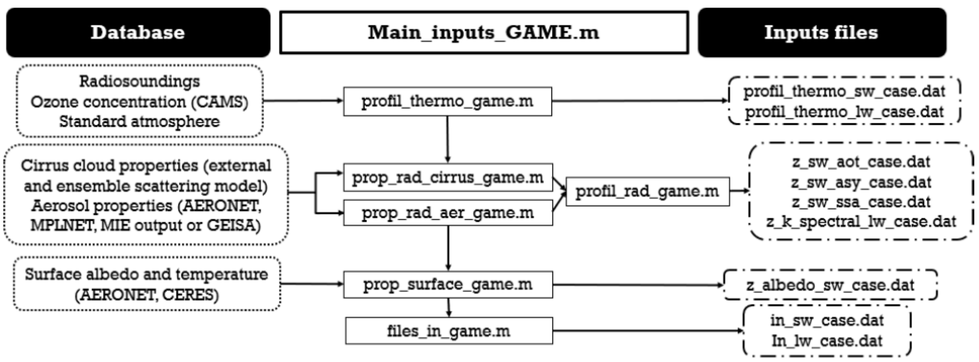


Figure 2. Scheme of the Matlab main script which creates the input files of the GAME model. The data albedo file is only created if the shortwave spectrum is considered.

2.2.2. prop_rad_cirrus_game.m

In case that the atmospheric scene contains an ice cloud, its optical scattering properties are calculated with the self-consistent scattering model for cirrus clouds, as I have done during part of my thesis (Gil-Diaz et al., 2024b). First, the wavelengths and coefficients of the parameterizations are read out. Second, the wavelengths are read from the GAME model. Third, the vertical profile of the ice water content in the whole cloud is calculated, after interpolating the extinction and temperature with the vertical distribution of the GAME model. Finally, the parameterizations are applied and the optical scattering properties of the cirrus cloud are calculated.

The inputs of this function are: cloud base height (cbh), cloud top height (cth), vertical distribution of lidar data (v_h_lidar), vertical distribution of temperature in cirrus cloud (v_temp_cld), vertical distribution of cloud extinction (v_ext_cld). The units of these inputs are: the heights in km, the temperature in K and the extinction in m^{-1}. The files read by this function are: lambda_Baran.txt, sca_coef_Baran.txt, abs_coef_Baran.txt, asym_coef_Baran.txt, lambda_GAME_SW.txt and lambda_GAME_LW.txt, z_p_GAME_LW.txt. The outputs of this functions are: matrices with the cloud extinction coefficient (m_ext_cld), asymmetry factor (m_asy_cld), single scattering albedo (m_ssa_cld), scattering (m_sc_cld), absorption (m_abs_cld) and the vertical distribution of the ice water content (v_iwc_cld). The vertical distribution of the cloud has been interpolated with a vertical distribution extending from 0 to 20 km, with 1 km resolution. The rows of the matrices indicate the height over which the cloud extends and the columns the GAME wavelength, in increasing order. Thus, the first row corresponds to 0 km and the first column to 0.2 μm by default. If the optical scattering properties of the clouds with a higher spatial resolution are desired, the user must change the function slightly.

2.2.3. prop_rad_aer_game.m

In case aerosols are considered in the atmospheric scene, this function is used to calculate their optical scattering properties from AERONET data. For data download, see Appendix D.2. First, the spectral and vertical distributions are read from the model in the shortwave and longwave spectra. Second, an AERONET file containing the variables: aerosol optical depth, single scattering albedo and asymmetry factor for the date of the atmospheric scene is read. If the time of the case is at night (between 6 a.m. and 6 p.m.), the aerosol properties of the previous days are averaged until there is sufficient data. The AERONET wavelengths are then interpolated with the GAME wavelengths using the Ängstrom exponent. Third, an aerosol file is read from MPLNET to analyze the vertical distribution of aerosols. In the case of time 00 UTC, in order to make temporal averages centred on that hour, the file of the previous day is read. In the case of no data, an equidistant vertical distribution up to 1 km height is assumed. Fourth, the aerosol properties in the shortwave spectrum are distributed over the entire atmospheric vertical profile. Fifth, the products of the MIE code or an external database like GEISA for continental aerosols are read with the optical scattering properties in the longwave spectrum and the properties are spectrally interpolated. Depending on the file containing data from an external database to be read, the Matlab function has to be modified accordingly.

The inputs of this function are: date (datenum), bool_lw and bool_mie. The files read by this function are: lambda_GAME_SW.txt, lambda_GAME_LW.txt, z_p_GAME_SW.txt, z_p_GAME_LW.txt, Barcelona_AERONET_INVERSION_L15.all, MPL#_V3_L15_AER_# and the aerosol optical scattering properties from GEISA database (opt_aerosol_lw.dat) or the output of the MIE algorithm (coef_vs_lambda_#.dat). The outputs of this functions are: matrices with the aerosol extinction coefficient (m_ext_aer_sw, m_ext_aer_lw), asymmetry factor (m_asy_aer_sw, m_asy_aer_lw) and single scattering albedo (m_ssa_aer_sw, m_ssa_aer_lw) in both spectra. The vertical distribution of the aerosols has been interpolated with a vertical distribution of each spectrum. The rows of the matrices indicate the height over which the aerosols extend and the columns the GAME wavelength, in increasing order. Thus, in the shortwave spectrum, the first row corresponds to 0.05 km and the first column to 0.2 μm by default.

2.2.4. profil_rad_game.m

This script combines the vertical profiles of the optical scattering properties of cirrus clouds and aerosols in the planetary boundary layer. First, the files with the vertical and spectral distributions are read. Second, the optical scattering properties of cirrus clouds and aerosols are combined, if both are considered in the simulation. Third, the aerosol optical depth in the shortwave spectrum is calculated from the extinction and the vertical height distribution. Fourth, the optical scattering properties of the files that will serve as inputs to the GAME model are written.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, m_ext_cld, m_asy_cld, m_ssa_cld, m_ext_aer_sw, m_ssa_aer_sw, m_asy_aer_sw, m_ext_aer_lw, m_ssa_aer_lw and m_asy_aer_lw. The units of the extinction is m^{-1}. The files read by this function are: lambda_GAME_SW.txt, lambda_GAME_LW.txt, z_p_GAME_SW.txt and z_p_GAME_LW.txt. This function does not have any output, but creates the GAME input files those contain the optical scattering properties of the atmospheric scene (z_aot_sw_case.dat, z_asy_sw_case.dat, z_ssa_sw_case.dat, z_k_spectral_lw_case.dat). If the simulation is performed in the shortwave spectrum, this Matlab function creates three GAME input files, independently of the vertical distribution of aerosols and cirrus clouds. Each of these files follows the same structure, as shown in Figure 4.

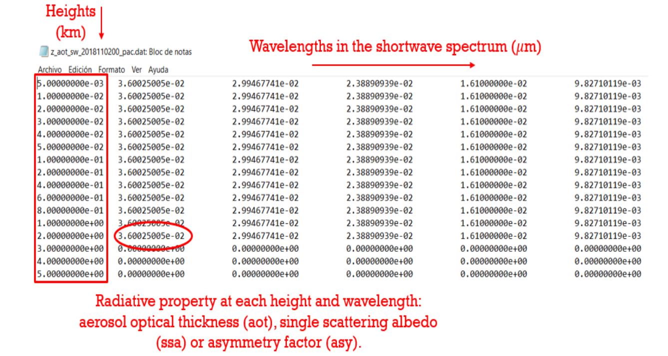


Figure 4. Example of structure of the data files: z_aot_sw_case.dat, z_asy_sw_case.dat, z_ssa_sw_case.dat, which are the GAME input files in the shortwave spectrum that contains the optical scattering properties.

In contrast, in the longwave spectrum, k input files are created with the optical scattering properties, one for each height. Its structure is shown in Figure 5.

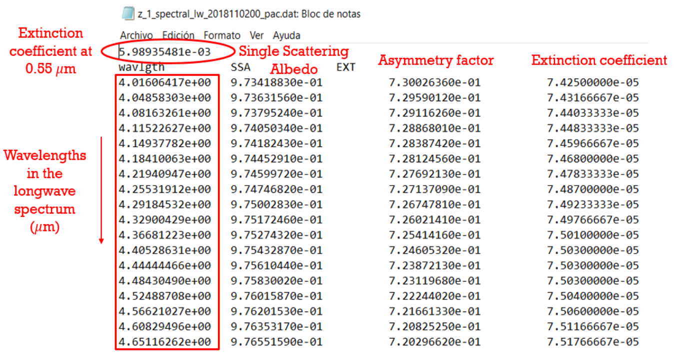


Figure 5. Example of structure of the data file: z_k_spectral_lw_case.dat, which is the GAME input files in the longwave spectrum that contains the optical scattering properties.

2.2.5. *prop_surface_game.m*

This script calculates the surface properties of the atmospheric scene. First, an AERONET file containing the albedo in the shortwave spectrum is read. Subsequently, a seasonal average of the season of the year in which the atmospheric scene is located is made and the albedo values are spectrally interpolated. It is established that for wavelengths longer than 1.02 μm, the surface albedo has no spectral dependence. Second, the albedo in the shortwave spectrum is writing in a file, Third, a NOAA20 file containing the surface emissivity and temperature are read. To calculate the surface albedo in the longwave, the following equation is used: 1 - emissivity. For data download, see Appendix D.3. The pixel closest to the location of the atmospheric scene is found and the data corresponding to that date is analyzed. Changing the location of the simulation can be done by modifying the variables lat_place and lon_place. By default, the surface albedo is set to 0.0157 and the surface temperature to 301.15 K.

The inputs of this function are: date (datenum), bool_sw, bool_lw, lat_place and lon_place. The files read by this function are: lambda_GAME_SW.txt, Barcelona_AERONET_INVERSION_L15.all, NOAA20.nc. The outputs of this function are the surface albedo and temperature in the longwave spectrum. Moreover, if the simulation is performed in the shortwave spectrum, this Matlab function creates a GAME input file that contains the surface albedo in the shortwave spectrum (z_albedo_sw_case.dat), whose structure is shown in Figure 6.

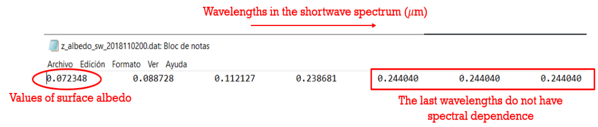


Figure 6. Example of structure of the data file: z_albedo_sw_case.dat, which is the GAME input file in the shortwave spectrum that contains the surface albedo.

As the surface albedo in the shortwave spectrum comes from AERONET and the spectral range of the example extends up to 4 μm, in order to have values in the larger wavelengths, the spectral dependence is removed.

2.2.6. *files_in_game.m*

This script creates the "IN" text files inputs to the GAME model. For this function to work properly, it is necessary that the previous functions had created the text files with the data: thermodynamic profiles, optical scattering properties and albedo if the shortwave spectrum is considered. In addition, it is necessary to adjust the read and write paths of the text files.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, albedo_lw, sst, lat_place and lon_place. Where the sst is the skin surface temperature in the longwave spectrum. The files read by this function are: lambda_GAME_SW.txt, lambda_GAME_LW.txt and the files z_sw_aot_case.dat, that contains the aerosol optical thickness at the 0.55 μm, which is the GAME work wavelength. This function does not have any output, but creates the GAME IN input files that contains the names of the input files, the vertical distribution of the atmospheric scene and several basic values of the simulation. The structure of the GAME IN input file in the shortwave spectrum is shown in Figure 7.

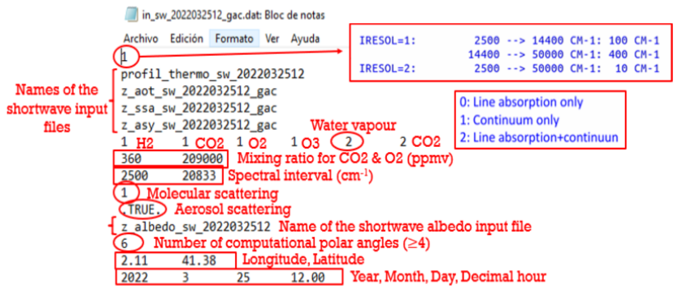


Figure 7. Example of structure of the data file: in_sw_case.dat, which is the GAME input file in the names of the input files, the vertical distribution of the atmospheric scene and several basic values of the simulation.

The structure of the GAME IN input file in the longwave spectrum is shown in Figure 8.

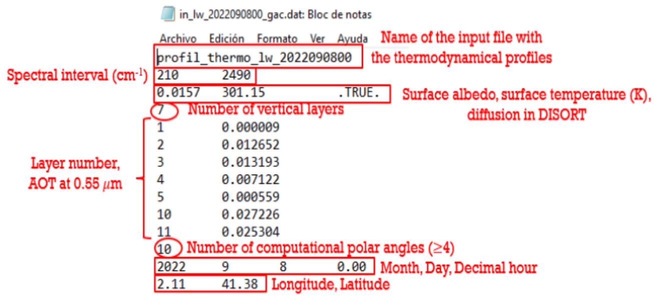


Figure 8. Example of structure of the data file: in_lw_case.dat, which is the GAME input file in the names of the input files, the vertical distribution of the atmospheric scene and several basic values of the simulation.

2.3. Execution

Once all the inputs for the GAME model are created, the text files are passed to CALCULA (see website https://tsc.upc.edu/en/it-services/computing-services) using the FileZilla program. Additionally, a text file called dates_GAME.txt, whose structure is shown in Figure 9.

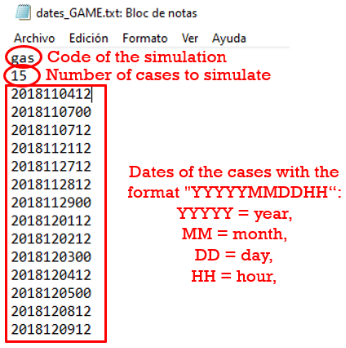
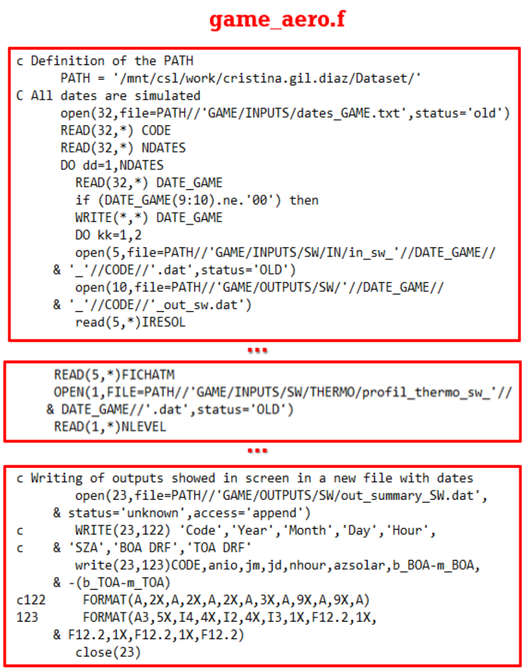


Figure 9. Example of structure of the text file: dates_GAME.txt, which contains the dates of the cases.

The first row is the code of the simulation, which indicates that type of atmospheric scene is: only gases (gas), only aerosols (aer), only a cirrus cloud (cld) and gases, aerosols and a cirrus cloud (gac). The second row is the number of cases to be executed and in the rest of the rows, the date of each case with the format "YYYYYMMDDHH", where YYYYY is the year, MM the month, DD the day and HH the hour.

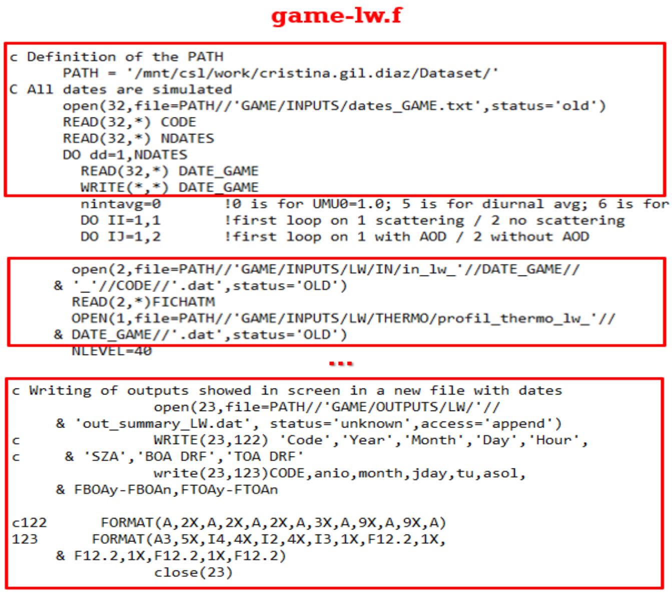
The main scripts of the GAME model have been modified to:
(1) Fix typos in the outputs files.
(2) Run cases in the longwave spectrum at nighttime.
(3) Automatically read several cases, one after the other.

An example of the modification of the GAME main scripts is shown in Figure 10 and Figure 11, for the shortwave and longwave spectra, respectively.




```
c Definition of the PATH
      PATH = '/mnt/csl/work/cristina.gil.diaz/Dataset/'
C All dates are simulated
      open(32,file=PATH//'GAME/INPUTS/dates_GAME.txt',status='old')
      READ(32,*) CODE
      READ(32,*) NDATES
      DO dd=1,NDATES
        READ(32,*) DATE_GAME
        if (DATE_GAME(9:10).ne.'00') then
        WRITE(*,*) DATE_GAME
        DO kk=1,2
        open(5,file=PATH//'GAME/INPUTS/SW/IN/in_sw_'//DATE_GAME//
     & '_'//CODE//'.dat',status='OLD')
        open(10,file=PATH//'GAME/OUTPUTS/SW/'//DATE_GAME//
     & '_'//CODE//'_out_sw.dat')
        read(5,*)IRESOL
...
        READ(5,*)FICHATM
        OPEN(1,FILE=PATH//'GAME/INPUTS/SW/THERMO/profil_thermo_sw_'//
     & DATE_GAME//'.dat',status='OLD')
        READ(1,*)NLEVEL
...
c Writing of outputs showed in screen in a new file with dates
        open(23,file=PATH//'GAME/OUTPUTS/SW/out_summary_SW.dat',
     & status='unknown',access='append')
c       WRITE(23,122) 'Code','Year','Month','Day','Hour',
c    & 'SZA','BOA DRF','TOA DRF'
        write(23,123)CODE,anio,jm,jd,nhour,azsolar,b_BOA-m_BOA,
     & -(b_TOA-m_TOA)
c122     FORMAT(A,2X,A,2X,A,2X,A,3X,A,9X,A,9X,A)
123      FORMAT(A3,5X,I4,4X,I2,4X,I3,1X,F12.2,1X,
     & F12.2,1X,F12.2,1X,F12.2)
        close(23)
```

Figure 10. Example of the modifications made in the GAME shortwave main script (game_aero.f).




```
c Definition of the PATH
      PATH = '/mnt/csl/work/cristina.gil.diaz/Dataset/'
C All dates are simulated
      open(32,file=PATH//'GAME/INPUTS/dates_GAME.txt',status='old')
      READ(32,*) CODE
      READ(32,*) NDATES
      DO dd=1,NDATES
        READ(32,*) DATE_GAME
        WRITE(*,*) DATE_GAME
        nintavg=0      !0 is for UMU0=1.0; 5 is for diurnal avg; 6 is for
        DO II=1,1      !first loop on 1 scattering / 2 no scattering
        DO IJ=1,2      !first loop on 1 with AOD / 2 without AOD

       open(2,file=PATH//'GAME/INPUTS/LW/IN/in_lw_'//DATE_GAME//
     & '_'//CODE//'.dat',status='OLD')
       READ(2,*)FICHATM
       OPEN(1,file=PATH//'GAME/INPUTS/LW/THERMO/profil_thermo_lw_'//
     & DATE_GAME//'.dat',status='OLD')
       NLEVEL=40
...
c Writing of outputs showed in screen in a new file with dates
          open(23,file=PATH//'GAME/OUTPUTS/LW/'//
     & 'out_summary_LW.dat', status='unknown',access='append')
c         WRITE(23,122) 'Code','Year','Month','Day','Hour',
c    & 'SZA','BOA DRF','TOA DRF'
          write(23,123)CODE,anio,month,jday,tu,asol,
     & FBOAy-FBOAn,FTOAy-FTOAn

c122      FORMAT(A,2X,A,2X,A,2X,A,3X,A,9X,A,9X,A)
123      FORMAT(A3,5X,I4,4X,I2,4X,I3,1X,F12.2,1X,
     & F12.2,1X,F12.2,1X,F12.2)
          close(23)
```

Figure 11. Example of the modifications made in the GAME longwave main script (game-lw.f).

Note that you will also must change the paths in the sun.f and sub.f files in the game-sw directory in order to be able to run the model in the shortwave spectrum.

If you do not have the GAME model installed in your Linux environment or in your CALCULA directory, 1) download the GAME model from the NextCloud path: Lidar < upclidar_Software < GAME < GAME, 2) transfer the GAME model to your Linux environment or CALCULA directory, 3) place it in the game-sw directory with the cd command, 4) compile the model with the command make clean < make, 5) place it in the game-lw directory with the cd command, 6) compile the model with the command . /compil game-lw, 7) place it in the mie directory with the cd command, 8) compile the model with the command compil < mie_net_LW.

Once the GAME model has been installed, it can be run with the execute_GAME.sh file, either in the shortwave or longwave spectrum. To run the bash file, it is necessary to change the path where the location of the model is indicated. The structure of the execution file is shown in Figure 12.

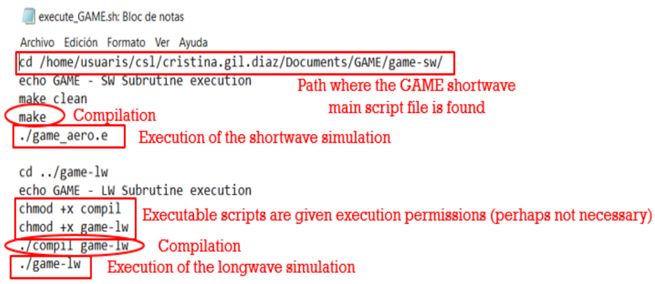


Figure 12. Example of structure of the bash file: execute_GAME.sh, which executes the GAME model automatically.

Once the files necessary to run the cases automatically have been explained, the execution mechanism will be detailed step by step. First, the GAME model is installed if it is not already installed in your CALCULA directory. To do this, download the model found in the Nextcloud lidar group folder (Lidar < upclidar_Software < GAME < GAME). Secondly, you transfer the model to your CALCULA directory via Filezilla. For the connection to the CALCULA server with Filezilla, please refer to the manual provided by UPC (https://tsc.upc.edu/en/it-services/computing-services/research-groups-specifications/csl-commsenslab-group). If you do not have access to CALCULA, you should write to the IT department. Thirdly, you pass also the input files from the local to the CALCULA server and adapt the execution files to the simulations you would like to do. Finally, you run the execute_GAME.sh file in a Linux terminal on the server. To access the Linux terminal, go to Dashboard - CALCULA OnDemand (https://ondemand.tsc.upc.edu) and open a CALCULA Interactive session. Within the session, go to System < LXTerminal and click. Once the session has been opened, you must go to the folder where the GAME model is located with the cd command and execute the file execute_GAME.sh that you will find in the same directory. As an example, Figure 13 shows a screenshot of the terminal.

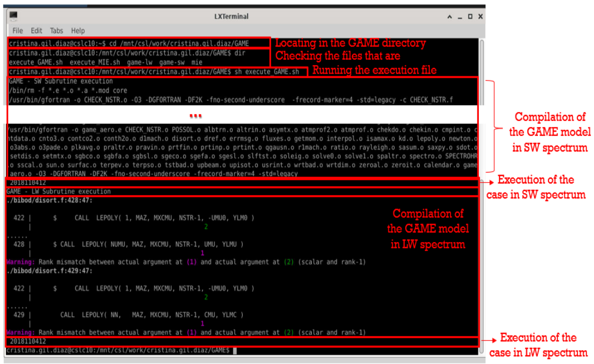


Figure 13. Example of execution of the GAME model.

If a 'strange' error occurs during compilation, it may be because the model's environment is not suitable. Make sure that the environment has the following dependencies:
- libgfortran5
- gfortran
- build-essential

In case of an error in the execution of the GAME model not associated with the inputs, I recommend that you ask the IT service of the department.

2.4. Outputs

The GAME model creates a data file for each simulation run on each spectrum with the radiative fluxes at each vertical layer. Both output files are identical. The structure of the output files is shown in Figure 14.

ALTITUDE (KM)	PRESSION (hPa)	FLUX DW (W/M2)	FLUX UP (W/M2)	DIR FLUX (W/M2)	NET FLUX (W/M2)	HEAT RATE (K/DAY)
0.00	1008.03	563.95	93.21	514.55	-470.75	0.00
0.01	1008.02	563.95	93.21	514.55	-470.75	0.00
0.02	1008.01	563.95	93.21	514.55	-470.75	0.00
0.03	1003.12	564.24	93.46	514.91	-470.79	0.00
0.04	1003.11	564.24	93.46	514.91	-470.79	0.00
0.05	1001.90	564.32	93.52	515.01	-470.80	0.00
0.10	995.90	564.68	93.83	515.48	-470.85	0.00
0.20	984.30	565.39	94.43	516.40	-470.96	0.00
0.40	960.70	566.82	95.65	518.28	-471.17	0.00
0.60	937.70	568.21	96.83	520.14	-471.37	0.00
0.80	915.60	569.54	97.96	521.95	-471.57	0.00
1.00	894.10	570.83	99.07	523.73	-471.76	0.00
2.00	790.60	577.03	104.36	532.62	-472.67	0.00
3.00	697.40	582.62	109.12	541.15	-473.51	0.00
4.00	613.60	587.71	113.40	549.33	-474.31	0.00
5.00	538.00	592.32	117.29	557.14	-475.04	0.00
6.00	469.50	596.54	120.82	564.63	-475.72	0.00
7.00	408.10	600.35	124.00	571.72	-476.35	0.00
8.00	352.90	603.80	126.86	578.44	-476.95	0.00
9.00	303.60	606.91	129.42	584.75	-477.49	0.00
10.00	259.80	609.68	131.70	590.62	-477.99	0.00
11.00	221.80	612.10	133.67	595.95	-478.44	0.00
12.00	189.30	614.18	135.35	600.69	-478.83	0.00
13.00	161.90	615.94	136.76	604.84	-479.18	0.00
14.00	138.00	617.48	137.98	608.59	-479.50	0.00
15.00	117.80	618.80	139.01	611.86	-479.79	0.00
16.00	100.90	619.92	139.86	614.69	-480.06	0.00
17.00	86.20	620.91	140.59	617.23	-480.32	0.00
18.00	73.40	621.81	141.23	619.52	-480.58	0.00
19.00	62.50	622.63	141.76	621.56	-480.87	0.00
20.00	53.30	623.52	142.20	623.52	-481.31	0.00

Figure 14. Example of structure of GAME output file: case_out_sw.dat or case_out_lw.dat.

This output file is a text file containing the radiative fluxes on each spectrum, downward, upward, direct, net and the heating rate for each vertical layer considered. The header indicates the variable and its unit, being W/m^2 for the fluxes and K/day for the heating rate.

In addition, a summary file with the direct radiative effects of the atmospheric scene at bottom-of-the-atmosphere and top-of-the-atmosphere, together with the code, date and solar zenith angle (SZA) of the case are also provided by the GAME model. These output variables are stored in a pre-existing data file, called out_summary_SW.dat and out_summary_LW.dat, the structure of which is shown in Figure 15.

CODE	YEAR	MONTH	DAY	HOUR	SZA(º)	BOA DRF (W/M2)	TOA DRF (W/M2)
gas	2018	11	4	12.00	57.12	0.00	0.00
gac	2018	11	4	12.00	57.12	6.33	13.89
gac	2018	11	4	12.00	57.12	6.33	13.89
gas	2018	11	4	12.00	57.12	0.00	0.00
gas	2018	11	7	0.00	154.43	0.00	0.00
gas	2018	11	7	12.00	58.02	0.00	0.00
gas	2018	11	21	12.00	61.57	0.00	0.00
gas	2018	11	27	12.00	62.74	0.00	0.00
gas	2018	11	28	12.00	62.91	0.00	0.00
gas	2018	11	29	0.00	159.73	0.00	0.00
gas	2018	12	1	12.00	63.38	0.00	0.00

Figure 15. Example of structure of GAME output file: out_summary_SW.dat or out_summary_LW.dat.

3. MIE algorithm

The Mie scattering algorithm is a computational method used to describe the scattering, absorption, and extinction of electromagnetic waves by spherical particles. Based on the theory developed by Gustav Mie in 1908, this code provides an exact solution for Maxwell's equations when electromagnetic waves interact with homogeneous spheres, making it particularly useful for studying aerosols, cloud droplets, and other particulate matter in the atmosphere. Unlike Rayleigh scattering, which applies to particles much smaller than the wavelength of light, Mie scattering is applicable across a broader range of particle sizes, making it suitable for analyzing interactions with particles comparable to or larger than the wavelength. It is typically used for wavelengths between 0.2 and 40 μm. This code computes several key parameters, including the scattering coefficient, absorption coefficient, and asymmetry parameter, which are crucial for accurately modeling radiative transfer in the atmosphere. In combination with radiative transfer models like GAME and ARTDECO, the Mie scattering code provides the optical scattering properties of atmospheric aerosols in the longwave spectrum. In the following sections, we will describe the automated algorithm developed for the creation of inputs for the creation of inputs in Matlab and subsequent execution of the program in Fortran.

3.1. Inputs

The inputs to the MIE code come from AERONET and are:
- **Size distribution of aerosols:** includes the effective geometrical radius with its associated standard deviation and the concentration.
- **Spectral refractive index:** includes the real and imaginary parts of the refractive index in function of the wavelength. The refractive index needs to be in the longwave spectrum and not only in the shortwave. Otherwise, the algorithm would make a spectral extrapolation and the resulting optical scattering properties would probably not make much sense.

3.2. Matlab script for inputs

The Matlab script creates the input files for the MIE algorithm, which is called Main_inputs_MIE.m. First, the AERONET file is read with the necessary variables. Second, the properties closest to the date of the atmospheric scene are searched, if the time is daytime. Otherwise, properties from at least two days are averaged to obtain the nighttime properties of the aerosols. Finally, the properties are written to text files. It is convenient to add refractive index values to the longwave spectrum of the dominant aerosol in the atmospheric scene if one wants to obtain the optical scattering properties of the aerosols in that spectrum. To do so, it will be necessary to modify the Matlab script or to add the refractive index manually to the text file created by the Matlab script.

The input of this function is the date (datenum). The file read by this function is Barcelona_AERONET_INVERSION_L15.all. The outputs of this function are the size distribution of the aerosols (distribution-char_case.dat) and the spectral refractive index (n_vs_lambda_case.dat).The structure of both input files is shown in Figure 16 and Figure 17, respectively.

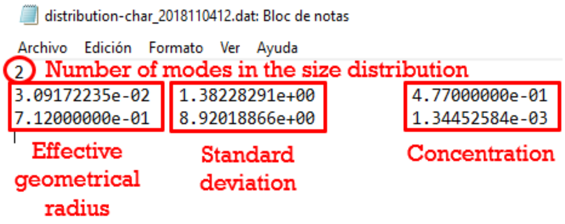


Figure 16. Example of structure of MIE input file: distribution-char_case.dat.

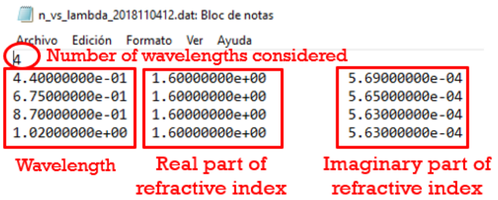


Figure 17. Example of structure of MIE input file: n_vs_lambda_case.dat.

3.3. Execution

Once all the inputs for the MIE algorithm are created, the text files are passed to CALCULA using the FileZilla program. Additionally, a text file is created containing in the first row the number of cases to be executed and in the rest of the rows, the date of each case with the format "YYYYYMMDDHH", where YYYYY is the year, MM the month, DD the day and HH the hour, as is shown in Figure 18.

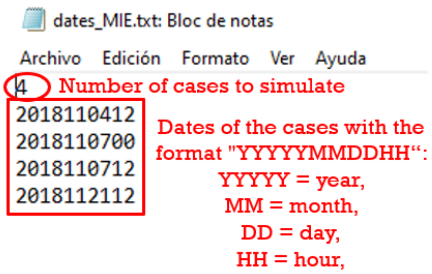


Figure 18. Example of structure of the text file: dates_MIE.txt, which contains the dates of the cases.

The main script of the MIE code has also been modified to read automatically several cases, one after the other. An example of the modifications made are shown in Figure 19.

mie_net_LW.f

```
c Definition of the PATH
      PATH = '/mnt/csl/work/cristina.gil.diaz/Dataset/'

C All dates are simulated
      open(32,file=PATH//'MIE/INPUTS/dates_MIE.txt',status='old')
      READ(32,*) NDATES

      DO dd=1,NDATES
        READ(32,*) DATE_GAME
        WRITE(*,*) DATE_GAME
c Initialization

      open (unit=99,file=PATH//'MIE/OUTPUTS/'//
     & 'coef_vs_lambda_'//DATE_GAME//'.dat',status='replace')
```

...

```
      open(1,file=PATH//'MIE/INPUTS/REFRACTIVE_INDEX/'
     & //'n_vs_lambda_'//DATE_GAME//'.dat')
      READ(1,*)nind
      DO i=1,nind
          READ(1,*)wawi(i),rni(i),rii(i)
      ENDDO
      close(1)

c Reading of characteristics of modal distribution (mean
c To change !!
      open(1,file=PATH//'MIE/INPUTS/DISTRIBUTION/'
     & //'distribution-char_'//DATE_GAME//'.dat')
          READ(1,*) AERTYPE
c         WRITE(*,*) 'Number of modes'
c         WRITE(*,*) AERTYPE
          DO i=1,AERTYPE
            READ(1,*)rmean(i),sig(i),Ni(i)
          ENDDO
          close(1)
```

Figure 19. Example of the modifications made in MIE algorithm (mie_net_LW.f).

As the MIE code is inside the GAME model, if you do not have already installed the GAME model in your CALCULA directory, see Section 2.3. Once the model has been downloaded and placed in the CALCULA directory, you have to pass also the input files from the local to the CALCULA server and adapt the file dates_MIE.txt. Finally, you run MIE code file in a Linux terminal on the server. To access the Linux terminal, go to Dashboard - CALCULA OnDemand (https://ondemand.tsc.upc.edu) and open a CALCULA Interactive session. Within the session, go to System < LXTerminal and click. Once the session has been opened, you must go to the folder where the MIE code is located with the cd command and execute the files compil < mie_net_LW that you will find in the same directory. As an example, Figure 20 shows a screenshot of the terminal.

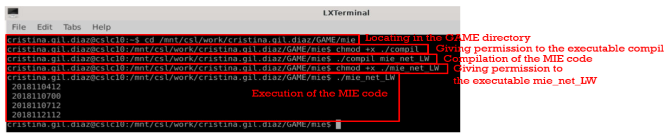


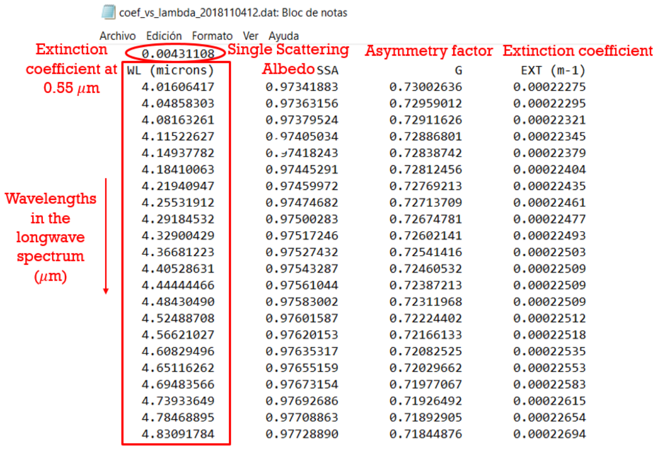
Figure 20. Example of execution of the GAME model.

If a ‘strange’ error occurs during compilation, it may be because the model’s environment is not suitable. Make sure that the environment has the following dependencies:
- libgfortran5
- gfortran
- build-essential

In case of an error in the execution of the MIE code not associated with the inputs, I recommend that you ask the IT service of the department.

3.4. Outputs

The MIE code creates a data file for each simulation with the optical scattering properties of the aerosols in the longwave spectrum. The structure of the output file is shown in Figure 21.



coef_vs_lambda_2018110412.dat: Bloc de notas

Archivo Edición Formato Ver Ayuda

0.00431108

WL (microns)	SSA	G	EXT (m-1)
4.01606417	0.97341883	0.73002636	0.00022275
4.04858303	0.97363156	0.72959012	0.00022295
4.08163261	0.97379524	0.72911626	0.00022321
4.11522627	0.97405034	0.72886801	0.00022345
4.14937782	0.97418243	0.72838742	0.00022379
4.18410063	0.97445291	0.72812456	0.00022404
4.21940947	0.97459972	0.72769213	0.00022435
4.25531912	0.97474682	0.72713709	0.00022461
4.29184532	0.97500283	0.72674781	0.00022477
4.32900429	0.97517246	0.72602141	0.00022493
4.36681223	0.97527432	0.72541416	0.00022503
4.40528631	0.97543287	0.72460532	0.00022509
4.44444466	0.97561044	0.72387213	0.00022509
4.48430490	0.97583002	0.72311968	0.00022509
4.52488708	0.97601587	0.72224402	0.00022512
4.56621027	0.97620153	0.72166133	0.00022518
4.60829496	0.97635317	0.72082525	0.00022535
4.65116262	0.97655159	0.72029662	0.00022553
4.69483566	0.97673154	0.71977067	0.00022583
4.73933649	0.97692686	0.71926492	0.00022615
4.78468895	0.97708863	0.71892905	0.00022654
4.83091784	0.97728890	0.71844876	0.00022694

Figure 21. Example of structure of MIE output file: coef_vs_lambda_case.dat.

This output file is a text file containing the single scattering albedo, asymmetry factor and extinction in function of the longwave in the longwave spectrum. The header indicates the variable and its unit, being μm for the wavelength and m^{-1} for the extinction.

4. Atmospheric Radiative Transfer Database for Earth Climate Observation (ARTDECO)

The Atmospheric Radiative Transfer Database for Earth and Climate Observation package (ARTDECO; https://www.icare.univ-lille.fr/artdeco/) is a numerical tool that gathers models and data for the 1D simulation of Earth atmosphere radiances and fluxes from the ultraviolet to thermal infrared range (0.2-50 μm). It is developed and maintained at the Laboratoire d'Optique Atmosphérique (LOA) and distributed by the data and services center AERIS/ICARE (University of Lille), and funded by the TOSCA program of the French space agency (CNES). In ARTDECO, users can either access a library for the scene or use their own description through ASCII input files. Optical properties for aerosols and clouds can be computed. Then, the user can choose among available models to solve the radiative transfer equation and to compute radiative quantities corresponding to the scene. ARTDECO is thus a flexible tool for remote sensing or radiative forcing applications, such as sensitivity studies, development and optimization of retrieval algorithms, evaluation of the future instruments performances, etc.

The ARTDECO package is used to simulate the radiative fluxes in different layers of the atmosphere, for a given atmospheric scene (with aerosols and/or cirrus clouds), using the DISORT model (Stamnes et al., 1988). One of the advantages of the ARTDECO environment is that it allows us to solve the radiative transfer equation in two ways: 1) by introducing our own phase matrix as a function of wavelength; 2) by using the Henyey-Greenstein function (Henyey and Greenstein, 1941), given extinction, single scattering albedo and asymmetry factor values over the whole spectral range in which the simulation will be done. In most cases, due to the lack of knowledge of the phase matrix of cirrus clouds and aerosols with the observational measurements with which we work, the second option is chosen, even though the Henyey-Greenstein function does not represent a good approximation to the real phase function, especially for ice crystals. Upward and downward radiative fluxes are calculated at different vertical Levels: 31 layers (0.005-20 km) in the shortwave (SW, 0.2-4 μm) and 40 layers (0-100 km) in the longwave (LW, 4-50 μm) spectra. These spectral/vertical ranges are adjustable, together with their spectral/vertical resolution depending on the type of simulation being performed. For instance, if you are validating a pyranometer, you can adjust the spectral range of the simulation to the measuring range of that instrument. Alternatively, if you need to analyze an aerosol layer or cloud with a higher vertical resolution, additional vertical layers can be added into the model's vertical distribution.

Cirrus direct radiative effects at the bottom-of-the-atmosphere (BOA) and top-of-the-atmosphere (TOA) can be calculated as:

$$BOA\ DRE = (F^{BOA}_{gas+cirrus}\downarrow - F^{BOA}_{gas+cirrus}\uparrow) - (F^{BOA}_{gas}\downarrow - F^{BOA}_{gas}\uparrow) \quad (3)$$

$$\begin{aligned} TOA\ DRE &= (F^{TOA}_{gas+cirrus}\downarrow - F^{TOA}_{gas+cirrus}\uparrow) - (F^{TOA}_{gas}\downarrow - F^{TOA}_{gas}\uparrow) \\ &= -(F^{TOA}_{gas+cirrus}\uparrow - F^{TOA}_{gas}\uparrow) \end{aligned} \quad (4)$$

Where $F_{gas+cirrus}$ and F_{gas} are the radiative fluxes with and without the cirrus cloud, respectively. The ↓ and ↑ arrows indicate whether the fluxes are downward or upward, respectively. The simplification of Eq. 4 implies the assumption that the amount of the incoming solar radiation at the TOA is equal for both cases with and without aerosols. With this convention, a negative sign of DRE implies a cirrus cooling effect independently of whether it occurs at the BOA or at the TOA.

The characteristics of the simulations performed with the ARTDECO model are shown in Table 1.

Configuration	
Radiative transfer equation solver	DISORT
Mathematical aproximation	Trucation of Legendre
Incident radiation spectrum	Kurudz medium
K-distribution parametrization	gamesw_1, gamelw_1
Solar configuration mode	Solar Zenith Angle
Surface type	Lambertian

Table 2. Summary of the characteristics of the simulations performed with the ARTDECO model.

These are the general characteristics of the simulations carried out with the ARTDECO model. The simulations have been set up to match as closely as possible those of the GAME model, as the GAME model has been validated with observations in multiple studies (Barragan et al., 2016; Granados-Muñoz et al., 2019; Sicard et al., 2022). The ARTDECO model has multiple options. I invite you to explore the different configurations and the data library of the ARTDECO model.

4.1. Inputs

The inputs of the ARTDECO model are similar to those of the GAME model:
- **Thermodynamic profiles of the atmosphere:** height (km), pressure (hPa), temperature (K), water vapor mixing ratio (ppmv) and ozone concentration (ppmv).
- **optical scattering properties of aerosols and/or clouds:** wavelength (μm), aerosol optical depth, extinction (m^{-1}), single scattering albedo and asymmetry factor.
- **Land surface properties:** albedo and temperature (K).

4.2. Matlab script for inputs

The Matlab script is called "Main_inputs_ARTDECO.m" and has five default boolean configuration variables and two variables associated

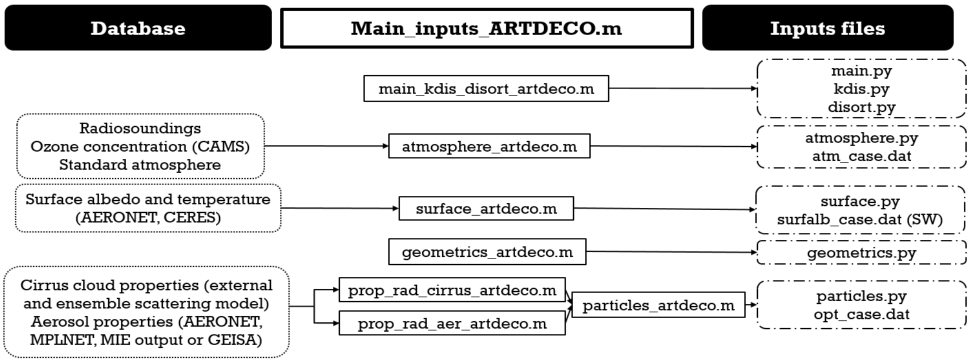


Figure 22. Scheme of the Matlab main script which creates the input files of the ARTDECO model. The data albedo file is only created if the shortwave spectrum is considered.

to the location of the atmospheric scene:
- **bool_cld:** it is true if clouds are considered in the simulation.
- **bool_aer:** it is true if aerosols are considered in the simulation.
- **bool_sw:** it is true if the simulation is made in the shortwave spectrum (0.2 - 4 μm).
- **bool_lw:** it is true if the simulation is made in the longwave spectrum (4 - 50 μm).
- **bool_mie:** it is true if the optical scattering properties for aerosols in the longwave spectrum are obtained with the MIE code from ARTDECO.
- **lat_place:** latitude of the place where the atmospheric scene is located.
- **lon_place:** longitude of the place where the atmospheric scene is located.

In addition, an array is provided with the dates of the aerosol and/or cloud scenes to be simulated (v_date) and the geometrical properties of the clouds are read: base and top heights (v_cbh, v_cth). These properties can be read from data measured with the MPL or can be created manually. Remember that the properties of a cirrus clouds must follow the restrictions of (Gil-Díaz et al., 2024a). The cirrus clouds and aerosols must be vertically separated in the simulation, as it is not known how their optical scattering properties could be combined.

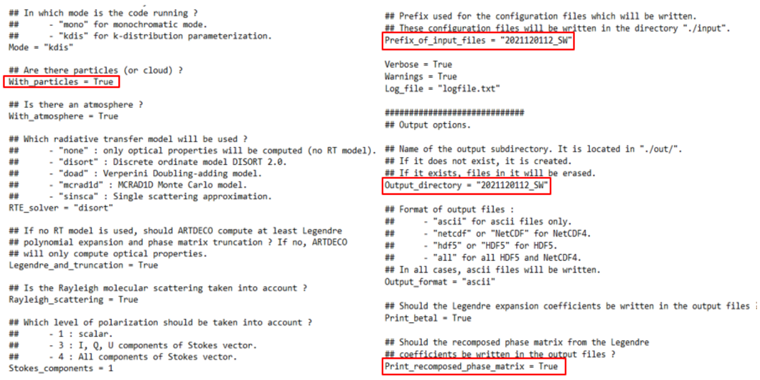
This script is composed of seven functions. The first function (main_kdis_disort_artdeco.m) sets up the directories for all Python input files and generates the following files: main.py, kdis.py, and disort.py. The second function (atmosphere_artdeco.m) generates the atmospehre.py file and creates the data file with the thermodynamic profiles of the atmosphere. The third function (surface_artdeco.m) generates the surface.py file and creates the data file with the surface albedo in the shortwave spectrum. The fourth function (geometrics_artdeco.m) generates the geometrics.py file. The fifth and sixth functions (prop_rad_cirrus_artdeco.m and prop_rad_aer_artdeco.m) calculate the optical scattering properties of cirrus clouds and aerosols, respectively. The seventh function (particles_artdeco.m) generates the particles.py and creates the data file with the optical scattering properties of the cirrus cloud and the aerosols considered in the atmospheric scene.

A scheme of the structure of this script, with the database employed and input files created, is shown in Figure 22.

4.2.1. main_kdis_disort_artdeco.m

First, create the directory where all Python input files will be stored. Second, copy the generic disort.py and kdis.py files, depending on the type of simulation, either in the shortwave or longwave spectrum. Finally, read the base main.py file, modify some specific fields to the atmospheric scene that will be simulated, and place it in the directory with the other Python input files.

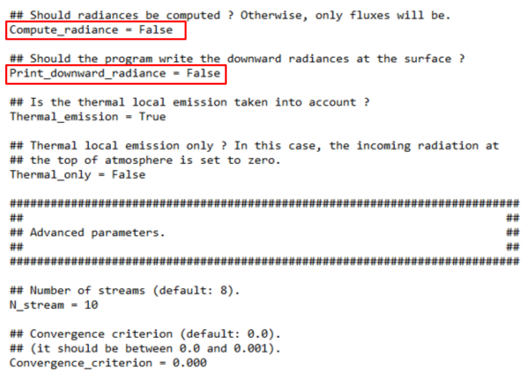
The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw and bool_lw. The files read by this function are: main.py, disort.py, kdis_sw.py and/or kdis_lw.py. The structure of the main.py file is shown in Figure 23.



```
## 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 = True

## 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.
##      - "mcrad1d" : MCRAD1D Monte Carlo model.
##      - "sinsca" : Single scattering approximation.
RTE_solver = "disort"

## 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 = True

## 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

## Prefix used for the configuration files which will be written.
## These configuration files will be written in the directory "./input".
Prefix_of_input_files = "2021120112_SW"

Verbose = True
Warnings = True
Log_file = "logfile.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 = "2021120112_SW"

## Format of output files :
##      - "ascii" for ascii files only.
##      - "netcdf" or "NetCDF" for NetCDF4.
##      - "hdf5" or "HDF5" for HDF5.
##      - "all" for all HDF5 and NetCDF4.
## In all cases, ascii files will be written.
Output_format = "ascii"

## Should the Legendre expansion coefficients be written in the output files
Print_betal = True

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

Figure 23. Example of structure of the main.py file.

The structure of the disort.py file is shown in Figure 24.



```
## 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 = True

## 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 = 10

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

Figure 24. Example of structure of the disort.py file.

The structure of the kdis.py file is shown in Figure 25.

```
## Name of the k-distribution parameterization. The program will
## then search for the file "./lib/kdis/reference/kdis_<name>.dat".
K_distribution_parameterization = "gamesw_1"

## List of gas taken into account for the gaseous absorption.
## If no gas are taken into account, Gas_list = []
Gas_list = ["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 = "mum"
Wavelength_start = 0.2
Wavelength_end = 4.0

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

Figure 25. Example of structure of the kdis.py file.

4.2.2. *atmosphere_artdeco.m*

First, the function reads the base file atmosphere.py and modifies it with the name of the data file that will store the thermodynamic profiles for the simulated case. Second, it creates the data file by reading the vertical height distribution of the simulation spectrum, either shortwave or longwave. The files containing the vertical distributions for these spectra are named z_p_ARTDECO_SW.txt and z_p_ARTDECO_LW.txt, respectively. Third, the function reads the file containing the vertical distribution of ozone (CAMS_O3.nc). To obtain this data, see Appendix D.1. The algorithm finds the data closest to the location of the atmospheric scenario, reads it, and converts ozone concentration units from kg/kg to ppmv. Forth, it reads a file with the 1976 standard atmosphere (COESA, 1976) to fill the atmospheric profiles at heights not reached by the radiosonde and where observational ozone data is unavailable. Fifth, the radiosonde data for that specific date is read. After all data is loaded, the ozone concentration values are vertically interpolated with the radiosonde data. Six, the vertical profile of the water vapor mixing ratio is calculated, and all variables are interpolated once again with the heights in the model's vertical distribution. Finally, null values are replaced with values from the 1976 standard atmosphere, and the model's thermodynamic input files are generated.All input files are located in the same folder named "ARTDECO\INPUTS", except for the radiosondes which are located in a folder named "Dataset\Radiosoundings". Likewise, the output files are written to a folder named "ARTDECO\OUTPUTS\SW" or "ARTDECO\OUTPUTS\LW", for the shortwave and longwave spectrum, respectively. The location of the files can be changed by changing the paths.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, lat_place and lon_place. Where the last two variables indicate the location of the atmospheric scene. The files read by this function are: atmosphere.py, z_p_ARTDECO_SW.txt, z_p_ARTDECO_LW.txt, CAMS_O3.nc, thermo_profil_us67.txt and DORSBCNT0001_#. This function does not have any output, but creates the Python input file atmosphere.py and the data file atm_case.dat, which is an input file of the ARTDECO model. These input files have the structure shown in Figure 26.

atmosphere.py

```
## Name of the atmospheric profile. The program will then be read in
## "./lib/atm/reference/atm_<name>.dat".
Atmospheric_profile = "2021120112_sw_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", 360],
                    ["o2", 2.09e5]]
```

atm_case.dat

```
# AFGL atmospheric constituent profile. tropical. ( AFGL-TR-86-0110)
#
# number of sampled altitude :
 50
# number of sampled gas :
 2
# sampled gas :
 h2o
 o3
#     z(km)      p(mb)       T(K)     air(cm-3)      h2o(cm-3)      o3(cm-3)
     120.00   2.0000e-05   380.00   4.2885520e+11   8.4500010e+04   2.1125000e+02
     115.00   4.0000e-05   299.70   8.7001670e+11   2.0894400e+05   4.3530000e+03
     110.00   6.0000e-05   241.60   1.9396310e+12   5.4348000e+05   9.7050000e+04
     105.00   1.3000e-04   212.00   4.4413940e+12   1.5113000e+06   8.8900010e+05
     100.00   2.9000e-04   190.70   1.0976380e+13   4.3920000e+06   4.3920000e+06
      95.00   6.9000e-04   184.30   2.7038020e+13   1.4612400e+07   1.3530000e+07
      90.00   1.7200e-03   177.00   7.0382870e+13   5.9865500e+07   3.6623600e+07
      85.00   4.4000e-03   177.10   1.7994760e+14   2.3413000e+08   9.0050000e+07
      80.00   1.1000e-02   184.80   4.3112440e+14   9.0593990e+08   1.4236200e+08
      75.00   2.6000e-02   201.80   9.3317690e+14   3.0815400e+09   1.6808400e+08
      70.00   5.8000e-02   218.90   1.9190840e+15   8.6400000e+09   5.7600000e+08
      65.00   1.2100e-01   236.00   3.7135150e+15   2.0066400e+10   2.4154000e+09
      60.00   2.3900e-01   253.10   6.8393930e+15   4.1064000e+10   7.5284000e+09
      55.00   4.5600e-01   263.40   1.2538940e+16   7.5300000e+10   2.2590000e+10
      50.00   8.5400e-01   270.20   2.2892030e+16   1.3746000e+11   6.4148000e+10
      47.50   1.1600e+00   269.60   3.1163770e+16   1.8402100e+11   1.0760550e+11
      45.00   1.5900e+00   264.80   4.3490160e+16   2.4806400e+11   1.9584000e+11
      42.50   2.2000e+00   259.40   6.1427740e+16   3.3808500e+11   3.6267300e+11
      40.00   3.0500e+00   254.00   8.6971700e+16   4.5255600e+11   6.5272490e+11
```

Figure 26. Example of structure of the input files: atmosphere.py and atm_case.dat, which is the ARTDECO input data file that contains the thermodynamical profiles.

4.2.3. *surface_artdeco.m*

This function generates the input surface.py file and the data file with the albedo values if the simulation is in the shortwave spectrum. First, an AERONET file containing the albedo in the shortwave spectrum is read. Subsequently, a seasonal average of the season of the year in which the atmospheric scene is located is made and the albedo values are spectrally interpolated. It is established that for wavelengths longer than 1.02 μm, the surface albedo has no spectral dependence. Second, the albedo in the shortwave spectrum is writing in a file, Third, a NOAA20 file containing the surface emissivity and temperature are read. To calculate the surface albedo in the longwave, the following equation is used: 1 - emissivity. For data download, see Appendix D.3. The pixel closest to the location of the atmospheric scene is found and the data corresponding to that date is analyzed. Changing the location of the simulation can be done by modifying the variables lat_place and lon_place. By default, the surface albedo is set to 0.0157 and the surface temperature to 301.15 K.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, lat_place and lon_place. The files read by this function are: surface.py, lambda_GAME_SW.txt, Barcelona_AERONET_INVERSION_L15.all, NOAA20.nc. This function does not have any output, but creates the Python input file surface.py. Moreover, if the simulation is performed in the shortwave spectrum, it also generates a data input file that contains the surface albedo in the shortwave spectrum (surfalb_case.dat). An example of the structure of the inputs files: surface.py and surfalb_case.dat are shown in Figure 27.

surface.py

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

## 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 = "2021120112_sw"

## In the case of a lambertian surface with a constant albedo.
Surface_albedo = 0

## Surface temperature (K).
## -1 to take the same temperature as the lowest layer
##  in the atmospheric profile.
Surface_temperature = 288.041687

###############################################
## 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
```

surfalb_case.dat

```
#
# Surface albedo : fresh ice
#
# Number of sampled values :
 208
# wavelength         albedo
#  (microns)
    0.20000000    0.60000002
    0.20242915    0.60200000
    0.20408164    0.60299999
    0.20576131    0.60399997
```

Figure 27. Example of structure of the input files: surface.py and surfalb_case.dat, which is the ARTDECO data input file in the shortwave spectrum that contains the surface albedo.

As the surface albedo in the shortwave spectrum comes from AERONET and the spectral range of the example extends up to 4 μm, in order to have values in the larger wavelengths, the spectral dependence is removed.

4.2.4. geometrics_artdeco.m

First, the function reads the base file geometrics.py and secondly, it modifies it according to the simulation to be performed. In our case, the solar zenith angle of our atmospheric scene is added. Other configurations are possible and the user is encouraged to explore them.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, lat_place and lon_place. The files read by this function are: geometrics.py. This function does not have any output, but creates the Python input file geometrics.py, whose structure is shown in Figure 28.

```
## 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 = [63.3874359,]

## 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 = [
    [2.11, 41.39, 355, 12.0],
    ]
```

Figure 28. Example of structure of the geometrics.py file.

4.2.5. prop_rad_cirrus_artdeco.m

In case that the atmospheric scene contains an ice cloud, its optical scattering properties are calculated with the self-consistent scattering model for cirrus clouds, as I have done during part of my thesis (Gil-Diaz et al., 2024b). First, the wavelengths and coefficients of the parameterizations are read out. Second, the wavelengths are read from the GAME model. Third, the vertical profile of the ice water content in the whole cloud is calculated, after interpolating the extinction and temperature with the vertical distribution of the GAME model. Finally, the parameterizations are applied and the optical scattering properties of the cirrus cloud are calculated.

The inputs of this function are: cloud base height (cbh), cloud top height (cth), vertical distribution of lidar data (v_h_lidar), vertical distribution of temperature in cirrus cloud (v_temp_cld), vertical distribution of cloud extinction (v_ext_cld). The units of these inputs are: the heights in km, the temperature in K and the extinction in m^{-1}. The files read by this function are: lambda_Baran.txt, sca_coef_Baran.txt, abs_coef_Baran.txt, asym_coef_Baran.txt, lambda_GAME_SW.txt and lambda_GAME_LW.txt, z_p_GAME_LW.txt. The outputs of this functions are: matrices with the cloud extinction coefficient (m_ext_cld), asymmetry factor (m_asy_cld), single scattering albedo (m_ssa_cld), scattering (m_sc_cld), absorption (m_abs_cld) and the vertical distribution of the ice water content (v_iwc_cld). The vertical distribution of the cloud has been interpolated with a vertical distribution extending from 0 to 20 km, with 1 km resolution. The rows of the matrices indicate the height over which the cloud extends and the columns the GAME wavelength, in increasing order. Thus, the first row corresponds to 0 km and the first column to 0.2 μm by default. If the optical scattering properties of the clouds with a higher spatial resolution are desired, the user must change the function slightly.

4.2.6. prop_rad_aer_artdeco.m

In case aerosols are considered in the atmospheric scene, this function is used to calculate their optical scattering properties from AERONET data. For data download, see Appendix D.2. First, the spectral and vertical distributions are read from the model in the shortwave and longwave spectra. Second, an AERONET file containing the variables: aerosol optical depth, single scattering albedo and asymmetry factor for the date of the atmospheric scene is read. If the time of the case is at night (between 6 a.m. and 6 p.m.), the aerosol properties of the previous days are averaged until there is sufficient data. The AERONET wavelengths are then interpolated with the ARTDECO wavelengths using the Ängstrom exponent. Third, an aerosol file is read from MPLNET to analyze the vertical distribution of aerosols. In the case of time 00 UTC, in order to make temporal averages centred on that hour, the file of the previous day is read. In the case of no data, an equidistant vertical distribution up to 1 km height is assumed. Fourth, the aerosol properties in the shortwave spectrum and the aerosol vertical distribution are stored in different variables.

The inputs of this function are: date (datenum). The files read by this function are: lambda_ARTDECO_SW.txt, z_p_ARTDECO_SW.txt, Barcelona_AERONET_INVERSION_L15.all and MPL#_V3_L15_AER_#. The outputs of this function are: a cell array with the vertical profiles of the weight of lidar extinction, the aerosol optical depth at 0.55μm and the extinction coefficient at 0.55μm (data_aeronet_prof) and another cell array with the aerosol optical scattering properties at SW ARTDECO wavelength (data_aeronet_interpn).

4.2.7. particles_artdeco.m

First, the function reads the base file particles.py and the vertical and spectral distribution of the ARTDECO model. Second, it modifies the python file by adding the layers of aerosols and clouds, either shortwave or longwave. If aerosols are to be added in a simulation in the LW spectrum, the optical scattering properties of aerosols from the GEISA database will be considered. Depending on the file containing data from an external database to be read, the Matlab function has to be modified accordingly. Third, it generates data input files with the optical properties of the clouds in both spectra and of the aerosols only in the SW spectrum, since the properties in the LW have been defined invariant.

The inputs of this function are: date (datenum), bool_cld, bool_aer, bool_sw, bool_lw, prop_cld and prop_aer. The files read by this function are: particles.py, z_p_ARTDECO_SW.txt, z_p_ARTDECO_LW.txt, lambda_ARTDECO_SW.txt, lambda_ARTDECO_LW.txt and opt_X.dat. This function does not have any output, but creates the input files particles.py and opt_case.dat, whose structures are shown in Figure 29.

particles.py

```
## 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 = [
   ["2018120212_1_pbl_sw", False, True, True, 0.00002, "layer",0.005],
   ["2018120212_2_pbl_sw", False, True, True, 0.00002, "layer",0.010],
   ["2018120212_3_pbl_sw", False, True, True, 0.00008, "layer",0.020],
   ["2018120212_4_pbl_sw", False, True, True, 0.00008, "layer",0.030],
   ["2018120212_5_pbl_sw", False, True, True, 0.00008, "layer",0.040],
   ["2018120212_6_pbl_sw", False, True, True, 0.00008, "layer",0.050],
   ["2018120212_7_pbl_sw", False, True, True, 0.00319, "layer",0.100],
   ["2018120212_8_pbl_sw", False, True, True, 0.00668, "layer",0.200],
   ["2018120212_9_pbl_sw", False, True, True, 0.01223, "layer",0.400],
   ["2018120212_10_pbl_sw", False, True, True, 0.01079, "layer",0.600],
   ["2018120212_11_pbl_sw", False, True, True, 0.00805, "layer",0.800],
   ["2018120212_12_pbl_sw", False, True, True, 0.00346, "layer",1.000],
   ["2018120212_13_pbl_sw", False, True, True, 0.00377, "layer",2.000],
   ["2018120212_1_cld_sw", False, True, True, 0.00725, "layer",12],
   ["2018120212_2_cld_sw", False, True, True, 0.04290, "layer",13],
         ]
```

opt_case.dat

```
#
# Optical properties to be used by ARTDECO
#
# Used model to obtain that properties is:
 unknown
# Used material is:
 unknown
# Number of phase matrix elements :
          4
# number of wavelengths
          2
#       lambda      nteta        Cext             SSA              g
    0.4120000E+00   1900    1.0000000E+00    1.0000000E+00    -0.3276800E+05
    0.5500000E+00   1900    1.0000000E+00    1.0000000E+00    -0.3276800E+05
# Phase matrix
```

Figure 29. Example of structure of the input files: particles.py and opt_case.dat, which is the ARTDECO input data file that contains the optical scattering properties of the cirrus cloud or aerosols.

In the input data file opt_case.dat it is always necessary to add the optical scattering properties at the extremes of the spectral range in which the simulation will be done and at the wavelength of 0.55μm.

4.3. Execution

Once all the inputs for the ARTDECO model are created, the text files are passed to CALCULA (see website https://tsc.upc.edu/en/it-services/computing-services) using the FileZilla program. Additionally, a text file called dates_ARTDECO.txt, whose structure is shown in Figure 30.

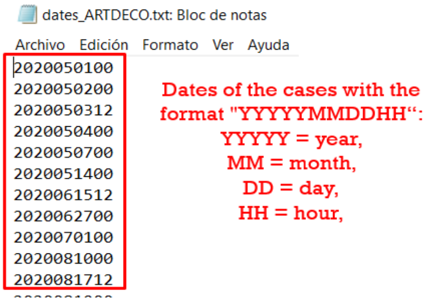
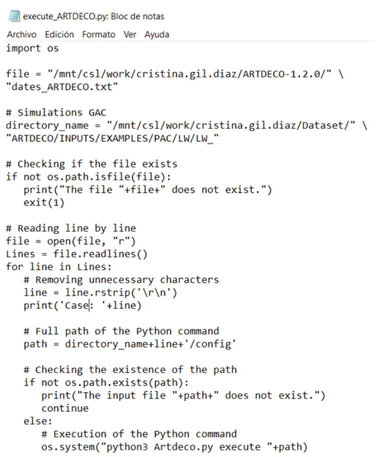


Figure 30. Example of structure of the text file: dates_ARTDECO.txt, which contains the dates of the cases.

If you do not have the ARTDECO model installed in your Linux environment or in your CALCULA directory, 1) download the GAME model from the NextCloud path: Lidar < upclidar_Software < ARTDECO < ARTDECO-1.2.0, 2) transfer the ARTDECO model to your Linux environment or CALCULA directory, 3) place it in the ARTDECO-1.2.0\ARTDECO-1.2.0 directory with the cd command, 4) compile the model with the command make cleanall && make artdeco > & compil.log.

Once the ARTDECO model has been installed, it can be run with the execute_ARTDECO.py file with the command python3 execute_ARTDECO.py. Depending on the simulations to be done, the user has to change the Python file that runs the ARTDECO model. The structure of the execution file is shown in Figure 31.




```
import os

file = "/mnt/csl/work/cristina.gil.diaz/ARTDECO-1.2.0/" \
"dates_ARTDECO.txt"

# Simulations GAC
directory_name = "/mnt/csl/work/cristina.gil.diaz/Dataset/" \
"ARTDECO/INPUTS/EXAMPLES/PAC/LW/LW_"

# Checking if the file exists
if not os.path.isfile(file):
   print("The file "+file+" does not exist.")
   exit(1)

# Reading line by line
file = open(file, "r")
Lines = file.readlines()
for line in Lines:
   # Removing unnecessary characters
   line = line.rstrip('\r\n')
   print('Case: '+line)

   # Full path of the Python command
   path = directory_name+line+'/config'

   # Checking the existence of the path
   if not os.path.exists(path):
      print("The input file "+path+" does not exist.")
      continue
   else:
      # Execution of the Python command
      os.system("python3 Artdeco.py execute "+path)
```

Figure 31. Example of structure of the Python file: execute_ARTDECO.py, which executes the ARTDECO model automatically.

Once the files necessary to run the cases automatically have been explained, the execution mechanism will be detailed step by step. First, the ARTDECO model is installed if it is not already installed in your CALCULA directory. To do this, download the model found in the Nextcloud lidar group folder (Lidar < upclidar_Software < ARTDECO < ARTDECO-1.2.0). Secondly, you transfer the model to your CALCULA directory via Filezilla. For the connection to the CALCULA server with Filezilla, please refer to the manual provided by UPC (https://tsc.upc.edu/en/it-services/computing-services/research-groups-specifications/csl-commsenslab-group). If you do not have access to CALCULA, you should write to the IT department. Thirdly, you pass also the input files from the local to the CALCULA server and adapt the execution files to the simulations you would like to do. Finally, you run the execute_ARTDECO.py file in a Linux terminal on the server. To access the Linux terminal, go to Dashboard - CALCULA OnDemand (https://ondemand.tsc.upc.edu) and open a CALCULA Interactive session. Within the session, go to System < LXTerminal and click. Once the session has been opened, you must go to the folder where the GAME model is located with the cd command and execute the file execute_ARTDECO.py that you will find in the same directory. As an example, Figure 32 shows a screenshot of the terminal.

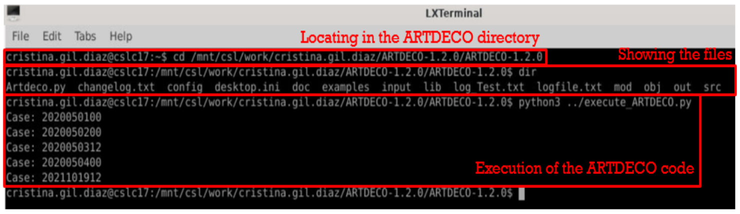


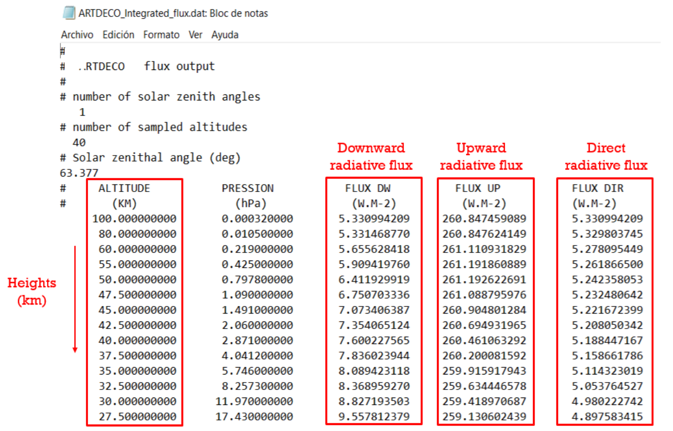
Figure 32. Example of execution of the ARTDECO model.

If a 'strange' error occurs during compilation, it may be because the model's environment is not suitable. Make sure that the environment has the following dependencies:
- libgfortran5
- gfortran
- build-essential

In case of an error in the execution of the GAME model not associated with the inputs, I recommend that you ask the IT service of the department.

4.4. Outputs

The outputs of the ARTDECO model can be very diverse, for example: files with the radiative fluxes integrated in each vertical layer of the atmospheric scene, values of net radiative heating rate and warming rate in different vertical layers, values of the K-distribution bands in each spectral range, values of the optical scattering properties of clouds or aerosols if the phase matrix with spectral dependence or the MIE code has been used, etc. All these output files are described in detail in the ARTDECO manual, which can be found in the Lidar < upclidar_Software < ARTDECO folder of NextCloud. In this section, only an example of the structure of the output ARTDECO_Integrated_Flux.dat file containing the radiative fluxes integrated in the different vertical layers of the atmospheric scene is shown in Figure 33.




```
#
#  ..RTDECO   flux output
#
# number of solar zenith angles
   1
# number of sampled altitudes
  40
# Solar zenithal angle (deg)
63.377
#     ALTITUDE         PRESSION        FLUX DW        FLUX UP          FLUX DIR
#       (KM)             (hPa)         (W.M-2)        (W.M-2)          (W.M-2)
    100.000000000      0.000320000    5.330994209    260.847459089    5.330994209
     80.000000000      0.010500000    5.331468770    260.847624149    5.329803745
     60.000000000      0.219000000    5.655628418    261.110931829    5.278095449
     55.000000000      0.425000000    5.909419760    261.191860889    5.261866500
     50.000000000      0.797800000    6.411929919    261.192622691    5.242358053
     47.500000000      1.090000000    6.750703336    261.088795976    5.232480642
     45.000000000      1.491000000    7.073406387    260.904801284    5.221672399
     42.500000000      2.060000000    7.354065124    260.694931965    5.208050342
     40.000000000      2.871000000    7.600227565    260.461063292    5.188447167
     37.500000000      4.041200000    7.836023944    260.200081592    5.158661786
     35.000000000      5.746000000    8.089423118    259.915917943    5.114323019
     32.500000000      8.257300000    8.368959270    259.634446578    5.053764527
     30.000000000     11.970000000    8.827193503    259.418970687    4.980222742
     27.500000000     17.430000000    9.557812379    259.130602439    4.897583415
```

Figure 33. Example of output of the ARTDECO model, which contains the radiative fluxes integrated in the different vertical layers of the atmospheric scene.

For more examples of different ARTDECO outputs you can find them in the examples folder within the model.

A. Summary of files for GAME execution

Data files needed to compute the GAME input files:
- **abs_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the absorption coefficient of the cirrus clouds.
- **asym_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the asymmetry factor of the cirrus clouds.
- **Barcelona_AERONET_INVERSION_L15.all:** file downloaded from AERONET covering the whole study time period with values of surface albedo and aerosol properties: datetime, aerosol optical depth, single scattering albedo, asymmetry factor, effective geometrical radius with its associated standard deviation and aerosol concentration. The AERONET working wavelengths are: 0.44, 0.675, 0.87, 1.02 μm.
- **CAMS_O3.nc:** netcdf file that contains the vertical profile of ozone concentration.
- **coef_vs_lambda_#.dat:** data file with the output of the MIE code. It contains in the first line the extinction coefficient at 0.55 μm and in the next lines, the single scattering albedo, the asymmetry factor and extinction in function of the wavelength.
- **DORSBCNT0001_#:** file that contains the thermodynamical profiles measured by radiosoundings.
- **lambda_Baran.txt*:** text file that contains the spectral distribution of the ensemble scattering model for cirrus clouds.
- **lambda_GAME_SW.txt*:** text file that contains the spectral distribution of the GAME model in the shortwave spectrum.
- **lambda_GAME_LW.txt*:** text file that contains the spectral distribution of the GAME model in the longwave spectrum.
- **MPL#_V3_L15_AER_#:** netcdf file that contains the optical properties of aerosols measured by the Micro Pulse Lidar. It is employed to get the vertical distribution of aerosols.
- **NOAA20.nc:** netcdf file that contains the surface emissivity and temperature measured by the NOAA-20 satellite.
- **opt_aerosol_lw.dat:** data file from GEISA database, that contains the optical scattering properties of continental aerosols in the longwave spectrum.
- **sca_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the scattering coefficient of the cirrus clouds.
- **thermo_profil_us67.txt:** text file that contains the thermodynamical profiles of the 1976 standard atmosphere.
- **z_p_GAME_SW.txt*:** text file that contains the vertical distribution of the GAME model in the shortwave spectrum.
- **z_p_GAME_LW.txt*:** text file that contains the vertical distribution of the GAME model in the longwave spectrum.

* Text files with the configuration of the GAME model. # Symbol that indicates the existence of additional prefixes or suffixes in the name of the file data.

Files for the GAME model execution:
- **dates_GAME.txt:** text file that contains the code of the simulation, the number the cases and the dates of the cases that will be simulated.
- **execute_GAME.sh:** bash file that executes the GAME model.

Main scripts of the GAME model:
- **game_aero.f:** fortran file that is the main script of the GAME model for the shortwave spectrum.
- **game-lw.f:** fortran file that is the main script of the GAME model for the longwave spectrum.

Output files of the GAME model:
- **case_out_sw/lw.dat:** data file that contains the radiative fluxes at each vertical layer of the atmospheric scene in the shortwave/longwave spectrum.
- **out_summary_sw/lw.dat:** data file that contains the direct radiative effects at bottom-of-the-atmosphere and top-of-the-atmosphere, together with the characteristics of the case.

B. Summary of files for MIE execution

Data files needed to compute the MIE input file:
- **distribution-char_case.dat:** data file that contains the number of modes of the size distribution of the aerosols, together with their respective effective geometrical radius, standard deviation and concentration.
- **n_vs_lambda_case.dat:** data file that contains the real and imaginary parts of the refractive index in function of the wavelength.

File for the MIE model execution:
- **dates_MIE.txt:** text file that contains the number the cases and the dates of the cases that will be simulated.

Main script of the MIE code:
- **mie_net_LW.f:** fortran file that is the main script of the MIE code.

Output file of the MIE model:
- **coef_vs_lambda_case.dat:** data file that contains the optical scattering properties of aerosols in the longwave spectrum.

C. Summary of files for ARTDECO execution

Data files needed to compute the ARTDECO input files:
- **abs_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the absorption coefficient of the cirrus clouds.
- **asym_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the asymmetry factor of the cirrus clouds.
- **atmosphere.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **atm_X_ppmv.dat:** data file which serves as a base file to be modified by an automatization routine to add the thermodynamical profiles of the atmospheric scene to be simulated.
- **Barcelona_AERONET_INVERSION_L15.all:** file downloaded from AERONET covering the whole study time period with values of surface albedo and aerosol properties: datetime, aerosol optical depth, single scattering albedo, asymmetry factor, effective

geometrical radius with its associated standard deviation and aerosol concentration. The AERONET working wavelengths are: 0.44, 0.675, 0.87, 1.02 μm.
- **CAMS_O3.nc:** netcdf file that contains the vertical profile of ozone concentration.
- **disort.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **DORSBCNT0001_#:** file that contains the thermodynamical profiles measured by radiosoundings.
- **geometrics.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **kdis_lw.py:** Python file which serves as a base Python file to be modified by an automatization routine for the LW simulations.
- **kdis_sw.py:** Python file which serves as a base Python file to be modified by an automatization routine for the SW simulations.
- **lambda_Baran.txt*:** text file that contains the spectral distribution of the ensemble scattering model for cirrus clouds.
- **lambda_ARTDECO_SW.txt*:** text file that contains the spectral distribution of the ARTDECO model in the shortwave spectrum.
- **lambda_ARTDECO_LW.txt*:** text file that contains the spectral distribution of the ARTDECO model in the longwave spectrum.
- **main.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **MPL#_V3_L15_AER_#:** netcdf file that contains the optical properties of aerosols measured by the Micro Pulse Lidar. It is employed to get the vertical distribution of aerosols.
- **NOAA20.nc:** netcdf file that contains the surface emissivity and temperature measured by the NOAA-20 satellite.
- **opt_X.dat:** data file which serves as a base file to be modified by an automatization routine to add the optical scattering properties of the atmospheric structure to be considered in the simulation, either an aerosol or cloud layer.
- **opt_aerosol_lw.dat:** data file from GEISA database, that contains the optical scattering properties of continental aerosols in the longwave spectrum.
- **particles.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **sca_coef_Baran.txt:** text file that contains the coefficients of the ensemble scattering model for cirrus clouds to calculate the scattering coefficient of the cirrus clouds.
- **surface.py:** Python file which serves as a base Python file to be modified by an automatization routine.
- **surfalb_X_sw.dat:** data file which serves as a base file to be modified by an automatization routine to add the surface albedo provided by AERONET for SW simulations.
- **thermo_profil_us67.txt:** text file that contains the thermodynamical profiles of the 1976 standard atmosphere.
- **z_p_ARTDECO_SW.txt*:** text file that contains the vertical distribution of the ARTDECO model in the shortwave spectrum.
- **z_p_ARTDECO_LW.txt*:** text file that contains the vertical distribution of the ARTDECO model in the longwave spectrum.

* Text files with the configuration of the ARTDECO model. # Symbol that indicates the existence of additional prefixes or suffixes in the name of the file data.

Files for the ARTDECO model execution:
- **dates_ARTDECO.txt:** text file that contains the dates of the cases that will be simulated.
- **execute_ARTDECO.py:** Python file that executes the ARTDECO model.

Output files of the ARTDECO model:
- **ARTDECO_Integrated_Flux.dat:** data file that contains he radiative fluxes integrated in the different vertical layers of the atmospheric scene.

D. Data download

D.1. Ozone data

To obtain ozone data in a vertical profile, the CAMS European air quality reanalysis is used. The steps to download the data are: 1) go to the web page where all Copernicus datasets are displayed (https://ads.atmosphere.copernicus.eu/datasets), 2) select the product type as Reanalysis and the Spatial coverage as Europe, 3) select the CAMS European air quality reanalyses, 4) click on the Download tab, 5) select the Ozone variable, all levels, the desired date, the MONARCH model type (corresponds to the BSC model) and the type of reanalysis (if it is a recent date, only the Interim reanalysis option is available; in (Gil-Diaz et al., 2024b) Validated reanalysis was used because the database was not recent), 6) click on the blue login/register to accept licenses button, 7) if you do not have a user, register, 8) click on the blue submit form button, 9) download the data when it is ready.

D.2. AERONET data

To obtain the optical scattering properties of the aerosols and their size distribution, it is necessary to download the data from the AERONET website (https://aeronet.gsfc.nasa.gov/). The steps to download the data are: 1) click on the tab Aerosol Inversions (V3) < Download Tool, 2) select the AERONET station with the help of the Region Selection, Country/Province Region and AERONET Site Selection drop-downs, 3) click on the blue name of the AERONET station, 4) select the dates you are interested in with the help of the START and END drop-downs, 5) select the variables All Products Except Phase Function and U27 to make sure that all necessary products are downloaded, 6) click on Almucantar Sky Scan Scenario, 7) choose the level of the data you want (if there is level 2 data), 8) select the level of the data you want (if there is level 2 data. 0 select that level), 8) click on All points and the grey Download button, 9) you will get a new tab in your browser and click OK, 10) go to your Downloads directory and there you will find a zip file with the data.

D.3. Surface data

To obtain the surface properties, it is necessary to download the data from the CERES website (https://ceres.larc.nasa.gov/data/). The steps to download the data are: 1) Click on the red Order Data button in the Single Scanner Footprint (SSF) section, 2) select NOAA-20 (satellite with most recent data), 3) Click again on the red Order Data button, 4) select from the Surface Parameters section the variables Surface LW emissivity, 5) select the variable Surface Skin Temperature from the GEOS-5 Atmosphere Parameters section, 6) in the Spatial reoslution section, select the spatial range where you want the data, 7) in the Time Range section, select the date range, 8) in the Email Address section, enter your email address, 9) click the light blue Add to Chart button, 10) in the drop-down tab, enter again your email address and the letter code displayed on the screen, 11) click the light blue Submit button (For this you need a password that if you are not registered, will be sent to you by email, 12) go to the top of the web page and click on the Shopping Chart (TBD) window, 13) click on the light blue Submit Order button, 14) when the data is downloaded you will be sent an email with the link to download the data.

Referencias

Barragan, R., Romano, S., Sicard, M., Burlizzi, P., Perrone, M.R., and, Comeron, A.: Estimation of mineral dust direct radiative forcing at the european aerosol research lidar network site of lecce, Italy, during the chArmex/ADRIMED summer 2013 campaign: Impact of radiative transfer model spectral resolutions, J. Geophys. Res., 121(17), 10,237-10,261, https://doi.org/10.1002/2016JD025016, 2016.

COESA: U.S. Standard Atmosphere, U.S. Government Printing Office, Washington, D.C, 1976.

Gil-Díaz, C., Sicard, M., Comerón, A., Camilo, D., Muñoz-porcar, C., Rodríguez-gómez, A., Lewis, J.R., Ellsworth, J., and, Lolli, S.: Geometrical and optical properties of cirrus clouds in Barcelona, Spain: Analysis with the two-way transmittance method of 5 years of lidar measurements, Atmos. Meas. Tech., 1–31, https://doi.org/10.5194/amt-17-1197-2024, 2024a.

Gil-Díaz, C., Sicard, M., Sourdeval, O., Saiprakash, A., Muñoz-Porcar, C., Comerón, A., Rodríguez-Gómez, A., Fortunato dos Santos Oliveira, D.C.: Study of optical scattering properties and effects of high-altitude cirrus clouds in Barcelona, Spain with 4 years of lidar measurements, Atmos. Chem. Phys., https://egusphere.copernicus.org/preprints/2024/egusphere-2024-2131/, 2024b.

Granados-Muñoz, M.J., Sicard, M., Papagiannopoulos, N., Barragán, R., Bravo-Aranda, J.A., and, Nicolae, D.: Two-dimensional mineral dust radiative effect calculations from CALIPSO observations over Europe, Atmos. Chem. and Phys., 19(20), 13157–13173, https://doi.org/10.5194/acp-19-13157-2019, 2019.

Henyey, L.G., and Greenstein, J.L.: Diffuse radiation in the galaxy. Astrophys. J., 93, 70-83, https://doi.org/10.1086/144246, 1941.

Lacis, A.A., and, Oinas, V.: A description of the correlated k distributed method for modeling nongray gaseous absorption, thermal emission, and multiple scattering in vertically inhomogeneous atmospheres, J. Geophys. Res., 96, 9027-9063, https://doi.org/10.1029/90JD01945, 1991.

Sicard, M., Córdoba-Jabonero, C., López-Cayuela, M.Á., Ansmann, A., Comerón, A., Zorzano, M. P., Rodríguez-Gómez, A., and, Muñoz-Porcar, C.: Aerosol radiative impact during the summer 2019 heatwave produced partly by an inter-continental Saharan dust outbreak - Part 2: Long-wave and net dust direct radiative effect, Atmos. Chem. and Phys., 22(3), 1921–1937. https://doi.org/10.5194/acp-22-1921-2022, 2022.

Stamnes, K., Tsay, S.-C., Wiscombe, W., and, Jayaweera, K.: Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=eb8a5a97ff53cd60b9de11a51a345b5c7492d342, 1988.