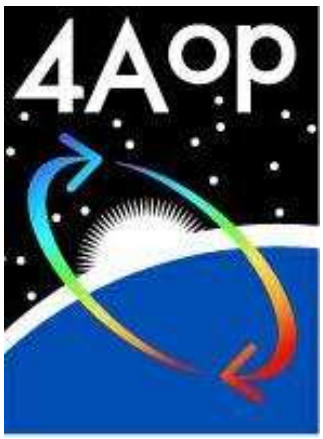

Advanced utilisation manual

Software Version 4AOP v1.5

April 2019

NOVELTIS

Parc Technologique du Canal
2, Avenue de l'Europe
31520 Ramonville-Saint-Agne
France

cnes
De l'Espace pour la Terre

Centre National d'Etudes Spatiales
2, place Maurice Quentin
75039 Paris Cedex 01
France

LMD

Ecole Polytechnique
91128 Palaiseau Cedex
France

THALES

Thales Services SAS
290, Allée du lac
31670 Labège
France

1. INTRODUCTION 3

1.1. PURPOSE 3
1.2. HOW TO USE THIS DOCUMENT 3
1.3. PEOPLE AND ACKNOWLEDGMENTS 4
1.4. REFERENCING 4A/OP IN PUBLICATION 4

2. DESCRIPTION OF THE 4A/OP-SUB MODULES 5

2.1. FLOW CHART OF THE SOFTWARE 5
2.2. GENERAL DESCRIPTION 5
2.3. DESCRIPTION OF THE FORTRAN 4A/OP-SUBROUTINE STRUCTURE 6
2.3.1. Structure of input parameters 6
2.3.2. Structure of output data 6
2.3.3. Modification of 4A/OP-Sub structure fields 7
2.4. DIRECTORY DESCRIPTION 8
2.5. DATA 9
2.5.1. Input Data 9
2.5.2. Output Data 9
2.6. DESCRIPTION OF THE SOFTWARE PROCESS MANAGEMENT 10
2.6.1. Make description 10
2.6.2. The file Makefile 10

3. HOW TO USE 4A/OP 12

3.1. GENERAL 12
3.1.1. Execution 12
3.1.2. CPU runtime 13
3.2. USER INPUT 14
3.2.1. Simulation definition parameters 14
3.2.2. Atmospheric profiles 19
3.2.3. Spectral emissivity / reflectance / brdf 25
3.2.4. Instrument Spectral Response Function (ISRF) 28
3.2.5. Aerosol / ice cloud physical parameters 32
3.3. OUTPUT 36
3.3.1. Output file description 36
3.3.2. Examples 45
3.3.3. Result visualization and jacobian unit recommendation 49
3.3.4. Statistics tool 50
3.4. USE 4A/OP SUBROUTINE MODE 52
3.4.1. Input sub-structures description 52
3.4.2. Modifying the 4A/OP-sub structure fields 54

4. SOFTWARE MAINTENANCE 56

4.1. VERSION CONTROL 56
4.2. ERROR CONTROL 56

5. REFERENCES 57

6. ACRONYMES 58

1. INTRODUCTION

1.1. Purpose

The software 4A/OP version subroutine, called 4A/OP-sub hereafter, is provided for an enhancement by NOVELTIS,

Numerous works have been achieved in order to improve the readability of the software code and to favour its portability and its maintenance. The following main tasks have been carried out:

- Inventory of fixtures: description and analysis of the full 4A/OP source code in order to estimate its quality; setup of test cases (and non-regression tests) in order to check the conformity of the consolidated code results with the original code results;
- Addition of scientific functions:
  - User-defined spectral emissivity/reflectance/brdf functions;
  - The plan-parallel atmosphere assumption has been replaced by a spherical atmosphere approximation;
  - Hole convolution option for computer time saving in some particular cases;
  - Implementation of user option for simulation of the solar contribution to spectral radiance and Jacobians;
  - Implementation of user option for radiative transfer simulation in limb geometry including refraction,
  - Implementation of scattering for molecular, aerosol and cloud (cirrus,…) contribution;
  - Simulation of polarized light;
  - Jacobian computation of temperature profile, atmospheric molecule profiles, surface pressure, emissivity and temperature and aerosol optical thickness;
- Code industrialization:
  - Setup of a revision control and error management,
  - Addition of comments,
  - Fortran 90 porting,
  - Implementation of the compliance with the Fortran Standards;
- Software promotion:
  - Reference documentation,
  - Graphical User Interface (GUI),
  - Web site http://www.noveltis.fr/4AOP/ including an on-line registration form,
  - Distribution with maintenance and assistance; User licence conditions.

The purpose of the Advanced utilisation manual for 4A/OP-subroutine is to provide more detailed instructions for using the model 4A. It also contains information on the internal data, parameterisations, procedures and the physics used. It should both educate users about what the software does, and instruct them how to use it. The role of this guide is to support the software by explaining as clear as possible the process management and the parameters that are playing a major part.

1.2. How to use this document

The "Advanced utilisation manual" is intended to bring more informations help user to begin using 4A/OP-sub. It is divided into three parts:

1. a detailled description of the 4A/OP (section 2),
2. an advanced use of 4A/OP in script mode (section 3),

3. the explanation of the software maintenance procedure (section 4).

1.3. People and acknowledgments

The following persons were involved in building and improving the 4A model, as well as in writing the current Reference Documentation.

Authors from LMD: Raymond Armante
Noëlle A. Scott
Virginie Capelle
Alain Chédin

Authors from NOVELTIS: Emilien Bernard
Laure Chaumat
Carsten Standfuss
Bernard Tournier

4A/OP-subroutine was developed under funding by CNRS and from CNES programs. Further development is supported by CNES.

1.4. Referencing 4A/OP in Publication

If a user is presenting results obtained using 4A/OP, the most complete way to reference it would be to reference the following:

- Scott, N.A. and A. Chedin, 1981: A fast line-by-line method for atmospheric absorption computations: The Automatized Atmospheric Absorption Atlas*. J. Appl. Meteor.*, 20, 802-812.

2. DESCRIPTION OF THE 4A/OP-SUB MODULES

2.1. Flow chart of the software

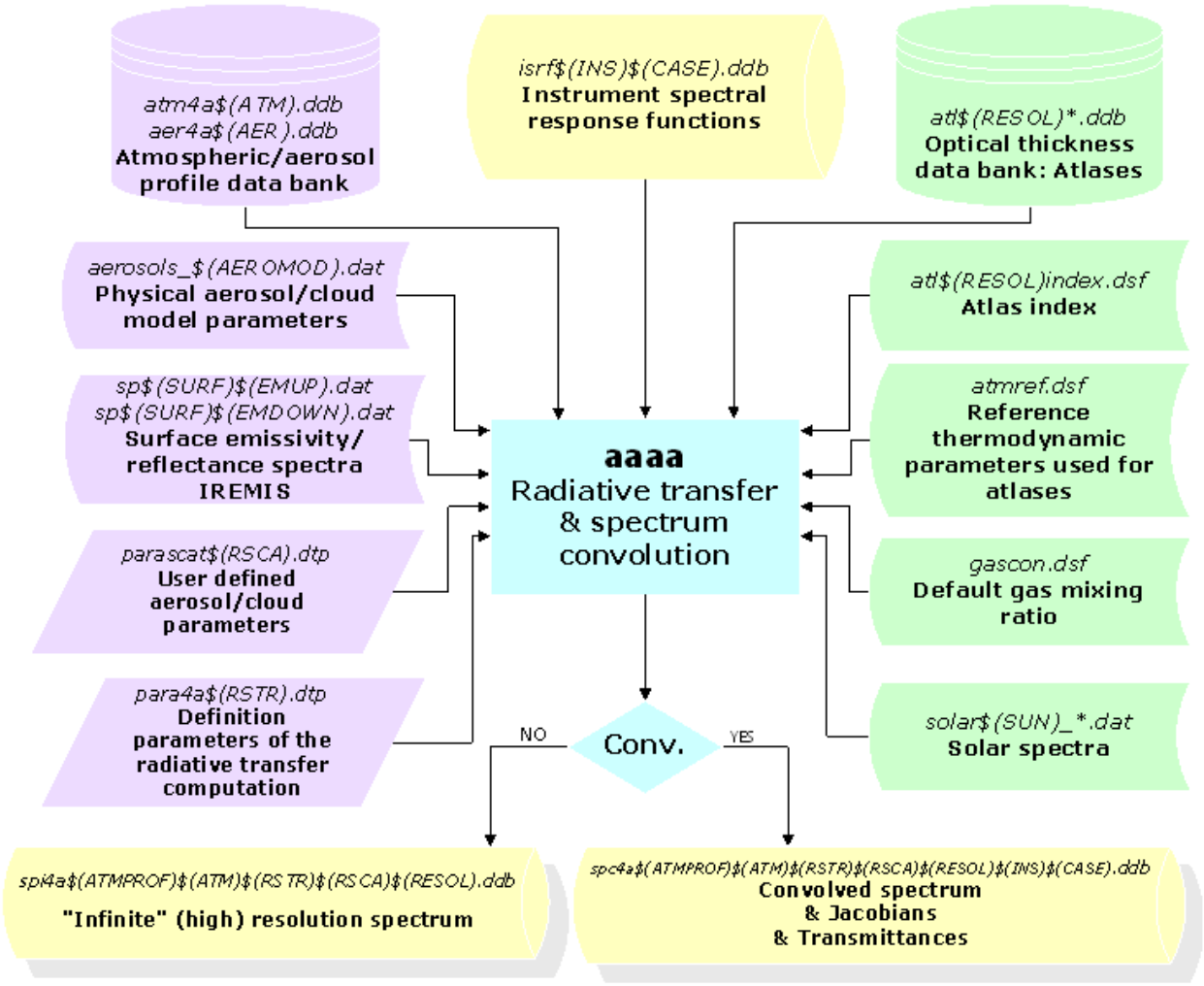

Figure 1: Flow chart of the software 4A/OP (* for several files).

2.2. General Description

The 4A/OP-subroutine code inherites the processing from its previous version (`aaaa` code). However, this new version consists of 5 separate routines (see Table 1) that are called successively in a main program called `mainaaaa`. The program `mainaaaa` can also be used for a stand alone application.

Table 1: list of 4A/OP-sub routines

readparam_aaaa
readscene
check_aaaa
compute_aaaa
write_aaaa

The `readparam_aaaa`, `readscene` and `check_aaaa` subroutines are used to initialize the processing of the 4A/OP-subroutine code by reading all input files (as it was done for the `aaaa` program) and preparing the 4A/OP-subroutine fortran structure that contains all necessary input parameters to run the 4A radiative transfer with the `compute_aaaa` subroutine. Thus, the output data are written in the same output files than for the `aaaa` program by means of the subroutine `write_aaaa`.

2.3. Description of the Fortran 4A/OP-Subroutine Structure

2.3.1. Structure of input parameters

All input parameters that are necessary for the 4A radiative transfer are allocated in the memory by means of an unique structure called `Params` which is organized in the 12 following sub-structures :

```
type :: Params
   type(PAtmRef)       :: atmref       ! reference atmospheric profiles used for atlas
   type(PDefaultGas)   :: gas          ! atmospheric profiles used as default values.
   type(PProfile)      :: profile      ! atmospheric profiles used for the simulation
   type(PSimDef)       :: simdef       ! run characterization parameters
   type(PScatSim)      :: scatsim      ! parameters for scattering simulation
   type(PMolecule)     :: mol          ! Molecule line simulation
   type(PAtlasIndex)   :: index        ! Atlas index input parameters
   type(PEmissivities) :: emis         ! Surface emissivity/reflectance parameters
   type(PBrdf)         :: brdf
   type(PIsrf)         :: isrf
   type(PFilenames)    :: fnames
   type(Pcontinua)     :: continuum
   type(type_Exponential)  :: exponential
   integer(kind=LONG)  :: nopro ! Index number of the used atmospheric profile
   character(len=6)    :: instru ! Simulated instrument
   logical(kind=1)  :: flag_lambda ! flag lambda
end type Params
```

These sub-structures are described in details in section 3.4.

Note: After the initialization step, the user can change, if necessary, the value of any field of the input parameter Fortran structure before calling the `compute_aaaa` subroutine (see below the section 2.3.3 and the section 3.4).

2.3.2. Structure of output data

The output 4A/OP-sub data are gathered in the two different structures containing the scalar mode (`Results`) data and the polarized (`Results_p`) data if the polarization mode is activated.

```
  ! Results
  type :: Results
     :: iadcnv              ! number of convolved points
     :: crad                ! convolved radiance spectrum
     :: csun                ! convolved sun radiance
     :: cnu                 ! convolved wave number grid
     :: ctjac               ! jacobians in temperature
     :: cdto                ! convolved transmittance (type Transmittances)
     :: cpjac               ! jacobians in pressure
     :: cajac               ! jacobians in aerosol optical thickness
     :: ccjac               ! jacobians in molecule concentration
```

```
    :: csjac               ! jacobians in surface emissivity/reflectance
    :: ptop                ! pressure of top of atmosphere
    :: numsel, jacsel      ! molecule selection for radiance and jacobian computation
    :: ndsd, ndsf          ! number of the atlases spectral domain (start and stop)
    :: irds, ilds, nmax    ! record index, record length, spectrum sample number
    :: omin, omax          ! spectral limit (cm-1)
    :: tdown               ! temperature of down surface
    :: respec              ! spectral resolution of high resolution spectrum
    :: level1, level2      ! limit of the optical path
End type Results
```

```
! Results_p
type :: Results_p
    :: nb_stokes           ! number of stoke vecteurs (max = 4)
    :: crad_p              ! polarized convolved radiance spectrum
    :: ccjac_p             ! polarized jacobians in molecule concentration
    :: csjac_p             ! polarized jacobians in surface emissivity/reflectance
    :: cajac_p             ! polarized jacobians in aerosol optical thickness
    :: cdto_p              ! polarized convolved transmittances
    :: csun_p              ! polarized convolved sun radiance
    :: ctjac_p             ! polarized jacobians in temperature
    :: cpjac_p             ! polarized jacobians in pressure
end type Results_p
```

2.3.3. Modification of 4A/OP-Sub structure fields

The interest of the 4A/OP-subroutine code is the possibility to modify a field of the 4A/OP-subroutine input parameter structure inside the fortran code without the need to read a new input data file. This ability might be very useful when the radiative transfer 4A needs to be processed in an iterative scheme. This is the case, for example, of inversion code based on iterative process of the radiative transfer like Optimal Estimation Method.

As an example, we described at the section 3.4 how to modify some fields of the input parameters fortran structure `Params` before and/or after the calling of radiative transfer module `computeaaaa`.

2.4. Directory description

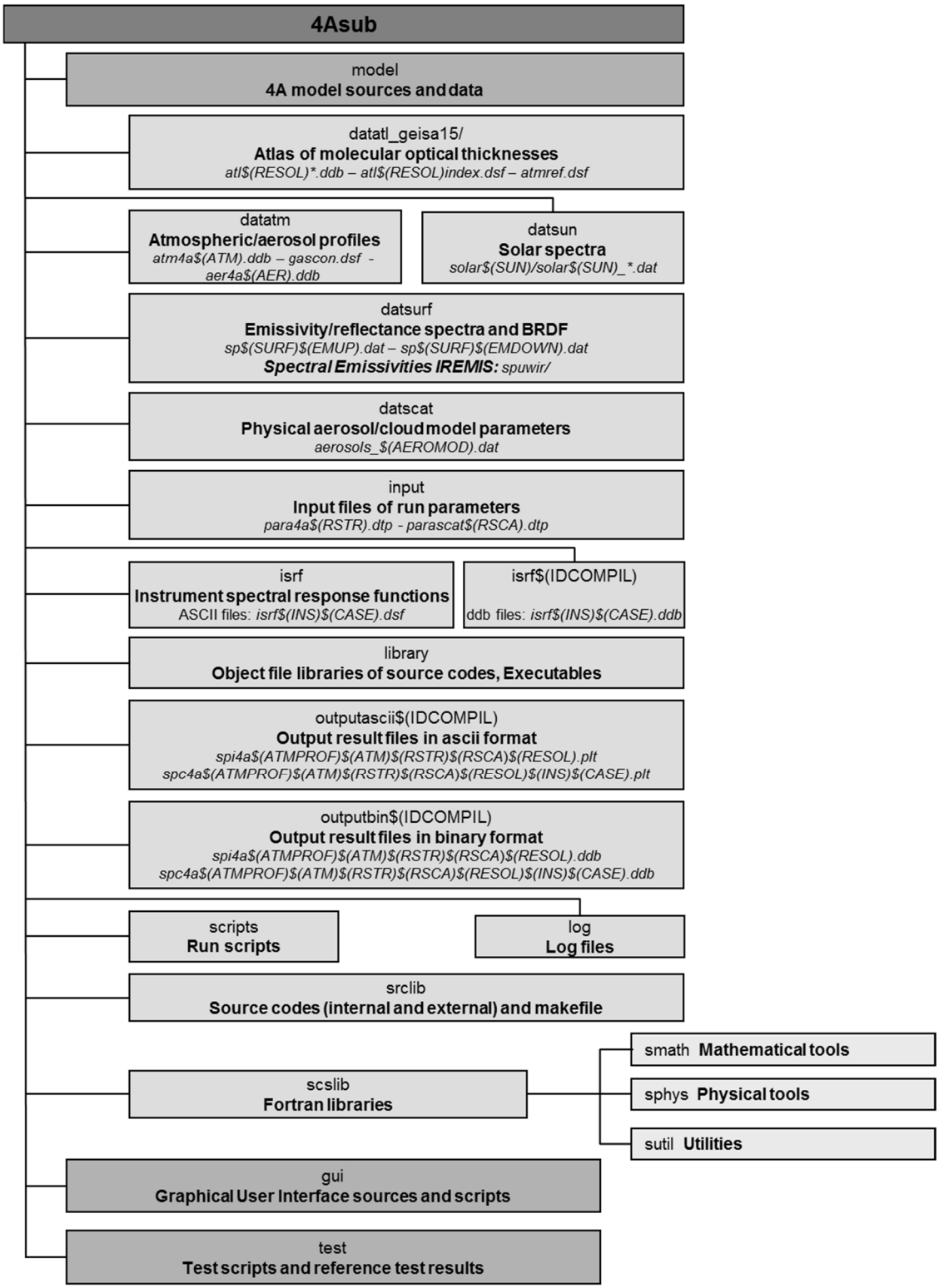

Diagram 1: Directory description of 4A/OP (* for several files).

2.5. Data

2.5.1. Input Data

Optical thicknesses in atlases.

Reference thermodynamical parameters.

Default gas mixing ratio.

Atmospheric profiles.

Simulation definition parameters.

Instrument spectral response functions.

Input data include algorithm fixed input data and user input data.

2.5.1.1. Algorithm fixed input data

Optical thickness atlases.

Reference thermodynamical parameters used for the computation of the atlases.

Default gas mixing ratio.

2.5.1.2. User defined input data

Atmospheric profiles on user-defined pressure levels (see section 3.2.2)

Spectral emissivity (see section 3.2.3).

Instrument spectral response functions (see section 3.2.4) if it is necessary.

Aerosol models and cloud types (see section 3.2.5) if necessary.

Simulation definition parameters (see section 3.2.1).

2.5.2. Output Data

Output data always include radiances and other quantities for an instrument simulation case.

2.5.2.1. "Pseudo-Infinite" case

High-resolution radiance spectra

Jacobians on user-defined pressure layers: partial derivatives of the radiance with respect to the temperature, the gas mixing ratio, the surface emissivity/reflectance, surface temperature, surface pressure, and the aerosol optical thickness.

Transmittances on user-defined pressure layers: transmittances of the main, the secondary and the solar view path. The secondary view path transmittances (atmospheric emission reflected from the reflexion layer) are output only when there is a secondaru view path. The solar view path are only output in the 'DOWNUP' view path and the 'DOWN' view path when the main view path is directly at the Sun (in that case main view path = solar view path). Transmittances are defined per each level (prior v1.5 they were cumulative starting from the observer point).

2.5.2.2. "Instrument" case

Convolved radiance spectra.

Jacobians on user-defined pressure layers: partial derivatives of the radiance with respect to the temperature, the gas mixing ratio, the surface emissivity/reflectance, surface temperature, surface pressure, and the aerosol optical thickness.

Transmittances on user-defined pressure layers: transmittances of the main, the secondary and the solar view path. The secondary view path transmittances (atmospheric emission reflected from the reflexion layer) are output

only when there is a secondaru view path. The solar view path are only output in the 'DOWNUP' view path and the 'DOWN' view path when the main view path is directly at the Sun (in that case main view path = solar view path). Transmittances are defined per each level (prior v1.5 they were cumulative starting from the observer point).

2.6. Description of the Software Process Management

The operator management is performed by the way of the Unix software Make. Management controls are grouped in the files makefile describe below.

2.6.1. Make description

The processing of the operator sequence is managed by the software Make. This software allows the administration of executables by creating a temporal dependency between executables. The compilation is performed if the date of the last modification of the source file is posterior to the date of the executable creation.

Because of its convenient faculties, the software Make is also used for the execution. It makes the execution of any operator possible if its output interface file does not exist, or has been created at a date anterior to the creation date of its input interface file. This way of using the software Make requires a good definition of the interface file names.

2.6.2. The file Makefile

The file makefile organises the hierarchic execution of dependent operators from the building of instrument functions and atmospheric profiles, if that is required, to the radiative transfer computation by using the 4A method.

The set of dependencies linking the interface files between the main operators is grouped in only one main file makefile. When the creation date of the output interface file of a main operator is anterior to the one of the input interface file, this main operator runs after the required associated operators have been executed.

Variables describing the main operator state are initialised in the main file makefile and in the run shell-script.

Table 2: description of the directory of files makefile

Primary directories	File Names	Contents
srclib	makefile	Execution makefile of the main operator 4A: manages file's links and executions. It also manages compilation by using the instructions include Makefile.4a and include makefile_ifct_atm
	Makefile.4a / Makefile.4a$(IDCOMPIL).inc	Compilation makefile: manages the whole compilation and creates libraries of object files. The variable IDCOMPIL is the identifier for the library built.
	makefile_ifct_atm	Compilation makefile for the formatting programs of the atmospheric profiles and instrument functions (atmformat and ifct) and for the reading and ASCII writing programs of 4A output (lirespc4a and lirespi4a)

The user defines the variables related to the experiment in a run shell-script that are passed on to the execution makefile and also by setting up the required parameter files corresponding to the user input variables. A shell-script runs controls of the main file makefile. A target is specified in the script to select the output level of the code. Once the run is finished, the user can examine the result files.

Each operator is linked with files of parameters and data (measured, estimated or specified). The following main source dependency tree (Diagram 2) describes the dependency between main processes.

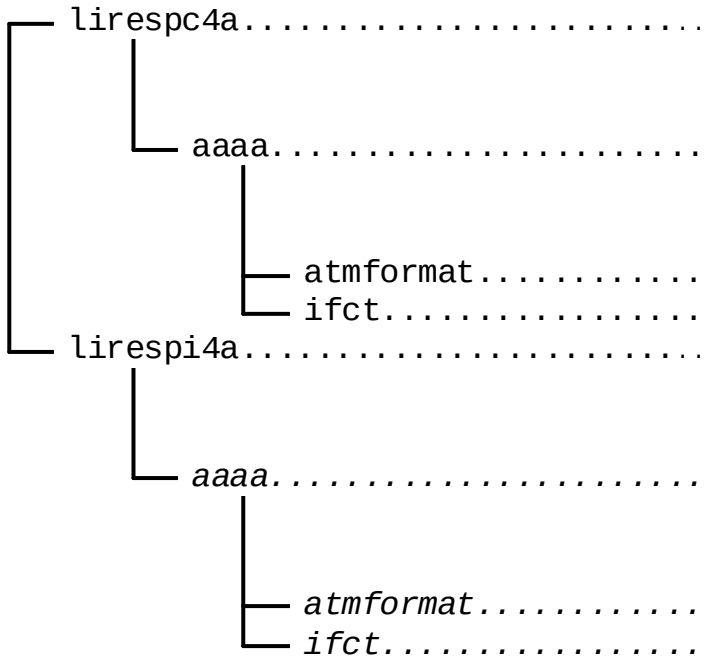

Diagram 2: Main source dependency tree of the software 4A/OP.

3. HOW TO USE 4A/OP

4A/OP processing is controlled by a set of global variables that identifies required input data files and files of run parameters. The user defines these variables through a script file and they are passed on to the execution makefile.

The execution principle is presented first. User input files, global variables and parameters are described in section 3.2. The following section presents the output results.

In the variable tables, the "Type" indication follows the notation described below:

int	Integer (4bytes)
r	Single precision real (4bytes)
d	Double precision real (8bytes)
char string	Character Character string

The "Type" field may also define the name of a structure which is describe in the part 2.3.

In the variable tables, the "Source/Destinat°" indication follows the notation described below:

Ini	Initialisation file (read via makefile)

Convention for additional notations:

NA	Not Applicable

3.1. General

3.1.1. Execution

3.1.1.1. Using the GUI

After 4A/OP has been successfully installed, the program `mainaaaa` can be easily executed either by using the Graphical User Interface or by typing the file name of a run shell script. See the file `README` in the user installation directory.

The simplest way to use 4A/OP is to launch its Graphical User Interface. Under the GUI path (gui/), write:

```
./src/4Agui
```

This program is in the `gui/src` directory of the system installation. Make sure this directory is in your PATH or launch `4Agui` from this directory. See document [DR1].

3.1.1.2. Using the script mode

If the user wants to use the software 4A/OP "by hand", the user can use the script `run4a_example` as an example:

```
./run4a_example
```

in directory `model/scripts/` or by invoking `run4a_example` by using the complete path to `scripts/` whilst being in a different directory.

Before starting the program, the user has to manually adjust this script file for other simulation conditions and to set up several user interface files in order to specify the simulation conditions. These user interface files are described in the next section.

In the running script, a target is specified in order to point at the process that the user wants to execute (`make target`). Presently, in order to compute the radiative transfer and produce ASCII result files, the target name is:

in the case of "infinite" resolution spectra or convolved with an instrument function:

```
make runliresp4a
```

The distinction between "infinite" and convolved spectra is done by setting the INS variable:

- INS=inf for the "infinite" resolution spectra;
- INS is set to the name of the instrument function (ISRF) for the convolved spectra; for example: iasi1c, gauss, modis,…

If the user wants BINARY output only, the corresponding instruction is:

```
make runsp4a
```

The entire instruction used in the script is (for a convolved spectrum):

```
make -f ../srclib/makefile runliresp4a INS=$INS CASE=$CASE ATMPROF=$ATMPROF ATM=$ATM
RSTR=$RSTR RSCA=$RSCA NUMIN=$NUMIN NUMAX=$NUMAX UNIT=$UNIT IDCOMPIL=$IDCOMPIL
RESOL=$RESOL RATL=$RATL JOB=$JOB
```

where the list of variables are global variables described in section 3.2.1.

The instruction "make help" (from the directory `model/srclib/`) produces a list of executions available when using the program make.

3.1.2. CPU runtime

The computation time for a spectrum alone is reasonable but depends on the machine used. For Jacobians, it can take a long time to get results. The most time consuming step of the simulation is the convolution. When Jacobian calculation is selected a convolution is done for each molecule and each level, then the computation is significantly longer. The computation time is a function of the number of levels, the number of implied molecular species, the spectral resolution and the instrumental function for the convolution case. Here is some computing time examples for two different machines in the case of a IASI spectrum:

Table 3: CPU runtime examples

Machine	IASI spectrum alone	IASI spectrum + 4 Jacobians
Linux Xeon Bipro 3.4 GHz (32bits - Fedora)	About 35 seconds	About 2.5 minutes
Linux Xeon 7 proc. 3 GHz (64bits -Debian)	About 21 seconds	About 2 minutes
Unix Sun V880 900 MHz	About 110 seconds	About 30 minutes

These values are given for TIR spectra simulation without any scattering process.

Eventually, taking into account the scattering may considerably increase (up to 15 times depending on spectral regions) the computing time with respect to 4A/OP-sub without scattering.

A way to attenuate this slowing down has been to perform a contraction of the optical thicknesses in each layer before calling the multiple scattering programs (DISORT, LIDORT, VLIDORT). This contraction is left to the choice of the user and is defined by the value of the RESPEC parameter (§3.2.1.). Since this value may impact the quality of the results, the value for this parameter has to be carefully chosen.

As a consequence, and in order to correctly and coherently assess the impact of aerosols with respect to any reference "clear sky" situation, it is recommended to compute this "clear sky" radiance using 4A/OP-sub with scattering RESPEC parameter.

3.2. User input

Four user input files are necessary to completely specify all 4A/OP input data. The user has the choice to use the default installed files or to use her/his own files provided that they are compatible with the required input format. For the two binary files (atmosphere state and instrument functions), the default procedure is to install ASCII files that are automatically formatted into binary files via interface subroutines. The user is free to modify these subroutines in order to code her/his own ASCII file format reading.

This section describes the format of the input files that are directly read by the main program 4A.

3.2.1. Simulation definition parameters

3.2.1.1. Global variables

Global variables are initialised through the run script file and are passed to the file makefile. They are listed in the Appendix A of the **Installation and Quick start guide document** [DR1].

3.2.1.2. Run characterization parameters

The file containing the run parameters is named `para4a$RSTR.dtp`, where `$RSTR` is the value of the variable defining the experiment parameters, and it is stored in the directory `input`. An example of such a file is displayed below. Parameters introduced after limb geometry implementation are highlighted in bold letters. Please report to the Appendix A of [DR1] for a description of every parameter.

The modifications in detail:

- The variable TYP_ANG has been removed. The user option to define zenith angles in degrees or as secant values is removed. All angles have now to be defined in degrees.
- The variable ESDIST is removed. Instead of being read the Earth-sun distance is now computed as function of the specified day.
- The variable PHI (relative azimuth angle in degrees) is replaced by the local azimuth angle of the observer PHI_V, and the local solar azimuth angle PHI_S. PHI_V has no function if no scatter effects are simulated (FLAG_SCAT='NO').

```
'AAAA'
' Geometry Identifier (EVIEW/LIMB)                 GEOMID'     'EVIEW'
' Viewing configuration (UP, DOWN, DOWNUP, UPDOWN) TRAJET'     'DOWNUP'
' Upper limit of simulation (hPa/km)                 PZUP'       0.05
' Lower limit of simulation (hPa/km)               PZDOWN'     1013.00
' Observation level (hPa if EVIEW/km if LIMB)       PZOBS'       0.05
' Index for the Emissivity of the upper level       EMUP'       'constant' 1
' Temperature of the upper level                     TUP'       4.
' Surface choice Emissivity/Reflectance (emis/refl)  SURF'      'refl'
' Index for the Emis./Reflect. of the lower level  EMDOWN'      'constant' 0.3
' Temperature of the lower level                    TDOWN'       0.0
' Geometric tangent height (km) (LIMB only)         ZGTAN'       0.
' Viewing "zenith" angle (deg)                      TETA1'       0.
' Reflexion/emission angle (deg)                    TETA2'      55.
' Viewing azimuth angle (deg) (scattering case only) PHIV'       0.
' Sun angle geometry definition (OWN/GEO)     TYP_ANG_SUN'     'GEO'
' Solar zenith angle (deg) (OWN only)               TETAS'      30.
' Solar azimuth angle (deg) (OWN only)               PHIS'       0.
' Observation date (yymmdd)                         DATE'   '110402'
' Observation hour (decimal hour UT: 0:24) (GEO only)  TU'     10.0
' Latitude (deg) (-90:90)                            LAT'     43.6
' Longitude (deg) (-180:180) (GEO only)              LON'      1.6
' Atmospheric Refraction                           REFRA'     'NO'
' Altitude of lowest atm. pressure level (km)     ZATMBOT'       0.
' Continua (H2O, N2, O2) contribution              CONTIN'     'NO'
' Spectral resolution                              RESPEC'       0.
```

```
' First wave number of the simulation                FORIG'      5140.
' Last wave number of the simulation                 FSTOP'      5150.
' Width of each Atlas spectral band                  DELDS'       15.
' Optical thickness interpolation in temperature    INTOPT'     'YES'
' Indicator of hole algorithm for the convolution     HOLE'      'NO'
' Spectral shift introduction due to the ISRF        SHIFT'      'NO'
' Value of the global spectral shift                   RNU'        0.
' Transmittance calculation flag                     FPOID'      'NO'
' Jacobian calculation flag                          JACOB'      'NO'
' Selection indexes for the 40 gases of the GEISA database (1 or 0)   IDSEL'
```

- Variables SECTTA1, SECTTA2, SETTAS are renamed TETA1, TETA2, TETAS respectively. Their specification in degrees is now obligatory.

- A new option (parameter TYP_ANG_SUN) is implemented. Possible settings are (a) 'OWN' if the user specifies the local solar coordinates (TETAS, PHIS) directly, (b) 'GEO' if the user specifies date, hour and geographical coordinates (DATE, TU, LAT, LON). In option (b), TETAS and PHIS are computed in the code.

- The new variable DATE specifies the day of the simulation in the form 'yymmdd'. For instance the specified year is not exploited (all years are considered identical). DATE specification is mandatory for the computation of the Earth-Sun distance and of the Doppler shift of the solar irradiance spectrum.

- The new variables TU, LAT, LON specify Universal Time in hours, latitude and longitude in degrees, respectively. In option 'GEO', these are inputs to the computation of the local solar coordinates. While TU and LON have no function in option 'OWN', the latitude LAT has to be specified in all cases, since LAT is an input to the computation of the Earth rotation speed component towards the sun.

Note: In option 'OWN', the user is allowed specifying a combination of the parameters TETAS, PHIS, DATE and LAT that cannot happen: While any combination of TETAS and PHIS happens continuously somewhere on the Earth, there is no choice for the latitude where a given combination happens at a given day.

The complete list of the input parameters is given at the Table 14 of the **Installation and Quick start guide document** [DR1].

- **Viewing and geometric configuration definition**

This section defines the terminology used for the viewing and geometric configuration of the radiative transfer computation in Table 4. The various viewing configurations are illustrated in the document "Installation and Quickstart User Guide".

Table 4: Terminology for the viewing and geometric configuration of 4A/OP.

<table>
<tr><th>VIEW PATH</th><th>TETA1</th><th>TETA2</th><th>TETAS</th></tr>
<tr><td>Earth View-UP</td><td>Viewing (sat.) zenith angle
(at surface z=0)</td><td rowspan="2">-</td><td>-</td></tr>
<tr><td>Earth View-DOWN</td><td>Viewing (obs.) zenith angle
(at surface z=0)</td><td rowspan="2">Solar zenith angle
(at surface z=0)</td></tr>
<tr><td>Earth View-DOWNUP</td><td>Viewing (sat.) zenith angle
(at surface z=0)</td><td rowspan="2">Reflexion incidence angle
(scaled to surface z=0)</td></tr>
<tr><td>Earth View-UPDOWN</td><td>Viewing (obs.) zenith angle
(at surface z=0)</td><td>-</td></tr>
<tr><td>Limb-UP</td><td rowspan="4">< 1 or ≥ 90°

zgtan must be specified</td><td>-</td><td>-</td></tr>
<tr><td>Limb-DOWN</td><td>-</td><td>-</td></tr>
<tr><td>Limb-DOWNUP</td><td>55° incidence zenith angle at cloud top
(cloud above tangent point)</td><td>solar zenith angle at cloud top
(cloud above tangent point)</td></tr>
<tr><td>Limb-DOWNUP</td><td>set to teta1
(limb view through tangent point)</td><td>-</td></tr>
</table>

VIEW PATH	TETA1	TETA2	TETAS
Limb-UPDOWN		55° incidence nadir angle at cloud bottom (cloud above tangent point)	-

Important note 1: User-defined angles need to be scaled to surface at z=0 in order to take into account spherical atmospheres (see document [DR2]). The zenith angle at ground (z=0) is given by:

$$\theta_0 = \arcsin [(R+z)/R \sin\theta(z)]$$

where R is the Earth radius and $\theta(z)$ the zenith angle as function of altitude z.

Important note 2: If the user has specified in configuration DOWN a solar zenith angle different from the viewing zenith angle, no sunlight is observed if the sun is completely outside the instrument's field of view. The simulation result will be identical to that obtained in nighttime conditions.
If the user has specified in configuration DOWN a solar zenith angle equal to the viewing zenith angle, the simulation considers that the instrument's field of view is fully occupied by the solar disk (TETAS is set to 90, TUP to 6000K and EMUP to 100). If the user wishes to simulate a situation where the solar disk partly covers the instrument's field of view, he has to adjust the following specifications:

- Specify TETAS as for a nightime situation;
- Specify the upper temperature TUP to the solar surface temperature (6000K);
- Specify the upper emissivity EMUP according to the fraction of the solar disk within the instrument's field of view. Do not forget to generate the file spemis$(EMUP).dat if it does not exist already.

- **Meteorological parameters**

A new possibility is given to the user to handle some input paramerers to be defined from a meteorological profile. These parameters are: TETA1, PHIV, TETAS, PHIS, ZGTAN, ZATMBOT, DATE, LAT, LON, EMUP, EMDOWN. They have to be set in the parameter file `para4a$RSTR.dtp`. In the case of using a meteorological profile, the user has to define the parameter in the header of the atmospheric profile and set to -999 this parameter in the parameter file.

The use of the index for the emissivity of the upper and lower level have been modified. For both of the fields 'EMDOWN' and 'EMUP' you have to define one or two parameters. The first one precise how the emissivity is defined and the second, if it exists, the value of the emissivity.

- For the upper level (top of of the atmosphere), there is four different ways to fill the parameter EMUP:
  - 'constant' value
  - 'snyder' index
  - 'user' character string

- For the lower level (surface), the definition of the parameter EMDOWN depends on the surface property:
  - If the surface property is 'emis', then five different ways are possible to define the parameter:
    - 'constant' value
    - 'snyder' index
    - 'atmsph'
    - 'uwiremis'
    - 'user'
  - If the surface property is 'refl':
    - 'constant' value
    - 'user' character string
  - If the surface property is 'brdf':
    - 'user' character string

The operation mode for this options are the following:

- 'constant' value: the written value is directly read by 4A/OP as a constant emissivity value for all the wave numbers of the simulation. Therefore there is no corresponding emissivity file in the repertory datsurf/.
- 'snyder' index: the index corresponds to a emissivity class associated to a infrared surface emissivity spectrum deduced from Snyder *et al.* [DR4]. These parameters must match with the emissivity file 'spemis_snyderXX.dat' in the repertory datsurf/.
- 'atmsph': the atmospheric profile is read by 4A/OP in the atmospheric file defined by the user. Then the emissivity file 'spemis_atmsphXXXX.dat' is used, with XXXX the profile number. It contains spectral variable emissivities given on a range of wave numbers.
- 'user' character string: the user could set her/his own emissivity file. To this end, she/he has to indicate a character string in the file containing the run parameters and create the related emissivity file in datsurf/ : 'spemisX.dat' (or 'spreflX.dat' or 'spbrdfX.dat'), where X is the character string.
- 'uwiremis': the emissivity file is taken from the IREMIS database in the repertory datsurf/spuwir. The choosen day of the simulation has to correspond to the day of the file used.

Example:

If the atmospheric profile file includes the DATE, LAT and LON parameters, so these parametes are set to -999 in the `para4a$RSTR.dtp` file.

For the preparation of the atmospheric profile file please report to the section 3.2.2.

3.2.1.3. Aerosol / ice cloud run characterization parameters

- **parascat$RSCA.dtp**

The variable `$RSCA` initialized through the makefile identifies the parameter file `parascat$RSCA.dtp` that contains the aerosol/ice cloud microphysical and optical user parameterisations.

Specifying `$RSCA` to the empty string, 4A/OP is executed in the same way as the reference code by reading the file `parascat.dtp`, delivered with the software package.

The file containing the user defined aerosol/ice cloud parameters (`parascat$RSCA.dtp`) is stored in the directory `input`. An example is displayed below. The file is read by the routine `parascat.f90`. The number of read lines depends on the value of FLAG_SCAT. If equal to 'NO', only the first line is read, if equal to 'SOS', the first eight lines are read, otherwise (FLAG_SCAT equal to 'DIS', 'LID' or 'VLI'), all 20 lines must be specified.

In case that `FLAG_SCAT` is equal to 'DIS' or 'SOS', the 5th line specifies the reference wavelength for which the aerosol optical thickness is specified.

The aerosol/ice profile is read in an external file in the `datatm` sub-directory : `aer$AER.dsf` where the global variable `$AER` is defined in the run shell script. The following variables are read in this external file: `n_lay_aero`, `aot_tot/aot_ref`, `p_lay_aero`, `aot_rwl`, `omeaero`, `gaero`. In this case the variable `aeromod` is read in this file.

> The values of `aeromod` are : '`undef`', '`powonu`' or the name of the aerosol/cirrus OPAC model which defines the files in the `datscat` sub-directory: `aerosols_$AEROMOD.dat`.

Each aerosol layer is defined by its upper and lower pressure limits (see line 6 in the example above).

And it is important to note that the reference AOT refers now to the entire aerosol layer, while in former versions it was applied to each atmospheric layer constituting an aerosol layer.

```
' Disort(DIS),SOS(SOS),Lidort(LID),Vlidort(VLI) or no scat(NO)   FLAG_SCAT'        VLI
' VLI Output radiance component: total(0), direct(1), diffuse(2)  RAD_COMP'        0
' Flag for the scattering mode: total(0), single(1), multiple(2) SCAT_MODE'        0
' Flag for FAST-SCATTERING (0=NO, 1=YES)                          FLAG_FAST'         0
' Speed level of FAST-SCATTERING (1->6 default:inf=2/conv=5)      LEVEL_FAST'        0
' AOT of profile                                                  AOT_RESCALE'  0.2
' Power of nu (powonu) Model Paramameter k : AOT(nu)*(nu0/nu)^k  K_AERO'           0
```

```
' Number of DIS/LID phase function Legendre coeffs.              N_MOM'            16
' Number of DIS/LID phase function Legendre coeffs for LSI LOW mode  N_MOM_LOW'    4
' Rayleigh scattering simulation flag (1=YES)                    FLAG_RAY'         1
' Polarisation flag (0=No, 1=Yes, 2=Yes_no_circular)             FLAG_POL'         0
' Number of DIS/LID streams (even and >2)                        N_STREAM'         8
' Number of DIS/LID streams (even and >2) for LSI LOW mode       N_STREAM_LOW'     2
' DIS/LID conv. crit. for beam source ]0.d0,0.1d0]               ACCUR'            0.009
```

Equivalence between the atmospheric layer and the pressure levels:

This is useful to help to make the link between the previous and current version of 4A/OP.

Aerosol layers are defined by the upper and lower pressure limits instead of the upper and lower atmospheric layer number. The atmospheric layer is defined by two atmospheric pressure limits: level 1 (up) and level 2 (bottom). Thus, the atmospheric layer *i* is defined between the pressure level p_i and p_{i+1}. For instance: the atmospheric layer 35 has to be defined by the two pressure limits p_{35}=724.78hPa and p_{36}=800hPa.

Warning:

This warning concerns the **use of the RESPEC parameter** (in the para4a$RSTR.dtp file) in the case of a run requiring the use of **4AOP with DISORT/LIDORT/VLIDORT** (in order to include scattering for aerosol contribution for example). It is recalled that, in script mode, this can be done by setting up the parameter RSCA as follows: RSCA ≠ ' ' (empty string).

Eventually, the call of DISORT/LIDORT/VLIDORT may considerably increase (up to 15 times depending on spectral regions) the computing time with respect to 4AOP in stand alone.

A way to attenuate this slowing down has been to perform a contraction of the optical thicknesses in each layer before calling DISORT/LIDORT/VLIDORT. This contraction is left to the choice of the user and is defined by the value of the RESPEC parameter. Since this value may impact the quality of the results, the value for this parameter has to be carefully chosen.

For indication only:

- A contraction at 0.02 cm^{-1} (RESPEC=0.02) is an acceptable compromise between computing time (about 3 times slower "only" than 4AOP) and accuracy for "window" channels but can lead to errors greater than 5K in channels sounding the upper part of the atmosphere.
- A contraction of 0.002 cm^{-1} (RESPEC=0.002) increases the computing time by about 15 times, but provides a good accuracy for all the wavelengths range (error smaller than 0.2K, except in region at 15 µm, where it can reach 0.5K).

As a consequence, and in order to correctly and coherently assess the impact of aerosols with respect to any reference "clear sky" situation, it is recommended to compute this "clear sky" radiance using 4AOP with DISORT/LIDORT/VLIDORT. (ie RSCA ≠ ' ')

Warning (2): SZA close to 0

The user has to be prevent that some particular solar zenith angles are forbidden when using LIDORT or VLIDORT. The empirical value of 0.13° has been found to be the limit below that the results of the simulation give NaN. A warning message display in such a case and the run stop. So it is recommended to specify SZA greater than 0.013°. However, this value is not warranty to be fixed for any geophysical configuration. In general any values of SZA too close to 0 using LIDORT or VLIDORT could generate bad results.

SCATFAST mode:

A way to accelerate the computing time in case of the use of 4AOP in scattering mode (FLAG_SCAT≠NO) has been recently implemented. The acceleration mode, called SCAT-FAST mode, is implemented in 4A/OP since the version 2016.1.0. It currently works for LIDORT only (validation for VLIDORT is in progress). The user can report to the section 3.1.3.2 of [DR1].

SCATTERING TYPE:

A new option is available in the list of parameter to control the type of scattering calculation. The parameter SCAT_MODE allows the user to choose between 3 modes: total (0), single (1) or multiple (2) scattering. This is

compatible with LIDORT and VLIDORT. Note that the sum of single and multiple scattering are equal to total scattering.

- **aerosols_$(AEROMOD).dtp**

Line 18 of the aerosol single scattering data files contains a layer number, which has been used as standard layer where aerosol is simulated if the user has specified a layer number 0.
This functionality is maintained despite switching to the specification of pressure limits to characterise an aerosol layer.
The original files contain in line 20 the upper pressure limit of the reference layer number specified line 18. By consequence, we have added in all files the lower limit.

Example lines 19, 20 for aerosol model "inso00".

New:

```
# Pressure limits of the aerosols layer
# 584.80 651.04 [hPa]
```

Old:

```
# Mean pressure of the aerosols layer
# 584.80 [hPa]
```

This modification is relevant only in the case that the user specifies 0 hPa as one of the pressure limits of any aerosol layer (see document [DR3] function "readaerosols"). Otherwise, both versions of the aerosol single scattering data files can be used.

3.2.1.4. Parameters for polarization simulation

In case of scattering simulation, it is possible to use the code Vlidort in order to simulate the radiative transfer for polarized light. Then, the user has to set the FLAG_SCAT variable equal to `'VLI'`. The choice of the simulation of the polarization is done by the FLAG_POL parameter in the **`parascat$RSCA.dtp`** file.

An example of parametrization of scattering by aerosols is displayed below:

```
' Disort(DIS),SOS(SOS),Lidort(LID),Vlidort(VLI) or no scat(NO)   FLAG_SCAT'        VLI
' VLI Output radiance component: total(0), direct(1), diffuse(2)  RAD_COMP'        0
' Flag for the scattering mode: total(0), single(1), multiple(2) SCAT_MODE'        0
' Flag for FAST-SCATTERING (0=NO, 1=YES)                          FLAG_FAST'        0
' Speed level of FAST-SCATTERING (1->6 default:inf=2/conv=5)      LEVEL_FAST'       0
' AOT of profile                                                  AOT_RESCALE'  0.2
' Power of nu (powonu) Model Paramameter k : AOT(nu)*(nu0/nu)^k  K_AERO'           0
' Number of DIS/LID phase function Legendre coeffs.              N_MOM'            16
' Number of DIS/LID phase function Legendre coeffs for LSI LOW mode  N_MOM_LOW'    4
' Rayleigh scattering simulation flag (1=YES)                    FLAG_RAY'         1
' Polarisation flag (0=No, 1=Yes, 2=Yes_no_circular)             FLAG_POL'         0
' Number of DIS/LID streams (even and >2)                        N_STREAM'         8
' Number of DIS/LID streams (even and >2) for LSI LOW mode       N_STREAM_LOW'     2
' DIS/LID conv. crit. for beam source ]0.d0,0.1d0]               ACCUR'            0.009
```

In the case of polarisation simulation with the Vlidort code, it is necessary to characterize the aerosol optical parameters in polarization mode for the aerosol/cirrus type considered. Several input parameter files (`datscat/aerosols_XXXX.dat`) are already present in the delivered 4A package and contain all the necessary input parameters in order to simulate the polarized components of the scattering by a given aerosols/cirrus.

The user can choose between several types of aerosols/cirrus by defining, in the input aerosol profile file `aer4a$AER.dsf`, the XXXX variable corresponding to the adequate aerosol/cirrus model file. The user can also create its own aerosol input parameter file by providing a new parameter file with the appropriate format.

3.2.2. Atmospheric profiles

The file describing the atmosphere profiles used for a 4A run are placed in the directory **`datatm`**.

3.2.2.1. Content

The file of atmospheric profiles contains necessarily the user-defined temperature profile and the corresponding pressure levels. The surface pressure and the surface temperature are also specified. The default installed file contains humidity and ozone profiles too.

The user is free to specify any gas mixing ratio profile provided that this gas is defined in the GEISA database (see appendix C in document [DR1]). If the user defines any gas mixing ratio profile and if the user wants to simulate the radiative transfer by including the effect of some gases, the gas mixing ratios are defined through the default gas concentration file `gascon.dsf` (placed in the same directory `datatm`).

For a satellite measurement simulation from a radiosonde atmospheric profile, we recommend the user to extend the profile until 0.1hPa at least in order to avoid wrong extrapolations in the radiative transfer computation. This extend can be made by using profile shapes from climatology databases for example.

Since the 4A/OP 2016.1.0 version, it is possible to give more information in the atmospheric input file, like typically meteorological profile data. Indeed, it is possible to give all the temporal and spatial information related to a profile. The parameters for the geometry, the surface, geographic coordinates and the date linked to the profile can be added in the header of the file. The following parameters are concerned: TETA1, PHIV, TETAS, PHIS, DATE, LAT, LON, ZGTAN, ZATMBOT, EMUP, EMDOWN. These parameters are set by default in the parameter file `para4a$RSTR.dtp`. In the case of defining them in the atmospheric input file atm4a$ATM.dsf, they have to be set to -999 in the parameter file `para4a$RSTR.dtp` (see section 3.2.1).

Since the 4A/OP 1.4 version, it is possible to give atmospheric profile define on pressure layers instead of pressure levels. In this case, the file contains pressure levels, temperature and mixing ratio profiles defined on pressure layers.

The database is stored in a file named: **`atm4a$ATM.ddb`**.

`$ATM` = extension index of the database type given by the user (default possible values are test, test2, satigr_v4.0_moyclas, etc: see section 3.2.1).

3.2.2.2. Format

The sub-routine `atmsph` allows reading this kind of files. It is placed in the directory **`srclib`** [DR3]. The program `atmformat` [DR3] together with the subroutine `lecatmdsf` [DR3] support the user for the creation of these files in the appropriate format from a user-defined file `atm4A$ATM.dsf`.

The atmosphere binary files are formatted according to the following characteristics:

Suffix: `.ddb`;

Access: direct binary;

Length of a record block (buffer): 1024 words;

Length of a word: 32 bits for real variables and for integer variables;

Header records describe the file structure.

These files are composed of several first header records of maximum length 1024*4 bytes and a variable number of record blocks dependent on the number of pressure levels. Numbers in parentheses (1), (2), (3) and (4) indicate the real values of the corresponding variables in the used default files, respectively natm, 1+i, 1+natm+i and 1 + 2*natm +i. Variable names in parentheses (nlevel), (nbcor), etc are the names used in the sub-routine `atmsph` (see their definition [DR3]) and in the program `atmformat`.

Table 5: format description of atmospheric database files for level profiles

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
1	natm (1)	Number of atmospheric profiles	-	1	Int	4
1 + i* (2)	iloc	Index number of the first record block	-	natm (3072 max)	Int	4

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
1 + natm + i* (3)	ilas	Number of significant words in the last record block	-	natm (3072 max)	Int	4
1 + 2*natm + i* (4)	nben	Number of record blocks	-	natm (3072 max)	Int	4
7+j**	buff	Profile characteristics	-	nmbuf	Real	4*dim
	buff(1)	Number of pressure levels of the profiles (nlevel)	-	1		4
	buff(2)	Number of various gas (nbcor)	-	1		4
	buff(2+nc, nc=1,nbcor)	Gas index (idcor), according to the GEISA database convention	-	2		4*2
	buff(nbcor+3)	Index of the profile (nopro)		1		4
	buff(nbcor+4)	Surface temperature (tdown)	K	1		4
	buff(nbcor+5)	Surface pressure (pzdown)	hPa	1		4
	buff(nbcor+6)	viewing zenith angle (tta_v)	deg	1	Real	4
	buff(nbcor+7)	viewing azimutal angle (phi_v)	deg	1	Real	4
	buff(nbcor+8)	solar zenith angle (tta_s)	deg	1	Real	4
	buff(nbcor+9)	solar azimutal angle (phi_s)	deg	1	Real	4
	buff(nbcor+10)	geometric tangent height (zgtan)	km	1	Real	4
	buff(nbcor+11)	altitude of the lowest pressure level (zatmbot)	km	1	Real	4
	buff(nbcor+12)	longitude*100 (lon)	-	1	Int	4
	buff(nbcor+13)	latitude*100 (lat)	-	1	Int	4
	buff(nbcor+14)	date (date)	yymmdd	1	Int	4
	buff(nbcor+15)	surface emissivity (emdown)	-	1	Int	4
	buff(nbcor+16+ (nl-1)*(2+nbcor), nl=1,nlevel)	Pressure levels (pl)	hPa	nlevel		4*dim
	buff(nbcor+17+ (nl-1)*(2+nbcor), nl=1,nlevel)	Temperature profile (tl)	K	nlevel		4*dim
	buff(nbcor+17+nc +(nl-1) *(2+nbcor), nl=1,nlevel,	Gas concentration profile (rol)	g/g	nlevel*nbcor		4*dim

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
	nc=1,nbcor)					

* $1 \leq i \leq (natm)/1024 + 1$; ** $1 \leq j \leq (nmbuf)/1024 + 1$, $nmbuf = nlevel*(nbcor+2)+nbcor+5$

Table 6: format description of atmospheric database files for layer profiles

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
1	natm (1)	Number of atmospheric profiles	-	1	Int	4
1 + i* (2)	iloc	Index number of the first record block	-	natm (3072 max)	Int	4
1 + natm + i* (3)	ilas	Number of significant words in the last record block	-	natm (3072 max)	Int	4
1 + 2*natm + i* (4)	nben	Number of record blocks	-	natm (3072 max)	Int	4
7+j**	buff	Profile characteristics	-	nmbuf	Real	4*dim
	buff(1)	Number of pressure layers of the profiles (nlayer)	-	1		4
	buff(2)	Number of various gas (nbcor)	-	1		4
	buff(2+nc, nc=1,nbcor)	Gas index (idcor), according to the GEISA database convention	-	2		4*2
	buff(nbcor+3)	Index of the profile (nopro)		1		4
	buff(nbcor+4)	Surface temperature (tdown)	K	1		4
	buff(nbcor+5)	Surface pressure (pzdown)	hPa	1		4
	buff(nbcor+6)	viewing zenith angle (tta_v)	deg	1	Real	4
	buff(nbcor+7)	viewing azimutal angle (phi_v)	deg	1	Real	4
	buff(nbcor+8)	solar zenith angle (tta_s)	deg	1	Real	4
	buff(nbcor+9)	solar azimutal angle (phi_s)	deg	1	Real	4
	buff(nbcor+10)	geometric tangent height (zgtan)	km	1	Real	4
	buff(nbcor+11)	altitude of the lowest pressure level (zatmbot)	km	1	Real	4
	buff(nbcor+12)	longitude*100 (lon)	-	1	Int	4
	buff(nbcor+13)	latitude*100 (lat)	-	1	Int	4
	buff(nbcor+14)	date (date)	yymmdd	1	Int	4

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
	buff(nbcor+15)	surface emissivity (emdown)	-	1	Int	4
	buff(nbcor+16+ (nl-1)*(2+nbcor), nl=1,nlayer)	Lower pressure level limits (pl)	hPa	nlayer		4*dim
	buff(nbcor+17+ (nl-1)*(2+nbcor), nl=1,nlayer)	Upper pressure level limits (pl)	hPa	nlayer		4*dim
	buff(nbcor+18+ (nl-1)*(2+nbcor), nl=1,nlayer)	Temperature profile (tl)	K	nlayer		4*dim
	buff(nbcor+18+nc +(nl-1) *(2+nbcor), nl=1,nlayer, nc=1,nbcor)	Gas concentration profile (rol)	g/g	nlayer*nbcor		4*dim

* $1 \leq i \leq (natm)/1024 + 1$; ** $1 \leq j \leq (nmbuf)/1024 + 1$, nmbuf = nlayer(nbcor+2)+nbcor+5*

3.2.2.3. Generating your own atmospheric profiles

A preprocessing program of 4A/OP-sub, **`atmformat`**, is able to convert user-defined atmospheric profiles written in ASCII files into a binary-formatted database, that is then an input to 4A/OP-sub. This binary database, is named **`atm4a$ATM.ddb`** ($ATM has been defined in the global variables, section 3.2.1.1) and placed in the directory **`model/datatm.`**

By default, 4A/OP-sub uses the file **`atm4asatigr_v4.0_moyclas.ddb`** built as follows:

- the default pressure levels come from file **`pressions4A.dsf`**, placed in the **`datatm`** directory of the system installation;
- the reference thermodynamical parameters (temperature, H_2O and O_3) are those from file **`atm4asatigr_v4.0_moyclas.dsf`**, in your **`datatm`** directory;
- the default gas concentrations also come from file **`gascon.dsf`** also in your **`datatm`** directory.

Now, see how to input user-defined atmospheric profiles on user-defined pressure levels or user-defined pressure layers.

Two steps are needed to achieve this. You have to:

(1) define your own ASCII formatted file **`atm4a$ATM.dsf`**;

(2) modify the **`lecatmdsf.f90`** module in your **`srclib`** directory according to your own format.

- **<u>First step: preparation of the user atmosphere file</u>**

This step allows you to prepare your own-designed **`atm4a$ATM.dsf`** file in ASCII format; This file has to be put in your **`datatm`** directory.

This file needs to contain the following parameters (given here with the variable name in Fortran syntax) (see for example in your **`datatm`** directory how is written the ASCII sample file **`atm4asatigr_v4.0_moyclas.dsf`**):

- **`tsol and psol`**: the surface temperature and pressure;
- **`nbcor`**: the number of involved molecules of user-defined gas profiles;

- **idcor(1:nbcor)**: indentication numbers of the molecules (according to the GEISA notation) corresponding to the user-defined gas profiles. You are free to specify any gas mixing ratio profile provided that this gas is defined in the GEISA database;
- **nlevel**: number of pressure levels;
- **p(1:nlevel)**: the user-defined pressure levels in hPa;
- **t(1:nlevel)**: the user-defined temperature profile in Kelvin;
- **rolvl(1:nlevel,idcor(1:nbcor))** : the user-defined gas mixing ratio profiles in g/g.

These parameters can be repeated for several atmospheric profiles, of user-defined number equal to **nopro**.

- **Second step : modification of the appropriate atmosphere reading subroutine**

This step consists in modifying the **lecatmdsf.f90** module to read correctly your **atm4a$ATM.dsf**.
This routine needs to be fed with the parameters defined previously. For example, for sample file corresponding to $ATM='satigr_v4.0_moyclas', following lines have been added in the module:

```
! Atmospheres "tigr"
! ------------------
if (atm == 'satigr_v4.0_moyclas') then
   read (jin1, fmt='(I12,2F8.3,2F6.0)', end=2000)
   read (jin1, fmt='(6E12.5)', end=2000) (t(j),j=1,40)
   read (jin1, fmt='(6E12.5)', end=2000) tsol,psol
   read (jin1, fmt='(6E12.5)', end=2000) (rolvl(i,1),i=1,40)
   read (jin1, fmt='(6E12.5)', end=2000) (rolvl(i,3),i=1,40)
   sectta = 1.
   iat    = iat + 1
   nopro  = iat
   nlevel = 40
end if
```

For file containing profiles on pressure layers, you have to add the flag parameter "flag_pc" to indicate to 4AOP that profiles are defined on pressure layers instead of pressure levels.For example, following lines have been added to the module to read "_layer_inv" atmospheric profile file:

```
! Atmosphere "_layer_inv"
if (atm(1:10) == '_layer_inv' ) then
   print*, 'atmosphere de type _layer_inv'
   nbcor=2
   do nc=1,nbcor
     idcor(nc)=nc
   end do
   !Warning : here, nlevel contains the number of layers, not levels
   read(jin1,*,end=2000) nlevel
   print*, 'nlayer lu', nlevel
   do nl=1,nlevel
     read(jin1,*,end=2000) p(nl),p(nl+1),t(nl),(rolvl(nl,i),i=1,nbcor)
   end do
   print*, 'profil lu'
   tta_v=0._DOUBLE
   tsol=t(nlevel)
   psol=p(nlevel)
   iat=iat+1
   nopro=iat
   flag_pc = 1
end if
```

3.2.3. Spectral emissivity / reflectance / brdf

The files describing the infrared surface property spectrum (emissivity for TIR, reflectance/albedo and BRDF for SWIR and NIR) for different types of surface used for a 4A/OP-sub run are placed in the directory **`model/datsurf.`** The user can choose between 3 types of surface properties by defining the variable **SURF** in the `para4a*.dat` input file:

- Case **`SURF = 'emis'`** : surface property is emissivity
- Case **`SURF = 'refl'`** : surface property is reflectance/albedo
- Case **`SURF = 'brdf'`** : surface property is brdf: **works only with VLIDORT**

Then the variables **`EMDOWN`** and **`EMUP`** indicate the files that will be used for the characterization of the surface (background) and top of atmosphere (foreground) properties.

3.2.3.1. Emissivity data files

3.2.3.1.1 Content

A **`readme.txt`** file explains the emissivity files already available. These files can also be either user-defined or default files can be used. The basis of all the emissivity files is 'spemis'. Three types are possible for surface emissivity files:

- a file containing spectral variable emissivities given on a range of wave numbers.. The standard installation of 4A/OP-sub contains the infrared surface emissivity spectrum deduced from Snyder *et al.* [DR4] and presented in Table 6. Fourteen emissivity classes are defined, associated to the IGBP surface type classification. Thus the file suffix of the emissivity file begins with '_snyder' and is followed by a number which represents the emissivity class.
- A atmospheric emissivity file; it also contains spectral variable emissivities given on a range of wave numbers. Each of this file corresponds to a specific atmospheric profile. That is why the file suffix starts with '_atmsph' and is followed by a value XXXX corresponding to a atmospheric profile.
- You can add in the **`datsurf`** directory your own-defined surface emissivity files, respecting the ASCII format. The basic of the emissivity file remains the same: 'spemis'. Then you have to put the character string you want, for example 'sahara' for the resulting emissivity file: spemissahara.dat

  For new spectral emissivities you can add your own emissivity database, provided that the files are written in the right format (see files **`spemis101.dat`** to **`spemis114.dat`** for examples in your **`datsurf`** directory) and the corresponding emissivity suffixes XXX are **superior to 114**.

Table 6: Emissivity class definition (adapted from Snyder *et al.*, 1998). Season notification: W – winter, Sp – spring, Su – summer, F – fall.

IGBP surface properties classification	Season	Snyder emissivity classes	EmissivityCode
Water (17), Wetlands (11)	all	Water	1
Snow, Ice (15)	all	Snow, Ice	2
Barren/Desert (16)	all	Arid bare soil	3
Crops (12), Tundra (18)	all	Organic bare soil	4
Open shrubs (7)	F,W	Senescent sparse shrubs	5
Open shrubs (7)	Sp,Su	Green sparse shrubs	6
Savannah (9) ; Closed Shrubs, Grasslands, Crop/Mosaic (6,10,14)	W,Sp F,W	Senescent grass savannah	7
Savannah (9) ; Closed Shrubs, Grasslands, Crop/Mosaic (6,10,14)	Su,F Sp,Su	Green Grass Savannah	8
Woody Savannah (8)	F,W	Senescent Woody Savannah	9
Woody Savannah (8)	Sp,Su	Green Woody Savannah	10
Deciduous broadleaf and Mixed forest (4,5)	F,W	Senescent Broadleaf Forest	11
Deciduous broadleaf and Mixed forest (4,5) ; Evergreen Broadleaf Forest (2)	Sp,Su all	Green Broadleaf Forest	12
Deciduous Needle Forest (3)	F,W	Senescent Needle Forest	13
Deciduous Needle Forest (3) ; Evergreen Needle Forest (1)	Sp,Su all	Green Needle Forest	14

3.2.3.1.2 Format

The sub-routine `lireemis` allows reading this kind of files. It is placed in the directory **`srclib`**.

These files are formatted according to the following characteristics:

Suffix: `.dat`;

Access: sequential formatted (ASCII).

These data files are composed of one header line and two columns of a variable number of lines.

Table 7: format description of emissivity spectra

Number of the line	Parameter name	Description	Units	Dimension	Type	Format
1	EmisDim	Number of emissivity channels	-	1	Int	i4
2 to emisDim+1	emis (fisrt column)	Wave numbers	cm^{-1}	emisDim	Real (double)	Free
	emis (second column)	Mean spectral emissivity	-	emisDim	Real (double)	Free
	third to sixth columns	Statistics on emissivity (not read)	-	emisDim	Real (double)	Free

3.2.3.2. Reflectance / albedo data files

In SWIR and NIR spectral regions, reflectance (or albedo) are commonly used to define the surface properties. In the same way as emissivity data files, the user can create her/his own surface reflectance data file The basis of the reflectance (or albedo) file is 'sprefl' ant then the user can add her/his specified character string.

3.2.3.3. BRDF data files

The Bidirectional Reflectance Distribution Function (BRDF) gives the reflectance of a target as a function of illumination geometry and viewing geometry. The BRDF depends on wavelength and is determined by the structural and optical properties of the surface.

The user can also choose to define the surface properties as a BRDF by adding files in the directory **`model/datsurf`** with filename **`spbrdfXXX.dat`**.

Important: the BRDF mode is working only with VLIDORT which is activated by the user by setting the variable FLAG_SCAT in the parascat$(RSCA).dtp parameter file (see section 3.2.1.3).

Table 8: Kernels available in VLIDORT

#	Kernel name	# parameters	type	Source
1	Lambertian	0	Scalar	----
2	Ross-thick	0	Scalar	Wanner *et al.*, 1995
3	Ross-thin	0	Scalar	MODIS
4	Li-sparse	2	Scalar	MODIS
5	Li-dense	2	Scalar	MODIS
6	Roujean	0	Scalar	MODIS
7	Rahman(RPV)	3	Scalar	Rahman *et al.*, 1993
8	Hapke	3	Scalar	Hapke, 1993
9	Cox-Munk	2	Scalar	Cox/Munk, 1954
10	GISS Cox-Munk	2	Vector	Mishchenko/Travis 1997
11	BPDF 2009	2	Vector	Maignan *et al.*, 2009

3.2.3.3.1 Preparation of the input files

The files **`spbrdfXXX.dat`** are formatted in the following form:

- N_BRDF_KERNELS, (k ≤ 3) : kernel number (k) to be combined;
- BRDF_NAMES(k), (k ≤ 3) : name of the kernel to be combined (second colum of Table 8) ;
- WHICH_BRDF(k), (k ≤ 3): index of the kernel to be combined (first colum of Table 8) ;
- N_BRDF_PARAMETERS (k), (k ≤ 3) : number of parameters for each kernel (third colum of Table 8) ;
- BRDF_PARAMETERS(k, N_BRDF_PARAMETERS (k)), (k ≤ 3) : parameters of each kernel, could be set to 0 if not used;
- BRDF_FACTORS(k), (k ≤ 3) : the amplitude of each kernel.

The other parameters used in the BRDF calculation, like the geometry, the number of Stokes components,…, are defined in the two definition parameter files: **`para4a$RSTR.dtp`** and **`parascat$RSCA.dtp`**.

3.2.3.4. IREMIS database

Since version 2016.1.0, 4A/OP includes the **IREMIS database for the surface spectral emissivities** [DR9] . This database can be chosen only from geographic coordinates given by the couple **latitude and longitude and by a date**. These parameters have to be selected in the input parameters file (`para4a$RSTR.dtp`). To use the

IREMIS surface spectral emissivities, the parameter **`EMDOWN` has to be set to -999**. Otherwise, the value set to `EMDOWN` is the emissivity/albedo corresponding to the XXX of the files in the datsurf directory.

The IREMIS database coming along the 4A/OP distribution is limited to the year 2012. In the case of a different year, the user has to download the emissivity IREMIS from the web site http://cimss.ssec.wisc.edu/iremis/. The IREMIS database in 4A is stored in the directory datsurf/spuwir/.

3.2.4. Instrument Spectral Response Function (ISRF)

The file describing the instrument functions used for a 4A run are placed in the directory **`isrf$IDCOMPIL`**. Pre-processing of the instrument functions is quite similar to pre-processing of atmospheric profiles: starting from a user-defined ASCII formatted file **`isrf$(INS)$(CASE).dsf`**, the subroutine **`lecifctdsf`** converts it into a binary file named **`isrf$(INS)$(CASE).ddb`,** that can be interpreted by 4A. This subroutine is contained in the fortran file named **`ifct$(INS)$(CASE).f90`**.

3.2.4.1. Content

The file of instrument functions contains the function characteristics in the file header and the functions as a function of the wave numbers in the file body. The instrument functions are stored in a file named: **`isrf$INS$CASE.ddb`**.

`$INS` = Indicator (name) of the simulated instrument (ISRF) (default possible values are inf, test, test2, test3, iasi1c, etc. ; « inf » for "pseudo-infinite" spectrum (an ISRF file is not required), « iasi1c » for a level 1c IASI spectrum, or « test » and « test2 » for test functions; see test cases in document [DR1] "Appendix B".

`$CASE` = Indicator of the instrument function version (case) for a given instrument defined by the variable `INS` (see section 3.2.1.

Important note: In the file, at least two instrument functions have to be defined at two different wave numbers included in the simulated spectral domain (even if they are the same).

Three types of instruments are possible: radiometer, interferometer or spectrometer. The second category of instrument is divided into two types: a constant sampling step and a non constant sampling step in wavenumber. An example of a radiometer ISRF is contained in the file **`isrftest.dsf`** (in the directory **`model/isrf$IDCOMPIL`**), an example of an interferometer/spectrometer ISRF with a constant sampling step is given by the file **`isrftest2.dsf`** and an example of an interferometer

3.2.4.2. Format

The sub-routine `lireisrf` allows reading this kind of files. It is placed in the directory **`srclib`**. The program `ifct` [DR3] supports the user for the creation of these files in the appropriate format from an user-defined file `isrf$INS$CASE.dsf` (see section 3.2.4.3 below).

These files are formatted according to the following characteristics:

Suffix: `.ddb`;

Access: direct binary;

Length of a record block (buffer): 1024 words;

Length of a word: 32 bits for real variables and for integer variables.

These files are composed of four first header records of maximum length 1024*4 bytes and a variable number of record blocks dependent on the number of instrument functions and their size.

Table 9: format description of instrument function files

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
1	fisrfu	File name of the instrument functions	-	1	String	2
1	titre	Description of the	-	1	String	2

Number of the record	Parameter name	Description	Units	Dimension	Type	Size (bytes)
		instrument functions				
1	nbk	Number of spectral bands	-	1	Int	4
1	itypeconv	Indentifier of the convolution type	-	1	Int	4
2	nufdeb	First central wave number of the ISRFs	cm^{-1}	nbk (3 max)	Real	4*dim
2	nuffin	Last central wave number of the ISRFs	cm^{-1}	nbk (3 max)	Real	4*dim
2	dnds	Spectral step of central wave numbers	cm^{-1}	nbk (3 max)	Real	4*dim
2	nknu	Number of ISRFs	-	nbk (3 max)	Int	4*dim
2	pdnuco	Sampling step of the ISRFs	cm^{-1}	1	Real	4*dim
2	pascon	Convolution sampling step: step of the ISRF application	cm^{-1}	1	Real	4*dim
2	kfmo	Number of description points of the ISRF	-	nknu*nbk (1000*3 max)	Int	4*dim
3	nufo	Central wave numbers of each ISRF	cm^{-1}	nknu*nbk (1000*3 max)	Real	4*dim
3	wnuo	Width of each ISRF	cm^{-1}	nknu*nbk (1000*3 max)	Real	4*dim
4	dnuo	Spectral shift of each ISRF	cm^{-1}	nknu*nbk (1000*3 max)	Real	4*dim
4	irecf	Record number of the beginning of each ISRF	-	nknu*nbk (1000*3 max)	Int	4*dim
4+j*	crp	Instrument functions	-	30001 max	Real	4*dim

* *$1 \leq j \leq kfmo/1024 + 1$, ((kfmo(k,nb), k=1,nknu(nb)),nb=1,nbk)*

Note:

Since the version 4AOP-sub-1.1, the parameter PASCON is included in the isrf configuration file and not anymore in the general parameter file (para4a.dtp). We just recall here some remarks for PASCON: PASCON only used if itypeconv=2; it must be >RESPEC.

3.2.4.3. Generating your own ISRF file

The document [DR3] aims at supporting users to set up the instrument function files. However, some points could remain unclear for new users who are not familiar with Fortran code. That is why the present section proposes to help users to install her/his own ISRF files.

- **First step: preparation of the user ISRF file**

You need to set up your file **`isrf$(INS)$(CASE).dsf`**. Be careful that:

1. The file name has to correspond to the instrument name **`INS`** (and optionally to the version of the instrument specified by the variable CASE). For example, if **`$INS=instru`** and **`$CASE`** not used (empty character string), the name of the file is: **`isrfinstru.dsf`**. It is placed in the directory **`isrf`**;
2. Then it is recommended to include a header in the file as in the ASCII file **`isrftest.dsf`** for example.
    a. First line: number of total functions.
    b. Line above each function: index number of the function, corresponding central wave number, number of description points of the function (number of samplings).
    c. The entire definition spectral domain of the functions has to be included in the limits of the radiative transfer simulation. The entire spectral domain of the functions are defined by the first and last central wave numbers and the function width.
    d. The spectral shift as an option.

- **Second step: preparation of the appropriate ISRF reading subroutine**

The user has to prepare the corresponding reading subroutine **`lecifctdsf`** located in the file named **`ifct$(INS).f90`**. The easiest way is to take as an example an existing subprogram, for example **`ifcttest.f90`**:

1. In the directory **`srclib`**, the user has to copy **`ifcttest.f90`** in **`ifctinstru.f90`**.
2. Then, it is recommended to edit the new file in order to correct some parameters if it is necessary in particular:
    a. the ISRF sampling step: change the variable **`pdnuco`** (0.1) into the user-defined sampling step value.
    b. the convolution type: change the variable **`itypeconv`** into the user-defined convolution type value. The value of **`itypeconv`** is choosen among 3 possibilities: CONV_RADIOMETER, CONV_INTERF_CS and CONV_INTERF_NCS. These constants are defined in the module instruments:
        i. CONV_RADIOMETER: used in case of a radiometer with n channels (Météosat, HIRS…);
        ii. CONV_INTERF_CS: used in case of an interferometer/spectrometer with a constant sampling step in wave number (IASI: step = 0.25cm^{-1},…);
        iii. CONV_INTERF_NCS: used in case of an interferometer/spectrometer with a non constant sampling step in wave number (AIRS,…).

Now, all is ready to compile and create automatically the binary file at the execution of the program 4A by specifying the correct instrument function name ("instru" in our example). Indeed the ISRF formatting program **ifct** (conversion from ASCII to 4A binary format) is compiled with the correct reading subroutine (**`ifct$(INS).f90`**) corresponding to the specified instrument INS. At the moment of the execution of the program 4A the instrument function binary file is created (**`isrfinstru.ddb`**) and passed to 4A as an input.

To compile the new subroutine created the user has to execute the following instruction (under the srclib directory):

```
make installisrf IDCOMPIL=$IDCOMPIL INS=$INS
```

with IDCOMPIL is the name of the library already created under the library/ directory and INS the name of the instrument function created.

As an example:

```
make installisrf IDCOMPIL=.calcul1_pgi_64b INS=iasi1c
```

3.2.4.4. Special case of the wavelength ISRF

Since the version of 4A/OP 2016.1.1, the user is able to use an ISRF defined in wavelength.

If this is the case, the user has to prepare the ISRF material by following the previous steps (see section 3.2.4.3) but with an additional step: the conversion from the wavelength scale to the wavenumber scale.

A new generic subroutine is available to convert the ISRF and the spectral parameters defined in the **Table 9** from wavelength to wavenumber.

This subroutine is called `convifct_wltown` and is defined in the module `mod_wltown.f90`.

The user can follow one of the example coming with the 4A/OP distribution: one parameterised ISRF function (`ifctgauss_lambda.f90`) or one completely defined ISRF function (`ifctoco-2_lambda.f90`).

The main steps to convert the user ISRF defined in wavelength into wavenumber are listed below:

- Reading the user defined ISRF function: input file is `isrf$INS$CASE.dsf` (defined in the `isrf/` subdirectory);
- For the parameterised ISRF case: calculation of the ISRF (Gaussian function for example);
- Determination of the global parameters in wavelength: number of ISRF, number of description points of each ISRF, ISRF sampling step, central wavelength of each ISRF, width of each ISRF;
- Conversion of each ISRF from wavelength to wavenumber grid; call the subroutine `convifct_wltown`: the result is an ISRF function defined on a irregular wavenumber grid;
- If necessary, conversion from irregular grid to regular grid in wavenumber by linear interpolation and normalisation of each ISRF; The user can use the subroutine `get_wltown` to calculate the wavenumber irregular grid from the regular wavelength grid;
- Writing in the output binary file: `isrf$INS$CASE.ddb`.

The user has to take care that some implementation may be necessary before and/or after the call to this subroutine depending on his/her ISRF.

The **Table 10** gives the description of the input and output of the subroutine convifct_wltown.

Table 10: format description of the subroutine convifct_wltown

Parameter name	input/output	Description	Units	Dimension	Type	Size (bytes)
INPUT						
dl_lamb	i	Sampling step of the ISRF in wavelength	µm	1	Real	4
width	i	Definition domain (width) of the ISRF in wavelength	µm	1	Real	4
lamb	i	Central wavelength of the ISRF	µm	1	Real	4
ldim_lamb	i	Number of description points of the ISRF in wavelength	-	1	Int	4
ldimp_lambda	i	Half of the number of description points of the ISRF in wavelength (number of points in the 1/2 width of the ISRF)	-	1	Int	4
isrf_lamb	i	Values of the ISRF in wavelength	-	Cst_kmax	Real	4*dim
ddfr_lambda	i	Spectral shift of the ISRF in wavelength	µm	1	Real	4
dnu	i	Sampling step of the ISRF in wavenumber	cm^{-1}	1	Real	4
OUTPUT						
wnu	o	Definition domain (width) of the ISRF in wavenumber	cm^{-1}	1	Real	4
nu	o	Central wave numbers of the ISRF	cm^{-1}	1	Real	4
kfmo	o	Number of description points of the ISRF in wavenumber	-	11	Int	4
ddfr	o	Spectral shift of the ISRF in wavenumber	cm^{-1}	1	Real	4
pg_irreg	o	Values of the ISRF in wavenumber normalised	-	Cst_kmax	Real	4*dim

3.2.5. Aerosol / ice cloud physical parameters

The files describing aerosol/cloud physical parameters, separately for a given aerosol model/cloud type, are placed in the directory **`model/datscat.`**

3.2.5.1. Content

Aerosol physical parameters are provided in files, separately for a given aerosol model, placed in the directory `datscat`. The nomenclature is `aerosols_$(AEROMOD).dat` , where $(AEROMOD**)** is the identifier of the aerosol model/cloud type, from top to bottom layer ($(AEROMOD**)** should be of maximum 10 characters length). In the directory datscat there are 2 types of models: the aerosol/cirrus OPAC models and the undef/powonu types.

3.2.5.1.1 Aerosol/cirrus OPAC models

Files for 12 aerosol OPAC models are included in the software package:

- inso00 Insoluble;
- micm00 Mineral (coarse mode);
- mitr00 Mineral (transported);
- soot00 Soot;

- ssam70 Sea salt (accumulation), 70% relative humidity;
- ssam95 Sea salt (accumulation), 95% relative humidity;
- sscm70 Sea salt (coalescence), 70% relative humidity;
- sscm95 Sea salt (coalescence), 95% relative humidity;
- suso70 Sulfate, 70% relative humidity;
- suso95 Sulfate, 95% relative humidity;
- waso70 Water soluble, 70% relative humidity;
- waso95 Water soluble, 95% relative humidity.

The user may define new aerosol models but must provide aerosol parameter files in the same format.

Cirrus extinction is modelled in the same way as aerosol extinction. For the user's convenience, cloud physical parameters are provided for 13 different cirrus cloud models/size distributions from two different sources. These files have the same nomenclature as corresponding aerosol physical parameter files `aerosols_$(AEROMOD).dat`.

The spectral coverage of the three ice cloud models from OPAC [DR5] is limited to wave numbers above 1000 cm^{-1} with very poor sampling in the thermal Infrared.

- cir100 Liou size distribution 30-1300 µm at -25°C
- cir200 Liou size distribution 30-1300 µm at -50°C
- cir300 as cir200, with additional small-size particles down to 2 µm

The spectral coverage of the ten selected models from [DR6] based on the size distribution model [DR7] is limited to the range between 780 and 1500 cm^{-1} with good sampling in between.

The selected models correspond to five effective diameters (10, 30, 50, 70, 90 µm) and two shapes (aggregate and columns).

- agg010 Aggregate, effective diameter 10 µm
- …
- agg090 Aggregate, effective diameter 90 µm
- col010 Columns, effective diameter 10 µm
- …
- col090 Columns, effective diameter 90 µm

3.2.5.1.2 Undef/powonu types

In the case the user is interested by simple models, the seletion of 'undef' or 'powonu' types if recommended. Report to section 3.2.1.3.

Since the version 4A/OP 1.2, the 2 types 'undef' and 'powonu' could be use in case of polarization mode (`FLAG_POL≠0`). 4A/OP does not take as input the phase matrix directly but rather the Greek coefficient which will be used to calculate the phase matrix in the VLIDORT software.

The user may define his/her own Greek coefficient in dedicated files. These files will have the same nomenclature as for the 12 aerosol models or cirrus model but has to include the string 'undef' or 'powonu' in their name. The following nomenclature has to be respected:

- case 'undef': `aerosols_undefXXXXX.dat` with `XXXXX` a 5 character string. If `XXXXX` is empty the default files will be read: `aerosol_undef.dat`;
- case 'powonu': `aerosols_powonuYYYY.dat` with `YYYY` a 4 character string. If `YYYY` is empty the default files will be read: `aerosol_powonu.dat`.

These files are placed in the directory `datscat`.

Some example are provided with 4AOP package. These files are: `aerosols_undef.dat` and `aerosols_powonu.dat`. These files only contains the Greek coefficients. The default is the same as the `waso70` aerosol model. The user is free to create new files.

3.2.5.2. Format

3.2.5.2.1 Aerosol/cirrus OPAC models

The sub-routine `readaerosol` allows reading this kind of files. It is placed in the directory **`srclib`.**

These files are formatted according to the following characteristics:

- Suffix: `.dat;`
- Access: sequential formatted (ASCII).

These data files are composed of

- 24 header lines: at the 16th line of the header, the value corresponds to the reference aerosol/cloud optical depth at 1000 cm^{-1}.

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
aotref	Reference aerosols optical depth at 10 µm	r	µm	-		

- nine columns of a variable number of lines:

Table 11: format description of aerosol/ice cloud physical parameters file

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
wl	Wavelengths for which aerosol model properties are provided	r[ksmax]	µm	-		in increasing order
ecoef	Extinction coefficient at wavelengths wl	r[ksmax]	km^{-1}	-		
scoef	Scattering coefficient at wavelengths wl	r[ksmax]	km^{-1}	o	*NOT USED*	
acoef	absorption coefficient	r[ksmax]	km^{-1}		*NOT USED*	
ssa	Single scattering albedo at wavelengths wl	r[ksmax]	-	-		
asp	Asymmetry parameter at wavelengths wl	r[ksmax]	-	-		
eno	normalized extinction coefficient	r[ksmax]	-		*NOT USED*	
rir	refractive index - real part	r[ksmax]	-		*NOT USED*	
rii	refractive index - imaginary part	r[ksmax]	-		*NOT USED*	

- 117 lines which are read but not used
- Part of the phase matrix:
  - the 209th line is read to get the number of coefficient: ncoef

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
ncoef	Number of Greek	int	-	-		

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
	coefficient					

- the 211th line is read to get the the number of moments (nmom) and the number of wavelength (nwn): these 2 variables are the dimension of the Greek matrix;

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
nmom	Number of moments	int	-	-		
nwn	Number of wavelength	int	-	-		

- the 214th line is read to get the numerical wavelength: this is the spectral grid on which the Greek coefficient are defined;

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
wl	Spectral wavelength grid on which the Greek coefficient are defined	r[Cst_ksmax]	µm	-		

- body: the Greek coefficient matrix with the dimension nmom*nwn

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
gcoef	Greek coefficient	r[0:Cst_maxmom,Cst_ksmax,Cst_maxgreek]	-	-	The Greek coefficients are read in an increase wavelength order and then converted in a increase wavenumber order	

3.2.5.2.2 Undef/powonu types

The sub-routine `readgreekcoef` allows reading this kind of files. It is placed in the directory **`srclib`**.

These files are formatted according to the following characteristics:

- Suffix: `.dat;`
- Access: sequential formatted (ASCII).

The format of the Greek coefficient is the same as for the Greek coefficient of the aerosol OPAC model (see the previous section 3.2.5.2.1).

These data files are composed of :

- 15 header lines with :
  - the 3 first are free to the user;
  - the 9th line is read to get the number of coefficient: ncoef
  - the 11th line is read to get the the number of moments (nmom) and the number of wavelength (nwn): these 2 variables are the dimension of the Greek matrix;
  - the 14th line is read to get the numerical wavelength: this is the spectral grid on which the Greek coefficient are defined;

- body: the Greek coefficient matrix with the dimension nmom*nwn

The standard dimension is nmom=120 moments and nwn=61 wavelength.

3.2.5.3. Generating your own Aerosol / ice cloud profile

The user has to specify an aerosol/cirrus profile file that define the type of aerosol/cirrus model for each layer of the profile. The nomenclature of the file is **`aer4a$AER.dsf`** where the globa variable **AER** (see section 3.2.1.1 indicates which aerosol profile file is used for the simulation.

The choice of using a profile file, or not, can be done by means of the parameter **FLAG_PROF** in the input file **`parascat$RSCA.dtp`**, where **RSCA** is a global variable defined in the run shell script. If **FLAG_PROF** is set to zero, then the optical aerosol parameters are defined directly in the file **`parascat$RSCA.dtp`**.

You can add in the **`datscat`** directory your own-defined aerosol physical parameters files, respecting the following ASCII format:
You can copy one of the aerosol physical parameters file in the **`datscat`** directory and change at the 16th line of the header, the value corresponding to the reference aerosol/cloud optical depth at 1000 cm^{-1} and from the 25th line, the nine columns corresponding to:

- the wavelengths for which aerosol model properties are provided;
- the extinction coefficient;
- the scattering coefficient;
- the absorption coefficient;
- the single scattering albedo;
- the asymmetry parameter;
- the normalized extinction coefficient;
- and the real and imaginary part of the refractive index.

Be sure the name of the new file is `aerosols_XXXX.dat`.

3.2.5.4. Mathematical model

The user can also use (or add) a mathematical model in order to characterize the spectral variability of optical parameters of the different types of aerosols/cirrus. At present time, one unique and simple mathematical model is implemented in the 4A/OP-sub code. This model is described in details in a document [DR8]. In such a model, aerosol induce pure scattering effect and the spectral variability of the extinction AOT coefficient is given by a simple power law:

$$\boldsymbol{AOT(\nu) = AOT(\nu_0)\left[\frac{\nu}{\nu_0}\right]^k}$$

Where $\boldsymbol{\nu}$ is the wave number, $\boldsymbol{\nu_0}$ the reference wave number where the AOT is defined and $\boldsymbol{k}$ the degree of the power law. The nomenclature of this model is `XXXX = 'powonu' or 'undef'`. To select this type of aerosol model, the user has to indicate this identifier in the aerosol profile file `aer4a$AER.dsf`.

3.3. Output

3.3.1. Output file description

3.3.1.1. Content

Radiative transfer computation considers each middle point of the layers *pc(nl)* delimited by the atmospheric vertical levels *p(nl)* and *p(nl+1)* specified by the user. Output Jacobians are given for these specific layers.

4A/OP produces only one binary file containing all results. An interface program (document [DR3]) transforms the binary file into ASCII files. Two simulation configurations are possible: "Pseudo-infinite" case and "Instrument" case. The output file names produced are listed in the table (Table 12) below according to the simulation case.

Output data in ASCII files always include radiances, brightness temperature, reflectance (in case of SURF="refl") and wavelengths and can also include other quantities if demanded (Jacobians and/or transmittances) for an instrument simulation case.

The output unit for Jacobians depends on the user choice (see global variable UNIT in Appendix A of [DR1]). The Jacobian unit specified in tables of the following section is the original unit (UNIT=5).

Table 12: 4A/OP output file description

Simulation case	Content	Binary file names In directory outputbin$IDCOMPIL	ASCII file names In directory outputascii$IDCOMPIL
"Pseudo-infinite"	High-resolution radiances	spi(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL).ddb	spi(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL).plt
"Instrument"	Convolved radiances	spc(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE).ddb	spc(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE).plt
"Jacobians"	- Partial derivatives of the radiance (Jacobians) with respect to the temperature	dtj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the molecule mixing ratio	dcj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the surface emissivity	dsj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the surface pressure	dpj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the aerosol optical thickness	daj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the aerosol asymmetric parameter "g aero"	dgj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
	- Jacobians with respect to the aerosol single scattering albedo "ssa aero"	dwj(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	
"Transmittances"	Transmittances for the main view path, and eventually the secondary view path and/or the solar view path	dto(JOB)(ATMPROF)(ATM)(RSTR)$(RSCA)$(RESOL)(INS)(CASE)$(UNIT).plt	

3.3.1.2. Format

Only ASCII file format is described (see also document [DR3] for the description of the source code). The output ASCII files are formatted according to the following characteristics:

Suffix: `.plt`;

Access: sequential formatted (ASCII).

The constant values on the tables below are defined in a constant module `model`. The constants used are:

- Cst_nlevelmax = 101: Maximum number of pressure levels;
- Cst_ncjacmax = 5: Maximum number of molecules allowed in a spectral domain to calculate their Jacobians.

- **<u>High-resolution radiance spectra</u>**

These data files are composed of three columns of a variable number of lines. The file name is `spi$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL).plt`.

Table 13: format description of 4A/OP output ASCII file for the high-resolution radiance spectra

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
1 to (numax-numin)/pdnu+1	ond (fisrt column)	Wave numbers	cm^{-1}	2**19+1	r	e16.8
	spi (second column)	Radiances	W/ (m^2.strd.cm^{-1})	2**19+1	r	e16.8
	temp (third column)	Brightness temperature	K	2**19+1	r	e16.8
	reflect_p	reflectance	-	2**19+1	r	e16.8
	lambda	Wavelength	[µm]	2**19+1	r	e16.8
	spi_mum_p	Spectral radiance	$Wm^{-2}sr^{-1}µm^{-1}$	2**19+1	r	e16.8

- **Convolved radiance spectra**

These data files are composed of three columns of a variable number of lines. The file name is `spc$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE).plt`.

Table 14: format description of 4A/OP output ASCII file for the convolved radiance spectra

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
1 to (numax-numin)/pascon+1	ond (fisrt column)	Wave numbers	cm^{-1}	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers	r	e16.8
	crad (second column)	Radiances	W/ (m^2.strd.cm^{-1})	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers	r	e16.8
	temp (third column)	Brightness temperature	K	Allocated in lirespc4a Size: number of convolution wave numbers	r	e16.8
	reflect	reflectance	-	Allocated in lirespc4a Size: number of convolution wave numbers		e16.8
	lambda	wavelength	microns		r	e16.8

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
	crad_mum	Spectral radiance	$Wm^{-2}sr^{-1}\mu m^{-1}$	Allocated in lirespc4a Size: number of convolution wave numbers	r	e16.8

- **Convolved Jacobians with respect to temperature**

These data files are composed of six columns of a variable number of lines. The file name is `dtj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`.

Table 15: format description of 4A/OP output ASCII file for the convolved jacobians wrt temperature

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1, i = 1 to (numax-numin)/pascon +1	pc(nl), nl = 1 to level2-level1+1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	cnu (fourth column)	Wave numbers	cm^{-1}	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers	r	e16.8
	ctjac (fifth column)	Jacobians in temperature	W/ (m^2.strd. cm^{-1})/K	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers * (Cst_nlevelmax+1)	r	e16.8
	nl (sixth column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					

By using Jacobian unit 7, the temperature Jacobians can now be expressed in $[Wm^{-2}sr^{-1}\mu m^{-1}K^{-1}]$ per atmospheric layer (in $[Wm^{-2}sr^{-1}\mu m^{-1}K^{-1}]$ for the surface temperature Jacobian).

- **Convolved Jacobians with respect to gas mixing ratio**

These data files are composed of seven columns of a variable number of lines. The file name is `dcj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`.

At the surface level (nl=level2), the Jacobians in gas mixing ratio ccjac do not exist. At the surface level, ccjac is set to the Jacobian with respect to the surface emissivity and is written in a new file.

Table 16: format description of 4A/OP output ASCII file for the convolved jacobians wrt gas mixing ratio

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
nc*[(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1], i = 1 to (numax-numin)/pascon +1, nc=1, jselds	pc(nl), nl = 1 to level2-level1+1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	cnu (fourth column)	Wave numbers	cm^{-1}	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers	r	e16.8
	ccjac (fifth column)	Jacobians in gas mixing ratio	$W/(m^2.strd.cm^{-1})/(g/g)$	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers * (Cst_nlevelmax+1)*Cst_ncjacmax	r	e16.8
	nc (sixth column)	Gas index number	-	1	i	I3
	nl (seventh column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					
For nc+1 ≠ nc	Line-jump					

The surface emissivity Jacobians have been removed from these files (removal of the last level in each Jacobian block over atmospheric layers at a given wavenumber and for a given molecule).

The gas mixing ratio Jacobians can now be expressed in two new units:

Using Jacobian unit 6 in [$Wm^{-2}sr^{-1}cm$] per atmospheric layer for a 10% increase of the mixing ratio.

Using Jacobian unit 7 in [$Wm^{-2}sr^{-1}\mu m^{-1}$] per atmospheric layer for a 10% increase of the mixing ratio.

- **<u>Convolved Jacobians with respect to the surface emissivity/reflectance</u>**

These data files are composed of 4 columns of a variable number of lines. The file name is `dsj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`.

At the surface level (nl=level2), the Jacobians in gas mixing ratio ccjac do not exist. At the surface level, ccjac is set to the Jacobian with respect to the surface emissivity.

New formatted user output file. Equivalent information was dispersed so far in the gas mixing ratio Jacobian files.

Table 17: Format description of 4A/OP output ASCII file for the convolved Jacobians wrt surface emissivity.

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
i = 1 to (numax-numin)/pascon +1	pc(level2-level1+1) (first column)	Surface layer pressure	hPa	1	r	e16.8
	ond(i) (second column)	Wave numbers	cm^{-1}	Cst_maxcnv	r	e16.8
	csjac(i) (third column)	Surface emissivity Jacobians	user specified	Cst_maxcnv	r	e16.8
	nl (fourth column)	Surface layer number	-	1	i	i3

The units of the surface emissivity Jacobians (controlled by the Jacobian unit identifier) is unchanged, apart from a new option:

By using Jacobian unit 7, the surface emissivity Jacobians can now be expressed in [$Wm^{-2}sr^{-1}\mu m^{-1}$] for 100% increase of surface emissivity.

Furthermore, in reflectance mode, the Jacobians (in all Jacobian units) are expressed with respect to surface reflectance rather than surface emissivity.

- **Convolved Jacobians with respect to aerosol**

`daj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`

New formatted user output file: the format is equivalent to the format of convolved Jacobians in temperature.

Table 18: Format description of 4A/OP output ASCII file for the convolved Jacobians wrt aerosol

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1, i = 1 to (numax-numin)/pascon +1	pc(nl), nl = 1 to level2-level1+1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	ond(i) (fourth column)	Wave numbers	cm^{-1}	Cst_maxcnv	r	e16.8
	cajac(i,nl) (fifth column)	Jacobians in aerosol	W/ (m^2.strd. cm^{-1})	Cst_maxcnv* (Cst_nlevelmax+1)	r	e16.8
	nl (sixth column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					

The units of the aerosol Jacobians (controlled by the Jacobian unit identifier) are TBD.

- **Convolved Jacobians with respect to the surface pressure**

`dpj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`

Note: there are two kinds of Jacobians with respect to the surface pressure:

- First kind: this jacobian is the sensivity of the radiance for variation of surface pressure only;
- Second kind:this jacobien is the sensivity of the radiance when all the pressure levels expand or contract.

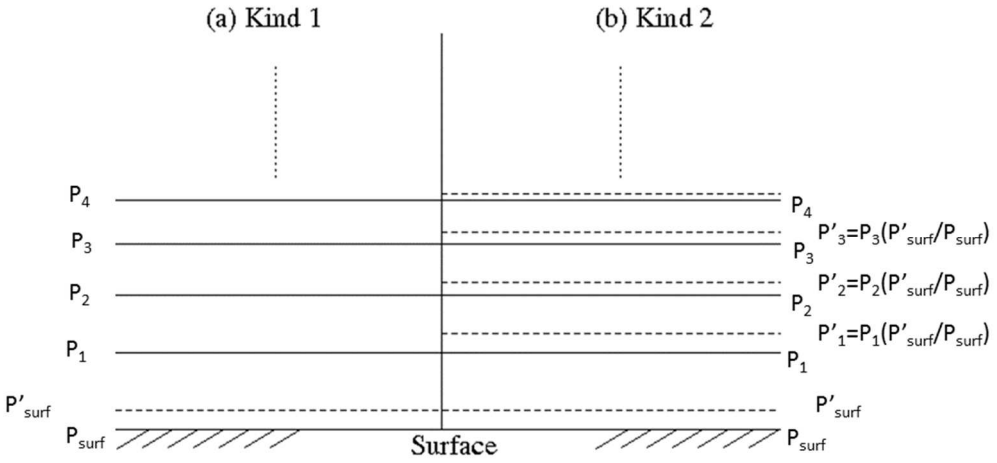

Figure 2: (a) the Jacobian (first kind) is the sensitivity of the radiance for variation of surface pressure; (b) the Jacobian (second kind) is the sensitivity of the radiance when all the pressure levels expand or contract.

New formatted user output file: the format is equivalent to the format of convolved Jacobians wrt temperature.

Table 19: Format description of 4A/OP output ASCII file for the convolved Jacobians wrt surface pressure.

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
i = 1 to (numax-numin)/pascon +1	pc(level2-level1+1) (first column)	mean layer pressure	hPa	1	r	e16.8
	pl(level2-level1+1) (second column)	Pressure of level level2-level1+1	hPa	1	r	e16.8
	pl(level2-level1+2) (third column)	Pressure of level level2-level1+2	hPa	1	r	e16.8
	ond(i) (forth column)	Wave numbers	cm^{-1}	Cst_maxcnv	r	e16.8
	cpjac(i,1) (fifth column)	Kind 1 :Surface Jacobians	W/ (m^2.strd. cm^{-1}.hPa)	Cst_maxcnv	r	e16.8
	cpjac(i,2) (sixth column)	Kind 2: Surface Jacobians	W/ (m^2.strd. cm^{-1}.hPa)	Cst_maxcnv	r	e16.8

- **Convolved Jacobians with respect to the single scattering albedo of aerosols**

`dwj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`

New formatted user output file: the format is equivalent to the format of convolved Jacobians wrt aerosols.

Table 20: Format description of 4A/OP output ASCII file for the convolved Jacobians wrt single scattering albedo of aerosols.

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1, i = 1 to (numax-numin)/pascon +1	pc(nl), nl = 1 to level2-level1+1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	ond(i) (fourth column)	Wave numbers	cm^{-1}	Cst_maxcnv	r	e16.8
	cwjac(i,nl) (fifth column)	Jacobians in single scattering albedo of aerosols	W/ (m^2.strd. cm^{-1})	Cst_maxcnv* (Cst_nlevelmax+1)	r	e16.8
	nl (sixth column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					

- **Convolved Jacobians with respect to the asymmetric parameter of aerosols**

`dgj$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE)$(UNIT).plt`

New formatted user output file: the format is equivalent to the format of convolved Jacobians wrt the single scattering albedo of aerosols.

Table 21: Format description of 4A/OP output ASCII file for the convolved Jacobians wrt asymmetric parameter of aerosols.

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1, i = 1 to (numax-numin)/pascon +1	pc(nl), nl = 1 to level2-level1+1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	ond(i) (fourth column)	Wave numbers	cm^{-1}	Cst_maxcnv	r	e16.8
	cgjac(i,nl) (fifth column)	Jacobians in asymmetric parameter of aerosols	W/ (m^2.strd. cm^{-1})	Cst_maxcnv* (Cst_nlevelmax+1)	r	e16.8
	nl (sixth column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					

- **Convolved transmittances**

These data files are composed of six columns of a variable number of lines. The file name is `dto$(JOB)$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(RESOL)$(INS)$(CASE).plt`.

Table 22: format description of 4A/OP output ASCII file for the convolved transmittances

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
(i-1)*(level2-level1+1)+i to i*(level2-level1+2)-1, i = 1 to (numax-numin)/pascon+1	pc(nl), nl = 0 to level2-level1 (first column)	Pressure of the layer between pl(nl) and pl(nl+1)	hPa	Cst_nlevelmax+1	r	e16.8
	pl(nl) (second column)	Pressure of level nl	hPa	Cst_nlevelmax	r	e16.8
	pl(nl+1) (third column)	Pressure of level nl+1	hPa	Cst_nlevelmax	r	e16.8
	cnu (fourth column)	Wave numbers	cm^{-1}	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers	r	e16.8
	cdto%to1 (fifth column)	Transmittances for the main view path	-	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers * (Cst_nlevelmax+1)	r	e16.8
	cdto%to2 (sixth column)	Transmittances for the secondary view path (when relevant, otherwise empty)	-	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers * (Cst_nlevelmax+1)	r	e16.8
	cdto%tosun (seventh column)	Transmittances for the solar view path (when relevant, otherwise empty)	-	Allocated in readconvdata (module spectrumio) Size: number of convolution wave numbers * (Cst_nlevelmax+1)	r	e16.8
	nl (eighth column)	Level pressure number	-	1	i	i3
i*(level2-level1+2)	Line-jump					

- **Log file printings concerning Doppler shift**

Log file printings concerning Doppler shift have been added. Here below corresponding to the parameterisation example section 3.2.1:

```
  SOLAR COORDINATES and DOPPLER SHIFT
  Latitude                     :  43.6
  Longitude                    :  1.6
  Day (YYMMDD)                 : 110402
  UT [h]                       :  10.0
  Solar zenith angle           :  46.438335
  Solar azimuth angle          :  137.65839
  Earth-Sun Distance [AU]      :  0.99988633
  Relative solar spectral shift :  -1.1608100222259624E-6
```

```
*****************************************************
 * Spectral domain number          :  343
 * First wave number of the domain :  5130  cm-1
 * Number of molecules involved    :  5
 *  Implied molecule identification:  1 2 4 6 11
 *****************************************************
                  ...
 First wavenumber :  5130.0
 Doppler shifted  :  5129.994045044586
```

3.3.2. Examples

The examples provide illustrations of both the input and output files. See the content of each input file for a description of simulation definition parameters (section 3.2.5).

3.3.2.1. Example 1

As a first example, here are the script which performs the required calculations and the corresponding input parameter file (`para4a$RSTR.dtp`) that cause 4A/OP to compute the very high-resolution radiance from 719 to 721 cm^{-1} ("infinite" spectrum):

```
make -f makefile runlirespi4a ATMPROF=0001 ATM=test RSTR=test1 INS=inf
NUMIN=719 NUMAX=721
```

Corresponding input parameter file: `para4atest1.dtp`

```
'AAAA'
' Geometry Identifier (EVIEW/LIMB)                  GEOMID'     'EVIEW'
' Viewing configuration (UP, DOWN, DOWNUP, UPDOWN) TRAJET'       'UP'
' Upper limit of simulation (hPa/km)                 PZUP'        0.05
' Lower limit of simulation (hPa/km)                PZDOWN'     1013.25
' Observation level (hPa if EVIEW/km if LIMB)       PZOBS'        0.05
' Index for the Emissivity of the upper level        EMUP'       'constant' 1
' Temperature of the upper level                      TUP'        4.
' Surface choice Emissivity/Reflectance (emis/refl)  SURF'      'emis'
' Index for the Emissivity of the lower level      EMDOWN'       'constant' 1
' Temperature of the lower level                    TDOWN'        0.0
' Geometric tangent height (km) (LIMB only)         ZGTAN'        0.
' Viewing "zenith" angle (deg)                      TETA1'        0.
' Reflexion/emission angle (deg)                    TETA2'        55.
' Viewing azimuth angle (deg) (scattering case only) PHIV'        0.
' Sun angle geometry definition (OWN/GEO)     TYP_ANG_SUN'     'OWN'
' Solar zenith angle (deg) (OWN only)              TETAS'       90.
' Solar azimuth angle (deg) (OWN only)              PHIS'        0.
' Observation date (yymmdd)                         DATE'   '110103'
```

```
' Observation hour (decimal hour UT: 0:24) (GEO only)  TU'      12.0
' Latitude (deg) (-90:90) (GEO only)                   LAT'      43.6
' Longitude (deg) (-180:180) (GEO only)                LON'       1.6
' Atmospheric Refraction                             REFRA'      'NO'
' Altitude of lowest atm. pressure level (km)     ZATMBOT'        0.
' Continua (H2O, N2, O2) contribution               CONTIN'     'YES'
' Spectral resolution                               RESPEC'        0.00
' First wave number of the simulation                FORIG'      719.
' Last wave number of the simulation                 FSTOP'      721.
' Width of each Atlas spectral band                  DELDS'       15.
' Optical thickness interpolation in temperature   INTOPT'     'YES'
' Indicator of hole algorithm for the convolution    HOLE'      'NO'
' Spectral shift introduction due to the ISRF       SHIFT'      'NO'
' Value of the global spectral shift                   RNU'        0.
' Transmittance calculation flag                     FPOID'      'NO'
' Jacobian calculation flag                          JACOB'      'NO'
' Selection indexes for the 40 gases of the GEISA database (1 or 0)   IDSEL'
'   h2o   co2    o3   n2o    co   ch4    o2    no   so2   no2'
    1    , 1   , 1   , 1   , 1   , 1   , 0   , 0   , 0   , 0   ,
'   nh3   ph3  hno3    oh    hf   hcl   hbr    hi   clo   ocs'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  h2co  c2h6  ch3d  c2h2  c2h4  geh4   hcn  c3h8  c2n2  c4h2'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hc3n  hocl    n2 ch3cl  h2o2   h2s hcooh  cof2   sf6  c3h4'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  ho2  clono2  xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hdo    xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  cfc11 cfc12 ccl4                                          '
    1    , 1   , 1   ,
' Weighting coefficients of the mixing ratio         ROCOEF'
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
' Selection indexes for the gas Jacobian calculation (1 or 0) IDJAC'
'   h2o   co2    o3   n2o    co   ch4    o2    no   so2   no2'
    1    , 0   , 1   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'   nh3   ph3  hno3    oh    hf   hcl   hbr    hi   clo   ocs'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  h2co  c2h6  ch3d  c2h2  c2h4  geh4   hcn  c3h8  c2n2  c4h2'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hc3n  hocl    n2 ch3cl  h2o2   h2s hcooh  cof2   sf6  c3h4'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  ho2  clono2  xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hdo    xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  cfc11 cfc12 ccl4                                          '
    0    , 0   , 0   ,
```

The spi4asub0001testtest1hr15.plt file contains the 4A/OP output corresponding to this input. Columns 1 through 3 are: the wave numbers (*cm*$^{-1}$), the radiance at the top of the atmosphere (*W/(m*^{2} *str cm*$^{-1}$)) and the corresponding brightness temperature (*K*). The results for this first example are shown in Figure 3. A good way to verify that

4A/OP is operating correctly on your system is to use your favourite graphics software to read spi4asub0001testtest1hr15.plt and compare the results visually.

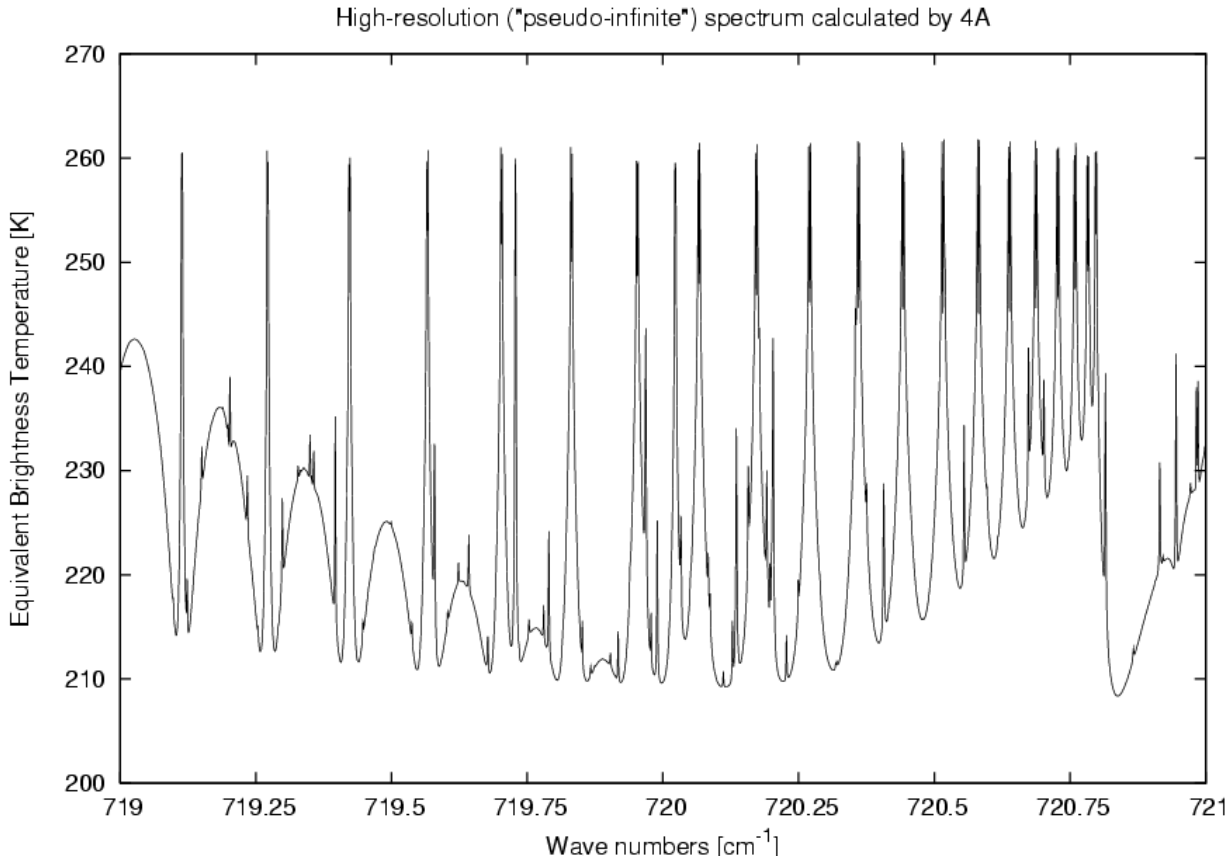

Figure 3: High-resolution radiance computed with 4A from 719 to 721 cm^{-1} in equivalent brightness temperature (test1).

3.3.2.2. Example 2

The second example is the computation of the convolved radiance that would be measured by the instrument IASI, from 645 to 2760 cm^{-1}.

```
make -f makefile runlirespc4a NUMIN=645 NUMAX=2760 RSTR=test8 INS=iasi1c
ATM=test ATMPROF=0001
```

Corresponding input parameter file: para4atest8.dtp

```
'AAAA'
' Geometry Identifier (EVIEW/LIMB)                  GEOMID'     'EVIEW'
' Viewing configuration (UP, DOWN, DOWNUP, UPDOWN) TRAJET'       'UP'
' Upper limit of simulation (hPa/km)                 PZUP'        0.05
' Lower limit of simulation (hPa/km)               PZDOWN'     1013.25
' Observation level (hPa if EVIEW/km if LIMB)       PZOBS'        0.05
' Index for the Emissivity of the upper level        EMUP'       'constant' 1
' Temperature of the upper level                      TUP'        4.
' Surface choice Emissivity/Reflectance (emis/refl)  SURF'      'emis'
' Index for the Emissivity of the lower level      EMDOWN'       'constant' 1
' Temperature of the lower level                    TDOWN'        0.0
' Geometric tangent height (km) (LIMB only)         ZGTAN'        0.
' Viewing "zenith" angle (deg)                      TETA1'        0.
' Reflexion/emission angle (deg)                    TETA2'       55.
' Viewing azimuth angle (deg) (scattering case only) PHIV'        0.
' Sun angle geometry definition (OWN/GEO)     TYP_ANG_SUN'     'OWN'
```

```
' Solar zenith angle (deg) (OWN only)              TETAS'       90.
' Solar azimuth angle (deg) (OWN only)              PHIS'        0.
' Observation date (yymmdd)                         DATE'   '110103'
' Observation hour (decimal hour UT: 0:24) (GEO only)  TU'      12.0
' Latitude (deg) (-90:90) (GEO only)                  LAT'      43.6
' Longitude (deg) (-180:180) (GEO only)               LON'       1.6
' Atmospheric Refraction                            REFRA'      'NO'
' Altitude of lowest atm. pressure level (km)     ZATMBOT'        0.
' Continua (H2O, N2, O2) contribution              CONTIN'     'YES'
' Spectral resolution                              RESPEC'        0.00
' First wave number of the simulation               FORIG'      645.
' Last wave number of the simulation                FSTOP'     2760.
' Width of each Atlas spectral band                 DELDS'       15.
' Optical thickness interpolation in temperature   INTOPT'     'YES'
' Indicator of hole algorithm for the convolution    HOLE'      'NO'
' Spectral shift introduction due to the ISRF       SHIFT'      'NO'
' Value of the global spectral shift                  RNU'        0.
' Transmittance calculation flag                    FPOID'      'NO'
' Jacobian calculation flag                         JACOB'      'NO'
' Selection indexes for the 40 gases of the GEISA database (1 or 0)   IDSEL'
'   h2o   co2    o3   n2o    co   ch4    o2    no   so2   no2'
    1    , 1   , 1   , 1   , 1   , 1   , 1   , 0   , 0   , 0   ,
'   nh3   ph3  hno3    oh    hf   hcl   hbr    hi   clo   ocs'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  h2co  c2h6  ch3d  c2h2  c2h4  geh4   hcn  c3h8  c2n2  c4h2'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hc3n  hocl    n2 ch3cl  h2o2   h2s hcooh  cof2   sf6  c3h4'
    0    , 0   , 1   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  ho2  clono2  xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hdo    xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  cfc11 cfc12 ccl4                                          '
    1    , 1   , 1   ,
' Weighting coefficients of the mixing ratio         ROCOEF'
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
    1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 ,
' Selection indexes for the gas Jacobian calculation (1 or 0) IDJAC'
'   h2o   co2    o3   n2o    co   ch4    o2    no   so2   no2'
    1    , 0   , 1   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'   nh3   ph3  hno3    oh    hf   hcl   hbr    hi   clo   ocs'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  h2co  c2h6  ch3d  c2h2  c2h4  geh4   hcn  c3h8  c2n2  c4h2'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hc3n  hocl    n2 ch3cl  h2o2   h2s hcooh  cof2   sf6  c3h4'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  ho2  clono2  xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  hdo    xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx   xxx'
    0    , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   , 0   ,
'  cfc11 cfc12 ccl4                                          '
    0    , 0   , 0   ,
```

You can compare your results written in the file `spc4asub0001testtest8hr15iasi1c.plt` to those shown in Figure 4.

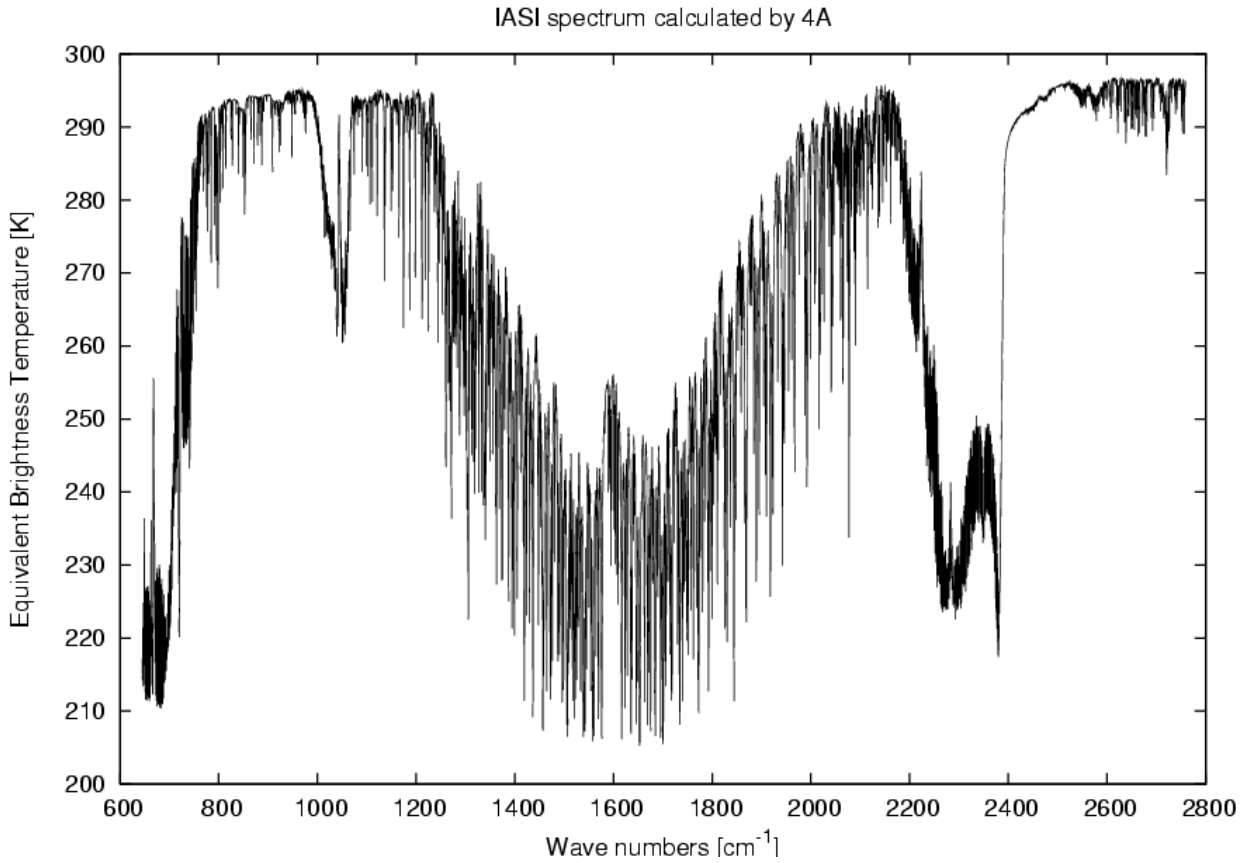

Figure 4: Convolved radiance (in in equivalent brightness temperature) that would be measured by the instrument IASI, from 645 to 2760 cm^{-1} (test8).

3.3.3. Result visualization and jacobian unit recommendation

Currently ASCII result files are produced and their visualization can be performed with the free Unix tool Gnuplot, or with IDL, or with some other plotting program.

The output unit for Jacobians depends on the user choice (see global variable UNIT in the Appendix A of [DR1] and the subroutine *converjacob* in document [DR3]). The output unit in binary file is the computation unit ($W/(m^2.strd.cm^{-1})/K/layer$ or $W/(m^2.strd.cm^{-1})/(g/g)/layer$): UNIT=5); for ASCII extraction, it can be converted into another unit available in the subroutine *converjacob*. Among the six possible units, we recommend the user to visualize Jacobians with normalized values, that is for UNIT=1: NedT at a reference temperature (usually 280K) in order to be independent of the temperature profile and per kilometre in order to be independent of the radiative transfer discretization. The corresponding unit for each Jacobian type is the following (detailed in the table following):

- Temperature: Kelvin (Nedt at tref) per Kelvin.Km;
- Gas mixing ratio: Kelvin (Nedt at tref) per Km for 10% variation of the mixing ratio;
- Emissivity: Kelvin (Nedt at tref) for 1% variation.

However, the user is free to visualize Jacobians in any other unit by converting it with his own tool.

Jacobian units in ascii outputfiles from lirespc as function of UNIT user parameter setting as of 2/11/2012.

Jacobian UNIT	Surface temperature	Surface emissivity/ reflectance	Temperature profile	Mass mixing ratio profile	AOT profile	Surface pressure
0	NEdT@Tref [K/K]	NEdT@Tref [K/1%]	NEdT@Tref [K/K/hPa]	NEdT@Tref [K/(g/g)/hPa]	[W/(m²srμm) /1]	NEdT@Tref [K/hPa]
1			NEdT@Tref [K/K/km]	NEdT@Tref [K/10%/km]		

2	[dB/K]	[dB/1%]	[dB/K/km]	[dB/10%/km]		[dB/hPa]
3	NEdT@Tbright [K/K]	NEdT@Tbright [K/1%]	NEdT@Tbright [K/K/km]	NEdT@Tbright [K/10%/km]		NEdT@Tbright [K/hPa]
4			NEdT@Tbright [K/K/hPa]	NEdT@Tbright [K/(g/g)/hPa]		
5	[W/(m^{2}srcm^{-1})/K]	[W/(m^{2}srcm^{-1}) /100%]	[W/(m^{2}srcm^{-1})/ K/hPa]	[W/(m^{2}srcm^{-1})/ (g/g)/hPa]		[W/(m^{2}srcm^{-1}) /hPa]
6				[W/(m^{2}srcm^{-1})/ 10%/hPa]		
7	[W/(m^{2}srμm)/ K]	[W/(m^{2}srμm)/ 100%]	[W/(m^{2}srμm)/ K/hPa]	[W/(m^{2}srμm)/ 10%/hPa]		[W/(m^{2}srμm)/ hPa]

3.3.4. Statistics tool

A statistics tool is available for the computation of the difference between two spectra and the corresponding statistics (see document [DR3] for the source code description). It does work for convolved spectra only and both spectra must be sampled with the same sampling step.

As an example, here is a script which performs the required calculations that cause the statistics program to compute the difference between a spectrum to be compared stored in the file `spc4a$(SPC1).ddb` and a reference spectrum stored in the file `spc4a$(SPC2).ddb`:

```
make -f makefile runstatanalys TREF=280 WCLASS=50 WNMIN=645 WNMAX=2760
SPC1=0002test8iasi1c SPC2=0001test8iasi1c
```

See the Appendix A of [DR1] for the description of the required input variables.

Two ASCII files are generated and stored in the directory `outputascii`:

- `stat_$(SPC1)_$(SPC2).plt`: difference in radiance and in NedT;
- `classstat_$(SPC1)_$(SPC2).plt`: statistical characteristics of the difference per wave number class with the width WCLASS.

The output format of these files is described below.

3.3.4.1. File of the spectra difference

This file is composed of five columns of a variable number of lines (nbwnScomspect: number of spectra samples). The file name is `stat_$(SPC1)_$(SPC2).plt`.

Table 23: format description of the first output ASCII file for the spectra difference

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
1 to nbwnScomspect	wnScomspect (fisrt column)	Wave numbers	cm^{-1}	Allocated in statanalys Size: number of convolution wave numbers	d	e21.12
	RdScomspect (second column)	Spectrum (radiance) to be compared	W/ (m^2.strd.cm^{-1})	Allocated in statanalys Size: number of convolution wave numbers	r	e21.12
	RdSrefspect (third column)	Reference spectrum	W/ (m^2.strd.cm^{-1})	Allocated in statanalys Size: number of convolution wave numbers	r	e21.12

	diff (fourth column)	Normalised difference between both spectra in NedT at TREF	K	Allocated in statanalys Size: number of convolution wave numbers	d	e21.12
	RdScomspect – RdSrefspect (fifth column)	Difference between both spectra	W/ (m^2.strd.cm^{-1})	Allocated in statanalys Size: number of convolution wave numbers	r	e21.12

3.3.4.2. File of the statistical characteristics per class

This file is composed of six columns of a variable number of lines (nbClass: number of classes calculated according to WCLASS). The file name is `classstat_$(SPC1)_$(SPC2).plt`.

See appendix document [DR2] for description of percentiles.

Table 24: format description of the second output ASCII file for the statistical characteristics per class of the spectra difference

Number of the line	Parameter name	Description	Units	Dimension (maximum)	Type	Format
1 to nbClass	wnSclass (fisrt column)	Wave numbers corresponding to the classes	cm^{-1}	Allocated in statcharclass Size: number of convolution wave numbers	d	e21.12
	rmsClass(1) (second column)	Root mean square of the difference in NedT per class	K	Allocated in statcharclass Size: number of convolution wave numbers	d	e21.12
	rmsClass(2) (third column)	Standard deviation of the difference in NedT per class	K	Allocated in statcharclass Size: number of convolution wave numbers	d	e21.12
	percentil(1) (fourth column)	Percentiles at 90 %	-	Allocated in statcharclass Size: number of convolution wave numbers	r	e21.12
	percentil(2) (fifth column)	Percentiles at 95 %	-	Allocated in statcharclass Size: number of convolution wave numbers	r	e21.12
	percentil(3) (sixth column)	Percentiles at 100 %	-	Allocated in statcharclass Size: number of convolution wave numbers	r	e21.12

3.4. Use 4A/OP subroutine mode

3.4.1. Input sub-structures description

This is the description of the sub-structures of the 4A/OP-sub input parameter structure `Params`.

```
! atmref
type :: PAtmRef
   :: psolref, tsolref, secref
   :: plref, pcref
   :: tlref, tcref
   :: rolref, rocref
end type PAtmRef

! gas
type :: PDefaultGas
   :: nlevgas
   :: plgas
   :: rolgas
end type PDefaultGas

! profile
type :: PProfile
   real(kind=SIMPLE)                                          :: pdown
   real(kind=SIMPLE), dimension(Cst_nlevelmax)                :: pl, tl, dp
   real(kind=SIMPLE), dimension(Cst_nlevelmax,Cst_ncorps)     :: rol
   integer(kind=LONG), dimension(Cst_ncorps)                  :: idcor
   real(kind=SIMPLE)                                          :: tdown
   integer(kind=LONG)                                         :: nlevatm
   integer(kind=LONG)                                         :: nopro
   integer(kind=LONG)                                         :: lat
   integer(kind=LONG)                                         :: lon
   integer(kind=LONG)                                         :: date
   character(len=3)                                           :: flag_pc
end type PProfile

! simdef
type :: PSimDef
   :: pzup, pzdown, pzobs
   :: tdown, tup
   :: teta1, teta2
   :: phi_v, phi_s, theta_s
   :: t_u,
   :: latdeg, londeg
   :: zgtan, zatmbot
   :: respec
   :: forig, fstop
   :: rnu
   :: pascon
   :: cemdown, cemup
   :: geomid, refra, yymmdd
   :: trajet, contin, jacob, conv, hole
   :: shift, fpoid, intopt
   :: Surf
   :: typ_ang_sun
   :: delds                                  ! Width of each Atlas spectral band
end type PSimDef

! scatsim
type :: PScatSim
   :: nlaero, nmom, nstream
   :: iflag_ray
   :: iflag_prof
   :: iflag_pol
   :: noproaer
```

```
    :: aotrwl
    :: aotref, omeaero, gaero
    :: playaero
    :: accur
    :: prnt(5)
    :: flag_scat
    :: aeromod
    :: disheader
    :: racine4, racine5
    :: faero
    :: layaero
    :: nwn                          !
    :: wn                           !
    :: aot                          !
    :: omega                        !
    :: g                            !
    :: greek_nwn                    !
    :: greek_wn                     !
    :: greek_coef                   !
    :: kaero                        ! AOT(nu) = AOT(nu0)*(nu0/nu)^(k)
end type PScatSim

! mol
type :: PMolecule
    :: idsel, idjac
    :: rocoef
end type PMolecule

! index
type :: PAtlasIndex
  integer(kind=LONG), dimension(Cst_dsindexmax)            :: ifreq1
  integer(kind=LONG), dimension(Cst_dsindexmax)            :: ncor
  integer(kind=LONG), dimension(Cst_ncmax,Cst_dsindexmax)  :: numcor
end type PAtlasIndex

! emis
type :: PEmissivities
    :: nonde
    :: tabemup, tabemdown
end type PEmissivities

! isrf
type :: PIsrf
    :: nbk,itypeconv
    :: irdeb
    :: irecf, kfmo
    :: nknu
    :: nufdeb, nuffin, dnds
    :: nufo, wnuo, dnuo
    :: pdnuco
end type PIsrf

! fnames
type :: PFilenames
   character(len=500)  :: fisrf
   character(len=500)  :: fresui, fresuc
   character(len=500)  :: atlasFilesRoot
   character(len=500)  :: solarFileRoot
   character(len=500)  :: solarFileSuffix
   character(len=500)  :: fbrdf
end type PFilenames

! continua
type :: Pcontinua
      character(len=500)  :: fcontH2O, fcontN2, fcontO2, fcontCO2
      integer(kind=LONG)  :: nb_contH2O, nb_contN2, nb_contO2, nb_contCO2
```

```
      real(kind=DOUBLE) , dimension(:) , allocatable          :: freqch2o
      real(kind=DOUBLE) , dimension(:) , allocatable          :: ratih2o
      real(kind=DOUBLE) , dimension(:) , allocatable          :: coselfh2o
      real(kind=DOUBLE) , dimension(:) , allocatable          :: frach2o

      real(kind=DOUBLE) , dimension(:) , allocatable          :: SigRefN2
      real(kind=DOUBLE) , dimension(:) , allocatable          :: B0N2
      real(kind=DOUBLE) , dimension(:) , allocatable          :: Beta0N2
      real(kind=DOUBLE) , dimension(:) , allocatable          :: B0N2_H2O
      real(kind=DOUBLE) , dimension(:) , allocatable          :: Beta0N2_H2O
      real(kind=DOUBLE) , dimension(:) , allocatable          :: SigRefO2
      real(kind=DOUBLE) , dimension(:) , allocatable          :: B0O2
      real(kind=DOUBLE) , dimension(:) , allocatable          :: Beta0O2
      real(kind=DOUBLE) , dimension(:) , allocatable         :: SigRefCO2
      real(kind=DOUBLE) , dimension(:) , allocatable         :: B0CO2_296
      real(kind=DOUBLE) , dimension(:) , allocatable         :: B0CO2_218
      real(kind=DOUBLE) , dimension(:) , allocatable         :: B0CO2_H2O_270
      real(kind=DOUBLE) , dimension(:) , allocatable         :: B0CO2_H2O_310

      real(kind=DOUBLE) , dimension(:,:) , allocatable         :: contH2O
      real(kind=DOUBLE) , dimension(:,:) , allocatable         :: dtcontH2O
      real(kind=DOUBLE) , dimension(:,:) , allocatable         :: dccontH2O

      real(kind=DOUBLE) , dimension(:,:) , allocatable         :: contO2
      real(kind=DOUBLE) , dimension(:,:) , allocatable         :: contN2

  end type Pcontinua

type :: PExponential
   :: filename
   :: Interp_Mode
   :: N_Exp
   :: Exp_Step
   :: Exp
   :: Fct
   :: Fct_Slope_A1
   :: Fct_Spline_A1
   :: Fct_Spline_A2
   :: Fct_Spline_A3
end type PExponential
```

3.4.2. Modifying the 4A/OP-sub structure fields

Here is an example of a fortran code where the 4A/OP-subroutine input parameter structure `Params` is modified before and after the calling of the module `computeaaaa`.

The subroutine `modify_Params` is used to modify some fields of the `Params` structure with respect to field values of the `Results` structure. The outputs are then put into the `Params_update` variable that is the same structure as `Params` but with field values modified. The other subroutine, `modify_Params`, allows to update the `Params` structure with the field values of the structure `Params_update`.

```
write(jlist,*)
write(jlist,*) '----------------------------------------'
write(jlist,*) 'Begin of the radiative transfer computations (iterative mode)'
write(jlist,*) '----------------------------------------'
write(jlist,*)
write(jlist,*)
```

```
! Initilization
call readaaaa(jinput, jlist, Params, ErrCode)
call checkaaaa(jlist, Params, ErrCode)

! Compute 4A radiative tranfer
do i = 1, nb_iteration

     call computeaaaa(jlist, Params, Results, Results_p, ErrCode)
     call modify_Params(Params, Results, Params_update)
     call update_Params(Params_update, Params)

enddo

! Write results
call writeaaaa(jlist, Params, Results, Results_p, ErrCode)


write(jlist,*)
write(jlist,*)
write(jlist,*) '----------------------------------------'
write(jlist,*) 'End of the radiative transfer computations'
write(jlist,*) '----------------------------------------'
write(jlist,*)
```

```
! modify_params (example of modifying the Profile%pdown field)
test_radiance = Results%crad(1) - measured_radiance(1)
if ( test_radiance <= 0 ) then
     Params_update%Profile%pdown = Params%Profile%pdown + 1.0
else
     Params_update%Profile%pdown = Params%Profile%pdown
endif
```

```
! update_params (example of updating the Profile%pdown field)
Params%Profile%pdown = Params_update%Profile%pdown
```

4. Software maintenance

4.1. Version control

A revision control has been set up by the way of the control management tool SVN (Subversion) in order to administer modifications of source files. Non-regression tests are also available and are presented in appendix (Appendix 2 document [DR1])

A nomenclature is proposed to name the different versions. The name of the current software is 4A2000. It is considered that is the version 0.0: the current software version is named 4A2000 v0.0 (written later: *vmodel v I.r*). Conditions for incrementing *vmodel*, *I* (Issue) and *r* (review) are suggested below:

- *r*: minor changes (additional comments for example, bug fix): *4A2000 v0.1 (for example);*
- *I*: changes in the implementation with no major modification (error management addition for example, translation in Fortran90): *4A2000 v1.0;*
- *vmodel*: major modifications (improvement of the physics): *4A2002 v0.0.*

The graphical user interface and the web site must be follow this nomenclature such as:

- GUI: GUI *vmodel-I* vG.g for the 4A version *vmodel vI.r* and the GUI version vG.g;
- WEB: WEB *vmodel-I* vW.w for the 4A version *vmodel vI.r* and the WEB version vW.w.

The graphical interface and the web site must have their own version (vG.g and vW.w respectively) as well as the 4A version. They will be attached to a version of 4A (*vmodel-I*).

4.2. Error control

A component to manage errors has been set up: the module `error.f90` [DR3].It notifies an error with a message and the name of the corresponding sub-routine. Document [DR3] `section "error management"` presents the glossary builder that produces a list of controlled errors that can occur during a 4A run.

5. REFERENCES

Numéro	Titres
[DR1]	SETG-MU-4AOP-010-TS_Installation and Quickstart Guide.
[DR2]	SETG-MU-4AOP-013-TS_Scientific manual.
[DR3]	NOV-FE-0035-NT-031_Algorithm manual.
[DR4]	Snyder, W.C., et al.,1998: Classification-based emissivity for land surface temperature measurement from space. Int. J. Remote Sensing, vol. 19, 2753-2774.
[DR5]	The software package OPAC (Optical Properties of Aerosols and Clouds) http://www.lrz-muenchen.de/~uh234an/www/radaer/opac.html
[DR6]	Guignard, A., Stubenrauch, C. J., Baran, A. J., and Armante, R.: Bulk microphysical properties of semi transparent cirrus from AIRS: a six years global climatology and statistical analysis in synergy with CALIPSO and CloudSat, Atmos. Chem. Phys. Discuss., 11, 24671-24725, doi:10.5194/acpd-11-24671-2011, 2011.
[DR7]	Mitchell, D. L., A. Macke, and Y. Liu, 1996: Modelling Cirrus Clouds: II) Treatment of radiative properties. J. Atmos. Sci., 53, 2967-2988.
[DR8]	O. Lezeaux, 2013: Validation du code d'inversion de MicroCarb et étude pour une prise en compte efficace des aérosols, NOV-7054-NT-2714-v2.0.
[DR9]	Seemann, S.W., E. E. Borbas, R. O. Knuteson, G. R. Stephenson, H.-L. Huang, 2008: Development of a Global Infrared Land Surface Emissivity Database for Application to Clear Sky Sounding Retrievals from Multi-spectral Satellite Radiance Measurements.J. Appl. Meteor. Climatol., Vol. 47, 108-123.

6. ACRONYMES

- **4A**: Automatized Atmospheric Absorptions Atlas;
- **4A/OP-sub**: subroutine version of 4A;
- **CNES**: Centre National d'Etudes Spatiales (http://www.cnes.fr/);
- **CNRS**: Centre National de la Recherche Scientifique (http://www.cnrs.fr/);
- **GEISA**: Gestion et Etude des Informations Spectroscopiques Atmosphériques (Management and Study of Atmospheric Spectroscopic Information);
- **ISRF**: Instrument Spectral Response Function;
- **IASI**: Infrared Atmospheric Sounding Interferometer;
- **IGBP**: International Geosphere-Biosphere Program;
- **LMD**: Laboratoire de Météorologie Dynamique (http://lmd.polytechnique.fr/);
- **MODIS**: Moderate Resolution Imaging Spectroradiometer;
- **NedT**: Noise equivalent temperature difference;
- **RAM**: Random Access Memory;
- **RTM**: Radiative Transfer Model;
- **TIGR**: Thermodynamic Initial Guess Retrieval: Climatological library of about 1800 representative atmospheric situations from radiosonde reports.