Quick Start Guide

Software Version 4AOP2012 v1.0

March 2012

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

Table of contents

1. Introduction

1.1. Purpose

The software 4A/OP is provided for an enhancement by NOVELTIS, in accordance with the convention signed between CNES, LMD/CNRS and NOVELTIS.

The objective of the software is to allow the fast simulation of the radiative transfer in particular over the infrared range either with a "pseudo-infinite" (high) resolution or with a spectral resolution of the simulated instruments (low resolution).

The 4A model is a line-by-line model. It allows a rapid computation of the radiative transfer without accuracy loss thanks to a prior creation of an optical thickness database. This database named Atlas contains the monochromatic optical thicknesses for the various atmospheric constituents (Scott and Chédin, 1981: [Ref. 1]). The atlases are created once and for all by using the line-by-line and layer-by-layer model, STRANSAC (Scott, 1974: [Ref. 3]; [Ref. 2]) in its latest 2000 version with up to date spectroscopy from the GEISA spectral line catalogue (Jacquinet-Husson et al., 1999, 2003, 2008 and 2011: [Ref. 4], [Ref. 5], [Ref. 6], [Ref. 7]). This concept has been developed and maintained at the Laboratoire de Météorologie Dynamique (LMD) under the abbreviation 4A for Automatized Atmospheric Absorption Atlas.

4A output is the radiance spectrum in a user-defined spectral domain in the infrared region; the usual spectral domain is between 600 and 3000 cm^{-1}. 4A can be used for a wide variety of surface and earth atmospheric conditions; its use could be extended to extra-terrestrial atmospheric conditions. Spectra are computed at high spectral resolution (the nominal spectral resolution is 5.10^{-4} cm^{-1}, but it can be changed by users). They can also be convolved with various types of instrument function. Partial derivatives of the radiance with respect to the temperature and gas mixing ratio can be also computed. They allow the model coupling with an inversion algorithm for the atmospheric constituent retrieval from infra-red radiance measurements.

4A heritage is from 4A2000 with its initial development at LMD supported under CNRS/ENS and CNES programs. CNES and CNRS/ENS support the continued advancements of 4A through a subcontract with NOVELTIS. NOVELTIS is in charge of the industrialization and distribution of the 4A software. The aim is to complete an easy-to-use product for several research group users that is also easy to integrate in operational processing chains. 4A/OP software is a version of 4A for distribution to registered users.

Numerous works have been achieved in order to improve the readability of the software code and to favour its portability and its maintenance. The available operational version includes:

- Regular updating and improvements;
- Graphical User Interface (GUI);
- Reference documentation;
- Website http://www.noveltis.fr/4AOP/ including an on-line registration form;
- Distribution with maintenance and assistance;
- Additional scientific functions:
  - User-defined spectral emissivity functions;
  - Spherical atmosphere;
  - Solar contribution;
  - Limb viewing including refraction;
  - Scattering for aerosol and cirrus contribution.

This Quick Start Guide for 4A/OP is intended for beginners with the 4A/OP software: it provides a quick way inside the most common 4A/OP features. For larger understanding of 4A/OP, one should refer to the Reference Documentation ([Ref. 10]) for 4A/OP where one can find the comprehensive description of the internal data, parameterisations, procedures and the physics used.

1.2. How to use this document

The 4A/OP Quick Start Guide is intended to help new user to begin using 4A/OP. It is divided into five parts: the installation guide, the preparation of the simulation, the first use of 4A/OP in script mode and the use of the graphical user interface (GUI; that has been developed from the existing code in order to make the software use easier), the visualization of the simulation results and the way to modify the user input files. These different parts help the new user through several examples to explore the main 4A/OP capabilities.

1.3. Definitions, acronyms and abbreviations

4A	Automatized Atmospheric Absorptions Atlas
CNES	Centre National d'Etudes Spatiales
CNRS	Centre National de la Recherche Scientifique
GEISA	Gestion et Etude des Informations Spectroscopiques Atmosphériques: Management and Study of Atmospheric Spectroscopic Information [Ref. 4], [Ref. 5]
IASI	Infrared Atmospheric Sounding Interferometer
IGBP	International Geosphere-Biosphere Program
LMD	Laboratoire de Météorologie Dynamique
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 2000 representative atmospheric situations from radiosonde reports [Ref. 8]

1.4. People and acknowledgments

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

Authors from LMD: Raymond Armante, Noëlle A. Scott

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

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

1.5. Referencing 4A/OP in publications

If a user is presenting results obtained by 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;
- L. Chaumat, C. Standfuss, B. Tournier, E. Bernard, R. Armante and N. A. Scott, 2012: 4A/OP Reference Documentation, NOV-3049-NT-1178-v4.3, NOVELTIS, LMD/CNRS, CNES, 310 pp.

2. Installing 4A/OP

The installation phase includes the installation of the algorithm folder tree with available files and the compilation of the full source code.

2.1. What is required ?

The 4A/OP distribution package is supplied in a single tar file `4AOP-version.tar.bz2`, where version corresponds to the current 4A/OP Software version number (presently `4AOP-2012-1-0`). The file is compressed using `bzip2`, available at no cost from the Free Software Foundation.

The following files are sufficient for the installation process:

- README (file containing important information);
- 4A`OP-version.tar.bz2` (4A/OP as compressed (zip) tar file).

Installation of 4A/OP on a UNIX system is performed using UNIX b`unzip2` and `tar` commands and the GNU Make program.

`4AOP-version.tar.bz2` is a compressed tar file containing the entire 4A/OP program (possible exception: large data base files which might be stored separately). This comprises several subdirectories with FORTRAN 90 source files (`*.f90`), a GNU-`Makefile`, UNIX shell scripts, user input files (`*.dtp`) and several other files including all necessary input and test data for 4A/OP.

2.1.1. Compilers

The code has been developed in Fortran 77 on SUN workstations and has been ported in Fortran 90 on SUN Ultra Sparc workstations under Solaris using the Forte 6.2 Fortran 90 compiler. 4A/OP has also been tested on PC Linux using the pgf90 compiler (Portland Group Fortran 90 compiler), g95/gfortran (GNU compiler), ifort (Intel compiler) and f90 (SUN compiler). The 4A/OP source code is FORTRAN 90 compatible and contained within several files `*.f90`.

Note: The different Fortran 90 compilers are proposed in the following order: f90 (SUN) (Linux or SunOS), ifort (INTEL) (Linux only), pgf90 (PGI) (Linux only), gfortran (GNU) (Linux only) and g95 (GNU) (Linux or SunOS). Fortran compiler is limited to 32-bit mode if it is not default mode (except in ifort case, UNKNOWN 32-bit limitation option). The user can define his/her choice by >> export PATH_F90=... (compiler with full path).

Moreover, a new routine written in C has been introduced and needs that a C compiler is installed on your system. 4A/OP has been tested on PC Linux using the gcc compiler (GNU compiler) and cc (SUN compiler).

Note: The different C compilers are proposed in the following order: cc (SUN) (Linux or SunOS) and gcc (GNU) (Linux or SunOS). C compiler is limited to 32-bit mode if it is not default mode. The user can define his/her choice by >> export PATH_CC=... (compiler with full path).

Important note: 4A/OP has been developed and tested on a big-endian architecture (Sun). In particular the optical thickness atlases are Sun binary files. On little-endian architecture (i.e. x86), be sure your compiler has the option that swaps bytes at file reading.

2.1.2. Tcl/Tk

The 4A/OP GUI needs that Tcl/Tk (at least version 8.4) is installed on your system. See www.tcl.tk for more information on Tcl/Tk.

2.1.3. Gnuplot

The visualization part of the 4A/OP GUI needs Gnuplot software to plot graphs. The GUI has been tested with the following version of Gnuplot: Unix version 4.0 patchlevel 0. See www.gnuplot.info for more information on Gnuplot.

2.1.4. Resources

The 4A executable occupies about one megabyte on disk only. It uses about 200 megabytes of memory. The associated files (sources and user input files) occupy about 20 megabytes of disk space. There are also the atlas files that can require up to 6.5 gigabytes of disk space. This volume depends on the user simulation needs.

The size of the output files is a function of the input simulation parameters, which you control via the file described in section Part I Section 5.2.5.2 "Run characterization parameters" of the Reference Documentation (Jacobian calculations can use lot of disk space).

2.2. Installation procedure

Invoking the UNIX commands bunzip2 and tar creates the 4A/OP main directory `4AOP-version/` just below the current directory including all necessary subdirectories. Simply type

```
bunzip2 4AOP-version.tar.bz2
tar xvf 4AOP-version.tar
```

in order to uncompress and "untar" `4AOP-version.tar.bz2`.

Directory `4AOP-version/` should now have been created and should contain the following files and subdirectories:
`Makefile`, `README`, `README.user.in`, `VERSION`, `configure`, `configure.in` and the directories `model/`, `gui/` and `doc/`.

There are two levels of installation: system and user. If you only want a single-user installation, read section 2.2.1 and then go directly to section 2.3. In case of multi-user, section 2.2.2 will describe you how to proceed to install for the different users.

2.2.1. System installation

2.2.1.1. Summary

Here are the instructions to build the main installation of 4A/OP, from the main directory `4AOP-version/`:

```
./configure
make
```

The user has to install atlas files if they are not included in this archive (i.e. the directory `model/datatl` is empty). If the user atlas directory is `/aaaa/atlas`, the user can do either

```
cp /aaaa/atlas/* model/datatl
```

or

```
rmdir model/datatl
ln -s /aaaa/atlas model/datatl
```

2.2.1.2. Details

The configure step creates files (like makefiles) that need system configuration information. If this step leads to incorrect settings, you can modify these files by hand. In particular, you need to edit `Makefile.4a` for compiler settings.

The make step compiles. The compilation generates libraries, object files and executables.

2.2.1.3. Contents of the system installation

`4AOP-version.tar.bz2` contains the entire 4A/OP program (possible exception: large data base files which might be stored separately). There are several directories that should be included in the installation (i.e. the directory where `./configure` and `make` have been done; see above):

`model/` The model itself. This comprises several subdirectories with Fortran 90 source files, makefiles, UNIX shell scripts, user input files and several other files including all necessary input and test data for 4A/OP.

`gui/` The graphical user interface for the model. This comprises Tcl/Tk sources and library, gnuplot scripts and on-line help for the GUI.

`doc/` The documentation for 4A/OP.

Be sure then that the directory `model/` contains the following subdirectories:

datatl/	datscat/	isrf/	outputascii/	scslib/
datatm/	guiparam/	lib/	outputbin/	srclib/
datemis/	input/	libexe/	scripts/	

The subdirectory `model/libexe/` should contain the executable files `aaaa, atmformat, lirespc4a, lirespi4a, statanalys`.

Six user input files are necessary to completely specify all 4A/OP input parameters. These files are read by the 4A/OP main program `aaaa` through several subroutines which set up all input parameters required for the simulation definition.

- One user interface file has a predefined filename: para4a.dtp. It is located in the subdirectory input. This ascii file has to be edited manually (or interactively if the user does use the GUI) in order to set several switches (e.g., trace gas selection, convolution mode), and to set various computational parameters, to specify the wave number range of interest, etc. (see section 3.1.2.1);
- Another user interface file is needed if aerosol contribution is required; its predefined filename is parascat.dtp. It is located in the subdirectory input. This ASCII file has to be edited manually (if the user does not use the GUI) in order to set several parameters, to specify the aerosol model, the number of aerosol layers, etc. (see section 3.1.2.2);
- Four other user input files are necessary:
  - One deals with the atmospheric state (profiles) to be simulated (see section 6.1), located in the subdirectory datatm,
  - One with the spectral emissivity (see section 6.2), located in the subdirectory datemis,
  - One with the aerosol physical parameters (see section 6.4), located in the subdirectory datscat,
  - And one with the instrument function (see section 6.3) for an instrument simulation case, located in the subdirectory `isrf`.

2.2.2. User installation

If a user wants to install a local 4A/OP working directory, go to the 4A/OP installation directory (i.e. the directory where `./configure` and `make` have been done; see above) and type:

```
make INS_DIR=/home/user/4AOP installuser
```

where `/home/user/4AOP` is the 4A/OP install directory for the user. See the `README` file in the user directory (`/home/user/4AOP` in our example) for instructions to use 4A/OP as a user.

2.3. Was the installation successful ?

In order to check if 4A/OP has been correctly installed, some reference input files have been defined and some reference output files created (on SUN station !). All relevant test input files are located in directory `model/input/` and test output files are located in directory `model/outputascii/`.

A UNIX shell script in the directory `model/srclib/`, `CasTest` (for details, see part III.3 in the Reference documentation), directs 4A/OP to produce output corresponding to some sample problems.

1. Execute 4A/OP for several tests:

   ```
   cd model/srclib
   ./CasTest
   ```

   Please note that this run test as well as the test sequence described below will use the following user input files which already exist in `model/input/` as well as in `model/datatm/`, `model/datemis/`, `model/datscat/` and `model/isrf/`:

   ```
   input/para4atest*.dtp
   input/parascat*.dtp
   datatm/atm4atest*.dsf
   datemis/spemis*.dat
   datscat/aerosols_*.dat
   isrf/isrf*.dsf
   ```

2. Now the following files should have been created:

   ```
   datatm/atm4atest*.ddb
   isrf/isrf*.ddb
   outputbin/spc4a*test*.ddb
   outputbin/spi4a*test*.ddb
   outputascii/spc4a*test*.plt
   outputascii/isrf*.plt
   srclib/CasTest.log
   ```

3. The output is written to files spi4a0001testtest1infb1.plt, spc4a0001testtest1testb1.plt, etc, in the directory outputascii/. To ensure the code is operating properly, you should compare your results with the contents of the corresponding files provided in the release package, named spi4a0001testtest1infb1.plt_ref, spc4a0001testtest1testb1.plt_ref, etc, which were generated by 4A/OP on our Solaris Sun workstation. The contents of these files should be essentially identical with the corresponding *_ref files (note that the numerical results may differ somewhat depending on the computer system and then the compiler used).

The script performs fourteen tests for testing different simulation conditions with 4A. These tests are based on the `~/input/para4atest*i.dtp` (*i = 1,14) input files. After each execution of 4A/OP, the corresponding `sp*i.plt_ref` reference files in `~/outputascii/` and the newly generated `sp*i.plt` files are compared using the UNIX `diff` command. The script performs also three more tests for testing the atmospheric profile and instrument function building.

For all tests, the basic scenario is the same. They only differ in wave number settings, simulated atmosphere state, convolution mode selection, simulated instrument, Jacobian computation (only in Test5) and scattering effects. Test8 is the most time consuming.

3. Preparing the simulation

After 4A/OP has been successfully installed, the program `aaaa` can be easily executed either by using the Graphical User Interface (GUI) or in script mode. See the file `README` in the user installation directory.

Preparing a 4A/OP simulation requires (Figure 1):

- setting up some global variables and simulation definition parameters;
- providing 4A/OP with some input files (atmospheric profiles, emissivities and instrumental function).

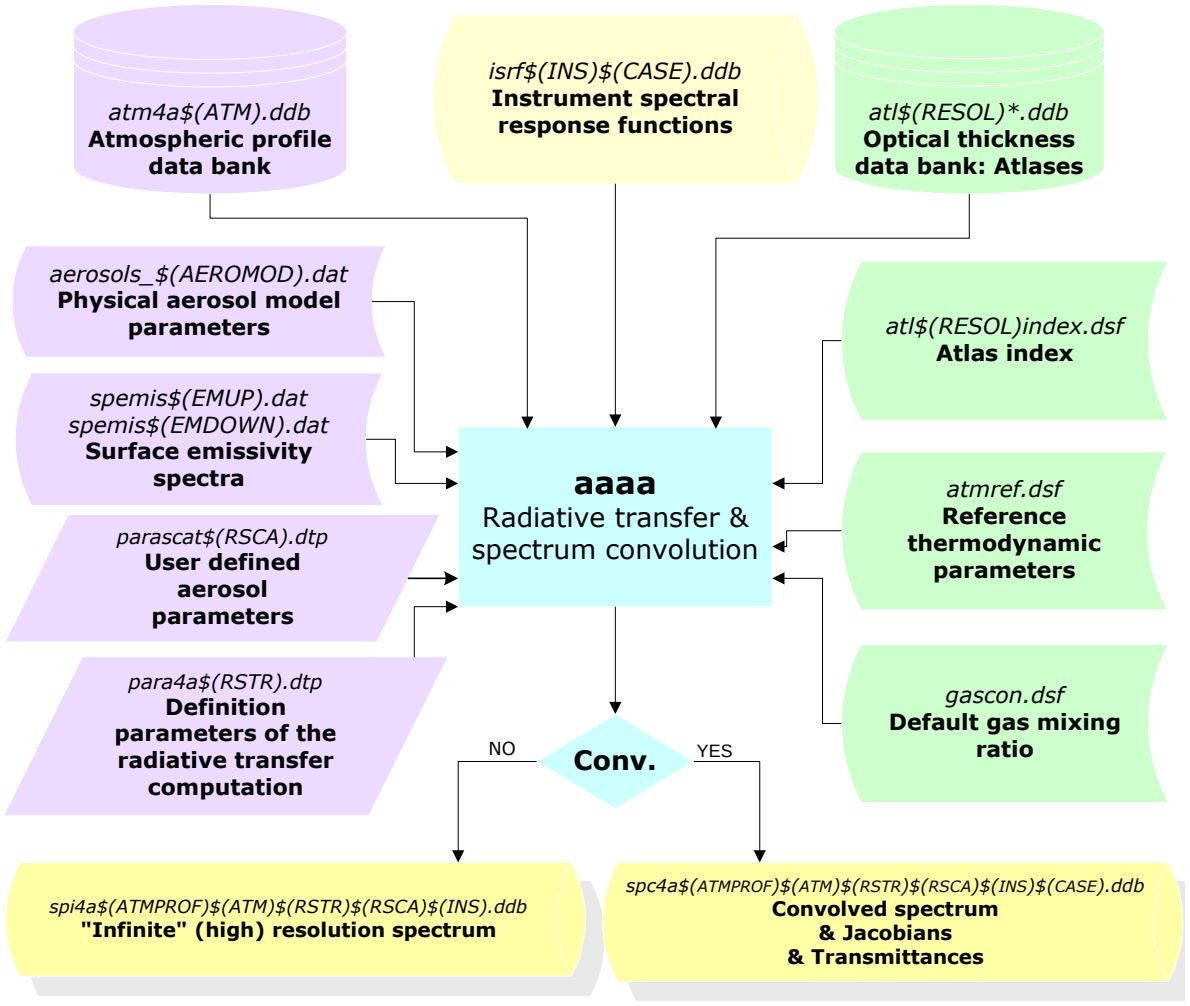

Figure 1: Software flowchart (inputs, outputs & processes).

In this section, input files are already provided and we concentrate on setting up the global variables and simulation definition parameters. See section 6 for more insight in the preparation of new input files.

In script mode, the global variables and the definition parameters are defined manually respectively in the run shell script and in one (or two) parameter file(s). When running 4A/OP in GUI mode, these inputs are defined interactively in menus.

3.1. Preparing the simulation in script mode

3.1.1. Global variables

These variables are user-defined in the run shell script and have to be changed manually in this file. We use as an example the run shell script "run4a_example" located in "model/scripts" directory in your user directory. Open this file. The following lines in the script represent the global variables needed by 4A/OP:

```
set INS=test
set ATM=test
set ATMPROF=0001
set RSTR=example
set RSCA=
set NUMIN=645
set NUMAX=950
set UNIT=4
```

More in detail, the simulation definition parameters are:

- **INS** is the indicator of the simulated instrument (instrument function ISRF) stored in the file named `isrf$(INS)$(CASE).ddb` (in `model/isrf`);
- **ATM** is the indicator of the atmospheric database stored in the file `atm4a$(ATM).ddb` (in `model/datatm`);
- **ATMPROF** is the index of the atmospheric profile read in the atmospheric database stored in the file named `atm4a$(ATM).ddb`.The range of profile indexes and their meaning depend on the atmospheric database;
- **RSTR** is the indicator of the parameter file `para4a$(RSTR).dtp` (in `model/input`);
- **RSCA** is the indicator of the parameter file `parascat$(RSCA).dtp` (in `model/input`);
- **NUMIN** is the beginning in wave number of the extraction for the conversion from the output binary file to an ASCII file;
- **NUMAX** is the end in wave number of the extraction into an ASCII file;
- **UNIT** is the Jacobian unit index for Jacobian ASCII outputs (see the meaning of the different indexes in Table 4 of appendix).

The complete list of global variables is found in Table 4 of the appendix.

3.1.2. Simulation definition parameters

3.1.2.1. Run characterization parameters

You will then modify the model parameter file, named **para4a$RSTR.dtp** (`$RSTR` having been defined as global variable, see above section 3.1.1) and stored in the **input** directory. To refer to the complete meaning of these parameters, please refer to Table 4 and Table 5 of the appendix. Such a file is displayed below (**para4aexample.dtp** in your **input** directory):

```
'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'       100
' Temperature of the upper level                     TUP'        4.
' Index for the Emissivity of the lower level     EMDOWN'       100
' Temperature of the lower level                    TDOWN'        0.0
' Geometric tangent height (km) (LIMB only)         ZGTAN'        0.
```

```
' Angle type definition  (ANGLE,SECANT,UNDEF)     TYP_ANG'     'ANGLE'
' Viewing angle (in degrees or secant value)      SECTTA1'        0.
' Reflexion/emission angle (same unit as above)   SECTTA2'        0.
' Solar zenith angle (same unit as above)       SECTTASUN'       90.
' Earth-Sun distance in astronomical units         ESDIST'        1.
' 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'      910.
' Width of each Atlas spectral band                 DELDS'       15.
' Optical thickness interpolation in temperature   INTOPT'     'YES'
' Indicator of convolution with the ISRF             CONV'     '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.
' Convolution step                                 PASCON'        0.25
' 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  c3h6'
    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   ,
'  xxx   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  c3h6'
    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   ,
'  xxx   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   ,
```

3.1.2.1.1 Viewing and geometric configuration definition

This section defines the terminology used for the viewing and geometric configuration of the radiative transfer computation in Table 1. The various viewing configurations are illustrated by Figure 2 to Figure 10.

TRAJET	SECTTA1	SECTTA2	SECTTAS
Earth View-UP	Viewing (sat.) zenith angle (at surface z=0)	-	-
Earth View-DOWN	Viewing (obs.) zenith angle (at surface z=0)		Solar zenith angle (at surface z=0)
Earth View-DOWNUP	Viewing (sat.) zenith angle (at surface z=0)	Reflexion incidence angle (scaled to surface z=0)	
Earth View-UPDOWN	Viewing (obs.) zenith angle (at surface z=0)		-
Limb-UP	< 1 or ≥ 90° zgtan must be specified	-	-
Limb-DOWN		-	-
Limb-DOWNUP		55° incidence zenith angle at cloud top (cloud above tangent point)	solar zenith angle at cloud top (cloud above tangent point)
Limb-DOWNUP		set to sectta1 (limb view through tangent point)	-
Limb-UPDOWN		55° incidence nadir angle at cloud bottom (cloud above tangent point)	-

Table 1: Terminology for the viewing and geometric configuration of 4A/OP.

Important note 1: User-defined angles need to be scaled to surface at z=0 in order to take into account spherical atmospheres (see section Part III 1.3 of the reference documentation). 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 θ(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 (SECTTASUN is set to 0, 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 SECTTASUN 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.

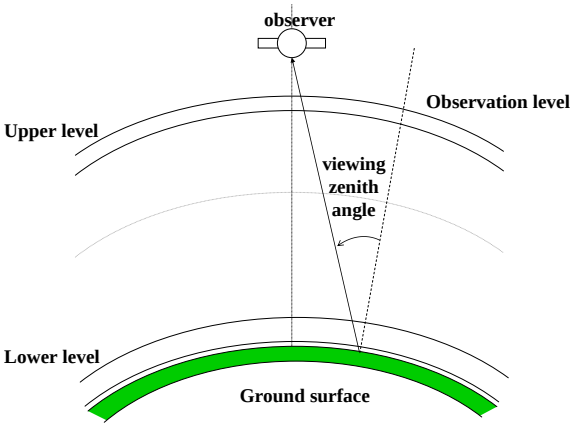

Figure 2: Nadir viewing (*TRAJET* = UP)

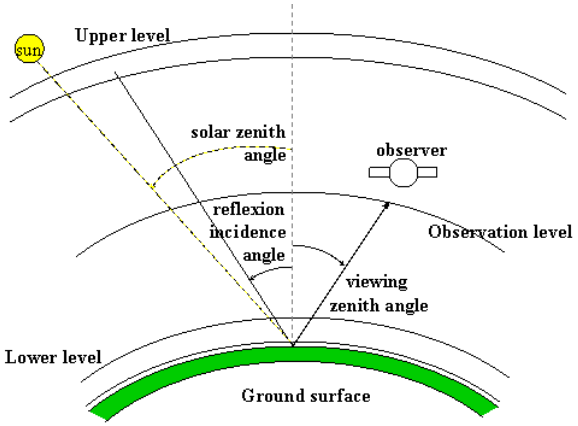

Figure 3: Viewing with ground surface reflexion (*TRAJET* = DOWNUP)

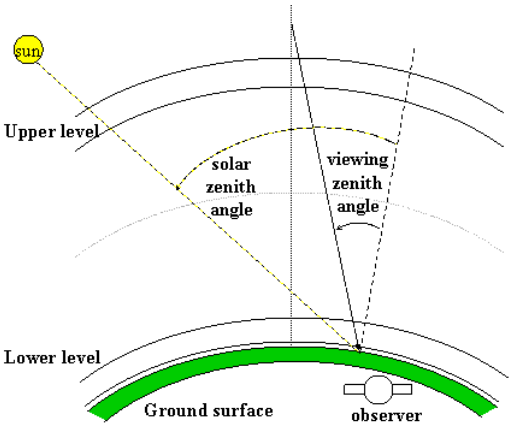

Figure 4: Zenith viewing (*TRAJET* = DOWN)

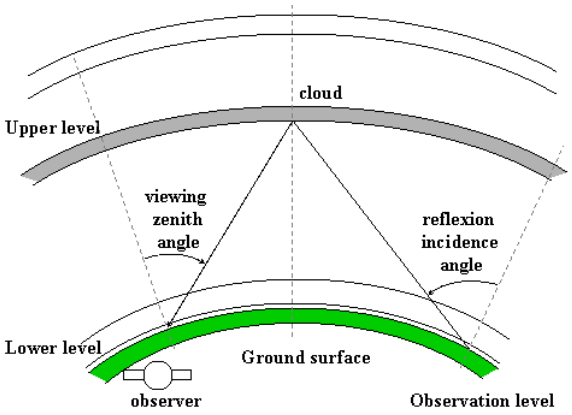

Figure 5: Viewing with cloud reflexion (*TRAJET* = UPDOWN)

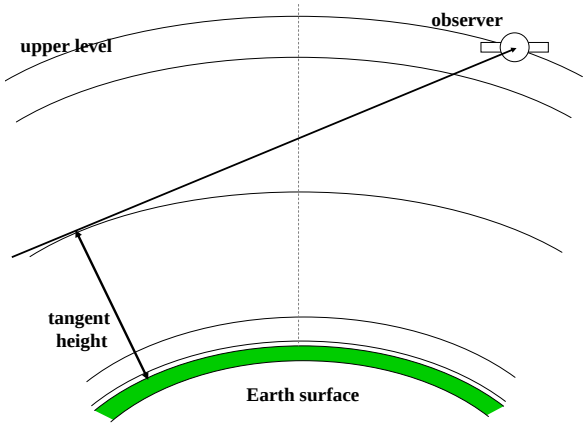

Figure 6: Limb view through the tangent point (*TRAJET* = DOWNUP). The user specifies background temperature and emissivity (cloud or sky), tangent height and the lower level below tangent height. The observer must not be specified above the background level.

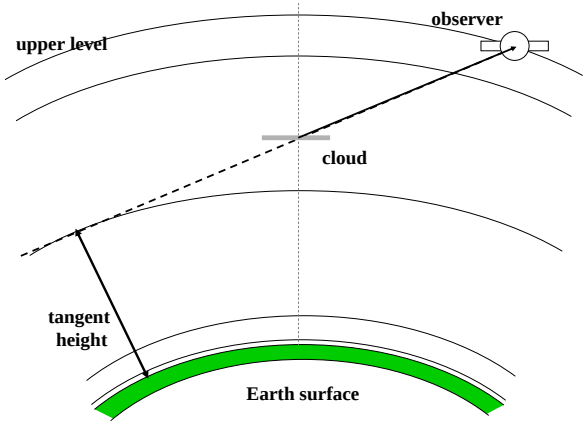

Figure 8: Limb view of a cloud top (*TRAJET* = UP). The user specifies cloud top temperature and emissivity, tangent height, and the lower level above tangent height.

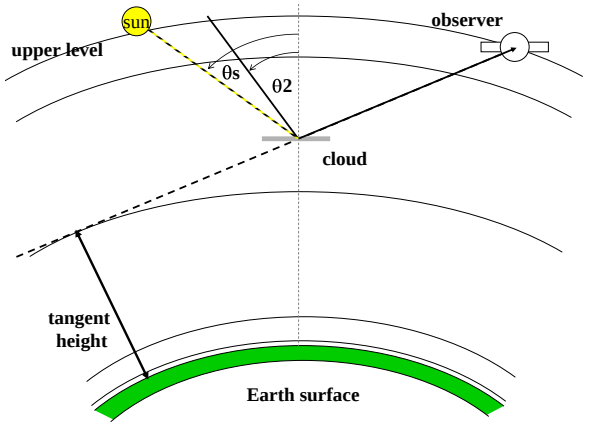

Figure 7: Limb view of a cloud top, including reflection of downward terrestrial and solar radiation (*TRAJET* = DOWNUP). The user specifies background temperature and emissivity (sky or cloud), cloud top temperature and emissivity, tangent height, the lower level above tangent height, and the solar zenith angle θs (if background is sky). θ2 is fixed to 55°. The observer must not be specified above the background level.

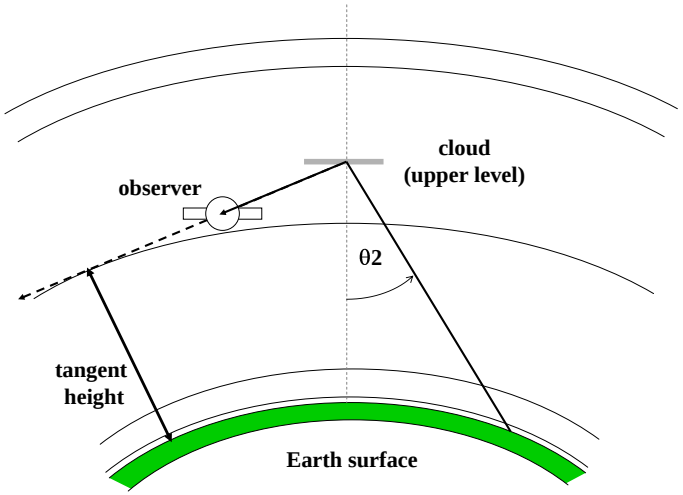

Figure 9: Backward limb view of a cloud bottom, including reflection of upward terrestrial radiation (*TRAJET* = UPDOWN). The user specifies background temperature and emissivity (Earth surface or lower cloud), cloud bottom temperature and emissivity, tangent height, and the upper level above tangent height. θ2 is fixed to 55°. The observer must be specified between upper level and tangent height.

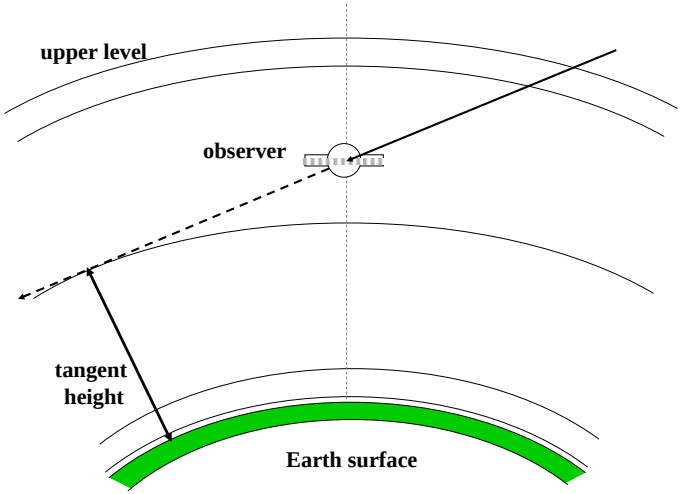

Figure 10: Backward limb view of sky or a cloud bottom (*TRAJET* = DOWN). The user specifies background temperature and emissivity, tangent height, and the observation level above tangent height.

3.1.2.2. Aerosol/ice cloud run characterization parameters

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 (this file is stored in the directory **`input`**). 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.

An example is displayed below, but you can refer to the complete meaning of these parameters with section 5.2.5.3 in the Reference Documentation.

```
' Disort(DIS), SOS(SOS) or no scattering (NO)    FLAG_SCAT'      'DIS'
' Number of aerosol layers (<=Cst_nlaeromax)       NLAERO'      2
' Aerosol model                                  AEROMOD'      'mitr00' 'soot00'
' reference optical thickness at 10 micron       AOTREF'      1.5 0.5
' Vertical level where aerosol is present        LAYAERO'      33 35 38 38
' Single scattering albedo (const)               OMEAERO'      0. 0.
' Asymmetry parameter (const)                      GAERO'      0. 0.
' Number of phase function Legendre coeffs.         NMOM'      10
' Number of DISORT streams (even and >2)          NSTREAM'      10
' DISORT conv. crit. for beam source ]0.d0,0.1d0]   ACCUR'      0.009
' DISORT print flag, input parameters             PRNT(1)'      .FALSE.
' DISORT print flag, fluxes                       PRNT(2)'      .FALSE.
' DISORT print flag, radiances, user polar ang.   PRNT(3)'      .FALSE.
' DISORT print flag, transmittivity/albedo        PRNT(4)'      .FALSE.
' DISORT print flag, phase function moments       PRNT(5)'      .FALSE.
' DISORT header string (<128 characters)        DISHEADER'      '4AOP_DISORT
test'
```

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** (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 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. 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. (ie `RSCA` ≠ ' ')

NB: A way to accelerate the computing time in case of the use of 4AOP with DISORT is in progress.

3.2. Preparing the simulation in GUI mode

The simplest way to use 4A/OP is to launch its Graphical User Interface:

```
./4Agui
```

This program is in the **`gui/bin`** directory of the system installation. Make sure this directory is in your user PATH or launch `4Agui` from this directory.

Launching it while open on the general page of 4A/OP GUI (see Figure 11).

The GUI allows the user to create a basic 4A/OP input file by selecting values with buttons, pull-down menus, and text fields. It should be particularly useful to the new user.

The GUI aims at wrapping the existing software (and the process management) in order to avoid source modifications. The dependency between parameters are kept and are implemented as a black box for the user. The GUI guides the user's choices according to the parameter dependencies: some entries (labels) are disabled if not required and warning messages appear if the user enters no compatible values with his/her experiment definition.

The programming language used to implement the GUI is Tcl/Tk (Tool command language / Tool kit), version 8.4 at least, and it uses the library BWidget.

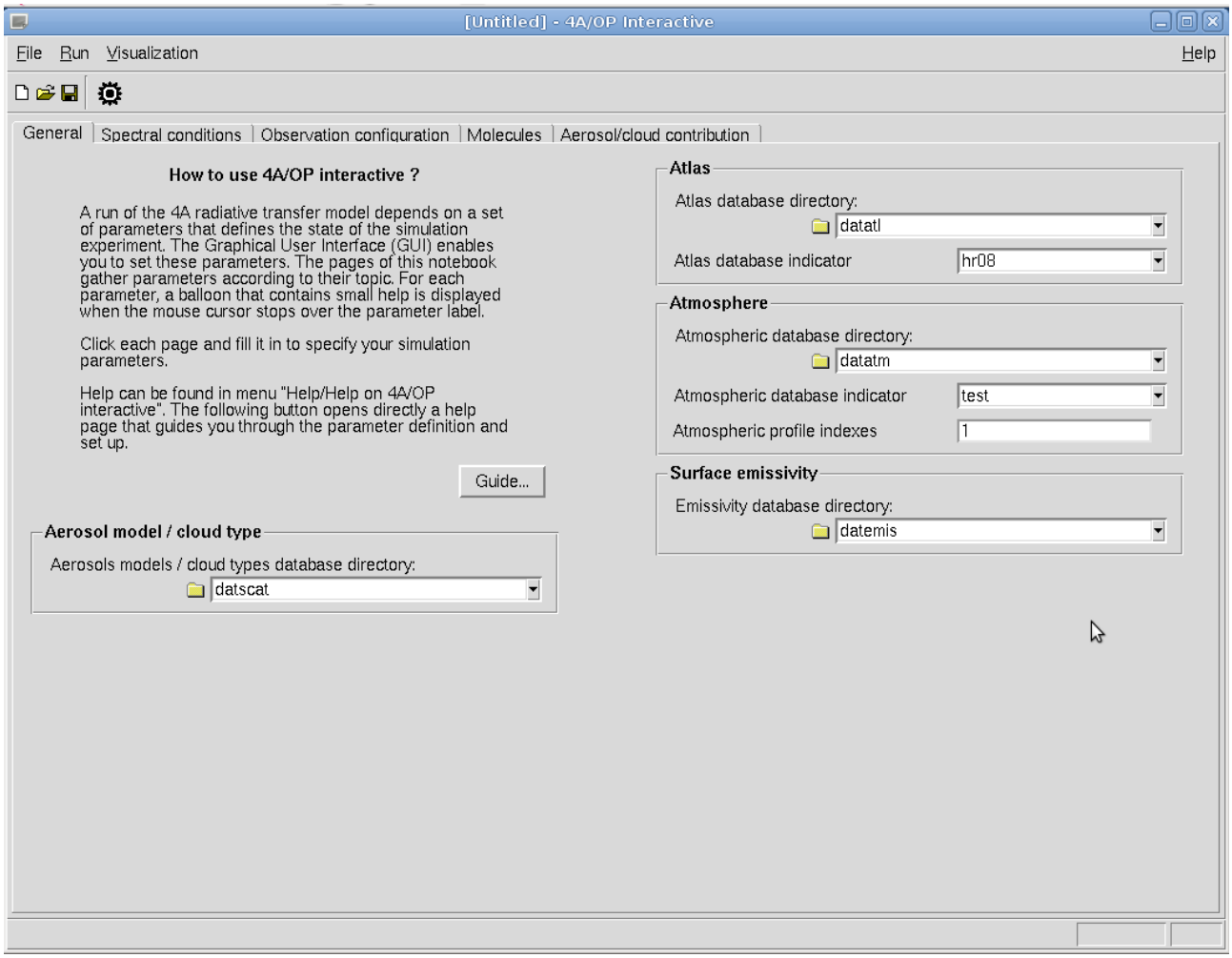

Figure 11: Screenshot of the GUI first page that shows the general parameters required for a 4A run.

3.2.1. Produce a GUI parameter file

The 4A/OP GUI allows through its five pages to define the simulation parameters: "General" (see Figure 11), "Spectral conditions" (see Figure 12), "Observation configuration" (see Figure 13), "Molecules" (see Figure 14) and "Aerosol/cloud contribution" (see Figure 15).

By clicking the button "Guide" available in the page "General", a help/guide page appears (see Figure 16). It explains step by step the use of the GUI for the parameter definition.

Go through the five pages of the GUI and adjust to your own simulation parameters.

When changing the values in five pages of the 4A/OP GUI, a parameter file with a format specific to the GUI is created in order to store all the parameters (global variables and definition parameters) required to run the 4A model from the GUI.

Save your experiment state into a file using "Save" or "Save as" (menu "File" of the GUI).

This GUI parameter file has extension ***.4a**.

3.2.2. Open an existing GUI parameter file

Available GUI parameter files can be open (menu "File / Open") for loading parameters that have been already used and recorded. Such parameter files are saved by default in the directory `model/guiparam/`.

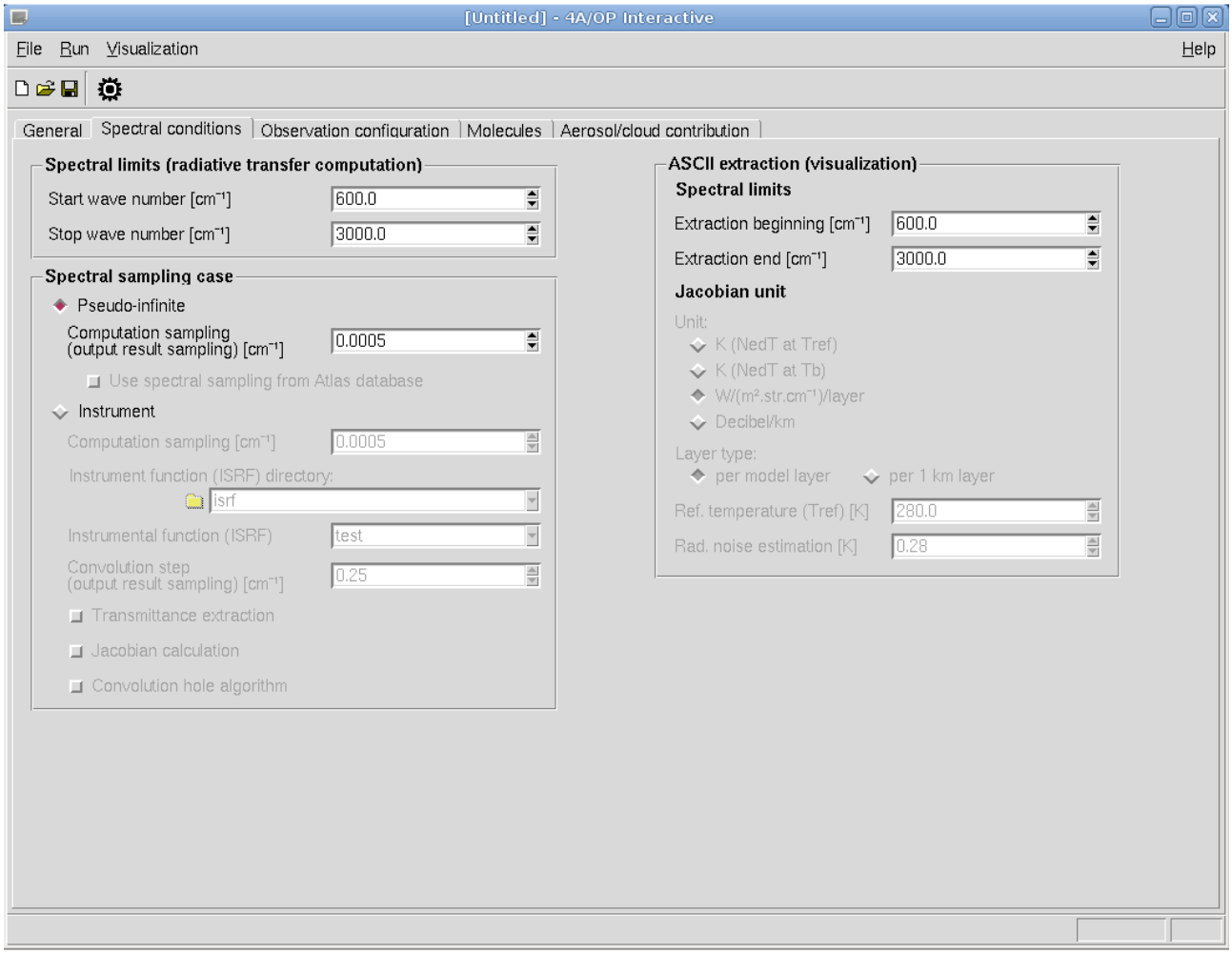

Figure 12: Screenshot of the page concerning the spectral condition parameters for the simulation.

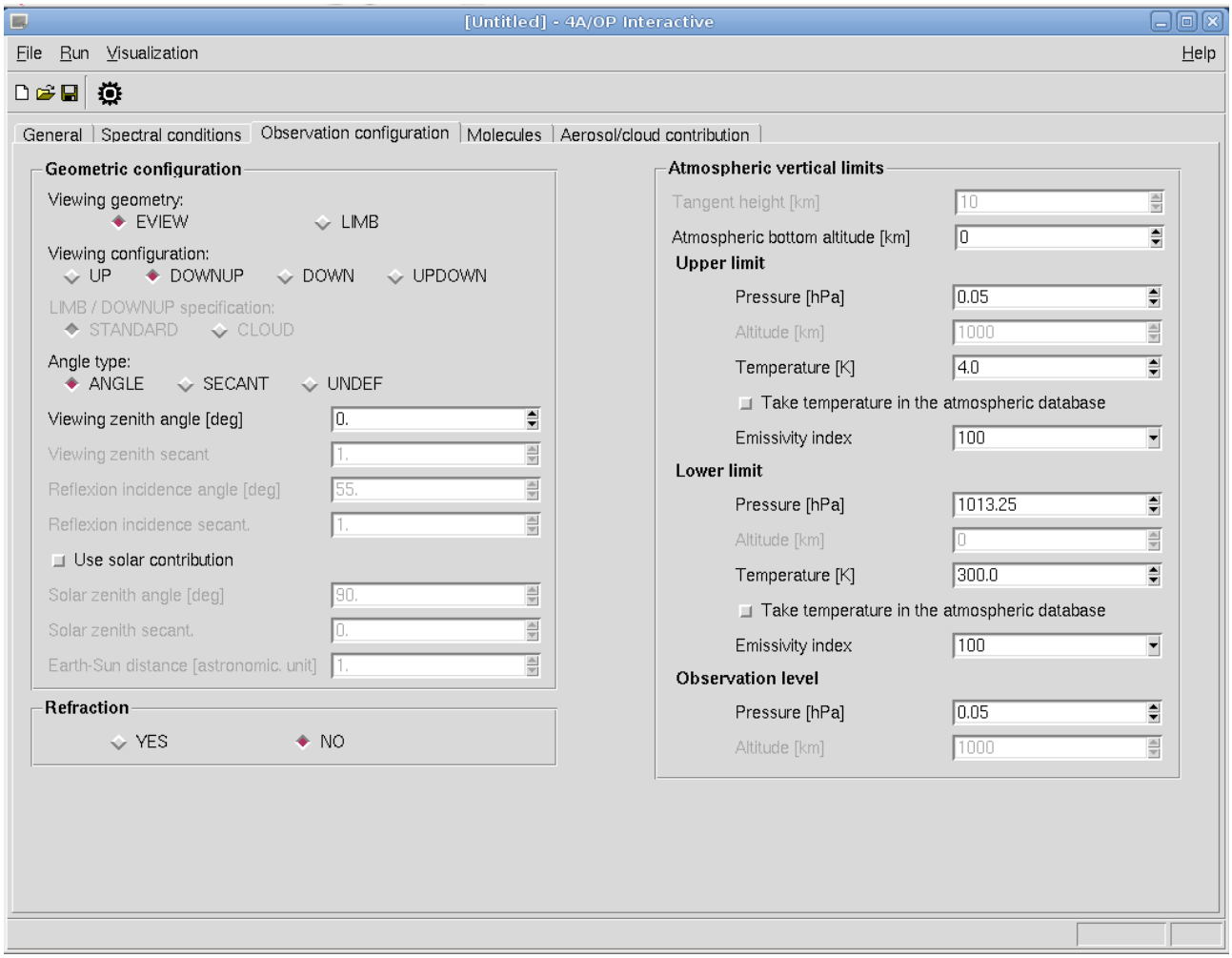

Figure 13: Screenshot of the page concerning the observation configuration parameters.

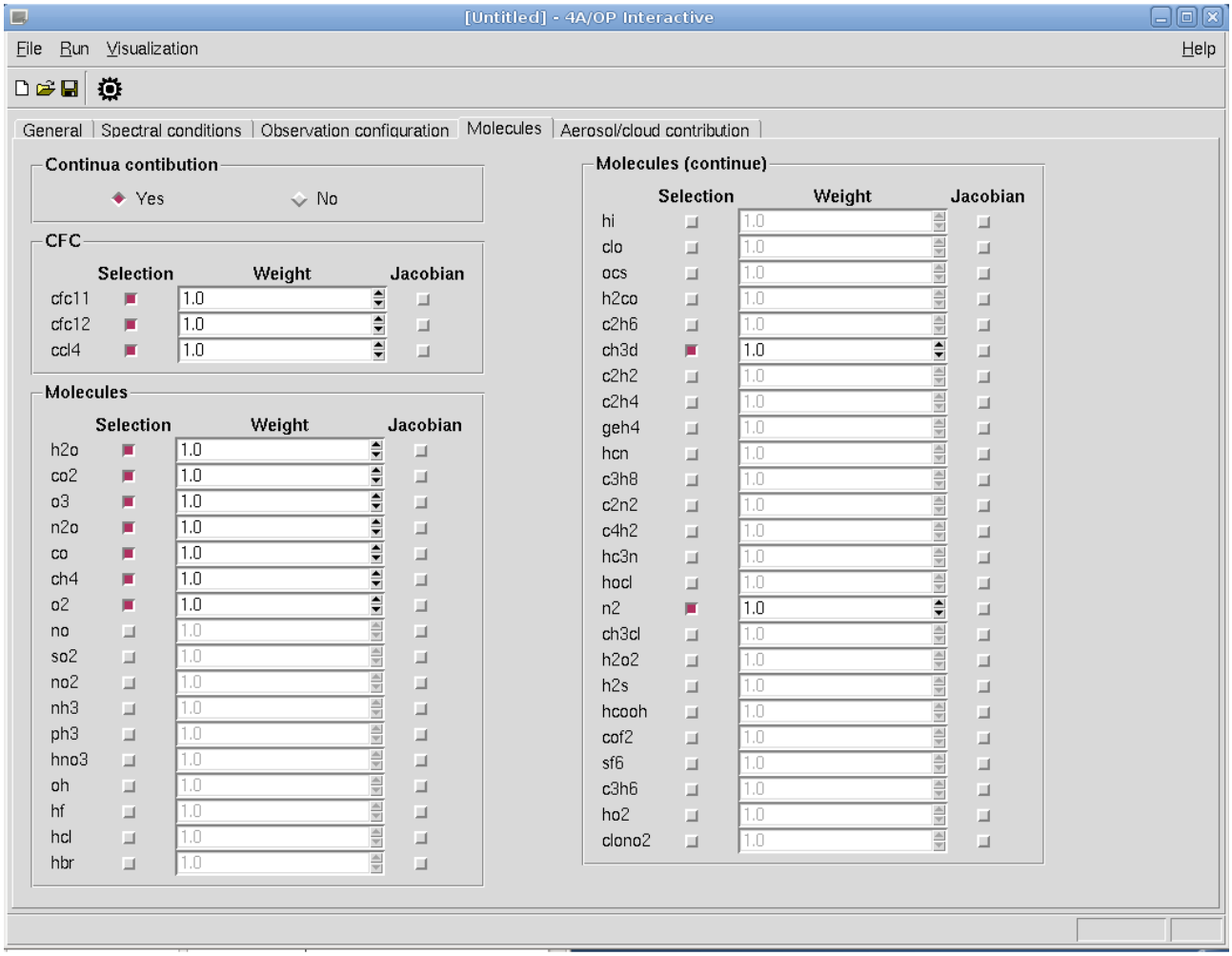

Figure 14: Screenshot of the page concerning the molecule parameters.

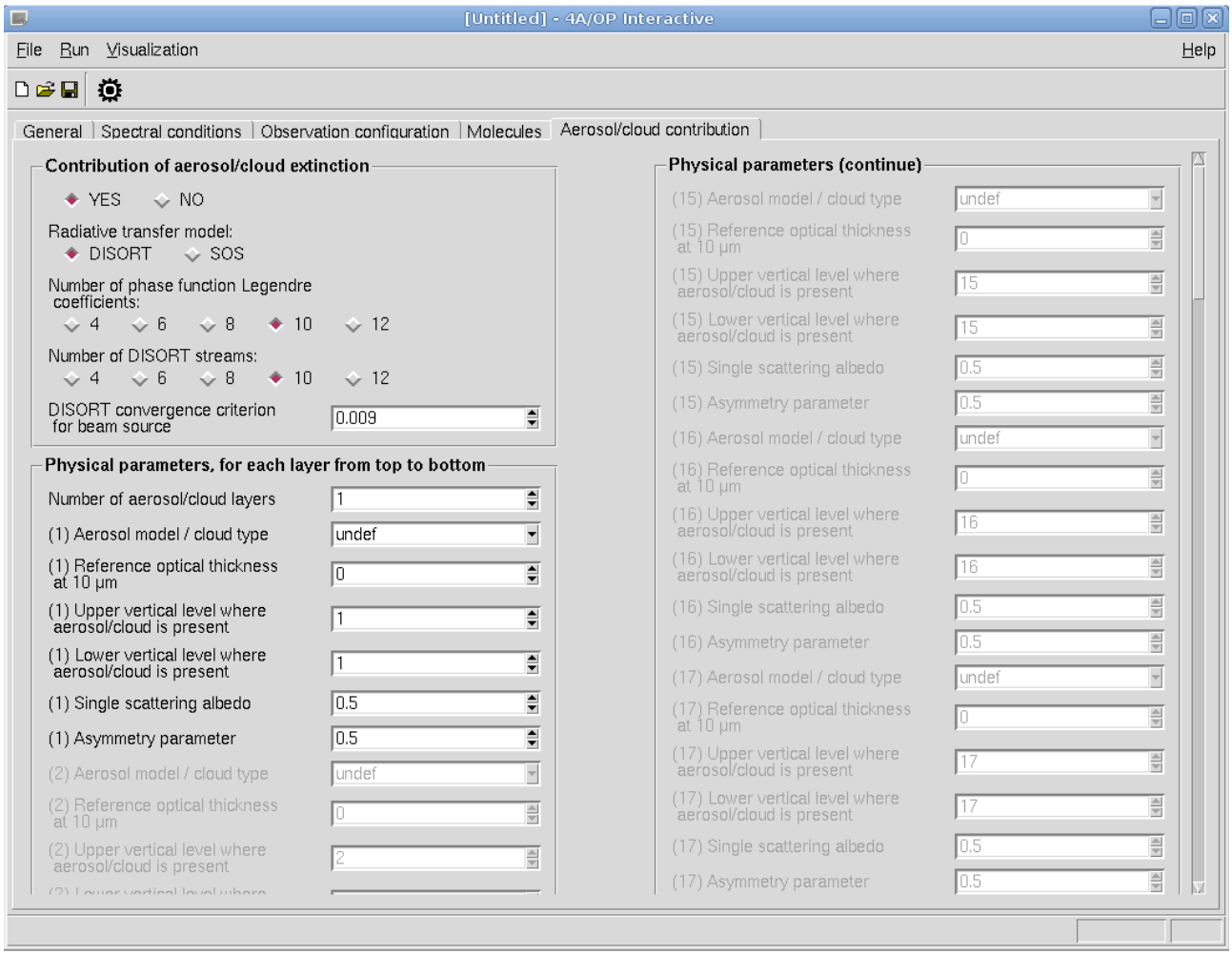

Figure 15: Screenshot of the page concerning the aerosol/cloud parameters.

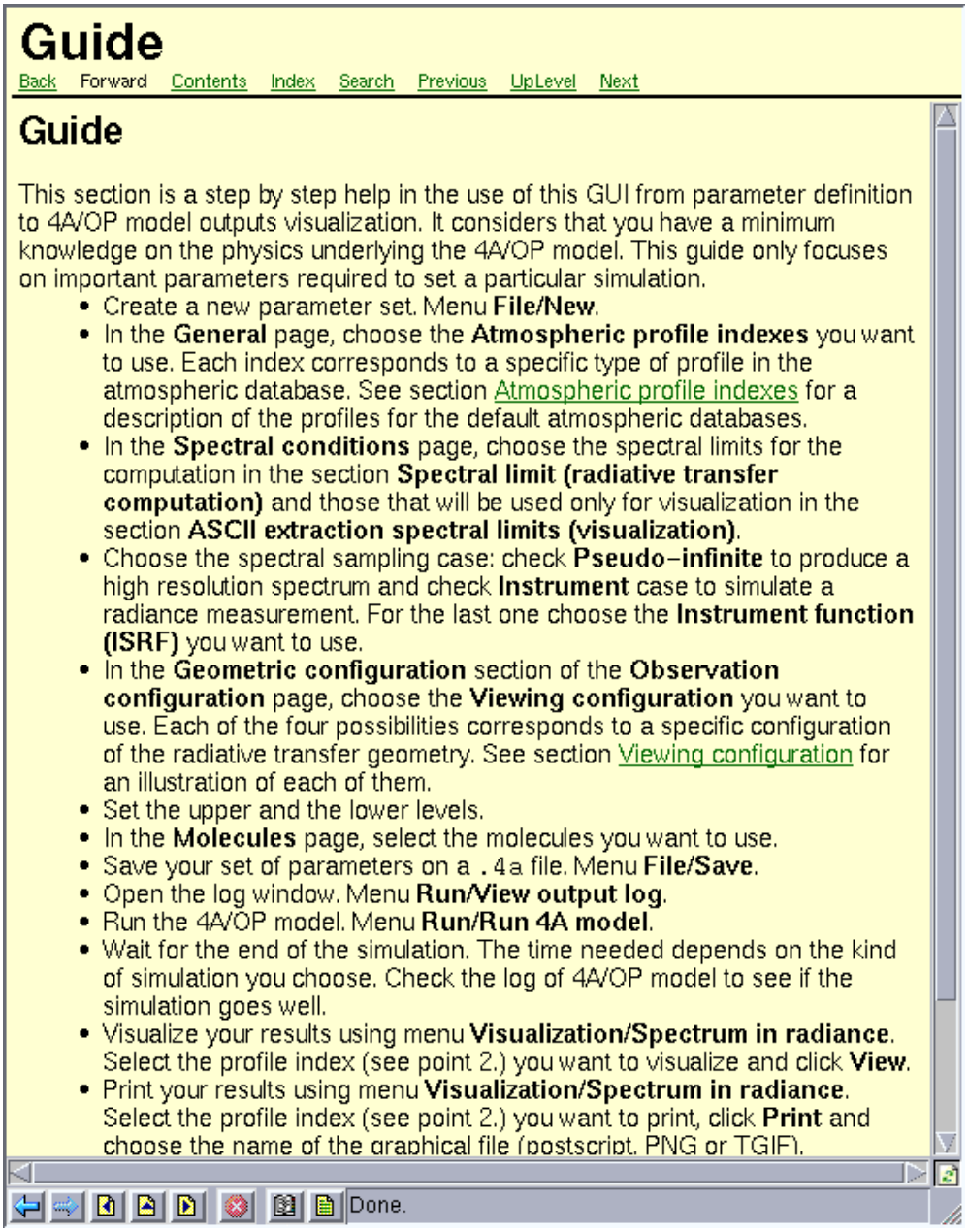

Figure 16: Screenshot of the GUI guide page that appears by clicking the button "Guide".

4. Running 4A/OP

4.1. Running 4A/OP in script mode

In the run shell 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: `runlirespi4a`
- in the case of spectra convolved with an instrument function: `runlirespc4a`

If the user wants BINARY output only, the corresponding instruction is: `make runspi4a`

`make runspc4a`

Going back to our example, you need to type:

```
./run4a_example
```

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

The entire command line executed in the run shell script is (for a convolved spectrum):

```
make -f ../srclib/makefile runlirespc4a INS=$INS CASE=$CASE ATMPROF=$ATMPROF
ATM=$ATM RSTR=$RSTR RSCA=$RSCA NUMIN=$NUMIN NUMAX=$NUMAX UNIT=$UNIT
```

where the variables are the global variables described above (section 3.1.1).

If you type the instruction "make help" (from the directory **`model/srclib`**), you get a list of executions available when using the program make.

4.2. Running 4A/OP in the GUI mode

Run of 4A/OP is launched via a menu choice.

Select `"Run 4A/OP"` in the menu `"Run"`.

The GUI produces a script file to run 4A from the user's choices presented above (see section 0). The variable **`$RSTR`** in script mode (cf. section 3.1.1) is equal to the character string defining the GUI parameter file (without extension). For example, if one creates a set of parameters that is saved under the name **`test.4a`**, the GUI creates the required 4A/OP parameter file called **`input/para4atest.dtp`** and a run shell script named **`scripts/test`**. Then, the model runs.

The user has the possibility to monitor the run development by watching the log window that can be open and closed at any time with the menu "Run / View output log" (see Figure 17). The model runs in background then the GUI remains active and the user can move all over the GUI. The user is able to stop the run at any time thanks to an aside "Stop" button.

Figure 17: Screenshot of the log window showing a run development example. This window appears by choosing the menu entry "Run / View output log".

5. Viewing 4A/OP simulation results

5.1. Outputs in script mode

4A/OP produces only one binary file containing all results. An interface program (sections II.1.1.4 and II.1.1.5 in the reference documentation) 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 below (Table 2) according to the simulation case.

Output data in ASCII files always include radiances and brightness temperature and can also include other quantities (Jacobians and/or transmittances) for an instrument simulation case.

We will now illustrate the 4A output capabilities for radiance spectra through two examples. In addition, the use of a statistic tool on 4A/OP outputs is also shown.

Simulation case	Content	Binary file names In directory outputbin	ASCII file names In directory outputascii
"Pseudo-infinite"	High-resolution radiances	spi4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(INS).ddb	spi4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(INS)b1.plt
"Instrument"	- Convolved radiances	spc4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)(INS)(CASE).ddb	spc4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)(INS)(CASE)b1.plt
	Possible additional quantities: - Partial derivatives of the radiance (Jacobians) with respect to the temperature		dtj4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)(INS)(CASE)b1.plt
	- Jacobians with respect to the molecule mixing ratio - Jacobians with respect to the surface emissivity		dcj4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)(INS)(CASE)b1.plt
	- Transmittances		dto4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)(INS)(CASE)b1.plt

Table 2: 4A/OP output file description

5.1.1. Output file description

5.1.1.1. High-resolution radiance spectra

Make a copy of the run shell script (in your **`scripts`** directory):

cp `run4a_example run4a_test1`

Then, edit directly the command lines in the new run shell script:

```
set INS=inf
set CASE=
set ATM=test
set ATMPROF=0001
set RSTR=test1
set RSCA=
set NUMIN=719
set NUMAX=721
```

and change the target `runlirespc4a` into `runlirespi4a` in the following command line:

```
make -f ../srclib/makefile runlirespi4a INS=$INS CASE=$CASE ATMPROF=$ATMPROF
ATM=$ATM RSTR=$RSTR RSCA=$RSCA NUMIN=$NUMIN NUMAX=$NUMAX UNIT=$UNIT
```

Then, simply run the script:

```
./run4a_test1
```

In this first example, 4A/OP computes the very high-resolution radiance from 719 to 721 cm^{-1} every $5x10^{-4}$ cm^{-1} ("infinite" spectrum). The corresponding input parameter file **`para4atest1.dtp`** in your **`input`** directory contains all model parameters for this simulation.

The spi4a0001testtest1infb1.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 18. A good way to verify that 4A/OP is operating correctly on your system is to use your favourite graphics software to read spi4a0001testtest1infb1.plt and compare the results visually.

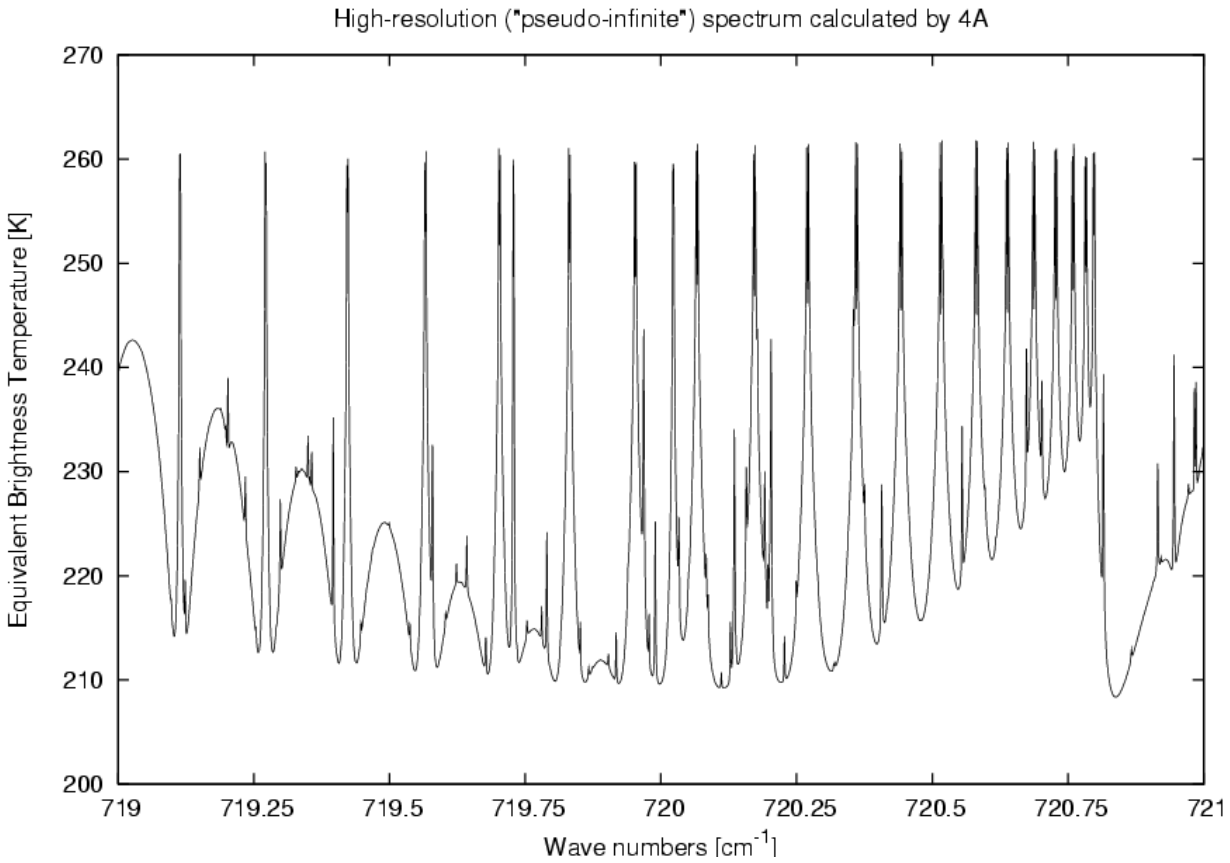

Figure 18: High-resolution radiance computed with 4A from 719 to 721 cm^{-1} in equivalent brightness temperature (test1).

5.1.1.2. Convolved outputs

5.1.1.2.1 Convolved radiance spectra

Now, make a new copy of the run shell script (in your **`scripts`** directory):

cp `run4a_test1 run4a_test8`

Then, edit directly the command lines in the new run shell script:

```
set INS=iasi1c
set CASE=
set ATM=test
set ATMPROF=0001
set RSTR=test8
set RSCA=
set NUMIN=645
set NUMAX=2760
set UNIT=1
```

Then, simply run the script:

```
./run4a_test8
```

In this second example, 4A computes the convolved radiance that would be measured by the instrument IASI, from 645 to 2760 cm^{-1} with resolution of $2.5x10^{-1}$ cm^{-1} (IASI spectrum sampling). The corresponding input simulation parameter file is **`para4atest8.dtp`** in your **`input`** directory.

The spc4a0001testtest8iasi1cb1.plt file contains the 4A/OP output corresponding to this input. Columns 1 through 3 are: idem as the first example above. The results for this second example are shown in Figure 19.

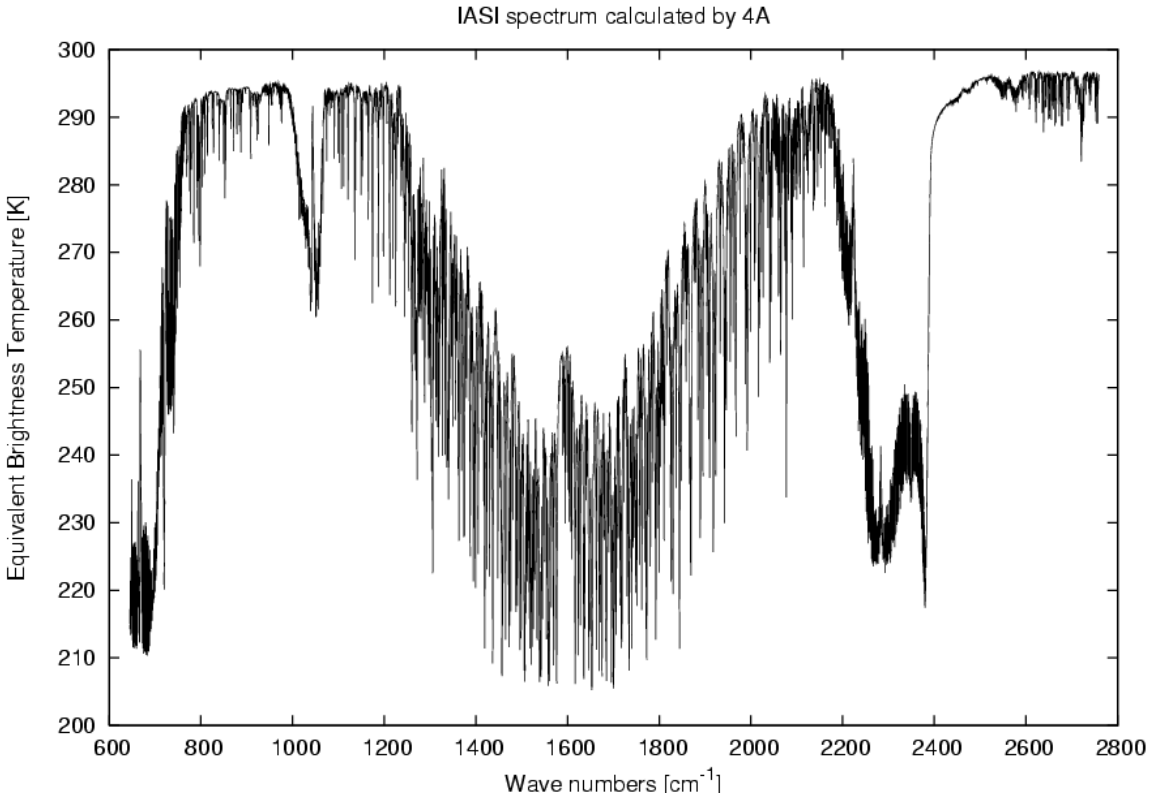

Figure 19: Convolved radiance (in in equivalent brightness temperature) that would be measured by the instrument IASI, from 645 to 2760 cm^{-1} (test8).

5.1.1.2.2 Convolved Jacobians

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

The output unit for Jacobians depends on the user choice (see global variable UNIT in Table 4 of the appendix). 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 available unit: 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:

- 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/her own tool.

Important note:

The convolved Jacobians with respect to the surface emissivity are contained in the file for the Jacobians with respect to gas mixing ratio `dcj4a$(ATMPROF)$(ATM)$(RSTR)$(RSCA)$(INS)$(CASE)b1.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.

5.1.2. Statistics tool

A statistics tool is available for the computation of the difference between two spectra and the corresponding statistics (see section II.1.2.10 in the reference documentation 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`.

Now, make a new copy of the run shell script (in your **`scripts`** directory):

cp `run4a_test8 run4a_test8b`

Then, edit directly the command line in the new run shell script to change the atmospheric profile from 0001 to 0002:

```
set ATMPROF=0002
```

Then, simply run the script :

```
./run4a_test8b
```

Then, you can compute the difference between 4A/OP outputs of test8 and test8b.

Create a new script file (as above) by replacing the command line by the following line:

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

See the complete list of simulation definition parameters in table 7 of the reference documentation (Section 5.2.5.1).

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 in the reference documentation in Section 5.3.4. Use your favourite graphics software to read the two files **`stat_0002test8iasi1c_0001test8iasi1c.plt`** and **`classstat_0002test8iasi1c_0001test8iasi1c.plt`**.

5.2. Outputs in the GUI mode

The visualization displays Gnuplot graphs of results that are the spectra, Jacobians (if they are calculated) and transmittances (if they are extracted). The "Print" entry (figure below) appears directly from the graph and allows to save it in a file (postscript, png or gif file).

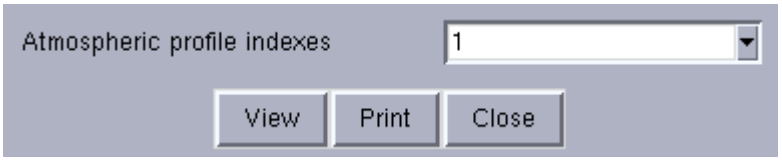

Figure 20: Screenshot of the window that appears by choosing the menu "Visualization / Spectrum in ...".

The visualization can also display Gnuplot graphs of used atmospheric profiles and the difference between two spectra.

5.2.1. Zoom

The user can select a new visualization range by zooming on the graph with the right mouse button. To go back to the previous range, the user can simply type "p" on the Gnuplot window.

5.2.2. User specific plots

You can personalize the graphs produced by the GUI. To do that:

1. Copy the default Gnuplot scripts which are installed in the directory gnuplot of the GUI installation. Put them on your own directory. Be sure that these scripts keep their original names;
2. Edit these scripts to fit your needs using standard Gnuplot commands;
3. Modify the field "Gnuplot scripts directory" in the "Preferences" window (menu "File/Preferences") to point at the directory of your Gnuplot scripts.

Original Gnuplot scripts contain some special codes used by the GUI which are names of variables that are substituted when the graph is actually plotted. They begin with the sign $ (dollar). For example, `$dataFile` is replaced by the name of the 4A/OP output file to plot.

5.2.3. Examples

Prepare the simulation in GUI mode (see section 0) according to the configuration in test 1 (see section 5.1.1.1): viewing the results will lead to a graphical output identical to Figure 21.

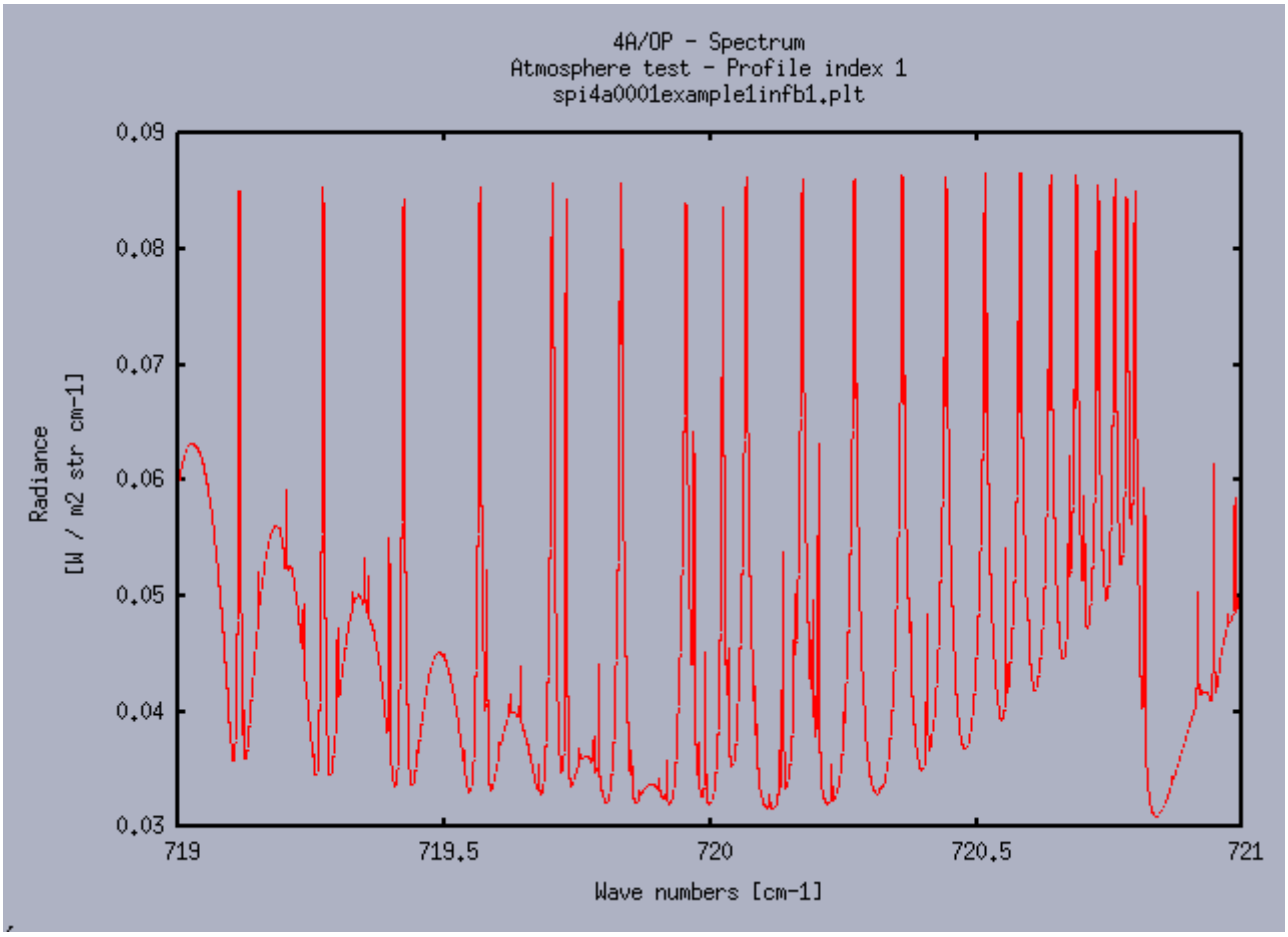

Figure 21: Screenshot of the Gnuplot window showing an example of a spectrum resulting from the current simulation. This window appears by choosing the menu entry "Visualization / Spectrum in radiance". The name of the ASCII file (*.plt) storing the spectrum is below the title "4A/OP – Spectrum".

Now, prepare the simulation in GUI mode by opening an existing parameter file (see section 3.2.2) that corresponds to example 2. In the "Spectral conditions" panel, enables the "Jacobian calculation" and choose as "Jacobian unit" the item "K (NedT at Tref)". Viewing the results will lead to a graph identical to Figure 22. Clicking on right mouse button will enable you to zoom in and restrain the selected range of wave numbers (Figure 23).

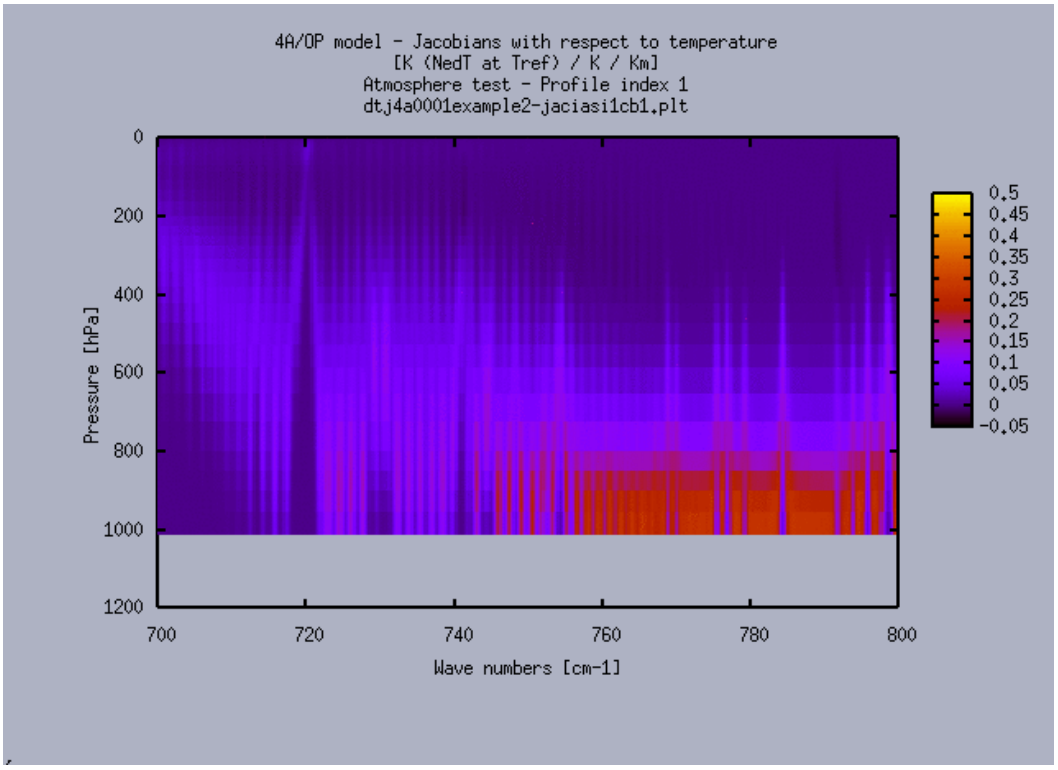

Figure 22: Screenshot of the Gnuplot window showing an example of Jacobians with respect to the temperature. This Gnuplot window appears by choosing the menu entry "Visualization / Jacobians (temperature)".

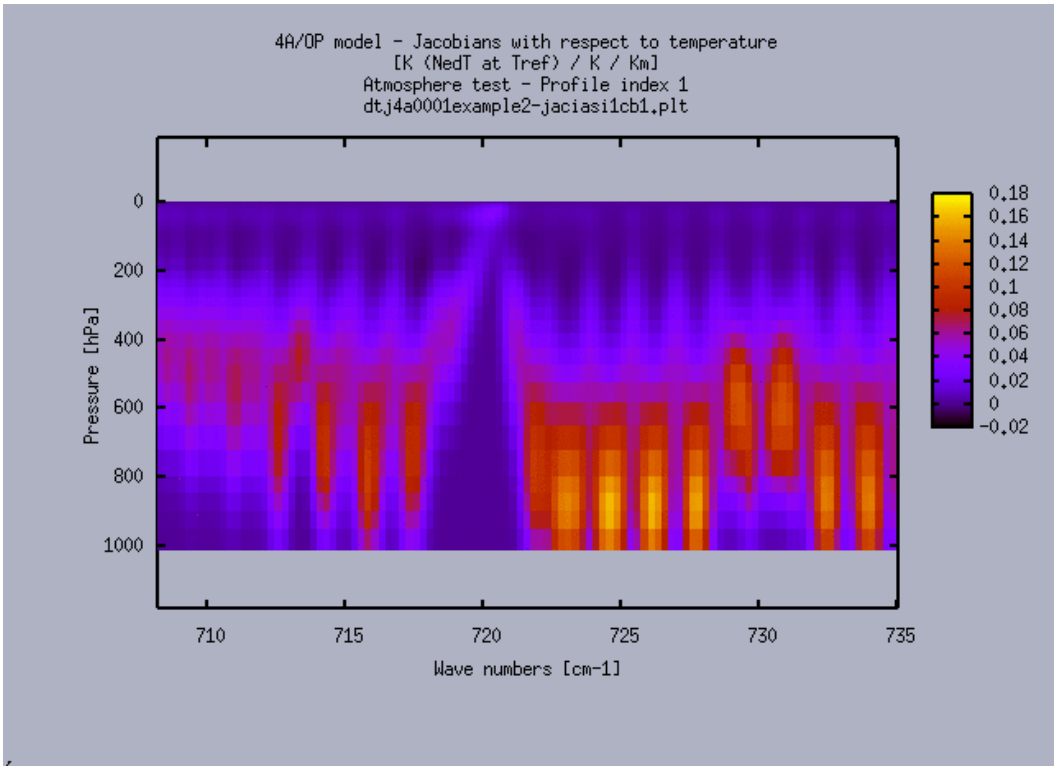

Figure 23: Same as Figure 22 for a selected range obtained by zooming on the graph with the right mouse button.

6. Generating your own 4A/OP input files

6.1. Atmospheric profiles

A preprocessing program of 4A, **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. This binary database, is named **atm4a$ATM.ddb** ($ATM has been defined in the global variables, section 3.1.1) and placed in the directory **model/datatm**.

By default, 4A/OP uses the file **atm4atest.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 **atm4atest.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.

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.

6.1.1. First step: preparation of the user atmosphere file

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 **atm4atest.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**.

6.1.2. Second step : modification of the appropriate atmosphere reading subroutine

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

```
! Atmosphere "test"
! -----------------
if (atm(1:4) == 'test') then
   read(jin1, *, end=2000) nopro, tsol, psol, sectta,           &
         (t(i),i=1,40),(rolvl(i,1),i=1,40),(rolvl(i,3),i=1,40)
   iat = iat + 1
   nlevel = 40
end if
```

For a complete description of the content of the binary database for atmospheric profiles (**`atm4a$ATM.ddb`**), please refer to the reference documentation (Section 5.2.1).

6.2. Spectral emissivity

The files describing the infrared surface emissivity spectrum for different types of surface used for a 4A/OP run are placed in the directory **`model/datemis`**. A **`readme.txt`** file explains the emissivity files.

These files can also be either user-defined or default files can be used.
Two types are possible for surface emissivity files here named `spemisXXX.dat`:

- a constant emissivity file; the range of the file suffix value `XXX` is between 000 and 100, that correspond to a constant emissivity value of 0 and 1 respectively;
- a file containing spectral variable emissivities given on a range of wave numbers; the file suffix value `XXX` has to be set superior to 100. The standard installation of 4A/OP contains the infrared surface emissivity spectrum deduced from Snyder et al. [Ref. 9] and presented in Table 3. Fourteen emissivity classes are defined, associated to the IGBP surface type classification: `XXX` is between 101 and 114.

You can add in the **`datemis`** directory your own-defined surface emissivity files, respecting the ASCII format of one of the two possible types:

- for new constant emissivities: you can copy one of the constant emissivity file in the **`datemis`** directory and change its second column to the constant emissivity value. Be sure the name of the new file is `spemisXXX.dat` where `XXX` is the new emissivity (i.e. 057 for a value of 0.57 for example). The corresponding emissivity suffix `XXX` should be in the range **000 – 100**;
- 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 **`datemis`** directory) and the corresponding emissivity suffixes XXX are **superior to 114**.

For a complete description of the surface emissivity files, please refer to the reference documentation (Section 5.2.2).

IGBP surface properties classification	Season	Snyder emissivity classes	EmissivityCode
Water (17), Wetlands (11)	all	Water	101
Snow, Ice (15)	all	Snow, Ice	102
Barren/Desert (16)	all	Arid bare soil	103
Crops (12), Tundra (18)	all	Organic bare soil	104
Open shrubs (7)	F,W	Senescent sparse shrubs	105
Open shrubs (7)	Sp,Su	Green sparse shrubs	106
Savannah (9) ; Closed Shrubs, Grasslands, Crop/Mosaic (6,10,14)	W,Sp F,W	Senescent grass savannah	107
Savannah (9) ; Closed Shrubs, Grasslands, Crop/Mosaic (6,10,14)	Su,F Sp,Su	Green Grass Savannah	108
Woody Savannah (8)	F,W	Senescent Woody Savannah	109
Woody Savannah (8)	Sp,Su	Green Woody Savannah	110
Deciduous broadleaf and Mixed forest (4,5)	F,W	Senescent Broadleaf Forest	111
Deciduous broadleaf and Mixed forest (4,5) ; Evergreen Broadleaf Forest (2)	Sp,Su all	Green Broadleaf Forest	112
Deciduous Needle Forest (3)	F,W	Senescent Needle Forest	113
Deciduous Needle Forest (3) ; Evergreen Needle Forest (1)	Sp,Su all	Green Needle Forest	114

Table 3: Emissivity class definition (adapted from Snyder et al., 1998).
Season notification: W – winter, Sp – spring, Su – summer, F – fall.

6.3. Instrument Spectral Response Function (ISRF)

The file describing the instrument functions used for a 4A/OP run are placed in the directory **`model/isrf`**. 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`**.

Three types of instrument 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`**), 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/spectrometer ISRF with a non constant sampling step is given by the file **`isrfairs0802.dsf`**

6.3.1. 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.

6.3.2. 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.

6.4. Aerosol/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`**. The nomenclature is aerosols_XXXX.dat, where `XXXX` is the identifier of the aerosol model/cloud type, from top to bottom layer (`XXXX` should be of maximum 6 characters length).

These files can also be either user-defined or default files can be used. Files for 12 aerosol models are included in the software package (Part I.[Ref. 10]).

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`. For a complete description of the aerosol/cloud physical parameters files, please refer to the reference documentation (Section 5.2.4).

7. Appendix : Simulation definition parameters

7.1. Global variables

Global variables are initialized through the run script file and are passed to the file makefile. They are listed in the table below. Please, also refer to section 5.2.5.1 in the Reference Documentation.

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
Radiative transfer computation						
JOB	Indicator of the program to use	string	-	makefile	JOB is fixed to 4a	4a
ATM	Indicator of the atmospheric database stored in the file atm4a$(ATM).ddb	char	-	makefile / directory datatm	(Presently, possible values are test, test2, satigr_v4.0_moyclas: TIGR atmosphere type)	test
ATMPROF	Index of the atmospheric profile read in the atmospheric database stored in the file named datatm/atm4a$(ATM).ddb. The range of profile indexes and their meaning depend on the atmospheric database.	int	-	makefile	Presently, possible values are 1 to 5. In the case of the total TIGR databank, not available with the default 4A/OP installation, 1≤ATMPROF≤ 1761 1 to 322: tropical class 323 to 710: warm temperate 711 to 1064: cold temperate 1065 to 1761: arctic class	0001
RSTR	Indicator of the parameter file para4a$(RSTR).dtp	char	-	makefile / directory input	(Presently, possible values are test1 to test14)	NA
RSCA	Indicator of the aerosol parameter file parascat$(RSCA).dtp	char	-	makefile / directory input	if empty string: no aerosol is considered	NA
RESOL	Indicator of the resolution of atlases described in the file atl$(RESOL)index.dsf	char	-	makefile / directory datatl	(Presently, possible values are hr09, that corresponds to the high resolution atlas database using spectroscopic parameters from the GEISA 2009 edition)	hr09
INS	Indicator (name) of the simulated instrument (instrument function ISRF) stored in the file named isrf(INS)(CASE).ddb	string	-	makefile / directory isrf	Only used if the convolution is performed (presently, possible values are inf, test, test2, iasi1c)	inf
CASE	Indicator of the instrument function version (case) for a given instrument defined by the INS	char	-	makefile / directory isrf	(Presently, possible values are " ")	" "

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
Reading the output of the radiative transfer and convert it into ASCII format						
NUMIN	Beginning of the extraction for the conversion (in wave number)	r	cm^{-1}	makefile	NUMIN must be ≥ *forig* (see Table 5)	635
NUMAX	End of the extraction for the conversion (in wave number)	r	cm^{-1}	makefile	NUMAX must be ≤ *fstop* (see Table 5)	2880
UNIT	Jacobian unit index for Jacobian ASCII outputs	i	-	makefile	• UNIT = 0: NedT at TREF - Kelvin per Kelvin.layer - Kelvin per layer for 1 g/g variation of the mixing ratio ro - Kelvin for 1% variation of the surface emissivity • UNIT = 1: NedT at TREF - Kelvin per Kelvin.Km - Kelvin per Km for 10% variation of ro - Kelvin for 1% variation of the surface emissivity • UNIT = 2: Decibel - Decibel per Kelvin.Km - Decibel per Km for 10% variation of ro - Decibel for 1% variation of the surface emissivity • UNIT = 3: NedT at brightness temperature Tb - Kelvin per Kelvin.Km - Kelvin per Km for 10% variation of ro - Kelvin for 1% variation of the surface emissivity • UNIT = 4: NedT at Tb - Kelvin per Kelvin.layer - Kelvin per layer for 1 g/g variation of ro - Kelvin for 1% variation of the surface emissivity • UNIT = 5: Initial unit - $W/(m^2\ str\ cm^{-1})$ per Kelvin.layer - $W/(m^2\ str\ cm^{-1})$ per layer for 1 g/g variation of ro - $W/(m^2\ str\ cm^{-1})$ for 100% variation of the surface emissivity	4
TREF	Reference temperature for conversion into NedT	r	K	makefile		280

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
TEBR	Radiometric noise estimation for conversion into Decibel	r	K	makefile		0.28
WCLASS	Width of the class in wave number for statistical characteristics of the difference between two spectra	d	cm^{-1}	makefile		50
WNMIN	Limits of the computation of the difference between two spectra in wave numbers	d	cm^{-1}	makefile		645
WNMAX		d	cm^{-1}	makefile		2760

Table 4: Global variables initialised through the file makefile

7.2. 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`. The table below (Table 5) describes every parameter. Please, also refer to section 5.2.5.2 in the Reference Documentation.

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
AAAA	Main run parameter that identifies the program to use	string (len=60 max)	-	Ini: parameter file para4a.dtp	Only for information: not used	AAAA
GEOMID	Geometry type identifier	string (len=6)	-	Ini: para4a.dtp	LIMB or EVIEW	EVIEW
TRAJET	Viewing configuration. The various configurations are defined in Table 1 and illustrated by Figure 2 to Figure 5	string (len=6 max)	-	Ini: parameter file para4a.dtp	UP, DOWNUP, DOWN, UPDOWN UP: nadir viewing DOWN: zenith viewing	UP
PZUP	TOA or cloud bottom pressure if GEOMID = 'EVIEW' TOA or cloud bottom height if GEOMID = 'LIMB'	r	hPa km	Ini: para4a.dtp		0.05
PZDOWN	Surface or cloud top pressure if GEOMID = 'EVIEW' Surface or cloud top height if GEOMID = 'LIMB'	r	hPa km	Ini: para4a.dtp	if "Earth view" and if *pzdown* undefined (negative): set *pzdown* to the surface pressure level read in the atmospheric database	1013.25
PZOBS	Pressure at observation level Height at observation level	r	hPa km	Ini: para4a.dtp	if geomid = 'EVIEW': pzup≤*pzobs*≤pzdown if geomid = 'LIMB': pzdown≤*pzobs*≤pzup	0.05
EMUP	Index for the	r	-	Ini: parameter	Used if *TRAJET*≠UP	100

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
	emissivity of the top level read in the file datemis/ spemis$EMUP.dat			file para4a.dtp		
TUP	Temperature of the top level	r	K	Ini: parameter file para4a.dtp	4 K in general (cold space temperature) If equal to 0, set to temperature of top level in atmospheric database.	4.
EMDOWN	Index for the emissivity of the lower level read in the file datemis/ spemis$EMDOWN.dat	r	-	Ini: parameter file para4a.dtp	Used if *TRAJET* ≠ DOWN	100
TDOWN	Surface temperature (temperature of the lower level)	r	K	Ini: parameter file para4a.dtp	If *TDOWN < 0* set to temperature of level2 If *TDOWN*=0: for level2 < nlevtam, *TDOWN* = temperature of level2 for level2 = nlevtam, *TDOWN* = surface temperature defined in the atmospheric database Else *TDOWN*=user input value	0.
ZGTAN	Geometric tangent height	r	km	Ini: para4a.dtp	Used if GEOMID= 'LIMB'	0.
TYP_ANG	Angle type definition for sectta1, sectta2 and secttasun (see below). This variable are also detailed in Table 1 and illustrated in Figure 2 to Figure 5	string (len=6 max)	-	Ini: parameter file para4a.dtp	ANGLE, SECANT or UNDEF ANGLE: secttax=1/cos(secttax), x = 1, 2 or sun SECANT: secttax=secttax UNDEF: sectta1=0, sectta2=1.7434468, secttasun=0	ANGLE
SECTTA1	Primary path angle (viewing angle) scaled to ground (altitude z=0).	r	degree	Ini: parameter file para4a.dtp	If *TYP_ANG*=SECANT, *SECTTA1* must be ≥1 If *TYP_ANG*=UNDEF, *SECTTA1* is set to 0 and then it is set to the value read in the atmosphere file datatm/ atm4a$(ATM).ddb	0. new: *sectta1* <1 or *sectta1* ≥90° admitted
	Or secant value	r	-			
SECTTA2	Secondary path angle (reflexion incidence angle) at z=0.	r	degree	Ini: parameter file para4a.dtp	Used if *TRAJET* = DOWNUP or UPDOWN	55.

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
	Or secant value	r	-		If *TYP_ANG*=SECANT, *SECTTA2* must be ≥1 If *TYP_ANG*=UNDEF, *SECTTA2* is set to 1.7434468 (cos(55°))	
SECTTASUN	Solar zenith angle	r	degree	Ini: parameter file para4a.dtp	Used if *TRAJET*= DOWN or DOWNUP If *TYP_ANG*=SECANT, *SECTTASUN* must be ≥1 If *TYP_ANG*=UNDEF, *SECTTASUN* is set to 0. To suppress the solar contribution, specify SECTTASUN outside [0,90[deg.	90.
	Or secant value	r	-			
ESDIST	Earth-Sun distance in astronomical units	r	astro. units	Ini: parameter file para4a.dtp	Natural values are [0.98, 1.02]	1
REFRA	Refraction computation identifier	string (len=6)	-	Ini: para4a.dtp	YES or NO	NO
ZATMBOT	Height of lower atmospheric level	r	km	Ini: para4a.dtp	if < 0: computed in atmsph through barometric height formula	0.
CONTIN	Continua (H_2O, N_2, O_2) contribution	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO If yes, the continua of the selected molecules (*IDSEL*=1) contribute to the radiative transfer computation	YES
RESPEC	Spectral resolution: contraction of the spectra before convolution or to define the spectral discretization with an "infinite" resolution simulation	r	cm^{-1}	Ini: parameter file para4a.dtp	If *RESPEC*<5.10^{-4}: - For *CONV*=YES, the resolution for the computation = 5.10^{-4} - For *CONV*=NO, the resolution for the computation = Atlas resolution Else *RESPEC* = user input value	0.00
FORIG	First wave number of the simulation	r	cm^{-1}	Ini: parameter file para4a.dtp	*FORIG* must be ≥ forig_atlas*	600.
FSTOP	Last wave number of the simulation	r	cm^{-1}	Ini: parameter file para4a.dtp	*FSTOP* must be ≤ fstop_atlas*	3000.
DELDS	Width of each Atlas spectral band	r	cm^{-1}	Ini: parameter file para4a.dtp	generally 15 cm^{-1}	15.
INTOPT	Indicator for interpolation in temperature of the optical thicknesses	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO Note: If no, the computation time is decreased but the results are deteriorated	YES
CONV	Indicator of convolution with the ISRF	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO ("infinite" resolution)	NO

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
HOLE	Indicator of hole algorithm use for the convolution (see Part III 1.4 in Reference Documentation Part I.[Ref. 10]	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO	NO
SHIFT	Indicator of a spectral shift introduction due to the ISRF	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO Only used if *CONV*=YES If yes, a spectral shift is applied due to ISRF not centred	NO
RNU	Value of the global spectral shift	r	cm^{-1}	Ini: parameter file para4a.dtp	Only used if *CONV*=YES Application of a possible global spectral shift of the spectra if *RNU* ≠ 0	0.
PASCON	Convolution step: step of the ISRF application	r	cm^{-1}	Ini: parameter file para4a.dtp	Only used if *CONV*=YES Must be consistent with the instrument function file header; *PASCON* only used if itypeconv=2 (see §6.3) it must be >*RESPEC*	0.
FPOID	Transmittance calculation flag	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO Jacobians and transmittances are not permitted if *CONV*=NO	NO
JACOB	Jacobian calculation flag	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or NO If *JACOB*=YES: - INTOPT must be YES - The Jacobian with respect to the temperature is always calculated. - The Jacobian with respect to the mixing ratio is calculated if the corresponding indicator is equal to 1 (see IDSEL and idjac below)	NO
IDSEL	Table to select molecules among the 42 gases available in the GEISA database and 3 CFCS coded between 61 to 63	int[63]	-	Ini: parameter file para4a.dtp	1 or 0 1=yes 0=no	0, …
ROCOEF	Table of weighting coefficients for the mixing ratio of the gases and CFCS	r[63]	-	Ini: parameter file para4a.dtp	0 ≤ *ROCOEF*	1.0, …
IDJAC	Table to select the calculation of the Jacobian for each gas in the 42 available gases and 3 CFCS	int[63]	-	Ini: parameter file para4a.dtp	1 or 0 (yes or no) Warning: The Jacobian calculation of a molecule is effective only if the corresponding molecule	0, …

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
					is selected too (idsel=1)	

Table 5: Run characterization parameters read in the file para4a.dtp

7.3. Directory description

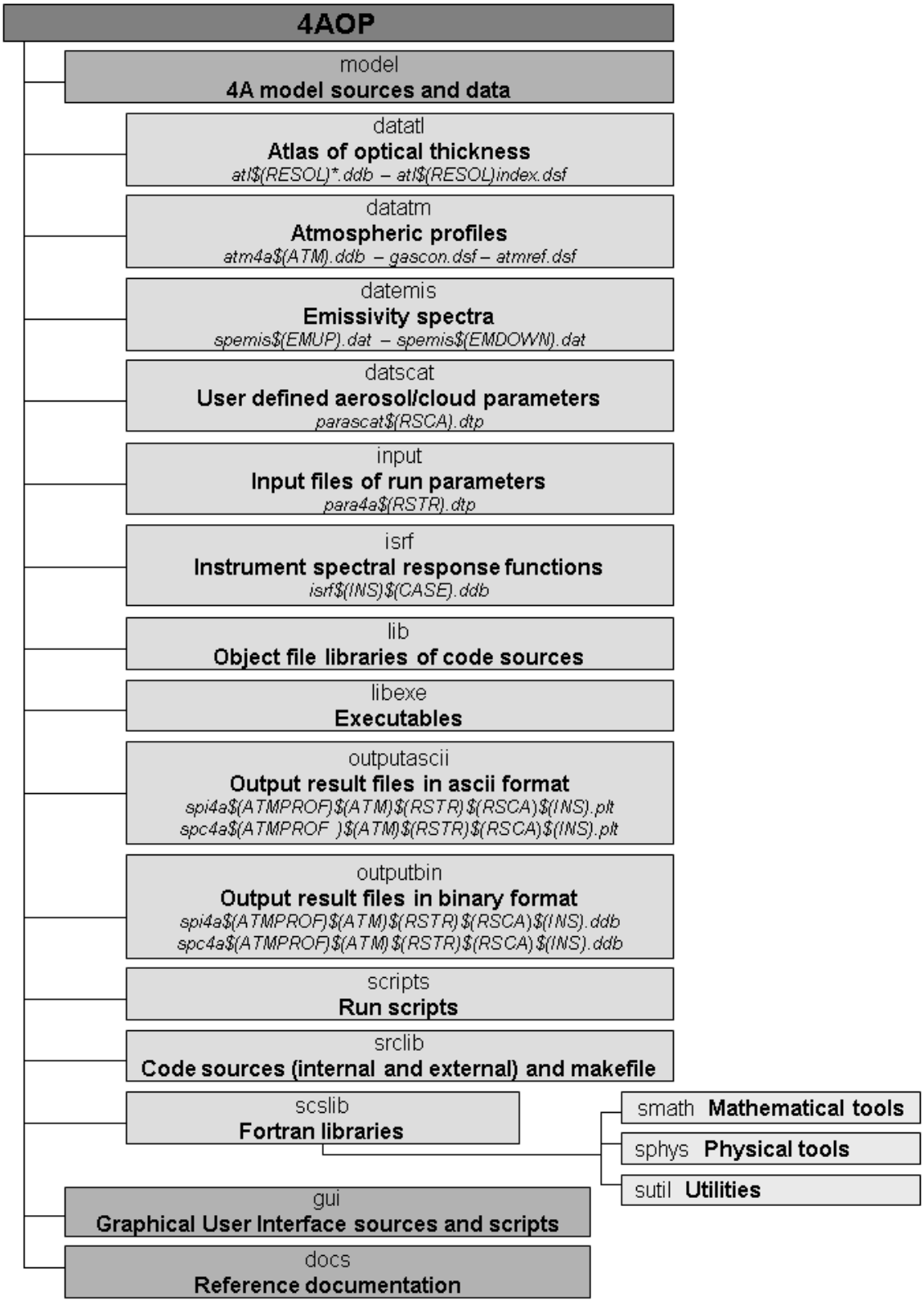

Diagram 1: Directory description of 4A/OP (* for several files).

8. References

[Ref. 1] 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.

[Ref. 2] Tournier, B., R. Armante, and N. A. Scott, 1995: STRANSAC-93 et 4A-93: Développement et validation des nouvelles version des codes de transfert radiatif pour application au projet IASI. Internal Rep. LMD, No. 201, LMD/CNRS, Ecole Polytechnique, PALAISEAU, France.

[Ref. 3] Scott, N. A., 1974: A direct method of computation of transmission function of an inhomogeneous gaseous medium: description of the method and influence of various factors. J. Quant. Spectrosc. Radiat. Transfer, 14, 691-707.

[Ref. 4] Jacquinet-Husson, N. *et al.*, 1999: The 1997 spectroscopic GEISA data bank. J. Quant. Spectrosc. Radiat. Transfer, 62, 205-254.

[Ref. 5] Jacquinet-Husson, N., N. A. Scott, A. Chedin, and A. A. Chursin, 2003: The GEISA spectroscopic database system revisited for IASI direct radiative transfer modelling. Atmospheric and Oceanic Optics, 16, N°3, 256-282.

[Ref. 6] Jacquinet-Husson, N. *et al.*, 2008: The GEISA spectroscopic database: Curent and future archive for Earth and planetary atmosphere studies. J. Quant. Spectrosc. Radiat. Transfer, 109, 1043-1059.

[Ref. 7] Jacquinet-Husson, N. *et al.*, 2011: The 2009 edition of the GEISA spectroscopic database. J. Quant. Spectrosc. Radiat. Transfer, 112, 2395-2445.

[Ref. 8] Chédin, A., *et al.*, 1985: The Improved Initialization Inversion Method: A High Resolution Physical Method for Temperature Retrievals from Satellites of the TIROS-N Series. J. Appl. Meteor., 24, 128-144.

[Ref. 9] Snyder, W.C., *et al.*, 1998: Classification-based emissivity for land surface temperature measurement from space. Int. J. Remote Sensing, vol. 19, 2753-2774.

[Ref. 10] L. Chaumat, C. Standfuss, B. Tournier, E. Bernard, R. Armante and N. A. Scott, 2012: 4A/OP Reference Documentation, NOV-3049-NT-1178-v4.3, NOVELTIS, LMD/CNRS, CNES, 310 pp.