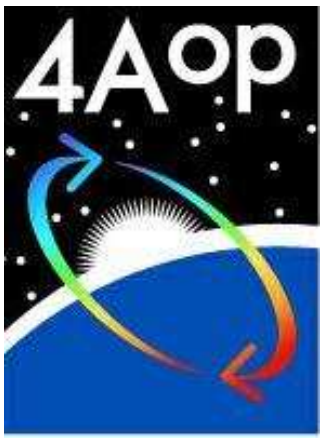


Installation and Quick Start Guide

Software Version 4AOP v1.5

© Software CNES – LMD (CNRS/ENS) - NOVELTIS

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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Table of Contents

1. INTRODUCTION 5

1.1. PURPOSE 5
1.2. HOW TO USE THIS DOCUMENT 6
1.3. PEOPLE AND AKNOWLEDGMENTS 6

2. INSTALLING 4A/OP 7

2.1. WHAT IS REQUIRED? 7
2.1.1. Required disk space 7
2.1.2. Compilers 7
2.1.3. Other components 7
2.1.4. Tcl/Tk 8
2.1.5. Gnuplot 8
2.1.6. NetCDF 8
2.1.7. Python 8
2.1.8. Supported computing environments 8
2.2. INSTALLATION PROCEDURE 8
2.2.1. Uncompress the .tar.gz file 8
2.2.2. Makefiles 9
2.2.3. Installation 9
2.3. WAS THE INSTALLATION SUCCESSFUL? 11
2.3.1. Installation check 11
2.3.2. Functional test cases 11
2.3.3. Graphical User Interface (GUI) 12

3. SCRIPT MODE RUNNING 13

3.1. PREPARING THE SIMULATION 13
3.1.1. Global variables 14
3.1.2. Required parameters 15
3.1.3. Parameters for scattering simulation 17
3.1.4. Viewing and geometric configuration definition 22
3.1.5. Work with 4A/OP in wavelength 27
3.2. RUNNING 4A/OP-SUBROUTINES 28
3.3. VIEWING 4A/OP-SUBROUTINE SIMULATION RESULTS 28

4. GRAPHICAL USER INTERFACE (GUI) 30

4.1. PRODUCE OR OPEN A GUI PARAMETER FILE 30
4.2. PRODUCE A SCRIPT FILE 30
4.3. USER'S CHOICES 30
4.4. SAVE 39
4.5. RUN 39
4.6. VISUALIZATION 40
4.6.1. Zoom 40
4.6.2. User specific plots 41
4.6.3. Examples 41
4.7. TEST 44
4.8. FUTURE IMPROVEMENTS 44
4.8.1. Run 44
4.8.2. Visualization 44

5. APPENDIX A: GENERAL DESCRIPTION 45

5.1. DIRECTORY DESCRIPTION 45

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

5.2. GLOBAL VARIABLES 46
5.3. RUN CHARACTERIZATION PARAMETERS 50
5.3.1. General simulation parameters 50
5.3.2. Scattering parameters 54
5.4. MAIN 4A/OP-SUB PROGRAM 57
5.5. 4A/OP-SUB INPUT PARAMETER SUBSTRUCTURES 57
5.6. MODIFYING THE 4A/OP-SUB STRUCTURE FIELDS 61

6. APPENDIX B: TEST CASES 63

7. APPENDIX C: SPECTROSCOPIC DATABASE GEISA 64

8. APPENDIX D: ON-LINE HELP OF THE GUI 65

8.1. OVERVIEW 65
8.2. GRAPHICAL USER INTERFACE DESCRIPTION 65
8.2.1. File menu 65
8.2.2. Run menu 65
8.2.3. Visualization menu 65
8.3. GUIDE 66
8.4. SPECIFIC HELP FOR SOME PARAMETERS 67
8.4.1. Viewing configuration 67
8.4.2. Atmospheric profile indexes 67
8.4.3. Surface type indexes 67
8.4.4. Handling the HDO isotope 69
8.5. RESOURCES 70
8.5.1. Disk usage 70
8.5.2. Processing time 70
8.6. BUG REPORTS 70

9. REFERENCES 71

10. ACRONYMS 73

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

1. INTRODUCTION

1.1. Purpose

The software 4A/OP subroutine, called 4A/OP-sub hereafter, is the subroutine version of the code 4A-SWIR-TIR, [DR10], [DR11], [DR12], [DR13] and [DR14].

The objective of the software is to allow the fast simulation of the radiative transfer in particular over the thermal and near 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: [DR15]). The atlases are created once and for all by using the line-by-line and layer-by-layer model, STRANSAC (Scott, 1974: [DR17]; [DR16]) in its latest 2012 version with up to date spectroscopy from the GEISA spectral line catalogue (Jacquinet-Husson *et al.*, 1999, 2003, 2008 2009, 2016: [DR5], [DR6], [DR7], [DR8], [DR9]). 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 and near-infrared region; the usual spectral domain is between 495 and 13290 cm^{-1} (20-0.76 µm). 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.

The 4A/OP-sub is an heritage of the 4A program for the needs of the inversion issues and was developed under funding by CNRS and from CNES programs. Further development is supported by CNES.

The main scientific functions included in 4A/OP-sub are listed below:

- User-defined spectral emissivity/reflectance/brdf functions;
- Spherical atmosphere;
- Solar spectral contribution;
- Limb viewing including refraction;
- Scattering for molecular, aerosol and cirrus contribution;
- Simulation in polarized light;
- Jacobian computation of temperature profile, atmospheric molecule profiles, surface pressure, emissivity and temperature and aerosol optical thickness;
- Convolution with instrument functions;
- The new possibility to define the input spectral limits and the instrumental functions in wavelength.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

A previous version of the 4A/OP-sub code has been delivered on March 2011 and corresponding to the 4A-SWIR program version described in ([DR11], [DR12]).

The 4A/OP code has been first released in 2015 in its 4A/OP-sub-2014-1.3.1 version and correspond to the new version of 4A/OP covering the infrared and near-infrared spectral region. The current version of 4A/OP is a new 2018 release named 4A/OP-1.4. The last update, version 1.2.1, was released in May 2018 after the previous one released in October 2017 and named 4A/OP-sub-2016-1.1.4 including the 4 patches from the 4A/OP-sub-2016-1.1 released in January 2017.

1.2. How to use this document

This document is intended for beginners with the 4A/OP-sub software: it provides a quick way inside the most common 4A/OP-sub features. For larger understanding of 4A/OP-sub functions, one should refer to the following documentation ([DR10], [DR11], [DR12], [DR13] and [DR14]).

The “Installation and Quickstart Guide” is intended to help new user to begin using 4A/OP-sub4A/OP-sub. It is divided into eight parts:

1. the installation guide (section 2),
2. the use of 4A/OP in script mode (section 3),
3. the use of the graphical interface (section 4),
4. a general description of the architecture (section 5),
5. some test cases and simulation examples (section 6),
6. a description of the spectroscopic database (section 7),
7. a presentation of the On line help GUI (section 8).

These different parts help the new user through several examples to explore the main 4A/OP-sub capabilities. For a more detailed use and scientific description of 4A/OP, please refer to the documents [DR1], [DR2] and [DR3].

1.3. People and aknowledgments

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

Authors from LMD:

Raymond ARMANTE

Noëlle A. SCOTT

Virginie CAPELLE

Alain CHEDIN

Authors from NOVELTIS:

Emilien BERNARD

Laure CHAUMAT

Carsten STANDFUSS

Bernard TOURNIER

Olivier LEZEAUX

Romain BERGOUGNOUX

Emmanuel DUFOUR

Sylvain BRUNATO

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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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. See also the file README in the user installation directory.

2.1. What is required?

The 4A/OP-sub distribution package is supplied in a single tar file 4AOP-1.5.tar.gz, where version corresponds to the current 4A/OP software version number (currently 4AOP-1.5). The file is compressed using gzip.

The following files are required for the installation process:

a. README (file containing important information);

b. 4AOP-1.5.tar.gz (4A/OP as a compressed tar file).

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

4AOP-1.5.tar.gz is a compressed tar file containing the entire 4A/OP-sub program. This includes 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-sub.

2.1.1. Required disk space

The 4A executables require about 5 Megabytes on disk only. The associated files (sources and user input files) require about 8 Gigabytes of disk space and test data also require 8 Gigabytes. In addition to this, atlas files require up to 40 gigabytes of disk space for all the TIR to UV-Vis atlas. This volume depends on the user simulation needs.

With all these components, the total amount of disk space required is about 55 GB.

The size of output files depends on the input simulation parameters, which you control via the files described in section 5.3 (Jacobian calculations can use lot of disk space).

2.1.2. 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-subroutine has also been tested on Linux using pgf90 (Portland Group Fortran 90 compiler), gfortran (GNU compiler), ifort (Intel compiler). The 4A/OP-subroutine source code is FORTRAN 90 compatible and contained within several *.f90 files.

Note: when compiling the code (see §2.2.3.1), the IDCOMPIL parameter allows the user to define and use his own environment (i.e. GNU, PGI or Intel compilers). Several makefiles are available (Makefile.4a.*IDCOMPIL*.inc).

Moreover, routines written in C have been added to the 4A/OP code and require a C compiler. 4A/OP has been tested on Linux using the gcc compiler (GNU compiler) and the icc compiler (Intel).

Important note: 4A/OP-sub has been developed and tested on a big-endian architecture (Sun). In particular, the optical thickness atlases are big-endian binary files. On little-endian architecture (i.e. x86 or x64), be sure your compiler has the option that swaps bytes at file reading.

2.1.3. Other components

The 4A/OP software includes four external tools which are used to compute the radiative transfer in case of scattering mode. These softwares are: SOS, DISORT (v2.1), SOS, LIDORT (v 3.5) and VLIDORT (v2.5). See document [DR3]. The compilation of these tools is included in the 4A compilation.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

2.1.4. Tcl/Tk

The 4A/OP GUI requires Tcl/Tk (at least version 8.4). See www.tcl.tk for more information on Tcl/Tk.

2.1.5. Gnuplot

The visualization part of the 4A/OP GUI needs Gnuplot to plot graphs. The GUI has been tested with the following version of Gnuplot: Unix version 4.6.2. See www.gnuplot.info for more information on Gnuplot.

2.1.6. NetCDF

The netCDF library OP (at least version 4.3.1) is required since the netCDF format is used by 4A/.

2.1.7. Python

Python 2.7.12 (or a newer version of Python 2) is required to launch the 4A/OP automatic tests. Python is not required for running 4A/OP itself.

2.1.8. Supported computing environments

Various computing environments have been tested and validated for the 4A/OP software installation.

The following environments are supported:

c. RH6: RedHat/CentOS 6.4 64bit:
d. Gcc 4.4.7 or icc 12.1.3
e. Gfortran 4.4.7 or ifort 12.1.3 (17.02)
f. Gnuplot 4.0
g. Tck/Tk 8.4
h. NetCDF 4.3.1
i. Python 2.7.12
j. RH7: RedHat/CentOS 7.2 64bit:
k. Gcc 4.9.3 or icc 12.1.3
l. Gfortran 4.9.3 or ifort 12.1.3 (17.02)
m. Gnuplot 4.0
n. Tck/Tk 8.4
o. NetCDF 4.4.1
p. Python 2.7.12

2.2. Installation procedure

2.2.1. Uncompress the .tar.gz file

Invoking the UNIX commands tar creates the 4A/OP-subroutine main directory 4AOP-1.5/ just below the current directory including all necessary subdirectories. Simply type the following command in order to uncompress and "untar" 4AOP-1.5.tar.gz:

```
tar -xzvf 4AOP-1.5.tar.gz
```

The 4AOP-1.5/ directory should now have been created and should contain the following files and subdirectories:

README, VERSION and the directories DDC, model/, gui/ and test.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

2.2.2. Makefiles

As stated in the chapter above the following compilers are supported:

- C compiler
  - GNU gcc (4.4.7 or 4.9.3)
  - Intel icc (12.1.3)
- Fortran compiler
  - GNU gfortran (4.4.7 or 4.9.3)
  - Intel ifort (12.1.3)

Depending on the compilers you want to use to build 4A/OP, you need to set the correct compilation parameters into a makefile that will then be used during the compilation process.

Some examples of makefiles are provided in the srclib sub-directory (Makefile.4a{IDCOMPIL}.inc files where {IDCOMPIL} is the configuration identifier of the compilation).

Here is an example of a simple makefile using gcc and gfortran:

```
# Makefile.4a

# GNU
CC      = gcc
FC      = gfortran

SHELL  = /bin/ksh
RANLIB = ranlib
AR = ar

INCLUDES = -I $(RDIR_S_MOD) -J $(RDIR_S_MOD)
CFLAGS   = -DSWAPIO -O3 -mieee-fp -ffloat-store -march=native
FFLAGS   = -O3 -fconvert=big-endian -mcmodel=large -cpp -ffloat-store -mieee-fp -march=native
LDFLAGS  = -O3 -fconvert=big-endian -mcmodel=large -cpp -ffloat-store -mieee-fp -march=native -Wl
-lnetcdf -lnetcdff
```

2.2.3. Installation

2.2.3.1. Compilation procedure

For the compilation, it is necessary to prepare the compilation configuration which depends on the compiler that will be used. In the sub-directory srclib of the directory model, there are some exemples of configuration definition in the files Makefile.4a{IDCOMPIL}.inc where {IDCOMPIL} is the configuration identifier of compilation. The user has to edit such a configuration file, adapt the configuration to its own computing environment and save this configuration. The information to be given are the fortran and C compilers paths and their options (CC, FC, CFLAGS, FFLAGS and LDFLAGS).

The working directory for the compilation is srclib/.

Thus, the compilation command is the following:

```
make all IDCOMPIL={your own parameters}
```

The compilation generates libraries, object files and executables. These compilation outputs are placed in the directory library/lib$IDCOMPIL and inside sub-directories libexe (executables), libobj (objects), libmod (modules) and lib (libraries). The subdirectory library/lib$IDCOMPIL/libexe/ should contain the executable files mainaaaa, atmformat, aerformat, lirespc4a, lirespi4a, calcStat, calcStatJ, ifctgauss, spjac and statanalys.

The compilation creates also directories used to store the results of the 4A simulation and the ISRF preparation: outputbin$IDCOMPIL, outputascii$IDCOMPIL and isrf$IDCOMPIL.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Example:

The user can compile with the configuration file coming with the distribution: IDCOMPIL=.gfortran, following:

```
make all IDCOMPIL=.gfortran
```

Important note: if the parameter IDCOMPIL is not set, the compilation procedure will use the configuration of the default file Makefile.4a.inc and the results of the compilation are put in the directory library/lib/ instead of library/lib$IDCOMPIL/. In this case, the compilation might stop with errors because the default configuration may not be compatible with the current system and compiler.

2.2.3.2. Atlas preparation

The atlases contain the pre-calculated optical thicknesses of atmospheric molecules for a given spectral domain. In this version of 4AOP-1.5, the atlases correspond to the NIR-SWIR-TIR spectral domain. If not already done (i.e. the directory model/datatl_geisa15 is empty), the user has to link atlas files into the system:

```
rmdir datatl_geisa15
ln –s {atlas path} datatl_geisa15
```

Note: the atlases are based on the last version of the spectroscopic database GEISA 2015 and covering TIR, SWIR and NIR spectral domain. The identification (via the global variable RESOL, see Table 7) of the atlases is:

RESOL=hr15

If an older version of the atlases has to be used, the two global variables RATL and RESOL (see Table 7) must be set accordingly.

2.2.3.3. Contents of the system installation

4AOP-1.5.tar.gz contains the entire 4A/OP-subroutine code (possible exception: large data base 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/ 4A/OP radiative transfer model. This includes 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/ 4A/OP graphical user interface. This includes Tcl/Tk sources and libraries, gnuplot scripts and on-line help for the GUI.

test/ Test scripts and reference test results.

The model directory should contain the following subdirectories:

datatl_geisa15/	Link to the Atlas directory (see above)
datatm/	Atmosphere data
datscat/	Scattering data
datsun/	Solar data
datsurf/	Surface data
input/	Input files
isrf/	ISRF files
library/	Compiled libraries
log/	Log files
outputascii/	ASCII output files
outputbin/	Binary output files

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

scripts/	launch scripts
scslib/	External library code
srclib/	Internal code

The model/library/lib$IDCOMPIL/libexe subdirectory should contain the following executable files: mainaaaa, atmformat, aerformat, lirespc4a, lirespi4a, statanalys, spjac, calcStat, calcStatJ, calcStatJsurf, calcStat_isrf, calcStat_statanalys and all the instrument functions executables corresponding to the isrf compiled and used: ifctgauss, ifctGauss, ifctiasi1c,…

2.3. Was the installation successful?

2.3.1. Installation check

In order to check that 4A/OP and its prerequisites are properly installed and that 4A/OP has everything it needs to run, use the test/check_installation.sh script.

First, go to the test directory:

```
cd {your 4A/OP installation path}/test
```

Then launch the script:

```
./check_installation.sh {ID_COMPIL} {OS}
```

Where {ID_COMPIL} is the compilation identifier that was used during the 4A/OP installation and {OS} is “RH6” or “RH7” (please choose the right OS value : RH6 for RedHat/CentOS 6 systems, RH7 for RedHat/CentOS 7 systems. If this parameter is not set, it defaults to RH7).

This script will check all 4A/OP prerequisites. Please make sure all prerequisites are properly installed before running 4A/OP.

This script will also launch some quick functional tests in order to check that 4A/OP is working.

2.3.2. Functional test cases

In order to check that 4A/OP is working in a more accurate way, various test cases (and the associated results) are provided with 4A/OP (see the 4A/OP test reference for more information about these tests).

A Python script is provided to handle the execution of all tests and the verifications of test results.

NB: Python 2.7.12 (or a newer version of Python 2) is required to run this script

First, go to the test directory:

```
cd {your 4A/OP installation path}/test
```

Many options are available, please use the following command to list them :

```
./launch_tests.py --help
```

Then launch the script, for example, to launch all tests, use the following command:

```
./launch_tests.py –i {ID_COMPIL}
```

where {ID_COMPIL} is the compilation identifier that was used during the 4A/OP installation.

The launch_tests.py script will then launch all the specified tests and compare the test results with the reference results. Please see the script output for more details.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

The following command compares the tests results with the references without relaunching the tests

```
./launch_tests.py –i {ID_COMPIL} --check_only
```

CAUTION: Executing all tests can take a considerable amount of time (several hours depending on the computing power available).

2.3.3. Graphical User Interface (GUI)

In order to test the good installation of the GUI the user could run several tests cases coming with the software. Currently, the four test cases are the same as the test cases executed in the software installation step (see section 2.2 and Appendix B: Test cases for the description of the test cases): the tests 1, 14, 15 and 16 have been implemented for the GUI. The input file for the GUI are in the gui/src/ directory:

test1.4a, castest14.4a, castest15.4a and castest16.4a.

To launch the GUI, the user has to move under the gui path and has to launch the executable:

```
cd 4AOP-1.5/gui
./src/4Agui
```

See section 4 for more details on the GUI.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

3. Script mode running

3.1. Preparing the simulation

After 4A/OP-sub has been successfully installed, the processing can be executed either by using the main script main_aaaa provided in the 4A/OP-sub package for a stand-alone use, either by using your own main script and program in the case of a user specific application.

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

q. setting up some global variables and simulation definition parameters;

r. providing 4A/OP-sub with some input files (atmospheric profiles, surface properties and instrumental function).

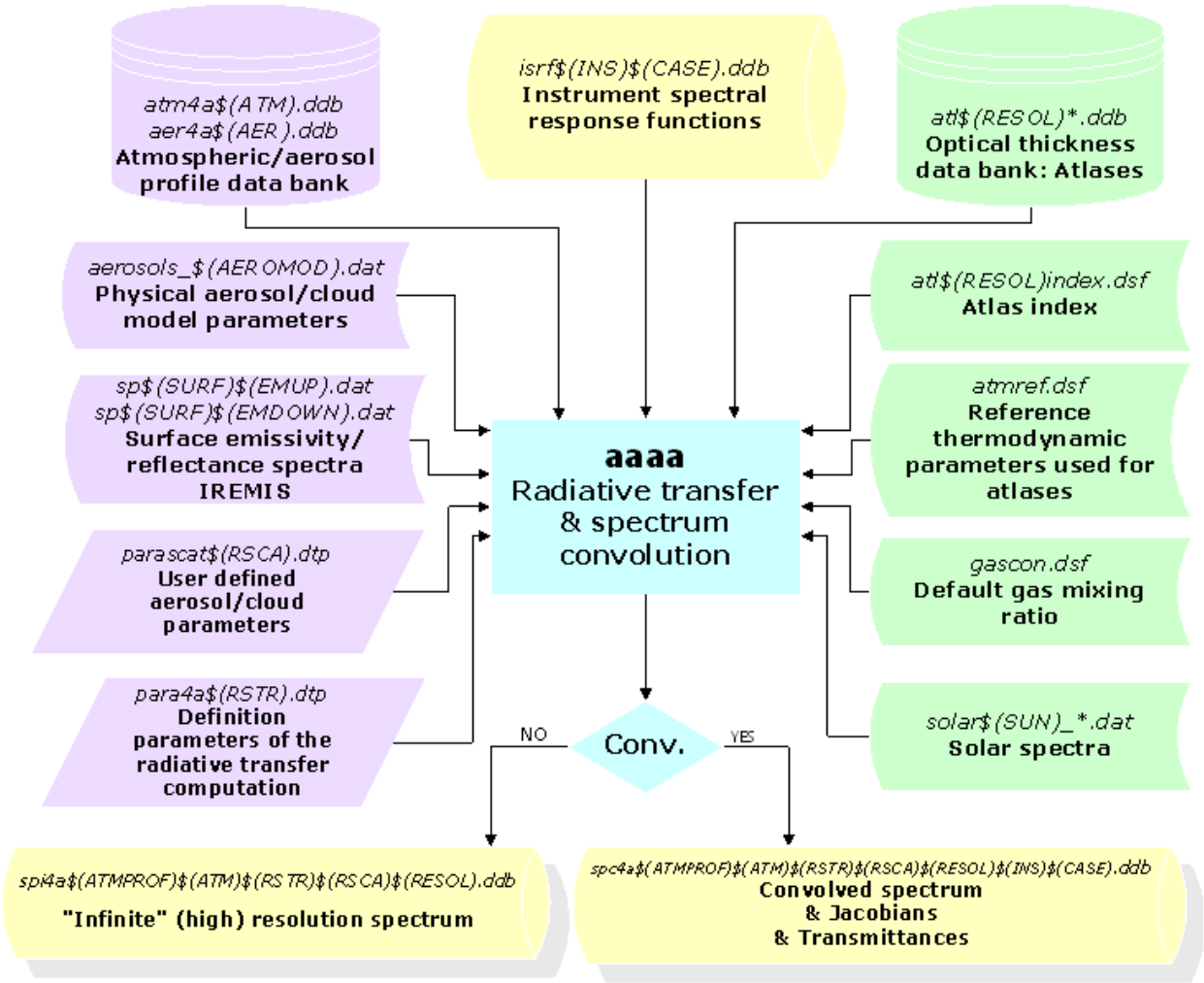


Figure 1: Software flowchart (inputs, outputs & processes).

Six user input files are necessary to completely specify all 4A/OP-sub input parameters (document [DR1]). These files are read by the 4A/OP-sub main program mainaaaa through several subroutines which set up all input parameters required for the simulation definition.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

s. One user interface file has a predefined filename: para4a$(RSTR).dtp. It is located in the subdirectory input. It is identified by the global variable RSTR. This ASCII file has to be edited manually 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 Table 8 and document [DR1]).

t. Another user interface file is needed if aerosol contribution is required; its predefined filename is parascat$(RSCA).dtp. It is located in the subdirectory input. It is identified by the global variable RSCA. This ASCII file has to be edited manually in order to set several parameters, to specify the aerosol model, the number of aerosol layers, etc. (see document [DR1]).

u. Four other user input files are necessary:

- One deals with the atmospheric state (profiles) to be simulated (see document [DR1]), located in the subdirectory datatm. It is identified by the global variable ATM;
- One with the spectral surface emissivity / reflectance (see document [DR1]) located in the subdirectory datsurf. It is identified by the two variables SURF and EMDOWN which are specified in the para4a$(RSTR).dtp parameter file;
- One with the aerosol physical parameters (see document [DR1]) located in the subdirectory datscat. It is identified by the global variable AER;
- And one with the instrument function (see document [DR1]) for an instrument simulation case, located in the subdirectory isrf. It is identified by the global variable ISRF.

In this section, input files are already provided and we concentrate on setting up the global variables and simulation definition parameters. See (see document [DR1]) 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).

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 test case n°15 of the run shell script CasTest_subroutine located in model/scripts directory in your user directory. Open this file. Go to the test case n°15. The following lines in the script represent the global variables needed by 4A/OP-sub:

```
set INS=gauss
set CASE=mini
set ATM=satigr_v4.0_moyclas
set ATMPROF=0002
set AER=
set AERPROF=
set RSTR=castest15
set RSCA=
set SUN="_planck"
set NUMIN=4800
set NUMAX=4900
set UNIT=4
set RATL  = '../datatl_geisa15'
set RESOL = 'hr15'
```

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$(ATM).ddb.The range of profile indexes and their meaning depend on the atmospheric database;
- **AER** is the indicator of the atmospheric aerosol database stored in the file aer4a$(AER).ddb (in model/datatm);
- **AERPROF** is the index of the atmospheric aerosol profile read in the atmospheric database stored in the file named aer4a$(AER).ddb.The range of profile indexes and their meaning depend on the atmospheric database;
- **RSTR** is the indicator of the parameter file para4a$(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 7 of appendix).
- **EXP** is the indicator to make the interpolation of exponential from tabulated exponential binary files or compute the exact exponential (see the meaning of the different indexes in Table 7 of appendix).
- **SUN** is the indicator to read the solar spectra (see the meaning of the different indexes in Table 7 of appendix)
- **RATL** is the path for atlas
- **RESOL** is the indicator for the atlas version (could be hr09 for version 2009, hr15 for version 2015)
- **IDCOMPIL** is the indicator of the used compiler. It is passed as an argument to the script
- **VERBOSE** is the indicator for the verbosity of 4A: if set (VERBOSE=1) it provides in the log file additional details as to what 4A/OP is doing.

The complete list of global variables can be found in Table 7 of the appendix.

3.1.2. Required 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 7 and Table 8 of the appendix.

Important note: Since the version 4AOP 1.1, both parameters PASCON and CONV have been removed from the para4a.dtp parameter definition file. The first one, PASCON, is now in the ISRF preparation process: you can define it in the isrfINSCASE.dsf ascii definition file or in the instrument dedicated subroutine ifct file (ifct$INS.f90). See the section 3.2.4 in the [DR1].

Such a file is displayed below (**para4_example.dtp** in your **input** directory):

```
'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'    1013.00
' 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 Emis./Reflect. of the lower level  EMDOWN'     'snyder' 1
' Temperature of the lower level                  TDOWN'     0.0
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

```
' Geometric tangent height (km) (LIMB only)        ZGTAN'       0.
' Viewing "zenith" angle (deg)                     TETA1'       1.
' 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'       0.
' 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) (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'     'NO'
' Spectral resolution                         RESPEC'     0.05
' First wave number of the simulation             FORIG'     3300.
' Last wave number of the simulation              FSTOP'     3315.
' 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  ,0  ,1  ,0  ,0  ,0  ,0  ,
'   nh3  ph3 hno3   oh   hf  hcl  hbr   hi  clo  ocs'
    1   ,0  ,0  ,0  ,0  ,0  ,0  ,0  ,0  ,0  ,
'  h2co c2h6 ch3d c2h2 c2h4 geh4  hcn c3h8 c2n2 c4h2'
    0   ,0  ,1  ,1  ,0  ,0  ,1  ,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  ,
' 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'
    0   ,1  ,0  ,0  ,0  ,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  ,
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

```
' cfc11 cfc12 ccl4                              '
  0   , 0   , 0   ,
```

3.1.3. Parameters for scattering simulation

The scattering can be taken into account in the 4A/OP-sub code by using different scattering code available that has to be chosen by the user among Disort, SOS, Lidort and Vlidort codes. The selection is done threw the parameter FLAG_SCAT in the input scattering parameter file **parascat$RSCA.dtp**.

The variable $RSCA initialized through the makefile identifies the parameter file **parascat$RSCA.dtp** that contains the parameters for Rayleigh and aerosol/ice scattering. The file (stored in the directory **input**) contains optical user parameterisations. Specifying $RSCA to the empty string, 4A/OP-sub is executed in the same way as the reference code (without scattering) by reading the file **parascat.dtp**, delivered with the software package.

An example of parametrization (**parascat$RSCA.dtp**) of scattering by aerosols is displayed below.

```
' Disort(DIS),SOS(SOS),Lidort(LID),Vlidort(VLI) or no scat(NO)   FLAG_SCAT'       LID
' 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
```

With the associated aerosol profile file (**aer4a$AER.dtp**) relative to a cirrus cloud :

```
1 0.766284
247.870  275.950  5.00-01 cir100  0.51  0.875                  # cirrus
```

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 **4A/OP-sub with aerosol scattering**.

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 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 affect 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.

In case of run without scattering simulation, RESPEC parameter is used for the contraction of the spectrum after radiative transfer computation and before convolution by an instrument function and/or output files writing.

If RESPEC is set to 0, no contraction is performed before radiative transfer computation. Radiative transfer computation (with or without scattering) is performed on spectral domain grid defined according to spectral domain sampling, molecule selection and solar spectrum sampling. A contraction is performed after radiative transfer computation and before convolution if HOLE convolution is not activated. The following table describes the

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

computation sampling according to RESPEC parameter and HOLE convolution activation, with or without scattering simulation:

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

4AOP	Without scattering (calcrad)		RT: Atlas irregular grid	Pseudo-infinite	RESPEC = 0		Spectrum at atlas resolution
					RESPEC > 0	Resampling at RESPEC (contract) if RESPEC < 5e10-4 -> RESPEC = 5e10-4	Spectrum at RESPEC sampling
				Convolution	HOLE	HOLE convolution (cnvspehole)	Spectrum at ISRF sampling
					without HOLE	Resampling at RESPEC (contract) if RESPEC < 5e10-4 -> RESPEC = 5e10-4	Convolution at RESPEC (cnvspeos, cnvspe)	Spectrum at ISRF sampling
	With scattering (DISORT, LIDORT, VLIDORT, SOS)	RESPEC = 0	RT: Atlas irregular grid	Pseudo-infinite			Spectrum at atlas resolution
				Convolution	HOLE	HOLE convolution (cnvspehole)	Spectrum at ISRF sampling
					without HOLE RESPEC = 5e10-4	Resampling at RESPEC (contract_diff)	Convolution at RESPEC (cnvspeos, cnvspe)	Spectrum at ISRF sampling
		RESPEC ≥ 5e10-4 Resampling at RESPEC (contract_sig)	RT: RESPEC grid	Pseudo-infinite			Spectrum at RESPEC sampling
				Convolution	HOLE	HOLE convolution (cnvspehole)	Spectrum at ISRF sampling
					without HOLE	Convolution at RESPEC (cnvspeos, cnvspe)	Spectrum at ISRF sampling

Table 1: Description of the different calculation paths and the corresponding computation sampling for the RT simulation.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

3.1.3.1. Polarization parameters

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'       LID
' 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
```

3.1.3.2. Scattering acceleration mode: SCATFAST mode

Since the 4A/OP 2016.1.0 version, a new mode is available for the scattering simulation. The running time in the scattering simulation can be important (> 1h and until dozens of hours) depending on the parameterization. The possibility to accelerate the calculation is given to the user by controlling few parameters in the input file. The method used to accelerate the scattering calculation is based on the k-distribution and in particular on the LSI method developed by O'Dell, 2010 [DR20], which has been made generic to adapt to 4A.

Briefly, the LSI method is based on three types of calculation: a low precision calculation processed on all the spectral samples, and a high and low precision calculation processed on the « bins ». The « bins » are upstream processed by the « double-k » classification.

Since the 4A/OP 1.2 version, the compatibility of the SCATFAST mode with VLIDORT is ensured. Therefore, the user can set FLAG_FAST to 1 and FLAG_POL to 2. The FLAG_POL=1 has not yet been tested.

The method has been validated with LIDORT and VLIDORT software.

Below, the instruction to use this method in 4A/OP is given.

The parameterization of the SCATFAST mode is controlled by three user's input parameters:

- **RESPEC**: used to accelerate the low precision calculation: in the *para4a$RSTR.dtp file*;
- **FLAG_FAST**: used to activate the acceleration mode: **0=NO, 1=YES**: in the *parascat$RSCA.dtp file*;
- **LEVEL_FAST**: used to control the level of the acceleration for the high precision calculation: values are between 1 and 6: **1=accurate, 6=fast** (this is a way to choose the good balance between speed and accuracy): in the *parascat$RSCA.dtp file* ;

The LSI method is validated only for the total scattering mode. Therefore, when the SCATFAST mode is activated, the flag **SCAT_MODE** is forced to zero (0 = total scattering), if differently set by the user in the input.

Note for number of stream/moments:

Recently, a modification has been made for the SCATFAST mode. The number of stream and moments for the high and low precision modes are let to the choice of the user and can be controlled in the `parascat$RSTR.dtp` file as it is the case for the classic mode without acceleration. These parameters are N_STREAM, N_MOM, N_STREAM_LOW and N_MOM_LOW.

In the first implementation of the method, the number of stream and moments was forced in the code and not available to the user. However, some problems can occur if the number of stream is too high (36 stream) with some specific aerosol models.

For no or low experienced users with this method, we give some standard values for the number of stream/nmom in the **Table** 2:

OPAC MODEL (DIRECTORY DATSCAT/)	*SCALAR MODE*				*POLARIZATION MODE*			
	N_STREAM	**N_MOM**	**N_STREAM_LOW**	**N_MOM_LOW**	**N_STREAM**	**N_MOM**	**N_STREAM_LOW**	**N_MOM_LOW**
waso70, inso00, ssam,…	16	32	2	4	36	72	2	4
mitr00	16	32	2	4	20	82	2	4

Table 2: Standard values for n_stream and n_mom parameters in the HIGH and LOW stream calculation for SCATFAST mode.

Examples

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

```
' Disort(DIS),SOS(SOS),Lidort(LID),Vlidort(VLI) or no scat(NO)   FLAG_SCAT'       LID
' 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'    1
' Speed level of FAST-SCATTERING (1->6 default:inf=2/conv=5)     LEVEL_FAST'    2
' 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
```

Some examples of using the SCATFAST scattering mode is given in the test cases coming with the distribution (test/scripts/swir_nir/ folder): the folders with the "lsi" extension contains the scripts that run the SCATFAST simulations for the MicroCarb widened bands in the presence of aerosol.

The SCATFAST scattering simulations with **polarization** are identified with the "vlidort_lsi_pola" extension. For example, the "test_gauss_scat_vlidort_lsi_pola_b1" folder contains the script that run the simulation of the radiative transfer in the first MicroCarb widened band, with the Gauss instrument, the SCATFAST scattering mode and with polarization.

3.1.3.3. Choice of output components (VLIDORT)

In the scattering mode when selecting VLIDORT (only) as the scattering code, it is possible to define an output component by combining the parameters RAD_COMP and SCAT_MODE. The RAD_COMP parameter allows to choose between a direct component (RAD_COMP=1), a diffuse component (RAD_COMP=2) or a total (direct + diffuse) component (RAD_COMP=0).

Moreover, the SCAT_MODE parameter allows defining more precisely the diffuse component. One can choose to take into account only single scattering (SCAT_MODE=1), only multiple scattering (SCAT_MODE=1), or both

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

(SCAT_MODE=0). The SCAT_MODE parameter does not affect which component are in the output when the output does not contain the diffuse component (when RAD_COMP=1).

These parameters affect the output radiances and jacobiens. The transmittances are not generated by VLIDORT and when generated by 4A in the scattering mode they correspond only to the direct component (forward scattering).

VLIDORT does not generate all components in all possible configurations. The direct component in the 'DOWN' and the 'UPDOWN' configurations is not calculated. In such case, it is directly 4A that will calculate the necessary fields and contribute to the output.

All possible configurations and the corresponding parameters are presented in the Table 3.

SCAT_MODE / RAD_COMP	TOTAL (0)	SINGLE (1)	MILTIPLE (2)
TOTAL (0)	down: dir (4A) + SS + MS up: dir (vli) + SS + MS	down: dir (4A) + SS up: *currently not possible*	down: dir (4A) + MS up: *currently not possible*
DIRECT (1)	down: dir (4A) up: dir (vli)	down: dir (4A) up: *currently not possible*	down: dir (4A) up: *currently not possible*
DIFFUSE (2)	down: SS + MS up: SS + MS	down: SS up: SS	down: MS up: MS

Table 3: All possible SCAT_MODE_RAD_COMP combinations and the corresponding output if selecting VLIDORT as the chosen scattering code. The referred path refers to the primary one: the mark '*DOWN*' includes the DOWN and the UPDOWN paths, and the mark '*UP*' includes the UP and the DOWNUP paths. SS refers to the single scattering component (calculated by VLIDORT), MS to the multiple scattering component (calculated by VLIDORT), and DIR to the direct component. The direct component could be calculated by VLIDORT (marked with *vli*) or 4A (via calcrad; marked with *4A*). Certain combinations in UP paths are not currently possible.

Note that the calculation of the **direct** component in the scattering mode via 4A (calcrad) takes into account diffusers (aerosols and air molecules) contrary to the same calculation in the mode without scattering (FLAG_SCAT=NO). The difference is also that 4A (calcrad) in the scattering mode does not take into account emissive/reflective bodies above the surface (like clouds) defined by parameters EMUP & TUP (para4a). This is consistent with VLIDORT that currently does not have possibility to take into account such bodies.

3.1.4. 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 Figure 2 to Figure 10.

VIEW PATH	*TETA1*	*TETA2*	*TETAS*
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 teta1 (limb view through tangent point)	-

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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

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

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 $\theta(z)$ 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 night-time 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 night-time 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.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

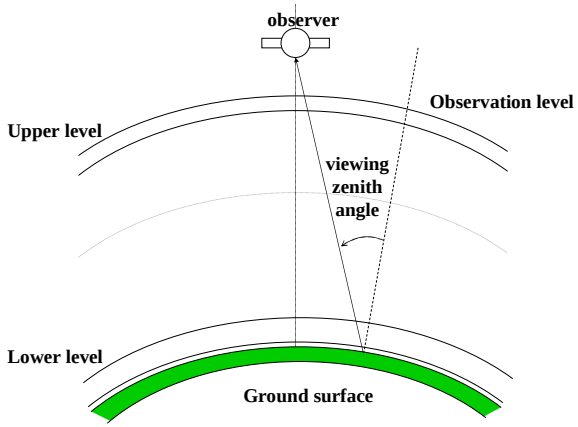


Figure 2: Nadir viewing (*TRAJET* = UP)

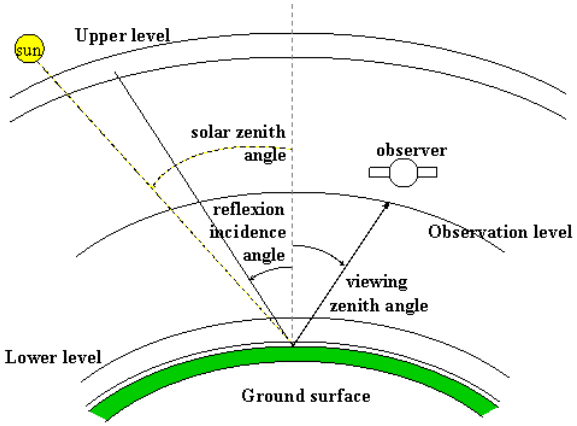


Figure 3: Viewing with ground surface reflexion (*TRAJET* = DOWNUP)

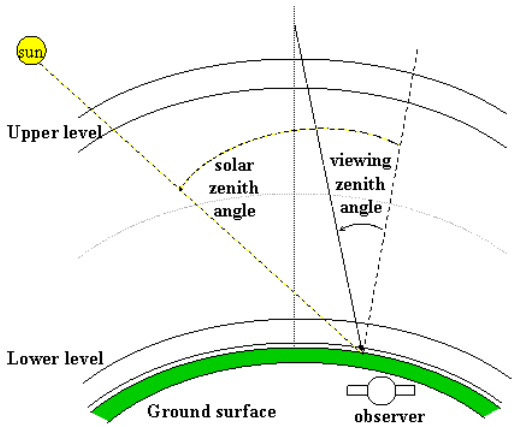


Figure 4: Zenith viewing (*TRAJET* = DOWN)

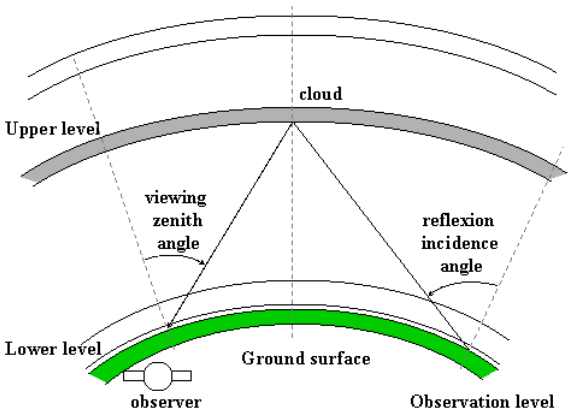


Figure 5: Viewing with cloud reflexion (*TRAJET* = UPDOWN)

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

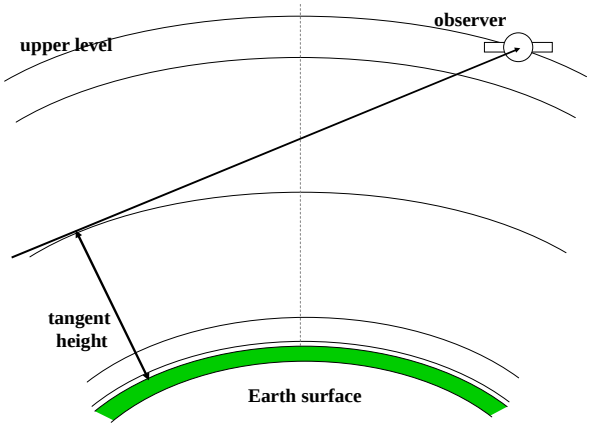


Figure 6: Limb view through the tangent point (*TRAJET* = DOWNUP). The user specifies background temperature and emissivity (cloud or sky), tangent height and the lower level below tangent height. The observer must not be specified above the background level.

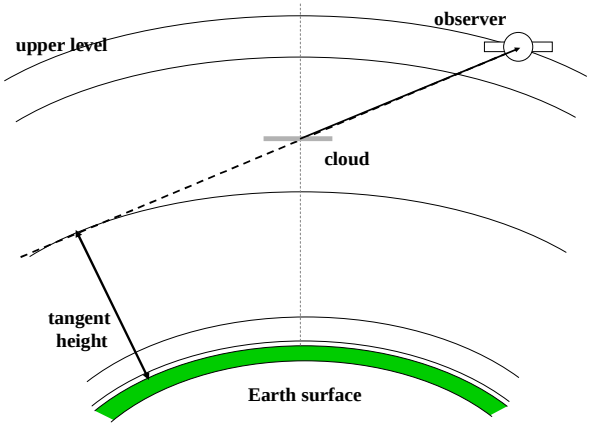


Figure 8: Limb view of a cloud top (*TRAJET* = UP). The user specifies cloud top temperature and emissivity, tangent height, and the lower level above tangent height.

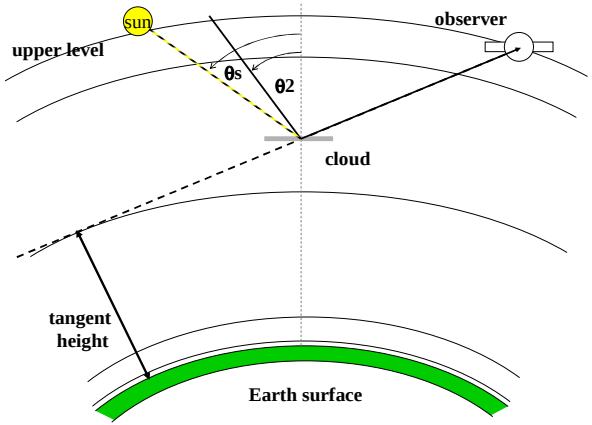


Figure 7: Limb view of a cloud top, including reflection of downward terrestrial and solar radiation (*TRAJET* = DOWNUP). The user specifies background temperature and emissivity (sky or cloud), cloud top temperature and emissivity, tangent height, the lower level above tangent height, and the solar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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

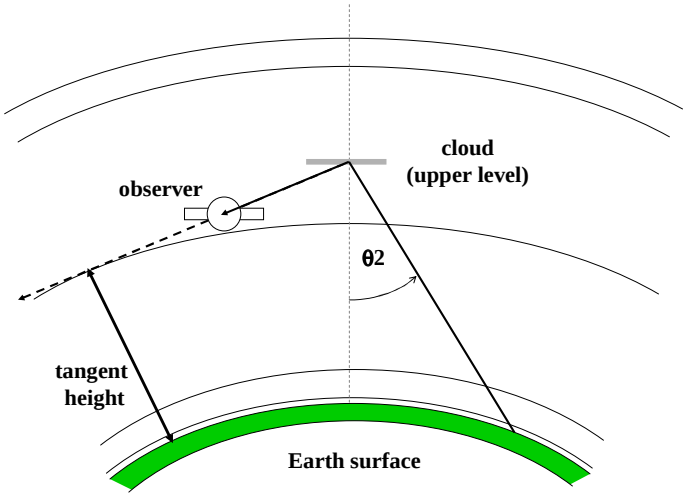


Figure 9: Backward limb view of a cloud bottom, including reflection of upward terrestrial radiation (*TRAJET* = UPDOWN). The user specifies background temperature and emissivity (Earth surface or lower cloud), cloud bottom temperature and emissivity, tangent height, and the upper level above tangent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

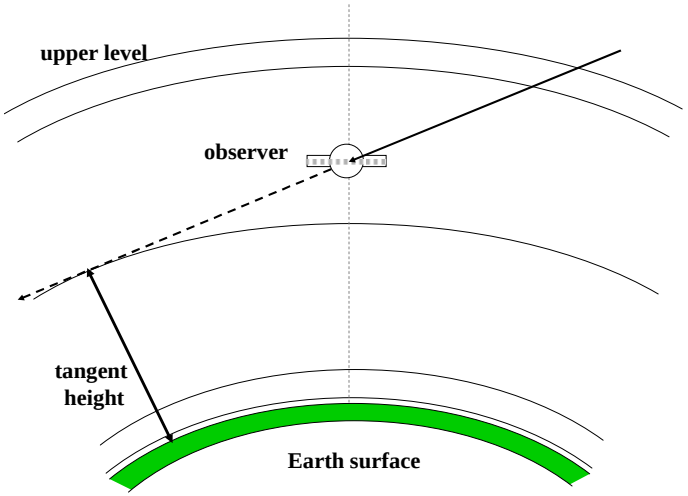


Figure 10: Backward limb view of sky or a cloud bottom (*TRAJET* = DOWN). The user specifies background temperature and emissivity, tangent height, and the observation level above tangent height.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

3.1.5. Work with 4A/OP in wavelength

Since the 4A/OP 1.2 version, the possibility is given to define the spectral parameters and the ISRF function in wavelength. Internally, 4A/OP continues to work in wavenumber. It is only the input parameters which can be now defined in wavenumber or wavelength and also the ISRF functions. The choice between both scales (wavenumber/wavelength) is controlled by the flag FLAG_LAMBDA.

The flag has 2 possible values:

- FLAG_LAMBDA = 0: 4A will read spectral parameters in wavenumber with the cm^{-1} unit;
- FLAG_LAMBDA = 1: 4A will read spectral parameters in wavelength with the micron (µm) unit;

Below, some explanation for the parameters in the both cases high resolution simulation and convolution simulation.

3.1.5.1. The basic parameters

The **Table 5** give the list of the variables which can be defined in one of the two scales, wavenumber or wavelength.

It is to notice that the parameter which controls the spectral sampling at high spectral resolution (RESPEC) has to be defined in wavenumber only.

FILE	Parameter name	Description
para4a$RSTR.dtp	forig	value in cm^{-1} or µm (according to the value of *flag_lambda*)
	fstop	value in cm^{-1} or µm (according to the value of *flag_lambda*)
4A execution script	numin	value in cm^{-1}
	numax	value in cm^{-1}
	lambdamin (new parameter)	value in µm
	lambdamax (new parameter)	value in µm
	flag_lambda	boolean : - 0 : wavenumber - 1 : wavelength

Table 5: 4A parameters which can be defined in one of the two scales (wavenumber/wavelength).

3.1.5.2. Convolution by an instrumental function

The user can prepare his/her own ISRF function by following the section 3.2.4 of [DR1].

In the case of wavelength mode, a conversion from wavelength to wavenumber of the ISRF function(s) has to be done. The conversion is done by a generic subroutine which takes as input the wavelength parameters of the ISRF and gives as output the ISRF converted in wavenumber. This subroutine is named convifct_wltown.f90. The user could report to the section 3.2.4 of [DR1] for more details.

As an example, the user can follow one of the two ISRF functions already defined in wavelength: see the subroutine ifctgauss_lambda.f90 for the Gaussian function and the subroutine ifctoco-2_lambda.f90.

3.1.5.3. Test case

Some examples of wavelength simulations can be find in the non-regression tests, in the folder test/scripts/swir_nir/. In the folder, the subfolders with the “gauss_lambda” extension contains the scripts that runs the simulations with the gauss instruments and bands defined in term of wavelength. For example, the “test_gauss_lambda_b1/” subfolder contains the script “test_gauss_lambda_b1.sh”. To launch this script, use the following command:

./test_gauss_lambda_b1.sh `$IDCOMPIL`

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

where the variable `IDCOMPIL` identifies the library compiled previously by the user (see 2.2.3.1).

The result are in the directory: `outputascii.$IDCOMPIL`

3.2. Running 4A/OP-subroutines

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, the following targets are available:

1. runliresp4a : Ascii file output, both cases of "infinite" resolution spectra or spectra convolved with an instrument function
2. runsp4a : Binary file output, both cases of "infinite" resolution spectra or spectra convolved with an instrument function
3. runatmformat : format in Binary file input file atm4a*.dsf of atmospheric molecule profiles
4. runaerformat : format in Binary file input file aer4a*.dsf of atmospheric aerosol profiles
5. runifct : format in Binary file input file ifct*.dsf of instrument function
6. runstatanalys : compute statistics on the difference of two spectra: spc4a$(SPC1) and spc4a$(SPC2)

As an example, the command line executed for a convolved spectrum simulation is:

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

where the global variable values define the configuration of the run (see section 5.2 for the global variable values or section 3.1.1 for an example).
For a specific application, the user can create a new shell script by choosing the target and the values of global variables in agreement with the application and relative input data files.

3.3. Viewing 4A/OP-subroutine simulation results

4A/OP-sub produces only one binary file containing all results. An interface program 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 according to the simulation case in the table below.

Output data in ASCII files always include radiances and brightness temperature and can also include other quantities id demanded (Jacobians and/or transmittances) for both cases, "Pseudo-infinite" or "Instrument" simulation case.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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	

Table 6: 4A/OP-sub output file description. Please see document [DR1] for a detailed description of the files' content.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

4. Graphical User Interface (GUI)

A Graphical User Interface (GUI) for 4A/OP is available. 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 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.

4.1. Produce or open a GUI parameter file

A parameter file with a format specific to the GUI is created in order to store all the parameters required to run the 4A model from the GUI. This file allows to save the experiment state chosen by the user. This parameter file can be saved every new experiment (menu file access: Save or Save as) into a file named filename.4a; the character string filename is suggested by the user if the file is saved with the menu file access "Save" or "Save as". Available GUI parameter files can be open (menu "File / Open") for loading parameters that have been already used and recorded. This kind of parameter file can be saved in the directory model/input/.

4.2. Produce a script file

The GUI produces a script file to run 4A from the user's choices presented below. The parameter file para4a$RSTR.dtp (see section 5.3) attached to the run experiment is created and remains hidden for the user. The variable $RSTR is equal to the character string filename. 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 an other file called scripts/test in the model installation directory (set with menu File/Preferences). Then scripts/test is executed.

It is important to notice that the parameter DELDS has been fixed to 15 cm^{-1} and the parameter SHIFT has been set to "NO": the user can not modify them.

4.3. User's choices

The user's choices are presented through a notebook that displays several pages for each type of parameters. Examples of pages are described by the following screenshots.

- Menu description
  - Menu File: New, Open, Save, Save as, Preferences, Exit;
    - Preferences: This functionality indicates the locations of both the 4A model sources and Gnuplot scripts. This allows the user the possibility to specify another location of the model sources (for the recording of the parameter, script and result files) or/and of the Gnuplot scripts (in order to enable the user to use its own scripts):
  - Menu Run: Run 4A/OP, View output log;
    - Run 4A/OP: It launches the 4A/OP model execution. This action requires that the current set of parameters has been saved on disk,
    - View output log: This functionality allows the visualization of the standard output and standard error of the run in progress ; that can be saved,

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Menu Visualization: This menu allows visualization (and then printing) of graphs of 4A/OP model products (spectra, jacobians, transmittances, atmospheric profiles and difference between two spectra). Those graphs are governed by Gnuplot scripts;
- Menu Help: Help on 4A interactive, Credits:
  - Help on 4A interactive: It is the on-line help of the 4A GUI.

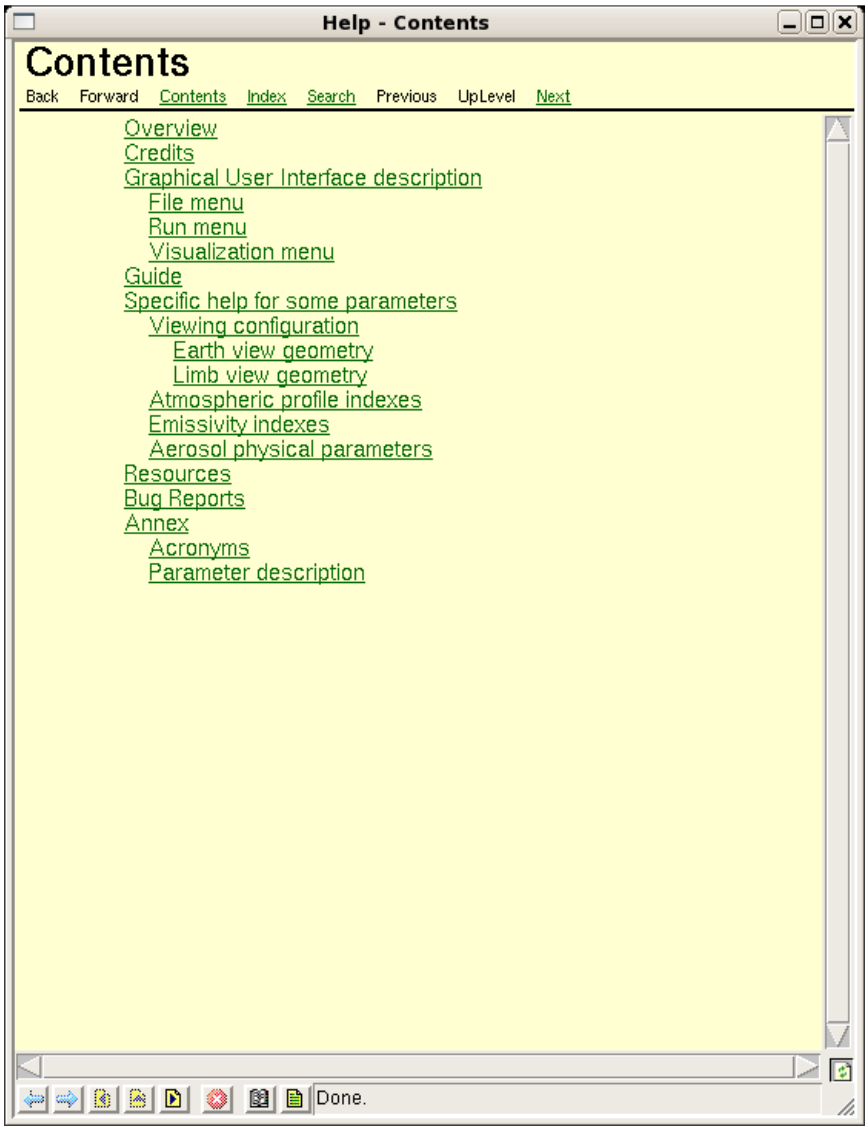


Figure 11: Screenshot of on-line help page that appears by choosing the menu "Help/Help on 4A/OP interactive".

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Page “General”

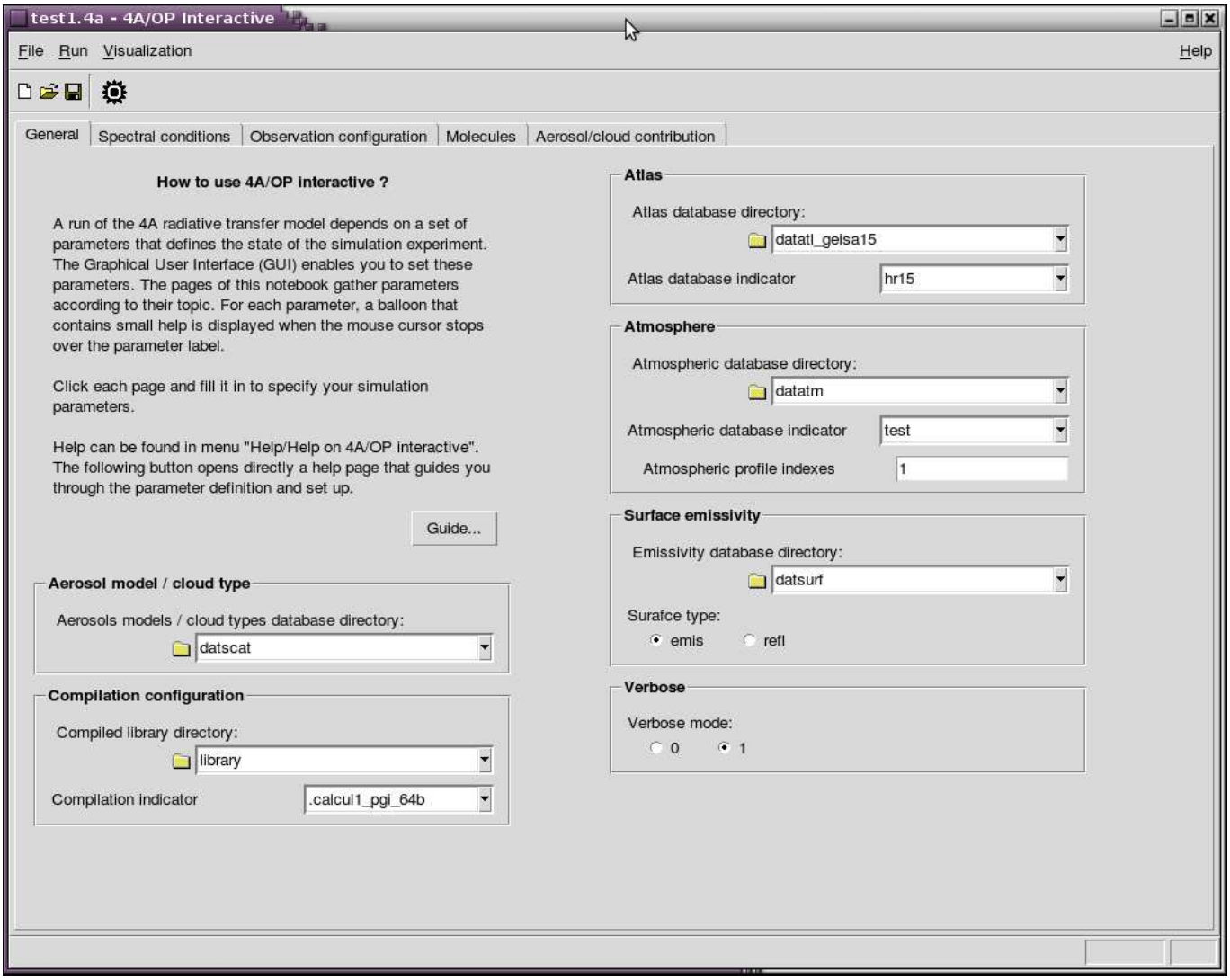
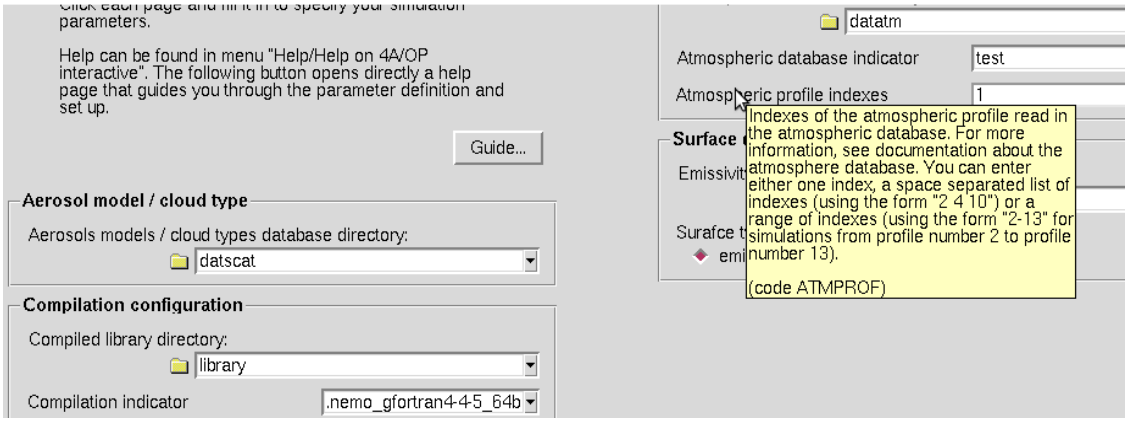


Figure 12: Screenshot of the GUI first page that shows the general parameters required for a 4A run.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Additional help on parameters:

  Additional help is available for every parameter and is displayed when the mouse pointer stops over the parameter label (a frame appears);



- Section “Atlas”:

  The user is able to choose the atlas to be used by 4A. Indeed, two types of atlas database are available up to now: the Atlas for the TIR spectral domain and the Atlas for the SWIR spectral domain. The first one is in the directory “datatl_tir” with the atlas indicator hr09 an dth second one is in the directory “datatl_swir” with the atlas indicator hr08;

- Section “Atmosphere”:

  The user is able to enter several atmospheric profile indexes if several profiles are available in the atmospheric database file chosen by the user. Then the model runs with a loop on several profiles;

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Guide:

  By clicking the button “Guide” available in the page “General”, a help/guide page appears. It explains step by step the use of the GUI from the parameter definition to the 4A/OP model output visualization.

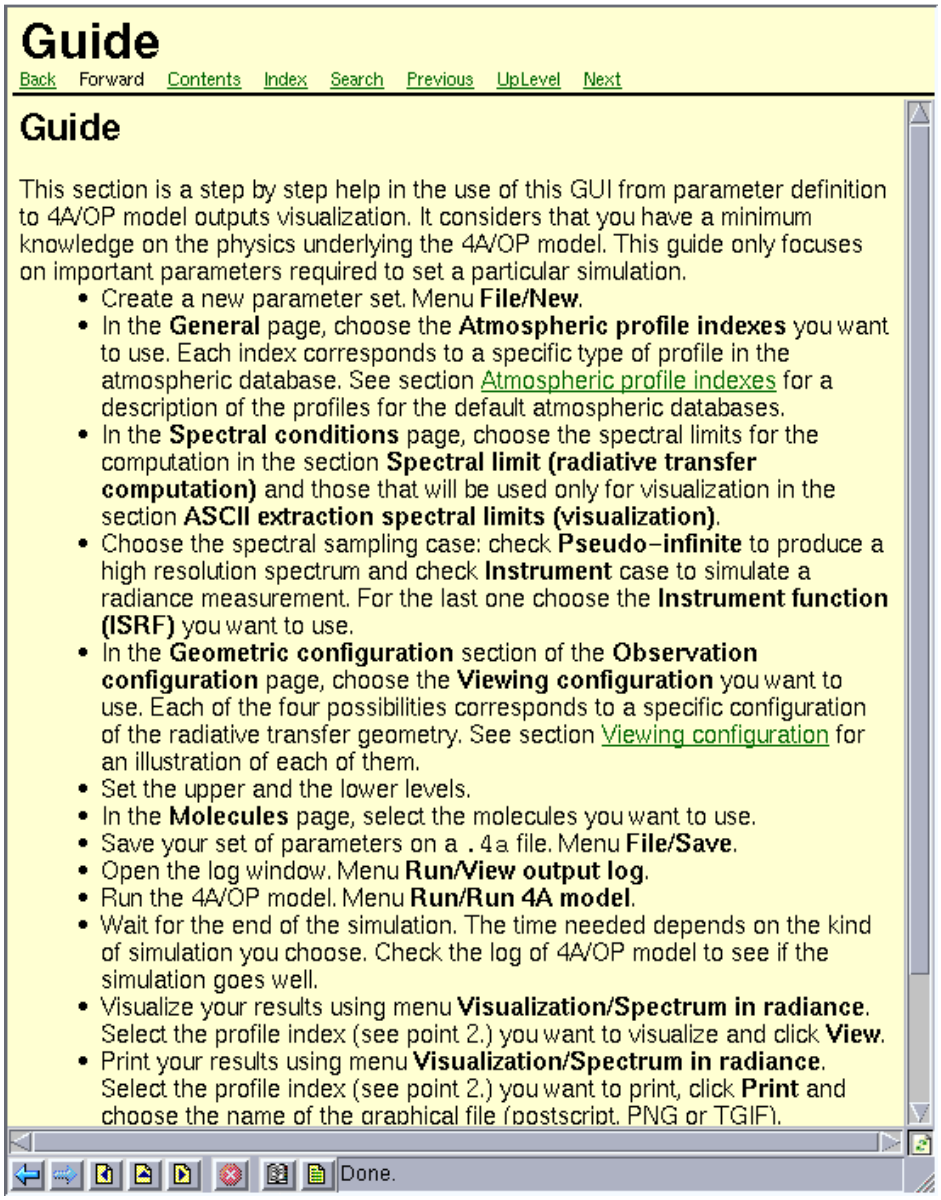


Figure 13: Screenshot of the GUI guide page that appears by clicking the button “Guide”.

- Page “Spectral conditions”

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

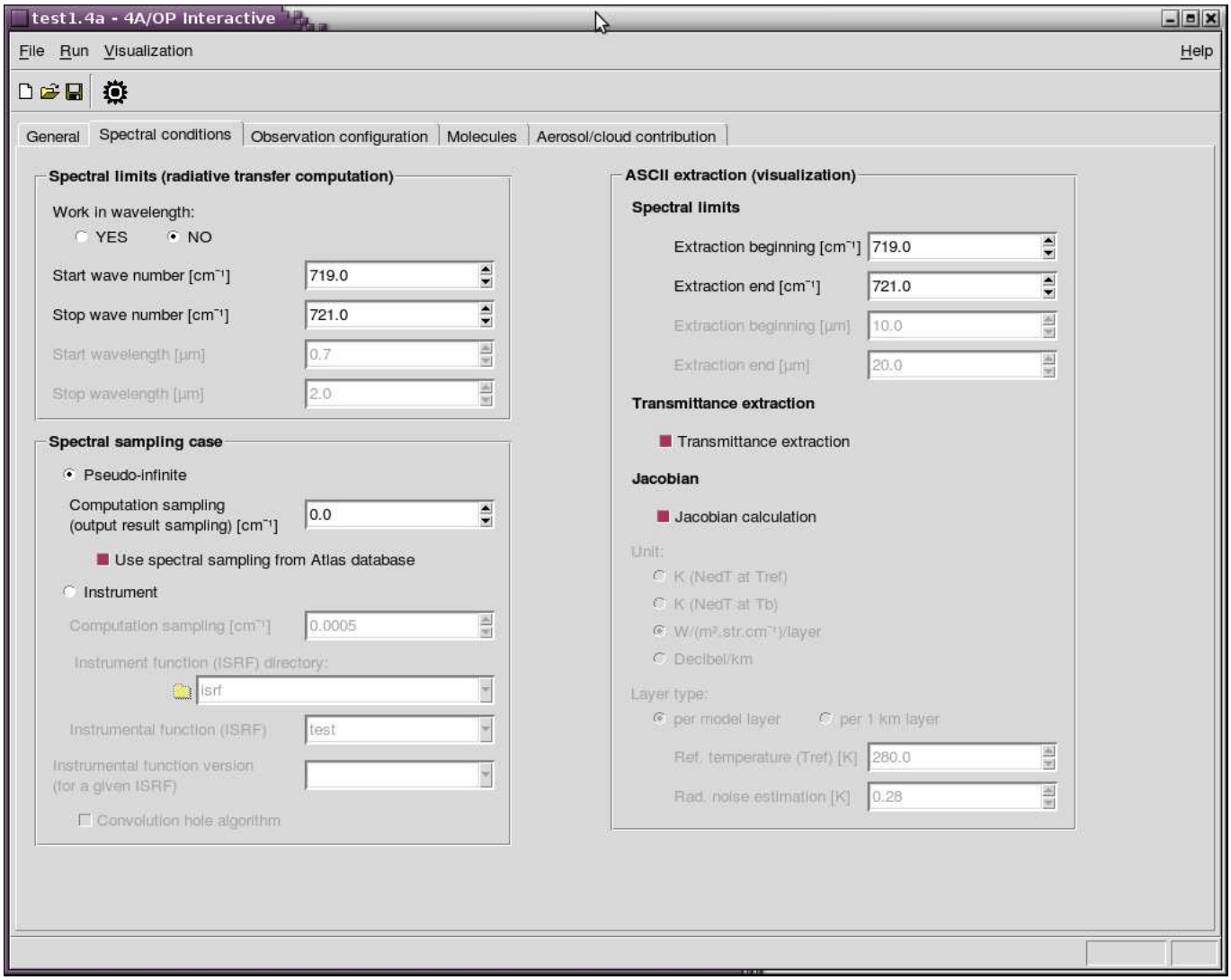


Figure 14: Screenshot of the page concerning the spectral condition parameters for the simulation.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Page “Observation configuration”

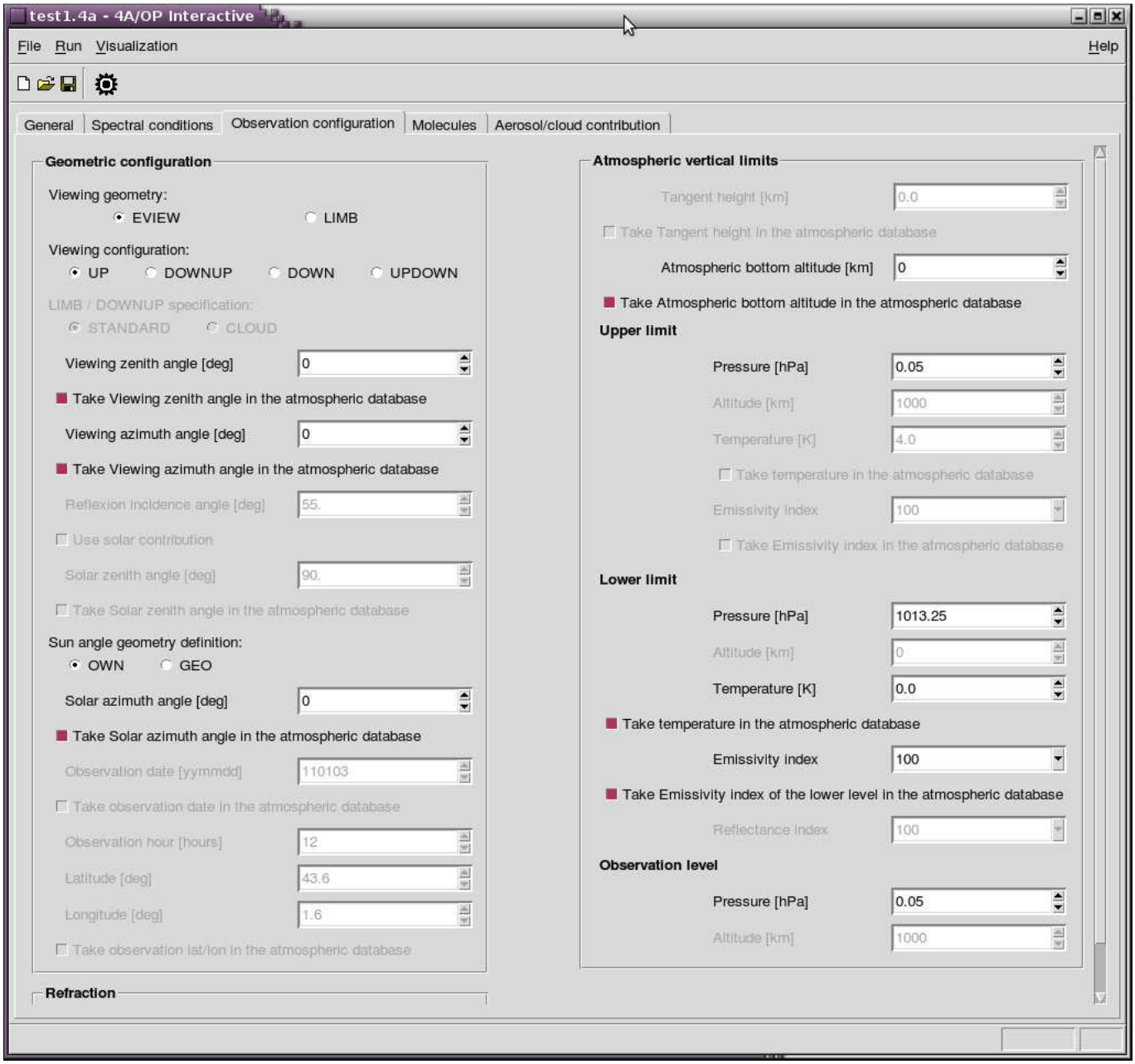


Figure 15: Screenshot of the page concerning the observation configuration parameters.

- Page “Molecules ”

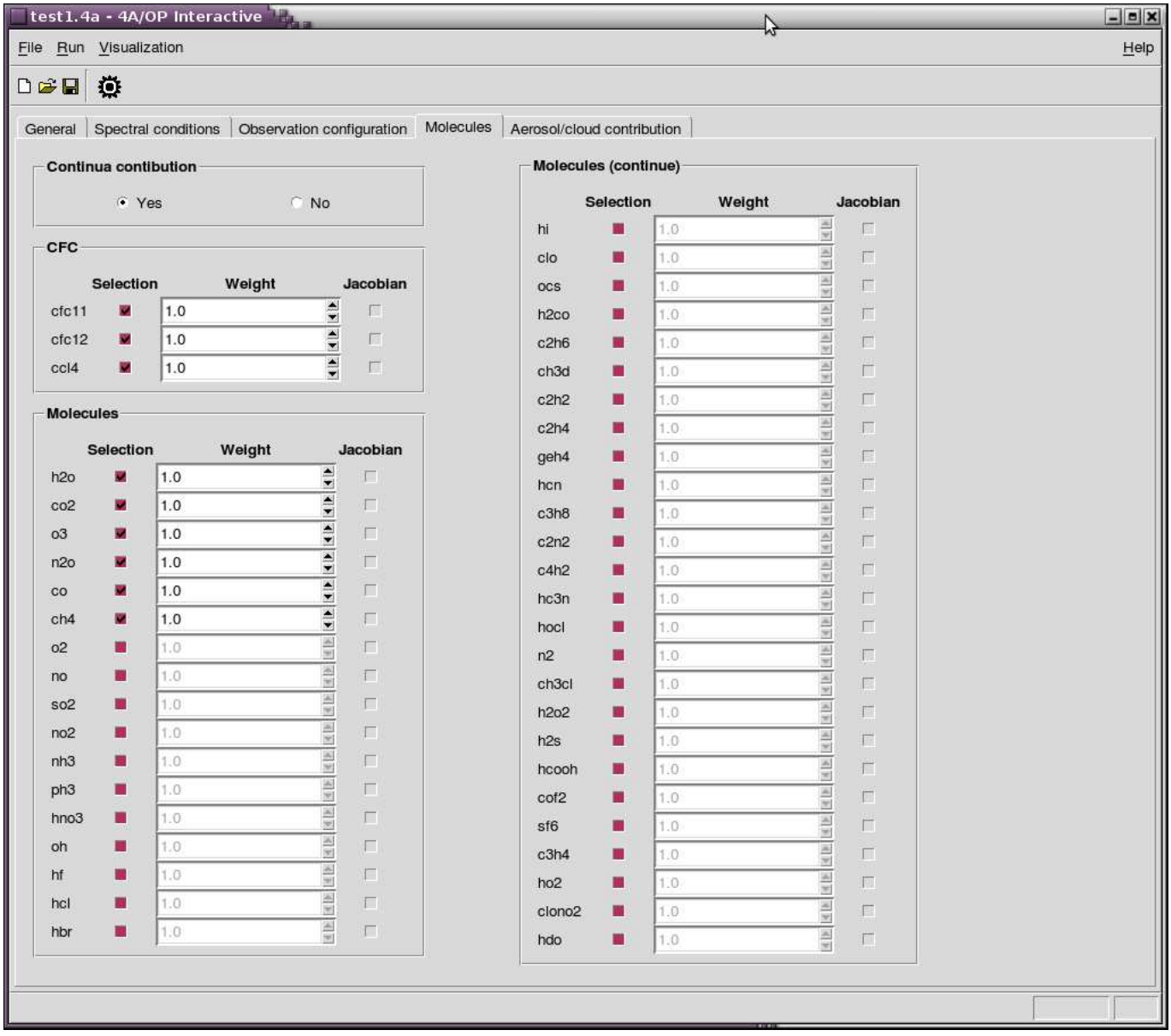


Figure 16: Screenshot of the page concerning the molecule parameters.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- Page “Aerosol/cloud contribution”

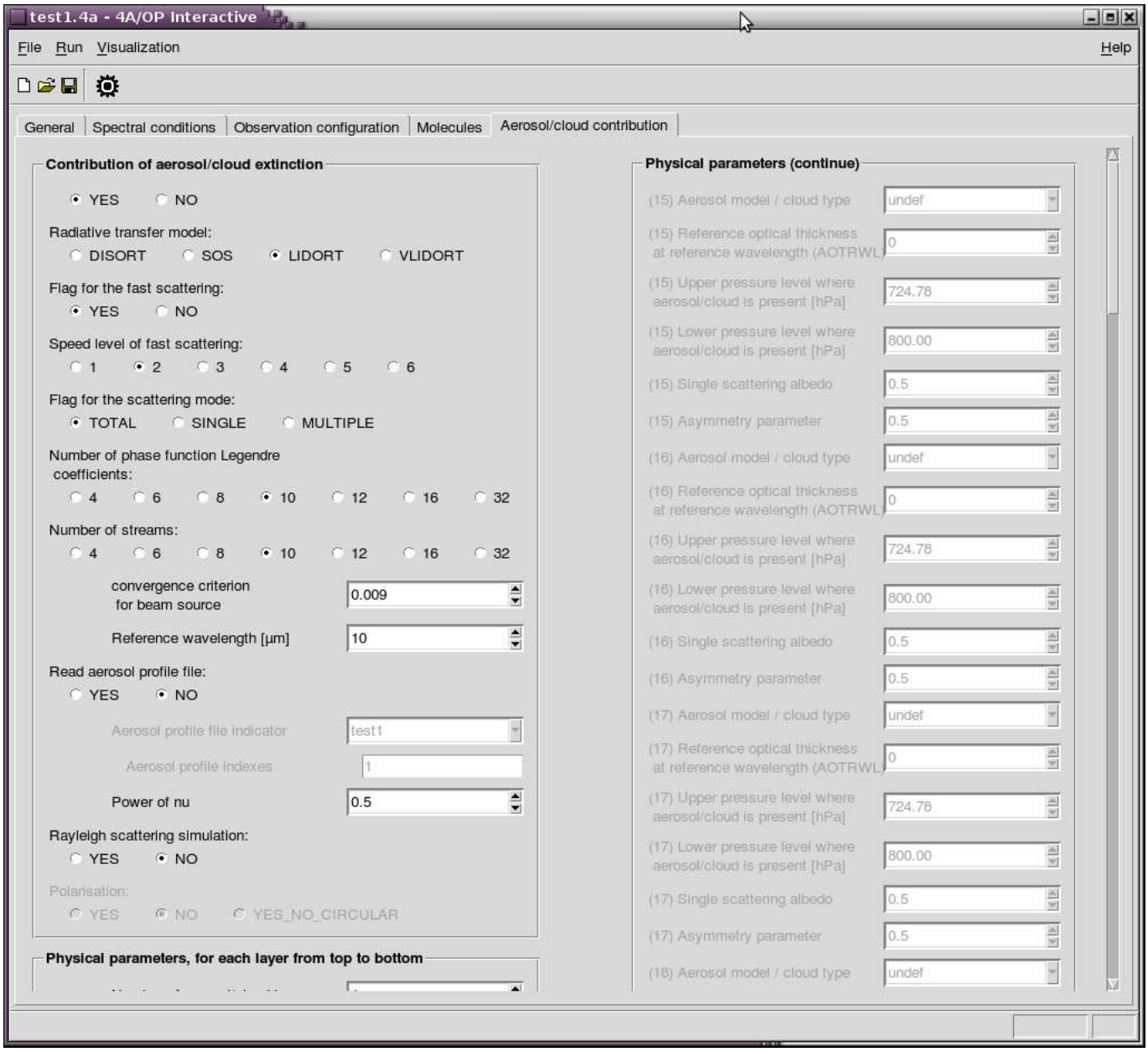


Figure 17: Screenshot of the page concerning the aerosol/cloud contribution parameters.

4.4. Save

The GUI enables the user to save the experiment parameters in order to use it again with the menu “File / Save as” (in the file $filename.4a: see section 4.1).

4.5. Run

The entry “Run 4A/OP” in the menu “Run” creates a run script and the parameter file para4a$filename.dtp. Then, the model runs.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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”. 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.

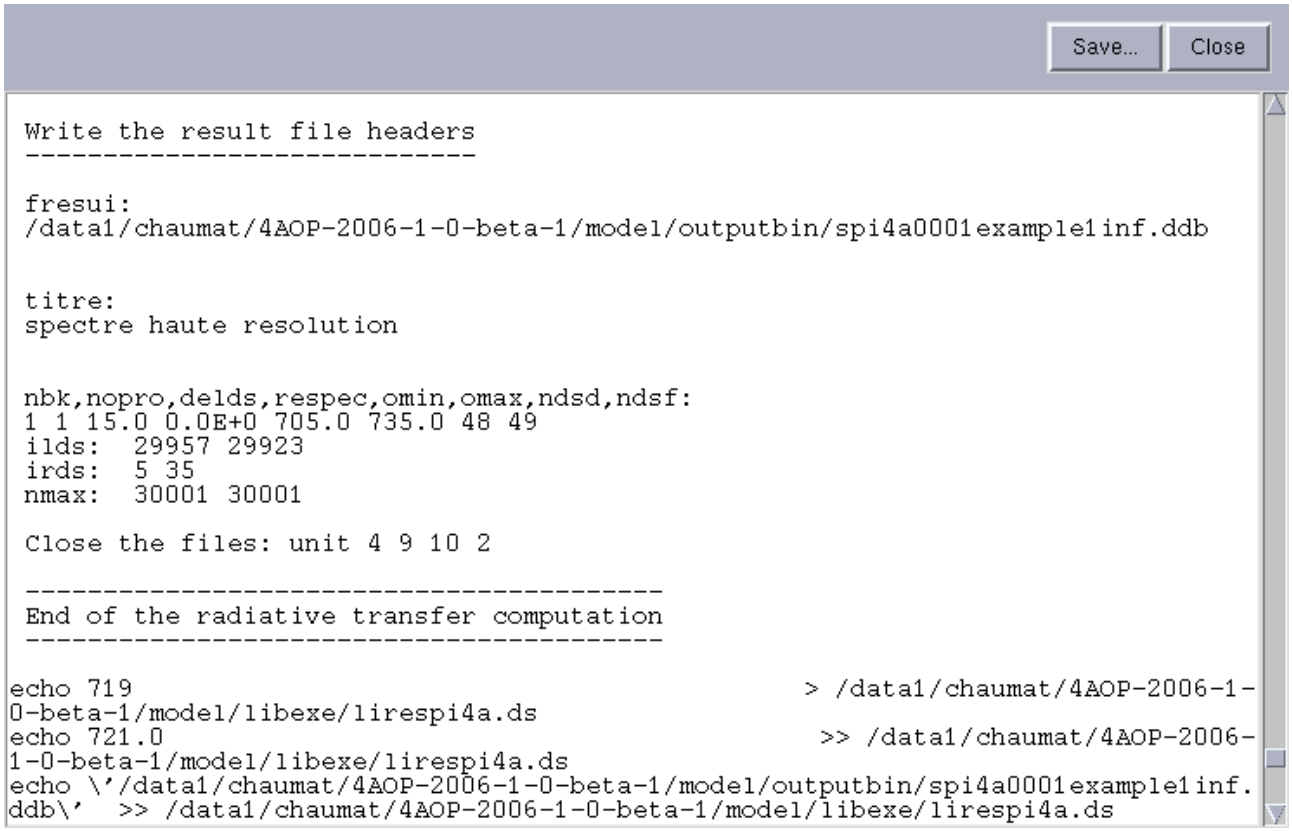


Figure 18: Screenshot of the log window showing a run development example. This window appears by choosing the menu entry “Run / View output log”.

4.6. Visualization

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).

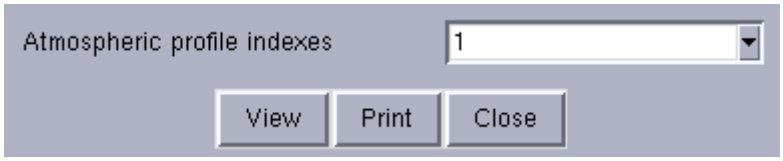


Figure 19: Screenshot of the window that appears by choosing the menu “Visualization / Spectrum in …”.

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

4.6.1. 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.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

4.6.2. 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.

4.6.3. Examples

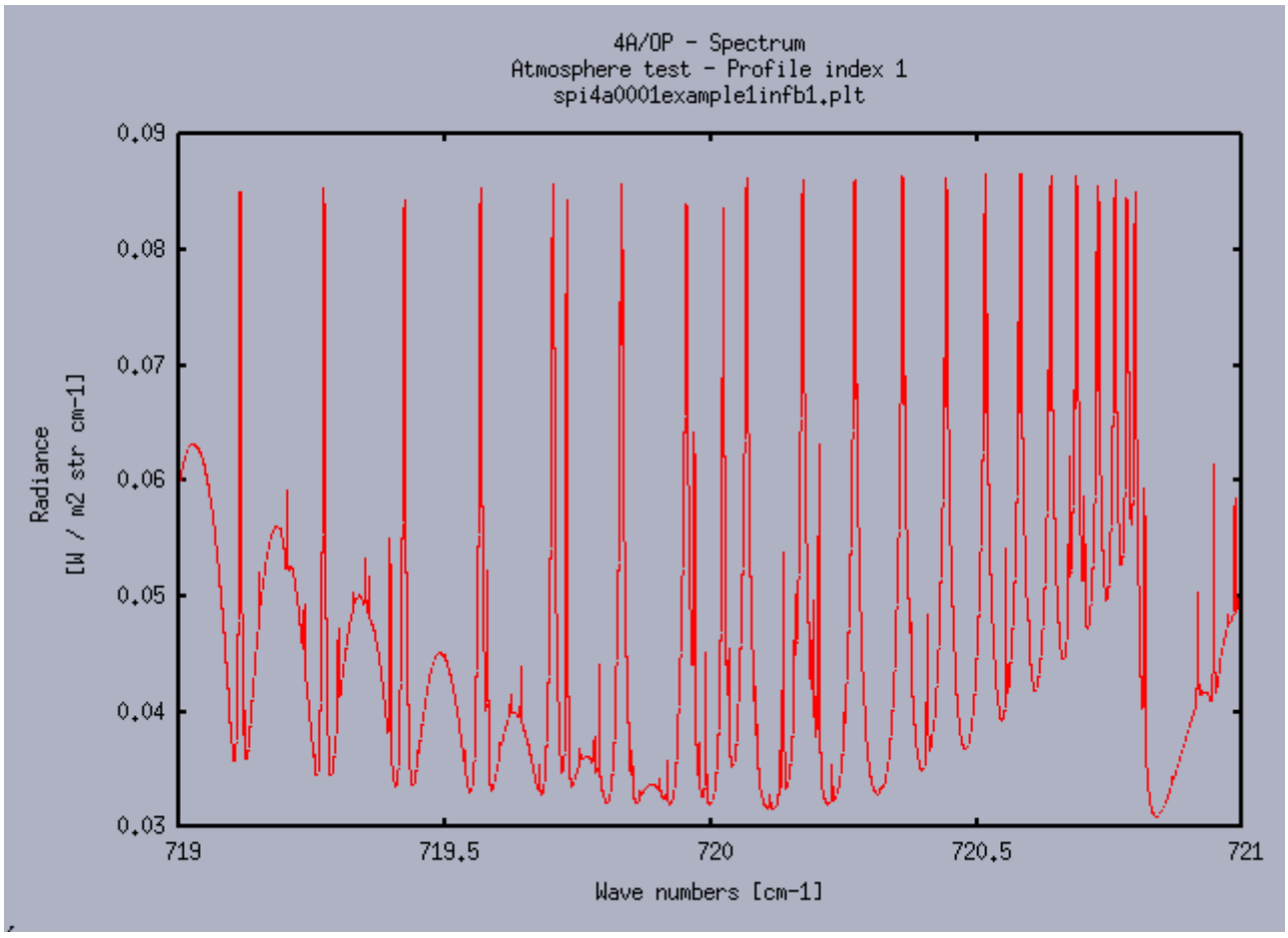


Figure 20: Screenshot of the Gnuplot window showing an example of a spectrum resulting from the current simulation. This window appears by choosing the menu entry “Visualization / Spectrum in radiance”. The name of the ASCII file (*.plt) storing the spectrum is below the title “4A/OP – Spectrum”.

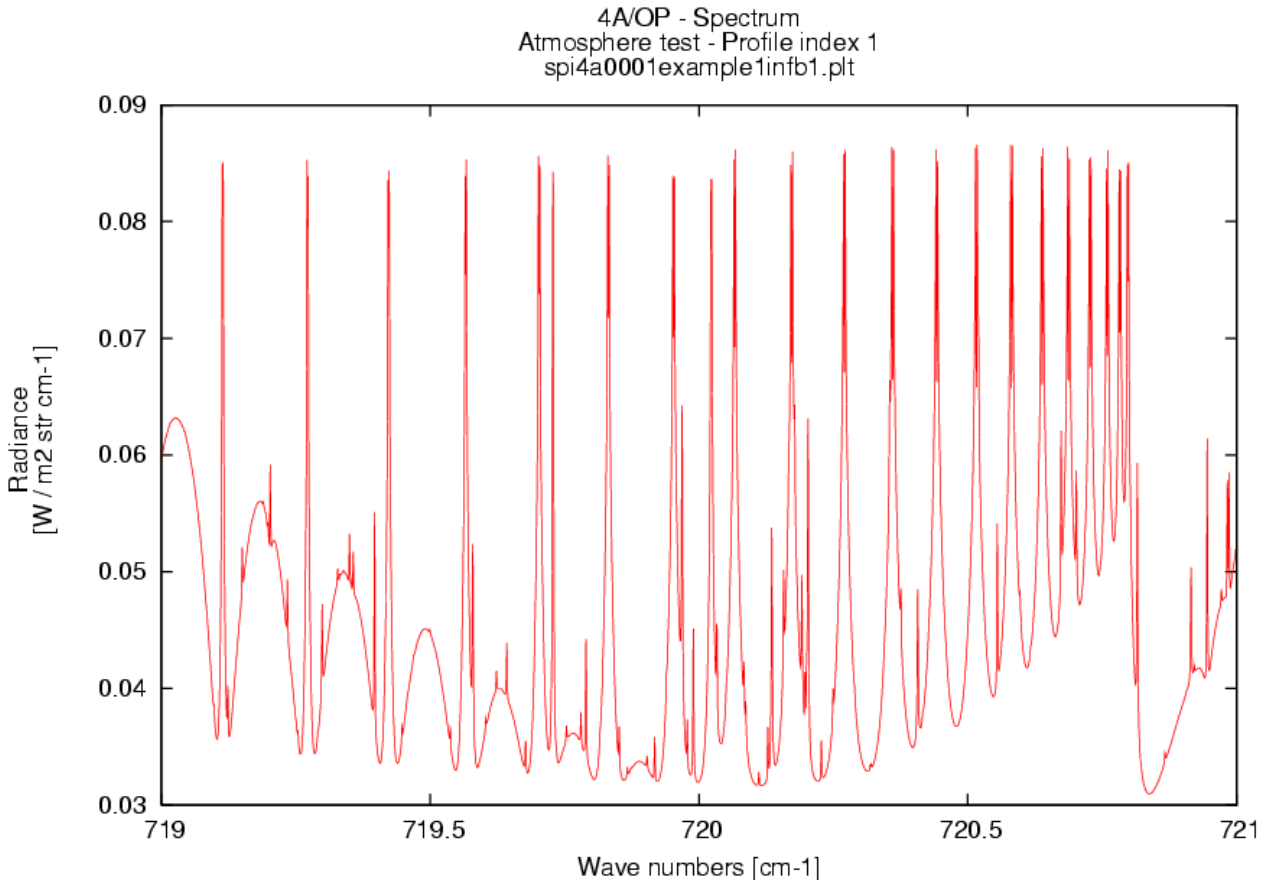


Figure 21: Display of the postscript file illustrating the spectrum of Figure 20. The postscript file is created by clicking the button "Print" in the window associated to the graph (see Figure 19).

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

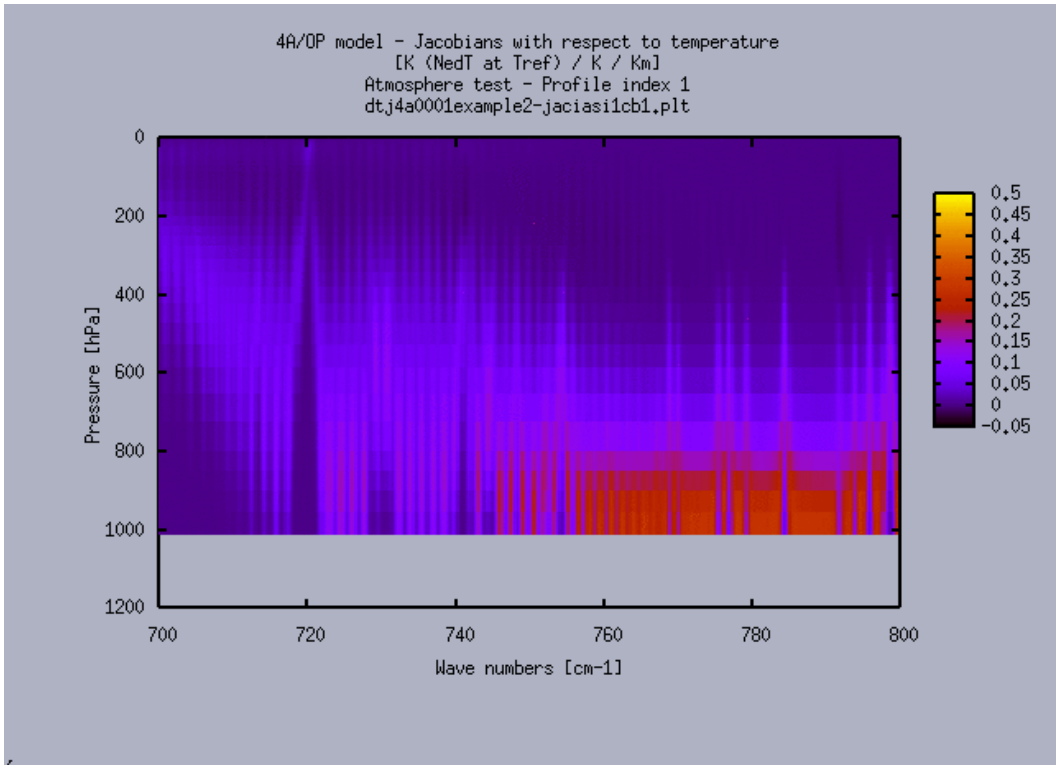


Figure 22: Screenshot of the Gnuplot window showing an example of Jacobians with respect to the temperature. This Gnuplot window appears by choosing the menu entry "Visualization / Jacobians (temperature)".

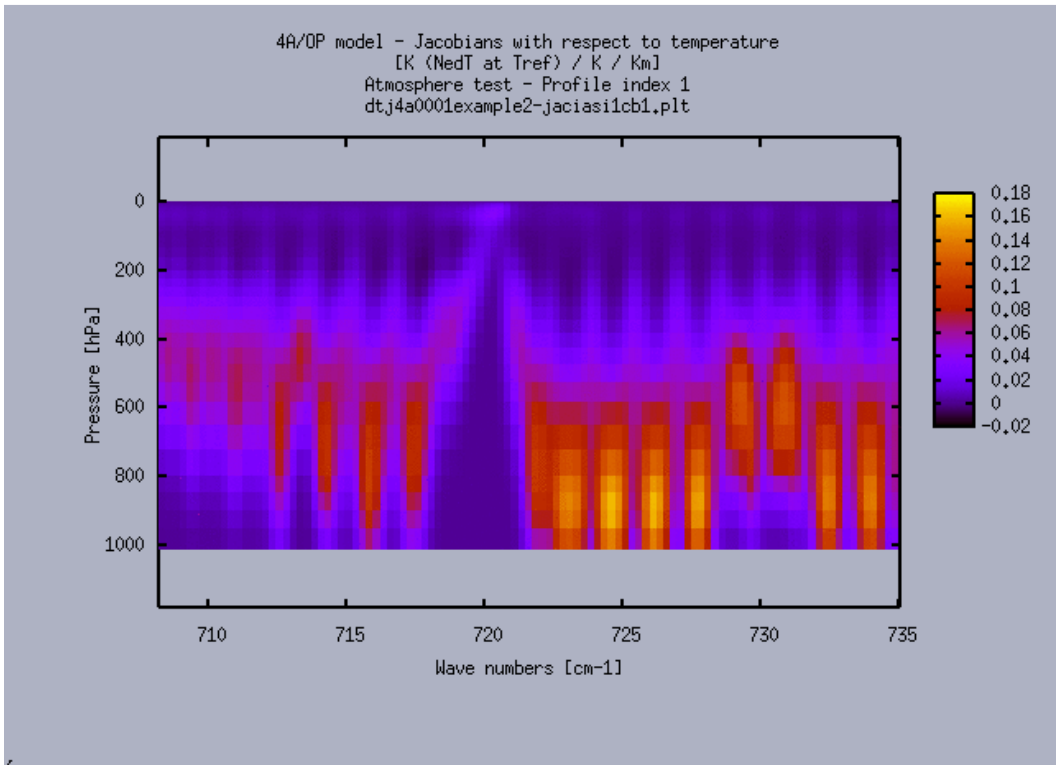


Figure 23: Same as Figure 22 for a selected range obtained by zooming on the graph with the right mouse button.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

4.7. Test

In order to test the good installation of the GUI the user could run several tests cases coming with the software. Currently, the four test cases are the same as the test cases executed in the sotware installation step (see section 2.3 and Appendix B: Test cases for the description of the test cases): the test 1, 14, 15 and 16 has been implemented for the GUI. The input file for the GUI are in the gui/src/ directory:

Test1.4a, castest14.4a, castest15.4a and castest16.4a.

4.8. Future improvements

4.8.1. Run

An extra useful functionality is the possibility to run the GUI from the shell with a parameter file, by using the following instruction for example:

% 4Agui filename.4a &

A practical option could be added in order to enable the user to run 4A with a loop on one or several parameters other than the atmospheric profiles.

4.8.2. Visualization

More functionalities can be added in the menu visualization access:

- View instrumental functions: in order to visualize the ISRF used in the simulation if it is the case of a convolved spectrum;
- View Jacobian for a 1D section of the map at a constant vertical level;
- View statistics;
- A succinct on-line help for Gnuplot.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

5. Appendix A: General description

5.1. Directory description

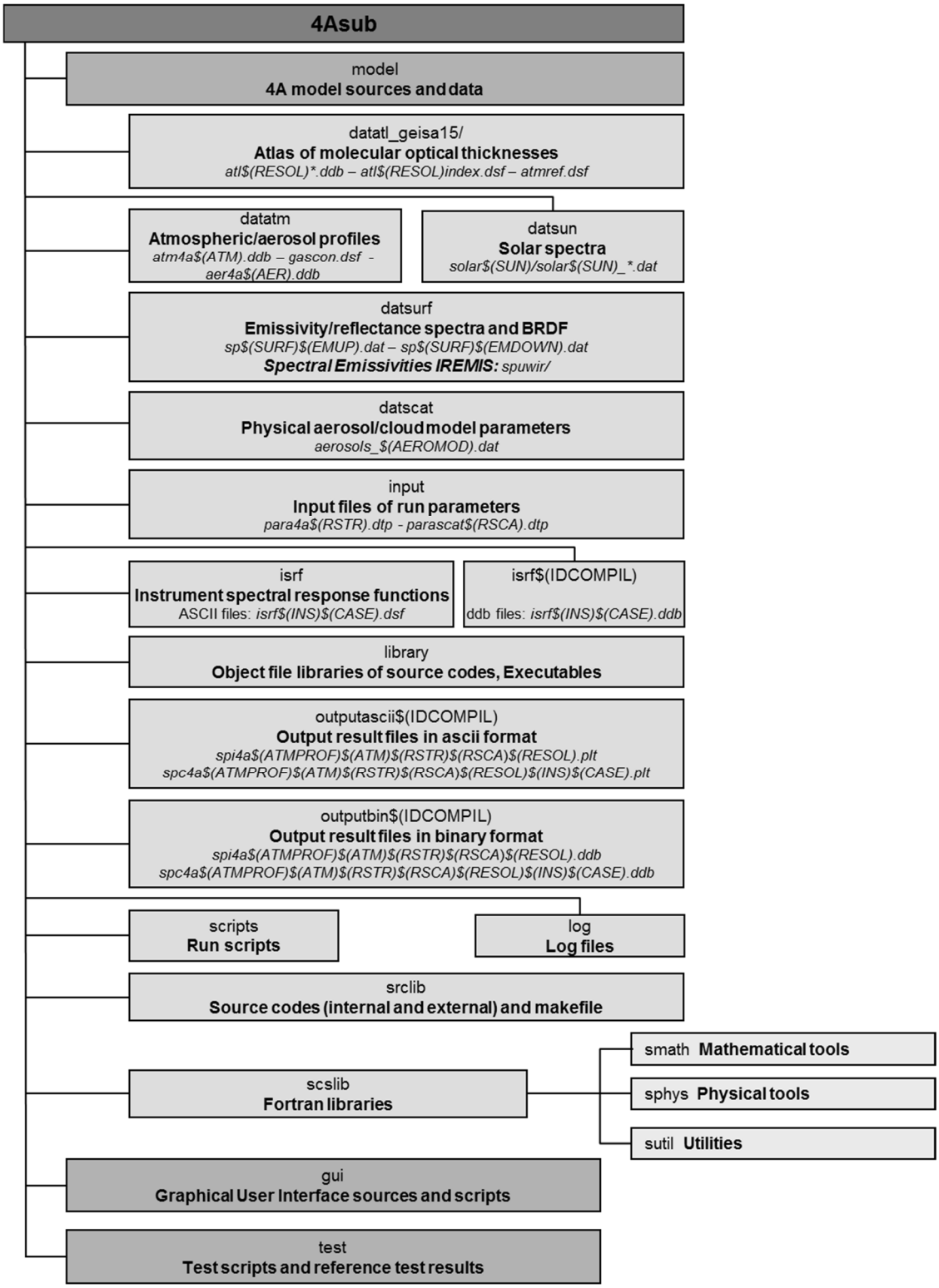


Diagram 1: Directory description of 4A/OP-sub (* for several files).

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

5.2. 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.

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 4asub	4a
ATM	Indicator of the atmospheric database stored in the file atm4a$(ATM).ddb	char	-	makefile / directory datatm	(Presently, possible values are test, test2, satigr_v4.0_moyclas: TIGR atmosphere type)	satigr_v4.0_moyclas
ATMPROF	Index of the atmospheric profile read in the atmospheric database stored in the file named datatm/atm4a$(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-sub 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
AER	Indicator of the aerosol profiles stored in the file aer4a$(AER).ddb	char	-	makefile / directory datatm	(Presently, possible values are test, test2, satigr_v4.0_moyclas: TIGR atmosphere type)	
AERPROF	Index of the aerosol profile read in the aerosol database stored in the file named datatm/aer4a$(AER).ddb	int	-	makefile		
RSTR	Indicator of the parameter file para4a$(RSTR).dtp	char	-	makefile / directory input	(Presently, 19 possible values are available)	NA
RSCA	Indicator of the aerosol parameter file parascat$(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 hr15, that corresponds to the high resolution atlas database using spectroscopic parameters from the GEISA 2015 edition)	hr15

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
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, test3, 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 " ")	" "
EXP	Indicator for the kind of interpolation of the exponential: needed to read the binary file Exponential_$(EXP).ddb.	string	-	makefile / directory input	(Presently, possible values are "Linear", "Exp", "Nearest" or "Spline")	Linear
SUN	Indicator for the solar spectra stored in the directory solar$(SUN)	string	-	makefile / directory datsun	(Presently, possible values are "_planck", "_kurucz", "_tooncss" or "_tooniss")	_planck
RATL	Path for the atlas database	string	-	makefile / directory datatl	The possible values are: 'datatl_geisa15' or any other string decided by the user	datatl
FLAG_LAMBDA	Flag to choose between the two scales: wavenumber or wavelength	i	-	makefile	- FLAG_LAMBDA = 0 : the parameters are defined in wavenumber; - FLAG_LAMBDA = 1 : the parameters are defined in wavelength;	0
IDCOMPIL	Indicator of the used compiler	string	-	makefile / directory library	It is user dependant. Defines the filename of the compilation configuration filename (Makefile.4a.$(IDCOMPIL).inc) and the library directory (library/lib$(IDCOMPIL))	" "
VERBOSE	Indicator for the verbosity of 4A/OP: control the details displayed in the log file	char	-	makefile	The possible values to set verbosity are 'true', 'True', 'TRUE' or 1. Otherwise verbosity is deactivted	

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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 8**)	635
NUMAX	End of the extraction for the conversion (in wave number)	r	cm^{-1}	makefile	NUMAX must be ≤ *fstop* (see **Table 8**)	2880
LAMBDAMIN	Beginning of the extraction for the conversion (in wavelength)	r	micron (µm)	makefile	LAMBDAMIN must be ≥ *forig* (*forig* defined in micron, see **Table 8**)	1 (µm)
LAMBDAMAX	End of the extraction for the conversion (in wavelength)	r	micron (µm)	makefile	LAMBDAMAX must be ≤ *fstop* (*fstop* defined in micron, see **Table 8**)	10 (µm)
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	4

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
					- W/(m² str cm⁻¹) per layer for 1 g/g variation of ro - W/(m² str cm⁻¹) for 100% variation of the surface emissivity • UNIT = 6: Initial unit - W/(m² str cm⁻¹) per Kelvin.layer - W/(m² str cm⁻¹) per layer for 10% variation of ro • UNIT = 7: - W/(m² str mum) per Kelvin.layer W/(m² str mum) per layer for 10% variation of ro	
TREF	Reference temperature for conversion into NedT	r	K	makefile		280
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 7: Global variables initialised through the file makefile

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

5.3. Run characterization parameters

5.3.1. General simulation 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 8**) describes every parameter. Please, also refer to the Reference Documentation [DR4].

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 4 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 emissivity of the top level read directly in parameter file para4a.dtp or in the file datsurf/ spemis$EMUP.dat	r	-	Ini: parameter file para4a.dtp	Used if *TRAJET*≠UP If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	100
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.
SURF	Surface choice: emissivity, reflectance or BRDF	string (len=4 max)	-	Ini: parameter file para4a.dtp		emis

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
EMDOWN	Index for the emissivity of the lower level read directly in para4a.dtp or in the file datsurf/ sp$SURF$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: parameter file para4a.dtp	Used if GEOMID= 'LIMB' If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0.
TETA1	Primary path angle (viewing angle) scaled to ground (altitude z=0).	r	degree	Ini: parameter file para4a.dtp	*TETA1* ≥ 0. and *TETA1* ≤ 90. If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0.
TETA2	Secondary path angle (reflexion incidence angle) at z=0.	r	degree	Ini: parameter file para4a.dtp	Used if TRAJET = *DOWNUP* or *UPDOWN*	55.
PHIV	Viewing azimuth angle (deg) (scattering case only)	r	degree	Ini: parameter file para4a.dtp	*PHIV* ≥ 0. and *PHIV* ≤ 360. If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0.
TYP_ANG_SUN	Sun angle geometry definition (OWN/GEO)	string (len=3 max)	-	Ini: parameter file para4a.dtp	If *OWN* : not used If *GEO* : used to calculate the solar geometry, *TETAS* and *PHIS*, from *DATE*, *TU*, *LAT* and *LON*.	OWN

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
TETAS	Solar zenith angle	r	degree	Ini: parameter file para4a.dtp	*TETAS* ≥ 0. and *TETAS* ≤ 90. Used only if *TYP_ANG_SUN* = *OWN* If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	90. (no Sun)
PHIS	Solar azimuth angle	r	degree	Ini: parameter file para4a.dtp	*PHIS* ≥ 0. and *PHIS* ≤ 360 Used only if *TYP_ANG_SUN* = *OWN* If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0.
DATE	Observation date	string (len=6 max)	-	Ini: parameter file para4a.dtp	Format : yymmdd If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0
TU	Observation hour	r	hour	Ini: parameter file para4a.dtp	decimal hour UT: 0:24 Used only if TYP_ANG_SUN = GEO	0
LAT	Latitude	r	degree	Ini: parameter file para4a.dtp	Used only if TYP_ANG_SUN = GEO If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0
LON	Longitude	r	degree	Ini: parameter file para4a.dtp	Used only if TYP_ANG_SUN = GEO If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	0
REFRA	Refraction computation identifier	string (len=6)	-	Ini: para4a.dtp	YES or NO	NO

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
ZATMBOT	Height of lower atmospheric level	r	km	Ini: para4a.dtp	if < 0: computed in atmsph through barometric height formula If -999 the value is taken from the atmospheric profile atm4a$ATM.ddb (see section 3.2.2 in [DR1])	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/wavelength of the simulation	r	cm^{-1}/µm	Ini: parameter file para4a.dtp	*FORIG* must be ≥ forig_atlas*	600.
FSTOP	Last wave number/wavelength of the simulation	r	cm^{-1}/µm	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
HOLE	Indicator of hole algorithm use for the convolution (see Part III 1.4 in Reference Documentation)	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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	Source / Destinat°	References / Remarks	Default value
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.
FPOID	Transmittance calculation flag	string (len=6 max)	-	Ini: parameter file para4a.dtp	YES or 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) - The Jacobian with respect to the surface emissivity is always calculated if any Jacobian w.r.t to the mixing ratio is calculated.	NO
IDSEL	Table to select molecules among the 43 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 43 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 is selected too (idsel=1)	0, …

* forig_atlas and fstop_atlas are the limits of the atlas in wave numbers and are defined in the file `atlas$(RESOL)index.dsf`

Table 8: Run characterization parameters read in the file para4a.dtp

5.3.2. Scattering parameters

The file containing the scattering run parameters is named parascat$RSCA.dtp, where $RSCA is the value of the variable defining the scattering experiment parameters, and it is stored in the directory input. The Table 9 describes every parameter. Please, also refer to the Advanced utilisation manual [DR1].

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	I/O	Source / Destinat°	References / Remarks	Default value
flag_scat	Aerosol presence flag	string (len=3)	-	o	calcrad	set to 'NO' in parascat.dtp	NO
Rad_comp	VLIDORT output radiance component	int	-	o	read_parascat init_vlidort	Possible values are: total(0), direct(1), diffuse(2)	0
scat_mode	Flag for the scattering mode	int	-	o	modaaaa	Values are: 0: total (single + multiple) scattering 1: single scattering 2: multiple scattering	0
flag_fast	Flag for FAST-SCATTERING	int	-	o	modaaaa	If set to 0: no scatfast scattering is done; if set to 1: scatfast scattering is processed	0
level_fast	Speed level of FAST-SCATTERING	int	-	o	classif_bin_LSI	Values between 1 to 6: 1 is the most precise and the slowest, 6 is the fastter and the less precise	2 in case of « pseudo-infinite » simulation 5 in the case of « convolution » simulation
flag_prof	Flag for the aerosol profile	int	-	o		If set to 0: aerosol profile is read in parascat; If set to 1: aerosol profile is read in the file datatm/aer$AER.dsf	0
n_lay_aero	Number of aerosol layers	int	dl.	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠ 'NO'	1
aeromod	Identifier of the aerosol model, from top to bottom layer	string(len=6) [Cst_nlaeromax]	-	o	readaerosols	If flag_scat≠ 'NO'	UNDEF
aottot/aotref	Aerosol optical thickness at	r	dl.	o	pre_disort	If flag_scat≠	0

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	I/O	Source / Destinat°	References / Remarks	Default value
	reference wave number defined by aot_rwl, from top to bottom layer	[Cst_nlaeromax]			pre_sos pre_lidort pre_vlidort	'NO': Initialisation to a spectrally constant value if positive	
p_lay_aero	Upper/lower pressure limits of aerosol layers	R[2,Cst_nlaeromax]	hPa	o	aerolay	If flag_scat≠ 'NO' And If flag_prof=0	1 1
aot_rwl	AOT reference wavelength	d	µm	o	readaerosols	If flag_scat≠ 'NO'	0
omeaero	Single scattering albedo, from top to bottom layer	r [Cst_nlaeromax]	dl.	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠ 'NO': Initialisation to a spectrally constant value if positive	0.5
gaero	Asymetry parameter, from top to bottom layer	r [Cst_nlaeromax]	dl.	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠ 'NO': Initialisation to a spectrally constant value if positive	0.5
kaero	Power of nu (powonu) model parameter k.	r	dl.	o	Init_aerosols	Used if aeromod = 'UNDEF' or 'powonu'	0.
nmom	Number of phase function Legendre polynoms	int	dl.	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠'NO'	10
nmom_low	Number of phase function Legendre polynoms for low precision LSI mode	int	dl.	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠'NO' and flag_fast=1	10
flagray	Rayleigh scattering simulation	int	-	o	pre_disort pre_sos pre_lidort pre_vlidort	If flag_scat≠'NO'	
flagpol	Polarisation flag	int	-	o	pre_vlidort	If flag_scat='VLI'	
nstream	Number of streams	int	dl.	o	pre_disort pre_lidort pre_vlidort	If flag_scat= 'DIS'	10
nstream_low	Number of streams for low precision LSI mode	int	dl.	o	pre_disort pre_lidort pre_vlidort	If flag_scat= 'DIS'	4

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Symbol	Descriptive Name	Type	Units	I/O	Source / Destinat°	References / Remarks	Default value
accur	DISORT convergence criterion if beam source is present	d	relative radiance unit	o	pre_disort	If flag_scat= 'DIS'	0.009
prnt	Header for prints, embedded in a DISORT banner	logical (kind=4) [5]	-	o	pre_disort	If flag_scat= 'DIS'	FALSE
disheader	Array of DISORT logical print flags	string (len=127)	-	o	pre_disort	If flag_scat= 'DIS'	4AOP_DISORT test

Table 9: Run characterization scattering parameters read in the file parascat.dtp

5.4. Main 4A/OP-Sub program

The main program available in the 4A package is the Fortran program mainaaa.f90 for which an extract is shown here below.

```
write(jlist,*)
write(jlist,*) '----------------------------------------'
write(jlist,*) 'Begin of the radiative transfer computation'
write(jlist,*) '----------------------------------------'
write(jlist,*)
write(jlist,*)

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

! Compute 4A radiative transfer
call computeaaaa(jlist, Params, Results, Results_p, ErrCode)

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

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

5.5. 4A/OP-Sub input parameter substructures

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

```
! Reference atmospheric parameters used for atlas computation
type :: PAtmRef
  real(kind=DOUBLE)                          :: psolref, tsolref, secref
  real(kind=DOUBLE), dimension(Cst_nlevref)          :: plref, pcref
  real(kind=DOUBLE), dimension(Cst_nlevref,Cst_nprof)   :: tlref, tcref
  real(kind=DOUBLE), dimension(Cst_nlevref,Cst_ncorps)  :: rolref, rocref
end type PAtmRef

! Default gas concentration profiles
type :: PDefaultGas
```

```
   integer(kind=LONG)                          :: nlevgas
   real(kind=DOUBLE), dimension(Cst_nlevelmax)          :: plgas
   real(kind=DOUBLE), dimension(Cst_nlevelmax,Cst_ncorps)  :: rolgas
 end type PDefaultGas

 type :: PEmissivities
   character(len=60)                    :: cemup
   character(len=60)                    :: cemdown
   integer(kind=LONG), dimension(2)           :: nonde
   integer(kind=LONG), dimension(2)           :: emflag
   real(kind=DOUBLE),  dimension(:,:), allocatable :: tabemup, tabemdown
 end type PEmissivities

 type :: PBrdf
   integer(kind=LONG)                      :: nbbrdf    ! number of BRDF model
   character(len=10)  , dimension(:), allocatable   :: brdf_name ! names of brdf model
   integer(kind=LONG) , dimension(:), allocatable   :: indexbrdf ! VLIDORT index for BRDF
   real(kind=DOUBLE) , dimension(:), allocatable    :: factorbrdf ! amplitudes for each BRDF model
   integer(kind=LONG), dimension(:), allocatable    :: nbparamperbrdf ! numbers of parameters for each BRDF
   real(kind=DOUBLE) , dimension(:,:), allocatable  :: parambrdf     ! values of BRDF parameters
 end type PBrdf


 ! Instrumental function characteristics
 type :: PIsrf
   integer(kind=LONG)                       :: nbk,itypeconv
   integer(kind=LONG)                       :: irdeb
   integer(kind=LONG), dimension(Cst_knumax,Cst_nbmax) :: irecf, kfmo
   integer(kind=LONG), dimension(Cst_nbmax)          :: nknu
   real(kind=DOUBLE),  dimension(Cst_nbmax)          :: nufdeb, nuffin, dnds
   real(kind=DOUBLE),  dimension(Cst_knumax,Cst_nbmax) :: nufo, wnuo, dnuo
   real(kind=DOUBLE)                        :: pdnuco
   real(kind=DOUBLE)                        :: pascon
   integer(kind=LONG)                       :: numreckfmo, numrecnufo, &
                                    numrecwnuo, numrecdnuo, &
                                    numrecirecf, numrecisrf
   integer(kind=LONG)                       :: ibuff_kfmo, ibuff_nufo, &
                                    ibuff_wnuo, ibuff_dnuo, &
                                    ibuff_irecf
   character(len=100)  :: fisrfu, titre

 end type PIsrf

 ! Molecule identification
 type :: PMolecule
   integer(kind=LONG), dimension(Cst_ncorps) :: idsel, idjac
   real(kind=DOUBLE), dimension(Cst_ncorps)  :: rocoef
 end type PMolecule

 ! Simulation definition parameters
 type :: PSimDef
   real(kind=DOUBLE)   :: pzup, pzdown, pzobs
   real(kind=DOUBLE)   :: tdown, tup
!    real(kind=DOUBLE)   :: sectta1, sectta2
   real(kind=DOUBLE)   :: tta_1, tta_2 !, secttas : not used
!    real(kind=DOUBLE)   :: theta_s  -> replaced by tta_s
   real(kind=DOUBLE)   :: t_u, phi_v, phi_s, tta_s, latdeg, londeg
   real(kind=DOUBLE), dimension(1)   :: zgtan
   real(kind=DOUBLE)   :: zatmbot
   real(kind=DOUBLE)   :: respec
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

```
    real(kind=DOUBLE)   :: forig, fstop, forig_temp, fstop_temp
    real(kind=DOUBLE)   :: rnu
!    real(kind=DOUBLE)   :: pascon
    !  new : from 4A-parallel
    character(len=500)  :: racine3, racine4, racine5
    character(len=60)   :: cemdown, cemup
    character(len=6)    :: geomid, refra, yymmdd
    character(len=6)    :: trajet, contin, jacob, conv, hole
    character(len=6)    :: shift, fpoid, intopt
    character(len=4)    :: Surf
    character(len=3)    :: typ_ang_sun
    ! Width of each Atlas spectral band
    real(kind=DOUBLE)   :: delds
    !  new : from 4A-parallel
    ! Spectral domain numbers
    integer(kind=LONG)  :: ndds, nfds
  end type PSimDef

  ! Scattering simulation parameters
  type :: PScatSim
    integer(kind=LONG)                    :: nlaero, nmom, nstream, nmom_low, nstream_low
    integer(kind=LONG)                    :: iflag_ray
    integer(kind=LONG)                    :: iflag_prof
    integer(kind=LONG)                    :: iflag_pol
    integer(kind=LONG)                    :: iflag_fast
    integer(kind=LONG)                    :: iscat_mode
    integer(kind=LONG)                    :: nclass_fast
    integer(kind=LONG)                    :: level_fast
    integer(kind=LONG)                    :: noproaer
    real(kind=DOUBLE)                     :: aotrwl
    real(kind=DOUBLE), dimension(Cst_nlaeromax)    :: aotref, omeaero, gaero
    real(kind=DOUBLE), dimension(2,Cst_nlaeromax)  :: playaero
    real(kind=DOUBLE)                     :: accur
    logical(kind=4)                       :: prnt(5)
    logical(kind=4)                       :: scatdirc
    character(len=3)                      :: rad_comp
    character(len=3)                      :: vliout
    character(len=3)                      :: flag_scat
    character(len=10),    dimension(Cst_nlaeromax) :: aeromod
    character(len=127)                    :: disheader
    character(len=500)                    :: racine4, racine5
    character(len=500)                    :: faero
    !
    integer(kind=LONG),dimension(2,Cst_nlaeromax)  :: layaero
    integer(kind=LONG)                    :: nwn        ! OL 2012/08/17 Aerosol retrieval
    real(kind=DOUBLE), dimension(:,:), allocatable :: wn         ! OL 2012/08/17 Aerosol retrieval
    real(kind=DOUBLE), dimension(:,:), allocatable :: aot        ! OL 2012/08/17 Aerosol retrieval
    real(kind=DOUBLE), dimension(:,:), allocatable :: omega      ! OL 2012/08/17 Aerosol retrieval
    real(kind=DOUBLE), dimension(:,:), allocatable :: g          ! OL 2012/08/17 Aerosol retrieval
    real(kind=DOUBLE), dimension(:,:,:,:), allocatable :: greek_coef   ! OL 2012/08/17 Aerosol retrieval
    !
    real(kind=DOUBLE)                     :: kaero  ! OL 2013/07/09 : AOT(nu) = AOT(nu0)*(nu0/nu)^(k)
  end type PScatSim

   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
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

```
      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 :: 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

!   ! Atmospheric profile
  type :: PProfile
    real(kind=DOUBLE)                           :: pdown
    real(kind=DOUBLE)                           :: tdown
    real(kind=DOUBLE), dimension(Cst_nlevelmax)           :: pl, tl, dp
    real(kind=DOUBLE), dimension(Cst_nlevelmax,Cst_ncorps)   :: rol
    integer(kind=LONG), dimension(Cst_ncorps)             :: idcor
    integer(kind=LONG)                          :: nlevatm
    integer(kind=LONG)                          :: nopro
    real(kind=DOUBLE)                           :: lat
    real(kind=DOUBLE)                           :: lon
    character(len=6)                           :: date
    integer(kind=LONG)                          :: flag_pc
    real(kind=DOUBLE), dimension(Cst_nlevelmax)           :: zl, refil
    real(kind=DOUBLE), dimension(0:Cst_nlevelmax)         :: pc, tc
    real(kind=DOUBLE), dimension(0:Cst_nlevelmax)         :: dzc, gc
    real(kind=DOUBLE), dimension(Cst_nlevelmax,Cst_ncorps)   :: roc
    real(kind=DOUBLE)                           :: ztan
    integer(kind=LONG)                          :: level1
    integer(kind=LONG)                          :: level2
    integer(kind=LONG)                          :: levobs
    real(kind=DOUBLE)                           :: sectta1
    real(kind=DOUBLE)                           :: sectta2
    real(kind=DOUBLE)                           :: secttas
    real(kind=DOUBLE), dimension(0:Cst_nlevelmax)         :: sectta1lev
    real(kind=DOUBLE), dimension(0:Cst_nlevelmax)         :: sectta2lev
```

```
  real(kind=DOUBLE), dimension(0:Cst_nlevelmax)         :: secttaslev
  real(kind=DOUBLE)                                     :: tta_v
  real(kind=DOUBLE)                                     :: phi_v
  real(kind=DOUBLE)                                     :: tta_s
  real(kind=DOUBLE)                                     :: phi_s
  real(kind=DOUBLE)                                     :: delta_phi
  real(kind=DOUBLE), dimension(1)                       :: zgtan
  real(kind=DOUBLE)                                     :: zatmbot
  real(kind=DOUBLE)                                     :: rshift_sun_nu
  real(kind=DOUBLE)                                     :: umu0
  real(kind=DOUBLE)                                     :: esdist
  character(len=60)                                     :: cemdown
end type PProfile

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

! Parameters used for the simulation
type :: Params
  type(PAtmRef)      :: atmref
  type(PDefaultGas) :: gas
  type(PProfile)       :: profile
  type(PSimDef)      :: simdef
  type(PScatSim)     :: scatsim
  type(PMolecule)    :: mol
  type(PAtlasIndex)  :: index
  type(PEmissivities) :: emis
  type(PBrdf)          :: brdf
  type(PIsrf)          :: isrf
  type(PFilenames)   :: fnames
  type(PContinNames) :: contfnames
  type(type_Exponential) :: exponential
  !
  ! Index number of the used atmospheric profile
  integer(kind=LONG) :: nopro
  !
  ! Simulated instrument
  character(len=6)   :: instru

  ! flag lambda
  logical(kind=1) :: flag_lambda

end type Params
```

5.6. Modifying the 4A/OP-Sub structure fields

Here is an example of a fortran code where the 4A/OP-sub input parameter structure Params is modified before and after the calling of the module computeaaaa. The subroutines 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, update_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)'
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

```
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
```

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

6. Appendix B: Test cases

A Python script can be used to launching the non regression tests in the directory test/ as described in section 2.3.2. The "test/" folder contains the following subfolders:

- Installation: contains the non regression tests to check the installation;
- Limb: contains one non regression test in one limb configuration;
- swir_nir: contains the non regression tests in the SWIR/NIR ranges and with different kinds of instruments (Gauss, OCO-2 for example);
- tir: contains the non regression tests in the TIR range and with different kinds of instruments (IASI, AIRS for example);
- wide_bands: contains the non regression tests with wide band intruments reponses (HIRS, MODIS for example).

The following Excel file contains more information about the non regression tests:



SETG-NREG-4AOP-0
012-TS_Ed1_Rev5.xl

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

7. Appendix C: SPECTROSCOPIC DATABASE GEISA

See [DR5] and [DR6].

The identification numbers of the molecules according to GEISA is given by:

H_2O 1	CO_2 2	O_3 3	N_2O 4	CO 5	CH_4 6	O_2 7	NO 8	SO_2 9	NO_2 10
NH_3 11	PH_3 12	HNO_3 13	OH 14	HF 15	HCl 16	HBr 17	HI 18	ClO 19	OCS 20
H_2CO 21	C_2H_6 22	CH_3D 23	C_2H_2 24	C_2H_4 25	GeH_4 26	HCN 27	C_3H_8 28	C_2N_2 29	C_4H_2 30
HC_3N 31	HOCl 32	N_2 33	CH_3Cl 34	H_2O_2 35	H_2S 36	HCOOH 37	COF_2 38	SF_6 39	C_3H_4 40
HO_2 41	$ClONO_2$ 42								
HDO 51									

Table 10: Molecule identification according to GEISA.

The identification numbers of the CFCS is given by:

CFC11 61	CFC12 62	CCl4 63							

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

8. Appendix D: On-line help of the GUI

This section is an extract of the on-line help available with the Graphical User Interface (GUI).

8.1. Overview

This documentation is the online help on the Graphical User Interface (GUI) for the 4A radiative transfer model.

Additional help is available for every parameter and is displayed when the mouse pointer stops over the parameter label as a balloon-help (a frame appears).

We place emphasis on the fact that the GUI is a layer on the model itself. It is designed to assist new users in understanding the 4A/OP model handling. With minimum knowledge, one can set, run and visualize a 4A/OP simulation with this GUI.

We refer the reader to the document "4A/OP – Advanced utilisation manual" (NOV-FE-0035-NT-030_Advanced utilisation manual) for accurate information on the model itself. It contains directives to run the model without this GUI and details some aspects of the physics of the model.

This GUI allows the user to do three main operations:

1. manage sets of parameters;
2. run the model with these sets;
3. visualize results of the model.

8.2. Graphical User Interface description

This section contains specific information on this GUI. Detailed description of classic functionalities (open/save/help menus, toolbar, etc.) is intentionally left aside.

8.2.1. File menu

The GUI manages sets of parameters using files with a specific format (extension .4a). This format allows the user to keep a set of parameters between sessions of the GUI.

Some parameters take their values according to the content of some directories of the 4A/OP model program installation. To avoid problem with this parameters, be sure that the root directory for the 4A/OP model is properly specified. This can be done using the menu **File/Preferences....** The preferences are saved in a file called .4Aguirc in home directory of the user.

8.2.2. Run menu

The menu **Run/Run 4A** launches the 4A/OP model execution. This action requires that the current set of parameters has been saved on disk (see the **File** menu). The user can monitor the 4A/OP model computation standard output by using the menu **Run/View output log**.

In order to run the model, the GUI creates a parameter file and a script that contains the macro definitions and the call to make utility that is used to actually run the model. The root name of the .4a file is used to name them. 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 an other file called scripts/test in the model installation directory (set with menu **File/Preferences**). Then scripts/test is executed.

8.2.3. Visualization menu

This menu allows visualization and printing of graphs of 4A/OP model products (spectra, jacobians, transmittances, atmospheric profiles and difference between two spectra). Those graphs are governed by Gnuplot scripts.

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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

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 plot. They begin with the sign $ (dollar). For example, $dataFile is replaced by the name of the 4A/OP output file to plot.

The same way, you can also access to the values of the parameters. For example in a Gnuplot script, $param_ATM represents the **Atmospheric database indicator** entry in the **Atmosphere** section of the **General** page which is identified by the code ATM. The code for each parameter can be found either in the section Parameter description or, for most parameters, in their balloon-help.

8.3. Guide

This section is a step by step help in the use of this GUI from parameter definition to 4A/OP model outputs visualization. It considers that you have a minimum knowledge on the physics underlying the 4A/OP model. This guide only focuses on important parameters required to set a particular simulation.

1. Create a new parameter set. Menu **File/New**;
2. In the **General** page, choose the **Atmospheric profile indexes** you want to use. Each index corresponds to a specific type of profile in the atmospheric database. See section 8.4.2 for a description of the profiles for the default atmospheric databases;
3. In the **Spectral conditions** page, choose the spectral limits for the computation in the section **Spectral limit (radiative transfer computation)** and those that will be used only for visualization in the section **ASCII extraction spectral limits (visualization)**;
4. Choose the spectral sampling case: check **Pseudo-infinite** to produce a high resolution spectrum and check **Instrument** case to simulate a radiance measurement. For the last one choose the **Instrument function (ISRF)** you want to use;
5. In the **Geometric configuration** section of the **Observation configuration** page, choose the **Viewing configuration** you want to use. Each of the four possibilities corresponds to a specific configuration of the radiative transfer geometry. See section Viewing configuration for an illustration of each of them;
6. Set the upper and the lower levels;
7. In the **Molecules** page, select the molecules you want to use;
8. Save your set of parameters on a .4a file. Menu **File/Save** ;
9. Open the log window. Menu **Run/View output log**;
10. Run the 4A/OP model. Menu **Run/Run 4A model**;
11. Wait for the end of the simulation. The time needed depends on the kind of simulation you choose. Check the log of 4A/OP model to see if the simulation goes well;
12. Visualize your results using menu **Visualization/Spectrum in radiance**. Select the profile index (see point 2.) you want to visualize and click **View**;
13. Print your results using menu **Visualization/Spectrum in radiance**. Select the profile index (see point 2.) you want to print, click **Print** and choose the name of the graphical file (postscript, PNG or TGIF).

We refer the reader to the document "4A/OP - Reference documentation" (NOV-3049-NT-1178) for information on the model and its parameters that can not be found in this help.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

8.4. Specific help for some parameters

This section gives information to complete the balloon-help description for several parameters. This helps new users to understand the meaning of tricky settings.

8.4.1. Viewing configuration

The parameter defining viewing configuration can be found in the **Geometric configuration** section of the **Observation configuration** page.

This parameter defines the configuration of the optical path for the radiative transfer computation. Its value determines the configuration of the source and the observer. It can be either UP, DOWNUP, DOWN or UPDOWN. The Figure 2 to Figure 5 show the configuration for each value as well as what are the upper, lower and the observation levels (see **Atmospheric vertical limits** section of the **Observation configuration** page).

8.4.2. Atmospheric profile indexes

The range of profile indexes and their meaning depend on the atmospheric database used which is selected by the parameter **Atlas database indicator**. See your 4A/OP administrator if you need a new database to be installed.

Here is the description of the profile indexes for some databases.

8.4.2.1. Default databases

This section concerns the databases that come along the default installation of 4A/OP which are arbitrary test database. A number of examples are stored in the datatm directory: test, test2, satigr_v4.0_moyclas. The possible profile indexes for them are 1 and 2 for the test atmosphere and 1 to 5 for the satigr atmosphere.

8.4.2.2. TIGR database

Here are the profile indexes available for the TIGR2000 database (available at LMD):

- 1 to 872: tropical class;
- 873 to 1260: warm temperate class;
- 1261 to 1614: cold temperate class;
- 1615 to 2311: arctic class.

This database is not part of the default installation of 4A/OP.

8.4.2.3. Atmospheric parameters

The following parameters can be set directly in the GUI or set in the atmospheric database files: TETA1, PHIV, TETAS, PHIS, DATE, LAT, LON, ZATMBOT, ZGTAN, TUP, EMUP, TDOWN, EMDOWN. In the case of setting in the atmospheric database files (atm4a$ATM.dsf), the values of these parameters are set to -999 in the parameter files para4a$RSTR.dtp.

8.4.3. Surface type indexes

The surface can be now modelled by taking into account the emissivity or the albedo. The albedo can be Lambertian or calculated with a BRDF model. The BRDF model is available only in the scattering mode by using VLIDORT module.

The input parameter files to define the surface type are located in the datsurf directory and can be modified by the user. Four types of files are stored:

- the emissivity files: spemisXXX.dat;
- the reflectance files: spreflXXX.dat;
- the brdf files: spbrdfXXX.dat.
- the surface emissivity files from IREMIS: spuwir/

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

Since version 2016.1.0, 4A/OP includes the **IREMIS database for the surface spectral emissivities** ([DR21]). 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).

The Table 11 gives a summary of the different possibilities for the parameterization of the top of the atmosphere.

SURF	EMUP	files in the datsurf directory
emis	**'constant'** value	Emissivity read directly in para4a.dtp
	'snyder' index	spemis_snyder***index***.dat
	'user' character_string	spemis***character_string.***dat

Table 11: Summary of the different possibilities for the upper level type parameterization. The EMUP entry is the parameter in the input file para4a$RSTR.dtp.

The Table 12 gives a summary of the different possibilities for the surface parameterization.

SURF	EMDOWN	files in the datsurf directory
emis	**'constant'** value	Emissivity read directly in para4a.dtp
	'snyder' index	spemis_snyder***index***.dat
	'atmsph'	spemis_atmsph***n°profile***.dat
	'uwiremis'	files under the spuwir/ directory
	'user' character string	spemis***character_string***.dat
refl	**'constant'** value	Emissivity read directly in para4a.dtp
	'user' character string	spref***lcharacter_string***.dat
brdf	**'user'**: **model name** of the BRDF: see Table 10 of [DR1]	spbrdf***$EMDOWN***.dat

Table 12: Summary of the different possibilities for the surface type parameterization. The SURF and EMDOWN entries are the parameters in the input file para4a$RSTR.dtp.

Here is a summary of the operation mode of all these options:

- 'constant' value: the written value in the parameter file para4a.dtp 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/. The written value may vary between 0 and 1 as an emissivity value (case 'emis') or as an albedo value (case 'refl').
- 'snyder' index: the index corresponds to a emissivity class associated to an infrared surface emissivity spectrum deduced from Snyder *et al. [DR19]*. 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' of the repertory datsurf/ is used, with XXXX the profile number. It contains spectral variable emissivities given on a range of wave numbers.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

- '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. For the new **surface emissivity IREMIS database** please refers to the official website: http://cimss.ssec.wisc.edu/iremis/.

A new option helps the user to choose the surface type to be used in the radiative transfer: **emissivity** or **reflectance** or **brdf**.

Here are the indexes of the infrared surface emissivity spectrum deduced from Snyder *et al.* [DR19] for the standard installation of 4A/OP:

- 'snyder' 1 – water;
- 'snyder' 2 - snow/ice;
- 'snyder' 3 - arid bare soil;
- 'snyder' 4 - organic bare soil;
- 'snyder' 5 - senescent sparse shrubs;
- 'snyder' 6 - green sparse shrubs;
- 'snyder' 7 - senescent grass savanna;
- 'snyder' 8 - green grass savanna;
- 'snyder' 9 - senescent woody savanna;
- 'snyder' 10 - green woody savanna;
- 'snyder' 11 - senescent broadleaf forest;
- 'snyder' 12 - green broadleaf forest;
- 'snyder' 13 - senescent needle forest;
- 'snyder' 14 - green needle forest.

Here are the **brdf** indexes for the standard installation of 4A/OP:

- desert – desert BRDF model;
- ocean – ocean BRDF model;
- veget – vegetation BRDF model.

We refer the reader who wants to define new surface type to the document [DR1] and in the main part of this document to the section :

- "Emissivity data files" for the emissivity;
- "Reflectance/albedo data files" for the reflectance;
- "BRDF data files" or the BRDF.

For example, if you want to create your own spectral emissivity, you can create an emissivity file in **Emissivity database directory**. Be sure that the name of the new file is spemisX.dat where X is the character string you have to define after the flag 'user' in the parameter file para4a.dtp. This is the same process for the reflectance.

8.4.4. Handling the HDO isotope

A new molecule has been introduced in 4A/OP via the atlases. The main isotope of H_2O, HDO, is now taking into account as a separate and independent molecule and could be handled in the input parameters of 4A/OP as the number 51 (see IDSEL in para4a$RSTR.dtp file).

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

The default behaviour is that the isotope HDO is taking into account automatically if the H_2O is selected. The atmospheric mixing ratio profile of HDO is the same as the H_2O profile. In this case the user does not has to deal with the HDO selection in para4a.

Otherwise, if the user wants to simulate only the HDO molecule, there are two possibilities:

- if the user does not have a mixing ratio profile for HDO, the H_2O profile is set to HDO:
  - the IDSEL has to be set to -1 for HDO (see **Table 8** or section 3.1.2);
- if the user has is own mixing ratio profile for HDO:
  - the IDSEL has to be set to 1 for HDO (this is a normal behaviour as for all other molecules);

The Table 13 sums up the handling of HDO in the input parameters.

IDSEL H2O/HDO	Mixing ratio profile used for HDO	Remark
1/0	H2O	no distinction between H2O and HDO
0/-1	H2O	Simulation of only HDO
0/1	own user profile	Simulation of only HDO

Table 13: H2O/HDO handling profile.

8.5. Resources

This section gathers resource considerations on disk usage and processing time of the simulation.

8.5.1. Disk usage

Jacobian calculations can use a lot of disk space. The amount is governed by the number of different wave numbers, the number of molecules and the number of levels. So setting large values for these parameters leads to large disk space use.

8.5.2. Processing time

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 and 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. So, for Jacobians, it can take a long time to get results. The **hole convolution** option allows to speed up convolution computations when Jacobian calculation is selected.

8.6. Bug Reports

If the Reference Documentation does not address your question or problem, then you should send a message detailing the issue. If it is a problem with compiling the program, make sure you include information on your compiler and computer (mainly the type of operating system and brand of compiler). If the program crashes, you should also provide any relevant input files.

Bug reports and development suggestions will be actively pursued.

Please report bugs in 4A/OP to 4AOP@noveltis.fr.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

9. References

Reference ID	Title
[DR1]	SETG-MU-4AOP-011-TS_Ed1_Rev5_Advanced_Utilisation_manual.
[DR2]	NOV-FE-0035-NT-032_Scientific manual.
[DR3]	NOV-FE-0035-NT-031_Algorithm manual.
[DR4]	L. Chaumat, C. Standfuss, B. Tournier, E. Bernard, R. Armante and N. A. Scott, 2013: 4A/OP Reference Documentation, NOV-3049-NT-1178-v5.0, NOVELTIS, LMD/CNRS, CNES, 307 pp.
[DR5]	Jacquinet-Husson, N. *et al.*, 1999: The 1997 spectroscopic GEISA data bank. J. Quant. Spectrosc. Radiat. Transfer, 62, 205-254.
[DR6]	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.
[DR7]	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.
[DR8]	Jacquinet-Husson, N. *et al.*, 2009: The 2009 edition of the GEISA spectroscopic database. J. Quant. Spectrosc. Radiat. Transfer, 112, 2395-2445.
[DR9]	Jacquinet-Husson, N. *et al.*, 2016: The 2015 edition of the GEISA spectroscopic database. J. Quant. Spectrosc. Radiat. Transfer, 327, 31-72.
[DR10]	L. Chaumat, C. Standfuss, B. Tournier, E. Bernard, R. Armante and N. A. Scott, 2013: 4A/OP Reference Documentation, NOV-3049-NT-1178-v5.0, NOVELTIS, LMD/CNRS, CNES, 307 pp.
[DR11]	L. Chaumat, E. Dufour, B. Tournier and C. Standfuss, 2010 : Extension de 4A/OP au domaine du proche infrarouge, NOV-3810-NT-8259_v2.0.
[DR12]	L. Chaumat and C. Standfuss , 2010: Développement de 4A/OP-SWIR, NOV-3865-NT-9881.
[DR13]	L. Chaumat, N. Lautier, O. Lezeaux and C. Standfuss, 2011 : Développement algorithmique et scientifiques de 4A/OP SWIR, NOV-3933-NT-11703_v1.2.
[DR14]	B. Tournier, C. Standfuss and T. Lalanne, 2012: Développement de 4A/OP SWIR, NOV-7139-NT-1519_v1.1.
[DR15]	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.
[DR16]	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.
[DR17]	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.
[DR18]	O. Lezeaux, E. Bernard and B. Tournier, 2013: 4A/OP-subroutine updates from release 18 to release 24, NOV-7285-MM-3199.
[DR19]	Snyder, W.C., et al.,1998: Classification-based emissivity for land surface temperature measurement from space. Int. J. Remote Sensing, vol. 19, 2753-2774.
[DR20]	O'Dell, C. W., 2010: Acceleration of multiple-scattering, hyperspectral radiative transfer calculations via low-streams interpolation, J. Geophys. Res., 115, D10206, doi:10.1029/2009JD012803
[DR21]	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

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

	Multi-spectral Satellite Radiance Measurements.J. Appl. Meteor. Climatol., Vol. 47, 108-123.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales

10. Acronyms

- **4A**: Automatized Atmospheric Absorptions Atlas;
- **4OP-sub 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.

© Documentation CNES – LMD (CNRS/ENS) – NOVELTIS – Thales