RTTOV v14 Graphical User Interface

Pascale Roquet, Sonia Péré,
Jean-Marie Lalande, Jérôme Vidot

Météo-France

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 7 September 2021, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, DWD and Météo France.

Change record			
Version	Date	Author / changed by	Remarks
1	06/09/2016	P. Roquet	First version
2	02/05/2017	P. Roquet	note about matplotlib version
3	07/03/2018	P. Roquet	Updates for version 12.2
4	11/02/2019	P. Roquet	Updates for version 12.3
5	20/02/20	P. Roquet	Updates for version 13.0
6	18/09/20	P. Roquet	Note about wxpython installation
7	19/10/20	J. Hocking	Updates after DRR
8	20/09/21	P. Roquet	Clarification about 1D-VAR window input parameters
9	16/11/21	P. Roquet	Updates after DRR
10	15/11/22	P. Roquet	Updates for version 13.2
11	15/09/23	S.Péré	Updates for RTTOV14 + use of pyrttov wrapper
12	17/10/24	S.Péré	Updates after Met Office review
13	28/11/24	S.Péré	Updates after DRR

Table of contents

1. Introduction

This guide provides instructions on installing and utilizing the graphical user interface for RTTOV, known as RTTOV GUI. This interface allows users to tune atmospheric profiles, execute RTTOV for a specific instrument to generate radiances and brightness temperatures, and promptly visualize the results.

With RTTOV GUI, the user will be able to:

- load and modify an atmospheric profile (gas content, aerosol content, hydrometeor content)
- change surface parameters and options
- load an emissivity and a BRDF atlas
- run the direct model and display the results (simulated radiances, brightness temperatures and, if computed, reflectances).
- run the Jacobian model (RTTOV-K) and display the results (derived T and gas content and Tskin only)
- perform an 1DVAR retrieval

However, there are certain limitations to RTTOV GUI:

- It cannot execute PC-RTTOV (direct and Jacobian model), which is not available in RTTOV wrapper.
- It cannot run the adjoint model (AD) or tangent linear model (TL).
- Users cannot input or modify hydrometeor or aerosol optical parameters profile (options rttov_hydro_opt_param and rttov_aer_opt_param).
- Users cannot input or modify cloud liquid water content for non-scattering MW simulations.

For an optimal experience, it is strongly recommended to use a sufficiently large screen (at least 1400 x 900), as RTTOV GUI generates numerous windows, some of which may be large.

Users are also encouraged to consult the RTTOV User Guide for additional guidance.

1.1. Installation

1.1.1. Python environment

The GUI runs on any system compatible with the RTTOV wrapper (see user guide) but additionally requires the following Python packages to be installed:

- python3 : minimum version 3.9

- wxpython : minimum version 4.2.0

- numpy minimum version : 1.24.2 (https://numpy.org/)
- matplotlib with backend_wxagg : minimum version 3.7.0 (http://matplotlib.org/)
- h5py : minimum version 3 .7.0 (http://www.h5py.org/)
- pypubsub : minimun version 4.0.3 (https://pypi.org/project/PyPubSub/)
- attrdict : minimum version 2.0.1 (needed by wxpython 4.2.0)

The most straightforward method for installing the python packages is to create a dedicated python environment with anaconda. Subsequently, install all the required python packages using the file named “rttovguienv.conda”.

You can use the script `install_python_for_gui.sh` to execute the installation of the python conda environment (python 3.9). This script will download the miniconda installer, install python, and create a python environment named “rttovgui” in a directory named “miniconda3” within your HOME directory. Additionally, the script will modify your “.bashrc” file to load python environment named “rttovgui”.

1.1.2. Compilation

After activating the python environment, compile RTTOV using the “rttov_compile.sh” script, following the instructions outlined in the RTTOV v14 users guide. When prompted, answer “y” to indicate that you want RTTOV GUI to be compiled. Note that conda usually typically includes its own versions of the HDF5, libraries (located, for instance, under ~/miniconda3/lib/). It is recommended to compile RTTOV against the conda version of these libraries to prevent any potential mismatches in library versions.

1.1.3. Starting the GUI

Modify the gui/rttov_gui.env configuration file (see next section : RTTOV GUI User Manual). Then you can start the RTTOV GUI by executing the following commands:

```
source ./rttov_gui.env
./rttovgui
```

1.1.4. MacOs

For MacOS, it is recommanded to include in your ~/.bash_profile:

```
export LC_ALL=en_US.UTF-8
export LANG=en_US.UTF-8
```

to prevent issues with matplotlib (unknown local:UTF-8).

2. RTTOV GUI User Manual

2.1. Configuration files

The rttov_gui.env file contains mandatory environment variables which are used by RTTOV GUI. This file must be customized to your specific installation.

Example of rttov_gui.env file:

```
# RTTOV GUI Environment
#
# Mandatory variables:
# --------------------

# RTTOV GUI installation directory

RTTOV_GUI_PREFIX=# absolute path to the rttov/gui directory e.g.
/home/user/rttov14/gui
export RTTOV_GUI_PREFIX

PATH=${RTTOV_GUI_PREFIX}:$PATH
export PATH

export PYTHONPATH=${RTTOV_GUI_PREFIX}:${RTTOV_GUI_PREFIX}/../wrapper/:$PYTHONPAT
H



# Optional environment variables : (the defaults are usually OK)
# ------------------------------

# Directory for rttov emissivity and BRDF atlases: this should be the directory
# containing the emis_data/ and brdf_data/ directories which hold the atlas data
sets

RTTOV_GUI_EMISS_DIR=${RTTOV_GUI_PREFIX}/../
export RTTOV_GUI_EMISS_DIR


# Working directory (for rttov gui temporary files)

RTTOV_GUI_WRK_DIR=$HOME/.rttov_14
export RTTOV_GUI_WRK_DIR


# Default directory for rttov coefficient files

RTTOV_GUI_COEFF_DIR=${RTTOV_GUI_PREFIX}/../rtcoef_rttov14
export RTTOV_GUI_COEFF_DIR
```

```
# Default directory for profile files

RTTOV_GUI_PROFILE_DIR=${RTTOV_GUI_PREFIX}/../rttov_test/profile-datasets-hdf
export RTTOV_GUI_PROFILE_DIR

system_info=`uname -s`
if [ ${system_info} = "Linux" ]
then
  export cmdpython=python
else
  echo "It seems that Linux is not your system..."
  echo "for python we will use pythonw"
  echo "correct this if necessary"
  export cmdpython=pythonw
fi

# ------------------------------------------------------------------------------

# If you installed the Python dependencies using conda the following can be igno
red.
# Otherwise you may need to specify the location of your wxPython installation:

# WXPYTHON_PREFIX=${RTTOV_DIR}/gui # Directory where wxPython is installed


# PYTHONPATH and LD_LIBRARY_PATH: these must point to your wxPython installation

# # #PYTHONPATH=${WXPYTHON_PREFIX}/lib/python3.9/site-packages/wx-4.2-gtk3:${WXP
YTHON_PREFIX}/lib/python3.9/site-packages/:${PYTHONPATH}
# export PYTHONPATH

# LD_LIBRARY_PATH=${WXPYTHON_PREFIX}/wxpython/lib:$LD_LIBRARY_PATH
# export LD_LIBRARY_PATH# RTTOV GUI Environment
#
```

2.2. *File created by the RTTOV GUI*

The RTTOV GUI software creates HDF5 files in the designated working directory, which is specified by the RTTOV_GUI_WRK_DIR environment variable (default is ~/.rttov_14). It is important to note that when running multiple instances of RTTOV GUI simultaneously, distinct working directory must be used for each instance

All these files are formatted in HDF5. You can examine them using software like HDFview, for instance. Exporting data to text file is also feasible using the h5dump command.

The files generated include:

- profile.h5, which contains the profile (gas concentration, aerosol types and concentration, hydrometeor types and concentration), surface information and RTTOV options.
- surface.h5, which contains information about surface emissivity/BRDF used as input, computed by RTTOV or modified by the user.

- radr.h5, which contains radiances computed by RTTOV.
- trns.h5, which contains transmittances computed by RTTOV.
- kmat.h5, which contains the Jacobian matrix computed by RTTOV_K.
- tmpFileErr.log, the error log file from the last RTTOV execution.
- tmpFileOut.log, the output log file from the last RTTOV execution.
- rttovgui.log, which is the RTTOV GUI log file.

2.3. *Starting the GUI*

As mentioned in section 1.1, RTTOV GUI can be started using the following commands:

```
source ./rttov_gui.env
./rttovgui&
```

The RTTOV main window is opened (*cf.* Figure 2.3.1). It provides the capability to load coefficient files via the RTTOV menu, open a profile using the File menu, adjusts options, profile and surface parameters as needed (accessible through the dedicated windows via the Windows menu), and execute the RTTOV direct model using the RTTOV menu. The main window also displays the application log.

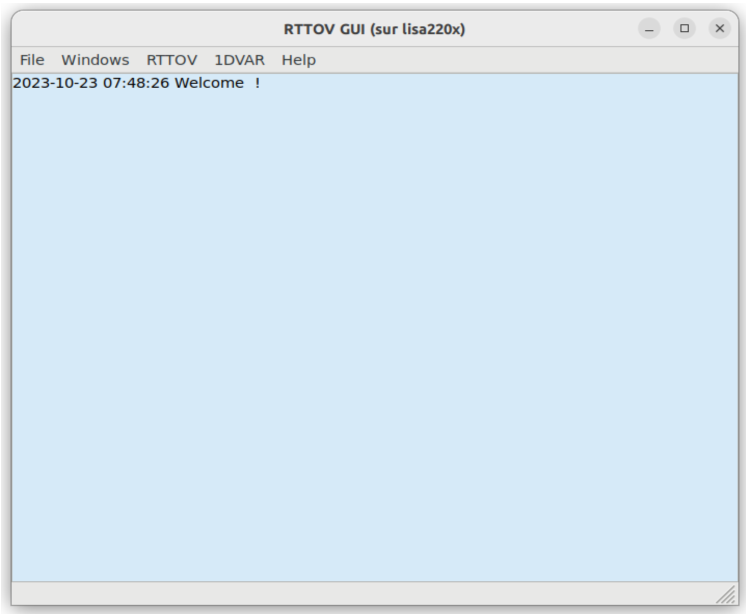


Figure 2.3.1: main window

2.4. *Loading RTTOV coefficient files*

The initial step involves selecting an instrument (i.e., IASI, AMSU, Meteosat, etc.) by picking up a coefficient file through the RTTOV menu “Load coefficient” (cf., Figure 2.4.1).

The RTTOV coefficients file is mandatory as this file contains the informations needed for computing the atmospheric transmittances from the atmospheric profiles. If you intend to incorporate aerosols, you must select an aertable optical property file (CAMS or OPAC). Similarly, if you want to run RTTOV with hydrometeors, an hydrotable optical property file is required. In cases where hydrometeor simulation are desired, some instruments support the MFASIS Neural Network capability to perform visible cloud scattering. This is done by selecting the corresponding SOLAR_SOLVER in the option windows.

It is imperative that all coefficient files are compatible, an error message will be displayed if there is any incompatibility.

After finalizing your selections, you can load the coefficient files either by clicking the "Load" button or by accessing the "File" menu and choosing the "Load coefficients" option.

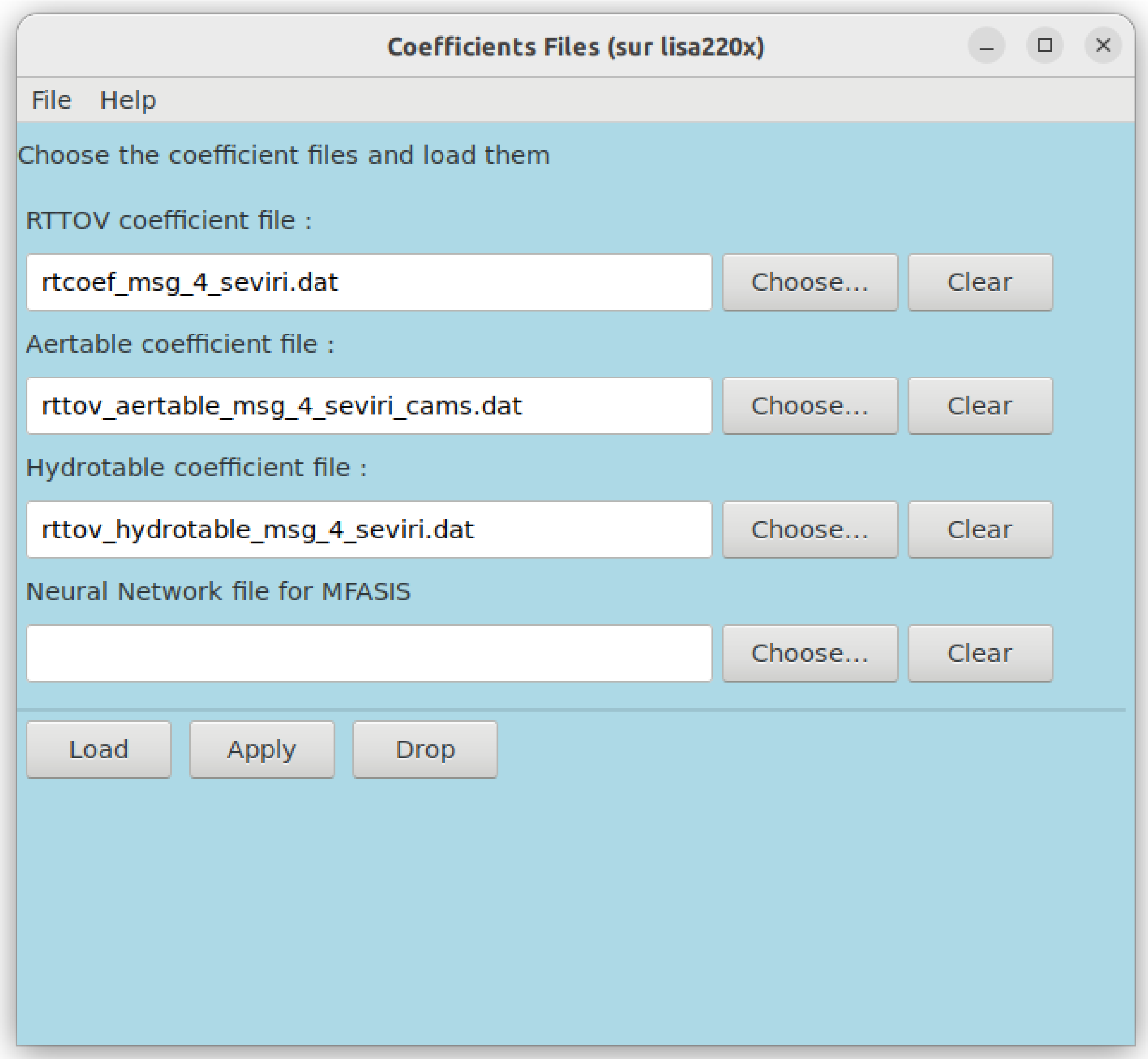


Figure 2.4.1: coefficient files window

After RTTOV successfully loads the coefficient files, an information window is displayed (cf. Figure 2.4.2). The log in the main window provides details extracted from the coefficient files, such as the wave numbers and the reference temperatures of the instrument.

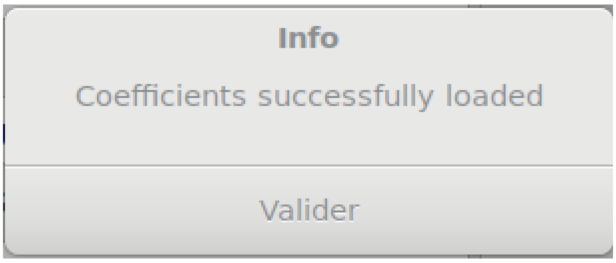


Figure 2.4.2: successful coefficient files loading message

2.5. *Open a Profile*

To open an atmospheric profile go to “Open Profile” option in the “File” menu of the main window (cf. Figure 2.5.1). Select the desired file. If it contains several profiles, you will be prompted to specify the profile number (cf. Figure 2.5.2)

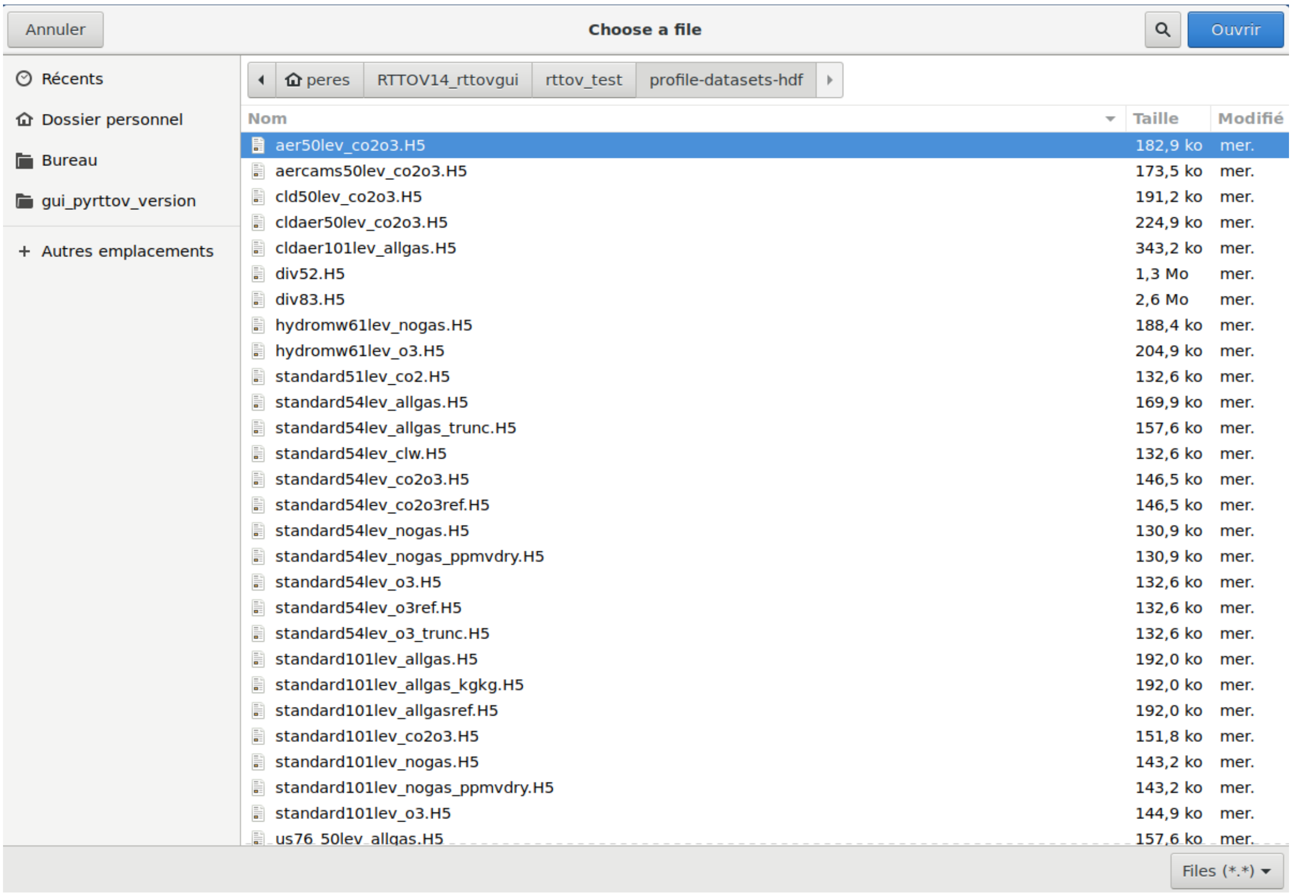


Figure 2.5.1: select a profile

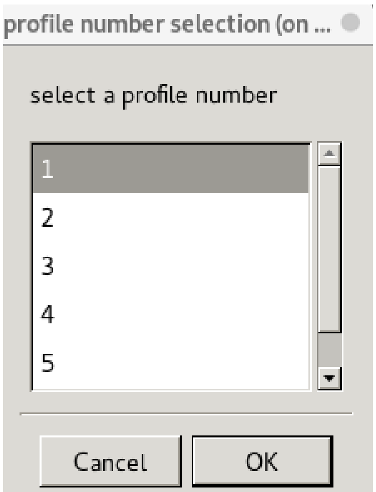


Figure 2.5.2: selection window

2.6. Adjusting the options

You can adjust the RTTOV options using the “Options Editor” window. Choose “Options Editor Window” from the main window (Figure 2.6.1).

The modifiable options encompass a subset of the options structure of RTTOV, as outlined in the RTTOV v14 User Guide (Annex J). The help menu provides additional information. Keep in mind that some options may be unavailable based on the loaded coefficient files or the profile content. The options are organised by themes and presented on tabs in the Options Editor window. The themes includes:

General configuration options:

“ApplyRegLimits”: If set to true, profile values beyond the limits of the training data for gas optical depth regression are truncated to the limits before being input to the regression.

“GasOpdepCalc”: If set to false, it deactivates the gas optical depth calculation, and simulations proceed with the assumption of zero optical depths. This feature is designed for users who are specifically investigating scattering scenarios.

“ADKBT” determines the units of Jacobians for thermal channels (wavelengths > 3 microns): if true Jacobians are calculated in terms of Brightness Temperature, if false Jacobians are calculated in terms of radiance.

“ADKRefl” determines the units of Jacobians for non-thermal channels (wavelengths < 3 microns): if true Jacobians are calculated in terms of reflectance, if false Jacobians are calculated in terms of radiance.

General RT options:

Each “data” option (“O3Data”, “CO2Data”, “N2OData”, “COData”, “CH4Data” and “SO2Data”) is enabled only when the RTTOV coefficient file has been computed for the corresponding gas (refer to the RTTOV User Guide for a complete description of the four sets of predictors available as coefficient files).

"Solar" option is deactivated if the standard coefficient file lacks solar channels.

"RayleighMaxWavelength": specify maximum channel wavelength for which to include Rayleigh scattering calculations.

"RayleightMinPressure": Rayleigh scattering is ignored in layers at pressures below this.

"RayleighSingleScatt" enables Rayleigh single scattering parametrisation.

"NlteCorrection" enables NLTE radiance correction.

"Refraction" enables atmospheric refraction in simulation.

"PlaneParallel" enables strict plane parallel geometry.

"RadDownLinTau" use linear-in-tau (true) or layer-average (false) for downwelling layer emission in radiative transfer equation integration.

"UseT2m" enables use of the 2m temperature profile variable.

"UseQ2m" enables use of the 2m temperature profile variable.

Scattering options:

"Hydrometeors" option becomes active when a hydrotable coefficient file is loaded and the profile contains hydrometeor concentration profiles.

"Aerosols" option becomes active upon loading an aertable optical property file and the profile contains aerosol content (CAMS or OPAC aerosols).

"BaranIceVersion": selection of Baran's ice microphysical-to-optical properties.

"IcePolarisation": polarised scattering factor for ice hydrometeors in the context of empirical polarisation mw_pol_mode (only relevant for MW sensors).

"MwPolMode": select treatment of polarised scattering in MW.

"ChouTangFactor": empirical factor used in Tang modification to Chou-scaling.

"ChouTangMod": enables Tang modification to Chou-scaling.

"ThermalSolver" is active only if Hydrometeors or Aerosols option has been selected.

"SolarSolver" is active only if Hydrometeors or Aerosols option has been selected as well as Solar option.

The "Dom" options, including "DomNstreams," "DomAccuracy," and "DomOpdepThreshold," are activated when either the ThermalSolver or SolarSover options (or both) are enabled, and the DOM model is selected for either one or both of them. The "RayleighMultiScatt" option is available only when the SolarSolver model is DOM, and a V13 predictor coefficient file is loaded along with a corresponding hydrotable/aertable optical property file.

Interpolation options:

"InterpMode": choice of vertical interpolation mode.

"PressureGradients": if set to true, the profile p_half and p are active variables in the TL/AD/K.

Cloud overlap options:

"OverlapParam": select cloud overlap scheme.

"ColThreshold": ignore cloud columns with weights below this value

"TwoColMaxFracMaxP": this is the maximum pressure applied when determining the effective cloud fraction and only applicable to the "TwoColMaxFrac" cloud overlap option.

Surface-related options:

"SolarSeaReflModel": select solar reflectance model for ocean surfaces.

"IrSeaEmisModel": select IR emissivity model for ocean surface.

"MwSeaEmisModel": select MW emissivity model for ocean surfaces.

"UseFoamFraction": set ocean foam fraction (the foam fraction has to be modified in the surface window).This option only applies to FASTEM.

"Lambertian": enable surfaces to be treated as a configurable mixture of Lambertian and specular reflectors.

"LambertianFixedAngle": use fixed effective zenith angle when computing Lambertian downwelling radiances.

After finalizing your options selection, save them by clicking the "Apply" button.

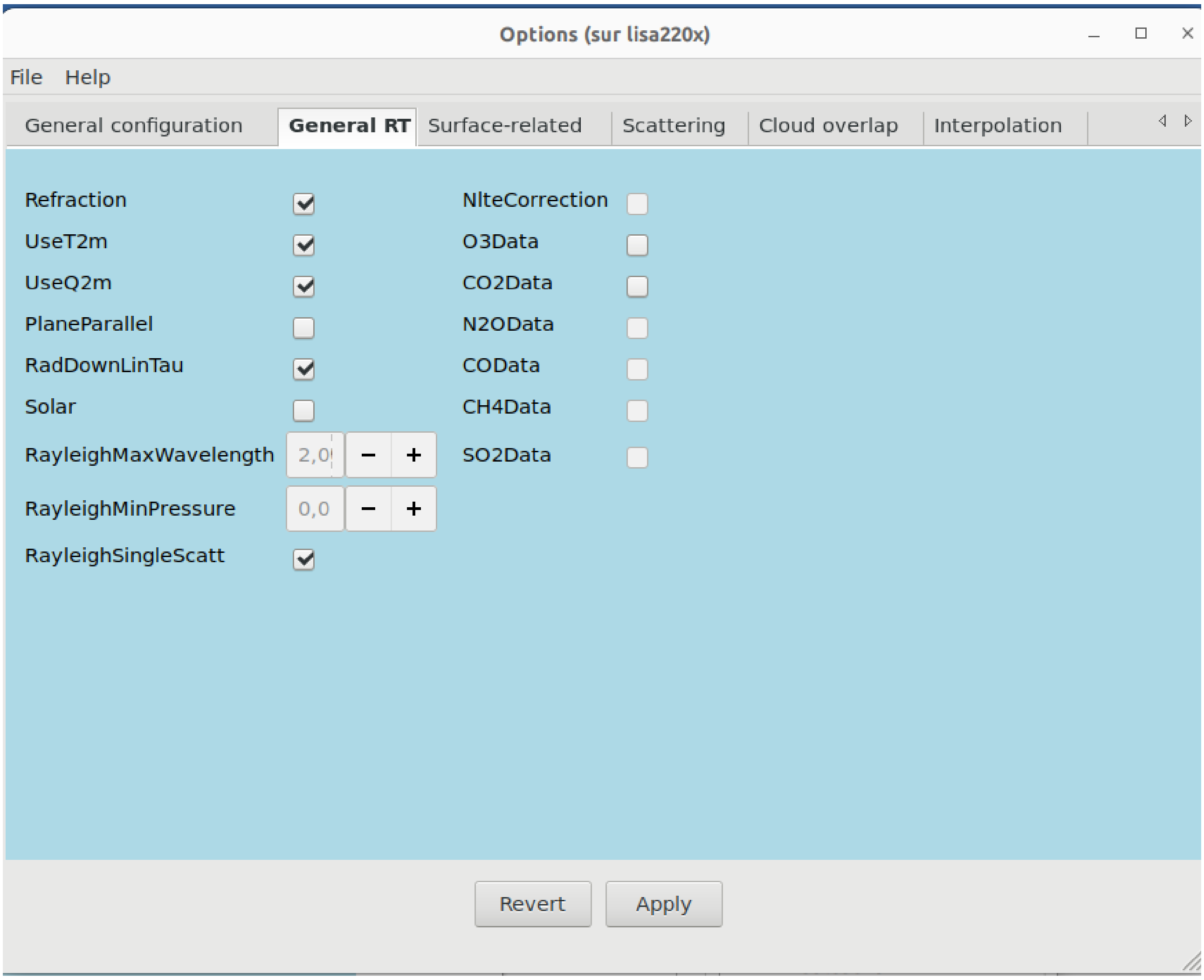


Figure 2.6.1: Options Editor window

2.7. Adjusting the atmospheric profile

You can edit the profile using the Profile Editor Window. Choose "Profile Editor Window" from the "Windows" menu in the main window.

Example of the profile editor window showing atmospheric gas profiles is illustrated in Figure 2.7.1.

Examples of hydrometeors and aerosols profiles are shown in Figures 2.7.2 and 2.7.3 respectively. Aerosol components and hydrometeor types are described in the RTTOV user guide.

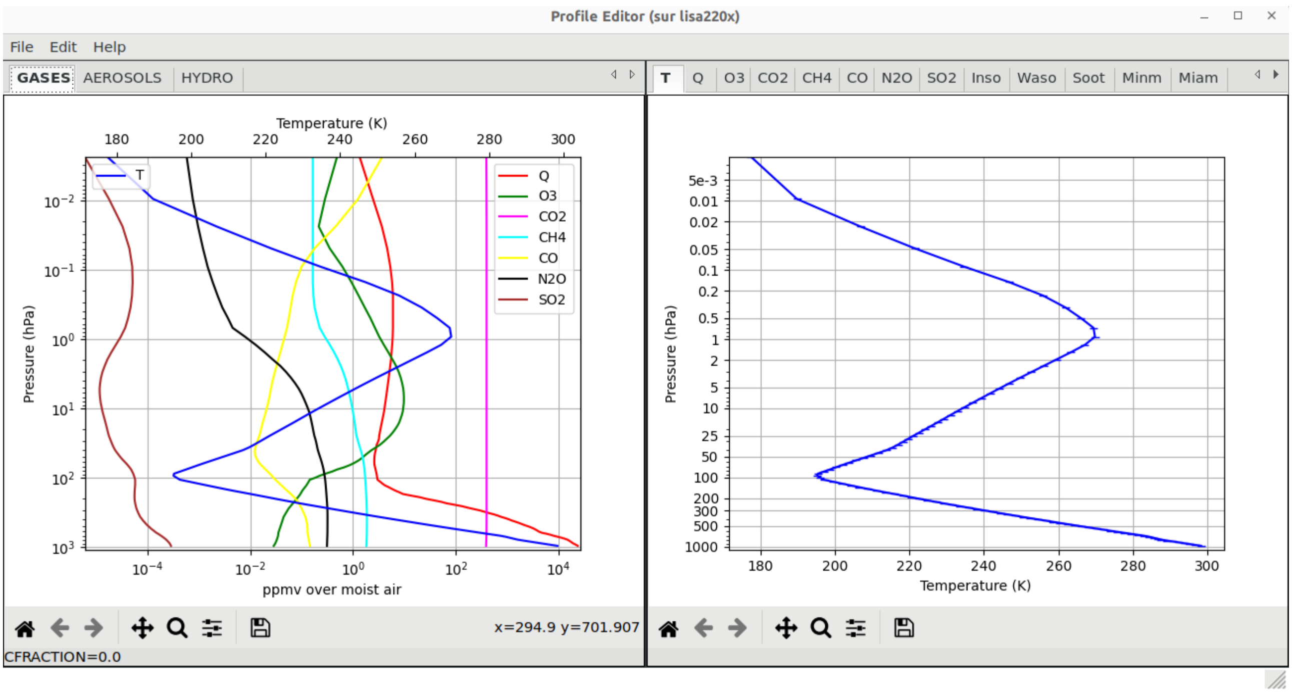


Figure 2.7.1: Profile Editor window for gases

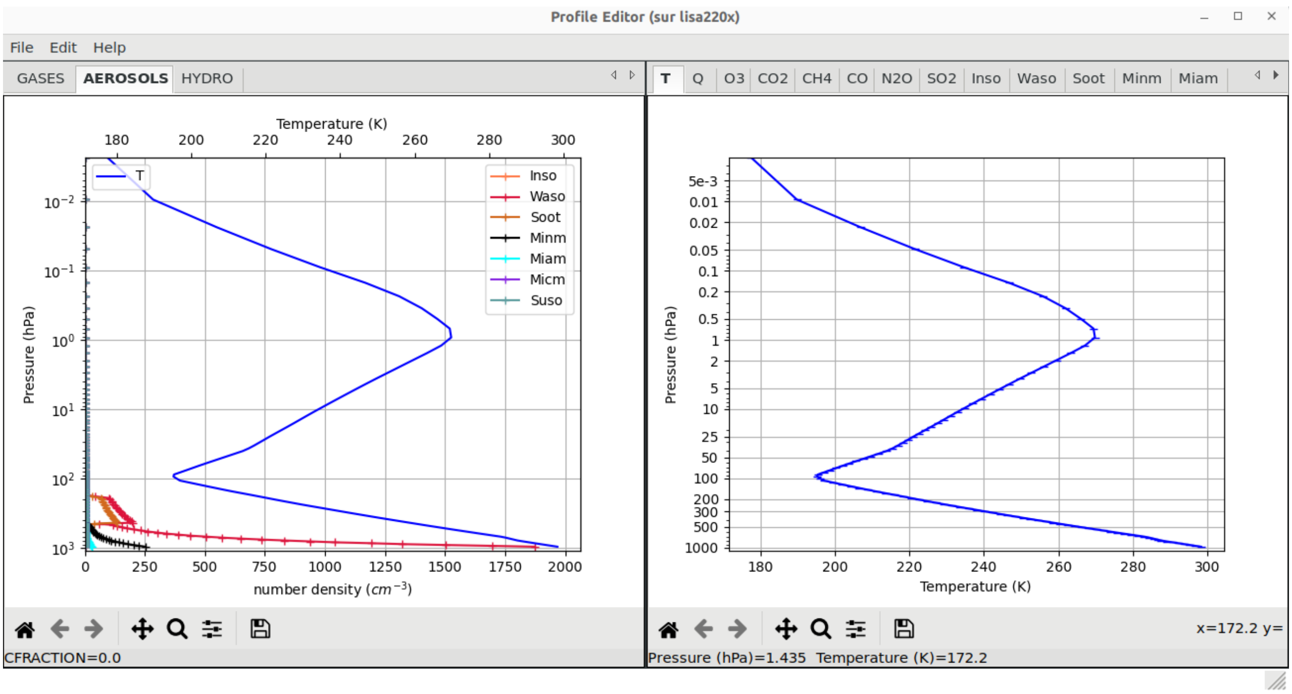


Figure 2.7.2: Profile Editor window for aerosols

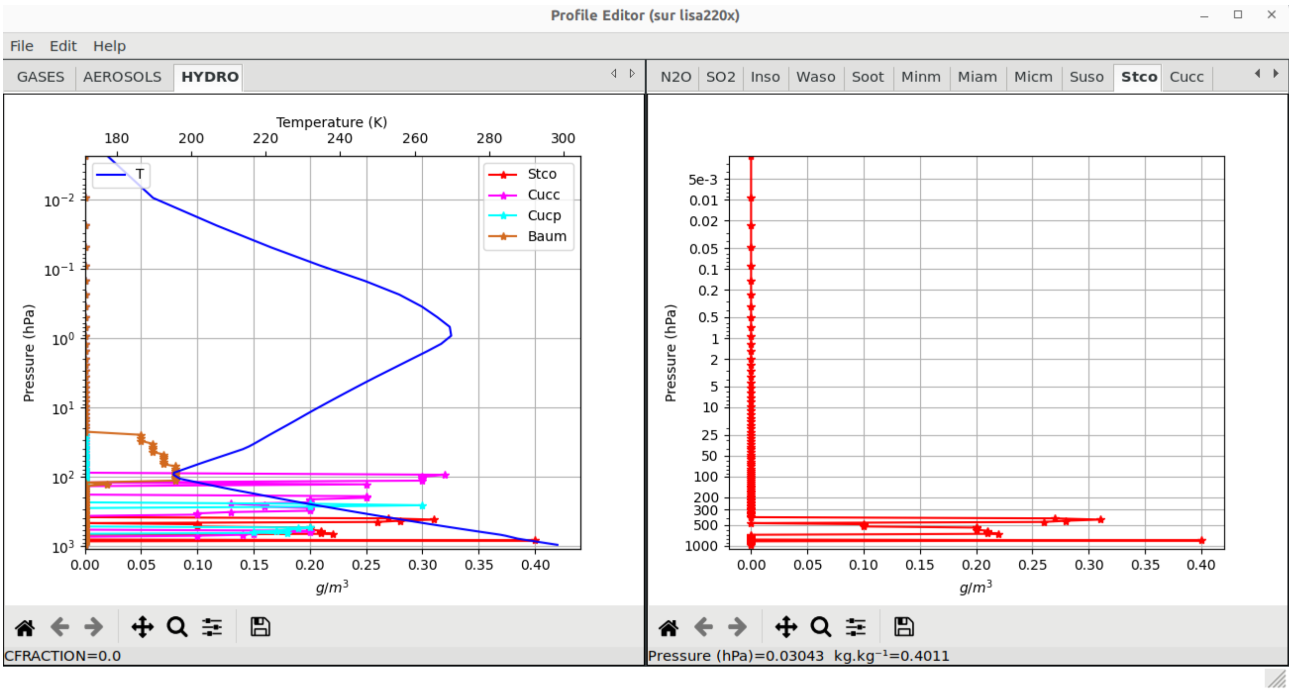


Figure 2.7.3: Profile Editor window for hydrometeor (profile used: cldaer101lev_allgas)

The window's left-hand panel displays profiles of atmospheric components categorized as Gases, Aerosols, and Hydro. The Temperature profile is consistently depicted across all three panel types. On the right-hand panel, individual profiles for different components are presented. To manually adjust a profile, click within the panel. Choose and view a component profile on the right panel, then click either the middle or right button to make modifications. The selected point on a profile's pressure level or layer will be relocated to the new value, with updates reflected in the left panel. Utilize the zoom feature in the matplotlib toolbar to zoom in or out :

matplotlib toolbar

Finally, apply your changes or save the profile for the next RTTOV execution.

Edit menu functionalities:

Undo: undo the last modification of the curve.
Redo: redo the last modification of the curve.

Insert: for aerosols or hydrometeors only, a click moves or creates a new point.
Remove: for aerosols or hydrometeors only, a click removes the nearest point of a layer.
Change profile values: apply scale*value + offset on the profile values between P min and P max.
Edit X axis: this menu item allows to modify the X axis bounds.
Add gas: with this menu item it is possible to add a gas.
Remove gas: with this menu item, it is possible to remove a gas.
Add aerosol: with this menu item, it is possible to add an aerosol model.
Remove aerosol: with this menu item, it is possible to remove an aerosol model.
Add hydrometeor: with this menu item, it is possible to add a hydrometeor.
Remove hydrometeor: with this menu item, it is possible to remove a hydrometeor.
Replace aerosol by clim: with this menu item, it is possible to replace all aerosols by an aerosol climatology. The user is allowed to choose between different types of climatology (cf. Figure 2.7.4).
Remove aerosol: with this menu item, it is possible to remove an aerosol model.
Add hydrometeor: with this menu item, it is possible to add a hydrometeor.
Remove hydrometeor: with this menu item, it is possible to remove a hydrometeor.

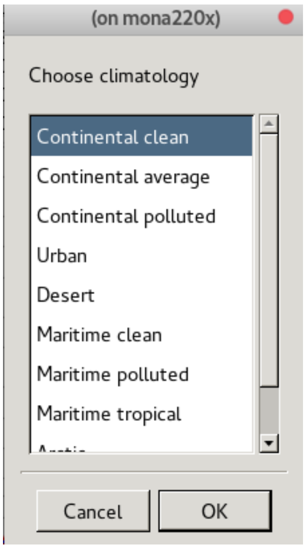


Figure 2.7.4: Climatology choices for aerosols (see RTTOV User Guide for full description)

2.8. *Editing the surface*

The “Surface Editor Windows” enables to adjust the surface parameters of the profile. Select the “Surface Editor Window” option from the windows menu located in the main window (cf. Figure 2.8.1).

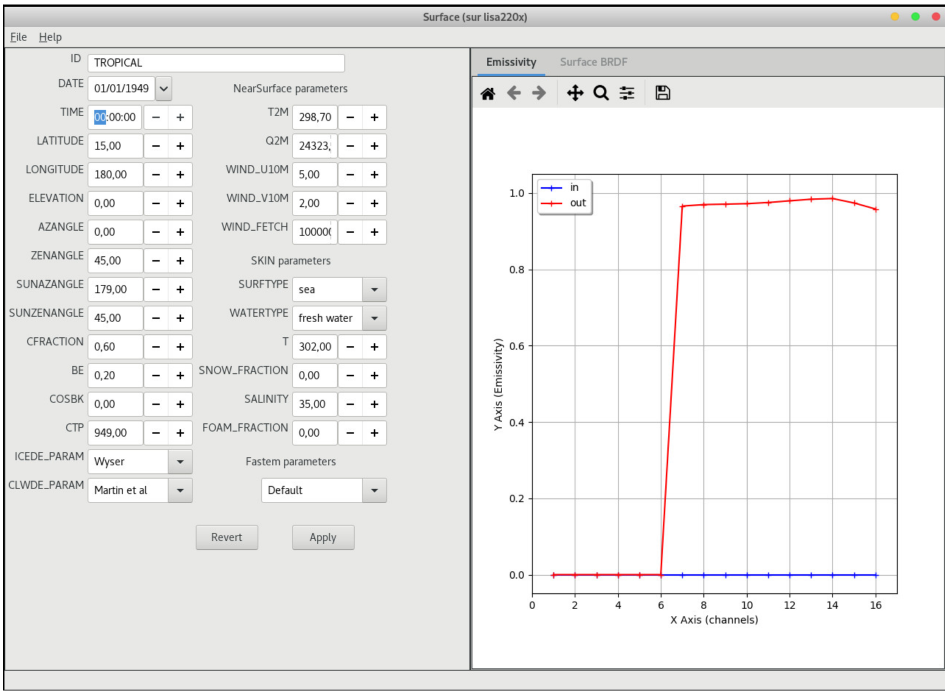
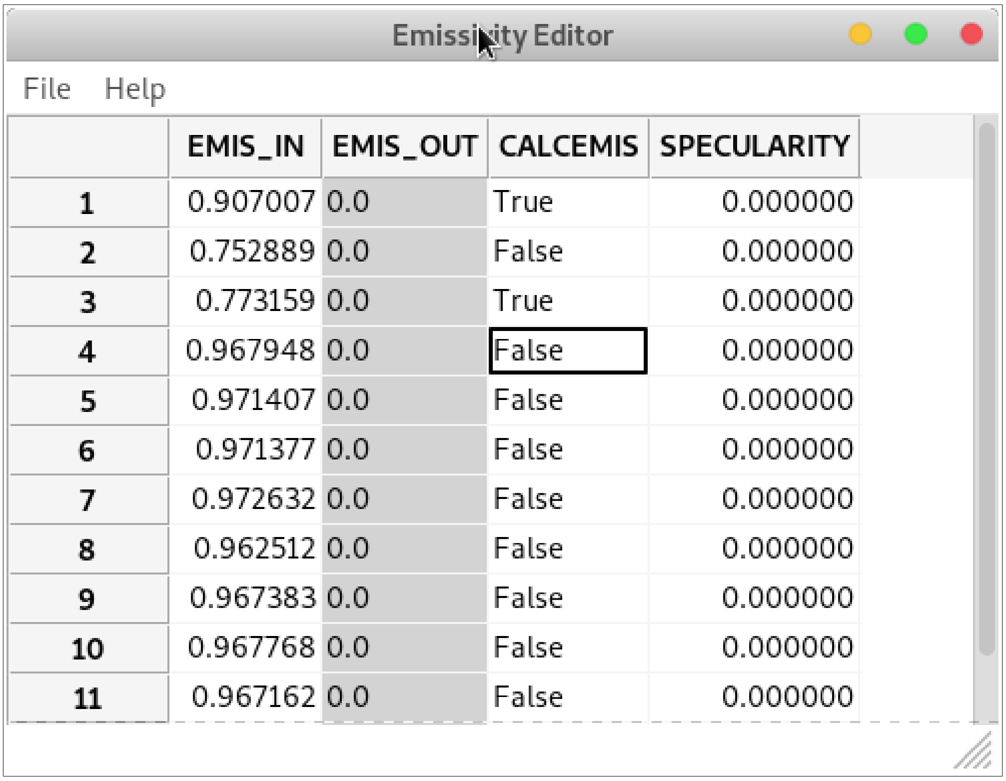


Figure 2.8.1: Surface Editor Window

Within the Surface Editor Window's File Menu, you can either load an atlas or make channel-specific adjustments to emissivity or BRDF values (cf. Figure 2.8.2). Computed emissivity/BRDF values are displayed in red on the right-hand panel, while input values are shown in blue. If an atlas is loaded, the RTTOV model does not calculate emissivity or BRDF values.

Please note that the CFRACTION parameter (cloud fraction for simple cloud) may have a non-zero value for many profiles supplied within RTTOV. It is used for the simple RTTOV cloud scheme (IR/VIS only): see RTTOV User Guide.



	EMIS_IN	EMIS_OUT	CALCEMIS	SPECULARITY
1	0.907007	0.0	True	0.000000
2	0.752889	0.0	False	0.000000
3	0.773159	0.0	True	0.000000
4	0.967948	0.0	False	0.000000
5	0.971407	0.0	False	0.000000
6	0.971377	0.0	False	0.000000
7	0.972632	0.0	False	0.000000
8	0.962512	0.0	False	0.000000
9	0.967383	0.0	False	0.000000
10	0.967768	0.0	False	0.000000
11	0.967162	0.0	False	0.000000

Figure 2.8.2: Edit Input Emissivity Values.

2.9. Running RTTOV Direct Model and working with the Radiance Window

You can now execute the RTTOV direct model by following these steps:

1. choose the RTTOV menu in the main window

2. select "Run direct." This action saves and replaces the currently edited profile before running the direct model, with the profile being stored in the RTTOV GUI working directory (~/.rttov_14/prof.h5) before each run.

Upon a successful run, a new window emerges—the radiance window, presenting the outcome of the RTTOV direct model. This window features three tabs: byRun, byChannel, and OnePlot. By default, the OnePlot tab is activated, displaying the brightness temperatures simulation (cf. Figure 2.9.1).

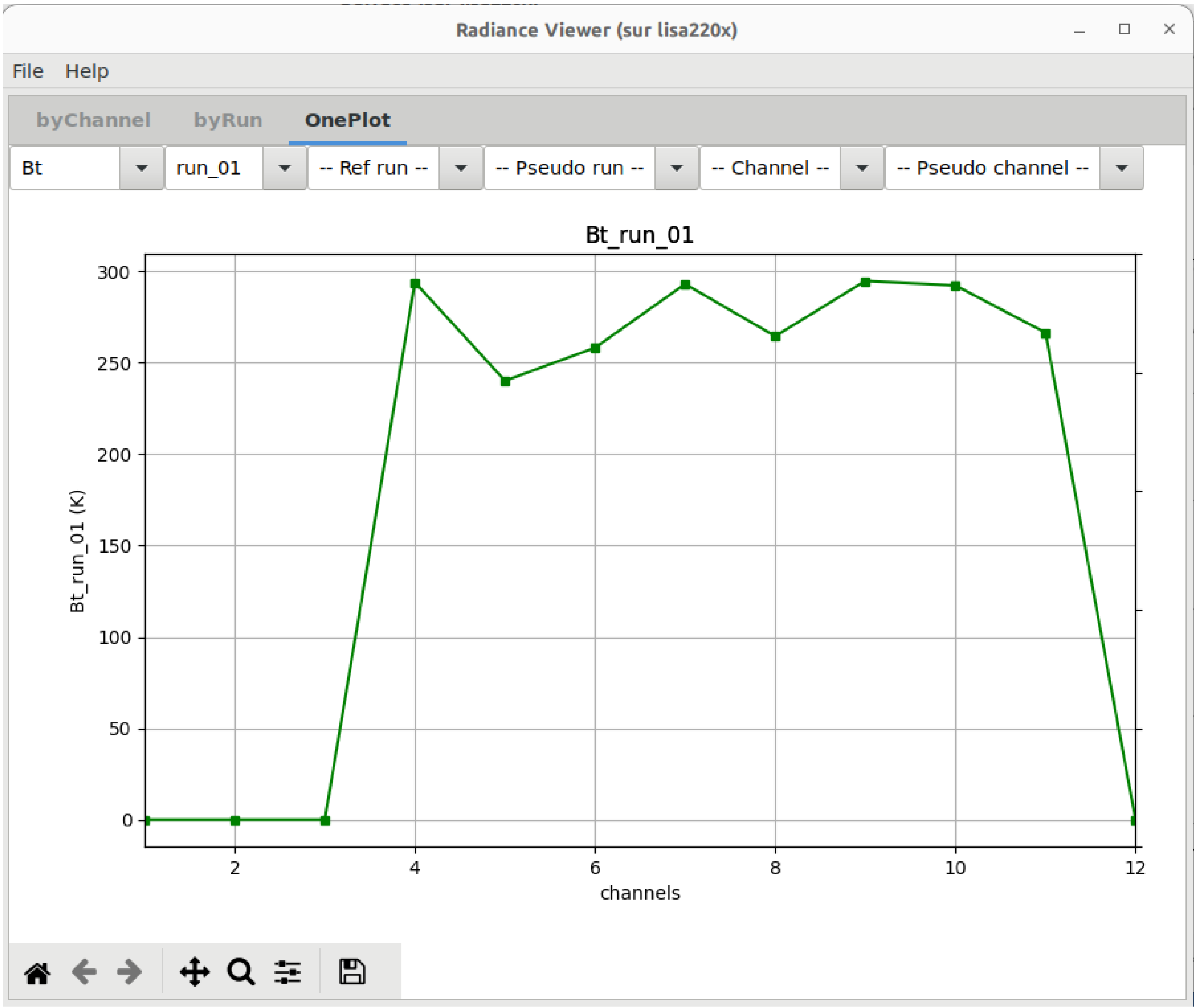


Figure 2.9.1: Radiance window OnePlot tab

The plots which can be displayed in the radiance window are:

- RadTotal: top of atmosphere radiance incorporating any cloud or hydrometeors present, either via the simple cloud scheme or full hydrometeor scattering, and including aerosols if present [mW.$(cm^{-1})^{-1}$.sr^{-1}.m^{-2}]
- RadClear: Clear-sky top of atmosphere radiance, including aerosols if present [mW. $(cm^{-1})^{-1}$. sr^{-1}. m^{-2}]
- Bt: Brightness temperature equivalent to total radiance [K]
- BtClear: Brightness temperature equivalent to clear radiance [K]
- Refl: Reflectance calculated from total radiance [-]

- ReflClear: Reflectance calculated from clear radiance [-]

It is possible to choose different types of visualisation:
OnePlot: View of one plot (Figure 2.9.1)
In this view you can select:

- **The parameter**: Bt, BtClear, RadTotal, RadClear, Refl, ReflClear: choose a plot type
- **-- Run --**: to choose a run
- **-- ref run:** to choose a reference run (for differences and comparison)
- -- **Pseudo Run --**: difference (minus) or overlay (versus) with reference run
- **-- Channel --**: to choose and display only one channel
- **-- Pseudo Channel --:** to display user's made combinations of channels

byRun: For each channel, view of all plots with all runs values
In this view you can select:

- **-- Channel --**: to choose a channel
- **-- Pseudo Channel --:** for channels combination: in this text field, user can write a formula (see the example displayed in the text field), which will become a “pseudo channel”.

byChannel: For each run, view of all charts with all channels values (Figure 2.9.2)
In this view you can select:

- **-- Run --**: to choose a run (needs at least 2 different runs)
- **-- Pseudo Run --**: to see the difference between with reference run (needs at least 2 different runs) (Figure 2.9.3)

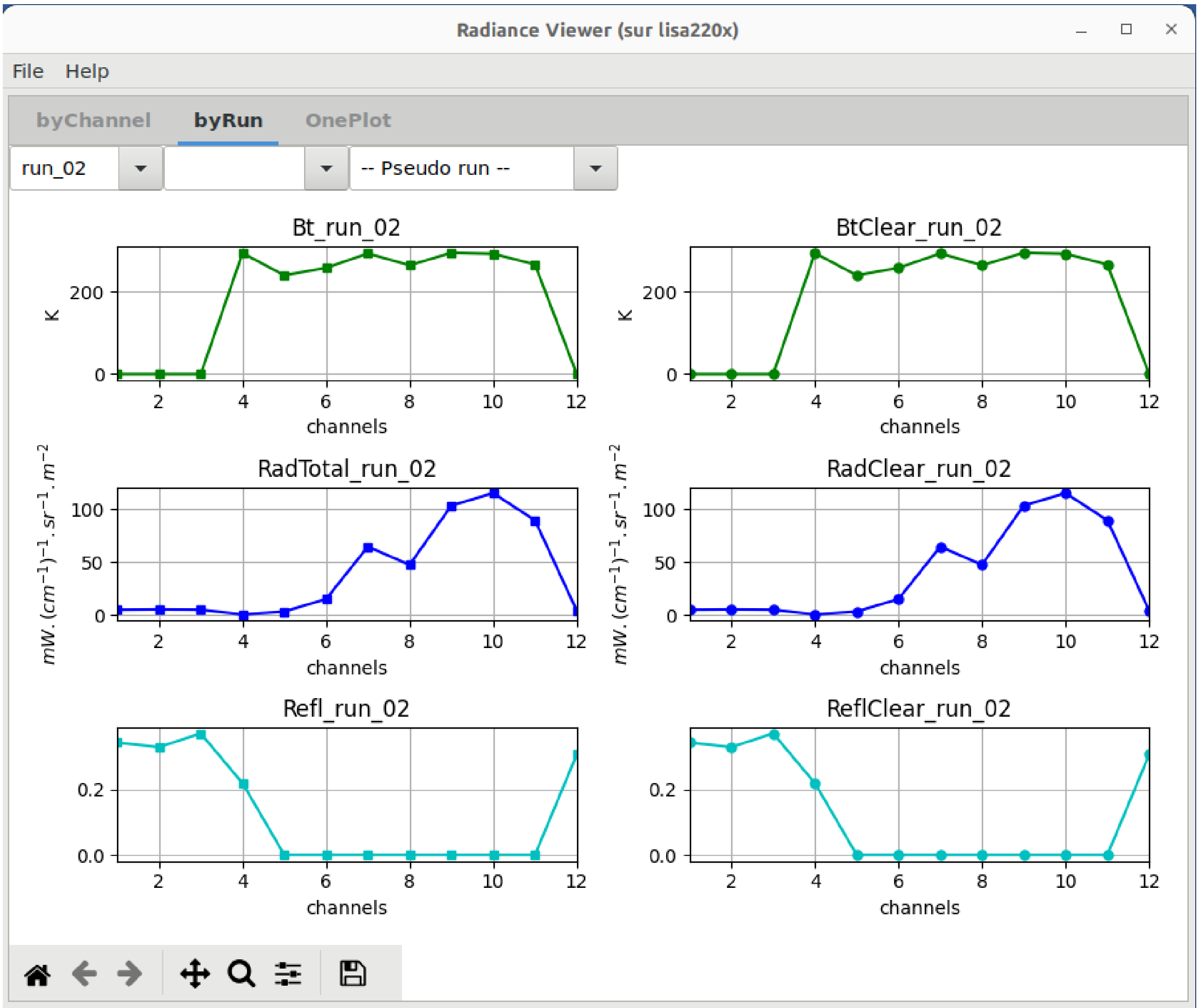


Figure 2.9.2: Radiance window showing differences between two runs of the direct model.

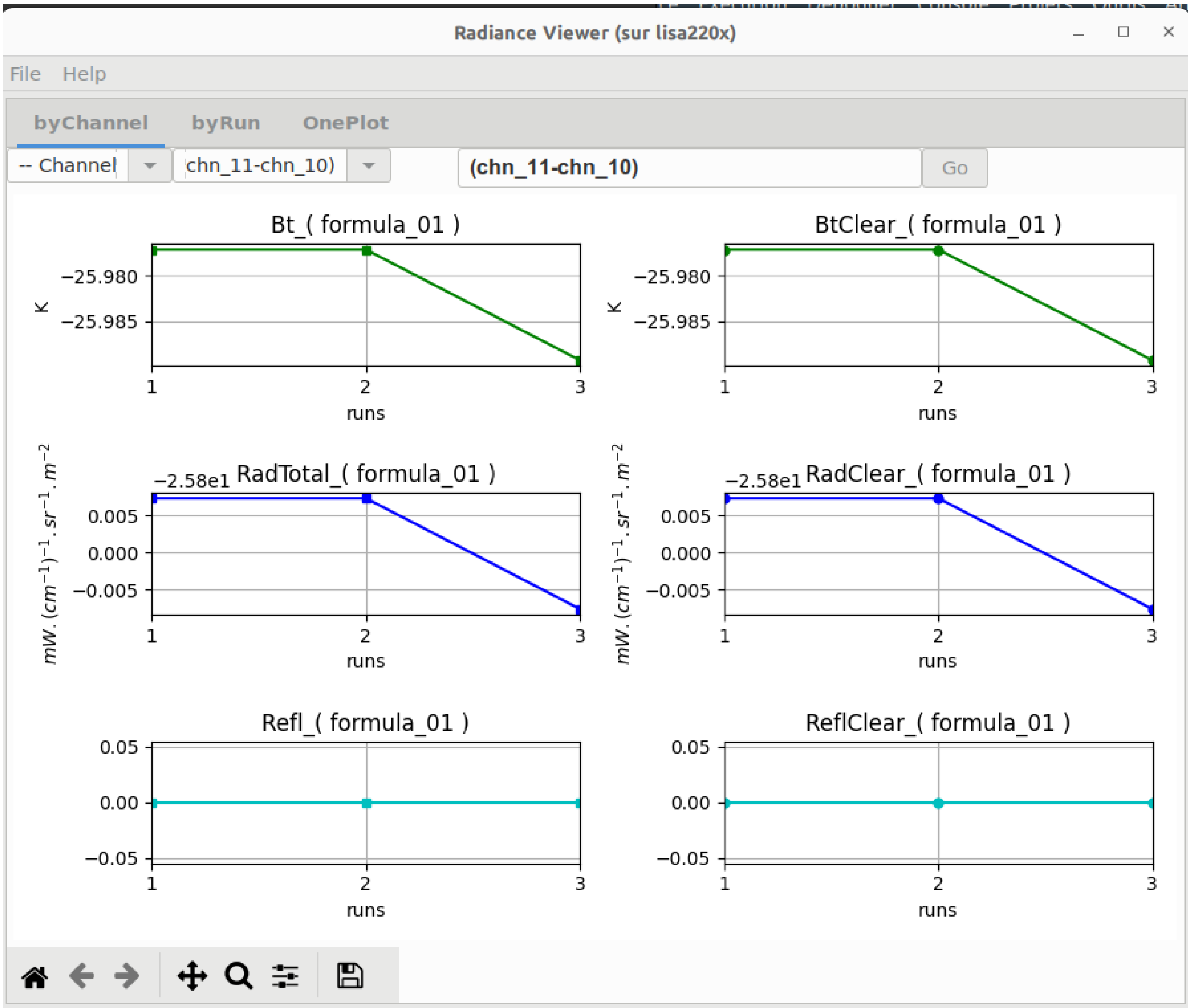


Figure 2.9.3: Radiance window (by Run view with a computed pseudo channel)

When running the software for high-resolution sounders such as IASI, AIRS or CrIS, a spectrum is displayed (cf. Figure. 2.9.5).

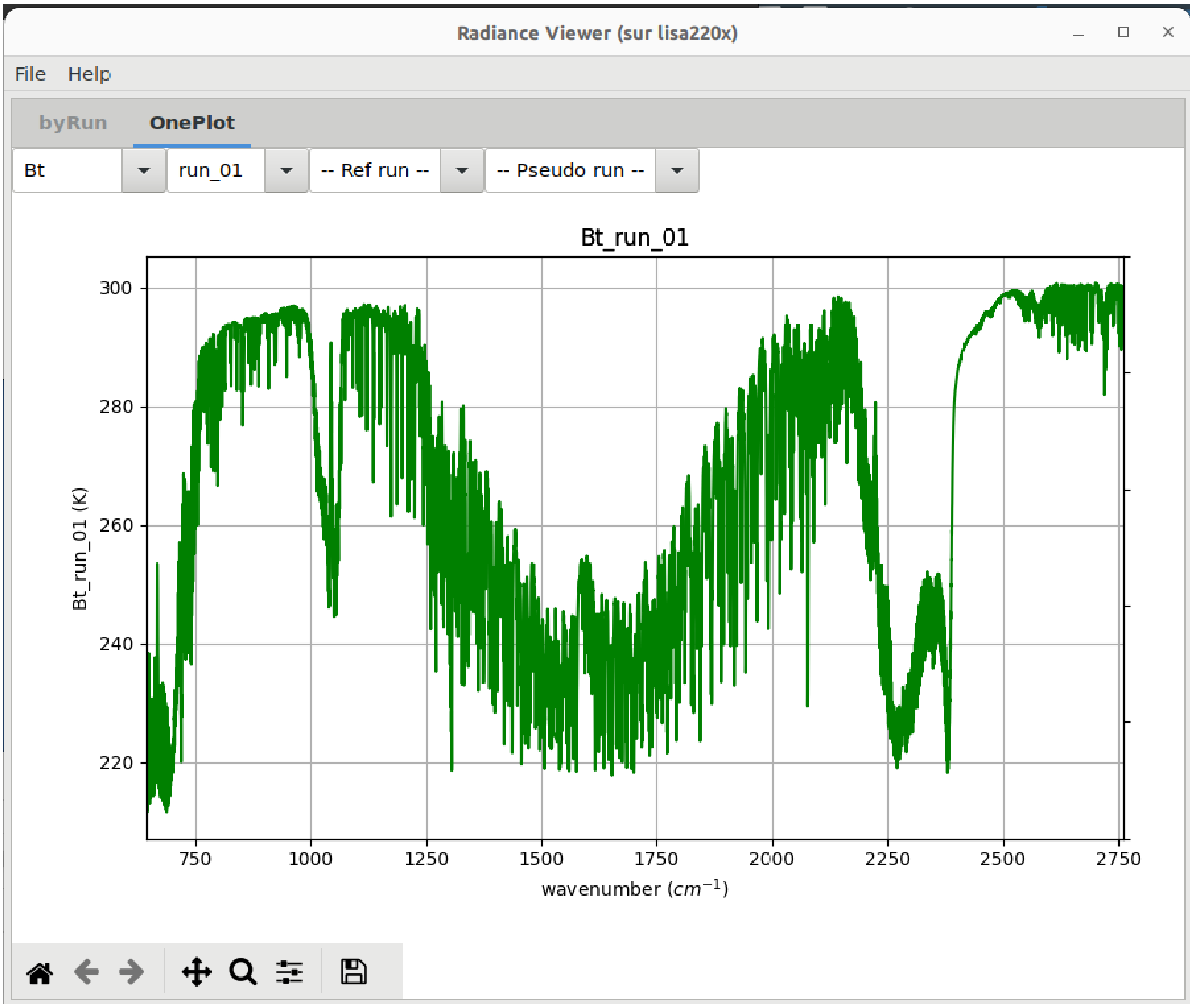


Figure 2.9.5: Radiance window for hyperspectral instruments

For any subsequent run you can select pseudo run such as “run_02 minus run _01” or “run_02 versus run_01” to see the spectrum differences (cf Figure. 2.9.6 spectrum difference between run 02 (with “Solar” option) and run 01(without “Solar” option) for IASI.

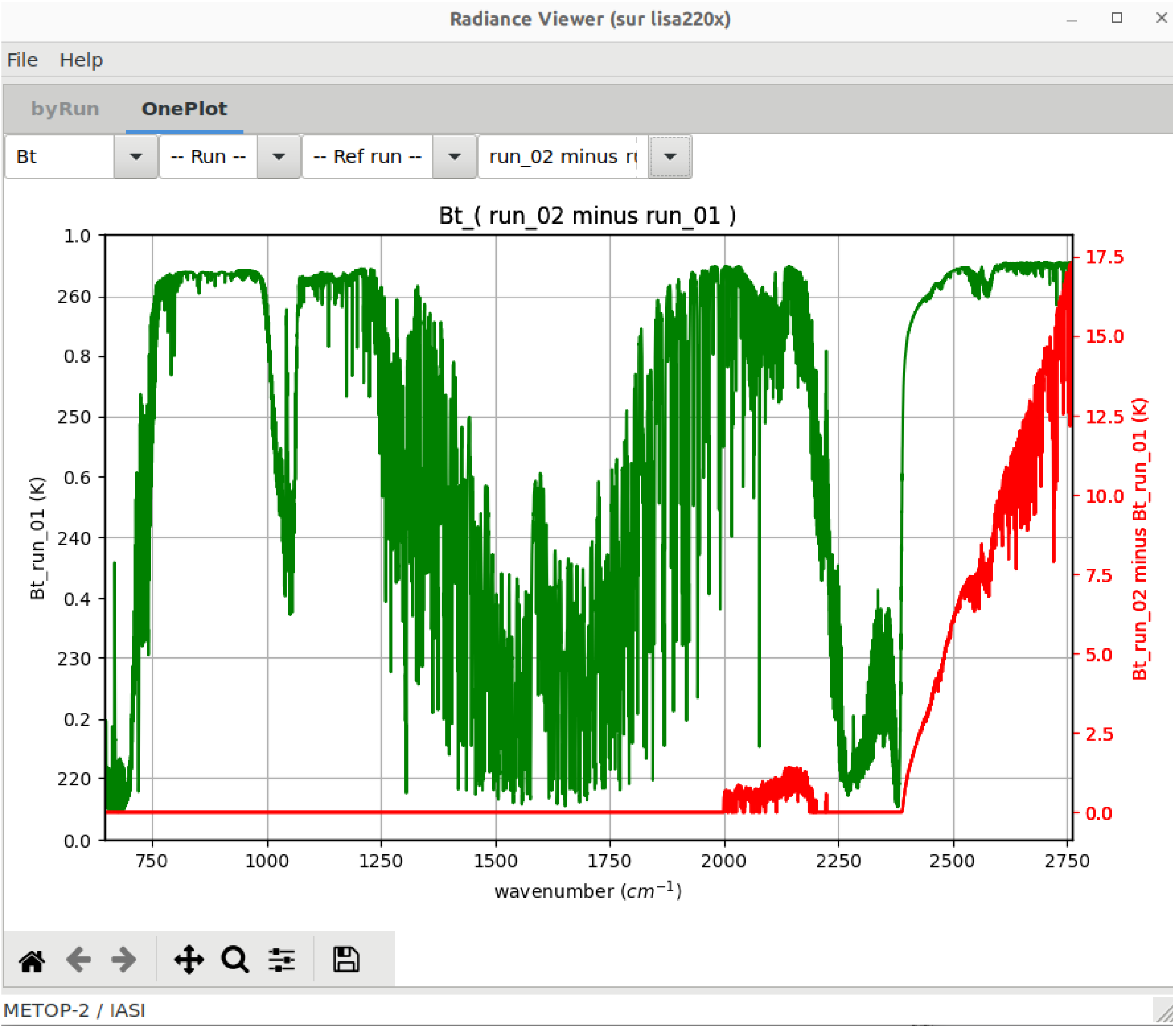


Figure 2.9.6: Radiance window for hyperspectral instruments: difference between 2 runs (with and without Solar option)

Remark:

If you aim to execute RTTOV with a different instrument, loading distinct coefficients file while having an active radiance window showing the outcomes of a prior RTTOV run, the radiance window becomes obsolete, and the GUI will automatically close it.

Radiance window command line functionality:

It is also possible to launch the radiance window separately from the RTTOV GUI using the radr.h5 file containing the results of a RTTOV run (this file is kept by default in the ~/.rttov_14 directory):

```
python rview/radianceframe.py radr.h5 [radr2.h5...]
```

Interactive navigation:

The toolbar at the bottom of the radiance window is the matplotlib toolbar which enables to zoom and navigate in the Figure.

2.10. *Running RTTOV K and working with the K-Matrix and the Kprofile windows.*

You can execute the Jacobian model of RTTOV by following these steps:

1. choose the RTTOV menu in the main window,
2. select "Run RTTOV K." This action saves and overwrites the profile, running the K model.

Upon successful completion, a new window, the K-Matrix Window, appears, displaying the result of the RTTOV K-Model (cf. Figure 2.10.1). With each RTTOV K run, a new window is generated.

The units of Jacobians depend on the “ADKBT” and “ADKRefl” options. For thermal channels (wavelengths > 3 microns), Jacobians are in terms of Brightness Temperature if ADKBT is true or radiance otherwise. For example, if ADKBT is true, then Jacobians for thermal channels are in units of $K.K^{-1}$ for temperature variables and $K.ppmv^{-1}$ or $K.(kg/kg)^{-1}$ for gases (depending on the profile gas_units). For non-thermal channels (wavelengths < 3 microns), Jacobians are in terms of reflectance if ADKRefl is true or radiance otherwise. For example, if ADKRefl is true, then Jacobians for non-thermal channels are in units of K^{-1} for temperature variables and $ppmv^{-1}$ or $(kg/kg)^{-1}$ for gases (depending on the profile gas_units).

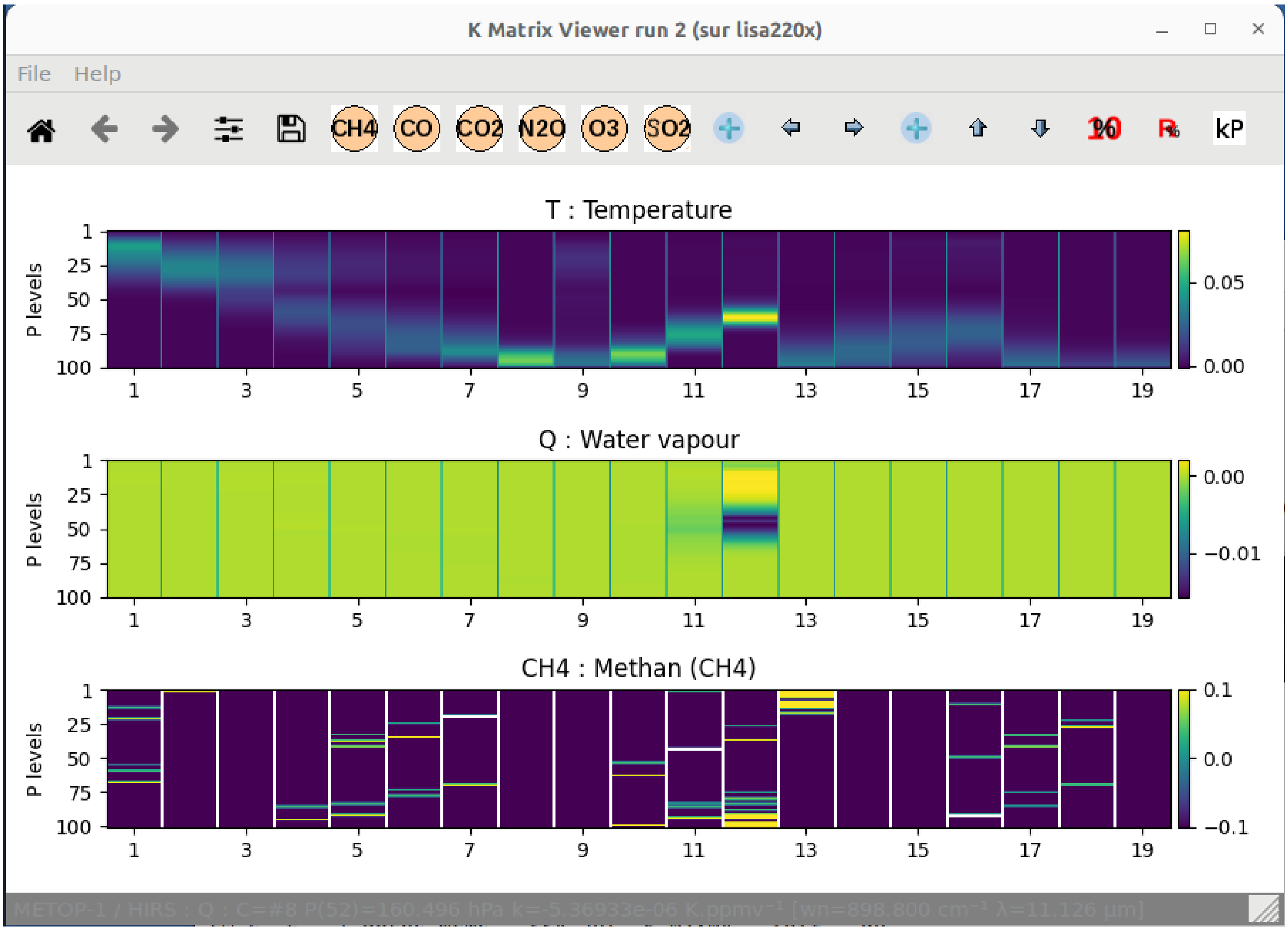


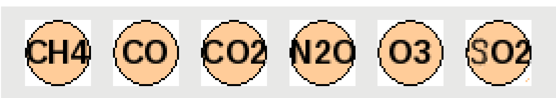
Figure 2.10.1: K-Matrix window

Move the mouse on a channel and enter P on your keyboard to display the K profile (Figure 2.10.2)

Icon Toolbars:

You can also use the Kp icon, in order to choose the channel of the K profile to be displayed.


This icon allows you to multiply by 0.1 all values.



These icons, if present, allow you to display the K matrix for the different gas.

The K-Profile windows enable the overlay of up to four K-profiles derived from four consecutive K-runs. In the illustrated example (refer to Figure 2.10.2), the influence of the vertical interpolation mode is evident. Run 1, utilizing mode 5 (Rochon/Log-lin WF), and run 2, employing mode 1 (Rochon/Rochon OD), showcase the impact on the computed K-profile for channel 3 of HIRS.

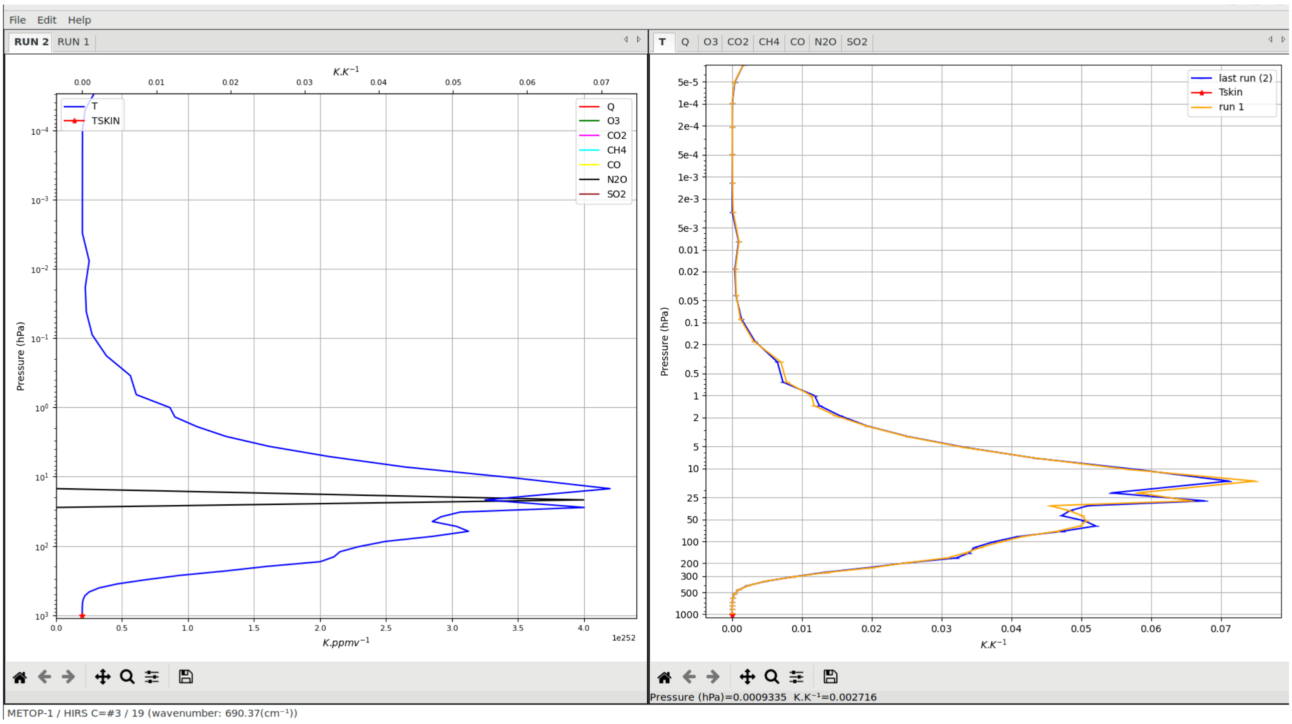


Figure 2.10.2: The K profile window

3. The 1DVAR functionality

This new functionality was added to RTTOV GUI for the 2015 NWP SAF-ECMWF satellite data assimilation course. Its purpose is to demonstrate to a student how the 1DVAR retrieval works.
The 1DVAR functionality uses Background and Observation error covariances matrices. These matrices are inherited from the NWP-SAF 1DVAR retrieval package and they can be customized if necessary: see the NWP-SAF 1DVAR retrieval package documentation. (https://nwp-saf.eumetsat.int/site/software/1d-var/).

In order to use the 1DVAR functionality you must use RTTOV GUI with a 54 levels profile, which

once opened from the file menu of the RTTOV GUI main window, will be considered as the background profile. It can be modified with the “profile editor” window.
The covariance observation error matrices come from the 1DVAR therefore the number of permitted instruments for the RTTOV GUI 1DVAR functionality is restricted: AMSUA, MHS, ATMS, HIRS, AIRS, CrIS, IASI, SSMIS. Choose a 54 levels coefficient file among these instruments.
With this 2 prerequisites, the “Configure 1DVAR” command of the “1DVAR” menu of the main window becomes available: select it.

Work now from the “1DVAR” window (Figure 3.1):

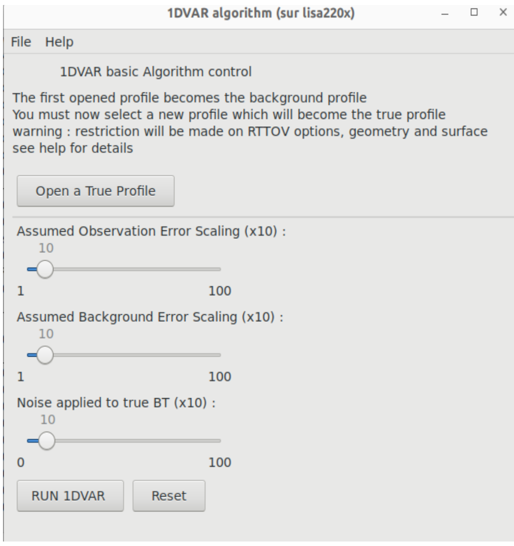


Figure 3.1: 1DVAR Window

1. Click on the “Open a true profile” in order to load what will be considered as the True profile (Xt): This true profile will be used to compute the simulated observed radiance, and to compare with the retrieved profile.
2. Change the value of the scaling factor (fb) for the background errors if necessary
3. Change the value of the scaling factor (fr) for the observation errors if necessary
4. Change the value of the maximum random noise if necessary
5. Click on the “RUN 1DVAR” button

The values of the 3 parameters that can be changed with the sliders are scaled by 10 on the interface to provide decimal precision: this means that if the user selects “85”, for example, with the slider, the actual value used by the algorithm will be 8.5.

The results are displayed in two windows:

- The retrieved profile window (Figure 3.2), the true profile is in black, the background in blue and the retrieval in red.

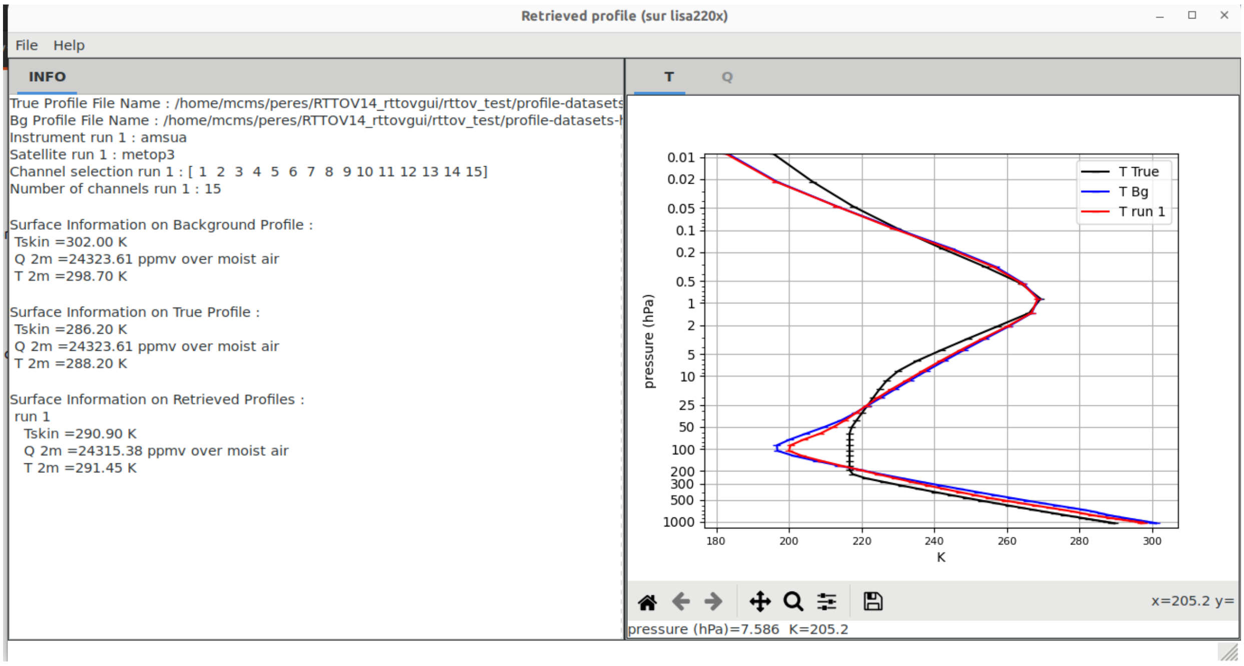


Figure 3.2: Retrieved profile window

- The brightness temperatures window (Figure 3.3), which displays the brightness temperatures obtained with the RTTOV direct model for the true and the background profiles, and the difference (True – Bg).

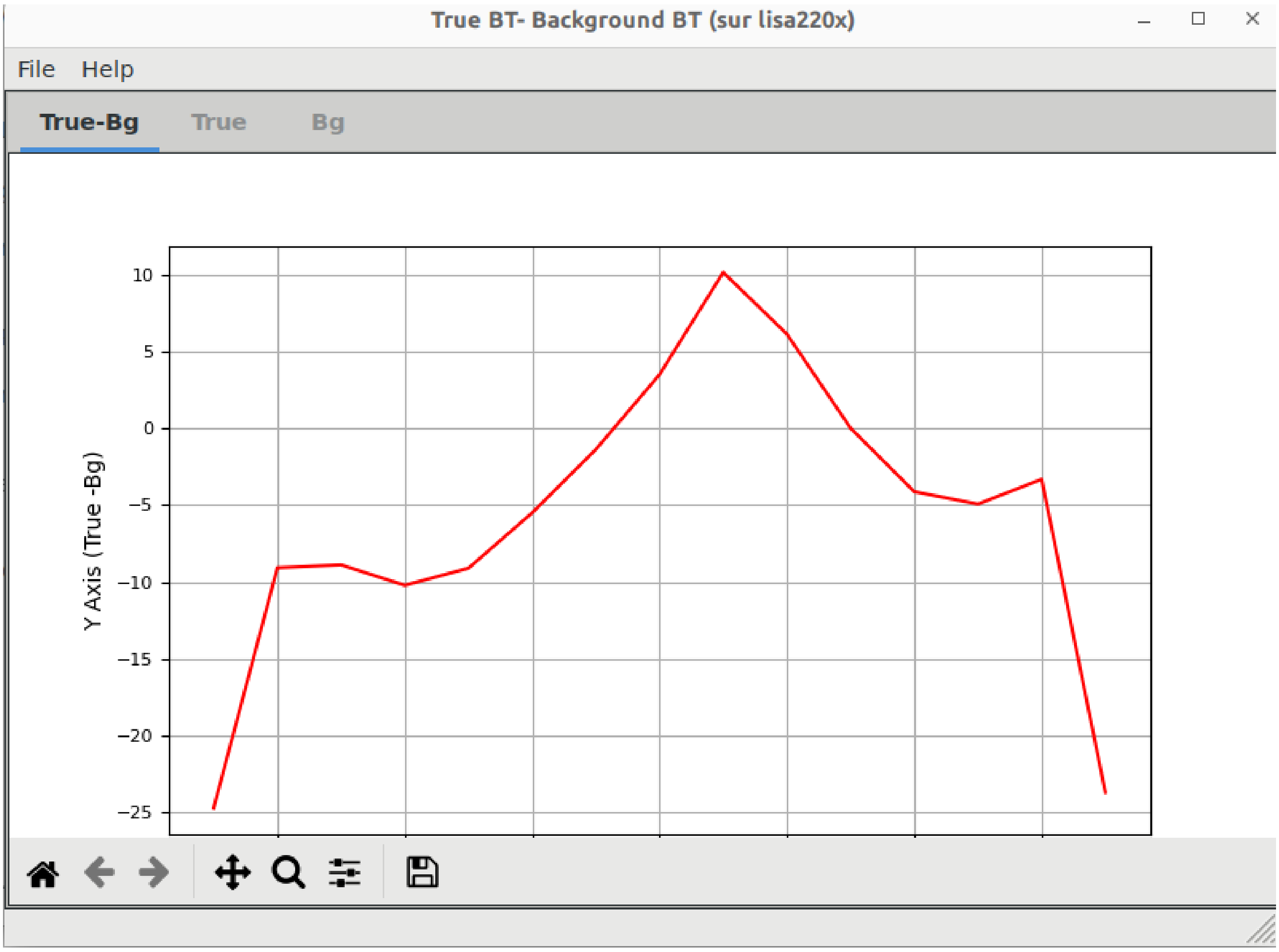


Figure 3.3: Brightness temperatures window

RTTOV GUI allows you to perform a channel selection on the instrument, this functionality can be very useful. Let us see a concrete case:
In order to visualize at which level a specific channel of an instrument may be useful for the retrieval process, you can run RTTOV-K for all channels of an instrument from the RTTOV menu of the main window. RTTOV GUI displays the K Matrix Window. This window shows the results of the Jacobian model for the different channels, for the temperature and for the gases concentrations. In this example (Figure 3.4) of a K matrix window for the METOP-3/AMSUA instrument, one can easily see that the influence of a slight modification of the temperature may be of some importance at the level where the Jacobian is the largest.

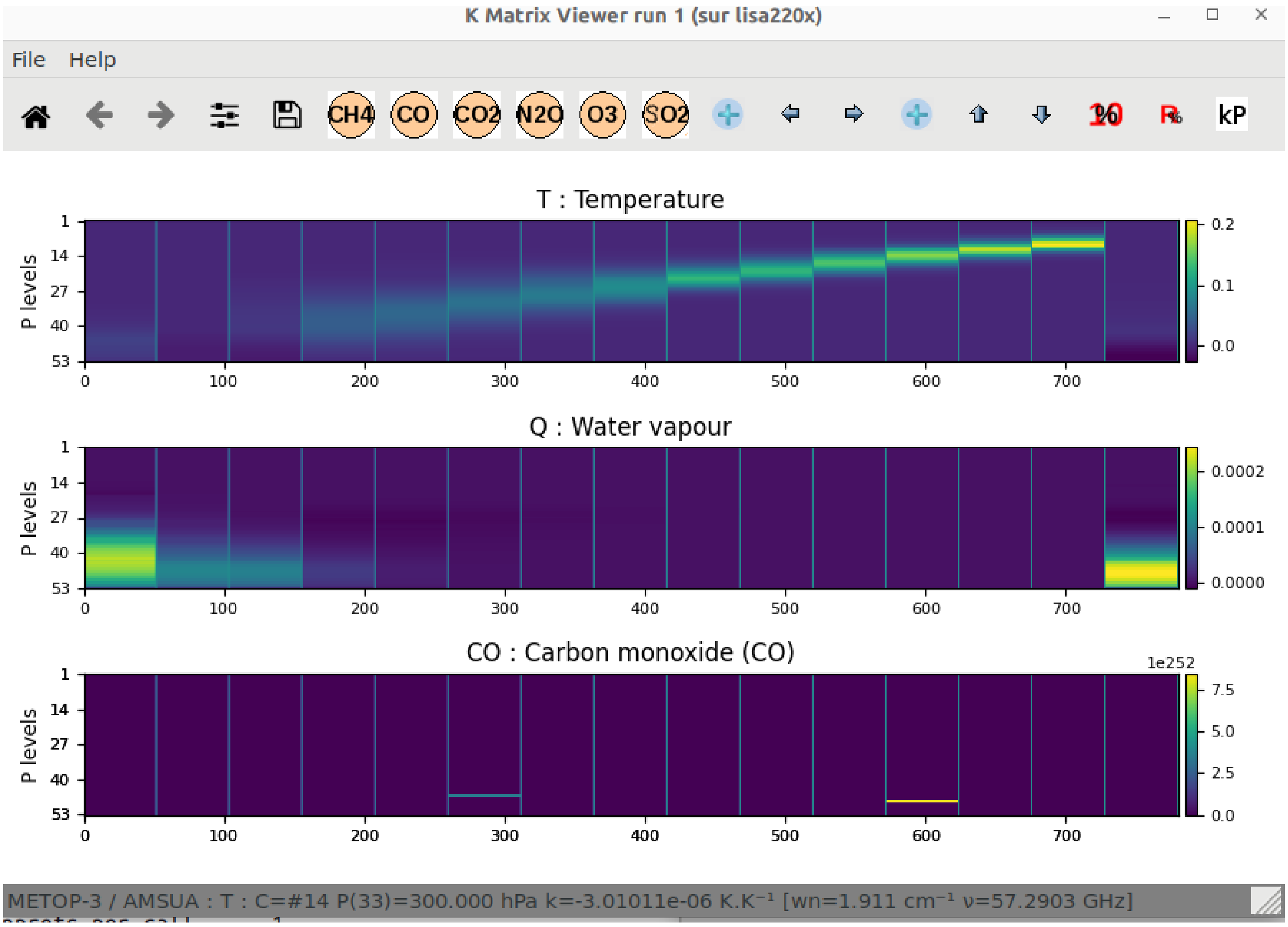


Figure 3.4: K Matrix window for the METOP-3/AMSUA instrument

From the K matrix viewer, the user can, with the “kP button”, select a channel in order to display the K profile. In Figure 3.5, on the K profile window for channel 12 of METOP-3/AMSUA, the peak around 10hPa on the Jacobian of temperature curve is clearly visible: this channel will add some important information in the retrieval process at this level.

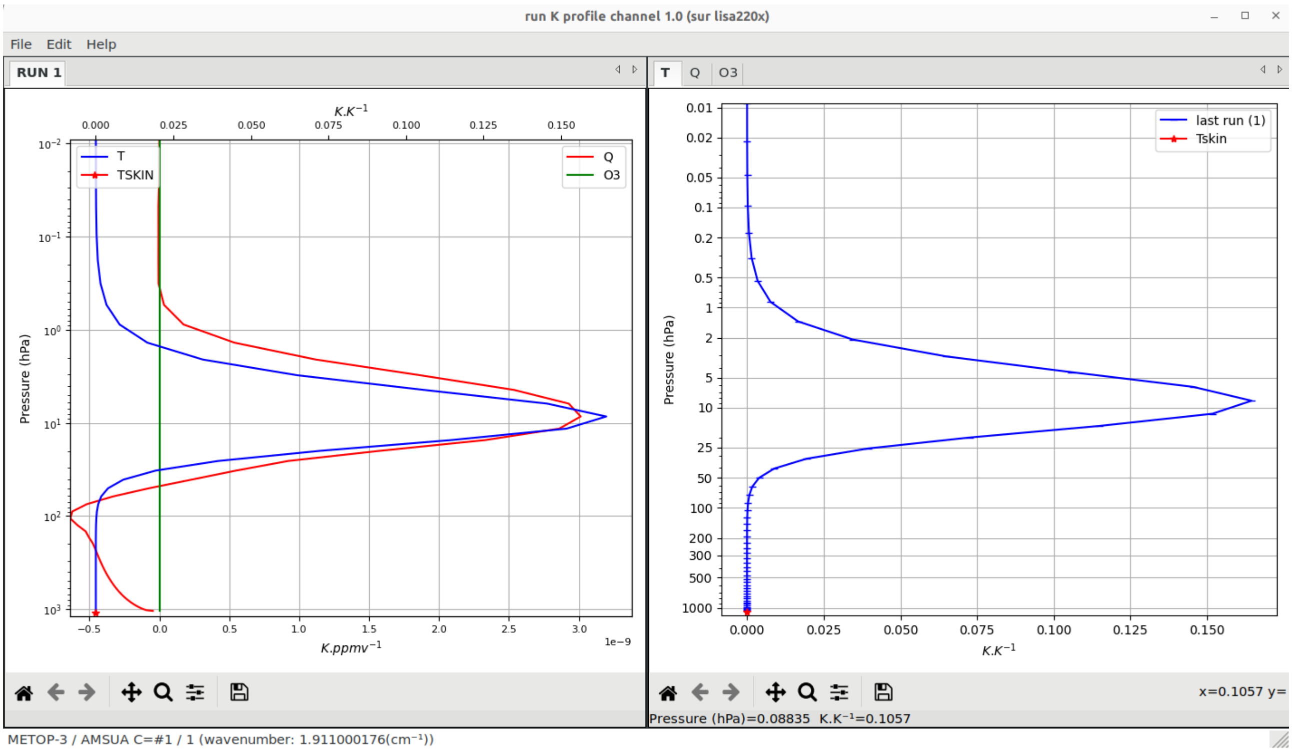


Figure 3.5: K profile window which displays a Jacobian for METOP-3/AMSUA channel 12

From the “RTTOV” menu of the main window, with the “Select Channels” command, the user can perform the channel selection and choose channel 12(Figure 3.6).

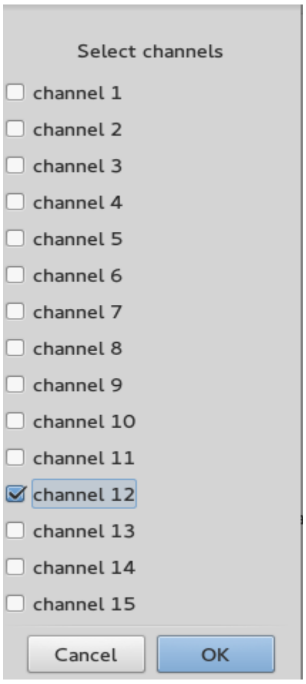


Figure 3.6: The select channels window.

The user can now perform the retrieval with just the METOP-3/AMSUA channel 12.
In the Figure 3.7, which is a retrieval for the METOP-3/AMSUA channel 12.

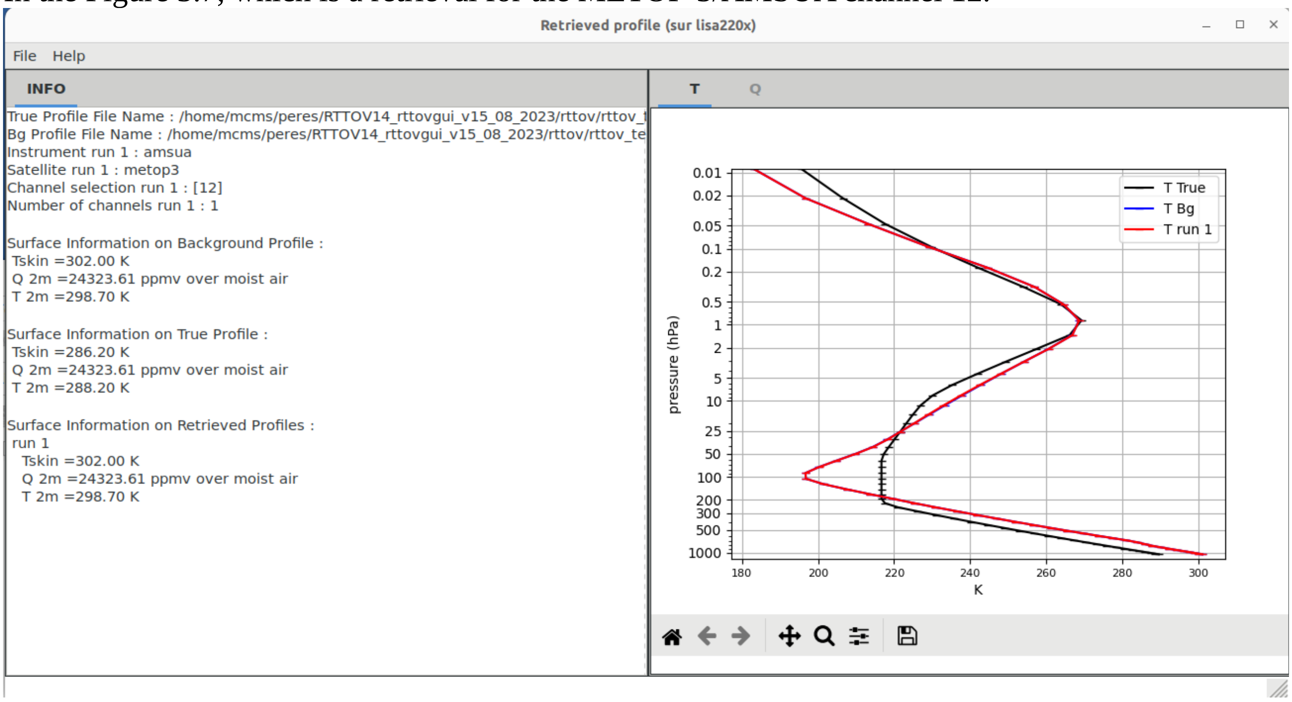


Figure 3.7: Retrieval example (zoomed) with METOP-3/AMSUA channel 12, and a background error scaling of 100.

Computations made by RTTOV GUI:

In the calculation below X_t is the true vertical profile, X_b the background vertical profile and X_r the linear 1DVAR retrieved profile. The true vertical profile is never known in the reality.
But the purpose is to show with RTTOV GUI, how it is possible to retrieve it from satellite observations. In the reality what are known are the observed brightness temperatures: they can be simulated in this experiment with the direct model of RTTOV from X_t :

The computation made by RTTOV GUI are:

1. Compute brightness temperatures Y_t from X_t (RTTOV direct model)

$$Y_t = RTTOV_{direct} (X_t) \quad [1]$$

2. Compute brightness temperatures Y_b from X_b (RTTOV direct model)

$$Y_b = RTTOV_{direct} (X_b) \quad [2]$$

3. Compute Jacobian matrix K and transpose K^T (RTTOV-K model):

$$K = RTTOV\text{-}K (X_b) \quad [3]$$

4. Apply scaling factor to background errors:

$$B = f_b \; B \quad [4]$$

5. Apply scaling factor to observation errors:

$$R = f_r \; R \quad [5]$$

6. Compute linear 1DVAR weights W:

$$W = B\,K^T\,[\,K\,B\;K^T + R\,]^{-1} \qquad [6]$$

7. Compute linear 1DVAR retrieved profile X_r:

$$X_r = X_b + W\;[\,(\,Y_t + N\,) - Y_b\,] \qquad \text{where } N = \text{random noise vector} \quad [7]$$

4. Input Profile file format

The input profiles for the RTTOV GUI can be of two different kind formats. The “native” profile format is HDF5, as the Fortran executable GUI command will only read such format. The other format is an ASCII format made of Python statements for the only necessary variables. The two formats are described below.

4.1. HDF5 profile file format

The HDF5 profile format allows to store a single profile or a profile dataset such as ECMWF83. It also allows to store RTTOV options.

The HDF5 top level structure is like the following (note all groups and datasets are in capital letters):

```
/PROFILES              Group
/PROFILES/0001         Group     First profile
/PROFILES/0002         Group     Optional
....
/PROFILES/9999         Group     Optional
/OPTIONS               Group     Optional
```

The /PROFILES/0001 group contains a copy of the RTTOV profile structure (see rttov_type.F90 and RTTOV User's manual). If the HDF5 file contains several profiles they should be numbered continuously and the group name is made of 4 digits with leading zeros. Under the profile number group, the variable names are HDF5 datasets (capital letters), substructures skin and NEAR_SURFACE are HDF5 subgroups which contains the relevant variables; see below table.

Name	**HDF5 type**	**Dimension**	**Comment**
ID	Dataset	{SCALAR}	User may give text ID to each profile
DATE	Dataset	{3}	Year Month Day
TIME	Dataset	{3}	Hour Minute Second
NLAYERS	Dataset	{1}	Number of atmospheric layers
NLEVELS	Dataset	{1}	Number of atmospheric levels
CTP	Dataset	{1}	Black body cloud top pressure (hPa)
CFRACTION	Dataset	{1}	Black body cloud fraction (0 - 1) 1 for 100% cloud cover
HYDRO	Dataset	{nlayers,nhydro} for MW or {nlayers,nhydro+1} for VIS/IR *nhydro is the number of particle types in the hydrotable, and the +1 for VIS/IR is for the Baran scheme*	optional
HYDRO_FRAC	Dataset	{nlayers,1}	Cloud fraction (0 - 1) 1 for 100%

			cloud cover
HYDRO_DEFF	Dataset	{nlayers,nhydro} *for VIS/IR only*	Units for gas profiles: 0 or less => ppmv over dry air, 1 => kg/kg over moist air, 2 => ppmv over moist air)
ICEDE_PARAM	Dataset	{1}	Scheme for IWC to eff diameter, Dg 1=Ou and Liou; 2=Wyser et al.; 3=Boudala et al; 4=McFarquhar et al.
AEROSOLS	Dataset	{nlayers,naer} *naer depends on the aertable file (13 for OPAC, 9 for CAMS)*	optional
P	Dataset	{nlayers}	Pressure (hPa)
P_half	Dataset	{nlevels}	Pressure (hPa)
T	Dataset	{nlayers}	Temperature
Q	Dataset	{nlayers}	Water vapour
O3	Dataset	{nlayers}	optional Ozone (O3)
CO2	Dataset	{nlayers}	optional
N2O	Dataset	{nlayers}	optional
CO	Dataset	{nlayers}	optional
CH4	Dataset	{nlayers}	optional
SO2	Dataset	{nlayers}	optional
GAS_UNITS	Dataset	{1}	Units for gas profiles :0 or less => ppmv over dry air, 1 => kg/kg over moist air, 2 => ppmv over moist air)
BE	Dataset	{1}	COMMENT = Earth magnetic field strength UNITS = Gauss
COSBK	Dataset	{1}	Cosine of the angle between the Earth magnetic field and wave propagation direction
LATITUDE	Dataset	{1}	Latitude (deg)
LONGITUDE	Dataset	{1}	Longitude (deg)
AZANGLE	Dataset	{1}	Satellite azimuth angle UNITS = degree (0-360 deg;

			east=90)
ZENANGLE	Dataset	{1}	Satellite zenith angle
SUNAZANGLE	Dataset	{1}	Sun azimuth angle
SUNZENANGLE	Dataset	{1}	Sun zenith angle
ELEVATION	Dataset	{1}	Surface elevation
MMR_AER	Dataset	{1}	Units for aerosol 0: cm^-3, 1: kg/kg
MMR_HYDRO	Dataset	{1}	Units for hydrometeor: 0: g/m^3, 1: kg/kg
NEAR_SURFACE	***Group***		
NEAR_SURFACE/Q2M	Dataset	{1}	Water vapour
NEAR_SURFACE/T2M	Dataset	{1}	Temperature (K)
NEAR_SURFACE/WIND_FETCH	Dataset	{1}	U 10m wind component (m/s)
NEAR_SURFACE/WIND_U10M	Dataset	{1}	V 10m wind component (m/s)
NEAR_SURFACE/WIND_V10M	Dataset	{1}	Wind fetch (metres)
SKIN	***Group***		
SKIN/FASTEM	Dataset	{5}	Land/sea-ice surface parameters for fastem
SKIN/FOAM_FRACTION	Dataset	{1}	Ocean foam coverage fraction passed to FASTEM
SKIN/SNOW_FRACTION	Dataset	{1}	Surface snow coverage fraction (0-1). Used only by IR emissivity atlas
SKIN/SURFTYPE	Dataset	{1}	0=land, 1=sea, 2=sea-ice
SKIN/SALINITY	Dataset	{1}	Practical ocean salinity unit (\%o) - FASTEM-4/5/6 only
SKIN/T	Dataset	{1}	Radiative skin temperature (K)
SKIN/WATERTYPE	Dataset	{1}	0=fresh water, 1=ocean water

The /OPTIONS group contains a copy of the RTTOV option structure (see rttov_type.F90 and RTTOV User's manual). There is only option structure in the HDF5 file even if several profiles are stored in. This /OPTIONS group is optional. All RTTOV option variables should be present. The logical variables are converted to integer datasets where “true” is converted to 1 and “false” converted to 0. Note that the RTTOV options substructures are packed all together in the same group. Here are the OPTIONS dataset names by alphabetic order:

```
ADKBt                                    Dataset{SCALAR}
ADKRefl                                  Dataset{SCALAR}
Aerosols                                 Dataset{SCALAR}
ApplyRegLimits                           Dataset{SCALAR}
BaranIceVersion                          Dataset{SCALAR}
CH4Data                                  Dataset{SCALAR}
CLWCloudTop                              Dataset{SCALAR}
CLWData                                  Dataset{SCALAR}
CO2Data                                  Dataset{SCALAR}
COData                                   Dataset{SCALAR}
ChouTangFactor                           Dataset{SCALAR}
ChouTangMod                              Dataset{SCALAR}
ColThreshold                             Dataset{SCALAR}
DomAccuracy                              Dataset{SCALAR}
DomNstreams                              Dataset{SCALAR}
DomOpdepThreshold                        Dataset{SCALAR}
GasOpdepCalc                             Dataset{SCALAR}
HydroFracTLAD                            Dataset{SCALAR}
Hydrometeors                             Dataset{SCALAR}
IcePolarisation                          Dataset{SCALAR}
InterpMode                               Dataset{SCALAR}
IrSeaEmisModel                           Dataset{SCALAR}
Lambertian                               Dataset{SCALAR}
LambertianFixedAngle                     Dataset{SCALAR}
MwPolMode                                Dataset{SCALAR}
MwSeaEmisModel                           Dataset{SCALAR}
N2OData                                  Dataset{SCALAR}
NlteCorrection                           Dataset{SCALAR}
O3Data                                   Dataset{SCALAR}
OverlapParam                             Dataset{SCALAR}
PerHydroFrac                             Dataset{SCALAR}
PermittivityParam                        Dataset{SCALAR}
PlaneParallel                            Dataset{SCALAR}
PressureGradients                        Dataset{SCALAR}
RadDownLinTau                            Dataset{SCALAR}
RayleighMaxWavelength                    Dataset{SCALAR}
RayleighMinPressure                      Dataset{SCALAR}
RayleighMultiScatt                       Dataset{SCALAR}
RayleighSingleScatt                      Dataset{SCALAR}
Refraction                               Dataset{SCALAR}
SO2Data                                  Dataset{SCALAR}
Solar                                    Dataset{SCALAR}
SolarSeaReflModel                        Dataset{SCALAR}
SolarSolver                              Dataset{SCALAR}
ThermalSolver                            Dataset{SCALAR}
TransmittancesOnly                       Dataset{SCALAR}
TwoColMaxFracMaxP                        Dataset{SCALAR}
UseFoamFraction                          Dataset{SCALAR}
UseQ2m                                   Dataset{SCALAR}
UseT2m                                   Dataset{SCALAR}
UseTskinEff                              Dataset{SCALAR}
```

```
UserAerOptParam                         Dataset{SCALAR}
UserHydroOptParam                       Dataset{SCALAR}
```

4.2. ASCII Text profile file

The RTTOV GUI is able to read an ASCII text profile file. This kind of file is made of Python statements for the only necessary variables. Thus the format should respect the Python syntax language. The arrays should be defined as NumPy arrays, the scalars can be pure Python scalars or NumPy variables.

The variable names are the ones described in the RTTOV Users guide for the profile structure (Annex J), capital letters; except for:

- hydrometeors where 2D hydrometeor array is replaced by 1D hydrometeor arrays, one for each hydrometeor short name (table 16 of Users guide)
- aerosols where 2D aerosols array is replaced by 1D aerosol arrays, one for each aerosol short name (table 19 of Users guide)

Units should be the ones described in RTTOV Users Guide for profile structure (Annex J)

3 arrays are mandatory, pressure, temperature and water vapour; all other RTTOV profile variables are optional, they will be set to default values at reading time.

The list of profile variables that can be set by the user is given below; with the default values. Default values aims to a clear atmosphere. Number of levels and layers are deduced from array sizes.

```
#" Mandatory arrays: P_HALF (hPa)on levels”

self["P_HALF"] = numpy.array([…])

#" Mandatory arrays: T(K), Q(ppmv) on layers"
self["T"] = numpy.array([...])
self["Q"] = numpy.array([...])

#"--------------------------------------------"
#" Optional profile variables "
#"--------------------------------------------"

#" Other Gases (gas units on layers)"
self["O3"]  = numpy.array([...])
self["CO2"] = numpy.array([...])
```

```
self["CH4"] = numpy.array([...])
self["CO"]  = numpy.array([...])
self["N2O"] = numpy.array([...])
self["SO2"] = numpy.array([...])

#" Aerosols opac(aerosol units (depend on mmr_aer) on layers)"
self["Inso"] = numpy.array([...]) # Insoluble
self["Waso"] = numpy.array([...]) # Water soluble
self["Soot"] = numpy.array([...]) # Soot
self["Ssam"] = numpy.array([...]) # Sea salt (acc mode)
self["Sscm"] = numpy.array([...]) # Sea salt (coa mode)
self["Minm"] = numpy.array([...]) # Mineral (nuc mode)
self["Miam"] = numpy.array([...]) # Mineral (acc mode)
self["Micm"] = numpy.array([...]) # Mineral (coa mode)
self["Mitr"] = numpy.array([...]) # Mineral transported
self["Suso"] = numpy.array([...]) # Sulphated droplets
self["Vola"] = numpy.array([...]) # OPAC Volcanic ash
self["Vapo"] = numpy.array([...]) # New Volcanic ash
self["Asdu"] = numpy.array([...]) # Asian dust
#" Aerosols cams(aerosol units(depend on mmr_aer) on layers)"
self["Bcar"] = numpy.array([...]) # Black carbon, fixed refractive index at all
wavelengths

self["Dust1"] = numpy.array([...]) # Dust, bin 1, 0.03-0.55 micron, refractive
index: Woodward 2001

self["Dust2"] = numpy.array([...]) # Dust, bin 2, 0.55-0.90 micron, refractive
index: Woodward 2001

self["Dust3"] = numpy.array([...]) # Dust, bin 3, 0.90-20.0 micron, refractive
index: Woodward 2001

self["Sulp"] = numpy.array([...]) # Ammonium sulphate

self["Ssa1"] = numpy.array([...]) # Sea salt, bin 1, 0.03-0.5 micron

self["Ssa2"] = numpy.array([...]) # Sea salt, bin 2, 0.50-5.0 micron
```

```
self["Ssa3"] = numpy.array([...]) # Sea salt, bin 3, 5.0-20.0 micron

self["Omat"] = numpy.array([...]) # Hydrophilic organic matter

#" Hydro VIS/IR sensors(Hydrometeor units (depend on mmr_hydro) on layers)"
self["Stco"] = numpy.array([...]) # Stratus Continental
self["Stma"] = numpy.array([...]) # Stratus Maritime
self["Cucc"] = numpy.array([...]) # Cumulus Continental Clean
self["Cucp"] = numpy.array([...]) # Cumulus Continental Polluted
self["Cuma"] = numpy.array([...]) # Cumulus Maritime
self["Clwdeff"] = numpy.array([...]) # Cloud liquid water type depends on Deff
self["Baum"] = numpy.array([...]) #Ice cloud Baum properties
self["Baran"] = numpy.array([...]) #Ice cloud Baran properties
self["Hydro_FRAC"]= numpy.array([...]) # Cloud Fraction (should be set if any
cloud)
self["Hydro_DEFF"]= numpy.array([...]) # Hydrometeor effective diameter only
applies to the "CLW Deff" and Baum ice types for VIS/IR

#" Hydro MW sensors (Hydrometeor units(depend on mmr_hydro) on layers)"
self["MWRain"] = numpy.array([...]) # Rain
self["MWSnow"] = numpy.array([...]) # Snow
self["MWGraupel"] = numpy.array([...]) # Graupel
self["MWClw"] = numpy.array([...]) # Liquid cloud
self["MWCiw"] = numpy.array([...]) # Ice cloud

#" Skin variables "
self["SKIN"]["T"]          = self["T"][-1]   # (K)
self["SKIN"]["SURFTYPE"]   = 1  # (0=Land, 1=Sea, 2=sea-ice)
self["SKIN"]["WATERTYPE"] = 1  #  (0=fresh water, 1=ocean water)
self["SKIN"]["SALINITY"]   = 37 # (%o)
self["SKIN"]["FOAM_FRACTION"] = 0.0
self["SKIN"]["SNOW_FRACTION"] = 0.0
self["SKIN"]["FASTEM"]     = numpy.array([0., 0., 0., 0., 0.]) # (5 parameters
Land/sea-ice)
```

```
#" 2m and 10m air variables "
self["NEAR_SURFACE"]["T2M"] = self["T"][-1] # (K)
self["NEAR_SURFACE"]["Q2M"] = self["Q"][-1] # (ppmv)
self["NEAR_SURFACE"]["WIND_U10M"] = 0 #  (m/s)
self["NEAR_SURFACE"]["WIND_V10M"] = 0 #  (m/s)
self["NEAR_SURFACE"]["WIND_FETCH"] = 100000 #  (m)

#" Simple cloud "
self["CTP"]       = 500.0 # (hPa)
self["CFRACTION"] = 0.0 # [0,1]    Clear sky is the default

#" Viewing geometry "
self["AZANGLE"]      = 0. # (deg)
self["ELEVATION"]    = 0. # (km)
self["LATITUDE"]     = 49.738  # (deg)  Lannion is 48.750,  -3.470
self["LONGITUDE"]    = -3.473  # (deg)  Exeter is  50.726,  -3.476
self["SUNAZANGLE"]   = 0. # (deg)
self["SUNZENANGLE"]  = 0. # (deg)
self["ZENANGLE"]     = 0. # (deg)

#" Magnetic field "
self["BE"]    = 0.3 # (Gauss)
self["COSBK"] = 1.
# units
self["MMR_HYDRO"] = 1 #Hydrometeor units: 0: "g/m3", 1: "kg/kg",
                        #None:"kg/kg"
self["MMR_AER"] = 1 #Aerosol units: 0: "cm^-3", 1: "kg/kg",
                        #None:"kg/kg"
self["GAS_UNITS"] = 2 #-1: ppmv over dry air, 0: ppmv over dry air,
                      # 1: kg/kg over moist air, 2: ppmv over moist air,
                      #None: ppmv over moist air
#" Miscellaneous "
self["ID"]="This is my profile"
self["DATE"] = numpy.array([2014, 04, 30], dtype=int) # Year, Month, Day
self["TIME"] = numpy.array([12, 0, 0], dtype=int)     # Hour, Minute, Second
```

4.3. How to create an HDF5 profile file from ASCII text profile file

The ASCII text profile files are Python files. A Python command named rttov_gui_import_ascii_profile.py allows the user to convert an ASCII text profile in an HDF5 profile file. This command makes use of RTTOV GUI Python software (Profile class), so cannot be used outside this framework.

```
usage: convert_python2hdf5.py [-h] -i INPUTF [-o OUTPUTF]
                                          [-g GROUP] [-v]

Import ASCII profile to HDF5 for RTTOV GUI

optional arguments:
  -h, --help            show this help message and exit
  -i INPUTF, --input-file INPUTF
                        input file name
  -o OUTPUTF, --output-file OUTPUTF
                        output file name
  -g GROUP, --group GROUP
                        internal HDF5 default is /PROFILES/0001/
  -v, --verbose         display profile variables
```

5. Reporting bugs

All main RTTOV GUI actions are logged in a file named "rttovgui.log" in the RTTOV_GUI_WRK_DIR directory . If the user encounters an issue during the RTTOV GUI usage, he should exit the program and copy the log file to a new name. Then this log file should be attached to any request to the help-desk or forum.

We encourage the users to share experiences through the RTTOV forum at https://nwp-saf.eumetsat.int/site/forums/forum/rttov/