RTTOV v11 Test Suite

Pascal Brunel
MétéoFrance

James Hocking
Met Office

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 1 December, 2006, 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, KNMI and Météo France.

Change record			
Version	Date	Author / changed by	Remarks
0.1	11/01/13	JH	v11 beta release version.
1.0	24/04/13	JH	v11 post-beta update.
1.1	16/05/13	JH	Post-DRI update.
1.2	13/06/14	JH	Updates for v11.2
1.3	22/09/15	JH	Updates for v11.3

Table of Contents

1. Introduction

NB The instructions for testing RTTOV given in the user guide should be sufficient for most users who simply wish to verify their RTTOV installation. This document provides comprehensive information about the test suite.

There are two parts to the RTTOV test suite: the first is a comprehensive and flexible test executable **`rttov_test.exe`** which is controlled via the **`rttov_test.pl`** script and allows most aspects of RTTOV to be configured and run from the command-line. This is described in detail in sections 1-7 of this document.

The second part consists of a set of stand-alone test executables, each with an associated script to run it. These comprise scripts to run the demonstration **`example_*fwd.exe`** and **`example_k.exe`** programs and additional tests for RTTOV-SCATT, the emissivity and BRDF atlases, and are described in section 8.

Section 9 gives an overview of a Python-based plotting utility which may be used to visualise the output from test runs carried out using **`rttov_test.pl`**.

The RTTOV v11 test suite is essentially the same as that for RTTOV v10. The principle difference is that more features of the code may be controlled via command-line parameters. The test suite allows most aspects of RTTOV to be configured either on the command-line or via input files.

RTTOV tests definition and scripts are located in the **`rttov_test`** subdirectory. The **`tests.0`** directory contains the data required to run the tests (these are atmospheric and ground data and a reference to the RTTOV coefficients). Test outputs for the **`myarch`** architecture are located in **`tests.1.myarch`**. Test references are kept in directories whose name ends with **`.2`** ; for instance, **`test_fwd.2`** contains the test references for the **`test_fwd.sh`** test script.

The following scripts and executables are involved in RTTOV tests execution:

- **`rttov_test.pl;`** this script requires Perl >= 5.6 to be installed as **`/usr/bin/perl`**.
- **`rttov_test.exe;`** this executable is created during the building of RTTOV. Its purpose is to run one or more tests and should be called using **`rttov_test.pl`**.
- **`rttov_conv_coef.exe;`** this executable is created during the building of RTTOV. Its purpose is to extract channels from coefficient files and/or to convert them to/from formatted/unformatted/HDF5 format.

By default, the predefined tests expect to find the **`rtcoef_rttov11`** coefficient directory in the RTTOV top directory containing following sub-directories:

- **`rttov7pred54L/`** v7 predictor files on 54 levels
- **`rttov7pred101L/`** v7 predictor files on 101 levels (hi-res sounders only)
- **`rttov8pred51L/`** v8 predictor files on 51 levels
- **`rttov8pred54L/`** v8 predictor files on 54 levels
- **`rttov8pred101L/`** v8 predictor files on 101 levels
- **`rttov9pred54L/`** v9 predictor files on 54 levels
- **`rttov9pred101L/`** v9 predictor files on 101 levels (hi-res sounders only)
- **`cldaer/`** IR cloud and aerosol scattering coefficient files
- **`mietable/`** MW scattering coefficient files
- **`pc/`** Principal Components coefficient files

Note that the vast majority of tests involving hyperspectral IR sounders expect the coefficient files in HDF5 format. The exceptions are the tests called in the ***`rttov_coef_io.sh`*** test script.

`rttov_test.pl` should always be run from the **`rttov_test/`** directory. Whenever this script is run, either the `ARCH` environment variable must be set or it must be supplied as an argument to **`rttov_test.pl`**. It is possible to ask for the list of tests defined in **`tests.0`**:

```
$ ./rttov_test.pl ARCH=ifort LIST_ONLY=1
                    +-------------------------------------------------------------------------------------------- number of Channels
                    |     +-------------------------------------------------------------------------------------- number of Profiles
                    |     |    +--------------------------------------------------------------------------------- number of Levels
                    |     |    | +------------------------------------------------------------------------------- calcEmis true
                    |     |    | |+------------------------------------------------------------------------------ calcRefl true
                    |     |    | ||+----------------------------------------------------------------------------- Aerosols
                    |     |    | |||+---------------------------------------------------------------------------- Clouds
                    |     |    | ||||                    +--------------------------------------------------------- Gas units
                    |     |    | ||||                    | +------------------------------------------------------- do Checkinput
                    |     |    | ||||                    | |+------------------------------------------------------ Apply reg limits
                    |     |    | ||||                    | ||+----------------------------------------------------- Reg limit extrap
                    |     |    | ||||                    | |||+---------------------------------------------------- space Top
                    |     |    | ||||                    | ||||+--------------------------------------------------- Gradients wrt pressure
                    |     |    | ||||                    | |||||+-------------------------------------------------- Interpolation mode
                    |     |    | ||||                    | ||||||  +----------------------------------------------- sWitchrad
                    |     |    | ||||                    | ||||||  |+---------------------------------------------- Refraction
                    |     |    | ||||                    | ||||||  ||+--------------------------------------------- use s2m%Q
                    |     |    | ||||                    | ||||||  |||+-------------------------------------------- Lambertian surface
                    |     |    | ||||                    | ||||||  ||||+------------------------------------------- Solar radiation
                    |     |    | ||||                    | ||||||  |||||+------------------------------------------ NLTE correction
                    |     |    | ||||                    | ||||||  ||||||+----------------------------------------- Fastem version
                    |     |    | ||||                    | ||||||  |||||||    +---------------------------------- scale TL/AD Increments
                    |     |    | ||||                    | ||||||  |||||||    |    +----------------------------- scale TL/AD Output
                    |     |    | ||||                    | ||||||  |||||||    |    | +--------------------------- number of Threads
                    |     |    | ||||                    | ||||||  |||||||    |    | |    +-------------------- Multiplicity
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    +--------------- call RTTOV N times
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | +------------- Channel by channel
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |+------------ Profile by profile
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||+----------- Temp allocation
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||+---------- All coef channels read
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||||+--------- Format of extracted coefs
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||||+-------- Ignore small values
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||||||+------- Direct
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||||||+------ Tangent linear
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||||||||+----- Adjoint
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||||||||+---- K-matrix
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||||||||||+--- Brute force k-matrix
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||||||||||+-- Tangent linear k-matrix
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | ||||||||||||+- Adjoint k-matrix
                    |     |    | ||||                    | ||||||  |||||||    |    | |    |    | |||||||||||||+ Taylor test
    TEST_ID         C     P    L ERAC       GASES        G CARTGI WRQLSNF    I    O T    M    N CPTAFIDTAKBTAT
    abi/001          60    6   54 X...  o3               2 X..X.1 X.X...5    1    1 0    1    1 ..............
    abi/201          96    6   54 X...  o3               2 X..X.1 X.X...5    1    1 0    1    1 ..............
    ahi/001          60    6   54 X...  o3               2 X..X.1 X.X...5    1    1 0    1    1 ..............
...
```

It is possible to filter the list with a regular expression; for instance:

```
$ ./rttov_test.pl LIST_ONLY=1 TEST_MATCH=avhrr
```

would display the list of tests whose name contains "**avhrr**".

2. High level options

The list of tests from the previous subsection shows the description of each test; on the left hand side the following information appears:

- The name of the test.
- The number of channels involved.
- The number of profiles.
- The number of levels.
- Whether RTTOV will calculate any surface emissivities internally.
- Whether RTTOV will calculate any surface reflectances (BRDFs) internally.
- Whether this test contains some aerosol data.
- Whether this test contains cloud data.
- The list of gases which appear in the test input data.

On the right hand side are listed the options which can be changed at run time:

- Gas_units RTTOV is assuming
- Setting of do_checkinput boolean.
- Setting of apply_reg_limits boolean to restrict profiles within regression limits.
- Setting of reg_limit_extrap boolean for profile extrapolation at the top of the atmosphere.
- Setting of spacetop boolean.
- Setting of lgradp boolean.
- The interpolation mode to be used (the interpolator is automatically switched on if required).
- Setting of the switchrad boolean.
- Setting of addrefrac boolean (atmospheric refraction).
- Setting of use_q2m boolean (use 2m water vapour value).
- Setting of do_lambertian boolean.
- Setting of addsolar boolean.
- Setting of do_nlte_correction boolean.
- The version of FASTEM to be used.
- Scale factor applied to TL/AD increments.
- Scale factor applied to TL/AD output.
- Number of threads; this option activates the RTTOV parallel routines. A value of 0 implies that the regular RTTOV high level routines will be run.
- Multiplicity; this expands the test data by some factor. A multiplicity of 10 applied to a test case with 2 profiles and 5 channels will cause the test to be run with 20 profiles with 5 channels each.
- Number of times the test case is run (within the same invocation of `rttov_test.exe`).
- Flag indicating test will be run on each channel separately.
- Flag indicating test will be run for all channels on each profile separately.
- Temp allocation flag. RTTOV is run with temporary data allocated outside the RTTOV high level routines (i.e. using the rttov_alloc_traj subroutine).

- Flag to indicate all channels are read from the coefficient file. RTTOV may then be called for a subset of channels. If not set only the required channels are read from the coefficient file.
- Format of extracted coefficients. It is possible to extract the coefficient data for the channels which are actually used in the test. The extracted coefficient data will be saved for future re-use in the **`coefs.1.myarch`** directory. The **`rttov_conv_coef.exe`** binary is used for extracting these data. If coefficient extraction has been activated (by passing COEF_EXTRACT=1), this item will indicate the format of the extracted coefficients: **`formatted (F), unformatted (U)`** or **`HDF5 (H)`**.
- "Ignoretiny" flag: if set the test suite will ignore small values when checking the data which can be useful for screening out insignificant differences from the output.
- Direct, tl, ad, k, k_bf, k_tl, k_ad. These flags activate some calculations:
  - **`DIRECT=1`** : the direct model
  - **`TL=1`** : the tangent linear model
  - **`AD=1`** : the adjoint model
  - **`K = 1`** : the K matrix computation
  - **`K_BF=1`** : calculation of an approximation of the K matrix using the direct model
  - **`K_TL=1`** : exact calculation of the K matrix using the tangent linear model
  - **`K_AD=1`** : exact calculation of the K matrix using the adjoint model
- Carry out Taylor test (cannot be used simultaneously with the other direct/tl/ad/k/etc options).

All available options can be listed by typing:

```
$ ./rttov_test.pl ARCH=ifort HELP=1

   + ARCH=...                          mandatory unless $ARCH environment variable is set

   + SESSION=...                       test session name (default: tests)

   + BIN=bin                           directory where binary executables are kept;
                                       this path is relative to RTTOV top directory
                                       (default: bin)

   + TEST_LIST=hirs/01,airs/51,...     comma separated list of tests to be run; it
                                       is also possible to define tests such as:
                                       hirs/01+airs/51; in this case, hirs/01 and
                                       airs/51 will be run from within the same
                                       executable (default: all tests)

   + TEST_MATCH=hirs                   regex to filter the tests

   + LIST_ONLY=1                       do not run tests, show list (default: 0/false)

   + COEF_EXTRACT=1                    extract needed coefficients data; this should not
                                       be used for tests where the number of channels
                                       varies from profile to profile (default: 0/false)

   + COEF_FORMAT=formatted             format for extracting coefficient data:
                                       formatted/unformatted/hdf5; has no effect unless
                                       COEF_EXTRACT=1 (default: formatted)

   + LALLCHANS=1                       force all channels to be read from the coefficient
                                       file; ignored if COEF_EXTRACT=1 (default: 0/false)

   + TEMP_ALLOC=1                      run RTTOV with temporary data allocated
                                       outside RTTOV (default: 0/false)

   + MULT=10                           number of channels and profiles is increased by
                                       a factor of MULT (default: 1)

   + NTIMES=10                         number times to run RTTOV (default: 1)

   + NTHREADS=2                        number of threads to run RTTOV (rttov_direct
                                       rttov_tl, rttov_ad, rttov_k) with; a value of 1 or
                                       more will force RTTOV to be called via the parallel
```

```
                                  interface (default: 0)

+ PRINT=1                         print results to disk (default: 1/true)

+ DIRECT=1 TL=1, AD=1, K=1, K_BF=1, K_TL=1, K_AD=1
                                  enables direct, tangent linear, adjoint, K matrix,
                                  brute force K matrix, tangent linear K matrix,
                                  adjoint K matrix (defaults: 0/false)

+ TAYLOR=1                        performs Taylor test; this cannot be run at the
                                  same time as any of the above tests
                                  (default: 0/false)

+ TAYLOR_BY_CHAN=1                if set Taylor test is performed per channel instead
                                  of per profile (default: 0/false)

+ CALC_RAD2=1                     calculate secondary radiances (only applies to
                                  rttov_direct) (default: 0/false)

+ TEST_REF=...                    provides reference data to check direct/tl/ad/k
                                  tests results against when CHECK=1 (default: none)

+ CHECK=1                         performs check between direct/tl/ad/k and the
                                  supplied TEST_REF reference data, performs internal
                                  consistency checks on k_bf/k_tl/k_ad/k, and checks
                                  that TAYLOR test output converges correctly
                                  (default: 1/true)

+ DOREAL=1                        performs k_bf/k comparison in real values, default
                                  is test in scaled integers (default: 0/false)

+ IGNORETINY=1                    ignores small absolute values and small relative
                                  differences when reporting differences
                                  (default: 0/false)

+ TINYABS=1.E-11                  with IGNORETINY ignore differences when values being
                                  compared are smaller than this (default: 1.E-11)

+ TINYREL=1.E-5                   with IGNORETINY ignore relative differences smaller
                                  than this (default: 1.E-5)

+ PRINT_ERROR=0                   print error in the test listing (default: 1/true)

+ SCALE_INC=2                     scale increments for TL/AD computations by
                                  a factor of SCALE_INC (default: 1.0)

+ SCALE_OUT=2                     scale TL/AD output of TL/AD computations by
                                  a factor of SCALE_OUT (default: 1.0)

+ SWITCHRAD=0                     sets the switchrad boolean (default: 1/true)

+ REFRACTION=1                    activates refraction (default: 0/false)

+ USE_Q2M=1                       use s2m%q input profile variable (default: 1/true)

+ SOLAR=1                         activates solar radiation (default: 0/false)

+ DO_NLTE=1                       sets the do_nlte_correction flag to true
                                  (default: 0/false)

+ CLDSTR_THRESH=-1.0              set the value of cldstr_threshold (default: -1.0)

+ DO_LAMBERTIAN=1                 computes reflected downwelling radiation lambertian
                                  reflected instead of specular (default: 0/false)

+ DO_LAMBERTIAN_MW=1              computes reflected downwelling radiation lambertian
                                  reflected instead of specular (MW-only) (default: 0/false)

+ DO_LAMBERTIAN_IR=1              computes reflected downwelling radiation lambertian
                                  reflected instead of specular (IR-only) (default: 0/false)

+ FASTEM_VERSION=6                set the version of FASTEM to use for MW emissivity
                                  calculations; if the value specified is not valid
                                  (see user guide for FASTEM versions), the version
                                  specified in the coef file is used
                                  (default: 5)
```

```
+ SUPPLY_FOAM_FRACTION=1          sets the supply_foam_fraction boolean which controls
                                  whether FASTEM uses the input foam fraction value in
                                  the input profile or not (default: 0/false)

+ ADDINTERP=1                     sets the addinterp boolean. NB if the input pressure
                                  levels differ to the coef file levels then the RTTOV
                                  interpolator is switched on automatically by the
                                  test suite so this switch is not usually required
                                  (default: 0/false)

+ INTERP_MODE=5                   sets the interpolation mode; see user guide for valid
                                  settings; has no effect if interpolation is off
                                  (default: 1)

+ REG_LIMIT_EXTRAP=1              sets the reg_limit_extrap boolean which, if true,
                                  extrapolates the input profile at the top of the
                                  atmosphere using the regression limits
                                  (default: 0/false)

+ LGRADP=1                        sets the lgradp boolean which is used to include
                                  variations wrt pressure if the internal RTTOV
                                  interpolation is used (default: 0/false)

+ SPACETOP=0                      sets the spacetop boolean (default: 1/true)

+ USER_CHECK_OPTS=1               run rttov_user_options_checkinput to check
                                  consistency between input options and coefs
                                  (default: 1/true)

+ USER_CHECK_PROF=1               run rttov_user_profile_checkinput to check input
                                  profiles are within limits (default: 0/false)

+ GAS_UNITS=1                     sets the gas units RTTOV is run with; the test
                                  suite will convert from INPUT_GAS_UNITS if they are
                                  different. This option sets profiles(:) % gas_units.
                                  Valid settings:
                                    2 = ppmv over moist air
                                    1 = kg/kg over moist air
                                    0 = compatibility mode
                                   -1 = ppmv over dry air
                                  The vast majority of test suite input profiles are
                                  in units of ppmv over moist air.
                                  (default: 2/ppmv over moist air)

+ INPUT_GAS_UNITS=1               specifies the gas units of the input test suite files.
                                  If unspecified on the commandline, the value of
                                  INPUT_GAS_UNITS is taken from gas_units.txt.
                                  If this file is not present, the default is ppmv over
                                  moist air. By specifying this on the commandline
                                  any value in gas_units.txt is over-ruled.
                                  If INPUT_GAS_UNITS differs from GAS_UNITS, the test
                                  suite will convert the input profiles before
                                  calling RTTOV. Valid settings:
                                    2 = ppmv over moist air
                                    1 = kg/kg over moist air
                                    0 = ppmv over dry air
                                  NB RTTOV is always run with units GAS_UNITS so, for
                                  example, Jacobians are in units of GAS_UNITS rather
                                  than the units of the input files.
                                  (default: defined by gas_units.txt, otherwise
                                  2/ppmv over moist air)

+ DO_CHECKINPUT=0                 sets the do_checkinput boolean (default: 1/true)

+ APPLY_REG_LIMITS=1              sets the apply_reg_limits boolean (default: 0/false)

+ VERBOSE=0                       sets the verbose boolean (default: 1/true)

+ PROF_BY_PROF=1                  run RTTOV a single profile at a time
                                  (default: 0/false)

+ CHAN_BY_CHAN=1                  run RTTOV a single channel at a time
                                  (default: 0/false)

+ PACK=directory-name
```

```
+ UNPACK=directory-name

+ DR_HOOK=1                    activates DR_HOOK (RTTOV has to be compiled with
                               DR_HOOK library, default: 0/false)

+ FTRACE=1                     activates FTRACE (RTTOV has to be compiled with
                               -ftrace option, default: 0/false)

+ MEMCHECK=1                   run RTTOV through valgrind's memcheck tool with
                               some default options for memory leak checking.
                               Valgrind must be installed on your system
                               (default: 0/false)

+ MASSIF=1                     run RTTOV through valgrind's massif tool with
                               some default options for testing peak memory
                               usage. Valgrind must be installed on your
                               system (default: 0/false)

+ ALLMOD=1                     shortcut: run all models
                               (same as DIRECT=1 TL=1 AD=1 K=1)

+ KCONS=1                      shortcut: run K consistency tests
                               (same as K=1 K_TL=1 K_AD=1)

+ CEU=1                        shortcut: extract coefficients to an unformatted
                               file (same as COEF_EXTRACT=1, COEF_FORMAT=unformatted)

+ CEH=1                        shortcut: extract coefficients to an HDF5
                               file (same as COEF_EXTRACT=1, COEF_FORMAT=hdf5)

+ CPU=1                        shortcut: turn off output for timing testing
                               (same as PRINT=0, VERBOSE=0)
```

We detail here the options we have not explained yet and whose meaning might not be obvious from the description above:

- **ARCH=myarch** specifies the architecture being tested. Test results will be saved in **tests.1.myarch** unless **SESSION** is specified.
- **BIN=install-myarch/bin** has to be specified is RTTOV has been compiled and installed elsewhere than at the top of the RTTOV distribution. This occurs when the **INSTALLDIR=install-myarch** parameter is specified on the command line of **make**. Note that BIN specifies a path relative to the RTTOV top level directory.
- **TEST_LIST=hirs/001,airs/001** is a comma separated list of tests. It is also possible to run several tests in the same execution of **rttov_test.exe** by join test ids with a "**+**"; for instance **TEST_LIST=hirs/001+avhrr/001+amsua/001** will make **rttov_test.exe** run these three tests together: data will be allocated for these tests, coefficients files will be read, calculations will be made and eventually data will be deallocated.
- **PRINT_ERROR=1** will print error messages as tests run (the default is true, but it can be deactivated).
- **DR_HOOK=1** will take care of setting the right options for activating DR_HOOK and saving its output.
- **PRINT=1** will cause **rttov_test.exe** to save its results (default is true, but it can be disabled): when running performance tests this should be turned off.
- **LALLCHANS=1** will force **rttov_test.exe** to load coefficient data for all channels and to manage a subset of them in the calculations.
- **ADDINTERP=1** will switch the RTTOV interpolation on. Unlike RTTOV v10 the test suite, v11 switches the interpolation on automatically whenever it is required, otherwise it is turned off. Therefore this flag is NOT generally required. It is intended for use by developers in order to switch the interpolation on in cases where the input pressure levels match those in the coefficient file (not generally recommended).

- **USER_CHECK_OPTS=0** will prevent the test suite from calling the *rttov_user_options_checkinput* subroutine which it does by default. Some test profile sets in the test suite contain trace gas profiles which are not relevant to some instruments. The *checkinput* subroutine will warn about this and the test will fail, but the test can be run successfully if this check is turned off (the unused trace gases are ignored).

3. Testing DIRECT/TL/AD/K consistency

When several of DIRECT, TL, AD, K, K_BF, K_TL, K_AD are activated, it is possible to check the consistency:

- of the calculations performed by the DIRECT, TL, AD and K models. In this case the output values are compared against test reference data specified by the TEST_REF option.
- of the K matrix calculations of K, K_BF, K_TL, K_AD; the K_BF which is computed in the subroutine **rttov_k_bf** is only an approximation of the K matrix calculated using finite differences; K_TL and K_AD (computed in **rttov_k_tl** and **rttov_k_ad**) should be identical to the K matrix (as computed by **rttov_k)**.

The option **CHECK=1** (set by default) activates this verification. The differences appear in the test log and are recorded in the test output directory. The differences when they appear may have to be more closely examined and may be caused by some rounding errors dependent on the processor and/or compiler. It is common for tests to report some differences other than those due to rounding errors which do not necessarily indicate problems with the code, particularly in relation to the internal consistency checks:

- The tangent linear and adjoint Jacobians should in theory be identical, but there may be differences in the least significant digits due to rounding errors and differences in the code paths.
- With the LGRADP=1 option, the pressure K Jacobian in the top-most and bottom-most layers will often differ slightly to the TL Jacobian. Small differences may also be observed in the cloud fraction ("cfrac") Jacobians for cloudy IR simulations.
- For Principal Components calculations the emissivity TL Jacobian is always zero and so will differ to the emissivity K Jacobian.

The comparisons between the K, K_TL and K_AD Jacobians are done exactly i.e. all differences are reported. However, the brute force Jacobian (K_BF) is expected to differ slightly from the others. By default, the BF comparison is done by scaling the Jacobian values up to integers and reporting differences which exceed a threshold specified in the code. If the DOREAL=1 option is specified then the BF comparison is carried out on real values, and differences are reported if they exceed 10% of the K matrix values. This can result in many differences being reported which do not necessarily indicate problems. The DOREAL option is intended for use by developers.
For the K/K_TL/K_AD consistency checks and the comparisons to reference data, the test suite can be configured to ignore small differences. This is enabled by setting **IGNORETINY=1**. There are two associated parameters: differences will not be reported for any output values which are smaller in absolute value than **TINYABS** (1.E-11 by default). Relative differences smaller than **TINYREL** (1.E-5 by default) are also ignored.

The TAYLOR argument can be used to test consistency between the direct and tangent linear (TL) code. It should not be supplied at the same time as the DIRECT/TL/AD/K arguments. This compares the TL output

with a “brute force” TL calculated with the direct model by perturbing the input profile. The comparison is repeated with decreasing perturbations, and the ratio of the real and brute force TLs should approach 1.0. The output of the Taylor test is written to the file **`taylor_test.log`**. The output is examined by the test script and the calculated ratios will be printed out for any profiles which do not appear to be converging to 1.0. This automated checking sometimes flags false positives (i.e. it flags differences where no problem exists), but has not been observed to give false negatives. Note that as the perturbations become very small, rounding errors begin to cause the ratio to deviate significantly from 1.0. The Taylor test ratio is calculated for a sum of radiances over all channels. To calculate separate ratios for each channel supply the **`TAYLOR_BY_CHAN=1`** argument to **`rttov_test.pl`**.

4. Looking at a test output

Run the following command (from the rttov_test/ directory):

```
$ ./rttov_test.pl ARCH=myarch TEST_LIST=hirs/001,avhrr/001 DIRECT=1 K=1

Start: 14/08/2015 09:47:59
                  +------------------------------------------------------------------------------------------ number of Channels
                  |     +------------------------------------------------------------------------------------ number of Profiles
                  |     |    +------------------------------------------------------------------------------- number of Levels
                  |     |    | +----------------------------------------------------------------------------- calcEmis true
                  |     |    | |+---------------------------------------------------------------------------- calcRefl true
                  |     |    | ||+--------------------------------------------------------------------------- Aerosols
                  |     |    | |||+-------------------------------------------------------------------------- Clouds
                  |     |    | ||||                    +----------------------------------------------------- Gas units
                  |     |    | ||||                    | +--------------------------------------------------- do Checkinput
                  |     |    | ||||                    | |+-------------------------------------------------- Apply reg limits
                  |     |    | ||||                    | ||+------------------------------------------------- Reg limit extrap
                  |     |    | ||||                    | |||+------------------------------------------------ space Top
                  |     |    | ||||                    | ||||+----------------------------------------------- Gradients wrt pressure
                  |     |    | ||||                    | |||||+---------------------------------------------- Interpolation mode
                  |     |    | ||||                    | ||||||  +------------------------------------------- sWitchrad
                  |     |    | ||||                    | ||||||  |+------------------------------------------ Refraction
                  |     |    | ||||                    | ||||||  ||+----------------------------------------- use s2m%Q
                  |     |    | ||||                    | ||||||  |||+---------------------------------------- Lambertian surface
                  |     |    | ||||                    | ||||||  ||||+--------------------------------------- Solar radiation
                  |     |    | ||||                    | ||||||  |||||+-------------------------------------- NLTE correction
                  |     |    | ||||                    | ||||||  ||||||+------------------------------------- Fastem version
                  |     |    | ||||                    | ||||||  ||||||    +-------------------------------- scale TL/AD Increments
                  |     |    | ||||                    | ||||||  ||||||    |    +--------------------------- scale TL/AD Output
                  |     |    | ||||                    | ||||||  ||||||    |    | +------------------------- number of Threads
                  |     |    | ||||                    | ||||||  ||||||    |    | |    +-------------------- Multiplicity
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    +--------------- call RTTOV N times
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | +------------- Channel by channel
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |+------------ Profile by profile
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||+----------- Temp allocation
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||+---------- All coef channels read
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||||+--------- Format of extracted coefs
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||||+-------- Ignore small values
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||||||+------- Direct
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||||||+------ Tangent linear
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||||||||+----- Adjoint
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||||||||+---- K-matrix
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||||||||||+--- Brute force k-matrix
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||||||||||+-- Tangent linear k-matrix
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | ||||||||||||+- Adjoint k-matrix
                  |     |    | ||||                    | ||||||  ||||||    |    | |    |    | |||||||||||||+ Taylor test
     TEST_ID        C     P    L ERAC       GASES        G CARTGI WRQLSNF    I    O T    M    N CPTAFIDTAKBTAT  REAL TIME   USER TIME STATUS
    hirs/001        114     6  54 X... o3               2 X..X.1 X.X...5    1    1 0    1    1 ......X..X....       1.19        0.07   OK
    avhrr/001        18     6  54 X... o3               2 X..X.1 X.X...5    1    1 0    1    1 ......X..X....       0.07        0.01   OK

End: 14/08/2015 09:48:00

Ran 2 tests, 2 = OK
```

Note that if your RTTOV has been compiled in a custom directory using the **`INSTALLDIR`** Makefile parameter, you have to use the **`BIN=...`** option as explained in the previous section. The **`rttov_test`** directory may contain a **`arch/myarch`** file holding the the myarch related environment variables to be exported before running the tests (this is optional, but may help debugging).

Two tests have been run separately: **hirs/001** and **avhrr/001**. It is possible to run them together by typing:

```
$ ./rttov_test.pl ARCH=myarch TEST_LIST=hirs/001+avhrr/001 DIRECT=1 K=1

     TEST_ID        C     P    L ERAC       GASES        G CARTGI WRQLSNF    I    O T    M    N CPTAFIDTAKBTAT  REAL TIME   USER TIME STATUS
    hirs/001        114    6  54 X... o3                2 X..X.1 X.X...5    1    1 0    1    1 ......X..X....
    avhrr/001        18    6  54 X... o3                2 X..X.1 X.X...5    1    1 0    1    1 ......X..X....       0.16        0.08   OK

End: 14/08/2015 09:48:59

Ran 1 tests, 1 = OK
```

The directory **tests.1.myarch/avhrr/001/out** contains the results of the **avhrr/001** test; it contains the following files:

```
$ find
./env.sh                        # compiler related environment variables (if any)
./run.sh                        # shell script to re-run the test by hand
                                # ( just type ./run.sh from the test output
                                #   directory )
./direct
./direct/transmission.txt       # transmission data produced by rttov_direct
./direct/radiance.txt           # radiance data producted by rttov_direct
./rttov_test.log                # standard error and output of rttov_test.exe
./rttov_test.txt                # namelist for rttov_test.exe
./interpolation.log             # log indicating whether interpolation was on or off
./gas_units.log                 # log indicating the assumed gas units
./k
./k/transmission.txt            # transmission data produced by rttov_k
./k/radiance.txt                # radiance data produced by rttov_k
./k/emissivity_k.txt            # gradient of the radiances relative to the
                                # emissivity
./k/profiles_k.txt              # gradient of the radiance relative to the
                                # atmospheric and ground data
```

Note that if you have run the combined test **hirs/001+avhrr/001**, then the log for the combined test, **rttov_test.log**, is dumped in the last test directory (**tests.1.myarch/avhrr/001/out**). Note however that test outputs are stored in separate test directories.

The **.txt** files contain the results of RTTOV calculations in a human readable format. For instance:

```
$ cat direct/radiance.txt
RADIANCE%TOTAL = (
  0.340210        88.8450
 )
RADIANCE%BT = (
   283.777        283.286
 )
RADIANCE%CLEAR = (
  0.340210        88.8450
 )
RADIANCE%BT_CLEAR = (
   283.777        283.286
 )
```

It is possible to have results saved in a directory whose name is not **tests.1.myarch**; for this purpose, it is necessary to append the **SESSION=mysession** argument to the list of parameters passed to **rttov_test.pl**; results will then be saved to the **mysession.1.myarch** directory.

5. Creating a new test

Test definition is located in the `tests.0` directory; every subdirectory of `tests.0` which contains an "`in`" subdirectory is interpreted by `rttov_test.pl` as a test definition.
This in subdirectory must contain the following files:

```
./lprofiles.txt      # profile list (chanprof(:)%prof)
./channels.txt       # channel list (chanprof(:)%chan)
./coef.txt           # coefficient file namelist
./profiles/001/atm
./profiles/001/atm/p.txt             # pressure levels ( hPa )
./profiles/001/atm/t.txt             # temperature ( K )
./profiles/001/atm/q.txt             # water vapour ( ppmv )
./profiles/001/atm/aerosli.txt       # ice cloud parameters
./profiles/001/atm/cloud0.txt        # simple cloud parameters
./profiles/001/ground
./profiles/001/ground/skin.txt       # skin parameters
./profiles/001/ground/s2m.txt        # s2m parameters
./profiles/001/ground/elevation.txt  # surface elevation ( km )
./profiles/001/be.txt                # magnetic field parameters
./profiles/001/angles.txt            # angles parameters
```

It may also contain the following files:

```
./calcemis.txt       # calcemis flags for RTTOV
./emissivity.txt     # emissivity
./calcrefl.txt       # calcrefl flags for RTTOV
./reflectance.txt    # surface reflectance (BRDF)
./profiles/001/gas_units.txt     # Gas units namelist
./profiles/001/atm/clw.txt       # Cloud liquid water ( kg/kg )
./profiles/001/atm/n2o.txt       # N2O ( ppmv )
./profiles/001/atm/co2.txt       # CO2 ( ppmv )
./profiles/001/atm/ch4.txt       # CH4 ( ppmv )
./profiles/001/atm/o3.txt        # O3 ( ppmv )
./profiles/001/atm/co.txt        # CO ( ppmv )
./profiles/001/atm/icede.txt     # ice particle effective diameter ( µm )
./profiles/001/atm/aerosl.txt    # aerosol concentrations
./profiles/001/atm/cfrac.txt     # cloud fraction
./profiles/001/atm/cloud.txt     # cloud liquid/ice ( kg/kg )
./profiles/001/datetime.txt      # date and time of profile
./aer_opt_param.txt              # aerosol optical parameter profiles
./cld_opt_param.txt              # cloud optical parameter profiles
```

For Principal component calculations:

```
./pcscores.txt                   # regression set and number of pcscores
./channels_rec.txt               # reconstructed channels (optional)
```

For tests involving pressure modulated cells (e.g. using the new PMC shift SSU coefficients) the following file is mandatory for the input of the cell pressures:

```
./pmc.txt                        # cell pressure for each channel
```

Creating a new test is just the matter of creating a new subdirectory of **`tests.0`**. The easiest is to copy a pre-existing test and modify it to suit your needs.

The configuration described above is a single profile configuration (profile 001): adding some more profiles involves creating other directories named 002, 003, etc with data laid out as described above.

The format of data within most of the above files is clear upon inspection of an existing example. The exceptions to this are the aerosol/cloud optical parameter files (**`aer_opt_param.txt/cld_opt_param.txt`**). The data are read directly into the ***`rttov_opt_param`*** Fortran type (see user guide) as follows:

- number of phase angles (single integer value)
- **`abs(:,:)`**
- **`sca(:,:)`**
- **`bpr(:,:)`**

Then, if ***`addsolar`*** is TRUE:

- **`phangle(:)`**
- ***`pha(:,:,:)`***

Optical parameters must be present for every channel/profile and every level specified by the test definition.

In order to run PC-RTTOV tests the **`pcscores.txt`** file should be present: this automatically sets addpc to true. The PC coefficient file must be specified in **`coef.txt`**.

In order to run visible/IR aerosol scattering tests either the **`aer_opt_param.txt`** or the **`aerosl.txt`** files should be present (the **`aer_opt_param.txt`** file takes precedence). If running simulations with the pre-defined aerosol types, the aerosol coefficient file must be specified in **`coef.txt`**.

In order to run visible/IR cloud scattering tests either the **`cld_opt_param.txt`** or the **`cloud.txt`** files should be present (the cld_**`opt_param.txt`** file takes precedence). For cloud simulations, there must also be a **`cfrac.txt`** file. If running simulations with the pre-defined cloud types, the cloud coefficient file must be specified in **`coef.txt`**.

It is generally recommended to provide **`calcemis.txt`**, **`emissivity.txt`**, **`calcrefl.txt`** and **`reflectance.txt`** files where applicable (i.e. emissivities should be supplied for thermal emission calculations and reflectances should be supplied for solar calculations; note that values must be present for *all* simulated channels in each of the these files). If any files are missing, calcemis defaults to false and emissivity to 1.0 for all channels, while calcrefl defaults to true for all channels.

6. Testing against a reference

A set of tests with reference output (where appropriate) are provided for users. These tests demonstrate the capabilities of RTTOV v11 for a range of instruments, and the comparison to the reference output can confirm that the code has been compiled correctly. References are provided in **`*.2`** directories.

We provide the following user test scripts:

- **`test_fwd.sh`** tests the forward model for a wide range of instruments
- **`test_rttov11.sh`** tests the full code (direct/TL/AD/K) for a range of instruments
- **`test_rttov11_hires.sh`** tests the full code for AIRS and IASI
- **`test_solar.sh`** tests solar calculations
- **`test_pc.sh`** tests the Principal Component calculations
- **`test_multi_instrument.sh`** tests RTTOV running for multiple instruments together
- **`test_zeeman.sh`** tests Zeeman code (using Zeeman coefficient files)
- **`test_coef_io.sh`** tests the coefficient input/output code (this test has no reference data)
- **`test_coef_io_hdf.sh`** tests the HDF5 coefficient input/output code (this test has no reference data)
- **`test_cpu.sh`** may be used for performance testing (this test has no reference data)

NB Due to the memory management of the Intel Fortran compiler, users compiling with ifort on Linux may need to increase the stack size by executing the following command before all tests will run correctly:

```
$ ulimit -s unlimited
```

In addition, when running the PC-RTTOV K model with multiple threads under ifort, users may need to increase the OMP stack size as well to allow tests to run:

```
$ export OMP_STACKSIZE=1000M
```

Note that the majority of the tests for hyperspectral IR sounders expect HDF5 format coefficient files. The only exceptions are the ***`test_coef_io.sh`*** and ***`test_coef_io_hdf.sh`*** scripts which require ASCII format files. It is not necessary to run every test script to validate your installation of RTTOV: running the ***`test_rttov11.sh`*** script is sufficient for this and this depends only on coefficient files included in the distribution.

The list of tests above may be run using a single script:

```
$ ./test_core.sh ARCH=myarch [BIN=install-myarch/bin]
```

Alternatively each script may be called individually. The test scripts can be run with any of the parameters described above for **`rttov_test.pl`** (though naturally some options will make comparison to the test reference output invalid). In particular, either the **`ARCH`** environment variable must be set or **`ARCH=myarch`** must be passed as an argument to the script. If you have specified **`INSTALLDIR=install-myarch`**, then the **`BIN=install-myarch/bin`** should be provided to the script.

We describe here **`test_rttov11.sh`**:

```
$ cat test_rttov11.sh
#!/bin/sh

# User test

# Tests the full RTTOV-11 code for various MW and IR instruments.

ARG_ARCH=`perl -e 'for(@ARGV){m/^ARCH=(\S+)$/o && print "$1";}' $*`
if [ ! "x$ARG_ARCH" = "x" ]; then
  ARCH=$ARG_ARCH
fi
if [ "x$ARCH" = "x" ];
then
  echo 'Please supply ARCH'
  exit 1
fi

set -x

SESSION=test_rttov11
OPTS="IGNORETINY=1 $*"
WHAT="DIRECT=1 TL=1 AD=1 K=1"
CHECK="CHECK=1 TEST_REF=$SESSION.2"

./rttov_test.pl SESSION=$SESSION $WHAT $CHECK ARCH=$ARCH $OPTS -- << EOF
  TEST_LIST=amsre/001
  TEST_LIST=amsua/001,amsua/021clw
  TEST_LIST=amsub/001
  TEST_LIST=msu/001                REFRACTION=1
  TEST_LIST=ssmis/001,ssmis/021
  TEST_LIST=windsat/001
  TEST_LIST=hirs/001               REFRACTION=1 APPLY_REG_LIMITS=1
  TEST_LIST=modis/021
  TEST_LIST=seviri/222             SOLAR=1
EOF
```

This test script runs several lists of tests, saves results to the **`test_rttov11.1.myarch`** directory and compares them to the reference **`test_rttov11.2`**.

Note that a reference is merely a test output directory renamed with a **`.2`**.

Some test scripts may use the **`COEF_EXTRACT=1`** argument which causes coefficients to be extracted to the **`coefs.1.myarch`** directory for efficiency. Repeated runs of the tests will be faster because these extracted coefficients will be used. However if changes are made in the **`rtcoef_rttov11`** directory (for example updating coefficient files), then **`coefs.1.myarch`** should be deleted to ensure the updated files are used by the tests.

A very thorough set of tests are employed by the developers to validate the code comprehensively. These include tests for every coefficient file provided with RTTOV v11, full tests of all capabilities of RTTOV (including the internal consistency checks described in section 3), checking consistency between the direct and TL using the Taylor test (described in section 3), checking linearity of the TL and AD code, and testing on diverse profile datasets. The scripts and reference output for these tests are not included in the RTTOV v11 package provided to users.

7. Testing performance

The **test_cpu.sh** script is for performance testing. The test cases are run for an increased number of profiles and are run several times so that the time spent in each subroutine is significant. No output data is written on the disk (**PRINT=0**) and verbose warnings are turned off (**VERBOSE=0**).

If RTTOV has been compiled with OpenMP (see user guide) it is possible to pass **NTHREADS=2**, **NTHREADS=3**, etc... to **test_cpu.sh** to see the impact on the real time when RTTOV parallel routines are invoked.

8. Additional test scripts

A number of additional test scripts are supplied which provide examples for running RTTOV, and allow testing of non-core components such as RTTOV_SCATT and the emissivity and BRDF atlases. These test scripts are described here.

Running examples of code calling RTTOV v11

There are several example programs in **src/test/** which demonstrate how to run the RTTOV forward model for various types of simulations:

example_fwd.F90	simple example for clear-sky simulations
example_atlas_fwd.F90	as **example_fwd.F90** but also using the emissivity and BRDF atlases
example_cld_file_fwd.F90	example for cloud simulations using pre-defined particle types
example_aer_file_fwd.F90	example for aerosol simulations using pre-defined particle types
example_aer_param_fwd.F90	example for aerosol simulations using explicit optical parameters
example_rttovscatt_fwd.F90	example for RTTOV-SCATT simulations
example_pc_fwd.F90	example for PC-RTTOV simulations
example_k.F90	example of calling the K model for clear-sky simulations

These examples may be used as a starting point for your own applications. Each of these programs may be run using a shell script with the name **run_example_*.sh** corresponding to the executable name. Near the top of each script is a small section where inputs may be configured such as the coefficient file and its location and the name of the input file(s) for profile data.

The scripts may be run by typing (for example):

```
$ ./run_example_fwd.sh ARCH=myarch [BIN=bindir]
```

Test reference output is in folders named **test_example_*.2/**. Input files for the scripts are in the **test_example_*.1/** directories, and these are also where the test outputs are written. The outputs consist of files named **output_example_*.dat.myarch** and diff files named **diff_example_*.myarch** showing the differences between the test outputs and the corresponding reference outputs. The diff files should typically have zero size although sometimes small differences are seen the least significant digits of some output values.

RTTOV_SCATT testing and example code

The **test_rttovscatt.sh** shell script may be used to verify the RTTOV_SCATT code. You may need to edit the first few lines of this script to specify the location of the RTTOV coefficient files (by default assumed to be in **rtcoef_rttov11/**). The script may then be run by typing:

```
$ ./test_rttovscatt.sh ARCH=myarch [BIN=bindir]
```

Test reference output is in **test_rttovscatt.2/**. Input files for the script are in the **test_rttovscatt.1/** directory, and this is also where the test output is written. The output consists of files named **output.NN.rttov11_scatt.myarch** and **diff.NN.myarch** (where NN is 01, 02, etc), the latter being diff files showing differences compared to the test reference data. The script will exit cleanly if no internal errors are found. The diff file should typically have zero size if no errors occurred.

There is also an example program **mw_scatt/example_rttovscatt.F90** demonstrating how to perform direct and Jacobian calculations with RTTOV_SCATT. Once **test_rttovscatt.sh** has been run, the required links to coefficient files are set up within **test_rttovscatt.1/**. You may then call **example_rttovscatt.exe** (located in **bin/**) from this directory to run the example code. Note there is no reference output for this example program.

Emissivity atlas testing

The emissivity atlas code is not compiled by default, so the instructions in the user guide should be followed to build the atlas test executables. The **test_iratlas.sh** and **test_mwatlas.sh** shell scripts may be used to verify the IR and TELSEM MW atlas code respectively: in each case, these test programs initialise the atlas, return emissivity values for a series of profiles/locations and then deallocate the atlas. You may need to edit the first few lines of each script to specify the location of the RTTOV coefficient files (by default assumed to be in **rtcoef_rttov11/**), and the location of the emissivity atlas data files (by default assumed to be in **emis_data/**). The test scripts require emissivity data for the month of August. All data files associated with the IR emissivity atlas are required for this test (including covariance and angular correction files). If RTTOV was compiled with HDF5 then the HDF5 format IR atlas files must be used, otherwise the netCDF format files must be used. The scripts may be run by typing:

```
$ ./test_iratlas.sh ARCH=myarch [BIN=bindir]
$ ./test_mwatlas.sh ARCH=myarch [BIN=bindir]
```

Test reference output is in **test_emisatlas.2/**. Input files for the scripts are in the **test_emisatlas.1/** directory, and this is also where the test output is written. The output consists of files named **output_iratlas.NN.myarch** and **output_mwatlas.NN.myarch**, where NN is 01, 02, etc. The script also writes diff files named **diff_iratlas.NN.myarch** and **diff_mwatlas.NN.myarch** showing the difference between the test output and the reference output. The difference files should have zero size.

BRDF atlas testing

The BRDF atlas code is not compiled by default, so the instructions in the user guide should be followed to build the atlas test executable. The **test_brdf_atlas.sh** shell script may be used to verify the BRDF atlas code: the test program initialises the atlas, returns BRDF values for a series of profiles/locations and then deallocates the atlas. You may need to edit the first few lines of the script to specify the location of the RTTOV coefficient files (by default assumed to be in **rtcoef_rttov11/**), and the location of the BRDF atlas data files (by default assumed to be in **brdf_data/**). The test script requires BRDF data for the month of August. If RTTOV was compiled with HDF5 then the HDF5 format BRDF atlas files must be used, otherwise the netCDF format files must be used. The script may be run by typing:

```
$ ./test_brdf_atlas.sh ARCH=myarch [BIN=bindir]
```

Test reference output is in **test_brdf_atlas.2/**. Input files for the scripts are in the **test_brdf_atlas.1/** directory, and this is also where the test output is written. The output is written to a file named **output_brdf_atlas.01.myarch**. The script also writes a diff file named **diff_brdf_atlas.NN.myarch** showing the difference between the test output and the reference output. The difference files should have zero size.

9. Visualising test suite output

A Python-based plotting tool is provided in the rttov **rttov_test/** directory which enables plots to be made of various input/output test suite data such as profile variables, radiances, weighting functions, and Jacobians. The code requires Python 2.7 or later, matplotlib and wxPython. It can be invoked from within the **rttov_test/** directory as follows:

```
$ ./rttov_test_plot.py
```

All interface widgets have associated tool-tips which are displayed when the mouse pointer is hovered over them. These give guidance on how to use the interface.

Before displaying test output a test should be run for the direct and K code. For example:

```
$ ./rttov_test.pl ARCH=ifort TEST_LIST=amsua/001 DIRECT=1 K=1 SWITCHRAD=1
```

The “Test dir” text boxes should then be set to the top level test output directory. For this example, this is “tests.1.ifort/amsua/001”.

The profile and channel list text boxes can be left blank to plot output for all profiles associated with the test and all channels defined for each profile or they can contain comma-separated lists of profile and channel numbers to plot. The profile and channel name text boxes can be used to optionally label the individual profiles and channels for the legend. If left blank default labels are used.

On the right-hand side of the interface various plot types can be selected along with associated parameters such as the profile variable to plot (for example for Jacobian plots).

It is also possible to make difference plots for most plot types: in this case select the “Difference plot” check-box and input another test directory name. The resulting plot will show the differences between corresponding profiles/channels in all selected tests on the left-hand side of the interface and the same profiles/channels in the difference test directory. Note that the two tests must have been run the for same

profiles/channels.

If you enable the “Plot stats” option, the GUI will plot any or all of the mean, standard deviation, RMS and maximum absolute value over all profiles for each channel: this is very useful when combined with difference plots for displaying statistics of the differences between two tests.

Finally, the plotted data may be extracted to a comma-separated ASCII file by clicking the “Write to file” button and providing an output file name when prompted. Additional comments can be written out in the file to help interpret the data: this is selected by checking the “Verbose file output?” check-box.