Radiance Simulator v4 Test Plan

James Hocking, Met Office, UK

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
1.0	07/04/2025	J. Hocking	First version

Table Of Contents

1 INTRODUCTION

This document defines the Test Plan for the NWP SAF Radiance Simulator in accordance with the guidelines set out in [RD-1]. The aim of the Test Plan is to verify that the requirements of the product specification in [RD-2] are met.

1.1 Reference documents

[RD-1] NWPSAF-MO-SW-002: Development Procedures for Software Deliverables
[RD-2] NWPSAF-MO-DS-053: Radiance Simulator v4 Product Specification
[RD-3] NWPSAF-MO-UD-061: Radiance Simulator v4 User Guide
[RD-4] NWPSAF-MO-UD-062: Radiance Simulator v4 Release Note

2 TESTED ITEMS

The following items are covered as defined in [RD-1]:

1. Coding
2. Module and Integration tests
3. Validation tests
4. Portability tests
5. Beta testing
6. User documentation

2.1 Coding

The majority of the code is written in Fortran-90 with the addition of some features from Fortran-2003. These additional features are supported by most modern compilers and have been approved for use in Met Office operational systems. The code is checked for conformance with the Fortran-2003 standard using compiler flags.

2.2 Module and Integration tests

The Radiance Simulator consists of a set of Fortran subroutines and modules which are compiled and linked to a single program unit. Independent module testing is not applicable in this case; all testing is performed as an integrated package.

A simple test case is included in the release package which runs RadSim for a single sensor in a basic configuration and the output is compared to reference data. This is used to ensure the compilation was successful and that RadSim runs successfully. This is performed for all compilers/platforms tested (see section on Portability below).

The primary integration testing is carried out using the first script described in the next section (Validation testing) which runs RadSim in a wide variety of configurations. This script is run for all compilers/platforms tested and ensures that the majority of RadSim options are tested across multiple platforms.

2.3 Validation tests

The purpose of the validation tests is to check that the main functions of the Radiance Simulator are performing correctly. As described in the Product Specification [RD-2], these are

- Ingest of NWP model fields (all formats)
- Ingest of observation data files (ASCII and netCDF)
- Interpolation of model fields to observation positions and times (both linear and nearest neighbour interpolation)
- Footprint simulation capability
- Setup and running of the radiative transfer model (RTTOV) for different sensors and simulation types (e.g. clear-sky, trace gases, cloudy, aerosol, PC-RTTOV, etc).
- Writing of results and associated data to an output file

The following test scenarios are performed (numbers given are test scenario IDs). Note that the various scripts referred to below are not included in the package distributed to users.

1. General validation test

   A shell script (*radsim_test.sh*) has been written which runs RadSim in multiple configurations in order to test the majority of the RadSim functionality. The script tests the following:
   - All input model types:
     - Met Office fieldsfiles and PPfiles
     - ECMWF GRIB and netCDF files (including GRIB CAMS fields)
     - ICON and ICON-ART GRIB files
     - HARMONIE-AROME GRIB files
     - JMA GRIB files
     - all NWP SAF profile datasets
   - MW sensor, VIS/IR radiometer, hyperspectral IR sounder, hyperspectral UV sounder
   - Clear-sky simulations, cloudy simulations, aerosol simulations, PC-RTTOV
   - Zeeman-enabled simulations
   - Use of emissivity and BRDF atlases
   - Simulations with solar radiation with fixed and RadSim-calculated solar angles
   - Simulations with fixed satellite zenith and azimuth angles and computed angles for a GEO sensor
   - Use of modified CO_2 background profile
   - Ingest of O_3, CO_2, N_2O, CO, CH_4, and SO_2 fields from applicable datasets
   - Use of RTTOV land surface emissivity and BRDF atlases
   - Simulations on the model grid
   - Simulations using observation data files with both bilinear and nearest neighbour spatial interpolation
   - Temporal interpolation (linear and nearest neighbour)

- Footprint simulations
- RTTOV direct and Jacobian model calls
  - includes all Jacobians: T, q, O3, Tskin, wind u/v, surface emissivity.
- CADS height assignments
- Geometric heights output
- Clear-sky BT/reflectance output

The following table details the RadSim runs performed in this test scenario.

Model data	Sensor	Simulation options
All NWP SAF profile datasets (inc. CAMS and MACC)	ATMS	K model, clear-sky, CLW absorption, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, multi_surface
	SEVIRI	K model, clear-sky, O3, co2_max_ppmv, fixed solar zen angle
		K model, clear-sky, O3, co2_max_ppmv, computed solar zen angles
		Direct model, hydrometeors, Chou-scaling+DOM
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff
	IASI	Direct model, clear-sky, all supported trace gases, CADS height output, no 2m T/q
		PC-RTTOV direct model, clear-sky, all supported trace gases, no 2m T/q
	GOME2	Direct model, clear-sky, O3, fixed solar zen angle, no 2m T/q
60L MACC, 137L CAMS	SEVIRI	Direct model, CAMS aerosols, hydrometeors, Chou-scaling+DOM
ECMWF (IFS) GRIB and netCDF	ATMS	K model, clear-sky, CLW absorption, on model grid, multi_surface
		K model, clear-sky, CLW absorption, obs data file, temporal interp, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, temporal interp, interp_nn_spatial/temporal, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, temporal interp, footprints, multi_surface
	SSMIS	Direct model, clear-sky, obs data file, Zeeman simulations, multi_surface
	SEVIRI	K model, clear-sky, O3, co2_max_ppmv, fixed solar zen angle, on model grid
		K model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, on model grid
		K model, clear-sky, O3, co2_max_ppmv, explicit solar zen angles, obs data file, temporal interp
		K model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, obs data file, temporal interp
		Direct model, clear-sky, O3, co2_max_ppmv, computer solar zen angles, computed geo sat angles, on model grid
		Direct model, clear-sky, O3, co2_max_ppmv, computer solar zen angles, computed geo sat angles, obs data file, temporal interp
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, temporal interp, interp_nn_spatial/temporal,
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file, temporal interp, interp_nn_spatial/temporal,
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, temporal interp, interp_nn_spatial/temporal,
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, temporal interp, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file, temporal interp, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, temporal interp, footprints

	IASI	K model, clear-sky, O3, obs data file, temporal interp, no 2m T/q
		Direct model, hydrometeors, O3, obs data file, temporal interp, no 2m T/q
		PC-RTTOV K model, clear-sky, O3, obs data file, temporal interp, no 2m T/q
		PC-RTTOV direct model, hydrometeors, O3, obs data file, temporal interp, no 2m T/q
ECMWF CAMS GRIB	SEVIRI	Direct model, CAMS aerosols, Chou-scaling+DOM, obs data file
		Direct model, CAMS aerosols, hydrometeors, Chou-scaling+DOM, obs data file
ICON GRIB	ATMS	K model, clear-sky, CLW absorption, on model grid, multi_surface
		K model, clear-sky, CLW absorption, obs data file, temporal interp, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, temporal interp, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, temporal interp, footprints, multi_surface
	SEVIRI	K model, clear-sky, O3, co2_max_ppmv, fixed solar zen angle, on model grid
		K model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, on model grid
		K model, clear-sky, O3, co2_max_ppmv, explicit solar zen angles, obs data file, temporal interp
		K model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, obs data file, temporal interp
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, temporal interp
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file, temporal interp
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, temporal interp
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, temporal interp, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file, temporal interp, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, temporal interp, footprints
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, use model Deff
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, use model Deff
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, use model Deff, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, use model Deff, footprints
	IASI	K model, clear-sky, O3, obs data file, temporal interp, no 2m T/q
		PC-RTTOV K model, clear-sky, O3, obs data file, temporal interp, no 2m T/q
ICON-ART GRIB	SEVIRI	Direct model, ICON-ART aerosols, Chou-scaling+DOM, obs data file
HARMONIE-AROME	ATMS	K model, clear-sky, CLW absorption, on model grid, multi_surface
		K model, clear-sky, CLW absorption, obs data file, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, multi_surface
		Direct model, hydrometeors, delta-Eddington, ARO scaling pol, obs data file, footprints, multi_surface
	SEVIRI	K model, clear-sky, co2_max_ppmv, fixed solar zen angle, on model grid
		K model, clear-sky, co2_max_ppmv, computed solar zen angles, on model grid
		K model, clear-sky, co2_max_ppmv, explicit solar zen angles, obs data file
		K model, clear-sky, co2_max_ppmv, computed solar zen angles, obs data file

		Direct model, hydrometeors, Chou-scaling+DOM, obs data file
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file
		Direct model, hydrometeors, Chou-scaling+DOM, obs data file, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW OPAC, obs data file, footprints
		Direct model, hydrometeors, Chou-scaling+MFASIS-NN, CLW Deff, obs data file, footprints
	IASI	K model, clear-sky, obs data file, no 2m T/q
		PC-RTTOV K model, clear-sky, obs data file, no 2m T/q
JMA and JRA55 GRIB	ATMS	Direct model, clear-sky, on model grid, multi_surface
		K model, clear-sky, obs data file, multi_surface
	SEVIRI	Direct model, clear-sky, O3, co2_max_ppmv, fixed solar zen angle, on model grid
		Direct model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, on model grid
		Direct model, clear-sky, O3, co2_max_ppmv, explicit solar zen angles, obs data file
		Direct model, clear-sky, O3, co2_max_ppmv, computed solar zen angles, obs data file
	IASI	K model, clear-sky, O3, obs data file, no 2m T/q
		PC-RTTOV K model, O3, clear-sky, obs data file, no 2m T/q
UM PP file	ATMS	K model, clear-sky, CLW absorption, obs data file, multi_surface
	SEVIRI	K model, clear-sky, co2_max_ppmv, computed solar zen angles, obs data file
	IASI	K model, clear-sky, obs data file, no 2m T/q
		PC-RTTOV K model, clear-sky, obs data file, no 2m T/q
UM fieldsfile	ATMS	K model, clear-sky, CLW absorption, on model grid, fcst_range, multi_surface
		K model, clear-sky, CLW absorption, obs data file, temporal interp, fcst_range, multi_surface
	SEVIRI	K model, clear-sky, co2_max_ppmv, computed solar zen angles, on model grid
		K model, clear-sky, co2_max_ppmv, computed solar zen angles, obs data file, temporal interp
	IASI	K model, clear-sky, obs data file, temporal interp, no 2m T/q
		PC-RTTOV K model, clear-sky, obs data file, temporal interp, no 2m T/q

Notes:

- IASI tests are for a subset of 302 channels.
- GOME2 tests are for a subset of 112 channels.
- ATMS, SSMIS, and SEVIRI tests are for all sensor channels.
- Solar radiation is always enabled for SEVIRI.
- For non-hyperspectral tests, the *combine_qcrttov* option is false so that RTTOV quality flags are output per channel. For hyperspectral tests, *combine_qcrttov* is true so that the flags are combined for all channels for each profile.
- All tests except those for NWP SAF profile datasets use emissivity and BRDF atlases.
- Obs data files are all in ASCII format.
- Output of profiles, emissivities, BRDFs, and geometric heights are enabled for all tests.

- Output of clear-sky radiances are enabled for all tests with hydrometeor scattering.
- For K model simulations all relevant Jacobians are output.

All outputs from all tests are validated against external RTTOV simulations (see below) except:

- PC-RTTOV tests
- Tests involving footprint simulations
- Tests involving computed GEO satellite angles

This script is run for all compilers/platforms being tested. Results are cross-compared to ensure results are consistent allowing for small compiler-related differences. This comparison is performed by a Python script (*radsim_compare_output.py*) which compares all datasets of interest and allows for absolute and relative tolerances to be specified such that only larger differences are reported. The default absolute tolerance is 1E-16 and the default relative tolerance is 1E-4%. A small number of compilers report maximum differences which exceed the default tolerance (1E-4%) for a few tests: the differences are still very small (order 1E-3% at most) and are not considered problematic.

In addition, a Python script has been created (*radsim_rttov_compare.py*) which accepts a RadSim configuration namelist file and the corresponding RadSim output netCDF file containing the profile data as output by RadSim and the calculated radiances. The script runs the corresponding simulations separately in RTTOV (using the RTTOV Python wrapper - see below) and compares the outputs with those from RadSim to ensure differences are within expected tolerances (differences are primarily due to the lower precision used in certain places in RadSim). This ensures that RadSim is running the RTTOV simulations as intended. This script has proven invaluable for uncovering errors during RadSim development. The *radsim_test.sh* shell script described above automatically runs this Python validation script for all applicable tests.

The RTTOV Python wrapper does not currently support PC-RTTOV simulations. Instead, these are validated by comparing the output brightness temperatures to those from equivalent standard RTTOV simulations to ensure the differences are within expected tolerances using the *radsim_compare_output.py* script.

The RTTOV Python wrapper also does not currently support the *calc_geo_sat_angles* functionality, and the comparison script is not configured to screen out satellite zenith angles larger than the RTTOV maximum so tests with the *calc_geo_sat_angles* option are not validated by the *radsim_rttov_compare.py* script. The GEO satellite angle calculations are validated in a separate test described below.

The footprint simulations cannot be validated by this Python script because the necessary profile data are not output by RadSim for these simulations (since there are multiple profiles per observation). The footprint simulation outputs are validated in a separate test described below.

2. OpenMP test

   This is tested as part of test scenario 1 above. The general validation test is run with multiple threads using OpenMP builds of RadSim and RTTOV, and with a single thread using an optimised non-OpenMP build, and the results are compared to ensure they are identical (to within expected tolerances). This test is primarily intended to ensure that the OpenMP multi-threading in RadSim (for footprint simulations) is working correctly, but it also checks that multi-threaded RTTOV simulations are running correctly through RadSim.

3. Footprints test

   A separate test script (*radsim_test_footprints.sh*) has been created to validate the footprint calculation. This uses netCDF ECMWF data at a lat/lon resolution of 5 degrees (this is very coarse but keeps the test run-times short and memory requirements low while fully testing the functionality). Four obs data files are used:
   a) all model grid points with small footprints (1km/0.75km semi-major/minor axes)
   b) all model grid points with large footprints (500km/375km)
   c) one obs per grid box, lying off the grid points, with small footprints (1km/0.75km)
   d) mixture of grid and off-grid points with mixture of footprint sizes

   Each observation is assigned a random azimuth angle which defines the orientation of the footprint ellipse.

   The shell script runs RadSim in 9 configurations for RTTOV cloudy simulations for MSG-4 SEVIRI IR channels:
   1. On the model grid.
   2. Using obs file a) without footprints enabled.
   3. Using obs file a) with footprints enabled.
   4. Using obs file c) without footprints enabled.
   5. Using obs file c) with footprints enabled.
   6. Using obs file b) with footprints enabled and writing out radiances (rather than BTs).
   7. On the model grid writing out radiances.
   8. Using obs file d) with footprints enabled and writing out footprint data using batches
   9. Using obs file d) with footprints enabled and reading footprint data from 8 using batches of a different size to those used in 8.

   The script confirms that runs 1-3 give identical results, runs 4-5 give identical results, and runs 8-9 give identical results (to within expected tolerances in each case): these comparisons are made using the netCDF *nccmp* utility. A separate Python script (*radsim_footprint_validation.py*) is used to compute footprint-averaged radiances from run 7 and ensures they are identical (to within expected tolerances) to the RadSim results from run 6.

4. Satellite and solar angle calculation validation

RadSim calculates satellite zenith and azimuth angles in the *radsim_geo_obs.py* script which generates obs datafiles for GEO sensors, and it calculates solar zenith and azimuth angles internally using the location and date/time of observations.

These calculations are validated by comparing the computed angles to those calculated by the Met Office operational Satellite Processing System (SPS). This comparison is performed using the test script *radsim_test_angle_calculations.sh*. It requires NWP model fields and a "slotstore" file from SPS containing the latitudes, longitudes and satellite and solar angles on the UM model grid. The test script uses *radsim_geo_obs.py* to generate an obs datafile for the grid points, runs RadSim using this, and compares the angles from RadSim with those from SPS. The obs data file is generated in netCDF format so this scenario also tests ingest of netCDF obs data files.

The solar angle calculation in RadSim follows that in SPS very closely and as such the differences in solar zenith angle should be very small (e.g. less than 0.2°). The differences are due primarily to the fact that RadSim operates at a resolution of minutes, while SPS performs calculations at a resolution of seconds. Differences in solar azimuth angles should generally be small, but at the subsolar point the angles are very sensitive to small differences in the calculation inputs and so points near the sub-solar point will exhibit larger differences.

The satellite zenith angle calculation in RadSim differs slightly in certain assumptions to that in SPS and as such the differences are slightly larger than for the solar zenith angle but should still be well below 1°. Similar considerations apply to the azimuth angle, with differences generally being small, but larger differences again being observed close to the sub-satellite point.

The aim of this test is to ensure that the angle calculations are producing sensible values. This test is run for one compiler with OpenMP.

A second RadSim run is performed computing the geostationary satellite angles internally at run-time instead of supplying them via the obs data file. The computed angles are compared with those from the first step. The test is successful if the angles from the two runs are identical to within expected tolerances (very small differences are observed due to differences in precision).

5. Miscellaneous options test

This test scenario is intended to check miscellaneous capabilities of RadSim. These are tested via a script *radsim_test_misc_options.sh*. Currently it runs two tests:

- test *use_all_atlas_months* option: simulations are run for an NWP SAF profile dataset with a MW sensor with this option set to true and false. The test is successful if the outputs differ.
- test specification of SSU PMC CO2 cell pressure specification: RadSim is run for non-PMC-shift SSU coefficients, and then for PMC-shift SSU coefficients

with no cell pressures specified, with the default (nominal) cell pressures for the given sensor specified, and finally with modified cell pressure specified. The outputs of the first three of these simulations should be identical and the results of the final test should differ.

The script runs the simulations, compares the outputs, and reports success or failure as appropriate.

6. Comparisons to previous version (RadSim v3.2)

The intention behind this test scenario is to ensure that each new RadSim version behaves in the same way as the previous one in respect of capabilities implemented in both versions. This test involves comparing outputs from test scenario 1 for the new version of RadSim with those from the previous version to ensure that differences are within expected tolerances. The simulations are configured to be as consistent as possible between the two versions.

Usually, it is possible to obtain identical or near-identical outputs from consecutive RTTOV releases. However, RTTOV v14.0 does not replicate outputs from RTTOV v13.2. The differences between RTTOV v13.2 and v14.0 have been assessed and are documented in the RTTOV v14 Science and Validation Report. In this case we expect larger differences than usual for this test.

For RadSim v4.1 and later, it will be possible to configure RadSim to replicate results from the previous release (e.g., v4.0) for all simulation types/configurations supported by both RadSim versions so outputs should be essentially identical.

7. Batching test

A script (*radsim_test_batching.sh*) has been created which performs a simple run for clear-sky ATMS simulations with 150000 random observations using model fields on a 1 degree lat/lon grid. All profiles are simulated in a single profile batch and all outputs are written in one output batch. The test is repeated with *max_profs* (maximum number of input profiles per batch) set to 70000 yielding three profile batches. The *max_array_size* (maximum size of data per output batch) is set to 1MB yielding 14 output batches. The two output files are compared using *nccmp* to ensure they are identical.

Note that the batching features are not intended to be triggered by such small numbers of profiles, the default values are much higher, but this test is run to verify that batching does not in any way affect the results by doing comparisons with single batch runs.

8. Heavy load test

Two tests are run to see how the Radiance Simulator performs under extreme conditions. These tests are run with one compiler with OpenMP enabled: completion of the RadSim runs without errors implies success.

The first test is to run clear-sky SEVIRI simulations (all 12 channels) for a very large number of data points (of order 10000000) corresponding to a full disc scan as generated by the new geostationary observation Python script. The obs data file is output in netCDF format so this scenario also tests ingest of netCDF obs data files. Such a large number of profiles would normally result in an excessive memory requirement to hold the interpolated model data, but the Radiance Simulator batches the input profiles to reduce this.

The second test is to run clear-sky IASI simulations (all channels) on a global model grid at a lat/lon resolution of 0.5 degrees (roughly 2.6×10^5 grid points) with emissivity output switched on. This would normally result in an excessive memory requirement to hold the calculated output data (over 16GB), but the Radiance Simulator batches the processing to reduce this. Note that this is separate and in addition to the batching of input profiles, each profile batch may have multiple processing batches depending on output data requirements.

9. Valgrind/memcheck test

The purpose of this test is to check RadSim for memory leaks and other memory errors. The test is run for one compiler (typically gfortran) *without* OpenMP for both RadSim and RTTOV. RadSim is compiled with the *-g* flag to include debugging symbols so that Valgrind can report the source (line number) of memory leaks and other issues.

The *bin/radsim_run* script is edited to run valgrind when the RadSim executable is called. For example:

```
valgrind --leak-check=full --show-reachable=yes \
/full/path/to/radsim/bin/radsim.exe $RADSIM_NL
```

The general validation test script (test scenario 1) is then run (without calls to the RadSim/RTTOV validation script) and the resulting Valgrind/memcheck output is examined to ensure no memory leaks are reported. Note that calls to the ecCodes library result in a large number of allocations classed as “still reachable” which affect all tests involving ingest of GRIB data: these are not related to RadSim and do not necessarily indicate a problem (see the Valgrind documentation). The ICON, HARMONIE, and JMA model data tests are excluded as GRIB input is covered by the ECMWF tests.

10. Timing tests

In most cases, the majority of the RadSim run-time is dominated by the calls to RTTOV, and since each version of RadSim is based on a different version of RTTOV, a comparison of RadSim run-times amounts to a comparison of RTTOV run-times. Each RTTOV release is compared to the previous one in terms of performance and the reports are available from the RTTOV web pages. However, some basic configurations are tested for RadSim v3.2 and v4.0 and the total run-times are recorded. In addition, an indicative timing for a footprint simulation is given as these are significantly more expensive. Note that timings are given for single-threaded runs: in practice use of multiple threads via OpenMP is recommended and reduces run-time significantly.

The timing simulations use ECMWF GRIB data for the region bounded by latitudes 45N to 65N and longitudes 15W to 5E (covering the UK and Ireland) at a resolution of 0.04 degrees. The grid has 251001 points. The observation data file is generated using the new GEO obs data file tool and represents MSG-4 SEVIRI pixels over this same area (though the nadir footprint radius is doubled to 3km to ensure the footprints are not empty at this model grid resolution): this comprises 126178 observations. Tests are run for SNPP ATMS and MSG-4 SEVIRI.

Timing tests performed:
- clear sky grid, ATMS
- cloudy grid, ATMS
- cloudy grid, SEVIRI IR channels
- cloudy obs with temporal interpolation, SEVIRI IR channels
- cloudy obs with temporal interpolation, SEVIRI VIS channels (MFASIS-NN)
- clear sky obs with temporal interpolation, SEVIRI VIS/IR channels
- clear sky obs with temporal interpolation and footprints, SEVIRI VIS/IR channels, writing footprint data
- clear sky obs with temporal interpolation and footprints, SEVIRI VIS/IR channels, reading footprint data from previous run

All tests are run for RadSim v3.2 and v4.0. The same RTTOV optical depth coefficients are used in v3.2 and v4.0, and RadSim/RTTOV options are selected to ensure the comparisons are fair between the two versions.

This test is intended to give an idea of run-times for RadSim v3.2 and v4.0, and since the majority of the run-time comes from RTTOV, there are no criteria for success/failure.

2.4 Portability tests

The general validation test (test scenario 1) is repeated for each of the following compilers and platforms:

- *ifort* v*17.0.1* on x86_64 virtual desktop running Red Hat Enterprise Linux version 7
- *pgfortran v18.7.0* on x86_64 virtual desktop running RHEL7
- *gfortran v11.2.0 and v8.1.0* on x86_64 virtual desktop running RHEL7
- *nagfor v6.1* on x86_64 virtual desktop running RHEL7
- *ifort v18.0.5* on Cray XC40 running Cray Linux

Results for all tests are cross-checked for consistency. The other validation tests are performed for a single compiler except where noted otherwise above. The RadSim v4.0 Test Log provides full details of all platforms and compiler flags tested.

2.5 Beta testing

RadSim v3 users were emailed a request for volunteer beta testers. Seven users responded to this call, but in the end only two submitted feedback. The aim is for beta testers to trial RadSim v4 in their existing application and assess how it compares to RadSim v3. They are also requested to test any new features in RadSim v4 which are relevant to their application. They are asked to assess RadSim v4 in terms of ease of use of the code and to verify that the relevant functionality works successfully. In addition, testers are asked to critically review the User Guide and other supporting documentation. Where significant problems are identified, the tester will be asked to re-test the code after changes have been made to confirm satisfactory operation.

Answers to the following questions will be collected and recorded below.

1. Is the software installation straightforward?
2. What hardware platform, operating system, and compiler did you use to build the executable? What versions of external libraries were used? If you ran any of the Python scripts, which version of Python are you using?
3. Did you encounter any problems building the package on your system?
4. How easy was it to set up and run the code? Did you encounter any problems?
5. Do you think the results are acceptable? Where relevant, how do they compare to RadSim v3 (bearing in mind the differences between RTTOV v13 and v14)?
6. Is the User Guide [RD-3] clear and comprehensive?
7. Do you have any suggestions for improvements to the software or documentation?

The beta testers were also asked to provide an overview of what features of RadSim they used such as which type(s) of NWP model fields, what types of RTTOV simulations, etc.

2.6 User Documentation

The Radiance Simulator documentation will be reviewed in-house by one or more people not on the development team.

3 REQUIREMENTS TRACEABILITY MATRIX

This section demonstrates how the requirements listed in section 8 of the Product Specification relate to the test plan.

Table 1: Requirements traceability

Requirement	**Testing method**	**Test plan reference**	**Test Scenarios**
RQ-4.1 Ingest of model fields	Test run	2.3	1, 9
RQ-4.2 Ingest of obs data files	Test run	2.3	1, 4, 8, 9
RQ-4.3 Spatial and temporal interpolation	Test run	2.3	1, 9
RQ-4.4 Footprint simulations	Test run	2.3	1, 3, 9
RQ-4.5 Calculate solar angles	Test run	2.3	4
RQ-4.6 Prepare/run RTTOV	Test run	2.3	1, 9
RQ-4.7 Generate outputs	Test run	2.3	1, 5, 6, 9

RQ-4.8 GEO obs data file tool	Test run	2.3	4, 8
RQ-8.1 Functionality	See RQ-4.1 - RQ-4.7 above	2.3	See above
RQ-8.2 Compilation / user instructions	Test install, beta test	2.2, 2.3, 2.4, 2.5	1
RQ-8.3 Test scripts successful	Test install, beta test	2.2, 2.5	N/A
RQ-8.4 Multi-threading via OpenMP	Test run	2.3	2
RQ-8.5 Performance	Test run	2.3	10
RQ-8.6 Portability	Test install, beta test	2.2, 2.3, 2.4, 2.5	1
RQ 8.7 Resources	Test run	2.3	7, 8
RQ-8.8 Language	Inspection, compiler flags	2.1, 2.4	N/A
RQ-8.9 Documentation	Review, beta test	2.5, 2.6	N/A
RQ-8.10 Interface requirements	Test install, beta test	2.2, 2.4, 2.5	N/A

To make clear the traceability of testing performed on new features in RadSim v4, the following table lists the major new capabilities (see the Release Note [RD-4]), the requirement(s) under which they fall, and the test scenarios in which they are tested. Where test scenario 1 is indicated, the new capability has been added to the testing carried out by the *radsim_test.sh* script described in detail above, and the relevant outputs are validated against external RTTOV simulations by the *radsim_rttov_compare.py* script.

New capability	**Requirement(s)**	**Test Scenarios**
Options for nearest neighbour spatial and temporal interpolation	RQ-4.3	1, 9
Output of clear-sky radiances for hydrometeor scattering simulations	RQ-4.7	1, 9
Output of geometric heights of both pressure half- and full-levels	RQ-4.7	1, 9
Ingest of pressure half- and/or full-levels	RQ-4.1	1, 9
2m T and q as optional inputs	RQ-4.1	1, 9
Reading a subset of forecast times from UM fieldsfiles	RQ-4.1	1, 9
Ingest of ICON-ART model data including aerosol fields	RQ-4.1	1, 9
Ingest of 60L MACC, 60L CAMS, and 137L CAMS profile datasets	RQ-4.1	1, 9
Implementation of RTTOV v14.0 interface	RQ-4.6	1, 9
New multi_surface option for heterogeneous surfaces	RQ-4.6	1, 9
Support for Zeeman-enabled simulations	RQ-4.2, RQ-4.6	1, 9
Output of RTTOV quality flags	RQ-4.7	1, 9
Obs data files in netCDF format	RQ-4.2	4, 8
Make ecCodes an optional dependency	RQ-8.2	1
Linking of RTTOV against an external LAPACK library (when RTTOV was compiled against one)	RQ-8.2	1

4 TEST RESULTS

4.1 Summary

The RadSim v4.0 Test Log is provided as a separate Annex to this document. It contains details on the specific tests performed including:

- information on compilers and compiler flags

- versions of external libraries
- portability test results
- basic installation test results
- test results for all test scenarios described above
- timing test results

The testing overall was successful and did not indicate any issues with the package.

4.2 Beta Testing

Volunteer beta testers were solicited from RadSim v3 users. Seven users expressed interest in beta testing RadSim, but in the end only two users provided any feedback after an 8-week beta test period.

Below are the (anonymised) email conversations with the beta testers who provided feedback. NWP SAF responses are provided in **red**.

4.2.1 Tester 1

1. Is the software installation straightforward?

Yes, it is. I compiled RTTOV v14 using the same compiler and compiler flags as for Radsim v13 :

- gfortran

-FFLAGS_NETCDF = -D_RTTOV_NETCDF -I$(NETCDF_PREFIX)/include

- LDFLAGS_NETCDF = -L$(NETCDF_PREFIX)/lib -lnetcdff -lnetcdf

- LDFLAGS_HDF5 = -L$(HDF5_PREFIX)/lib -lhdf5_hl -lhdf5 -lz

2. What hardware platform, operating system, and compiler did you use to build the executable? What versions of external libraries were used? If you ran any of the Python scripts, which version of Python are you using?

- Linux lisa220x 5.4.217-1.el8.elrepo.x86_64 #1 SMP Thu Oct 6 13:08:00 EDT 2022 x86_64 x86_64 x86_64 GNU/Linux
- gfortran
- netcdf-fortran-4.4.5,eccodes-2.12.5,hdf5-1.10.4,

3. Did you encounter any problems building the package on your system?

I had to export HDF5_DISABLE_VERSION_CHECK=1, before building the package.

[NWP SAF: this is most likely due to a conflict between the version of HDF5 used to compile RTTOV/RadSim (or the netCDF library), and the version being linked to dynamically at run-time.]

4. How easy was it to set up and run the code ? Did you encounter any problems?

Quite easy to run. To run my own experiment, I simply modify the radsim_check_install.nl input file. I've done exactly the same with Radsim v3.2 with the equivalent RTTOV options (cf user-guide).

5. Do you think the results are acceptable? Where relevant, how do they compare to RadSim v3.x? It would be helpful here if you can give a brief overview of how you used RadSim including the types of simulation you are running (e.g., which sensor(s), clear-sky, scattering, etc), what source of model data you are using, and which RadSim capabilities you are using.

- exp1 : using profiles from ECMWF (1440 x 721), sensor = Seviri

IR channels (micro-m)	Min of v3.2 -v4 (K)	Max of v3.2 -v4 (K)
3.92	-9.377716	8.544891
6.25	-1.0881042	0.63513184
7.35	-1.1123505	2.2615662
8.70	-4.94635	6.87442
9.66	-2.2637634	4.569397
10.8	-1.9293518	7.216217
12.0	-1.3746643	7.393341

[NWP SAF: the differences above are a little larger than we would expect for equivalent configurations. After discussion it was discovered that the user had not realised that some default options had changed (see comment below under question 6). When the test was repeated with equivalent configurations, the differences below were obtained which are in accordance with expectations.]

Updated results with equivalent configurations of v3.2 and v4.0:

IR channels (micro-m)	Min of v3.2 -v4 (K)	Max of v3.2 -v4 (K)
3.92	-1.1538086	4.8895874
6.25	-1.1372833	0.58813477
7.35	-1.2114105	1.6463318
8.70	-1.2491913	5.4564514
9.66	-0.75631714	3.7227478
10.8	-1.330307	5.607605
12.0	-1.3339386	5.6925354
13.4	-1.1640625	2.846405

Computation time :

RadSim v3.2 Chou-scaling –> 26 minutes
RadSim v4.0 delta-Eddington – > 48 minutes
RadSim v4.0 Chou-scaling – > 36 minutes
RadSim v4.0 Chou-scaling –> 29 minutes with col_threshold = 1.E-5.

6. Is the User Guide clear and comprehensive?

Yes. In Annex I, I would just add the default values for radsim v13 when it differs.

[NWP SAF: changes to default values have been highlighted in the table in Annex I which gives the mapping for v3.2 configuration namelist variables to v4.0 variables.]

4.2.2 Tester 2

1. Is the software installation straightforward?

Yes, the build and installation was straightforward. The required amendments to the Makefile.local file were easy to implement as described in the RTTOV User Guide.

2. What hardware platform, operating system, and compiler did you use to build the executable? What versions of external libraries were used? If you ran any of the Python scripts, which version of Python are you using?

- Met Office XCE/XCF machines: SUSE Linux Enterprise Server 11 (x86_64)
- Intel compiler 18.0.5
- Libraries:
  - HDF 1.8.13
  - Netcdf 4.3.2
  - ecCodes 2.6.0a
  - jasPer 1.900.1
- Python: 3.6.6

3. Did you encounter any problems building the package on your system?

No.

4. How easy was it to set up and run the code? Did you encounter any problems?

The RadSim User guide provided clear instructions. Annex I was particularly helpful!

I initially encountered the following error related to the "Pressure half-levels (rho levels)" (stash 407) fields:

```
fatal error in module rttov_check_profiles.F90:0353
pressure half-levels must be monotonically increasing from TOA down to
surface (profile number =           1)
```

To resolve this, I overwrote the "lblev" attribute for the "Pressure half-levels (rho levels)" (stash 407) fields so that level 71 was relabelled 1, level 70 was relabelled 2, …, level 1 was relabelled 71. This resolved this error but I am unsure if this was the right approach to take. It seems a bit odd that the numbering of "lblev" mattered for the half-levels but not for the full-levels.

[NWP SAF: in this application the UM fieldsfiles are provided to this system from a specific internal data feed. They are then pre-processed before being passed into RadSim. This issue is not related to RadSim, but to the data feed and/or the pre-processing.]

Roadblocks in processing Met Office UM fieldsfiles:

- Stash code 407 is not available from MOGREPS-UK model fieldsfiles.
- Stash code 407 is not available from UKV fieldsfiles with validity times at 30 mins past the hour (polar simulated imagery is generated for half-hourly leadtimes).
- Transitioning from using DOM (for AQUA) and MFASIS-LUT (all geostationary satellites) to MFASIS-NN:
    - MFASIS-NN coefficients are not available for SEVIRI channel 2.
    - AQUA MODIS channel 2 and NOAA20 channel 7 **visible channel processing failures**:
        - Attempting to generate simulated imagery from UKV or global models results in a segmentation fault/core dump.
        - RadSim has appeared to run successfully from UKV fieldsfiles for both AQUA and NOAA20 visible channels. However, this is only the case when all valid obs have invalid solar zenith angle (i.e. at nighttime)!
        - Simulated imagery has been successfully generated from GOES16 channel 3 and HIM9 channel 4 using the same configuration so it appears that the problem is specific to these polar satellites.
    - Visible imagery generation has been tested using the same configuration bar replacing MFASIS-NN with the DOM solver. For AQUA MODIS channel 2 and MSG SEVIRI channel 3, imagery generation is successful. So, if necessary, we could use this approach.

[NWP SAF: after discussion I added an option to compute approximate rho-level (half-level) pressures from theta-level (full-level) pressures if the model data file does not contain the rho levels. This is not recommended but is convenient for situations like this where users do not have the rho levels. In this case, the upstream data feeds will be updated in due course to include pressures on rho levels.

The failures were tracked down to running out of stack memory which is a known issue with the ifort compiler. This is simple to mitigate by modifying the environment to increase the size of the stack. I have added a note on this in section 3 of the user guide.

MFASIS-NN in RTTOV v14.0 does not support SEVIRI channel 2, but this is planned for v14.1 and will then be available in RadSim v4.1.]

5. Do you think the results are acceptable? Where relevant, how do they compare to RadSim v3.x? It would be helpful here if you can give a brief overview of how you used RadSim including the types of simulation you *are running (e.g., which sensor(s), clear-sky, scattering, etc), what source of model data you are using, and which RadSim capabilities you are using.*

Comparison of Global simulated MSG3 SEVIRI and UKV simulated AQUA MODIS imagery, relative to RadSim v3.1. Both RadSim instances run in a similar configuration: the Chou thermal solver for IR imagery, Baran2014 (IR) and Baum (vis) ice properties, seaice threshold of 1.0. The same BRDF and emissivity atlas (UW IR emissivity atlas) used by both (albeit in differing formats). For AQUA, both RadSim instances use the DOM solar solver for visible imagery; for MSG, RadSim v3.1 uses MFASIS LUT while RadSim v4 uses the DOM solar solver.

- Good consistency for IR imagery: high clouds fractionally less bright
- Good consistency for AQUA VIS imagery:
    - land surface somewhat brighter
    - high clouds fractionally less bright
- Differences more pronounced for MSG3 VIS imagery: reduced contrast in RadSim v4 imagery, with significantly less bright cloud. I hope that the differences might be reduced if we were able to run MFASIS-NN (if SEVIRI channel 2 is added to the coefficients file).

[NWP SAF: it is expected that MFASIS-LUT vs MFASIS-NN results will be largest. MFASIS-NN has better accuracy in reproducing the training simulations so this should be an improvement.]

6. Is the User Guide clear and comprehensive?

Yes, in general very clear and comprehensive!

7. Do you have any suggestions for improvements to the software or documentation?

The RTTOV User Guide doesn't appear to include specifics of the netcdf and hdf5 library versions, although they are provided in the RadSim User Guide.

[NWP SAF: the RTTOV user guide does mention netCDF v4.1 or higher is required in section 5.1. The HDF5 library is no longer an explicit requirement of RTTOV (or RadSim). It is used by the netCDF4 library. I have therefore removed the reference to a particular HDF5 version from the RadSim documentation as it is not relevant.]

We would appreciate the MFASIS-NN coefficients being made available for SEVIRI channel 2.

We would appreciate support to enable AQUA MODIS channel 2 and NOAA20 channel 7 processing using MFASIS-NN. Likely user-error at my end but perhaps some extra guidance would be helpful here.

[NWP SAF: as noted above, SEVIRI channel 2 will be supported in RTTOV v14.1 / RadSim v4.1, and the MFASIS failures were resolved.]

4.2.3 Summary

The following is a summary of the outcomes of the RadSim v4.0 beta testing:

- No major issues found, differences to RadSim v3.2 noted by users are within expected tolerances.
- Updated code to compute approximate UM rho (half-)level pressures when only theta (full-)level pressures are supplied.
- Updated user guide to highlight changes to namelist variable defaults.
- Updated user guide to recommend increasing stack memory in case of segmentation faults when using ifort builds.
- Updated user guide to remove reference to specific HDF5 library version.