Radiance Simulator v3 Top-level Design

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 December 2016, 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
0.1	15/01/2021	J. Hocking	Initial version
0.2	04/02/2021	J. Hocking	Footprints modelled as ellipses, ingest of JMA data
0.3	31/03/2021	J. Hocking	Updates after internal review

Table of Contents

1. INTRODUCTION

This document defines the top-level design for Version 3 of the Radiance Simulator, in accordance with the requirements of the NWP SAF.

1.1 Reference documents

[RD-1] NWPSAF-MO-DS-039, RTTOV Product Specification
[RD-2] NWPSAF-MO-DS-041, Radiance Simulator Product Specification
[RD-3] NWPSAF-MO-DS-051, Radiance Simulator User Guide

1.2 Design Drivers

Version 3 of the Radiance Simulator extends the capabilities of version 2 based on requests from users. As described in [RD-2], the RadSim v2.x minor releases made available additional capabilities of RTTOV including simulations of visible/near-IR channels and simulations using the Principal Components-based models implemented in RTTOV. Each release of RadSim is based on the latest version of RTTOV and exploits the new features available.

The key updates for RadSim v3 are as follows:

- Interface with RTTOV v13 enabling support for new RTTOV features.
- New optional outputs: additional Jacobians, height assignments consistent with the NWP SAF Cloud and Aerosol Detection Software package, and geometric altitudes of pressure levels as computed by RTTOV.
- Capability for satellite footprint simulations whereby output radiances are computed as the mean simulated radiance over the grid points falling within an ellipse of specified dimensions around the observation location. This is intended for use with convective-scale model fields i.e. where grid boxes are much smaller than satellite footprints.
- Orbit simulation capability for geostationary sensors.
- Ingest of fields from the HARMONIE model in GRIB format.
- Ingest of fields from JMA in GRIB format (clear-sky simulations only currently).

The preferred practice for implementing new features in RadSim is to integrate them into the existing code framework with as few changes as possible. Many updates do not require a large development effort, such as adding new outputs, or adding ingest of data from a new NWP model if the format is already supported, as in the case of GRIB for the HARMONIE fields and JMA data. Updating the RTTOV interface to that of the latest model is also usually a straightforward task.

The footprint simulation capability is implemented by maintaining separately the list of observations (as specified in the input obs data file) and the list of profiles to be simulated. A mapping is stored indicating to which observation each simulated profile pertains. In this way, multiple profiles (at model grid points) may be simulated for each observation. This design choice was made as it requires only relatively minor changes elsewhere in the code to accommodate the distinction between the list of observations and the list of profiles being simulated. For non-footprint simulations, the two lists are the same.

The orbit simulation capability requires additional information about the specific satellite in question. The simplest approach here seemed to be to create an external tool which would be used to generate an obs data file based on some ancillary data for the given satellite. This requires no changes to RadSim itself and avoids complicating the RadSim code unnecessarily. For geostationary (GEO) sensors, the observations are typically made for a fixed set of pixels (or fields of view) at specific locations whereas the observation locations for sensors in low-Earth orbits are much more complicated to calculate, requiring detailed knowledge of the individual orbit and sensor characteristics. The initial capability therefore focuses on GEO sensors: in principle, any GEO sensor can be supported. It requires an input netCDF data file containing at least the latitudes and longitudes of each pixel or field of view, ideally in the order in which they are scanned, and optionally other information. The tool then generates an obs data file for the full disc scan (or a subset thereof). The most significant task of the tool is the calculation of the satellite zenith and azimuth angles, and it also provides options for specifying the pixel scan times (relevant for runs using temporal interpolation and/or solar radiation). Sample netCDF files will be made available for download from the RadSim website for a selection of sensors (e.g. MSG-4 SEVIRI, GOES-16/17 ABI and Himawari-8 AHI). Files for other sensors may be made available later, and users can create their own.

2. SOFTWARE FUNCTIONS

The main components of the Radiance Simulator are:

- **Input** of control, observation and model data from external files
- **Field conversions** – calculation of humidity, identification of grid points with footprints, interpolation to observation locations and times etc
- **Initialisation and set-up** of the radiative transfer model (RTTOV);
- **Processing** - performing the simulation; looping over all observations, transferring model data to RT model profiles and calling the RT model.
- **Output** of the simulated radiances and other data.

The sequence of main functions is represented in Figure 1 below.

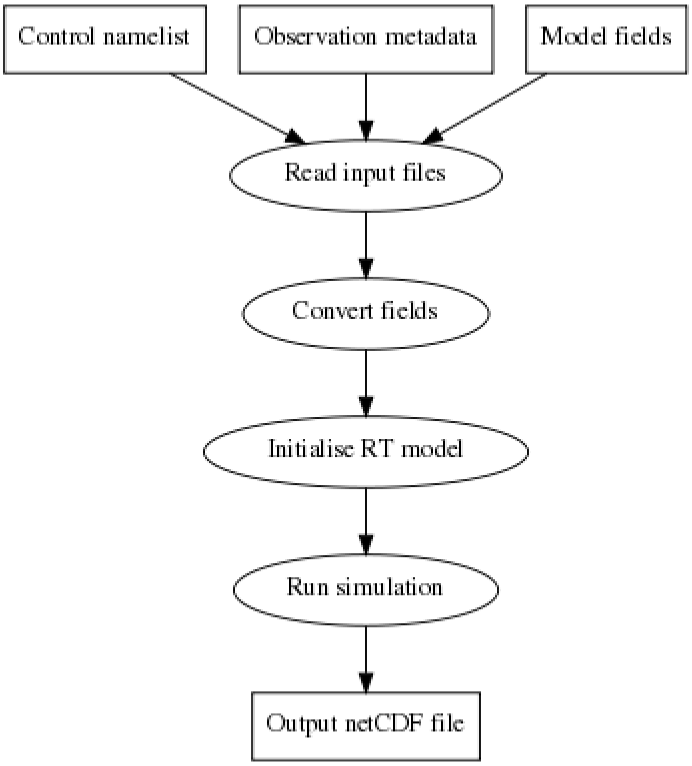

Figure 1: Main functions, input and output of the Radiance Simulator.

3. DATA FLOWS

3.1 Input, Initialisation and Set-up.

The simulations are controlled via a Fortran namelist configuration file. Users can run RadSim by calling a script which sets up environment variables and calls the RadSim executable. In this case the path to the namelist file is supplied by the user on the command line or in an environment variable. Alternatively, a Python script is available which can be used to generate a namelist file and then run RadSim using this namelist in a single call from the command line.

As a minimum, users will need to specify the following in the namelist file:

- Path to the model field data file.
- The location of the RTTOV coefficients file directories
- Satellite instrument identifiers (consistent with those used by RTTOV)

Optional configuration includes:

- Channels to use
- Path to an observation data file
- Run time options including cloudy VIS/IR and MW scattering simulations
- Output definitions including output file location and variables to be written out

Model fields can be provided in several formats:

- GRIB (data from ECMWF, DWD ICON model, HARMONIE model, data from JMA)
- NetCDF (data from ECMWF)
- Met Office fieldsfiles/PP files
- NWPSAF diverse profile datasets (60, 91 or 137 level sets are supported)

The optional observation data file allows simulations to be performed at observation locations (otherwise they are done at model grid/profile locations). The file contains latitude, longitude, satellite zenith angle and other meta-data for each observation (one observation per row). More details can be found in the User Guide [RD-3].

3.2 Processing / Simulation

The main processing steps are:

- Read in model field data or profile dataset
- Perform variable conversions as required. Conversions supported are: θ to T; dewpoint to specific humidity; relative humidity to specific humidity (and vice versa); orography to land/sea mask; rain/snow flux to mixing ratio.
- Set up RTTOV model
- Read in observation data file (if required)
- Identify grid points lying within each observation footprint (if required)
- Interpolate fields to the observation locations (bilinear interpolation for rectangular lat/lon grids, otherwise the nearest grid point to each observation is used) and times (linear interpolation).
- Rotate grid (if required)
- Calculate solar zenith and azimuth angles (if required)
- Call RTTOV for each profile to be simulated
- Set QC flags (to indicate valid RT calculation, observation within model domain, etc)
- Perform any calculations required for requested outputs (such as mean radiances over footprints and CADS height assignments)

3.3 Output

- Write output to a netCDF file

4. IMPLEMENTATION ASPECTS

4.1 Distribution

Source files will be distributed as a gzipped tar file. The file contains: Fortran source code and interface files, an installation script with configuration files for different compilers, example namelist and observation data files, a basic test script to test for successful implementation, Python scripts (for generating RadSim namelists and running RadSim from the command line, generating observation data files for geostationary sensors, and plotting datasets from RadSim output files), user documentation, and a readme file containing instructions on how to install and run the code.

4.2 External libraries

Fortran versions of the following libraries are required. These are straightforward to download and install if they are not already available. See the User Guide [RD-3] for more details.

- RTTOV (version 13)
- ecCodes (version 2.0.0 or later)
- netCDF (version 4.0 or later)
- HDF5 (version 1.8.8 or later) *

* Note: it is not always necessary to specify the path to the HDF5 library, this is platform dependent.

4.3 Compiler requirements

One of the following Fortran compilers is required:

- ifort - v15.0.0 or later
- pgfortran - v18.7-0 or later
- gfortran - v4.8.5 or later

Further details on compiler support are given in the User Guide (RD-3). Older compiler versions may work, but those given above are the versions tested and hence supported.

The Python scripts require Python v3.7 or later (they may also be compatible with earlier versions of Python3).

4.4 External file interfaces

The Radiance Simulator contains code to read NWP model fields from the following types of file:

- GRIB (ECMWF, DWD ICON, HARMONIE, JMA)
- NetCDF (ECMWF)
- Met Office fieldsfiles / PP files
- NWP SAF 60L profile dataset from ECMWF analyses
- NWP SAF 91L profile dataset from ECMWF analyses
- NWP SAF 137L profile dataset from ECMWF analyses

Other inputs are a Fortran namelist file for run-time options and an ASCII file containing observation meta-data. The ASCII file is of a format specific to the Radiance Simulator.

Output is to a netCDF file. The format is described in the User Guide [RD-3].

The Python script for generating observation data files for geostationary sensors requires ancillary data related to the specific sensor. This is provided in a netCDF file, the format of which is described in the User Guide [RD-3].

4.5 Build method

An installation (build) script is provided. Users need to specify the compiler and the location of external libraries in a configuration file before running the script.

4.6 Run-time considerations

It is possible to configure the Radiance Simulator to process extremely large datasets that may exceed the capabilities of the host platform. Built-in batching mechanisms usually mitigate the problem enough such that users are able to run these simulations successfully, but care should be taken when selecting output options. For example, if running simulations for a hyper-spectral sounder like IASI, channel-dimensioned arrays such as emissivity require huge amounts of memory, and also disk space if they are to be stored in the output file. There are two parameters that can be tuned to assist in getting the code to run to completion in the event of memory limitations – the maximum number of user-defined data points (observations) and the maximum array size for output data. These parameters define the sizes of the batches that are used in processing. The footprint simulation capability can also be demanding in terms of memory requirements, and this is controlled by reducing the number of observations processed per batch. Further details are given in the User Guide [RD-3].

Users are recommended to compile both RTTOV and RadSim with OpenMP support to take advantage of multiple threads. This is beneficial for all RadSim runs with a substantial number of profiles to simulate (e.g. thousands or more) since the radiative transfer calculations in RTTOV are usually the most time-consuming part of a RadSim run. When footprint simulations are enabled in RadSim, the process of identifying the model grid points within each footprint in RadSim is expensive and this can also take advantage of multiple threads via OpenMP. This is noted in the User Guide [RD-3].