ecRad radiation scheme: User Guide (draft)

Robin J. Hogan
European Centre for Medium Range Weather Forecasts, Reading, UK

Document version 1.1.0 (August 2018) applicable to *ecRad* version 1.1.x

If you have any queries about *ecRad* that are not answered by this document or by the information on the *ecRad* web site (`https://confluence.ecmwf.int/display/ECRAD`) then please email me at `r.j.hogan@ecmwf.int`.

Contents

1 Introduction **2**

1.1 What is *ecRad*? 2

1.2 License 2

1.3 Overview of this document 2

2 Using the offline radiation scheme **3**

2.1 Compiling the package 3

2.2 Running the offline radiation scheme 4

2.3 Configuring the radiation scheme 6

2.4 Configuring the offline package 9

Chapter 1

Introduction

1.1 What is *ecRad*?

ecRad is an atmospheric radiation scheme designed for computing profiles of solar (or *shortwave*) and thermal infrared (or *longwave*) irradiances from the surface up to the middle mesosphere. It is incorporated into the Integrated Forecasting System (IFS), the weather forecast model used operationally by the European Centre for Medium Range Weather Forecasts (ECMWF), in which it is used to compute radiative heating and cooling rates of the atmosphere and surface. An offline version of the scheme is available for educational and non-commercial research uses only.

A scientific overview of *ecRad* was provided by Hogan and Bozzo (2018). It incorporates the Rapid Radiative Transfer Model for GCMs (RRTMG; Iacono et al., 2008) for representing absorption by atmospheric gases and a flexible treatment of the optical properties of aerosol particles. Three different solvers capable of representing the effects of subgrid cloud structure: McICA (Pincus et al., 2003), Tripleclouds (Shonk and Hogan, 2008) and SPARTACUS (Hogan et al., 2016). It is coded in Fortran 2003 in a way that is efficient and flexible.

1.2 License

The Meteorological Services of ECMWF Member States are permitted to use *ecRad* (along with any IFS software) for any in-house purpose. Other researchers may apply for an institutional license to use the software, as described on the *ecRad* web site[*]. The license is the same as for the OpenIFS software and permits educational and non-commercial research use only, and prohibits redistribution. The full terms of the license are available on the *ecRad* web site.

1.3 Overview of this document

Chapter 2 describes how to compile and use the offline version of *ecRad*, which is essentially a Unix program that reads a configuration file and a NetCDF file containing a description of the atmospheric state, and outputs a NetCDF file containing the computed irradiance profiles.

At some future date this document will be expanded to include chapters describing the API (for incorporating *ecRad* into a larger Fortran program), the internal architecture and the detailed scientific documentation.

[*] `https://confluence.ecmwf.int/display/ECRAD`

Chapter 2

Using the offline radiation scheme

2.1 Compiling the package

The offline version of *ecRad* is designed to be used on a Unix-like platform. You will need a Fortran compiler that supports the 2003 standard such as `gfortran`. As a prerequisite, you will need to install the NetCDF library, including the Fortran interface (packages to install on a Linux system are typically called `libnetcdff-dev` or `libnetcdff-devel`). To run some of the tests, you will also need to install the `nco` utilities for manipulating NetCDF Files.

First unpack the package and enter the subdirectory as follows:

```
tar xvfz ecrad-1.1.0.tar.gz
cd ecrad-1.1.0
```

On a non-GNU platform you may need to untar and unzip the package using the `tar` and `gunzip` commands separately. The `README` file contains concise instructions on compilation and testing, while the `NOTICE` file outlines the license conditions. The subdirectories are as follows:

`radiation` The *ecRad* souce code for atmospheric radiation

`radsurf` The *ecRad* source code for surface radiation (under development)

`ifsaux` Source code providing a (sometimes dummy) IFS environment

`ifsrrtm` The IFS implementation of the RRTMG gas optics scheme

`utilities` Source code for useful utilities, such as reading NetCDF files

`driver` The source code for the offline driver program `ecrad`

`ifs` Source files from the IFS that are used to illustrate how *ecRad* can be incorporated into a large model, but note that these files are not used in the offline version

`mod` Where Fortran module files are written

`lib` Where the static libraries are written

`bin` Where the executable `ecrad` is written

`data` Contains configuration data files read at run-time

`test` Test cases including Matlab code to plot the outputs

`include` Automatically generated interface blocks for non-module routines

Compilation on different platforms using different compilers is facilitated by the various `Makefile_include.<prof>` files in the top-level directory: if you type

```
make
```

or

```
make PROFILE=gfortran
```

the code will be compiled using the `gfortran` compiler via the Makefile variables set in the `Makefile_include.gfortran` file. If everything goes to plan this should create the executable `bin/ecrad` and various static libraries in the `lib` directory. The other profiles are `pgi` (for the PGI compiler), `ecmwf` and `uor` (more specific configurations for the ECMWF and University of Reading computer systems). If you wish to add a new configuration for a different compiler, the cleanest way is to create a new `Makefile_include.<prof>` file modelled on one of the existing ones.

To compile in single precision, type

```
make PROFILE=gfortran SINGLE_PRECISION=1
```

To compile with debugging options (no optimization, bounds checking and initializing real numbers with not-a-number), type

```
make PROFILE=gfortran DEBUG=1
```

Finer tuning may be achieved by specifying the optimization and debugging flags explicitly, for example

```
make PROFILE=gfortran OPTFLAGS="-O1" DEBUGFLAGS="-g1 -pg"
```

Remember that if you change the compile settings you will probably want to recompile everything, in which case you first need to remove all compiled files with

```
make clean
```

2.2 Running the offline radiation scheme

To test the code, type

```
make test
```

which runs `make` in each of the subdirectories of the `test` directory. The `README` files in these directories provide more information on what they are doing, and some Matlab scripts are provided to visualize the outputs.

You will see in the output of the tests the command line in each invocation of *ecRad*, which is of the form

```
ecrad config.nam input.nc output.nc
```

where `ecrad` needs to be the full path to the *ecRad* executable, `config.nam` is a Fortran namelist file configuring the code, `input.nc` contains the input atmospheric profiles and `output.nc` contains the output irradiance (flux) profiles. The namelist file contains a `radiation` namelist that configures the *ecRad* scheme itself; the parameters available are described in section 2.3. The file also contains a `radiation_config` namelist that configures aspects of the offline package. Only the `radiation` namelist is used when *ecRad* is incorporated into an atmospheric model.

The input NetCDF file contains numerous floating-point variables listed in Table 2.1. The dimensions are shown in the order that they are listed by the `ncdump` utility, with the first dimension varying slowest in the file. Note that this is opposite to the internal Fortran ordering. Most variables are stored as a function of column and level (dimensions named `col` and `level` in Table 2.1, although the actual dimension names are ignored by `ecrad`). The `half_level` dimension corresponds to the mid-points of the levels, plus the top-of-atmosphere and surface, and so must be one more than `level`. The `level_interface` dimension excludes the top-of-atmosphere and surface so must be one less than `level`. The optional `sw_albedo_band` and `lw_emiss_band` dimensions allow for shortwave albedo and longwave emissivity to be specified in user-defined spectral intervals, but in the offline code these are ignored: the first element along these dimensions will be used for the entire shortwave and longwave spectrum.

Table 2.1: Main variables contained in the input NetCDF file to *ecRad*. Note that some variables are not required if they are not used by the particular solver selected, for example `iseed` is only needed if the McICA solver is specified. Also, only one of `o3_mmr` and `o3_vmr` should be provided. In addition to ozone, further gases can be specified in either mass mixing ratio (suffix `_mmr`) or volume mixing ratio (suffix `_vmr`) units, where the prefixes are `co2` (carbon dioxide), `n2o` (nitrous oxide), `co` (carbon monoxide), `ch4` (methane), `o2` (molecular oxygen), `cfc11` (CFC-11), `cfc12` (CFC-12), `hcfc22` (HCFC-22), `ccl4` (carbon tetrachloride) and `no2` (nitrogen dioxide). These further trace gases may either be specified as variable (dimensioned `col, level`) or constant (a scalar value in the file).

Variable	Dimensions	Description
`solar_irradiance`	-	Solar irradiance at Earth's orbit (W m^{-2})
`skin_temperature`	`col`	Skin temperature (K)
`cos_solar_zenith_angle`	`col`	Cosine of solar zenith angle
`sw_albedo`	`col, sw_albedo_band`	Shortwave albedo (if 1D then assumed spectrally constant)
`lw_emissivity`	`col, lw_emiss_band`	Longwave emissivity (if 1D then assumed spectrally constant)
`iseed`	`col`	Seed for McICA random-number generator (double precision)
`pressure_hl`	`col, half_level`	Pressure at half levels (Pa)
`temperature_hl`	`col, half_level`	Temperature at half levels (K)
`q`	`col, level`	Specific humidity (kg kg^{-1})
`o3_mmr`	`col, level`	Ozone mass mixing ratio (kg kg^{-1})
`o3_vmr`	`col, level`	Ozone volume mixing ratio (m^3 m^{-3}), used only if `o3_mmr` not provided
`q_liquid`	`col, level`	Liquid cloud mass mixing ratio (kg kg^{-1})
`q_ice`	`col, level`	Ice cloud mass mixing ratio (kg kg^{-1})
`re_liquid`	`col, level`	Liquid cloud effective radius (m)
`re_ice`	`col, level`	Ice cloud effective radius (m)
`overlap_param`	`col, level_interface`	Cloud overlap parameter
`fractional_std`	`col, level`	Fractional standard deviation of cloud optical depth
`inv_cloud_effective_size`	`col, level`	Inverse of cloud effective horizontal size for SPARTACUS solver (m^{-1})
`aerosol_mmr`	`col, aerosol_type, level`	Aerosol mass mixing ratio (kg kg^{-1})

All the test data store input fields in order of increasing pressure, i.e. starting at the top-of-atmosphere and working down to the surface. The output data are then provided using the same convention. If input data are provided in the opposite order then this should be automatically detected and under the bonnet the order is reversed before being passed to the radiation scheme. But if you use this convention then please test the results carefully as this option is not regularly tested.

The output NetCDF file contains the typical set of variables listed in Table 2.2. Clear-sky fluxes (i.e. computed on the same input profiles but in the absence of clouds) are provided if the `do_clear` namelist parameter is set to `true` (see section 2.3) while surface spectral shortwave surface fluxes are provided if `do_surface_sw_spectral_flux` is set to `true`. Note that if you want atmospheric heating rates then you will need to compute them yourself from the flux profiles.

Table 2.2: Variables contained in the output NetCDF file from *ecRad*, where all fluxes (or irradiances) have units of W m^{-2}. The `band_sw` dimension has the same size as the number of shortwave bands in the gas-optics scheme.

Variable	Dimensions	Description
`pressure_hl`	`col, half_level`	Pressure at half levels (Pa)
`flux_up_sw, flux_dn_sw`	`col, half_level`	Up- and downwelling shortwave fluxes
`flux_up_sw_clear, flux_dn_sw_clear`	`col, half_level`	Up- and downwelling clear-sky shortwave fluxes
`flux_dn_direct_sw`	`col, half_level`	Direct component of downwelling shortwave flux
`flux_dn_direct_sw_clear`	`col, half_level`	Direct component of downwelling clear-sky shortwave flux
`flux_up_lw, flux_dn_lw`	`col, half_level`	Up- and down-welling longwave fluxes
`flux_up_lw_clear, flux_dn_lw_clear`	`col, half_level`	Up- and down-welling clear-sky longwave fluxes

`lw_derivative`	`col, half_level`	Derivative of upwelling longwave flux with respect to surface value (Hogan and Bozzo, 2015)
`spectral_flux_dn_sw_surf`	`col, band_sw`	Downwelling surface shortwave flux in each band
`spectral_flux_dn_direct_sw_surf`	`col, band_sw`	Direct downwelling surface shortwave flux in each band
`spectral_flux_dn_sw_surf_clear`	`col, band_sw`	Clear-sky downwelling surface shortwave flux in each band
`spectral_flux_dn_direct_sw_surf_clear`	`col, band_sw`	Clear-sky direct downwelling surface shortwave flux in each band
`cloud_cover_sw`	`col`	Total cloud cover diagnosed by shortwave solver
`cloud_cover_lw`	`col`	Total cloud cover diagnosed by longwave solver

2.3 Configuring the radiation scheme

The detailed settings of *ecRad* are configured using the `radiation` namelist in the namelist file provided as the first command-line argument to the `ecrad` executable. The available namelist parameters are listed in Table 2.3. One of the most important is `directory_name`, which provides the absolute or relative path to the directory containing all the configuration files. This is the `data` directory at the top level of the *ecRad* package. Note that the default values listed in Table 2.3 may differ in some cases from the values used operationally in the IFS (see Table 2 of Hogan and Bozzo, 2018).

Table 2.3: Options for the `radiation` namelist that configures the radiation scheme. The type of each parameter can be inferred from its name: logicals begin with `do_` or `use_`, integers start with `i_` or `n_`, strings end with `_name`, and all other parameters are real numbers.

Parameter	**Default value**, other values	Description
General		
`directory_name`	**.**	Directory containing NetCDF configuration files
`do_sw`	**`true`**	Compute shortwave fluxes?
`do_lw`	**`true`**	Compute longwave fluxes?
`do_sw_direct`	**`true`**	Do direct shortwave fluxes?
`do_clear`	**`true`**	Compute clear-sky fluxes?
Gas and aerosol optics		
`gas_model_name`	**RRTMG-IFS**, Monochromatic	Gas optics model
`use_aerosols`	**`false`**	Do we represent aerosols?
`do_lw_aerosol_scattering`	**`true`**	Do longwave aerosol scattering?
`n_aerosol_types`		Number of aerosol types
`i_aerosol_type_map`		Vector of integers that map from aerosol types to types in the NetCDF aerosol optics file, where positive integers indexe hydrophobic types, negative integers index hydrophilic types and zero indicates a type should be ignored
Monochromatic scheme		
`mono_lw_wavelength`	**-1.0**	Wavelength of longwave radiation, or if negative, a broadband calculation will be performed
`mono_lw_total_od`	**0.0**	Zenith longwave optical depth of clear-sky atmosphere
`mono_sw_total_od`	**0.0**	Zenith shortwave optical depth of clear-sky atmosphere
Cloud optics		
`liquid_model_name`	**SOCRATES**, Slingo, Monochromatic	Liquid optics model, including the scheme in the SOCRATES radiation scheme and the older scheme of Slingo (1989)
`ice_model_name`	**Fu-IFS**, Baran2016, Yi, Monochromatic	Ice optics model, including the schemes of Fu (1996), Fu et al. (1998), Baran et al. (2016) and Yi et al. (2013)
`do_lw_cloud_scattering`	**`true`**	Do longwave cloud scattering?
`do_fu_lw_ice_optics_bug`	**`false`**	Reproduce bug in McRad implementation of Fu ice optics?

Solver		
`sw_solver_name`	Homogeneous, **McICA**, Tripleclouds, SPARTACUS	Shortwave solver
`lw_solver_name`	Homogeneous, **McICA**, Tripleclouds, SPARTACUS	Longwave solver
`overlap_scheme_name`	Max-Ran, **Exp-Ran**, Exp-Exp	Cloud overlap scheme
`use_beta_overlap`	**`false`**	Use Shonk et al. (2010) ‘β’ overlap parameter definition, rather than default ‘α’?
`cloud_inhom_decorr_scaling`	**0.5**	Ratio of overlap decorrelation lengths for cloud inhomogeneities and boundaries
`cloud_fraction_threshold`	$\mathbf{10^{-6}}$	Ignore clouds with fraction below this
`cloud_mixing_ratio_threshold`	$\mathbf{10^{-9}}$	Ignore clouds with total mixing ratio below this
`cloud_pdf_shape_name`	**Gamma**, Lognormal	Shape of cloud water PDF
`cloud_pdf_override_file_name`		Name of NetCDF file of alternative cloud PDF look-up table
`do_sw_delta_scaling_with_gases`	**`false`**	Apply delta-Eddington scaling to particle-gas mixture, rather than particles only
SPARTACUS solver		
`do_3d_effects`	**`true`**	Represent 3D effects when SPARTACUS solver selected
`n_regions`	2, **3**	Number of regions, where one is clear sky and one or two are cloud (the Tripleclouds solver always assumes three regions regardless of this parameter)
`do_lw_side_emissivity`	**`true`**	Represent effective emissivity of the side of clouds (Schäfer et al., 2016)
`sw_encroachment_name`	Minimum, **Computed**, Maximum	Encroachment (or ‘entrapment’) model (note that the behaviour in ecRad version 1.0.1 was ‘Maximum’ encroachment)
`do_3d_lw_multilayer_effects`	**`false`**	Maximum encroachment for longwave radiation?
`max_3d_transfer_rate`	**10.0**	Maximum rate of lateral exchange between regions in one layer, for stability of matrix exponential (where the default means that as little as e^{-10} of the radiation could remain in a region)
`max_gas_od_3d`	**8.0**	3D effects ignored for spectral intervals where gas optical depth of a layer exceeds this, for stability
`max_cloud_od`	**18.0**	Maximum in-cloud optical depth, for stability
`use_expm_everywhere`	**`false`**	Use matrix-exponential method even when 3D effects not represented?
`clear_to_thick_fraction`	**0.0**	Fraction of cloud edge interfacing directly to the most optically thick cloudy region
`overhead_sun_factor`	**0.0**	Minimum tan-squared of solar zenith angle to allow some ‘direct’ radiation from overhead sun to pass through cloud sides (0.06 used by Hogan et al., 2016)
Surface (under development)		
`use_canopy_full_spectrum_sw`	**`false`**	Perform canopy shortwave radiative transfer at full atmospheric spectral resolution
`use_canopy_full_spectrum_lw`	**`false`**	Perform canopy longwave radiative transfer at full atmospheric spectral resolution
Diagnostics		
`iverbosesetup`	0, 1, 2, **3**, 4, 5	Verbosity in setup, where 1=warning, 2=info, 3=progress, 4=detailed, 5=debug
`iverbose`	0, **1**, 2, 3, 4, 5	Verbosity in execution
`do_save_spectral_flux`	**`false`**	Save flux profiles in each band?
`do_save_gpoint_flux`	**`false`**	Save flux profiles in each g-point?
`do_surface_sw_spectral_flux`	**`true`**	Save surface shortwave fluxes in each band for subsequent diagnostics?
`do_lw_derivatives`	**`false`**	Compute derivatives for Hogan and Bozzo (2015) approximate updates?
`do_save_radiative_properties`	**`false`**	Write intermediate NetCDF file(s) of properties sent to solver (`radiative_properties*.nc`)?

When calling *ecRad* from within a model, the parameters listed in Table 2.3 are members of the `config_type` structure, and may be modified within the code at the appropriate place in the configuration stage. The exception is in the case of strings, which are prefixed by `_name` in the namelist. In the `config_type` structure there are equivalent integers to express these parameters, which can be changed using the named constants listed in Table 2.4.

Table 2.4: Integers in the `config_type` structure that represents the strings in Table 2.3, where a namelist parameter named `*_name` would be named `i_*` here.

Variable in `config_type`	Available named constants, **default**
`i_overlap_scheme`	`IOverlapMaximumRandom`, **`IOverlapExponentialRandom`**, `IOverlapExponential`
`i_solver_sw, i_solver_lw`	`ISolverHomogeneous`, **`ISolverMcICA`**, `ISolverSpartacus`, `ISolverTripleclouds`
`i_3d_sw_encroachment`	`IEncroachmentMinimum`, **`IEncroachmentComputed`**, `IEncroachmentMaximum`
`i_gas_model`	`IGasModelMonochromatic`, **`IGasModelIFSRRTMG`**
`i_liq_model`	`ILiquidModelMonochromatic`, **`ILiquidModelSOCRATES`**, `ILiquidModelSlingo`
`i_ice_model`	`IIceModelMonochromatic`, **`IIceModelFu`**, `IIceModelBaran2016`, `IIceModelYi`
`i_cloud_pdf_shape`	**`IPdfShapeGamma`**, `IPdfShapeLognormal`

When `ecrad` is run with the default verbosity settings, it outputs to the screen a summary of the configuration options, the files read and written and details of the aerosol mapping. The following is an example from the default test in the `test/ifs` directory.

```
-------------------------- OFFLINE ECRAD RADIATION SCHEME --------------------------
Copyright (C) 2014-2018 European Centre for Medium-Range Weather Forecasts
Contact: Robin Hogan (r.j.hogan@ecmwf.int)
Floating-point precision: double
General settings:
  Data files expected in "../../data"
  Clear-sky calculations are ON                           (do_clear=T)
  Saving intermediate radiative properties OFF            (do_save_radiative_properties=F)
  Saving spectral flux profiles OFF                       (do_save_spectral_flux=F)
  Saving surface shortwave spectral fluxes ON             (do_surface_sw_spectral_flux=T)
  Gas model is RRTMG-IFS                                  (i_gas_model=1)
  Aerosols are ON                                         (use_aerosols=T)
  Longwave derivative calculation is ON                   (do_lw_derivatives=T)
Cloud settings:
  Cloud fraction threshold = .100E-05                     (cloud_fraction_threshold)
  Cloud mixing-ratio threshold = .100E-08                 (cloud_mixing_ratio_threshold)
  Liquid optics scheme is SOCRATES                        (i_liq_model=2)
  Ice optics scheme is Fu-IFS                             (i_ice_model=2)
  Longwave ice optics bug in Fu scheme is OFF             (do_fu_lw_ice_optics_bug=F)
  Cloud overlap scheme is Exp-Exp                         (i_overlap_scheme=2)
  Use "beta" overlap parameter is OFF                     (use_beta_overlap=F)
  Cloud PDF shape is Gamma                                (i_cloud_pdf_shape=1)
  Cloud inhom decorrelation scaling = .500                (cloud_inhom_decorr_scaling)
Solver settings:
  Shortwave solver is McICA                               (i_solver_sw=1)
  Shortwave delta scaling after merge with gases OFF      (do_sw_delta_scaling_with_gases=F)
  Longwave solver is McICA                                (i_solver_lw=1)
  Longwave cloud scattering is ON                         (do_lw_cloud_scattering=T)
  Longwave aerosol scattering is OFF                      (do_lw_aerosol_scattering=F)
Reading ../../data/RADRRTM
Reading ../../data/RADSRTM
Reading NetCDF file ../../data/socrates_droplet_scattering_rrtm.nc
Reading NetCDF file ../../data/fu_ice_scattering_rrtm.nc
Reading NetCDF file ../../data/aerosol_ifs_rrtm_43R3.nc
```

```
Aerosol mapping:
   1 -> hydrophilic type 1: Sea salt, bin 1, 0.03-0.5 micron, OPAC
   2 -> hydrophilic type 2: Sea salt, bin 2, 0.50-5.0 micron, OPAC
   3 -> hydrophilic type 3: Sea salt, bin 3, 5.0-20.0 micron, OPAC
   4 -> hydrophobic type 1: Desert dust, bin 1, 0.03-0.55 micron, (SW) Dubovik et al 2002, (LW)...
   5 -> hydrophobic type 2: Desert dust, bin 2, 0.55-0.90 micron, (SW) Dubovik et al 2002, (LW)...
   6 -> hydrophobic type 3: Desert dust, bin 3, 0.90-20.0 micron, (SW) Dubovik et al 2002, (LW)...
   7 -> hydrophilic type 4: Hydrophilic organic matter, OPAC
   8 -> hydrophobic type 10: Hydrophobic organic matter, OPAC (hydrophilic at RH 20-30%)
   9 -> hydrophobic type 11: Black carbon, Olivier Boucher
  10 -> hydrophobic type 11: Black carbon, Olivier Boucher
  11 -> hydrophilic type 5: Ammonium sulfate, GACP
  12 -> hydrophobic type 14: Stratospheric sulfate (hydrophilic ammonium sulfate at RH 20-30%)
Reading NetCDF file ../../data/mcica_gamma.nc
Reading NetCDF file ecrad_meridian.nc
  Warning: variable co_vmr not found
  Warning: variable no2_vmr not found
Writing NetCDF file inputs.nc
Thread 0 processing columns 1-32
Writing NetCDF file ecrad_meridian_default_out.nc
------------------------------------------------------------------------------------
```

2.4 Configuring the offline package

In addition to the namelist parameters described in section 2.3 an additional set of parameters are available in the `radiation_config` namelist that are specific to the offline version of *ecRad* and are listed in Table 2.5.

Table 2.5: Options for the `radiation_config` namelist that configures additional aspects of the offline radiation scheme. All entries must be scalars. If an override parameter is present then it need not be included in the input file. The cloud effective sizes (used by the SPARTACUS solver) may be specified for low, middle and high clouds according to the cloud layer pressure p and the surface pressure p_0.

Parameter	Description
Execution control	
`nrepeat`	Number of times to repeat, for benchmarking
`istartcol`	Start at specified input column (1 based)
`iendcol`	End at specified input column (1 based)
`iverbose`	Verbosity in offline setup (default 2)
`do_parallel`	Use OpenMP parallelism? (default `true`)
`nblocksize`	Number of columns per block when using OpenMP
`do_save_inputs`	Sanity check: save input variables in `inputs.nc`
Override input variables	
`solar_irradiance_override`	Override solar irradiance (W m^{-2})
`skin_temperature`	Override skin temperature (K)
`cos_solar_zenith_angle`	Override cosine of solar zenith angle
`sw_albedo`	Override shortwave albedo
`lw_emissivity`	Override longwave emissivity
`fractional_std`	Override cloud optical depth fractional standard deviation
`overlap_decorr_length`	Override cloud overlap decorrelation length (m)
`inv_effective_size`	Override inverse of cloud effective size (m^{-1})
`low_inv_effective_size`	...for low clouds ($p > 0.8p_0$)
`middle_inv_effective_size`	...for mid-level clouds ($0.45p_0 < p \leq 0.8p_0$)
`high_inv_effective_size`	...for high clouds ($p \leq 0.45p_0$)
Scale input variables	
`q_liquid_scaling`	Scaling for liquid water mixing ratio
`q_ice_scaling`	Scaling for ice water mixing ratio
`cloud_fraction_scaling`	Scaling for cloud fraction (capped at 1)
`overlap_decorr_length_scaling`	Scaling for cloud overlap decorrelation length
`effective_size_scaling`	Scaling for cloud effective size

Bibliography

Baran, A. J., P. Hill, D. Walters, S. C. Hardiman, K. Furtado, P. R. Field and J. Manners, 2016: The impact of two coupled cirrus microphysicsradiation parameterizations on the temperature and specific humidity biases in the tropical tropopause layer in a climate model. *J. Climate,* **29,** 5299–5316.

Fu, Q., 1996: An accurate parameterization of the solar radiative properties of cirrus clouds. *J. Climate,* **9,** 2058–2082.

Fu, Q., P. Yang and W. B. Sun, 1998: An accurate parametrization of the infrared radiative properties of cirrus clouds of climate models. *J. Climate,* **11,** 2223-2237.

Hogan, R. J., and A. Bozzo, 2015: Mitigating errors in surface temperature forecasts using approximate radiation updates. *J. Adv. Model. Earth Syst.,* **7,** 836–853.

Hogan, R. J., and A. Bozzo, 2018:A flexible radiation scheme for the ECMWF model. *J. Adv. Model. Earth Syst.,* **10,** doi:10.1029/2018MS001364.

Hogan, R. J., S. A. K. Schäfer, C. Klinger, J.-C. Chiu and B. Mayer, 2016: Representing 3D cloud-radiation effects in two-stream schemes: 2. Matrix formulation and broadband evaluation. *J. Geophys. Res.,* **121,** 8583–8599.

Iacono, M. J., J. S. Delamere, E. J. Mlawer, M. W. Shephard, S. A. Clough and W. D. Collins, 2008: Radiative forcing by longlived greenhouse gases: Calculations with the AER radiative transfer models. *J. Geophys. Res.,* **113,** D13103, doi: 10.1029/2008JD009944.

Pincus, R., H. W. Barker, and J.-J. Morcrette, 2003: A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous clouds. *J. Geophys. Res. Atmos.,* **108,** 4376, doi:10.1029/2002JD003322.

Schäfer, S. A. K., R. J. Hogan, C. Klinger, J.-C. Chiu and B. Mayer, 2016: Representing 3D cloud-radiation effects in two-stream schemes: 1. Longwave considerations and effective cloud edge length. *J. Geophys. Res.,* **121,** 8567–8582.

Shonk, J. K. P., and R. J. Hogan, 2008: Tripleclouds: an efficient method for representing horizontal cloud inhomogeneity in 1D radiation schemes by using three regions at each height. *J. Climate,* **21,** 2352–2370.

Shonk, J. K. P., R. J. Hogan, J. M. Edwards and G. G. Mace, 2010: Effect of improving representation of horizontal and vertical cloud structure on the Earth's radiation budget: 1. Review and parameterisation. *Q. J. R. Meteorol. Soc.,* **136,** 1191–1204.

Slingo, A., 1989: A GCM parametrization for the shortwave radiative properties of water clouds. *J. Atmos. Sci.,* **46,** 1419–1427.

Yi, B., P. Yang, B. A. Baum, T. L'Ecuyer, L. Oreopoulos, E. J. Mlawer, A. J. Heymsfield and K.-K. Liou, 2013: Influence of ice particle surface roughening on the global cloud radiative effect. *J. Atmos. Sci.,* **70,** 2794–2807.