

ecRad radiation scheme: User Guide

Robin J. Hogan

European Centre for Medium Range Weather Forecasts, Reading, UK

Document version 1.5 (June 2022) applicable to *ecRad* version 1.5.x

This document is copyright © European Centre for Medium Range Weather Forecasts 2018–. 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 <i>ecRad</i> ?	2
1.2	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
2.2.1	Input file format	5
2.2.2	Output file format	6
2.3	Configuring the radiation scheme	7
2.3.1	Configuring aerosol optical properties	10
2.3.2	Configuring surface albedo and emissivity	10
2.3.3	Configuring cloud optical properties	11
2.3.4	Configuring the gas-optics model	12
2.4	Configuring the offline package	12
2.5	Describing cloud structure	13
2.6	Checking the configuration	15
3	Incorporating <i>ecRad</i> into another program	20

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. It is also used operationally by the German Weather Service in the ICON model. An offline version of the scheme is available under the open source Apache License, version 2, from these locations:

- <https://confluence.ecmwf.int/display/ECRAD>
- <https://github.com/ecmwf-ifs/ecrad>

A scientific overview of *ecRad* was provided by [Hogan and Bozzo \(2018\)](#), with a little more technical information available from [Hogan and Bozzo \(2016\)](#). Two gas-optics models are available: the Rapid Radiative Transfer Model for GCMs (RRTMG; [Iacono et al., 2008](#)), and the ECMWF correlated *k*-distribution scheme (ecCKD; [Hogan and Matricardi, 2022](#)). Three different solvers capable of representing the effects of subgrid cloud structure are available: McICA ([Pincus et al., 2003](#)), Tripleclouds ([Shonk and Hogan, 2008](#)) and SPARTACUS ([Hogan et al., 2016](#)). The treatment of cloud and aerosol optical properties is easily extensible. It is coded in Fortran 2003 in a way that is efficient, but also flexible in the sense that the solver and the optical models for clouds, aerosols and gases can be changed independently.

1.2 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. Chapter 3 describes how to incorporate *ecRad* into a larger Fortran program, such as an atmospheric model. At some future date this document will be expanded to include chapters describing the internal architecture and the detailed scientific documentation.

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 `libnetcdf-dev` or `libnetcdf-devel`). To run some of the tests, you will also need to install the NCO utilities for manipulating NetCDF Files.

First download and unpack the package, then enter the subdirectory; if you download the package from the *ecRad* web page then do this:

```
tar xvfz ecrad-1.x.y.tar.gz
cd ecrad-1.x.y
```

On a non-GNU platform you may need to `untar` and `unzip` the package using the `tar` and `gunzip` commands separately. If you want the most recent snapshot from GitHub then do the following:

```
git clone https://github.com/ecmwf-ifs/ecrad.git
cd ecrad
```

The `README.md` file contains concise instructions on compilation and testing, while the `NOTICE` file outlines the license conditions. The subdirectories are as follows:

radiation The *ecRad* source code for atmospheric radiation

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`

drhook Optional profiling and debugging support library used by the IFS

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

practical Practical exercises to help new users become familiar with *ecRad* as well as learning something about atmospheric radiative transfer

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. Using instead `PROFILE=pgi` will use the `Makefile_include.pgi` file to attempt to compile with the PGI compiler. If everything goes to plan this should create the executable `bin/ecrad` and various static libraries in the `lib` directory.

One common reason the code doesn't compile out of the box is that it can't find the NetCDF library files. Since version 1.2.0, the *ecRad* Makefile uses the `nf-config` script that comes with recent versions of the NetCDF library to create the Makefile variables `NETCDF_INCLUDE` and `NETCDF_LIB`. If `nf-config` is not available on your system, or it fails to correctly locate the NetCDF library files, then the cleanest way to fix this is to create a `Makefile_include.local` file that defines `NETCDF_INCLUDE` and `NETCDF_LIB` explicitly to contain arguments for the compile and link operations, respectively. Suppose you installed NetCDF in `/path/to/netcdf` and you use the `gfortran` compiler then your file might contain:

```
include Makefile_include.gfortran
NETCDF = /path/to/netcdf
NETCDF_INCLUDE = -I$(NETCDF)/include
NETCDF_LIB = -L$(NETCDF)/lib -lnetcdff -lnetcdf -Wl,-rpath,$(NETCDF)/lib
```

You should then be able to build the code with

```
make PROFILE=local
```

Examples of such configurations for the ECMWF and University of Reading computer systems may be found in `Makefile_include.ecmwf` and `Makefile_include.uor`.

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

The easiest and most fun way to become familiar with how to use *ecRad* is to do the practical exercises: enter the `practical` directory, read the `ecrad_practical.pdf` document there and follow the instructions. You will learn about how changes to atmospheric gases provide the radiative forcing that drives climate change, uncertainties in the representation of clouds in radiation schemes, and the impact of aerosols on surface fluxes and atmospheric heating rates.

To quickly test the code is compiled correctly, 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, described in section 2.4. Only the `radiation` namelist is used when *ecRad* is incorporated into an atmospheric model.

2.2.1 Input file format

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 (opposite to the Fortran convention). 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. Some variables can be omitted in which case default values will be used or these fields will be constructed according to `radiation_config` namelist parameters (section 2.4).

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 used by the McICA solver and `inv_cloud_effective_size` is only used by the SPARTACUS solver. 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), `cc14` (carbon tetrachloride) and `no2` (nitrogen dioxide). These further trace gases may either be specified as variable in space (dimensioned `col`, `level`) or constant (a scalar value in the file). To override the suffix indicating volume mixing ratio (e.g. to change it to `_mole_fraction`), set the namelist variable `vmr_suffix_str` as described in Table 2.4.

Variable	Dimensions	Description
<code>solar_irradiance</code>	-	Solar irradiance at Earth's orbit (W m^{-2})
<code>skin_temperature</code>	<code>col</code>	Skin temperature (K)
<code>cos_solar_zenith_angle</code>	<code>col</code>	Cosine of solar zenith angle
<code>sw_albedo</code>	<code>col</code> , <code>sw_albedo_band</code>	Shortwave albedo (if 1D then assumed spectrally constant)
<code>lw_emissivity</code>	<code>col</code> , <code>lw_emiss_band</code>	Longwave emissivity (if 1D then assumed spectrally constant)
<code>iseed</code>	<code>col</code>	Seed for McICA random-number generator (double precision, default: 1, 2, 3...)
<code>pressure_hl</code>	<code>col</code> , <code>half_level</code>	Pressure at half levels (Pa)
<code>temperature_hl</code>	<code>col</code> , <code>half_level</code>	Temperature at half levels (K)
<code>q</code> or <code>h2o_mmr</code>	<code>col</code> , <code>level</code>	Specific humidity (kg kg^{-1})
<code>h2o_vmr</code>	<code>col</code> , <code>level</code>	Water vapour volume mixing ratio (mol mol^{-1})
<code>o3_mmr</code>	<code>col</code> , <code>level</code>	Ozone mass mixing ratio (kg kg^{-1})
<code>o3_vmr</code>	<code>col</code> , <code>level</code>	Ozone volume mixing ratio (mol mol^{-1}), used only if <code>o3_mmr</code> not provided
<code>aerosol_mmr</code>	<code>col</code> , <code>aer_type</code> , <code>level</code>	Aerosol mass mixing ratio (kg kg^{-1})
<code>q_liquid</code>	<code>col</code> , <code>level</code>	Liquid cloud mass mixing ratio (kg kg^{-1})

<code>q_ice</code>	<code>col, level</code>	Ice cloud mass mixing ratio (kg kg^{-1})
<code>q_hydrometeor</code>	<code>hydro, col, level</code>	Hydrometeor mass mixing ratio (kg kg^{-1}), alternative to <code>q_liquid</code> and <code>q_ice</code>
<code>re_liquid</code>	<code>col, level</code>	Liquid cloud effective radius (m)
<code>re_ice</code>	<code>col, level</code>	Ice cloud effective radius (m)
<code>re_hydrometeor</code>	<code>hydro, col, level</code>	Hydrometeor effective radius (m), alternative to <code>re_liquid</code> and <code>re_ice</code>
<code>cloud_fraction</code>	<code>col, level</code>	Cloud fraction
<code>overlap_param</code>	<code>col, level_interface</code>	Cloud overlap parameter (default: compute from decorrelation length of 2 km)
<code>fractional_std</code>	<code>col, level</code>	Fractional standard deviation of cloud optical depth (default 0)
<code>inv_cloud_effective_size</code>	<code>col, level</code>	Inverse of cloud effective horizontal size for SPARTACUS solver (m^{-1})
<code>inv_inhom_effective_size</code>	<code>col, level</code>	Inverse of effective horizontal size of cloud inhomogeneities, for SPARTACUS solver (m^{-1}) (default: same as <code>inv_cloud_effective_size</code>)
<code>inv_cloud_effective_separation</code>	<code>col, level</code>	Alternative input to SPARTACUS if <code>inv_cloud_effective_size</code> not present (m^{-1})
<code>inv_inhom_effective_separation</code>	<code>col, level</code>	Alternative input to SPARTACUS if <code>inv_inhom_effective_size</code> not present (m^{-1})

Cloud properties may be specified either via `q_liquid`, `q_ice`, `re_liquid` and `re_ice`, but to support additional hydrometeor species (e.g. rain and graupel) you should instead use `q_hydrometeor` and `re_hydrometeor`. Typically the first two hydrometeor types contain liquid and ice properties, and this will be assumed if you have the older setting of `use_general_cloud_optics=false`.

All the test data in the `ecRad` package 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 variables describing cloud properties, particularly sub-grid cloud structure, are defined in detail in section 2.5.

2.2.2 Output file format

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). If you need diagnostic downward fluxes at the surface for just a subset of the spectrum (e.g. ultraviolet or photosynthetically active radiation) then they can be computed from the `spectral_flux_dn_*` variables, activated if namelist variable `do_surface_sw_spectral_flux` is set to `true`. In some contexts it is also useful to have fluxes in each of the shortwave albedo or longwave emissivity spectral intervals. These are named `canopy_flux_dn_*` and are activated if `do_canopy_fluxes_sw` or `do_canopy_fluxes_lw` are 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
<code>pressure_hl</code>	<code>col, half_level</code>	Pressure at half levels (Pa)
<code>flux_up_sw, flux_dn_sw</code>	<code>col, half_level</code>	Up- and downwelling shortwave fluxes
<code>flux_up_sw_clear, flux_dn_sw_clear</code>	<code>col, half_level</code>	Up- and downwelling clear-sky shortwave fluxes
<code>flux_dn_direct_sw</code>	<code>col, half_level</code>	Direct component of downwelling shortwave flux
<code>flux_dn_direct_sw_clear</code>	<code>col, half_level</code>	Direct component of downwelling clear-sky shortwave flux
<code>flux_up_lw, flux_dn_lw</code>	<code>col, half_level</code>	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
canopy_flux_dn_diffuse_sw_surf	col, sw_albedo_band	Downwelling diffuse surface shortwave flux in each albedo interval
canopy_flux_dn_direct_sw_surf	col, sw_albedo_band	Downwelling direct surface shortwave flux in each albedo interval
canopy_flux_dn_lw_surf	col, lw_emiss_band	Downwelling surface longwave flux in each emissivity interval
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
<i>General</i>		
<code>directory_name</code>	<code>.</code>	Directory containing NetCDF configuration files
<code>do_sw</code>	true	Compute shortwave fluxes?
<code>do_lw</code>	true	Compute longwave fluxes?
<code>do_sw_direct</code>	true	Do direct shortwave fluxes?
<code>do_clear</code>	true	Compute clear-sky fluxes?
<code>do_cloud_aerosol_per_sw_g_point</code>	true	Do we compute cloud, aerosol and surface shortwave optical properties per g point (not available for RRTMG)?
<code>do_cloud_aerosol_per_lw_g_point</code>	true	Do we compute cloud, aerosol and surface longwave optical properties per g point (not available for RRTMG)?
<i>Gas optics</i>		
<code>gas_model_name</code>	RRTMG-IFS, ECCKD, Monochromatic	Gas optics model
<code>gas_optics_sw_override_file_name</code>	(See section 2.3.4)	Path to an alternative shortwave ecCKD gas optics file
<code>gas_optics_lw_override_file_name</code>	(See section 2.3.4)	Path to an alternative longwave ecCKD gas optics file
<i>Aerosol optics</i>		
<code>use_aerosols</code>	false	Do we represent aerosols?
<code>use_general_aerosol_optics</code>	true	Support aerosol properties at an arbitrary spectral discretization (not just RRTMG)
<code>do_lw_aerosol_scattering</code>	true	Do longwave aerosol scattering?
<code>n_aerosol_types</code>		Number of aerosol types

<code>i_aerosol_type_map(:)</code>	(see section 2.3.1)	Vector of integers that map from aerosol types to types in the NetCDF aerosol optics file, where positive integers index hydrophobic types, negative integers index hydrophilic types and zero indicates a type should be ignored
<code>aerosol_optics_override_file_name</code>	(see section 2.3.1)	Path to an alternative aerosol optics file
<i>Monochromatic scheme</i>		
<code>mono_lw_wavelength</code>	-1.0	Wavelength of longwave radiation, or if negative, a broadband calculation will be performed
<code>mono_lw_total_od</code>	0.0	Zenith longwave optical depth of clear-sky atmosphere
<code>mono_sw_total_od</code>	0.0	Zenith shortwave optical depth of clear-sky atmosphere
<code>mono_lw_single_scattering_albedo</code>	0.538	Longwave cloud single scattering albedo
<code>mono_sw_single_scattering_albedo</code>	0.999999	Shortwave cloud single scattering albedo
<code>mono_lw_asymmetry_factor</code>	0.925	Longwave cloud asymmetry factor
<code>mono_sw_asymmetry_factor</code>	0.86	Shortwave cloud asymmetry factor
<i>Cloud optics</i>		
<code>liquid_model_name</code>	SOCRATES , Slingo, Monochromatic	Liquid optics model, including the scheme in the SOCRATES radiation scheme and the older scheme of Slingo (1989)
<code>ice_model_name</code>	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)
<code>use_general_cloud_optics</code>	true	Support arbitrary hydrometeor types (not just liquid and ice) with properties at an arbitrary spectral discretization (not just RRTMG)
<code>do_lw_cloud_scattering</code>	true	Do longwave cloud scattering?
<code>do_fu_lw_ice_optics_bug</code>	false	Reproduce bug in McRad implementation of Fu ice optics (Hogan et al., 2016)?
<code>liq_optics_override_file_name</code>		Path to alternative liquid optics file name
<code>ice_optics_override_file_name</code>		Path to alternative ice optics file name
<code>cloud_type_name(:)</code>	mie_droplet , baum-general-habit-mixture_ice	Optical property model name for each generalized hydrometeor species
<code>use_thick_cloud_spectral_averaging(:)</code>	true	Do we use 'thick' spectral averaging of Edwards and Slingo (1996) for each generalized hydrometeor species?
<i>Solver</i>		
<code>sw_solver_name</code>	Cloudless, Homogeneous, McICA , Tripleclouds, SPARTACUS	Shortwave solver; note that the homogeneous solver assumes cloud fills the gridbox horizontally (so ignores cloud fraction) while the cloudless solver ignores clouds completely
<code>lw_solver_name</code>	Cloudless, Homogeneous, McICA , Tripleclouds, SPARTACUS	Longwave solver
<code>overlap_scheme_name</code>	Max-Ran, Exp-Ran , Exp-Exp	Cloud overlap scheme; note that SPARTACUS and Tripleclouds only work with the Exp-Ran overlap scheme
<code>use_beta_overlap</code>	false	Use Shonk et al. (2010) ' β ' overlap parameter definition, rather than default ' α '?
<code>cloud_inhom_decorr_scaling</code>	0.5	Ratio of overlap decorrelation lengths for cloud inhomogeneities and boundaries
<code>cloud_fraction_threshold</code>	10^{-6}	Ignore clouds with fraction below this
<code>cloud_mixing_ratio_threshold</code>	10^{-9}	Ignore clouds with total mixing ratio below this
<code>cloud_pdf_shape_name</code>	Gamma , Lognormal	Shape of cloud water PDF
<code>cloud_pdf_override_file_name</code>		Name of NetCDF file of alternative cloud PDF look-up table
<code>do_sw_delta_scaling_with_gases</code>	false	Apply delta-Eddington scaling to particle-gas mixture, rather than particles only (see Hogan and Bozzo, 2018)
<i>SPARTACUS solver (these parameters have no effect for other solvers)</i>		
<code>do_3d_effects</code>	true	Represent cloud edge effects when SPARTACUS solver selected; note that this option does not affect entrainment, which is also a 3D effect

n_regions	2, 3	Number of regions used by SPARTACUS, 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_entrapment_name	Zero, Edge-only, Explicit , Non-fractal, Maximum	Entrapment model (Hogan et al., 2019); note that the behaviour in ecRad version 1.0.1 was ‘Maximum’ entrapment
do_3d_lw_multilayer_effects	false	Maximum entrapment 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	16.0	Maximum in-cloud optical depth, for stability
use_expm_everywhere	false	Use matrix-exponential method even when 3D effects not important, such as clear-sky layers and parts of the spectrum where the gas optical depth is large?
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)
overhang_factor	0.0	A detail of the entrapment representation described by Hogan et al. (2019)
<i>Surface</i>		
do_nearest_spectral_sw_albedo	false	Surface shortwave albedos may be supplied in their own spectral intervals: do we select the nearest to each band of the gas optics scheme, rather than using a weighted average?
do_nearest_spectral_lw_emiss	false	...likewise but for surface longwave emissivity
sw_albedo_wavelength_bound(:)	(see section 2.3.2)	Vector of the wavelength bounds (m) delimiting the shortwave albedo intervals
lw_emiss_wavelength_bound(:)	(see section 2.3.2)	Vector of the wavelength bounds (m) delimiting the longwave emissivity intervals
i_sw_albedo_index(:)	(see section 2.3.2)	Vector of indices mapping albedos to wavelength intervals
i_lw_emiss_index(:)	(see section 2.3.2)	Vector of indices mapping emissivities to wavelength intervals
do_weighted_surface_mapping	true	Do we weight the mapping from surface emissivity/albedo to g-point/band weighting by a reference Planck function (more accurate) or a constant weight in wavenumber (less accurate)?
<i>Diagnostics</i>		
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)?
do_canopy_fluxes_sw	false	Save surface shortwave fluxes in each albedo interval
do_canopy_fluxes_lw	false	Save surface longwave fluxes in each emissivity interval

Several aspects of Table 2.3 deserve further explanation, particularly those that are configured with vectors, and are covered in the following subsections.

2.3.1 Configuring aerosol optical properties

As shown in Table 2.1, aerosols are provided to *ecRad* in the form of the mass mixing ratios of a number of different aerosol types. The optical properties of an arbitrary number of hydrophilic and hydrophobic aerosol types are provided in a NetCDF file; for example `data/aerosol_ifs_rrtm_49R1.nc` in the *ecRad* package contains aerosol properties in the RRTMG bands, while `data/aerosol_ifs_49R1.nc` describes the properties at high spectral resolution and is used if `use_general_aerosol_optics` is `true`. The mapping between the input aerosol concentrations and the aerosol types in the optical-property file may be specified in the `radiation` namelist. The `n_aerosol_types` parameter specifies the number of aerosol concentrations to be provided, with a value of zero having the effect of deactivating aerosols. `i_aerosol_type_map` is a vector of integers of length `n_aerosol_types` indicating which aerosol type to select from the optical-property file. Negative numbers select hydrophilic types, whose optical properties vary with relative humidity, while positive numbers select hydrophobic types. Zero indicates that an input aerosol type is to be ignored. As an example, the IFS settings (in the `test/ifs` directory) are specified with:

```
aerosol_optics_override_file_name = 'aerosol_ifs_rrtm_46R1_with_NI_AM.nc'
n_aerosol_types = 12
i_aerosol_type_map = -1, -2, -3, 1, 2, 3, -4, 10, 11, 11, -5, 14
```

When *ecRad* is run, the output printed to the terminal includes a description of the aerosol mapping.

2.3.2 Configuring surface albedo and emissivity

A similar mechanism is used to describe how spectral intervals of the input `sw_albedo` and `lw_emissivity` should be interpreted. This is best explained by considering the configuration of the IFS in Cycle 47R1, which is described by the following namelist variables:

```
sw_albedo_wavelength_bound(1:5) = 0.25e-6, 0.44e-6, 0.69e-6, 1.19e-6, 2.38e-6
i_sw_albedo_index(1:6) = 1,2,3,4,5,6
do_nearest_spectral_sw_albedo = false
lw_emiss_wavelength_bound(1:2) = 8.0e-6, 13.0e-6
i_lw_emiss_index(1:3) = 1,2,1
do_nearest_spectral_lw_emiss = true
do_weighted_surface_mapping = false
```

The IFS describes surface albedo in six spectral intervals. The vector `sw_albedo_wavelength_bounds` here provides the wavelengths, in metres, of the five boundaries between these intervals, where the first interval is taken to include all wavelengths shorter than the first value (in this case $0.25 \mu\text{m}$) and the last includes all wavelengths longer than the last value (in this case $2.38 \mu\text{m}$). The vector `i_sw_albedo_index` specifies which of the elements of the input `sw_albedo` field should be used in each of the six spectral intervals. Surface emissivity is described similarly: there are three spectral intervals specified by the two boundaries in `lw_emiss_wavelength_bound`. The corresponding vector `i_lw_emiss_index` contains two occurrences of the index 1, indicating that the first element of `lw_emissivity` is used both for wavelengths smaller than $8 \mu\text{m}$ and wavelengths larger than $13 \mu\text{m}$ (i.e. outside the infrared atmospheric window). The second element is then used for wavelengths between these two boundaries. Thus even though there are three spectral intervals, only two elements are needed in `lw_emissivity`. The logicals `do_nearest_spectral_sw_albedo` and `do_nearest_spectral_lw_emiss` specify whether the bands of the gas optics scheme used in *ecRad* will use a single value of albedo or emissivity from the input fields (chosen to be the spectral interval with the largest overlap in wavenumber space with each band of the gas-optics scheme), or whether they will weight the spectral intervals by their overlap with each band of the gas-optics scheme. Finally, `do_weighted_surface_mapping` specifies whether to use a weighted mapping using the Planck function at a representative temperature of the sun or the earth in the shortwave and longwave, respectively; this is more accurate but this feature was not available before *ecRad* version 1.5 so should be `false` to reproduce earlier behaviour. The mapping from spectral interval to band is printed on standard output when *ecRad* is run, as shown in the example in section 2.6.

2.3.3 Configuring cloud optical properties

All parameterizations for the optical properties of hydrometeors are expressed in terms of effective radius, which is assumed to be defined by:

$$r_e = \frac{3V}{4A}, \quad (2.1)$$

where V is the total volume of liquid or solid ice per unit volume of air, while A is the total projected area of the particles per unit volume of air (units m^{-1}). For liquid and ice clouds this reduces to the familiar

$$r_{e,\text{liq}} = \frac{3}{4} \frac{\text{LWC}}{\rho_{\text{liq}} A_{\text{liq}}}; \quad r_{e,\text{ice}} = \frac{3}{4} \frac{\text{IWC}}{\rho_{\text{ice}} A_{\text{ice}}}, \quad (2.2)$$

where LWC and IWC are the liquid and ice water content, respectively (in kg m^{-3} , equal to the mass mixing ratio multiplied by the air density), and ρ_{liq} and ρ_{ice} are the densities of liquid water and solid ice. Effective radius is assumed to be horizontally constant within a gridbox, even if the water content varies.

Prior to version 1.5, the r_e dependence of the various optical property models for liquid droplets and ice particles were all represented parametrically in the code, with the coefficients of the parameterizations provided specifically for the spectral intervals of the RRTMG gas optics model in data files such as `socrates_droplet_scattering_rrtm.nc` and `fu_ice_scattering_rrtm.nc`. The optical models were selected via the `liquid_model_name` and `ice_model_name` namelist parameters, and could be overridden with `liq_optics_override_file_name` and `ice_optics_override_file_name`.

These various parameterizations are still available to use with the RRTMG gas optics model from version 1.5 onwards, but in order to support alternative gas optics models with a different band structure, and potentially more hydrometeor species than just liquid and ice clouds, a more flexible approach is taken when the user specifies `use_general_cloud_optics=true`. In this case the user should use the `cloud_type_name` namelist parameter to specify the cloud type and optical model of each hydrometeor species in the form of a string with the form `<OPTICAL-MODEL>_<HYDROMETEOR-TYPE>`. The default setting is equivalent to the following namelist entries:

```
cloud_type_name(1) = 'mie_droplet'
cloud_type_name(2) = 'baum-general-habit-mixture_ice'
```

The number of hydrometeor types (up to a maximum of 12) is taken from the number of `cloud_type_name` entries provided. `ecRad` then appends `_scattering.nc` to these strings and looks for files with these names in the data directory. These files contain look-up tables of the optical properties as a function of effective radius and wavenumber. At setup time, the optical properties are averaged to the spectral intervals used by the gas-optics scheme. In case above (and by default) only two cloud types are provided, so `ecRad` assumes the input fields `q_hydrometeor` and `re_hydrometeor` (see Table 2.1) contain two types, liquid and ice cloud. Data files are available in the `data` directory of the `ecRad` package to support the following optical models:

mie_droplet Cloud droplets described by Mie theory with effective radius in the range 1–50 μm ;

mie_rain Rain drops described by Mie theory with effective radius in the range 25–3000 μm ;

baum-general-habit-mixture_ice Ice particles using Baum’s ‘general habit mixture’ for effective radius in the range 5–60 μm ;

fu-muskatel_ice Unroughened ice particles using Fu’s model extended to the effective radius range 3–370 μm by Harel Muskatel;

fu-muskatel-rough_ice As above but for roughened ice particles.

Note that if the input effective radius is out of bounds, the nearest in-bound values are used.

Whether cloud optical properties will be computed per band (less accurate but required by RRTMG) or per g-point (more accurate) is governed by the `do_cloud_aerosol_per_sw_g_point` and `do_cloud_aerosol_per_lw_g_point` namelist options. By default, ‘thick averaging’ [Edwards and Slingo \(1996\)](#) is used when mapping the high spectral resolution optical properties to the bands or g-points of the gas-optics model. This was found by [Hogan and Matricardi \(2022\)](#) to give better cloud radiative effects over a wide range of cloud optical thicknesses for both ice and liquid clouds. To specify that you want the first hydrometeor type to use thick averaging and the second thin you would use:

```
use_thick_cloud_spectral_averaging(1) = true
use_thick_cloud_spectral_averaging(2) = false
```

2.3.4 Configuring the gas-optics model

By default the RRTMG gas-optics model is used, which uses 16 and 14 bands in the longwave and shortwave, respectively, containing a total of 140 and 112 g-points (spectral intervals) in the longwave and shortwave, respectively. The number of g-points determines how many quasi-monochromatic calculations are performed to cover the full broadband spectrum, and therefore the computational cost. To use the ecCKD scheme of [Hogan and Matricardi \(2022\)](#) instead, use the following namelist option

```
gas_model_name = 'ECCKD'
```

which will load the shortwave and longwave gas-optics models from NetCDF files (playing an equivalent role to the ‘spectral files’ of [Edwards and Slingo, 1996](#)). The default files are:

- ecckd-1.0_lw_climate_fsck-32b_ckd-definition.nc
- ecckd-1.0_sw_climate_rgb-32b_ckd-definition.nc

The first was generated using ecCKD version 1.0 for the longwave spectrum, uses the full-spectrum correlated- k (FSCK) band structure with a total of 32 g-points, and used both (‘b’) the ‘Evaluation-1’ and ‘Evaluation-2’ line-by-line datasets of [Hogan and Matricardi \(2020\)](#) in training the model. The second is similar for the shortwave except using the RGB band structure which contains five bands corresponding to the red, green and blue parts of the spectrum, plus bands for the UV and near infrared. Two other files are available in the ecRad data directory which can be activated with

```
gas_optics_lw_override_file_name = 'ecckd-1.2_lw_climate_narrow-64b_ckd_definition.nc'
gas_optics_sw_override_file_name = 'ecckd-1.2_sw_climate_window-64b_ckd_definition.nc'
```

These models each use a total of 64 g-points; the first uses the 13-band ‘narrow’ band structure of [Hogan and Matricardi \(2020\)](#) and the second uses a band structure containing separate bands for each of the windows in the near infrared ([Hogan and Matricardi, 2022](#)). These two models are more accurate but slower, so more suited for reference calculations. More models are available from the ecCKD web site <https://confluence.ecmwf.int/x/XwU0Dw>.

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 2.4. In general if these parameters are present in the namelist then they will override the corresponding variable provided in the input file.

Table 2.4: 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
<i>Execution control</i>	
<code>nrepeat</code>	Number of times to repeat, for benchmarking
<code>istartcol</code>	Start at specified input column (1 based)
<code>iendcol</code>	End at specified input column (1 based)
<code>iverbose</code>	Verbosity in offline setup (default 2)
<code>do_parallel</code>	Use OpenMP parallelism? (default true)
<code>nblocksize</code>	Number of columns per block when using OpenMP
<code>do_save_inputs</code>	Sanity check: save input variables in <code>inputs.nc</code>

<code>do_save_aerosol_optics</code>	Output computed aerosol optical property look-up table in <code>aerosol_optics.nc</code>
<code>do_correct_unphysical_inputs</code>	If input variables out of physical bounds, correct them and issue a warning
<code>vmr_suffix_str</code>	Suffix for variables containing volume mixing ratios (default <code>'_vmr'</code>)
<i>Override input variables</i>	
<code>solar_irradiance_override</code>	Override solar irradiance (W m^{-2})
<code>skin_temperature</code>	Override skin temperature (K)
<code>cos_solar_zenith_angle</code>	Override cosine of solar zenith angle
<code>sw_albedo</code>	Override shortwave albedo
<code>lw_emissivity</code>	Override longwave emissivity
<code>fractional_std</code>	Override cloud optical depth fractional standard deviation
<code>overlap_decorr_length</code>	Override cloud overlap decorrelation length (m)
<code>inv_effective_size</code>	Override inverse of cloud effective size (m^{-1})
<code>low_inv_effective_size</code>	...for low clouds ($p > 0.8p_0$, where p is pressure and p_0 surface pressure)
<code>middle_inv_effective_size</code>	...for mid-level clouds ($0.45p_0 < p \leq 0.8p_0$)
<code>high_inv_effective_size</code>	...for high clouds ($p \leq 0.45p_0$)
<i>Scale input variables</i>	
<code>q_liquid_scaling</code>	Scaling for liquid water mixing ratio
<code>q_ice_scaling</code>	Scaling for ice water mixing ratio
<code>cloud_fraction_scaling</code>	Scaling for cloud fraction (capped at 1)
<code>overlap_decorr_length_scaling</code>	Scaling for cloud overlap decorrelation length
<code>effective_size_scaling</code>	Scaling for cloud effective size
<code>h2o_scaling, co2_scaling...</code>	Scaling for specific humidity and carbon dioxide; equivalents available for <code>o3, co, ch4, n2o, o2, cfc11, cfc12, hfc22</code> and <code>cc14</code>
<i>Parameterize input variables</i>	
<code>cloud_inhom_separation_factor</code>	Set inhomogeneity separation scale to be this multiplied by cloud separation scale
<code>cloud_separation_scale_surface</code>	Surface cloud separation scale in pressure-dependent parameterization
<code>cloud_separation_scale_toa</code>	Top-of-atmosphere cloud separation scale in pressure-dependent parameterization
<code>cloud_separation_scale_power</code>	Power in cloud separation scale parameterization

2.5 Describing cloud structure

Probably more than any other 1D radiation scheme, *ecRad* allows the user to define in detail the statistical properties of the sub-grid cloud distribution, and in this section the relevant variables and namelist parameters are explained in more detail. In an operational context most of these variables need to be parameterized, but in developing new solvers we need to perform explicit radiation calculations on realistic high resolution 3D cloud fields, and compare them to *ecRad* simulations in which the profiles of these variables have been extracted from the 3D cloud fields. This has been done by Schäfer et al. (2016), Hogan et al. (2016) and Hogan et al. (2019). Explicit radiation calculations on a 3D cloud field can either be performed using the Independent Column Approximation (ICA) and compared to *ecRad*'s McICA or Tripleclouds solvers, or using a fully 3D solver (e.g. Monte Carlo) and comparing it to *ecRad*'s SPARTACUS solver. Note that *ecRad* can itself perform ICA calculations on 3D cloud fields, by flattening the two horizontal dimensions of a 3D dataset into a single 'column' dimension, and using the *ecRad*'s 'Homogeneous' solver in which any cloud is assumed to homogeneously fill each of the narrow columns (so cloud fraction is not used as it is implicitly taken to be 0 or 1).

The input variables describing the profile of cloud properties are given in the lower half of Table 2.1. The most basic are the liquid and ice mass mixing ratios (`q_liquid` and `q_ice`), which are gridbox-mean quantities, and the corresponding effective radii (`re_liquid` and `re_ice`) defined in section 2.3.3. These may alternatively be expressed by `q_hydrometeor` and `re_hydrometeor` if more than two hydrometeor types are to be represented.

Cloud fraction is simply the fractional horizontal area of a given model layer that contains cloud. The layers are assumed to be thin enough that cloud fraction is constant with height within a layer, i.e. cloud fraction by volume is equal to cloud fraction by area. The horizontal variability of cloud water content in a layer is specified by the fractional standard deviation (`fractional_std`), defined as the standard deviation of the in-cloud water content, divided by the in-cloud mean water content. The in-cloud mean water content is the gridbox-mean water content divided by cloud fraction. Note that since effective radius is assumed constant across a gridbox, cloud optical depth is proportional to water path and so `fractional_std` can also be thought of as the horizontal fractional standard deviation of cloud optical depth. Moreover, *ecRad* assumes that horizontal variations of liquid and ice water

content are perfectly correlated. As shown in Table 2.4, fractional standard deviation can be overridden through a namelist parameter; for example, in the IFS this value is set to 1.

Cloud overlap is needed by the Exp-Ran and Exp-Exp overlap schemes, and is specified at the interface (or half-level) between each layer by `overlap_param`, the overlap parameter as defined by Hogan and Illingworth (2000). To compute this at half-level $i + 1/2$ of a high-resolution 3D cloud field, you need the cloud fractions in the upper and lower lower layers, c_i and c_{i+1} , and the combined cloud cover of the cloud in these two layers, C . Then from Eqs. 1, 2 and 4 of Hogan and Illingworth (2000) you can compute the overlap parameter:

$$\alpha_{i+1/2} = \frac{C_{\text{rand}} - C}{C_{\text{max}} - C}, \quad (2.3)$$

where the combined cloud covers that would be obtained from the random and maximum overlap assumptions are

$$C_{\text{rand}} = c_i + c_{i+1} - c_i c_{i+1}; \quad (2.4)$$

$$C_{\text{max}} = \max(c_i, c_{i+1}). \quad (2.5)$$

Alternatively, cloud overlap can be parameterized as in most atmospheric models in terms of an overlap decorrelation length as shown in Table 2.4, which implements Eq. 5 of Hogan and Illingworth (2000). In addition to describing how cloud boundaries overlap, *ecRad* needs to know how sub-grid cloud inhomogeneities are vertically correlated. This cannot be specified at each layer, but is rather specified via the namelist variable `cloud_inhom_decorr_scaling` in Table 2.3, which gives the ratio of the decorrelation lengths for cloud inhomogeneities and cloud boundaries. The default value of 0.5 was obtained from observations of ice clouds by Hogan and Illingworth (2003).

The variables and parameters above are all used by the McICA and Tripleclouds solvers to represent cloud properties relevant for 1D radiative transfer. In order to use the SPARTACUS solver to represent 3D radiative effects, we also need a means to specify the *normalized cloud perimeter length*, L , in each model layer. If we imagine a horizontal slice through the sub-grid cloud field, then L is the total cloud perimeter length divided by the area of the domain, with units of inverse metres. This variable is not provided to SPARTACUS directly, since it tends to be strongly dependent on the cloud fraction. Rather we specify either the *cloud effective size*, C_S , or the *cloud effective separation*, C_X , which tend to be less dependent on cloud fraction. Normalized perimeter length is related to the former via Eq. 29 of Hogan et al. (2019):

$$L = 4c(1 - c)/C_S, \quad (2.6)$$

and to the latter via (Fielding et al., 2020)

$$L = 4 [c(1 - c)]^{1/2} / C_X, \quad (2.7)$$

where c is the cloud fraction. The variables $1/C_S$ and $1/C_X$ may be specified directly in the input file as `inv_cloud_effective_size` and `inv_cloud_effective_separation`, respectively. If both are present then the former will take precedence. The reason that reciprocals are provided is that then a value of zero (corresponding to C_S or C_X of infinity) indicates no 3D effects are to be simulated in a particular layer. If you have a high resolution cloud scene and you wish to run SPARTACUS on it then you need to compute the perimeter length from it (e.g. use a contouring function on a field containing 0 for clear sky and 1 for cloud, and then compute the length of the 0.5 contour), and knowing also cloud fraction you can invert (2.6) or (2.7).

In the context of an atmospheric model, we recommend that C_X is parameterized using the namelist parameters at the bottom of Table 2.4 scheme with the values of Fielding et al. (2020):

```
cloud_separation_scale_toa      = 14000.0, ! Value of C_X at top-of-atmosphere (m)
cloud_separation_scale_surface = 2500.0,  ! Value of C_X at surface (m)
cloud_separation_scale_power   = 3.5,     ! Describes pressure dependence of C_X
cloud_inhom_separation_factor  = 0.75    ! Defines size of cloud inhomogeneities
```

These numbers are used in the namelist in the `test/ifs` case. Note that the first number shown here, C_X^{TOA} , is valid for a model with a horizontal grid spacing of around 100 km, but this parameter was found by Fielding et al. (2020) to be dependent on horizontal grid spacing Δx in a way that can be fitted with

$$C_X^{\text{TOA}} = 1.62 \Delta x^{0.47}, \quad (2.8)$$

where both C_X^{TOA} and Δx are in km. The surface value of C_X can be assumed to be 2.5 km for all model resolutions.

2.6 Checking the configuration

When `ecrad` is run, it outputs to the screen a summary of the configuration options, the files read and written and details of the aerosol mapping. This can be used to check that *ecRad* has been configured as intended. The following is an example from typing

```
make test_default
```

in the `test/ifs` directory, in the case of `iverbosesetup=2` and `iverbose=1` in the radiation namelist:

```
----- OFFLINE ECRAD RADIATION SCHEME -----
Copyright (C) 2014-2020 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 ON                           (do_save_spectral_flux=T)
  Gas model is "RRTMG-IFS"                                  (i_gas_model=1)
  Aerosols are ON                                           (use_aerosols=T)
  Clouds are ON                                              (do_clouds=T)
Surface settings:
  Saving surface shortwave spectral fluxes OFF              (do_surface_sw_spectral_flux=F)
  Saving surface shortwave fluxes in abledo bands ON         (do_canopy_fluxes_sw=T)
  Saving surface longwave fluxes in emissivity bands ON      (do_canopy_fluxes_lw=T)
  Longwave derivative calculation is ON                      (do_lw_derivatives=T)
  Nearest-neighbour spectral albedo mapping OFF             (do_nearest_spectral_sw_albedo=F)
  Nearest-neighbour spectral emissivity mapping ON           (do_nearest_spectral_lw_emiss=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=2)
  Shortwave delta scaling after merge with gases OFF         (do_sw_delta_scaling_with_gases=F)
  Longwave solver is "McICA"                                 (i_solver_lw=2)
  Longwave cloud scattering is ON                            (do_lw_cloud_scattering=T)
  Longwave aerosol scattering is OFF                         (do_lw_aerosol_scattering=F)
Warning: turning on do_surface_sw_spectral_flux as required by do_canopy_fluxes_sw
Reading ../data/RADRRTM
Reading ../data/RADSRTM
Weighting of 6 albedo values in 14 shortwave bands (wavenumber ranges in cm-1):
  2600 to 3250: 0.00 0.00 0.00 0.00 0.00 1.00
  3250 to 4000: 0.00 0.00 0.00 0.00 0.00 1.00
  4000 to 4650: 0.00 0.00 0.00 0.00 0.69 0.31
  4650 to 5150: 0.00 0.00 0.00 0.00 1.00 0.00
  5150 to 6150: 0.00 0.00 0.00 0.00 1.00 0.00
  6150 to 7700: 0.00 0.00 0.00 0.00 1.00 0.00
  7700 to 8050: 0.00 0.00 0.00 0.00 1.00 0.00
  8050 to 12850: 0.00 0.00 0.00 0.93 0.07 0.00
  12850 to 16000: 0.00 0.00 0.48 0.52 0.00 0.00
  16000 to 22650: 0.00 0.00 1.00 0.00 0.00 0.00
  22650 to 29000: 0.00 0.99 0.01 0.00 0.00 0.00
  29000 to 38000: 0.00 1.00 0.00 0.00 0.00 0.00
  38000 to 50000: 0.83 0.17 0.00 0.00 0.00 0.00
  820 to 2600: 0.00 0.00 0.00 0.00 0.00 1.00
Mapping from 16 longwave bands to emissivity intervals: 1 1 1 1 2 2 2 1 1 1 1 1 1 1
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_46R1_with_NI_AM.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...
  5 -> hydrophobic type 2: Desert dust, bin 2, 0.55-0.90 micron, (SW) Dubovik et al. 2002...
  6 -> hydrophobic type 3: Desert dust, bin 3, 0.90-20.0 micron, (SW) Dubovik et al. 2002...
  7 -> hydrophilic type 4: Hydrophilic organic matter, OPAC
  8 -> hydrophobic type 10: Hydrophobic organic matter, OPAC (hydrophilic at RH=20%)
  9 -> hydrophobic type 11: Black carbon, OPAC
 10 -> hydrophobic type 11: Black carbon, OPAC
 11 -> hydrophilic type 5: Ammonium sulfate (for sulfate), GACP Lacis et al https://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
Performing radiative transfer calculations
Writing NetCDF file ecrad_meridian_default_out.nc
-----

```

Running

```
make test_ecckd_tc
```

uses the eCKD gas optics model and the Tripleclouds solver, producing the following:

```

----- OFFLINE ECRAD RADIATION SCHEME -----
Copyright (C) 2014- ECMWF
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 ON (do_save_spectral_flux=T)
  Gas model is "ECCKD" (i_gas_model=2)
  Aerosols are ON (use_aerosols=T)
  General aerosol optics ON (use_general_aerosol_optics=T)
  Clouds are ON
  Do cloud/aerosol/surface SW properties per g-point ON (do_cloud_aerosol_per_sw_g_point=T)
  Do cloud/aerosol/surface LW properties per g-point ON (do_cloud_aerosol_per_lw_g_point=T)
Surface settings:
  Saving surface shortwave spectral fluxes OFF (do_surface_sw_spectral_flux=F)
  Saving surface shortwave fluxes in abledo bands ON (do_canopy_fluxes_sw=T)
  Saving surface longwave fluxes in emissivity bands ON (do_canopy_fluxes_lw=T)
  Longwave derivative calculation is ON (do_lw_derivatives=T)
  Nearest-neighbour spectral albedo mapping OFF (do_nearest_spectral_sw_albedo=F)
  Nearest-neighbour spectral emissivity mapping OFF (do_nearest_spectral_lw_emiss=F)
  Planck-weighted surface albedo/emiss mapping ON (do_weighted_surface_mapping=T)
Cloud settings:
  Cloud fraction threshold = .100E-05 (cloud_fraction_threshold)
  Cloud mixing-ratio threshold = .100E-08 (cloud_mixing_ratio_threshold)
  General cloud optics ON (use_general_cloud_optics=T)
  Cloud overlap scheme is "Exp-Ran" (i_overlap_scheme=1)
  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 "Tripleclouds" (i_solver_sw=4)
  Shortwave delta scaling after merge with gases OFF (do_sw_delta_scaling_with_gases=F)
  Longwave solver is "Tripleclouds" (i_solver_lw=4)
  Longwave cloud scattering is ON (do_lw_cloud_scattering=T)

```

```

Longwave aerosol scattering is OFF                               (do_lw_aerosol_scattering=F)
Warning: turning on do_surface_sw_spectral_flux as required by do_canopy_fluxes_sw
Reading NetCDF file ../../data/ecckd-1.0_sw_climate_rgb-32b_ckd-definition.nc
ecCKD shortwave gas optics model: 250-50000 cm-1, 32 g-points in 5 bands
Look-up table sizes: 53 pressures, 6 temperatures, 0 planck-function entries
Gases:
  Composite of well-mixed background gases: no concentration dependence
  H2O: look-up table with 12 log-spaced mole fractions in range 0.160779E-06 to 0.160779E+00
  O3: linear concentration dependence
  CO2: linear concentration dependence
  CH4: linear concentration dependence relative to a mole fraction of 0.192100E-05
  N2O: linear concentration dependence relative to a mole fraction of 0.332000E-06
Reading NetCDF file ../../data/ecckd-1.0_lw_climate_fsck-32b_ckd-definition.nc
ecCKD longwave gas optics model: 0-3260 cm-1, 32 g-points in 1 bands
Look-up table sizes: 53 pressures, 6 temperatures, 231 planck-function entries
Gases:
  Composite of well-mixed background gases: no concentration dependence
  H2O: look-up table with 12 log-spaced mole fractions in range 0.160779E-06 to 0.160779E+00
  O3: linear concentration dependence
  CO2: linear concentration dependence
  CH4: linear concentration dependence relative to a mole fraction of 0.192100E-05
  N2O: linear concentration dependence relative to a mole fraction of 0.332000E-06
  CFC11: linear concentration dependence
  CFC12: linear concentration dependence
Surface shortwave albedo
Mapping from 6 values to 32 g-points
1: 0.00 0.00 0.00 0.00 0.72 0.28
2: 0.00 0.00 0.00 0.00 0.24 0.76
3: 0.00 0.00 0.00 0.00 1.00 0.00
4: 0.00 0.00 0.00 0.00 1.00 0.00
5: 0.00 0.00 0.00 0.14 0.86 0.00
6: 0.00 0.00 0.00 0.41 0.59 0.00
7: 0.00 0.00 0.00 1.00 0.00 0.00
8: 0.00 0.00 0.00 1.00 0.00 0.00
9: 0.00 0.00 0.00 1.00 0.00 0.00
10: 0.00 0.00 0.00 1.00 0.00 0.00
11: 0.00 0.00 0.00 1.00 0.00 0.00
12: 0.00 0.00 0.00 1.00 0.00 0.00
13: 0.00 0.00 0.00 0.29 0.56 0.15
14: 0.00 0.00 0.00 0.21 0.64 0.15
15: 0.00 0.00 0.00 0.00 0.80 0.20
16: 0.00 0.00 0.00 0.14 0.59 0.27
17: 0.00 0.00 0.00 0.00 0.00 1.00
18: 0.00 0.00 0.00 0.08 0.48 0.44
19: 0.00 0.00 0.00 0.91 0.09 0.00
20: 0.00 0.00 0.00 0.03 0.45 0.53
21: 0.00 0.00 0.00 0.00 0.13 0.87
22: 0.00 0.00 0.00 0.00 0.08 0.92
23: 0.00 0.00 0.00 0.00 0.32 0.68
24: 0.00 0.00 0.00 0.00 0.14 0.86
25: 0.00 0.00 0.00 0.00 0.00 1.00
26: 0.00 0.00 0.91 0.09 0.00 0.00
27: 0.00 0.00 1.00 0.00 0.00 0.00
28: 0.00 0.38 0.62 0.00 0.00 0.00
29: 0.00 1.00 0.00 0.00 0.00 0.00
30: 0.11 0.89 0.00 0.00 0.00 0.00
31: 0.23 0.77 0.00 0.00 0.00 0.00
32: 0.30 0.70 0.00 0.00 0.00 0.00
Surface longwave emissivity
Mapping from 2 values to 32 g-points
1: 0.03 0.97
2: 0.01 0.99
3: 0.11 0.89
4: 0.38 0.62
5: 0.90 0.10

```

```
6: 0.97 0.03
7: 0.99 0.01
8: 0.85 0.15
9: 1.00 0.00
10: 1.00 0.00
11: 0.99 0.01
12: 1.00 0.00
13: 0.98 0.02
14: 1.00 0.00
15: 1.00 0.00
16: 1.00 0.00
17: 0.93 0.07
18: 1.00 0.00
19: 1.00 0.00
20: 0.29 0.71
21: 1.00 0.00
22: 0.23 0.77
23: 1.00 0.00
24: 1.00 0.00
25: 0.02 0.98
26: 1.00 0.00
27: 1.00 0.00
28: 1.00 0.00
29: 1.00 0.00
30: 1.00 0.00
31: 1.00 0.00
32: 1.00 0.00
Shortwave cloud type 1:
Reading NetCDF file ../../data/mie_droplet_scattering.nc
  File: ../../data/mie_droplet_scattering.nc
  Weighting temperature: 5777.0 K
  SSA averaging: optically thick limit
  Spectral discretization: 32 g-points
  Effective radius look-up: 50 points in range 1.0- 50.0 um
  Wavenumber range: 250-49950 cm-1
Longwave cloud type 1:
Reading NetCDF file ../../data/mie_droplet_scattering.nc
  File: ../../data/mie_droplet_scattering.nc
  Weighting temperature: 273.1 K
  SSA averaging: optically thick limit
  Spectral discretization: 32 g-points
  Effective radius look-up: 50 points in range 1.0- 50.0 um
  Wavenumber range: 0-3250 cm-1
Shortwave cloud type 2:
Reading NetCDF file ../../data/baum-general-habit-mixture_ice_scattering.nc
  File: ../../data/baum-general-habit-mixture_ice_scattering.nc
  Weighting temperature: 5777.0 K
  SSA averaging: optically thick limit
  Spectral discretization: 32 g-points
  Effective radius look-up: 23 points in range 5.0- 60.0 um
  Wavenumber range: 250-49950 cm-1
Longwave cloud type 2:
Reading NetCDF file ../../data/baum-general-habit-mixture_ice_scattering.nc
  File: ../../data/baum-general-habit-mixture_ice_scattering.nc
  Weighting temperature: 273.1 K
  SSA averaging: optically thick limit
  Spectral discretization: 32 g-points
  Effective radius look-up: 23 points in range 5.0- 60.0 um
  Wavenumber range: 0-3250 cm-1
Reading NetCDF file ../../data/aerosol_ifs_48R1.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 7: Desert dust, bin 1, 0.03-0.55 micron, Woodward 2001, Table 2
```

```
5 -> hydrophobic type 8: Desert dust, bin 2, 0.55-0.90 micron, Woodward 2001, Table 2
6 -> hydrophobic type 9: Desert dust, bin 3, 0.90-20.0 micron, Woodward 2001, Table 2
7 -> hydrophilic type 4: Hydrophilic organic matter, OPAC
8 -> hydrophobic type 10: Hydrophobic organic matter, OPAC (hydrophilic at RH=20%)
9 -> hydrophobic type 11: Black carbon, OPAC
10 -> hydrophobic type 11: Black carbon, OPAC
11 -> hydrophilic type 5: Ammonium sulfate (for sulfate), GACP Lacis et al https://gacp.giss.nasa.gov/data\_sets
12 -> hydrophobic type 14: Stratospheric sulfate (hydrophilic ammonium sulfate at RH 20%-30%)
Reading NetCDF file ecrad_meridian.nc
Warning: variable co_vmr not found
Warning: variable no2_vmr not found
Performing radiative transfer calculations
Time elapsed in radiative transfer: 0.42291E-01 seconds
Writing NetCDF file ecrad_meridian_eckkd_tc_out.nc
-----
```

Chapter 3

Incorporating *ecRad* into another program

ecRad can be called within a larger program, and indeed it has been incorporated into several atmospheric models (the IFS, Meso-NH and ICON). Pending a full description here of how to do this, see the `ifs/radiation_setup.F90` in the *ecRad* package to see how it is configured in the IFS, and `ifs/radiation_scheme.F90` for how it is run.

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 3.1.

Table 3.1: 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 <code>config_type</code>	Available named constants, default
<code>i_overlap_scheme</code>	<code>IOverlapMaximumRandom</code> , <code>IOverlapExponentialRandom</code> , <code>IOverlapExponential</code>
<code>i_solver_sw</code> , <code>i_solver_lw</code>	<code>ISolverCloudless</code> , <code>ISolverHomogeneous</code> , <code>ISolverMcICA</code> , <code>ISolverSpartacus</code> , <code>ISolverTripleclouds</code>
<code>i_3d_sw_entrainment</code>	<code>IEntrapmentZero</code> , <code>IEntrapmentEdgeOnly</code> , <code>IEntrapmentExplicit</code> , <code>IEntrapmentExplicitNonFractal</code> , <code>IEntrapmentMaximum</code>
<code>i_gas_model</code>	<code>IGasModelMonochromatic</code> , <code>IGasModelIFSRRTMG</code> , <code>IGasModelECKD</code>
<code>i_liq_model</code>	<code>ILiquidModelMonochromatic</code> , <code>ILiquidModelSOCRATES</code> , <code>ILiquidModelSlingo</code>
<code>i_ice_model</code>	<code>IIceModelMonochromatic</code> , <code>IIceModelFu</code> , <code>IIceModelBaran2016</code> , <code>IIceModelYi</code>
<code>i_cloud_pdf_shape</code>	<code>IPdfShapeGamma</code> , <code>IPdfShapeLognormal</code>

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 microphysics–radiation parameterizations on the temperature and specific humidity biases in the tropical tropopause layer in a climate model. *J. Climate*, **29**, 5299–5316.
- Edwards, J. M., and Slingo, A., 1996: Studies with a flexible new radiation code: 1. Choosing a configuration for a large-scale model. *Q. J. R. Meteorol. Soc.*, **122**, 689–719.
- Fielding, M. D., S. A. K. Schäfer, R. J. Hogan and R. M. Forbes, 2020: Encapsulating cloud geometry for 3D radiative transfer and cloud turbulent mixing parameterizations. *To be submitted to Q. J. R. Meteorol. Soc.*
- 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. J. Illingworth, 2000: Deriving cloud overlap statistics from radar. *Q. J. R. Meteorol. Soc.*, **126**, 2903–2909.
- Hogan, R. J., and A. J. Illingworth, 2003: Parameterizing ice cloud inhomogeneity and the overlap of inhomogeneities using cloud radar data. *J. Atmos. Sci.*, **60**, 756–767.
- 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, 2016: ECRAD: a new radiation scheme for the IFS. *ECMWF Technical Memorandum 787*, available at <http://www.ecmwf.int/en/elibrary/16901-ecrad-new-radiation-scheme-ifs>
- 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.
- Hogan, R. J., M. D. Fielding, H. W. Barker, N. Villefranche and S. A. K. Schäfer, 2019: Entrapment: An important mechanism to explain the shortwave 3D radiative effect of clouds. *J. Atmos. Sci.*, **76**, 2123–2141.
- Hogan, R. J., and M. Matricardi, 2020: Evaluating and improving the treatment of gases in radiation schemes: the Correlated K-Distribution Model Intercomparison Project (CKDMIP). *Geosci. Model Dev.*, **13**, 6501–6521.
- Hogan, R. J., and M. Matricardi, 2022: A tool for generating fast *k*-distribution gas-optics models for weather and climate applications. Submitted to *J. Adv. Modeling Earth Sys.*
- 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.