NWP SAF

IRSPP User Manual

Document version 1.4

28th October 2024

IRSPP User Manual

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 7 December 2016, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, DWD and Météo France.

Change record			
Version	Date	Author / changed by	Remarks
0.1	09/11/2021	Nigel Atkinson	For internal review
0.2	01/12/2021	Nigel Atkinson	For external beta testing
0.3	18/01/2022	Nigel Atkinson	After beta testing
1.0	31/01/2022	Nigel Atkinson	Approved
1.1	16/02/2022	Nigel Atkinson	Add clarifications as recommended in review. Release with IRSPP v1.0
1.2	23/08/2022	Nigel Atkinson	Update for IRSPP v1.1
1.3	01/02/2023	Nigel Atkinson	Update for IRSPP v1.2
1.4	28/10/2024	Nigel Atkinson	Updates for IRSPP v1.3: Update BUFR sequence and flags. Support PC-RTTOV instead of HT-FRTC.

Table of Contents

1. INTRODUCTION

This document is the user manual for the Infrared Sounder Pre-processor, IRSPP. It is a software deliverable of the NWP SAF. Information on the package can be found at https://nwp-saf.eumetsat.int/site/software/irspp/. In this manual, sections 2 and 3 describe how to install the package, section 4 provides an overview of the functionality, section 5 lists the routines and control files, section 6 describes the input and output data, section 7 described in detail how to run each main program, section 8 presents the theory, section 9 describes the test cases and section 10 gives details of the BUFR sequence that may be used as output. Background information on the MTG mission and on the IRS instrument may be found on the EUMETSAT web site at https://www.eumetsat.int/meteosat-third-generation.

1.1 Reference documents

[RD-1]	IRSPP Product Specification, NWPSAF-MO-DS-037, available on the NWP SAF web site.
[RD-2]	IRSPP Top Level Design, NWPSAF-MO-DS-043, available on the NWP SAF web site.
[RD-3]	MTG IRS Level 0 & 1 Format Specification [IRSL1FS], EUM/MTG/SPE/10/0449
[RD-4]	MTG Test Data, https://user.eumetsat.int/resources/user-guides/mtg-test-data

Please note that for completeness some sections of [RD-2] are also included in this User Manual.

1.2 IRSPP releases

IRSPP v1.0 was an initial release designed to be compatible with the pre-launch simulated IRS data released by EUMETSAT in December 2019. It allowed users to start to prepare their pre-processing systems well in advance of the data becoming operational.

As more test datasets became available (more dwells, updated spectral grid, etc.), several IRSPP v1 update releases have been released, together with new test cases. These update releases also attempt to take account of feedback from users (e.g. bug reports). IRSPP v1.1 is the first update release, compatible with test data from July 2022 (TD-406). IRSPP v1.2 is compatible with test data from November 2022 (TD-417). IRSPP v1.3 is compatible with Principal Component test data from July 2024 (TD-521), while still supporting the full-spectrum test data in TD-417.

A second main release, IRSPP v2, is foreseen for release after launch. It will have been tested using in-orbit data. This version may also incorporate improvements to the functionality that have been requested by users.

2. INSTALLATION FROM SOURCE

2.1 Delivery

IRSPP is normally delivered as a gzipped tar file, e.g. `IRSPP_v1.3_source.tgz`. Unpack it using:

```
tar -xzf IRSPP_v1.3_source.tgz
```

The IRSPP top directory is referred to in this document as $IRSPP_HOME. We recommend that you define it as an environment variable, i.e.

```
cd IRSPP
export IRSPP_HOME=$PWD
```

Test cases are delivered separately, also as a gzipped tar file.

2.2 Prerequisites

IRSPP is intended to be built on a Linux platform. A Fortran90 compiler (e.g. gfortran) is required. These instructions assume the use of a ksh or bash environment. IRSPP makes use of the following libraries, which need to be installed on your system:

- hdf5 (with Fortran enabled)
- netcdf-c
- netcdf-f
- ecCodes

and the following optional library (if you want to generate your own eigenvectors):

- LAPACK (Linear Algebra PACKage). Current versions come with BLAS (Basic Linear Algebra Subprograms) included.

You will also need cmake (for building ecCodes) and zlib. These may be included in your Linux distribution. See Table 1 for access details.

Table 1: External libraries used in IRSPP

Library	Typical version	URL
ecCodes	2.27.1	https://www.ecmwf.int/en/computing/software
cmake	3.21.2 (needs 3.6 or higher)	https://cmake.org/download/
hdf5	1.10.7	https://support.hdfgroup.org/ftp/HDF5/releases/
netcdf-c	4.8.1	https://www.unidata.ucar.edu/downloads/netcdf/
netcdf-f	4.5.3	https://www.unidata.ucar.edu/downloads/netcdf/
lapack	3.10.0	http://www.netlib.org/lapack/
zlib	1.2.11	https://zlib.net/ (zlib is normally part of the Linux distribution)

If you have administrator privilege on your workstation, you should follow your normal procedures to install the libraries (e.g. using *sudo yum install*).

If you do not have administrator privilege, then a script *install_dependencies.sh* is provided in $IRSPP_HOME which can download and build the libraries, one at a time, and install them in a local directory. By default, they are built and installed in a directory $IRSPP_HOME/dependencies. If you want to install the libraries somewhere else, you can set an environment variable before running *./install_dependencies.sh*:

```
export IRSPP_DEPENDENCIES=directory path
```

You may also set the Fortran compiler at this stage (default is gfortran), e.g.

```
export FC=compiler   #gfortran or ifort
```

You should expect it to take about an hour to install all the dependency libraries.

If you are not sure what libraries are already installed on your workstation, proceed to the next step and go back to run *install_dependencies.sh* if any errors are reported.

Important note

IRSPP makes use of both the netCDF library and the hdf5 library. NetCDF also has an internal dependency on hdf5. If there are multiple versions of these libraries centrally installed on your workstation (and you did not use the *install_dependencies.sh* script) then you should make sure that the hdf5 library that you link to in IRSPP is the same version as was used internally in the netCDF. Otherwise you could get a run-time error.

For example, at the time of writing the ECMWF compute system has available both hdf5 v1.10.6 and v1.14.3, but only the 1.14.3 is compatible with netCDF v4.9.2.

If in doubt, you can create a dummy program (it doesn't need any content), compile it with the netCDF library linked in (via the flag -lnetcdff), then use **ldd** to examine the dependencies of the executable.

2.3 Building IRSPP

Next you should run the supplied *configure_irspp.sh* script. If your dependency libraries are centrally installed, or if you have installed them using *install_dependencies.sh* (section 2.2), then no arguments are needed:

```
cd $IRSPP_HOME
./configure_irspp.sh
```

The script will then attempt to locate the dependency libraries and their include files. If it doesn't find what is needed, then it will inform you.

Alternatively, you can explicitly specify the fortran compiler and provide the locations of the dependency libraries (LIB) and their include files (INC):

```
./configure_irspp.sh [options]
  options: --fortran-compiler=FORTRAN_COMPILER (default gfortran)
           --hdf5-lib=HDF5_LIB
           --hdf5-inc=HDF5_INC (optional)
           --netcdf-lib=NETCDF_LIB
           --netcdf-inc=NETCDF_INC (optional)
           --eccodes-lib=ECCODES_LIB
           --eccodes-inc=ECCODES_INC (optional)
           --lapack-lib=LAPACK_LIB (optional)
           --help (display this message)
```

It is possible to use environment variables instead of options: Highest priority: FC, HDF5_INSTALL_DIR, NETCDF_INSTALL_DIR, ECCODES_INSTALL_DIR. Lower priority: ECCODES_LIB, ECCODES_INC, HDF5_LIB, HDF5_INC, NETCDF_LIB, NETCDF_INC, LAPACK_LIB.

If LAPACK is not installed on your system then IRSPP will automatically build without it, but you will not be able to generate your own eigenvectors. Most users will not require this functionality.

The configure script generates a file Makefile.ARCH, which is used by all the Makefiles. It also creates a file *$IRSPP_HOME/bin/irspp_env.sh* which can be sourced at run-time (see section 7). This sets up any additions that may be needed to your PATH and (if necessary) LD_LIBRARY_PATH. It may also set up ECCODES_DEFINITION_PATH to access local versions of BUFR tables.

Having run *configure_irspp.sh*, you can then run the following command:

```
cd src; make
```

This will compile the software and create executables in the *$IRSPP_HOME/bin* directory. You can copy them elsewhere if you wish. There is no “make install” command.

3. INSTALLATION FROM BINARIES

In the past, all NWP SAF deliverables have been delivered as source files which the user has to compile. In response to requests from users, IRSPP will also be delivered as binaries, with the necessary dynamic libraries, built on a 64-bit Rad Hat Enterprise Linux 7 system (v7.9). This should be compatible with CentOS7. It has also been tested successfully on a RockyLinux 8.5 system. You can run the *hostnamectl* command to display information about your Linux system.

Save the *IRSPP_exec.tgz* file to a suitable directory, unpack it and set up your environment as follows:

```
tar -xzf IRSPP_v1.3_exec.tgz
cd IRSPP_exec
export IRSPP_HOME=$PWD
. $IRSPP_HOME/bin/irspp_env.sh
```

The *irspp_env.sh* performs three functions:

1. Adds $IRSPP_HOME/bin to the front of your PATH (if it is not already there)
2. Adds $IRSPP_HOME/lib to the front of your LD_LIBRARY_PATH
3. Defines ECCODES_DEFINITION_PATH and ECCODES_SAMPLES_PATH to access the local BUFR tables and the default tables and samples.

This should allow the executables to be run. You will be able to control the functionality of IRSPP using namelists (see following sections) but of course you will not be able to make any changes to the code.

In order to ensure that the IRSPP_exec can be run with no dependencies, the ecCodes “share” directory has been included in the tar file. If you have your own installation of ecCodes and prefer to use those tables then simply adjust the environment variables in *irspp_env.sh* so that they point to your own ecCodes shared files.

Please note that the binaries do *not* run on Suse Linux.

4. FUNCTIONALITY OVERVIEW

4.1 Functionality within IRSPP

The IRSPP package is designed for processing data from the MTG-IRS sensor, specifically the Principal Component (PC) and Spectral Sounding Sample (SSS) level 1b products. The PC product will be available from EUMETSAT in near real time (e.g. via EUMETCast satellite and terrestrial services[1]), whereas the SSS product is currently planned to be available only from the Data Centre.

The IRSPP package also incorporates some of the functionality that was present in an earlier NWP SAF deliverable, the “IASI PCA-based compression package”.

Typical workflows for IRSPP are shown in Figure 1 to Figure 5. For operational use, it is expected that only the first of these would be used. However, the ability for the users to generate, and use, their own eigenvectors is also provided, for consistency with the IASI PCA package.

Please note that IRSPP does not attempt to make any improvements to the IRS data quality: it is assumed that all necessary corrections will have been made in the level 1 processing that is run at EUMETSAT, i.e. all pixels use the same spectral grid and have been harmonised so that their spectral response functions are the same (Coppens et al., 2019). If, after launch, specific corrections are found to be necessary for NWP assimilation, these will be considered for a future software release.

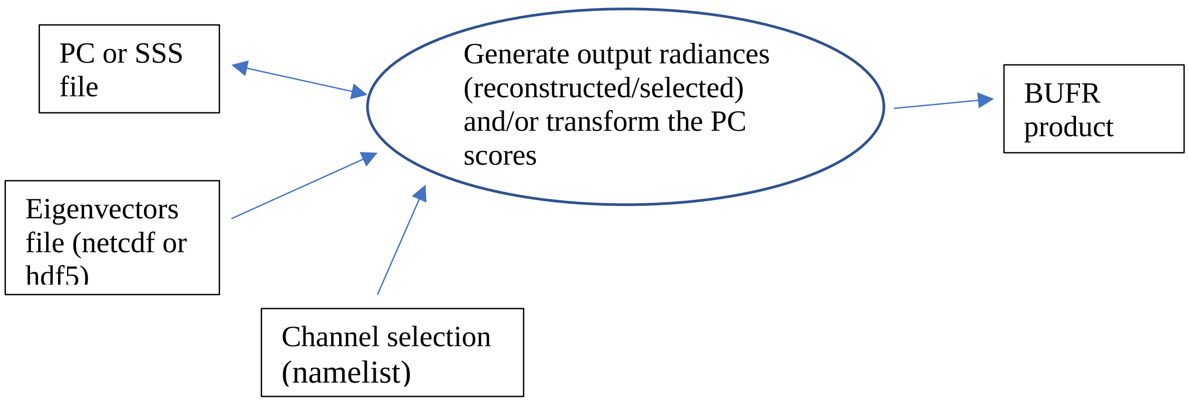

Figure 1: Workflow for the core task of processing IRS data for use in NWP

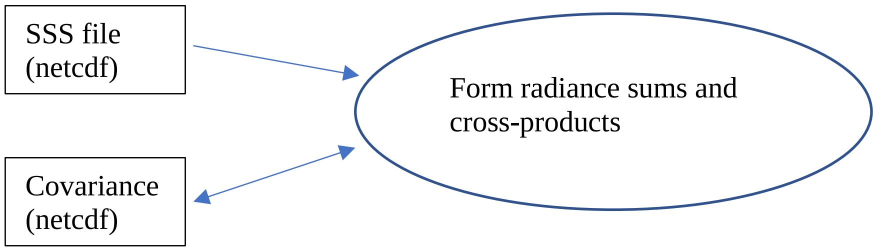

Figure 2: Workflow for generating covariance matrix from many SSS files

[1] https://user.eumetsat.int/data-access

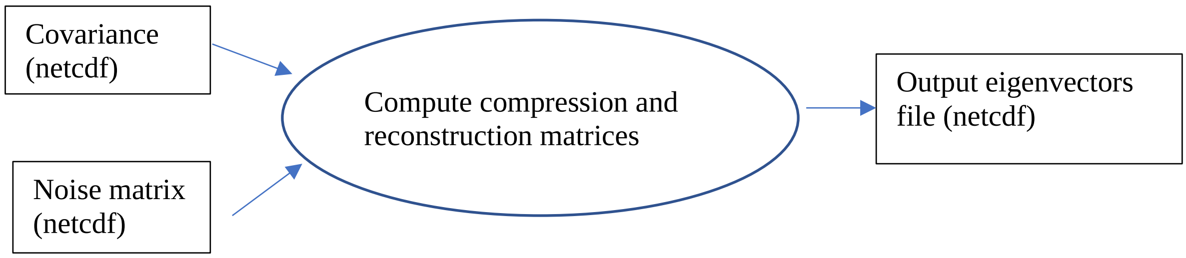

Figure 3: Workflow for generating user-defined eigenvectors

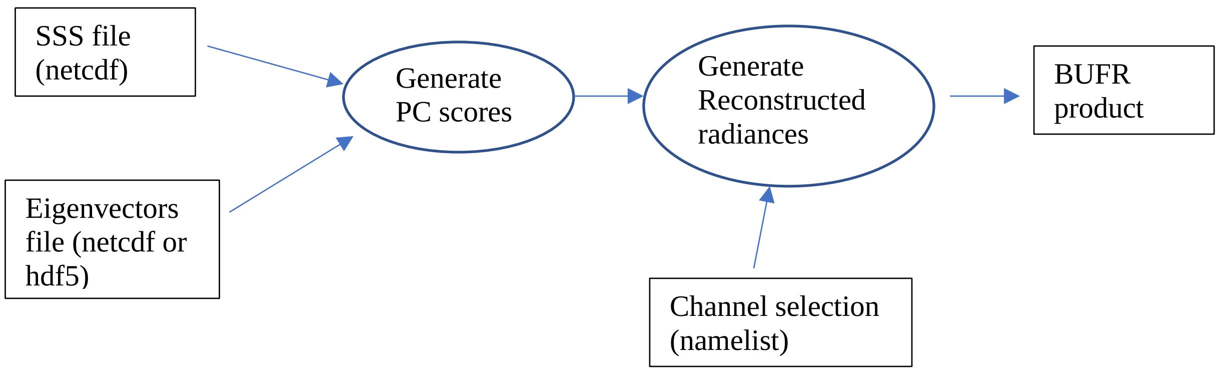

Figure 4: workflow for performing spectral filtering using the user's own eigenvectors

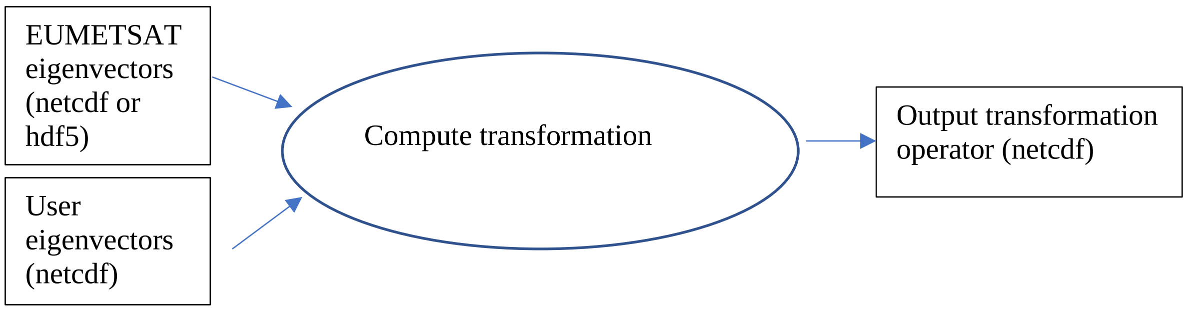

Figure 5: workflow for generating a transformation operator, for converting from one basis function to another

The following points should be noted:

- An input IRS file normally contains 1 dwell, i.e. 160 x 160 spectra, acquired in 10 seconds of observation time.
- There is an option to thin the data by selecting 1 spot in *n* spots or 1 line in *m* lines.
- Processing time is approximately proportional to the number of spots processed. If reconstructed radiances are being generated, processing time is also proportional to the number of channels in the user's channel selection.
- One instance of IRSPP runs on one thread.
- The expectation is that a dwell can be processed in less than 10 seconds (i.e. faster than real-time). But if increased throughput is required, the user can run several instances simultaneously, processing different data files.

- The software also supports processing of simulated IRS spectra from the NWP SAF's Radiance Simulator package (see discussion in section 8.7).

4.2 Relationships with RTTOV and assimilation systems

The NWP SAF deliverable RTTOV (Saunders et al., 2018; version 14 planned for release in late 2024) is a fast model that can be used to simulate IRS radiances in two ways:

1. Classical RTTOV: accurately simulates strongly-apodised radiances (e.g. Hamming-apodised) but does not work well with lightly-apodised radiances, due to the negative sidelobes of the spectral response function
2. PC-RTTOV (Matricardi, 2010) simulates Principal Component scores, from which reconstructed radiances can be generated. RTTOV v14 supports both clear-sky and cloudy/scattering PC simulations.

Previous versions of RTTOV supported the Havemann-Taylor Fast Radiative Transfer Code (HT-FRTC, Havemann et al., 2018), which is based on principal components. This model was capable of simulating lightly-apodised IRS radiances, but it is not supported in RTTOV v14.

The PC-RTTOV eigenvectors are Hamming-apodised, i.e. reconstructed radiances would also be strongly apodised. In principle, the PC regression (section 8) could be trained with lightly-apodised line-by-line spectra, but this has not been demonstrated. Currently, RTTOV only supports strongly apodised IRS spectra.

Assimilation systems typically generate simulated radiances based on model profiles (this step is outside the scope of IRSPP) and compare with the observed radiances. The function of IRSPP is to generate the observed radiances in a convenient form, either as reconstructed radiance for specified channels or as PC scores for an appropriate set of basis functions (not necessarily the same basis functions that are used for data transmission).

IRSPP would normally be used in near-real-time as a pre-processor, converting the incoming netCDF files (PC scores) to radiances or PC scores in BUFR format. The BUFR files would then be stored in a meteorological database. Some organisations may prefer to embed elements of IRSPP inside their assimilation code, but the details of that would be very dependent on the organisation's particular requirements, so are not discussed in this document.

IRSPP can also be used *off-line* to generate conversion matrices for PC basis function transformation, i.e. to generate a set of PCs that are optimised for model radiances. One way of doing this is to use the NWP SAF Radiance Simulator package to generate simulated radiances from model fields, and this has the advantage that there is a well-defined netCDF format for radiances that are presented to IRSPP. Alternatively, a transformation to the PC-RTTOV basis function set could be computed, see section 8.

5. SOFTWARE DESCRIPTION

5.1 Fortran code

The software comprises a collection of library routines, together with main programs that perform different tasks, according to the workflows of the previous section. The main routines are listed in Table 2 and the library routines are listed in Table 3.

Table 2: IRSPP main routines, in the bin directory

Main routine	Purpose
irs_main.f90	Processing IRS data for NWP (Figure 1)
irs_covariance.f90	Generating a covariance file (Figure 2)
irs_generate_pcs.f90	Computing eigenvectors from covariance (Figure 3)
irs_sss_filtering.f90	SSS filtering using user eigenvectors (Figure 4)
irs_transform_pcs.f90	Generate matrices that can be used to transform PC scores from one basis function to another (Figure 5)
format_ev.f90	Utility to convert eigenvectors file from the format of TD-521 to that of TD-417, which includes reconstruction operator
dummy.f90	Dummy routine, only compiled if LAPACK is not installed

Table 3: IRSPP library routines, in libirs directory

Routine	Purpose
Core routines	
irs_data_mod.f90	Definition of all the data held in common memory
irs_read_channelselection.f90	Read a namelist file containing a user-supplied channel selection
irs_read_irsdata.f90	Read netcdf file: PC product or full-spectra (SSS)
irs_read_irsdata_hdf5.f90	Read elements of the PC file that can’t be read using the netcdf API. Applies to certain enumerated data types.
irs_read_irsattr_hdf5.f90	Read attributes that are variable length strings.
irs_read_spectra.f90	Read only the spectra from an SSS file, ignoring all other information. Also accepts spectra from a RadSim file.
irs_read_static_eigenvectors.f90	Read static eigenvectors, etc., from an ancillary file in netCDF format
irs_read_eigenvectors_hdf5.f90	Read hdf5 version of the static eigenvectors
irs_read_noise.f90	Read only the noise from an ancillary file, ignoring all other information
irs_read_noise_hdf5.f90	Read noise from hdf5 version of the static eigenvectors
irs_apodise_eigenvectors.f90	Convert the static eigenvectors to a strong apodisation, applying a Hamming function on top of the original “light” apodisation.
irs_compute_rr.f90	Compute reconstructed radiances from PC scores, for the user-defined channels, using the static ancillary data and optionally including any dynamic eigenvectors
irs_append_netcdf.f90	Append output radiances to the original netcdf files. Creates a new group “derived”.
irs_write_bufr.f90	Generate a BUFR product containing the output radiances, and other information likely to be required by NWP applications.
irs_warmestfov.f90	As part of the spatial thinning, find the spectrum that has warmest radiance for a defined channel.

Covariance generation	
irs_covariance_readwrite.f90	Read from and write to a user-generated covariance file
irs_update_covariance.f90	Update a covariance using data from an SSS file
Eigenvector generation	
irs_write_eigenvectors.f90	Generate an eigenvector file in netCDF format containing compression matrix and reconstruction matrix
irs_write_eigenvectors_hdf5.f90	Generate an hdf5 version of the eigenvector file
compute_eigenvectors.f90	Generic routine to compute eigenvectors, and hence compression matrix and reconstruction matrix, from a covariance. Interfaces with LAPACK.
Compression	
irs_compute_pcscores.f90	Apply compression matrix to spectra that have been read from an SSS file, to generate PC scores
irs_compute_reconstruction_score.f90	Compute reconstruction score for the user-defined set of channels
Transformation	
irs_write_transformation.f90	Write data to a netcdf transformation file
irs_read_transformation.f90	Read data from a netcdf transformation file
irs_transform_pcs.f90	Convert PC scores from one basis function to another
irs_read_htfrtc_coef.f90	Read an HTFRTC coefficient file (no longer used but code retained for possible future use)
irs_read_pcrttov_coef.f90	Read a PCRTTOV coefficient file
Miscellaneous	
check.f90	Error handler for the netcdf API
matrix_inversion.f90	Matrix inversion using LAPACK
meansd_2d.f90	Mean and standard deviation of a 2-D array (used for the imager-mode radiances)

The Makefiles have a simple structure. Therefore, if you wish to write your own main program, to perform something specific, it should be straightforward to edit the Makefile in order to add your program to the build system.

5.2 Namelists

The requirements of the user are specified in two ways: by command arguments (e.g. “-i” or “-o” to give the names of the input/output files) and by namelists (for requirements that are the same from run to run – for example, the file of fixed eigenvectors).

The general structure of any namelist is as follows:

- First line: “&*name*” where *name* must match the Namelist definition in the Fortran. For example, the namelist that specifies BUFR parameters is called “&irs_bufr”
- Following lines: variables and their values. The variable names must match the Namelist definition in the Fortran. For example, “originating_centre = 74”
- Final line: “/”
- Comments can be used freely, with “!” at the start of the comment. New lines can also be used freely.

In most of the main programs, there is a command argument “-n” to specify the file name of a top-level namelist. The top-level namelist can in turn be used to specify the file names of lower level namelists.

An important namelist is the channel selection namelist. Its specification is shown in Table 4. Channels numbers start from 1 at the start of each band.

Table 4: Namelist: channels_namelist

Variable	Default	Comment
required_channels_lw	0	A list of channels, or -1 for all, or 0 for none
required_channels_mw	0	A list of channels, or -1 for all, or 0 for none

For example:

```
&channels_namelist

!list of channels in each band, or -1 for all or 0 for none
!New lines can be used freely

required_channels_lw =
1  11  21  31  41  51  61  71  81  91
101 111 121 131 141 151 161 171 181 191 ...

/
```

Note that each band is defined separately (starting from 1) in the channel selection namelist, but output BUFR files have a single set of numbers covering both bands (for consistency with other hyperspectral instruments like IASI and CrIS).

Another generic namelist is the BUFR specifications namelist, defined in Table 5.

Table 5: Namelist: irspp_bufr

Variable	Default	Comment
originating_centre	255	Should be set to 74 if local descriptors are used (which will be the case until the definitions are formally adopted by WMO)
sub_centre	0	
master_table	32	
local_table	0	Should be set to 2 to use local descriptors
local_subtype	0	
max_subsets	160	Can be used to control the length of BUFR messages
npc_out_lw	0	Number of PC scores to include, for LW
npc_out_mw	0	Number of PC scores to include, for MW

6. DATA

6.1 IRS data characteristics

The IRS spectral grid definition has changed during the evolution of the MTG programme, and users may encounter test datasets with different characteristics, see Table 6.

Table 6: Spectral grid definitions for IRS

Band	Version	Wavenumber range (cm^{-1})	Step (cm^{-1})	Number of samples
LW	Nominal	700 – 1210	0.625	817
	2022	679.703 – 1210.439	0.6031087	881
	“Final” (TBC)	679.49573 – 1210.01111	0.60561117	877
MW	Nominal	1600 – 2175	0.625	921
	2022	1599.769 – 2250.543	0.6036863	1079
	“Final” (TBC)	1599.90822 – 2250.40041	0.60510901	1076

The first set of test data released by EUMETSAT in 2019 conformed to the “nominal” spectral grid, but the products to be distributed post launch will use the “final” grid (based on optical path travel distance), in which the sampling step is slightly different for the two bands.

A second set of test data was released by EUMETSAT in July 2022, using what we have referred to above as the “2022” grid. This was followed by a new release in July 2024[2] on the same grid.

Data will normally be distributed in “dwells” of 160x160 spectra, where a complete earth disk comprises 280 dwells. For more information on the instrument, see https://user.eumetsat.int/resources/user-guides/mtg-irs-level-1-data-guide .

6.2 IRSPP input and output data

The main input data to IRS will normally be netCDF files in the format specified in [RD-3]. The software will accept:

- IRS PC files
- IRS SSS (full-spectrum) files

Each file contains data for 1 dwell (160x160 spectra, gathered in 10 seconds). EUMETSAT plan to make the PC files available in near-real-time, via EUMETCast, but the SSS files will only be available from the Data Archive.

[2] https://vuser.eumetsat.int/resources/user-guides/mtg-test-data

In addition, there will be ancillary data, notably the file of fixed eigenvectors. These will be either in netCDF or hdf5. (IRSPP v1.3 can read both versions).

Some users may want to ingest simulated IRS data that have been generated using a radiative transfer model. IRSPP is able to ingest netCDF radiance files generated with the NWP SAF's *Radiance Simulator* package. If a different format is needed (e.g. ASCII) then the user will need to modify the ingest routine.

The main outputs will be:

- BUFR files containing PC scores and/or reconstructed radiances
- There will be an option to write reconstructed radiances back to the input netCDF files

A BUFR sequence has been defined, see section 10. Note that two local descriptors are used to hold IRS-specific quality flags.

6.3 IRS channel selections

The channels to be written out by IRSPP, as reconstructed radiances, are specified via namelist (section 5.2). In earlier versions of IRSPP, a channel selection was provided based on the IASI "500-channel" set (based on Collard 1997). However, this is not ideal for IRS as many of the IASI water vapour channels are not available to IRS.

For the IRSPP v1.3 release, a channel selection based on the information-content study of Coopmann (2022) is used. This has 245 LWIR channels and 55 MWIR channels. See Figure 6.

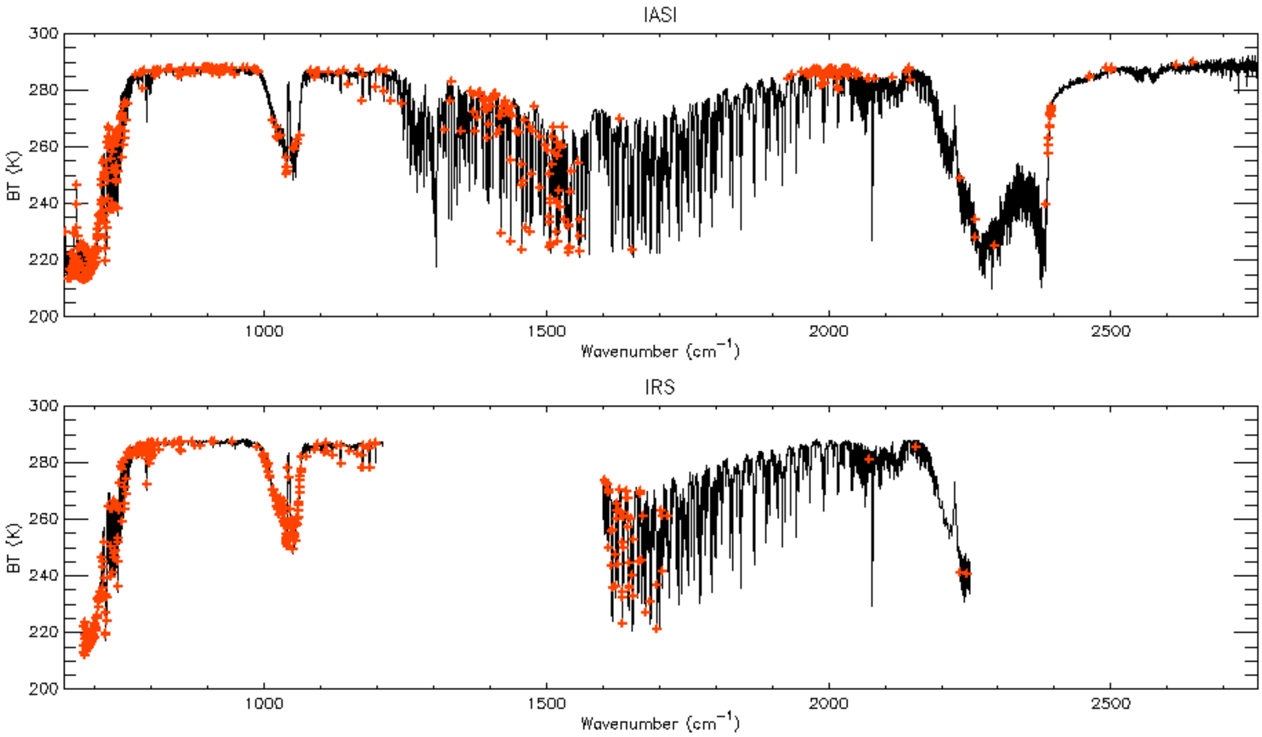

Figure 6: Top: typical IASI spectrum with 500 selected channels marked by red crosses. Bottom: the equivalent IRS spectrum, with 300 channels from Coopmann (2022) marked by red crosses.

Although the Coopmann (2022) selection is tailored to IRS and is reasonably realistic, it cannot be considered a “final” channel selection because:

- It uses the 2022 spectral grid (defined in Table 6), which is not the final one
- The study was based on raw radiances, not reconstructed radiances

7. RUNNING THE MAIN PROGRAMS

As mentioned previously, before starting any IRSPP session, you should source your environment using

```
. $IRSPP_HOME/bin/irspp_env.sh
```

7.1 irs_main.exe and irs_main_parallel.sh

Purpose: pre-processing for NWP

Usage: irspp_main.exe -n namelistfile [-i infile_irs | -io inoutfile_irs] [-o outfile_bufr]

Or: irs_main_parallel.sh [-n namelistfile] [-o outdir] nmax infiles

The *namelistfile* defines the variables listed in Table 7.

Table 7: Namelist: irspp_main_namelist

Variable	Default	Comment
infile_static		Static ancillary file, either netCDF or hdf5 (EUMETSAT have used two different formats for this file)
use_local_pcs	.false.	Use dwell-dependent PCs? See section 8.2
apply_hamming	.false.	Apply additional Hamming apodisation? See section 8.5
transform_pcs	.false.	Transform the basis function? If yes, then *infile_static* must be a transformation file – see sections 7.5 and 8.6
channel_selection_namelist		Defines the required channels
bufr_namelist		BUFR specifications
firstrow	1	Can be used for spatial thinning
firstcolumn	1	“
rowstep	1	“
columnstep	1	“
delta_row	0	Row increment defining a test box centred on the nominal FOVs
delta_column	0	Column increment defining a test box centred on the nominal FOVs
test_channel	0	Long-wave channel number used to define warmest radiance (e.g. channel 408 = approximately 10.8μm wavelength)
q	0.5	PC score quantisation (only applicable when using the *transform_pcs* option)

If the “-i” option is used then *infile_irs* is either an SSS (full-spectrum) file or a PC file, in netCDF format. These files normally hold 1 dwell, i.e. 160x160 spectra.

Alternatively, if the “-io” option is used, then the output radiance spectra, for the specified channels, are appended to the input file, in new groups “/data/LWIR/derived” and “/data/MWIR/derived”.

If the “-o” option is used, then an output BUFR file *outfile_bufr* is created. See Section 10 for details.

When processing IRS data in near-real-time, it is important that the average processing time per dwell is less than the measurement time of 10 seconds. Tests show that this is normally the case. However, IRSPP provides two methods of speeding up the processing should that be required:

- Set *rowstep* and/or *columnstep* to be greater than 1 in the main namelist. This spatially thins the output data.
- Use the wrapper script *irs_main_parallel.sh*. The user supplies the script with a list of files to be processed, and the script runs *irs_main.exe* for several files simultaneously, on different processors, up to a specified maximum number (*nmax*). This is faster than running the files sequentially. The script ends when all files have been processed.

The *delta_row*, *delta_column* and *test_channel* options are provided to allow the user to thin the data in a more intelligent way that simply selecting a fixed output grid, and also to avoid selection of two neighbouring FOVs, which may be undesirable since the instrument point spread function is significantly larger than the FOV spacing of 4km at the sub-satellite point. If row/column thinning is applied, the result is a grid of boxes, each with columnstep x rowstep samples. From each gridbox, only one FOV is selected: the warmest FOV from the central test box.

For example:

- firstcolumn=3, columnstep=5, delta_column=1, firstrow=3, rowstep=5, delta_row=1. Here the gridbox is 20x20km at SSP (5*5 samples); the warmest FOV is chosen from the central square 12x12km (3*3 samples).
- firstcolumn=4, columnstep=8, delta_column=2, firstrow=4, rowstep=8, delta_row=2. Here the gridbox is 32x32km at SSP (8*8 samples); the warmest FOV is chosen from the central square 20x20km (5*5 samples).
- firstcolumn = 4, columnstep = 8, delta_column = 2, firstrow = 2, rowstep = 3, delta_row = 1. Here the gridbox is 32x12km at SSP (8*3 samples); the warmest FOV is chosen from the central rectangle 20x12km (5*3 samples). This is a grid in which NS and EW spacings are more closely matched at high latitudes.

We anticipate that users could implement different thinning schemes depending on geographical region, i.e. based on dwell number.

7.2 irs_covariance.exe

Purpose: Create or update a covariance file, starting from full spectra

Usage: irspp_covariance.exe -n namelistfile -i infile_sss -o outfile_cov

The *namelistfile* defines the variables listed in Table 8.

Table 8: Namelist: irspp_cov_namelist

Variable	Default	Comment

firstrow	1	Can be used for spatial thinning
firstcolumn	1	“
rowstep	1	“
columnstep	1	“

Spectra taken from the input file are appended to the covariances in the output file. The input file would typically be a NetCDF SSS file containing the following datasets:

/data/lwir/measured/effective_radiance – dimension (wavenumber_lw, columns, rows)
/data/lwir/wavenumber (netCDF dimension)
/data/mwir/measured/effective_radiance – dimension (wavenumber_mw, columns, rows)
/data/mwir/wavenumber (netCDF dimension)
/data/dwell_row (netCDF dimension)
/data/dwell_column (netCDF dimension)

Note that the effective_radiance datasets are integers with *scale_factor* and *add_offset* attributes.

Alternatively, the code accepts a NetCDF radiance file generated by the NWP SAF *Radiance Simulator* (RadSim), containing:

/wavenumber – dimension (channels)
/radiance – dimension (obs, channels)
/obs (NetCDF dimension)
/channels (NetCDF dimension)

In this case, *firstcolumn* and *columnstep* in the namelist file should either be omitted or set to 1.

A third option could be for the user to create a customised version of *irs_read_spectra.f90*, if the input radiances are in a different format (e.g. ASCII).

7.3 irs_generate_pcs.exe

Purpose: Perform PC analysis on a covariance file, generating eigenvectors, etc.

Usage: irs_generate_pcs.exe -n namelistfile -i infile_cov -o outfile_pcs

The *namelistfile* defines the variables listed in Table 9.

Table 9: Namelist: irspp_generate_pcs_namelist

Variable	Default	Comment
infile_static		Static ancillary file
global_pcs_lw	150	Number of LW PCs to generate
global_pcs_mw	150	Number of MW PCs to generate

The static ancillary file is used only to obtain the noise profile for IRS. Normally a EUMETSAT-supplied file would be used, but a user-defined file could be substituted if a different noise profile was wanted (e.g. including model noise). In that case, the necessary datasets are:

/lwir/noise_normalisation – dimension (wavenumber_lw, wavenumber_lw)
/lwir/wavenumber (netCDF dimension)
/mwir/noise_normalisation – dimension (wavenumber_mw, wavenumber_mw)
/mwir/wavenumber (netCDF dimension)

The output file is similar to the EUMETSAT static ancillary files, but it contains some additional matrices (shown in blue below) to allow it to be easily used for compression as well as reconstruction:

/lwir/compression_operator
/lwir/reconstruction_operator
/lwir/mean_spectrum
/lwir/noise_normalisation
/lwir/inverse_noise

and similarly for /swir.

Note that IRSPP does not include the capability to generated local (dwell-dependent) PCs. Users who want to use local PCs should use the standard EUMETSAT PC product.

7.4 irs_sss_filtering.exe

Purpose: Convert full-spectrum IRS data to PC scores and, optionally, reconstructed radiances, using the PCs generated in section 7.3.

Usage: irs_sss_filtering.exe -n namelistfile -i infile_sss -o outfile_bufr

The *namelistfile* defines the variables listed in Table 10. In this case, the static ancillary file is the output from irs_generate_pcs.exe.

Table 10: Namelist: irspp_sss_filtering_namelist

Variable	Default	Comment
infile_static		Static ancillary file
channel_selection_namelist		Defines the required channels
bufr_namelist		BUFR specifications
firstrow	1	Can be used for spatial thinning
firstcolumn	1	"
rowstep	1	"
columnstep	1	"
q	0.5	PC score quantisation

7.5 irs_transform_pcs.exe

Purpose: Create a transformation from one PC basis function to another.

Usage: irs_transform_pcs.exe -n namelistfile -i eigenvectors_eumetsat -j eigenvectors_user -o outfile

The two eigenvectors files – one from EUMETSAT and the other either generated by the user or taken from PC-RTTOV – are used to create an output transformation file, according to the equations given in section 8.6.

The namelistfile defines the variables listed in Table 11.

Table 11: Namelist: irspp_transform_namelist

Variable	Default	Comment

apply_hamming	.true.	Controls whether or not a Hamming apodisation is to be applied to the EUMETSAT eigenvectors. Set this to true if transforming to PC-RTTOV.
max_pcs	400	Number of PCs to retain. Up to 400 for PC-RTTOV

8. PRINCIPAL COMPONENTS - THEORY

8.1 Converting between radiances and PC scores

Principal Components (PC) scores, **p**, are related to the measured radiance spectrum, **r** (column vector) and the reconstructed radiance spectrum, **r'**, as follows:

$$\mathbf{p} = \mathbf{E}^{T}\, \mathbf{N}^{-1}\, (\mathbf{r} - \mathbf{r_m}) \qquad (1)$$

$$\mathbf{r}' = \mathbf{r_m} + \mathbf{N}\,\mathbf{E}\,\mathbf{p} \qquad (2)$$

where $\mathbf{r_m}$ is a fixed mean radiance spectrum, **N** is the noise normalisation matrix (usually the matrix square root of the instrument noise covariance) and **E** is a truncated set of eigenvectors ($\mathbf{E}^{T}\,\mathbf{E} = \mathbf{I}$, where **I** is the identity matrix). Note that **E** is of rank {*nchan* rows, *npc* columns}.

To minimise the number of matrix multiplications being carried out in near-real-time, we define a compression operator $\mathbf{C} = \mathbf{E}^{T}\,\mathbf{N}^{-1}$ and a reconstruction operator $\mathbf{R} = \mathbf{N}\,\mathbf{E}$, so that:

$$\mathbf{p} = \mathbf{C}\,(\mathbf{r} - \mathbf{r_m}) \qquad (3)$$

$$\mathbf{r}' = \mathbf{r_m} + \mathbf{R}\,\mathbf{p} \qquad (4)$$

For self-apodised data, the noise matrix is expected to be diagonal (i.e. white noise in the interferogram), but the ancillary files supplied by EUMETSAT have a full-rank (nchan x nchan) noise matrix.

For computing reconstructed radiances from a set of (global) PC scores, the user only needs to apply equ. (4).

For more information on the principles of principal component analysis, see Antonelli et al., 2004, or Collard et al., 2010.

8.2 Global and local PCs

MTG-IRS PC files contain two types of PC scores: (i) global PCs that relate to a fixed set of eigenvectors held in an external ancillary file, and (ii) a small number of local PCs that are specific to the dwell. (A dwell comprises 160 x 160 spectra). The purpose of the local PCs is to ensure that any unusual signals (e.g. caused by localised factory emissions or volcanic eruptions) are accurately represented. See Hultberg et al., 2017a or 2017b.

IRSPP does not provide the facility to *compute* local PCs, as this is the function of the EUMETSAT ground segment. However, it does have the facility to use them when reconstructing radiances. In this case, equation (4) is modified to

$$\mathbf{r}' = \mathbf{r_m} + \mathbf{R_{global}}\,\mathbf{p_{global}} + \mathbf{R_{local}}\,\mathbf{p_{local}} \tag{4a}$$

8.3 Reconstruction scores

The PC reconstruction score, or PCR score, for a particular band, is defined as

$$\text{PCR score} = \frac{1}{nchan}\sqrt{\sum_{i=1}^{nchan}\left(\mathbf{N}^{-1}(\mathbf{r}' - \mathbf{r})\right)_i^2} \tag{5}$$

In the EUMETSAT PC product, the PCR score is computed over all channels in the band. If you are working with SSS data, the PCR score computed by IRSPP uses only the channels selected by the user (to save on computing resources).

If the PCR score is significantly greater than 1.0, that is an indication either that there are signals in the observed spectra that are not present in the eigenvectors, or that the instrument noise profile is unrealistic.

8.4 Creating the eigenvectors

The facility to create eigenvectors is an optional part of IRSPP and is included for consistency with the earlier NWP SAF “IASI PCA-based compression package”. The method implemented in IRSPP closely follows that implemented in the earlier package, except that data files are NetCDF rather than ASCII.

The process of creating eigenvectors, from a set of reference spectra, comprises (i) the formation of a noise-normalised covariance matrix, **Cov**, and (ii) computation of its eigenvectors and eigenvalues.

To create the covariance matrix, the method implemented in IRSPP is first to form the following sums, over all spectra in the training set:

$$\sum \mathbf{r} \qquad \text{and} \qquad \sum \mathbf{r}\,\mathbf{r}^{\mathbf{T}}$$

These quantities are held in a NetCDF data file. Further spectra can be easily added to the training set, if required. The number of spectra used should typically be of order 100000.

Then when the training set is complete (with *n* spectra) we compute the mean spectrum, $\mathbf{r_m}$, and form the noise-normalised covariance as follows:

$$\mathbf{r_m} = \frac{1}{n}\sum \mathbf{r} \tag{6}$$

$$\begin{aligned}\mathbf{Cov} &= \frac{1}{n}\sum\left(\mathbf{N}^{-1}(\mathbf{r} - \mathbf{r_m})\right)\left(\mathbf{N}^{-1}(\mathbf{r} - \mathbf{r_m})\right)^{\mathbf{T}} \\ &= \mathbf{N}^{-1}\left(\frac{1}{n}\sum \mathbf{r}\,\mathbf{r}^{\mathbf{T}} - \mathbf{r_m}\mathbf{r_m}^{\mathbf{T}}\right)\mathbf{N}^{-1}\end{aligned} \tag{7}$$

where *n* is the number of spectra. The eigenvectors and eigenvalues are related to each other, and to the covariance matrix, by:

$$\mathbf{Cov} = \mathbf{E}\,\mathbf{\Lambda}\,\mathbf{E}^{\mathrm{T}} \tag{8}$$

where **Λ** is a diagonal matrix containing the eigenvalues. IRSPP uses LAPACK routines to perform the eigenvalue decomposition. (Actually, interfaces to two LAPACK routines are provided, but they are functionally equivalent).

8.5 Apodisation

Apodisation is the process modifying the instrument's spectral response function (SRF) in order to control the sidelobes, typically in order to reduce "ringing". It can be performed as either a multiplication in the interferogram domain or a convolution in the spectral domain.
EUMETSAT products for IRS are based on "light apodisation". This produces a modest reduction in sidelobes while leaving the main lobe approximately unchanged. Some users prefer a heavier apodisation (e.g. Hamming) – heavy attenuation of the sidelobes at the expense of a broader main lobe. See Figure 7.

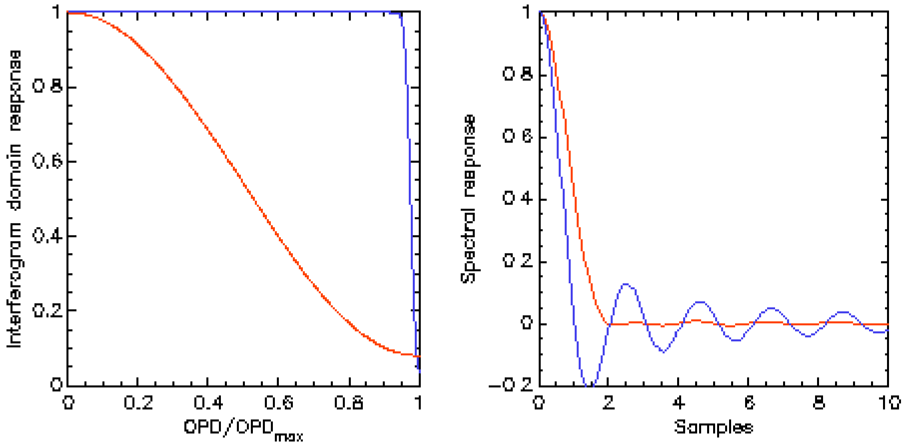

Figure 7: Comparison of light apodisation (blue) with Hamming apodisation (red), in the interferogram (left) and spectral (right) domains. OPD is the optical path difference.

In principle, apodisation is reversible, and does not change the information content of the spectra. However, different types of apodisation can be preferred for different applications. EUMETSAT's light apodisation is most easily computed in the interferogram domain: it is the convolution of a top-hat (with edge slightly below OPDmax) with a narrow gaussian. Whereas the Hamming is most easily computed in the spectral domain: simply convolve the spectrum with the 3-element array [0.23,0.54,0.23].

The first and last points require special treatment. If the input spectrum is x and the apodised spectrum is y then the software applies: $y(1) = \{0.54*x(1) + 0.23*x(2)\} / 0.77$, and $y(n) = \{0.23*y(n\text{-}1) + 0.54*y(n)\} / 0.77$.

It is clear that users who want a heavy apodisation, and are working with spectra, can easily generate one by applying the Hamming *on top of* the light apodisation supplied by EUMETSAT. This is computationally much easier than first removing the light apodisation (using FFTs) and then applying the Hamming. In practice, the radiance differences between the two approaches are very small (below noise levels – see Figure 8), but it is preferable for radiative transfer to take this into account by simulating the modified Hamming rather than the pure Hamming.

Users who are working with PC scores can alter the apodisation of the reconstructed radiances by apodising the reconstruction operator and mean radiance vector. For the global PCs, this can be a one-off operation.

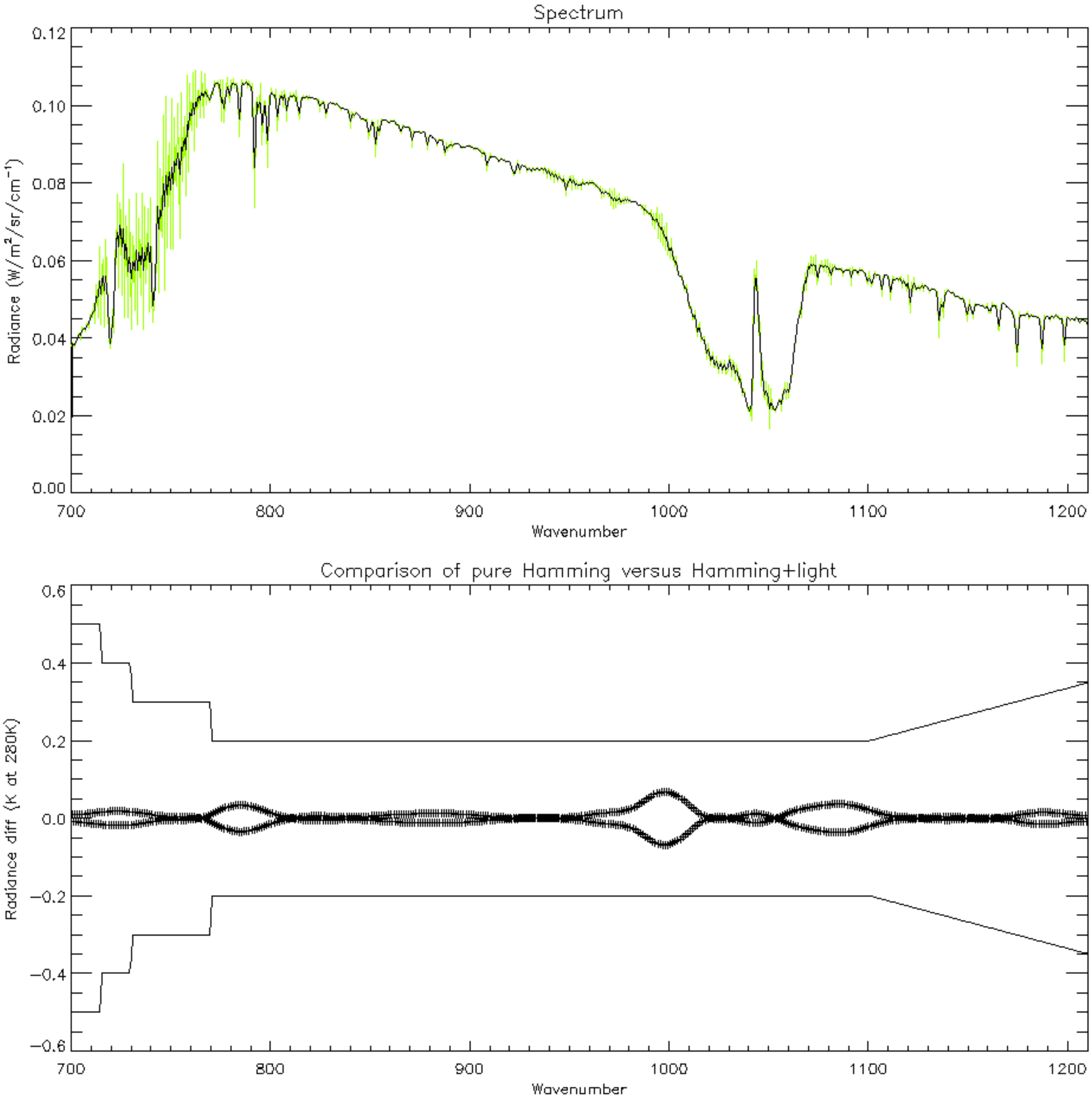

Figure 8: Top: a typical simulated LWIR spectrum with light apodisation (green) and Hamming apodisation (black). Bottom: crosses show the difference between pure Hamming and Hamming applied on top of light. The solid lines at top and bottom show the instrument noise specification.

8.6 Basis function transformation

For some applications it may be required to transform EUMETSAT's global PC scores into PC scores appropriate to a different basis function, e.g. based on model-generated spectra. The noise normalisations may also be different (e.g. the second may include forward model uncertainty). If subscript 1 refers to the initial scores and subscript 2 the final scores, then using (3) and (4) we have:

$$\mathbf{p}_1 = \mathbf{C}_1\,(\mathbf{r} - \mathbf{r}_{m,1}) \tag{9}$$

$$\begin{aligned} \mathbf{p}_2 &= \mathbf{C}_2\,(\mathbf{r} - \mathbf{r}_{m,2}) \\ &\approx \mathbf{C}_2\,(\mathbf{r}'_1 - \mathbf{r}_{m,2}) \\ &= \mathbf{C}_2\,\mathbf{R}_1\,\mathbf{p}_1 + \mathbf{C}_2(\mathbf{r}_{m,1} - \mathbf{r}_{m,2}) \end{aligned} \tag{10}$$

It is advisable for the number of PCs in the second basis function to be less than or equal to the number in the first, otherwise the PCs will not be independent. The transformation operator $\mathbf{C_2}$ $\mathbf{R_1}$ has *npc*$_2$ rows and *npc*$_1$ columns, while the adjustment to the mean $\mathbf{C_2(r_{m,1} - r_{m,2})}$ has *npc*$_2$ rows.

Having computed the new PC scores, we can then, if required, use $\mathbf{R_2}$ and $\mathbf{r_{m,2}}$ to compute the corresponding reconstructed radiances.

8.7 Support for PC-RTTOV

IRSPP v1.0 supported transformation to the PC basis function of the Havemann-Taylor Fast Radiative Transfer Code (HT-FRTC, Havemann et al., 2018). But this code has been withdrawn from RTTOV, and users who require the ability to forward-model PC scores are encouraged to use PC-RTTOV instead (Matricardi, 2010). In brief, PC-RTTOV works as follows:

1. For clear sky, in the “training” stage, a line-by-line radiative transfer model (LBLRTM) is used to generate high-resolution (0.001 cm^{-1}) spectra covering a wide frequency range, for 195000 model profiles.
2. For each supported instrument, the training spectra are convolved with the instrument line shape to produce polychromatic radiances.
3. For the scattering simulations (aerosols, clouds), instead of LBL the fast Chou-scaling solver is used with RTTOV v14, for the given instrument. There are 63135 training profiles for aerosols and 15000 training profiles for clouds, giving a total of 273135 training profiles for the RTTOV v14 PC coefficients.
4. 400 Principal Components are computed based on the polychromatic radiances.
5. A regression is computed to link PC scores to polychromatic radiances, for a specifically chosen set of “classical” RTTOV radiances. The result is stored in a PC coefficient file (e.g. *pccoef_mtg_2_irs-hamming-2mopd_v13p_landsea_6gas_clr_nlte_aer_hydro.nc*).
6. At run time, the user uses normal RTTOV (based on optical depth files) to simulate radiances for a set of channels specified in the PC coefficient file (the actual number of channels is user-configurable: a larger subset is slower but yields more accurate simulations), then the regression coefficients from step 5 are applied to compute the PC scores. Optionally, the PC scores can be used to compute reconstructed radiances for the user’s choice of channels.

IRSPP provides the capability of converting EUMETSAT’s PC product into equivalent PC-RTTOV scores (see section 8.6). The PC-RTTOV coefficient file for IRS (provided with RTTOV) includes eigenvectors (Hamming-apodised) and a 1-D noise profile (a “mean radiance” is not used). Thus we can compute the PC-RTTOV compression and reconstruction operators $\mathbf{C_2}$ and $\mathbf{R_2}$.

Note that in PC-RTTOV the two bands are combined together whereas the EUMETSAT PC product treats them separately. Therefore we need to merge the (apodised) EUMETSAT reconstruction operators ($\mathbf{R_{LW}}$ and $\mathbf{R_{MW}}$) before computing the transformation operator ($\mathbf{C_2}$ $\mathbf{R_1}$):

$$\mathbf{R_1} = \begin{pmatrix} \mathbf{R_{LW}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R_{MW}} \end{pmatrix} \qquad \textit{channels}\downarrow \quad \xrightarrow{\textit{scores}} \tag{11}$$

Also, the EUMETSAT PC scores for the two bands need to be concatenated together before they are transformed.

PC-RTTOV accommodates up to 400 PCs whereas EUMETSAT only provides 300. Mathematically, this is not a problem for the matrix multiplication in Eq. 10. But the user should be aware that the resulting PC scores will not be independent. IRSPP provides a namelist option to use only a user-defined number of leading PCs if required.

For the reconstruction operators (to convert from up to 400 PC-RTTOV PC scores to reconstructed radiance), a separate reconstruction matrix is used for each band.

Note that when encoding the transformed PC scores into BUFR, *npc_out_lw* should be set to the total number of PCs (up to 400) and *npc_out_mw* to 0, to avoid unnecessary duplication.
In conclusion, IRSPP supports the ingest and use of RTTOV's PC-RTTOV coefficient files, as described above, and illustrated in Figure 9. It is an open question as to whether there are scientific benefits in such a treatment.

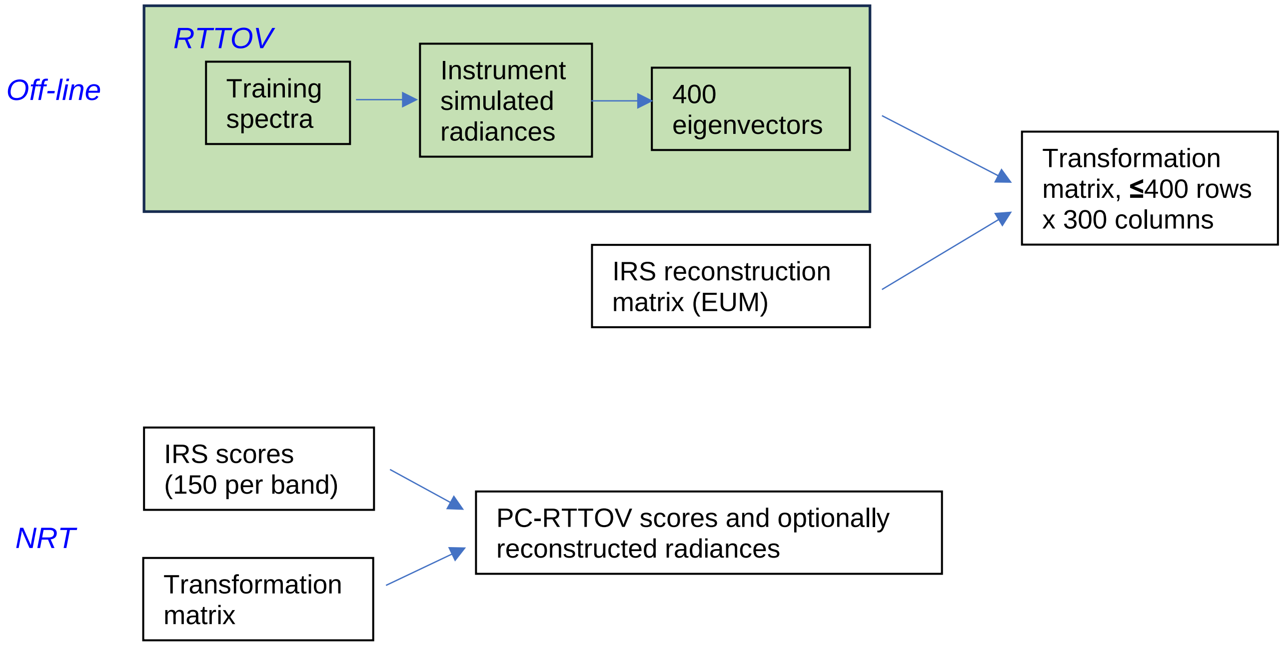

Figure 9: Top: off-line creation of a transformation matrix. Bottom: conversion of disseminated PC scores into equivalent PC-RTTOV scores

An alternative (but more time-consuming) way of using PC-RTTOV could be to use it to generate simulated spectra from a set of model profiles (e.g. using the NWP SAF Radiance Simulator, as mentioned in section 4.2, and noting that IRSPP supports the reading of RadSim netCDF output files). The user can then use the workflows of Figure 2 and Figure 3 to generate a covariance dataset and new PCs that are fully consistent with the model variables used in the simulation, and with the supplied noise profile. A transformation operator can then be computed for each band (as in section 8.6) to allow EUMETSAT PC scores to be readily transformed to the new PC scores.

9. TEST CASES

9.1 Common features

The test cases are delivered as a gzipped tar file. When unpacked, the following subdirectories will be seen:

- scripts – containing Fortran namelists and the calling bash scripts
- input – a single dwell of IRS data, in both PC and SSS format
- ancillary – eigenvector files
- plots

Before running a script, you need to define IRSPP_HOME:

`export IRSPP_HOME=[`directory where you installed IRSPP]

If you want to manually examine output files with *bufr_dump* (or another ecCodes tool) then you should set up your ECCODES_DEFINITION_PATH by typing:

```
. $IRSPP_HOME/bin/irspp_env.sh
```

Please cd to the "scripts" directory to continue, and run them from that directory. It is assumed that the namelists are in the same directory as the scripts. Detailed instructions for running the test cases are provided in a 00README.txt file.

The test cases distributed with IRSPP v1.3 all use the IRS "2022" spectral grid (Table 6).

9.2 Converting PC files to reconstructed radiance with BUFR output

To run:

```
./run_irs_main_parallel.sh
```

Input: PC files for LAC4 in netCDF from EUMETSAT
Output: each input file is converted to BUFR and placed in the output directory

Note that

- The script does not include the option to append reconstructed radiances to the input file. To use this facility you can call *irs_main.exe* directly
- The channel selection namelist is `required_channels_Coopmann.nl`

The maximum number of files that can be processed simultaneously is set in the script (variable *nmax*); users can experiment with different values.

A Python visualisation script, *plot_radiance_map.py*, is also provided, to allow the user to plot radiances as a map. It uses *numpy*, *matplotlib* and *cartopy*. See the README file for more information.

9.3 Generating covariance from SSS file

To run:

```
./run_irs_covariance.sh
```

Input: SSS file (all channels) in netCDF from EUMETSAT
Output covariance file in netCDF format (in the output directory): `irs_covariance.nc`

Note: if the output file already exists, the covariance in that file will be *updated*, and the number of used spectra increased.

Namelists used:

- `irs_covariance.nl` – this specifies whether any spatial thinning to be performed. Default is to use all spectra.

9.4 Generating new eigenvectors from a covariance file

To run:

```
./run_irs_generate_pcs.sh
```

Input: Covariance file from step 9.3

Output eigenvector file (in the output directory): `irs_pcs.h5`

Namelists used:

- `irs_generate_pcs.nl` – specifies the static file containing noise profile and the number of PCs to be generated in each band. This number is set to 120 in order to distinguish the resulting product from EUMETSAT's PC product that has 150 PCs.

9.5 Applying a user-generated eigenvector file to SSS input

To run:

```
./run_ irs_sss_filtering.sh
```

Inputs:

- Eigenvector file from step 9.4
- SSS file

Namelists used:

- `irs_sss_filtering.nl` – specifies the eigenvector file and lower-level namelists
- `required_channels_Coopmann.nl`
- `irs_bufr.nl`

Output file (in the output directory): `irs_sss_filtering.bufr` – containing PC scores and reconstructed radiances

Note that in this test case we have used the same SSS file that was used in step 9.3 to generate the covariances. Normally, generation on the covariances would be a separate step, using a dedicated set of training spectra.

This script also provides "reference" output for comparison using *irs_main.exe*: same channel selection but using raw radiances not reconstructed.

9.6 Converting PC file to PC-RTTOV basis function

The script *run_irs_pcrttov.sh* has two parts. Firstly it creates a matrix to transform from EUMETSAT PCs to the PC-RTTOV PCs. Then it applies the transformation matrix to a PC file and generates BUFR output containing a new set of PC scores, together with reconstructed radiances.

The script also processes the same PC file using the EUMETSAT PCs, so that the user can compare the results.

To run it:

```
./run_ irs_pcrttov.sh
```

9.7 Generating covariance from RTTOV / RadSim simulation

The input for this test is a RadSim file of simulated IRS radiances in netCDF format, generated from Met Office model fields.

The file contains 14211 spectra for a single model cycle, where the sample positions are based on a full disk of simulated IRS data (from EUMETSAT). The sample positions have been thinned to 8x8 spectra per IRS dwell (full resolution is 160x160 spectra), and only spectra where the satellite zenith angle is less than 70° have been selected.

RadSim generates 881 spectral samples in the long-wave band (679.703 cm^{-1} to 1210.439 cm^{-1}) and 1079 in the short-wave (1599.769 cm^{-1} to 2250.543 cm^{-1}). There is an attribute “wavenumbers” that contains this information.

We have configured RadSim to generate *Hamming-apodised* radiances.

The test script *run_irs_radsim.sh* generates a covariance file based on these simulated spectra. To run it:

```
./run_ irs_radsim.sh
```

One could then go on to generate eigenvectors and a transformation operator, but that is not implemented in this test as its main function is to demonstrate successful ingest of the RadSim file.

10. BUFR SEQUENCE

This section details the BUFR sequence provided in IRSPP. It supports inclusion of PC scores, radiances or both. The number of PCs or channels is set by delayed replication. Note that two locally-defined quality flags are used: *irsSpatialSampleQuality* and *irsDetectorSampleQuality*. The bit definitions follow those given in RD-3. IRSPP does not generate any quality information itself, it simply transfers the input quality flags into the BUFR. These flags are netCDF variables 'detector_sample_quality' and 'spatial_sample_quality', provided for the two bands [RD-3].

We have included the mean and standard deviation of “imager-mode radiances”, providing a measure of scene inhomogeneity. This reflects the fact that whereas a dwell comprises 160x160 IRS pixels, each of these actually comprises a 3x3 array and is reported as a broad-band radiance. We have re-used IASI BUFR descriptors that refer to AVHRR.
The sequence provides the possibility of including different geolocation for the two bands, though at present the EUMETSAT test data only includes one set of geolocation value (based on LW band).

```
001007 satelliteIdentifier (CCT C-5, 72=Meteosat-13=MTG-S1, 75=Meteosat-16=MTG-S2)
001033 centre (CCT C-1)
001034 subCentre (CCT C-12)
002019 satelliteInstruments (CCT C-8, 212 = IRS)
002020 satelliteClassification (334 = MTG)

301011 (year, month, day: 004001, 004002, 004003)
301012 (hour, minute: 004004, 004005)
207003 (increase scale and width)
004006 second
207000 (reset scale and width)
201135 (increase width from 8 to 15 bits)
005043 fieldOfViewNumber (1 to 25600 - 160x160)
201000 (reset width)
005041 scanLineNumber
005045 fieldOfRegardNumber (dwell)
002165 radianceTypeFlags (4=apodized, 5=unapodized)
202126 (decrease scale)
201132 (increase width)
007001 heightOfStation (m, to nearest 100m, geostationary height range)
201000 (reset width)
202000 (reset scale)
005066 spacecraftYaw (0 or 180 degrees)

124002 Replicate 24 descriptors 2 times (for the 2 bands)

008076 band (2=LW, 3=MW)
006029 waveNumber (start)
006029 waveNumber (end)
025140 startChannel
025141 endChannel
301021 (latitude, longitude: 005001, 006001)
007024 satelliteZenithAngle
005021 bearingOrAzimuth
007025 solarZenithAngle
005022 solarAzimuth
025142 channelScaleFactor
014047 scaledMeanAvhrrRadiance (imager mode, 3x3 pixels)
014048 scaledStandardDeviationAvhrrRadiance
033230 irsSpatialSampleQuality (14 bit flag table - local definition)
033231 irsDetectorSampleQuality (5 bit flag table - local definition)
025187 confidenceFlag (0=valid, 1=invalid, 15=missing)
207002 (increase scale and width)
040026 scoreQuantizationFactor
207000 (reset scale and width)
040016 residualRmsInBand
025062 databaseIdentification
101000 (replication)
031002 extendedDelayedDescriptorReplicationFactor
040017 nonNormalizedPrincipalComponentScore

008076 band (set to missing)

104000 (replication)
031002 extendedDelayedDescriptorReplicationFactor
201133 (increase bit width)
005042 channelNumber
201000 (reset bit width)
```

```
014044 channelRadiance

Quality flags:
 spatial_sample_quality, 14-bit (bit 1 defined here as most significant)
  Bit  Meaning
   1-4 reserved
   5   solar_straylight_correction_warning
   6   solar_straylight_warning
   7   noisy_detector_sample_warning
   8   undersaturated_detector_sample_warning
   9   saturated_detector_sample_warning
  10   dust
  11   cloudy
  12   limb_view
  13   space_view

 detector_sample_quality, 5-bit (bit 1 defined here as most significant)
  Bit  Meaning
   1   excluded_detector_sample
   2   noisy_detector_sample
   3   undersaturated_detector_sample
   4   saturated_detector_sample
```

It is a BUFR regulation that in a flag table the least significant bit remains unused, to allow “missing” to be uniquely defined (i.e. all bits set).

11. REFERENCES

Antonelli, P., H.E. Revercomb, L.A. Sromovsky, W.L. Smith, R.O. Knuteson, D.C. Tobin, R.K. Garcia, H.B. Howell, H.-L. Huang, and F.A. Best (2004): A principal component noise filter for high spectral resolution infrared measurements, J. Geophys. Res., 109(D23). https://doi.org/10.1029/2004JD004862

Collard, A. D. (1997): Selection of IASI channels for use in numerical weather prediction, Quarterly Journal of the Royal Meteorological Society, Volume 133, Issue 629, pages 1977-1991, October 2007 Part B. https://doi.org/10.1002/qj.178

Collard, A. D., A. P. McNally, F. I. Hilton, S. B. Healy, N. C. Atkinson (2010): The use of principal component analysis for the assimilation of high-resolution infrared sounder observations for numerical weather prediction, Quarterly Journal of the Royal Meteorological Society, Volume 136, Issue 653, pages 2038-2050, October 2010 Part B. https://doi.org/10.1002/qj.701

Coopmann, O., N. Fourrié, V. Guidard (2022): Analysis of MTG-IRS observations and general channel selection for numerical weather prediction models, Quarterly Journal of the Royal Meteorological Society, Volume 148, Issue 745, pages 1864-1885, April 2022, https://doi.org/10.1002/qj.4282

Coppens, D., B. Theodore, T. August, T. Hultberg, C. Goukenleuque, Jochen Grandell (2019): MTG-IRS: Scientific Improvements For a User-Friendly Mission. Proc. ITSC-22, https://itwg.ssec.wisc.edu/conferences/past-itsc-meetings/itsc-22-program/

Havemann, S., J-C. Thelen, J. P. Taylor, R. C. Harlow, 2018: The Havemann-Taylor Fast Radiative Transfer Code (HT-FRTC): A multipurpose code based on principal components, Journal of Quantitative Spectroscopy and Radiative Transfer, Volume 220, Pages 180-192, https://doi.org/10.1016/j.jqsrt.2018.09.008

Hultberg, T., August, T. and Lenti, F. (2017a): Local or global? How to choose the training set for principal component compression of hyperspectral satellite measurements: a hybrid approach. In Sensors, Systems, and Next-Generation Satellites XXI (Vol. 10423, p. 104231G). International Society for Optics and Photonics, https://doi.org/10.1117/12.2278349

Hultberg, T., August, T. and Lenti, F. (2017b): A global-local hybrid approach to retain new signals in hyperspectral PC products. Proc. ITSC-21, https://itwg.ssec.wisc.edu/conferences/past-itsc-meetings/itsc-21-program/