	Cloud and Aerosol Detection Software Version 3	Doc ID : NWPSAF-EC-UD-015 Version : v1.6 Date : 15/02/2024

Cloud and Aerosol Detection Software

Version 3

User Guide

Version 1.6
15/02/2024

Chris Burrows
ECMWF

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.

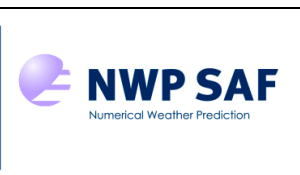| Change record			
Version	Date	Author / changed by	Remarks
1.0	27/04/2020	Reima Eresmaa	Initial version
1.1	02/06/2020	Reima Eresmaa	Clarify based on beta testers' reports
1.2	19/06/2020	Reima Eresmaa	Include more information in Tables I-IV
1.3	22/06/2020	Reima Eresmaa	Include aerosol detection comparison with visible satellite imagery
1.4	01/07/2020	Reima Eresmaa	Add references on aerosol type classification and trace gas detection. Include information on fast RT coefficients used in tuning the default parameter values.
1.5	06/07/2020	Reima Eresmaa	Emphasize the independence of each output flag type. Clarify on the aerosol detection accuracy over land, on the use of IR (10 μm) AOD estimate, and on the concept of channel height assignments. Clarify the equivalence of parameters with names used in the namelists.
1.6	15/02/2024	Chris Burrows	Update to cover extended functionality in CADS v3.2

TABLE OF CONTENTS

	Cloud and Aerosol Detection Software Version 3	Doc ID : NWPSAF-EC-UD-015 Version : v1.6 Date : 15/02/2024

1 INTRODUCTION

The Cloud and Aerosol Detection Software (CADS) is a flexible stand-alone cloud and aerosol detection package intended for radiance assimilation studies using high-spectral-resolution infrared (IR) sounders. The code was originally developed at the European Centre for Medium-range Weather Forecasts (ECMWF) to facilitate operational assimilation of AIRS (Atmospheric Infrared Sounder) radiances, and it has since been adapted to handle several other IR sounders. The software is updated regularly to maintain the capability to reproduce performance characteristics as implemented in the ECMWF 4D-Var data assimilation system. The software is consistent with the widely adopted practice of assimilating radiance data in the form of Brightness Temperature (BT), and it is equally applicable to original radiances and radiances reconstructed from principal component scores.

Since the first release, the software has undergone several updates to respond to user requests and developments in the ECMWF 4D-Var system. The most important milestones in the history of CADS are listed as follows:
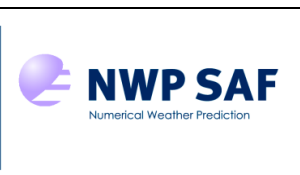
- Original release in 2007 to provide the detection of cloud and aerosol for AIRS and IASI (Infrared Atmospheric Sounding Interferometer) radiances (V1.1).
- Extend the processing capability to include the CrIS (Cross-track Infrared Sounder) in November 2011 (V1.3).
- In January 2015, introduce the use of collocated imager information as part of the cloud detection (V2.1).
- Make the aerosol detection independent of the cloud detection in November 2015 (V2.2).
- In February 2017, extend the processing capability to include the next generation European IR sounders (V2.3).
- Extend the processing capability to include the Chinese hyperspectral IR sounders (HIRAS; Hyperspectral Infrared Atmospheric Sounder, and GIIRS; Geostationary Interferometric Infrared Sounder) in March 2019 (V2.4).

Version 3.1 is a significant upgrade and provides the following new functionalities:

- Extending the processing capability to include the IKFS-2 (Infrared Fourier Spectrometer -2).
- Discriminating volcanic ash and Saharan dust from other types of aerosol.
- Making the flagging of aerosol-affected channels to account for the aerosol type.
- Introducing the detection of excessive trace gas concentration in IR sounder's field-of-view (FOV).
- Introducing the situation-dependent flagging of channels that are sensitive to underlying land surface.

Version 3.2 is a minor upgrade and provides the following new functionalities:

- Support is provided for HIRAS-2.

- Support is provided for IRIS.
- Support is provided for VIIRS imager clustering data which can augment the cloud detection for CrIS.
- Bug fix to avoid potential memory leaks in the aerosol detection code.

The output flags indicate channels that are affected by cloud, aerosol, excessive trace gas concentration, and sensitivity to underlying land surface at each observation location. For each of the four possible contaminants, the flagging is made independently of others. In other words, any combination of the four output flags is possible for each IR channel at each observation location.

While the software by design reproduces the cloud, aerosol, trace gas, and land sensitivity flagging as implemented in the ECMWF 4D-Var system, users are provided with considerable flexibility to make the processing better meet their needs. For each processing parameter, the source code assigns a default value, that may be overridden by specifying another value in a namelist file. Ideally, each user should optimise the various processing parameters locally to match the characteristics of their specific application. The optimal tuning is likely sensitive to systematic and random errors not just in observations, but also in forward modelling and NWP model background. The default parameter values provided in the software reflect the experience gained from IR radiance assimilation in the global NWP system of ECMWF. Table I lists the fast radiative transfer (RT) model coefficients that have been used when determining the default processing parameter for each supported IR sensor. The ECMWF RT modelling experience is solely from the RTTOV software.

The default processing parameters specified in the source code for AIRS, CrIS, IASI and IRIS are based on real measurement data and they are identical to those to be implemented in version CY48R1 of the Integrated Forecasting System (IFS) at ECMWF (the forthcoming ERA-6 reanalysis in the case of IRIS). This is also the case with the default cloud detection and land sensitivity detection settings for HIRAS, HIRAS-2, GIIRS, and IKFS-2: users however should be aware that the aerosol and trace gas detection settings provided for these Chinese and Russian sounders are inactive and will be revised later when sufficient experience from real measurement data has been collected. For the future European IR sounders, i.e. the IRS (Infrared Sounder) and IASI-NG (Infrared Atmospheric Sounding Interferometer – New Generation), respectively onboard the Meteosat Third Generation (MTG) and EUMETSAT Polar System – Second Generation (EPS-SG) platforms, all default processing parameters are based on simulated data only for now. The parameters for IRIS were tested as part of preparations for the ERA-6 reanalysis and are based on Poli and Brunel (2018).

Table I: Fast radiative transfer model coefficients used at ECMWF.

Sensor	RT Coefficients	Notes
AIRS	kCARTA v1.11	
CrIS	LBLRTM v11.1	LBLRTM v12.2 used from November 2019 onwards, but

		without changing the CADS processing parameters
HIRAS	LBLRTM v12.2	
HIRAS-2	LBLRTM v12.8	
GIIRS	LBLRTM v12.6	
IASI	kCARTA v1.11	
IASI-NG	LBLRTM v12.2	
IKFS-2	LBLRTM v12.2	
IRS	LBLRTM v12.2	
IRIS	LBLRTM v12.8	

2 THE DETECTION ALGORITHMS

The software consists of four main modules that are mutually independent and provide the detection of cloud, aerosol, excessive trace gas absorption, and sensitivity to land.

2.1 Cloud detection

A detailed description of the cloud detection algorithm for the assimilation of hyperspectral IR radiances in the ECMWF 4D-Var system is given in McNally and Watts (2003) and more recently in Eresmaa (2014); the latter reference describes the imager-assisted functionality of the cloud detection in detail.

The IR channels that are taken into consideration in the cloud detection algorithm are specified in a two-dimensional array (`N__Bands` in the code and namelist files). Up to five spectral bands (the first dimension of the array) can be defined. The number of channels included in each band should be defined in the array `N__Band_Size`. Within each spectral band, the cloud detection algorithm computes channel-specific background departures

$$\delta = BT_{ob} - BT_{bg}$$

i.e., the difference between the BT observation and its background-modelled counterpart. The background value BT_{bg} is calculated by applying a radiative transfer model on an atmospheric state estimate – most typically a short-range NWP forecast[1]. It is essential that BT_{bg} is calculated assuming no cloud, such that the radiative effect of cloud becomes detectable in δ. To reveal the cloud radiative effect, and to separate it from other contributions of δ (as far as possible), the IR sounder channels are ranked in vertical, using channel height assignments (see Appendix A) provided as input to the algorithm. The resulting vertically ranked array of δ is also smoothed by a moving average filter (the width of the filter in each spectral band is specified in the namelist parameter `N__Window_Width`). Using the smoothed and vertically ranked array of δ (hereafter δ_{smooth}), one of three possible cloud detection scenarios is chosen. Where collocated imager data is available, the choice of the scenario makes use of this as well. The three scenarios account for situations where (i) there is no cloud in the IR sounder's FOV, (ii) there is cloud in the FOV and it is warmer than underlying surface, and (iii) there is cloud in the FOV and it is colder than underlying surface. Once the scenario is chosen, the detection proceeds to the cloud search where cloud-affected channels are distinguished from clear ones.

2.1.1 Choosing the cloud detection scenario

The choice of the cloud detection scenario relies on δ_{smooth} and on collocated imager data if that is available. The diagrams in Figs. 1-3 illustrate typical characteristics of the three

---

[1] Note that it is user's responsibility to provide observed and background BT information as input. No radiative transfer calculations are performed inside CADS.

possible scenarios. In each, the heavy black line indicates δ_{smooth}, the highest-ranked channels being at far left and the lowest-ranked channels at right. Dashed lines show positions of two channels with special role in the cloud detection process: these correspond to approximate levels of the tropopause and the top of the planetary boundary layer (PBL).

Potential starting channels

Before the cloud detection scenario is chosen, three channels are identified in δ_{smooth} as potential starting channels for the cloud search. These are highlighted by red letters (A, B, C) in the diagrams below, and their definitions are as follows:

  A: Considering only channels ranked between the tropopause and the top of PBL, the channel at which δ_{smooth} reaches its minimum value
  B: Considering all channels ranked below the tropopause, the channel at which δ_{smooth} reaches its minimum value
  C: The bottom channel, i.e., the channel that has height assignment closest to the surface

2.1.2 Quick Exit scenario

The Quick Exit scenario corresponds to the hypothesis of a completely cloud-free FOV and it results in flagging all channels clear. Properties of δ_{smooth} in Quick Exit are illustrated in Fig. 1. As the Quick Exit scenario provides the highest possible number of clear (and consequently usable) channels for the assimilation, it is a desirable outcome of the cloud detection. However, several criteria need to be met for the Quick Exit to take place. These are listed as follows:

- Absolute value of δ_{smooth} must be smaller than a pre-defined departure threshold (`R__BT_Threshold` in the code) at all of the initially considered potential starting channels (A, B, C)
- δ_{smooth} must not exceed the departure threshold (`R__BT_Threshold`) anywhere among the channels assigned below the tropopause
- Among the channels in the long-wave window wavelengths (i.e., between channel indices specified in `N__Window_Bounds`), δ_{smooth} must not change more than a pre-defined window-gradient threshold (`R__Window_Grad_Threshold`)
- If collocated imager data is provided (and `L__Do_Imager_Cloud_Detection` is set to `.TRUE.`), the preliminary cloud flag derived from it must not indicate presence of cloud in the FOV
- The Quick Exit scenario must be allowed by the logical variable setting `L__Do_Quick_Exit=.TRUE.` (this is the default setting)
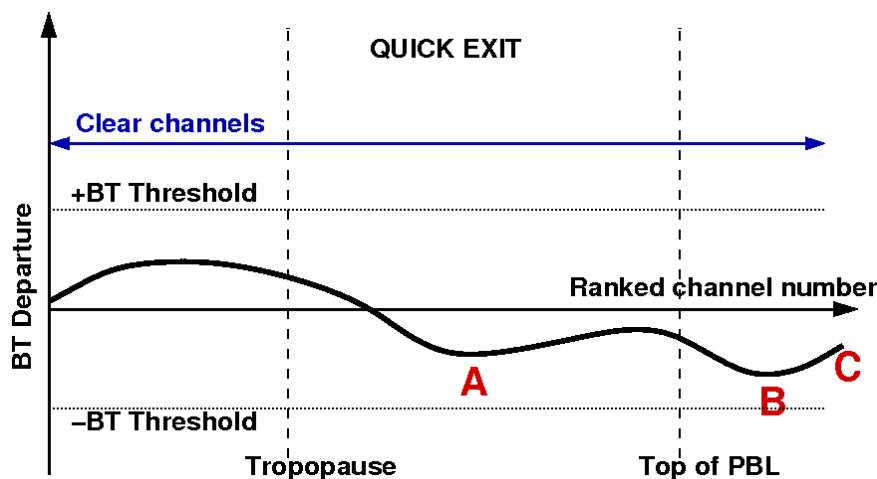

Figure 1: A schematic representation of δ_{smooth} in a Quick Exit Scenario.

2.1.3 Warm Start scenario

The Warm Start scenario accounts for cases where there is a warm cloud above a cold surface. The following criteria are met in a Warm Start scenario:

- The possibility of the Quick Exit scenario has been ruled out
- Absolute value of δ_{smooth} at channel A is less than the pre-defined departure threshold (`R__BT_Threshold`)
- δ_{smooth} at channel C channel is greater than the pre-defined departure threshold (`R__BT_Threshold`)

In the Warm Start scenario, cloud-affected channels are identified by starting the cloud search at channel C and proceeding upwards one channel at a time as long as either gradient or absolute value (or both) of δ_{smooth} is outside their respective thresholds (`R__Grad_Threshold` and `R__BT_Threshold` in the code). Around each channel, the gradient is calculated over an interval spanning several channels, as specified in `N__GradChkInterval`. In the example shown in Fig. 2, the cloud search finishes, and the lowest clear channel is found, near channel B (indicated by the red arrow).

There is another possible form of Warm Start scenario (so-called Abnormal Warm Start) that may take place in exceptional circumstances (δ_{smooth} exceeding the departure threshold `R__BT_Threshold` on all channels below the tropopause). Only channels ranked higher than the tropopause can potentially be diagnosed clear in such situations.
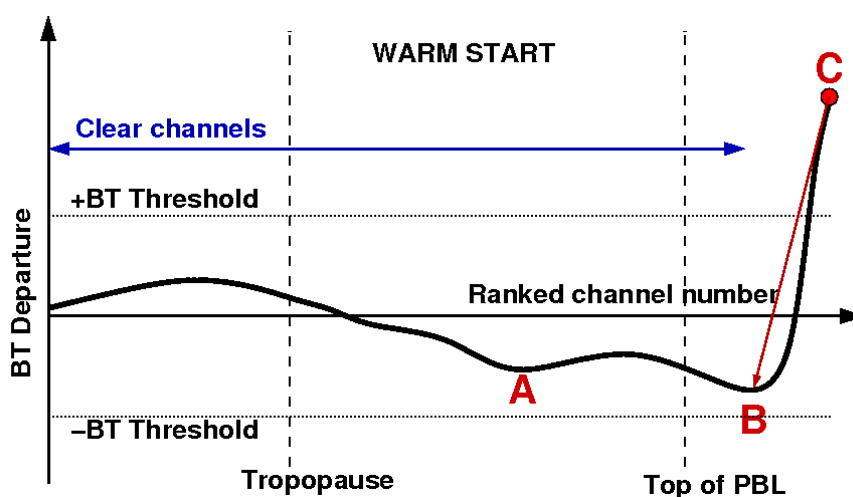

Figure 2: A schematic representation of δ_{smooth} in a Warm Start Scenario.

2.1.4 Cold Start scenario

The Cold Start scenario accounts for the most common situation, where there is a cold cloud above a relatively warm surface. The Cold Start takes place whenever neither Quick Exit nor Warm Start is applicable. Usually this means that δ_{smooth} is large negative – i.e., less than `-R__BT_Threshold` - at channel B at the very least. The Cold Start can also happen because of the gradient of the curve being too large among the long-wave window channels (i.e., larger than `R__Window_Grad_Threshold`), or because the preliminary cloud flag from collocated imager data suggests presence of cloud.

Figure 3 illustrates a typical Cold Start situation. Similarly to the Warm Start scenario, the Cold Start scenario aims to identify cloud-affected channels by starting the cloud search from one of the potential starting channels. Usually, the cloud search starts at channel A. It is only if neither the absolute value nor the gradient of δ_{smooth} suggests the presence of cloud at channel A that the search starts at channel B instead. The lowest clear channel is then found by proceeding upwards, in terms of the vertically-ranked channels, one channel at a time as long as either the gradient or absolute value (or both) of δ_{smooth} remains outside the gradient and departure threshold values (`R__Grad_Threshold` and `R__BT_Threshold`, respectively). The cloud search in a Cold Start scenario is visualized by the red arrow in Fig. 3.
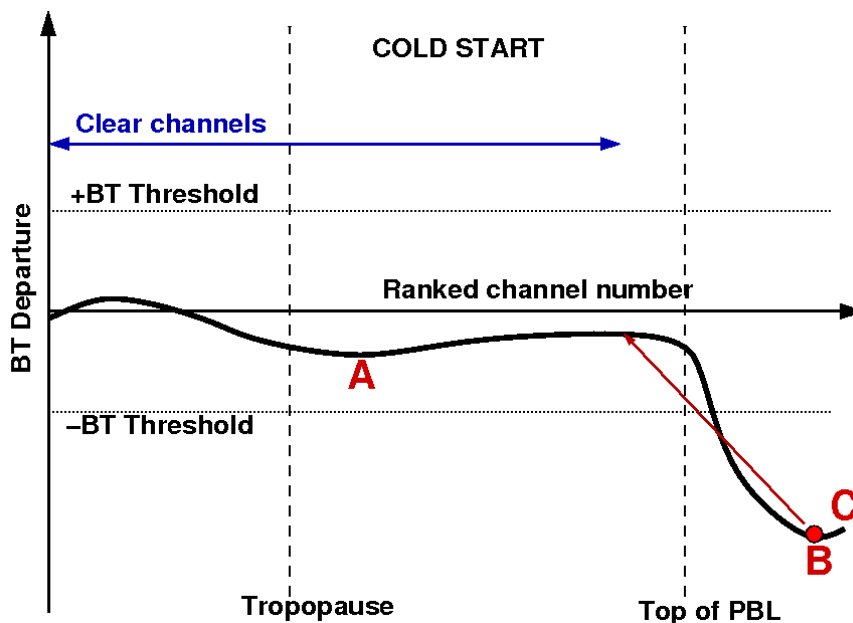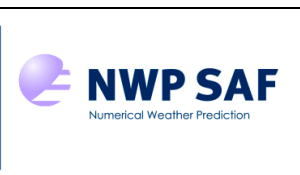

Figure 3: A schematic representation of δ_{smooth} in a Cold Start Scenario.

2.1.5 The preliminary cloud flag from collocated imager data

Whenever collocated imager data is available, there is a possibility to make use of it to obtain a preliminary cloud flag to assist the choice of the cloud detection scenario: it is a requirement of the Quick Exit scenario that this preliminary cloud flag supports the hypothesis of cloud-free FOV. The use of the collocated imager data is controlled by the namelist parameter `L__Do_Imager_Cloud_Detection`.

The imager-based cloud detection is currently implemented for IASI and CrIS, and it exploits mean and standard deviation of observed BT in clusters of AVHRR/VIIRS pixels. These clusters are thought to sit inside the sounder's FOV. Background-modelled BT estimates in the imager channels are required as additional input[2]. In the default configuration, the imager-based cloud flag uses data on the two long-wave IR channels of AVHRR/VIIRS.

For IR sounders other than IASI and CrIS, the imager-based cloud detection is currently not possible, so the preliminary cloud flag is always set clear.
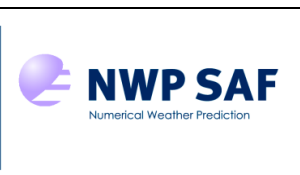
As explained in Eresmaa (2014), the imager-based cloud flag is determined from three separate checks, called *the homogeneity check*, *inter-cluster consistency check*, and *background departure check*. Clear imager-based flag is output only in the case that none of the three checks suggests presence of cloud.

2.1.6 Cross-band option

The cloud detection algorithm is best applied to individual bands rather than to combined input covering the entire infrared spectrum. Therefore, up to five bands are defined for each sounder covering the 15µm CO_2 absorption band, the O_3 absorption band, the 6.7µm H_2O absorption band, the long-wave side of the 4.3µm CO_2 absorption band, and the short-wave side of the 4.3µm CO_2 absorption band. The short-wave band is divided in two parts to distinguish between channels that are sensitive to solar radiation or non-LTE-effects and those that are not.

Optionally one may use the output of the cloud search in one band to infer cloud-affected channels in another band of channels. This is useful, for example, in the water vapour band where NWP background humidity errors may easily mask or mimic cloud signal. This is done by activating the "cross-band" (i.e., setting `L__Do_CrossBand=.TRUE.`; this is the default setting) option. When using the cross-band option one should consider whether the height assignments for spectrally widely separated channels compare meaningfully with each other. By default, the cross-band option is in use for the O_3 and H_2O absorption bands on all IR sounders, and for short-wave CO_2 absorption bands on interferometric IR sounders (i.e., all supported sounders except AIRS). In each of these cases, the referencing cloud search output comes from the long-wave CO_2 absorption band (i.e., `N__BandToUse(:)=1`).

---

[2] Again, provision of background BT for imager channels is on user's responsibility.

2.2 Aerosol detection

Aerosol detection consists of three steps. In the first step, contamination in the sounder FOV is diagnosed based on observed spectral BT differences in the long-wave IR window region (Letertre-Danczak, 2016). Four key channels are identified (denoted c_1, c_2, c_3, and c_4 in the following; in the code and namelist files, these are specified in the array `N__Aerosol_Chans`) and a representative BT value is determined around each. For the special case of AIRS, observed BTs at c_1, c_2, c_3, and c_4 are used as such; with interferometers, the effect of random instrument noise is mitigated by averaging over several neighbouring channels around the key channels (this is controlled by the namelist parameter `N__Mean_Aerosol_Chans`). From the representative BTs, two BT differences (c_1-c_2 and c_3-c_4) are formed and compared with pre-defined thresholds (set in the namelist parameter `R__Aerosol_TBD`). Aerosol contamination is diagnosed if, and only if, both BT differences fall below their respective threshold values. Because channel characteristics differ from one sounder to another, the thresholds are specific to each sensor. As far as possible, however, key channels in different sounders are chosen consistently at the same wavenumbers. The desired wavenumbers are 980 cm^{-1}, 1232 cm^{-1}, 1090.5 cm^{-1}, and 1234 cm^{-1} for c_1, c_2, c_3, and c_4, respectively.

Vidot (2017) has evaluated the performance of the first aerosol detection step by comparing the output cloud and aerosol flags of CADS v2.3 with the IASI aerosol product of Universite Libre de Bruxelles (ULB). In general, many more situations are found to contain aerosol in the ULB product than what is suggested by the CADS output. While this can be interpreted as a weakness in CADS to miss some aerosol situations, it should be emphasized that in majority of such cases the aerosol-affected channels were flagged as cloudy. These data would thus not be used in any subsequent retrieval or assimilation application.

In the case that no aerosol contamination is identified during the first detection step, all channels will be flagged clear of aerosol and no further steps are taken. Otherwise, the algorithm proceeds to the second step to determine the aerosol type (Eresmaa et al., 2020). For this purpose, representative BT is determined around two additional key channels (c_5 and c_6, the desired wavenumbers for these are 1168 cm^{-1} and 833 cm^{-1}, respectively). From the total of six key channels, three additional BT differences (c_5-c_2, c_3-c_1, and c_3-c_6) are formed. The detected aerosol is classified as volcanic ash in the case that the difference c_5-c_2 falls below its specific threshold value[3]. If the detected aerosol does not classify as volcanic ash, and if the differences c_3-c_1, and c_3-c_6 both fall below their respective thresholds, it is classified as Saharan dust. In the case that the detected aerosol is neither volcanic ash nor Saharan dust, it will remain unclassified and assigned the aerosol type index of other aerosol.

Figures 4 and 5 illustrate the output of this aerosol type classification in samples of Metop-B IASI data. The case studies in Fig. 5 indicate a reasonably good accuracy in detecting volcanic ash and Saharan dust over sea. The ash cloud coming from Raikoke volcano,

---

[3] This BT difference is consistent with the concept of "Ash Brightness Temperature Difference" as defined in Clarisse et al. (2010).

seen in the middle of bottom left panel, is well captured by the aerosol type classification scheme in the bottom right panel. Similarly, detections of Saharan dust above Red Sea, Arabian Sea, and Persian Gulf are in agreement with the imagery and there are very few false detections overall. The detections of other aerosol type tend to occur mostly in cloud-affected regions and may be false alarms though. While experience suggests good performance over sea, the classification tends to be unreliable over land. Therefore, the current practice is to reject all channels when any aerosol is found over land. Note however that the detection of any aerosol, i.e. the first step of the detection algorithm, is thought to be reliable both over land and sea (Vidot, 2017).

The third step is to distinguish between aerosol-affected and unaffected channels at each location where aerosol is found. As described in Eresmaa et al. (2020), the way this distinction is made depends on the aerosol type. In all cases, the affected channels are separated from unaffected ones by comparing channels' height assignments (see Appendix A) against a rejection threshold t_R. The height assignments are first normalized such that the highest- and lowest-ranked channels get values 0 and 1, respectively. The threshold t_R is then determined as follows:

- For volcanic ash aerosol, and for all aerosol types over land, t_R is set to 0, such that all channels get flagged as affected. For controlling the threshold setting over land, there is the namelist parameter `R__Land_Fraction_Thres` that determines the minimum land fraction considered for the IR FOV to be representative of land.

- For Saharan dust aerosol, t_R is determined as a function of IR (10 μm) Aerosol Optical Depth (AOD), that is estimated from the previously calculated BT difference c_3-c_4. The BT converts to the AOD estimate through the regression

$$AOD = p_1 + p_2 (c_3\text{-}c_4) + p_3 (c_3\text{-}c_4)^2,$$

where p_1, p_2, and p_3 are sensor-specific coefficients (`R__coef_AOD` in the code). The rejection threshold t_R follows from the AOD estimate through

$$t_R = (1/r_3) [(r_1/AOD) - r_2],$$

where r_1, r_2, and r_3 are, again, empirical coefficients (specified in the array `R__Rank_Thres_Coeff` in the code). The latter set of coefficients are currently set to r_1=-0.01, r_2=2.1, and r_3=-3.9. Figure 6 illustrates the conversion from the BT difference to AOD (on the left) and from AOD to t_R (on the right).
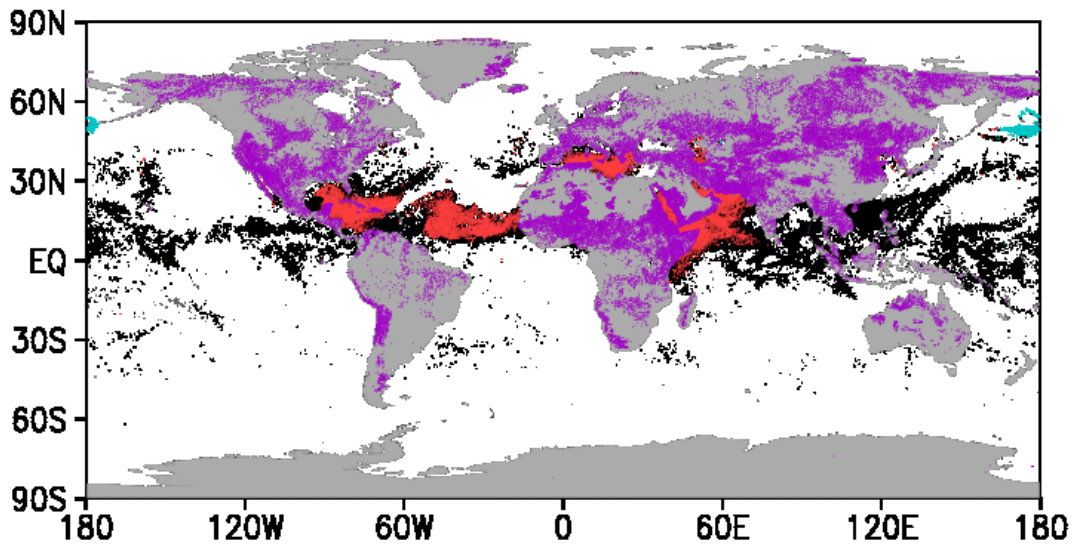
Figure 4: Aerosol type classification for Metop-B IASI on 23 June 2019. Red, cyan, and black dots are indicative of Saharan dust, volcanic ash, and unclassified aerosol, respectively. Purple dots indicate aerosol detected over land.
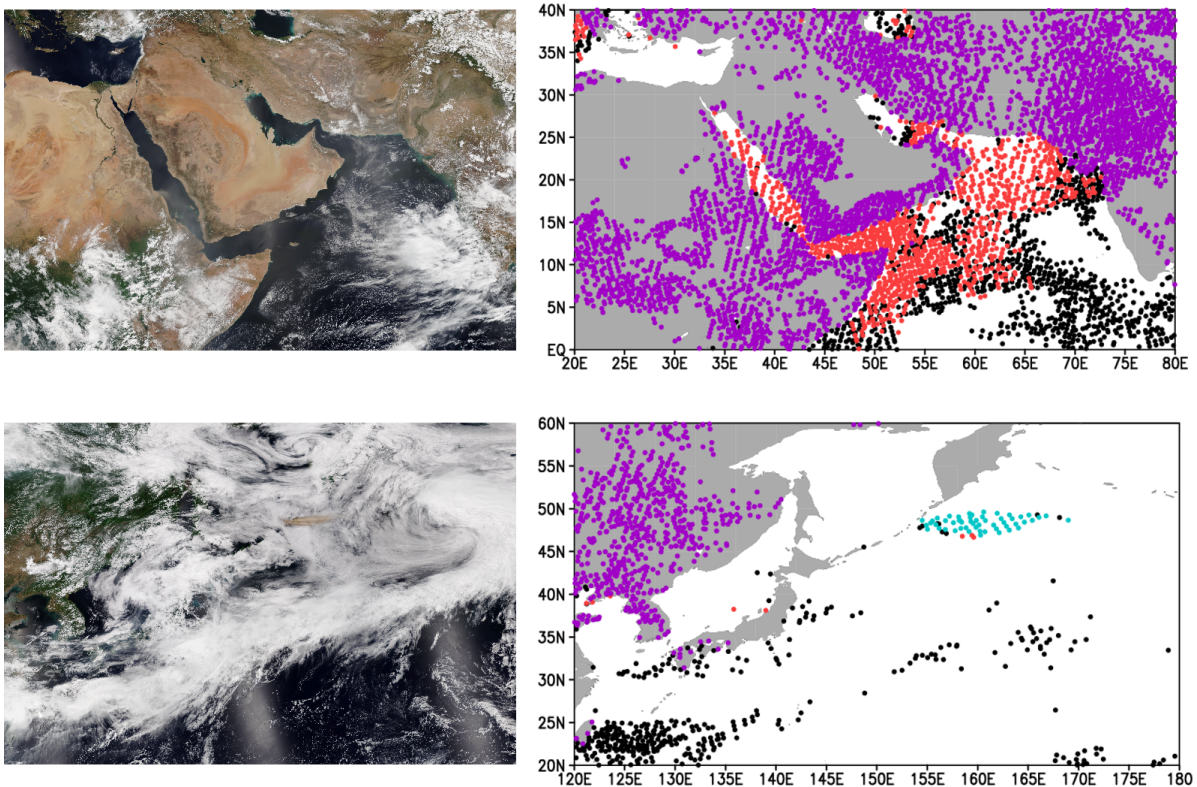

Figure 5: Comparison of aerosol type classification (right) with visible satellite imagery from Suomi-NPP VIIRS instrument (left) on 22 June 2019. Dots are coloured as in Fig. 4, and the two regions of interest are at Arabian Peninsula (top) and Northwestern Pacific Ocean (bottom).
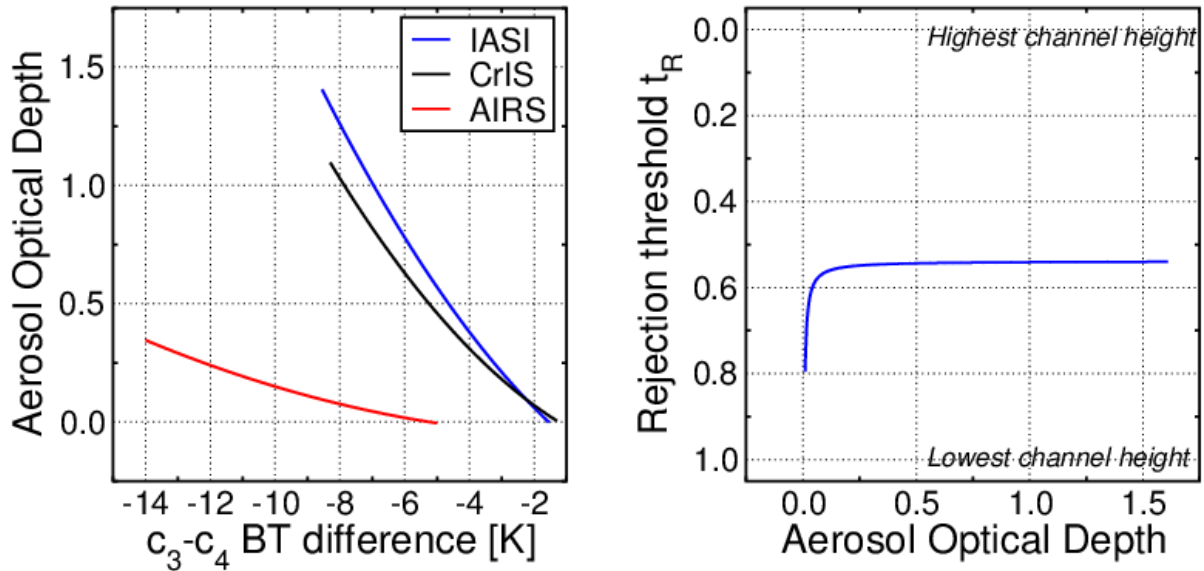
Figure 6: Conversion from BT difference to AOD (left) and from AOD to rejection threshold t_R (right).

- For other (unclassified) aerosol, stratospheric channels are considered free from contamination, while all tropospheric-sensitive channels are flagged as affected. The threshold value used here is currently set to t_R=0.4 (specified in variable `R__Unclassified_Thres` in the code).

2.3 Trace gas detection

The trace gas detection attempts to identify those locations where concentration of a particular trace gas exceeds the tolerance determined by climatological variability. The excessive trace gas concentrations have a significant potential to dramatically reduce the accuracy of the radiative transfer model in certain wavelengths. Each check carried out in the trace gas detection is tailored for one specific gas. The current implementation supports only the detection of excessive Hydrogen Cyanide (HCN). As an example, Fig. 7 shows locations where excessive HCN is found to affect CrIS radiances. The data shown in Fig. 7 was collected on 25-26 September 2019.
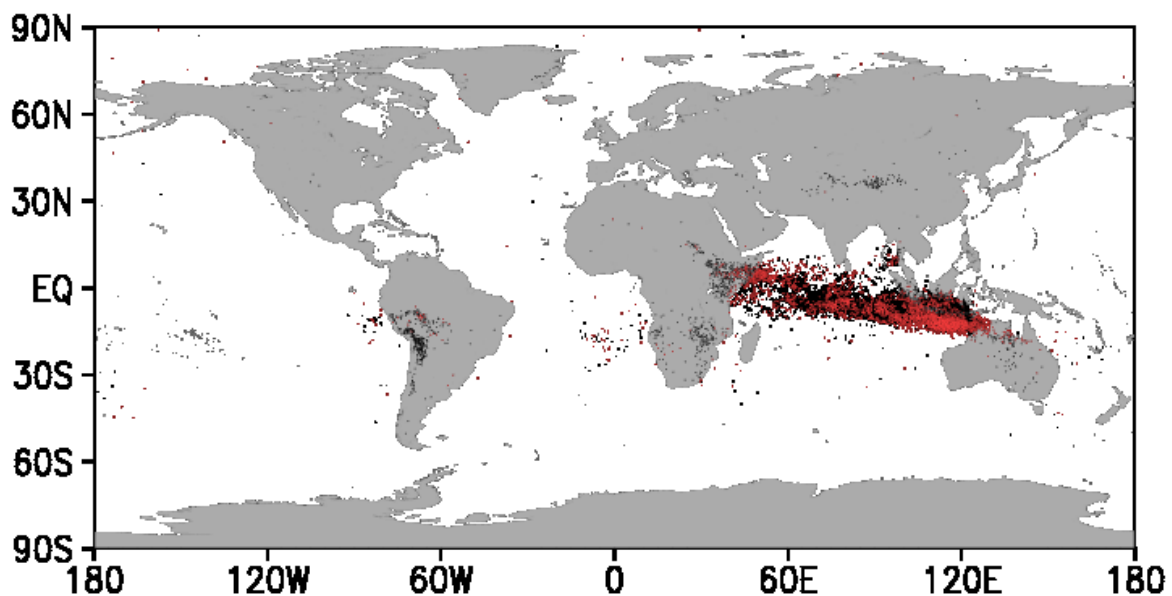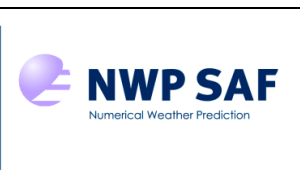

Figure 7. Locations of excessive HCN concentration affecting CrIS data in a two-day sample in September 2019. Black dots indicate simultaneous contamination from cloud, while red dots indicate HCN detection in the absence of cloud.

The trace gas detection method is described in Eresmaa et al. (2020). The input to the detection consists of the BT observation and its background departure δ,

$$\delta = BT_{ob} - BT_{bg}.$$

In the setup for the trace gas detection, three groups of channels are specified:

1) Tracer channels, that are chosen such that the sensitivity to the targeted trace gas is as high as possible. These are specified in the namelist array `N__Tracer_Channels`.

2) Control channels, that lack the sensitivity to the trace gas, but are otherwise similar to the tracer channels and can be thought to respond in the same way to given perturbations in temperature and humidity profiles. These are specified in the namelist array `N__Control_Channels`.

3) Flagged channels, that are positioned at the absorption lines of the trace gas and will be flagged as contaminated whenever excessive trace gas concentration is found. These are specified in array `N__Flagged_Channels`. There may be considerable overlap between channels listed in the tracer channel and flagged channel groups.

The trace gas detection algorithm computes the mean of observed BT and its background departure within the tracer and control channel groups. The means computed for the control channels are subtracted from those of the tracer channels, and the resulting differences are compared with pre-determined thresholds (`R__D_Obs_Threshold` and `R__D_Dep_Threshold` in the code and namelist files). In the case that both calculated parameters fall below their respective threshold values, presence of excessive trace gas is diagnosed. The trace gas flags are set to indicate contamination for those channels included in the flagged channels group.

2.4 Land sensitivity detection

Flagging for land sensitivity is provided as an option to enable users to reject potentially misbehaving channels from entering the retrieval or NWP analysis. In this context, the misbehaviour is attributed to the relatively large uncertainty in the description of land surface emission in the background modelling for BT – this may be due to difficulties in modelling surface emissivity and/or skin temperature over wide range of land surface types. The setup for the land sensitivity detection in CADS consists of specifying two threshold parameters. One is a threshold for the input land fraction (`R__Land_Fraction_Thres` in the code and namelist files), above which the land sensitivity detection activates. The other is the land sensitivity level threshold (`R__Level_Thres` in the code and namelist files). The algorithm normalizes all input channel height assignments (see Appendix A) such that the lowest-ranked channel is assigned with value 1.0. Those channels, that have their normalized height assignments in excess of the land-sensitivity level threshold, are flagged as land sensitive.

More generally, it is possible for the user to make the land sensitivity flags indicative of sensitivity to any surface type. This would be achieved by setting the land fraction threshold to 0, so that the land sensitivity detection activates at all locations.

3 DESCRIPTION OF THE CODE

The calling tree of CADS is shown in Fig. 8. Purpose of each module is discussed below.

- `CADS_Main` is the main program. All input data are collected, and all output are produced, inside the main program.

- `CADS_Abort` is a simple routine that allows aborting a program run in controlled way when necessary, such as if some of the required input files are missing or they contain corrupt data. Calls to the abort routine come either from the main program or from the CADS_Setup routines (see below).

- `CADS_Setup_Cloud` sets up the default configuration for the cloud detection and reads in associated sensor-specific cloud detection namelists if they are provided. The CADS software distribution contains sample cloud detection namelist files (`*_CLDDET.NL`) for all supported sensors. Parameters that can be modified through the cloud detection namelist file are listed in Table II.

- `CADS_Setup_Aerosol` sets up the default configuration for the aerosol detection and reads in sensor-specific aerosol detection namelists if they are provided. Sample aerosol detection namelist files (`*_AERDET.NL`) are provided for all supported sensors. Table III lists the parameters that can be modified through the aerosol detection namelist file.
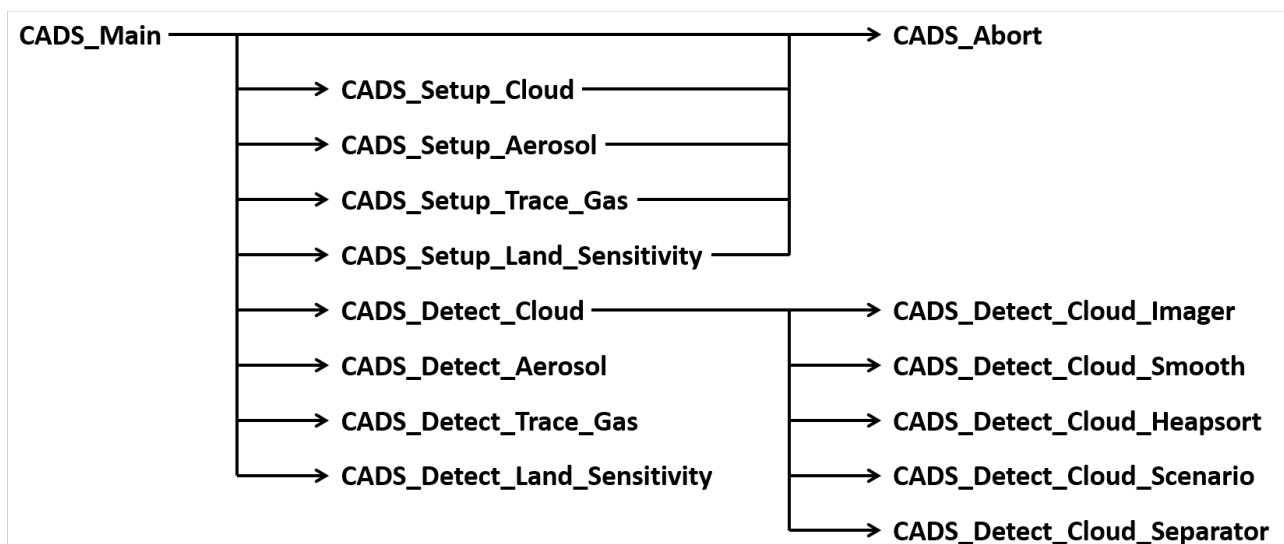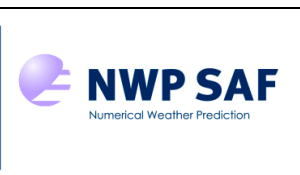

Figure 8: The CADS calling tree.

- `CADS_Setup_Trace_Gas` sets up the default configuration for the trace gas detection and reads in sensor-specific trace gas detection namelists if they are provided. Sample trace gas detection namelist files (`*_TRGASDET.NL`) are provided for all supported sensors. Table IV shows a listing of parameters that can be modified through these namelist files.

- `CADS_Setup_Land_Sensitivity` sets up the default configuration for the land sensitivity detection and reads in the associated namelist files if they are provided. Sample land sensitivity detection namelist files (`*_LANDSENSDET.NL`) are provided for all supported sensors, and the few parameters that are modifiable through these are listed in Table V.

- `CADS_Detect_Cloud` implements the cloud detection algorithm as described in Section 2.1. BT observations, their background-modelled conterparts, channel height assignments (see Appendix A), and auxiliary data from a collocated imager (when available) are taken as input. The algorithm computes departures and splits them in spectral bands, and then proceeds to determining flags to indicate cloud-contaminated channels at each location. Calls to external CADS_Detect_Cloud_* subroutines (see below) are made during the processing.

- `CADS_Detect_Cloud_Imager` determines the preliminary cloud flag if collocated imager data is provided (see Section 2.1.5 for reference).

- `CADS_Detect_Cloud_Heapsort` ranks the input BT departures in vertical according to input channel height assignments. The input to this subroutine consists of one spectral channel band at a time.

- `CADS_Detect_Cloud_Smooth` reduces the effect of random noise in the vertically ranked departure array. A moving average filter is applied, the width of the filter being one of the tuning parameters controlled by the cloud detection namelist.

- `CADS_Detect_Cloud_Scenario` chooses the cloud detection scenario from the input array of vertically ranked and smoothed departure information (as explained in Section 2.1). The preliminary imager-based cloud flag is used as additional input.

- `CADS_Detect_Cloud_Separator` finds the separating point between cloud-affected and unaffected channels in the array consisting of vertically ranked and smoothed departure data.

- `CADS_Detect_Aerosol` implements the aerosol detection as described in Section 2.2. Taking BT observations, channel height assignments, and land fraction as input, the detection algorithm diagnoses the aerosol type and estimates AOD, and eventually produces aerosol contamination flags for the input channels.

- `CADS_Detect_Trace_Gas` implements the trace gas detection as described in Section 2.3. BT observations and their simulated counterparts are taken as input, and channels affected by excessive trace gas concentration are indicated by the output flags.

- `CADS_Detect_Land_Sensitivity` implements the land sensitivity detection as described in Section 2.4. The input to this subroutine consists of land fraction and channel height assignments.

- `CADS_Module` defines parameters and computing structures used during the processing. Specifically, the module introduces four derived variable types to facilitate the detection of cloud, aerosol, trace gas, and land sensitivity.

The user may change the configuration of the detection by changing the values specified in the namelist files placed in directory CADS/namelist/. The parameters included in the namelist take precedence over their default values specified in the source code files. Tables II-V indicate the parameters that can be modified this way.

The namelist files are named using the syntax [SENSOR]_[TYPE]DET.NL, where [SENSOR] is the sensor name (AIRS, CRIS, GIIRS, HIRAS, HIRAS2, IASI, IASING, IKFS2, IRS or IRIS) and [TYPE] indicates the detection software component controlled by the file (either CLD, AER, TRGAS, or LANDSENS to control the detection of cloud, aerosol, trace gas, or land sensitivity, respectively).
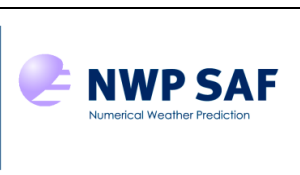
Table II: Cloud detection namelist parameters.

Variable name	Type	Unit	Explanation	Default
M__Sensor	INTEGER	-	Sensor ID number	
N__Num_Bands	INTEGER	-	Number of spectral bands	Sensor-specific
N__Band_Size	INTEGER Array of size [N__Num_Bands]	-	Number of channels in each spectral band	Sensor-specific
N__Bands	2D INTEGER Array of size [N__Num_Bands, MAXVAL(N__Band_Size)]	-	Channels included in each spectral band	Sensor-specific
N__Window_Width	INTEGER Array [N__Num_Bands]	Channels	Moving average smoothing filter width	Sensor-specific
N__Window_Bounds	2D INTEGER Array of size [N__Num_Bands,2]	-	Channel indices at the edges of the IR window	Sensor-specific
N__GradChkInterval	INTEGER Array [N__Num_Bands]	Channels	Number of channels for evaluating the gradient in smoothed BT departure	Sensor-specific
R__BT_Threshold	REAL Array [N__Num_Bands]	K	Maximum (absolute) smoothed BT departure allowed for clear data	Sensor-specific
R__Grad_Threshold	REAL Array [N__Num_Bands]	K	Maximum smoothed BT departure gradient allowed for clear data	Sensor-specific
R__Window_Grad_Threshold	REAL Array [N__Num_Bands]	K	Maximum smoothed BT departure gradient allowed inside the IR window in Quick Exit	0.4
L__Do_Quick_Exit	LOGICAL	-	Switch to enable/disable the Quick Exit scenario.	.TRUE.
L__Do_CrossBand	LOGICAL	-	Switch to enable/disable cross-band detection	.TRUE.
N__BandToUse	INTEGER Array [N__Num_Bands]	-	Spectral band used as reference in the cross-band cloud detection.	Sensor-specific
L__Do_Imager_Cloud_Detection	LOGICAL	-	Switch to enable/disable the use of collocated imager data	.TRUE. on IASI only
N__Num_Imager_Chans	INTEGER	-	Number of collocated imager channels	2*
N__Num_Imager_Clusters	INTEGER	-	Number of collocated imager pixel clusters	7*
N__Imager_Chans	INTEGER Array of size [N__Num_Imager_Chans]	-	Imager channel indices	[2,3]*
R__Stddev_Threshold	REAL Array of size [N__Num_Imager_Chans]	K	Maximum allowed BT standard deviation in the collocated imager data	[0.75;0.80]*
R__Coverage_Threshold	REAL	-	Minimum considered fractional coverage for clusters of imager data	0.03*
R__FG_Departure_Threshold	REAL	K	Maximum allowed BT departure in the collocated imager data	1.0*

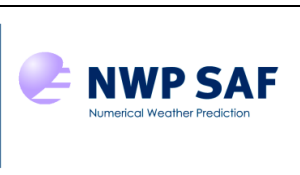* Default values for Imager cloud detection parameters are provided only for IASI.

Table III: Aerosol detection namelist parameters.

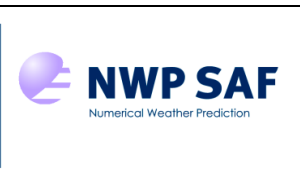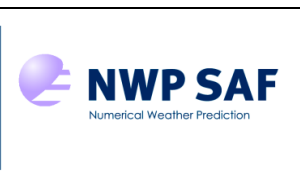| Variable name	Type	Unit	Explanation	Default
M__Sensor	INTEGER	-	Sensor ID number	
N__Num_Aerosol_Tests	INTEGER	-	Number of aerosol tests	3
N__Num_Aerosol_Chans	INTEGER Array of size [N__Num_Aerosol_Tests]	-	Number of channels used in each test	[4, 4, 4]
N__Aerosol_Chans	2D INTEGER Array of size [N__Num_Aerosol_Tests, MAXVAL(N__Num_Aerosol_Chans)]	-	Channel lists for each aerosol test	Sensor-specific
N__Mean_Aerosol_Chans	INTEGER	Channels	Smoothing interval width	Sensor-specific
R__Aerosol_TBD	2D REAL Array of size [N__Num_Aerosol_Tests, MAXVAL(N__Num_Aerosol_Chans)]	K	BT difference threshold values	Sensor-specific
N__Num_Regression	INTEGER Array of size [N__Num_Aerosol_Tests]	-	Number of regression coefficients in the AOD calculation	[3; 3; 3]
R__coef_AOD	2D REAL Array of size [N__Num_Aerosol_Tests, MAXVAL(N__Num_Regression)]	-	Coefficients for converting from spectral BT difference to AOD	Sensor-specific
R__Rank_Thres_Coeff	REAL Array of size 3	-	Coefficients for converting from AOD to a rejection threshold for Saharan dust	[-0.01;2.1;-3.9]
R__Unclassified_Thres	REAL	-	Rejection threshold for normalized channel height with unclassified aerosol	0.4
R__Land_Fraction_Thres	REAL	-	Threshold for land fraction at observation location.	0.5

Table IV: Trace gas detection namelist parameters.

Variable name	Type	Unit	Explanation	Default
M__Sensor	INTEGER	-	Sensor ID number	
N__Num_Trace_Gas_Checks	INTEGER	-	The number of trace gas checks to perform	1
N__Num_Tracer_Channels	INTEGER Array of size [N__Num_Trace_Gas_Checks]	-	The number of tracer channels	Sensor-specific
N__Tracer_Channels	2D INTEGER Array of size [N__Num_Trace_Gas_Checks, MAXVAL(N__Num_Tracer_Channels)]	-	Channel lists containing the tracer channels	Sensor-specific
N__Num_Control_Channels	INTEGER Array of size [N__Num_Trace_Gas_Checks]	-	The number of control channels	Sensor-specific
N__Control_Channels	2D INTEGER Array of size [N__Num_Trace_Gas_Checks, MAXVAL(N__Num_Tracer_Channels)]	-	Channel lists containing the control channels	Sensor-specific
N__Num_Flagged_Channels	INTEGER Array of size [N__Num_Trace_Gas_Checks]	-	The number of potentially affected channels	Sensor-specific
N__Flagged_Channels	2D INTEGER Array of size [N__Num_Trace_Gas_Checks, MAXVAL(N__Num_Tracer_Channels)]	-	Channel lists containing the potentially affected channels	Sensor-specific
R__D_Obs_Threshold	REAL Array of size [N__Num_Trace_Gas_Checks]	K	Thresholds for the observed tracer minus control difference	Sensor-specific
R__D_Dep_Threshold	REAL Array of size [N__Num_Trace_Gas_Checks]	K	Thresholds for the tracer minus control departure difference	Sensor-specific

Table V: Land sensitivity detection namelist parameters.

Variable name	Type	Unit	Explanation	Default
M__Sensor	INTEGER	-	Sensor ID number	
R__Land_Fraction_Thres	REAL	-	Threshold for land fraction at observation location (0=0% land; 1.0=100% land)	0.5
R__Level_Thres	REAL	-	Threshold for normalized channel height assignment (0.0=top of the model vertical grid; 1.0=lowest available channel)	0.9

4. COMPILING AND RUNNING THE CODE

The CADS software is written in `Fortran 90` and it is compiled by using a makefile that is provided as part of the software distribution. Typing `make` in the source directory `CADS/src` will compile the code and produce the executable `CADS` in the parent directory `../CADS`. It should be noted however that, depending on the local computing environment and `Fortran 90` compiler settings, the user may need to make slight modifications to the makefile.

The software expects to be provided with a single input file `cads_input.dat` that is written in `ASCII` format. Depending on whether collocated imager information is provided or not, the input file must be structured in one of two alternative described in sections 4.1 and 4.2.

A sample shell script and test datasets are provided to allow users to verify the correct performance on their local platform. These are placed at directory `CADS/Test_Runs/`. The test script is run from command line using the syntax

```
>> ./CADS_Test_Script.sh SENSOR
```

where `SENSOR` is either AIRS, CRIS, GIIRS, HIRAS, HIRAS2, IASI, IASING, IKFS2, IRS or IRIS.

4.1 Input format without collocated imager data

Header information

- Sensor number
- Number of IR sounder channels provided in the file
- List of the IR sounder channels
- Number of observations included in the file

After the header, the following information are given for each observation:

- Longitude and latitude (in degrees), land fraction, tropopause and boundary layer heights (in the same coordinate system as channel height assignments are given), and a sequential observation index
- BT observation on each channel
- BT simulation (e.g. from NWP model) on each channel
- Channel height assignments (these need to be specified such that higher-sensitive channels have smaller values than lower-sensitive ones)

4.2 Input format with collocated imager data included

Header information

- Sensor number
- Number of IR sounder channels provided in the file
- List of the IR sounder channels
- Number of observations included in the file
- Number of collocated imager channels provided in the file
- List of the imager channels
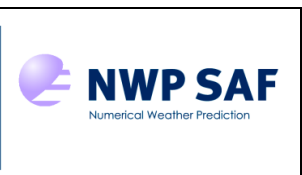- Number of collocated imager clusters

After the header, the following information are given for each observation:

- Longitude and latitude (in degrees), land fraction, tropopause and boundary layer heights (in the same coordinate system as channel height assignments are given), and a sequential observation index
- BT observation on each channel
- BT simulation (e.g. from NWP model) on each channel
- Channel height assignments (these need to be specified such that higher-sensitive channels have smaller values than lower-sensitive ones)
- Fractional coverage of each cluster
- Mean observed BT in each cluster and on each imager channel
- Overall imager BT standard deviation (on all imager channels) inside the IR sounder FOV
- BT simulation (e.g. from NWP model) on each imager channel

4.3 Output format

The software produces an output `ASCII` file that contains following information for each input FOV:

- Longitude, latitude, and the sequential number of the observation
- Cloud flags (separately for each channel): 0=clear; 1=cloud-affected
- The diagnosed aerosol type index: 0=No aerosol; 1=Saharan dust aerosol; 2=volcanic ash aerosol; 3=Unclassified aerosol; 4=Any aerosol over land
- Aerosol flags: 0=clear; 1=aerosol-affected
- Trace gas flags: 0=clear; 1=trace-gas-affected
- Land sensitivity flags: 0=no land sensitivity; 1=channel sensitive to land

5 MAKING CHANGES TO THE SOFTWARE

As far as those IR sounders supported in the original code are concerned, the easiest way to make changes to the processing is through modifying parameter values in the namelist files (see Tables II-V for reference). Alternatively, the user may modify the parameter values at the source code level, but with such approach special care must be taken to prevent the new values getting overridden by values set in the namelist file. When two different values are specified for the same parameter in the source code on the one hand and in the namelist file on the other, the latter gets precedence and is picked up by the software.

5.1 Introducing a new IR sounder

It is possible to introduce new IR sounders to the processing, but this requires substantial source code changes. Changes are needed in the following source code files:

- `CADS_Module.F90`: Somewhere near the existing lines that specify sounder numbers for the supported IR sensors, an extra line needs to be inserted with the form

  `INTEGER(KIND=4), PARAMETER :: INST_ID_[SENSOR] = [NN]`

  where [SENSOR] is a user-specified name of the new sounder and [NN] is a user-specified sensor number. Note that the sensor number must not conflict with any of the previously existing sensor numbers. If the assigned sensor number is smaller than 11 or larger than 98, either `JP__MIN_SENSOR_INDEX` or `JP__MAX_SENSOR_INDEX` must be updated accordingly.
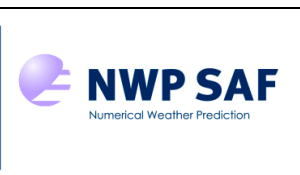
- `CADS_Setup_Aerosol.F90`: Inside the `SELECT CASE` statement that is opened at line 97 (and closed at line 349), a new `CASE` must be introduced with the form

  `CASE(INST_ID_[SENSOR])`

  and this must be followed by specification of default values for all those variables that are specified for previously existing sensors.

- `CADS_Setup_Cloud.F90`, `CADS_Setup_Land_Sensitivity.F90`, and `CADS_Setup_Trace_Gas.F90`: The `SELECT CASE` statements opened at lines 132, 83, and 96, respectively, must be augmented by new `CASE` definitions corresponding to `INST_ID_[SENSOR]`, and these must include new default values specified for all those variables that are already specified for previously existing sensors.

It should be noted that specification of meaningful default values for several variables listed in the four different `CADS_Setup_*` routines will require a considerable amount of experimenting with a test dataset collected from the new IR sounder. It will be helpful if

user can produce a reference dataset that consists of output flags from a supported IR sounder (i.e., AIRS, CrIS, HIRAS, HIRAS-2, GIIRS, IASI, IASI-NG, IKFS-2, IRS or IRIS). If possible, the reference dataset should be made representative of the same geographical area and time period as the test dataset from the new IR sounder. As a first approximation, the parameter values should be adjusted such that a crude correspondence is obtained for output flags for the new IR sounder against the reference dataset. Finer tuning to find the most appropriate tuning parameter values will likely require substantial experimentation covering, ideally covering several weeks or even months in time.

REFERENCES

Poli, P. and Brunel, P., 2018. Assessing reanalysis quality with early sounders Nimbus-4 IRIS (1970) and Nimbus-6 HIRS (1975). *Advances in Space Research* **62**(2), 245—264.

Clarisse, L., F. Prata, J.-L. Lacour, D. Hurtmans, C. Clerbaux, and P.-F. Coheur, 2010. A correlation method for volcanic ash detection using hyperspectral infrared measurements. *Geophys. Res. Lett.*, **37**, L19806, doi:10.1029/2010GL044828.

Eresmaa, R., 2014. Imager-assisted cloud detection for assimilation of Infrared Atmospheric Sounding Interferometer radiances. *Quarterly Journal of the Royal Meteorological Society*, **140**, 2342-2352, DOI:10.1002/qj.2304.

Eresmaa, R., J. Letertre-Danczak, and T. McNally, 2020. Screening routines for aerosol- and trace-gas-affected infrared radiances. *Proceedings of the 22nd International TOVS Study Conference, Saint-Sauveur, Canada.* 9 p.

Letertre-Danczak, J., 2016. The use of geostationary radiance observations at ECMWF and aerosol detection for hyper-spectral infrared sounders: 1st and 2nd years report. *EUMETSAT/ECMWF Fellowship Programme Research Report No. 40,* 18 p.

McNally, A.P. and P.D.Watts, 2003. A cloud detection algorithm for high-spectral-resolution infrared sounders. *Quarterly Journal of the Royal Meteorological Society*, **129**, 3411-3423.

Vidot, J., 2017: CADS Version 2.3 Aerosol Validation. *NWP SAF Report NWPSAF-MF-TV-001.* 9 p.

APPENDIX A. CHANNEL HEIGHT ASSIGNMENTS FROM RTTOV-12

The CADS software makes use of channel height information in the detection algorithms for cloud contamination, aerosol contamination, and land sensitivity. The channel height assignments indicate the level of a hypothetical cloud top that would alter the radiance measurement by a fixed percentage from the clear-sky radiance value. The channel height assignments can be expressed in terms of pressure or model level index, but the software assumes that lower numbers mean channels ranked higher in vertical. The channel height assignments, that are used in the test datasets included in CADS V3.1, correspond to the 137-level vertical coordinate system used at ECMWF.

In the ECMWF implementation of the cloud detection algorithm, channel heights are assigned using the information computed and saved in RTTOV-12 under the type definition `rttov_radiance`. Variables of this type contain calculated radiances and brightness temperatures for a variety of assumptions. Specifically, array `rttov_radiance%overcast` contains channel-specific radiances corresponding to the assumed presence of an opaque cloud top at each of the RTTOV model levels. Radiances corresponding to clear (cloud-free) conditions are stored in array `rttov_radiance%clear`. Height assigned to each channel matches the model level index at which the difference between the overcast and clear radiances is exactly 1% of the clear radiance. This is expressed as a real number and it is interpolated from the nearest two levels where the difference is just above and just below the desired 1% threshold.

Note that the height assignment is computed in radiance (rather than brightness temperature) space as this is the unit the `rttov_radiance%overcast` array uses. In typical atmospheric conditions, a 1% change in radiance corresponds to approximately 0.5K on wavelengths greater than 10μm but decreases to around 0.2K at 3μm.