ESA Contract Report

PEARL Cloud contract 4000128669/19/NL/CT

Contract Report to the European Space Agency

Assimilation system adaptation and maintenance for cloud radar and lidar observations

WP-1000 report

Preparations for EarthCARE Assimilation - Radar and Lidar Cloud Observations (PEARL Cloud)

Authors: Marta Janisková and Mark Fielding

Contract officer: Tobias Wehr

December 2022

Series: ECMWF ESA Contract Report Series

A full list of ECMWF Publications can be found on our web site under:
http://www.ecmwf.int/en/publications/

Contact: library@ecmwf.int

European Centre for Medium Range Weather Forecasts, Shinfield Park, Reading, RG2 9AX, UK

The information within this publication is given in good faith and considered to be true, but ECMWF accepts no liability for error or omission or for loss or damage arising from its use.

ABSTRACT

A key objective of this work package is to ensure that the system developments for cloud radar and lidar observations are maintained along with the evolving operational model at the European Centre for Medium-Range Weather Forecasts (ECMWF).

At the first step of this on-going effort, the previous assimilation system developments for these observations were adapted to the Integrated Forecasting System (IFS) model cycle, CY46R1, the cycle used at the beginning of this project in October 2019. After successful moving of those developments to the model cycle CY46R1 and based on experimentation done using this cycle, some adjustments and minor corrections have been required. The porting, debugging and testing required in order to make the updated system with cloud radar and lidar observations functional is described.

In the next step, the assimilation system was ported to the higher IFS model cycle, CY47R1, as part of preparations to include these developments in CY48R1 in order to appear passively in the operational code when releasing this cycle in 2022. Basic testing of the developments in CY47R1 has been done. That included a comparison of the analysis increments between cycles CY46R1 and CY47R1, as well as checking the bit reproducibility of the system when cloud radar and lidar observations are switched off (i.e. to ensure that the passive observations do not modify any integration of the operational code).

During the project, the system has been updated to use, on top of already included cloud radar reflectivity and lidar backscatter, for other observations such as cloud Doppler velocity, lidar extinction and lidar Rayleigh backscatter. The technical developments and modifications for that are also described in this report.

Finally, summary of modifications and corrections to the assimilation system for cloud radar and lidar observation to be able to use it in the most recent IFS model cycle, CY48R1, is provided. These were required because of: (i) starting to use the model cycle running on new computer at ECMWF'S new Data Centre in Bologna, (ii) running IFS Four-Dimensional Variational (4D-Var) data assimilation under the framework of the Object-Oriented Prediction System (OOPS) and (iii) corrections and modifications as outcome of extensive testing for cloud radar and lidar observations done when using still the cycle CY46R1. Basic testing of the system in CY48R1 is also discussed.

Contents

1 Introduction

Earth, Clouds, Aerosols and Radiation Explorer (EarthCARE) mission was conceived with the goal of advancing the understanding of the interaction between clouds and aerosols with Earth's radiation budget through the synergistic use of an on-board radar, lidar and a multi-spectral imager. However, as documented by a number of previous studies (Janisková et al., 2010, 2014; Janisková and Fielding, 2018), the detailed measurements of clouds provided by these instruments could also be beneficial to producing weather forecasts by helping to constrain the initial atmospheric state required in Numerical Weather Prediction (NWP).

The previous assimilation project at European Centre for Medium-Range Weather Forecasts (ECMWF) (Operational Assimilation of Space-borne Radar and Lidar Cloud Profile Observations for Numerical Weather Prediction, Janisková and Fielding, 2018) focused on developments towards direct assimilation and monitoring to exploit cloud radar and lidar data for their assimilation in NWP models. The project demonstrated the feasibility of Four-Dimensional Variational (4D-Var) assimilation of such observations using CloudSat (Stephens et al., 2002) and CALIPSO (Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations Winker et al., 2009) data and very limited use of simulated EarthCARE data. The successful real-time assimilation of EarthCARE observations relies on a number of further tasks, which are part of this project.

Crucial to ensure that the previous developments for assimilation of cloud radar and lidar observations are not lost, an on-going effort is required to align developments to the system related to these observations with operational model upgrades and other modifications to the data assimilation system. Without these updates, existing programming code can quickly become obsolete to the point that is unusable and in need of complete re-writes. Ensuring that the assimilation system is maintained along with the evolving ECMWF operational model is therefore a key objective.

The first step of this process was to update data assimilation system developed in the previous project (Janisková and Fielding, 2018) funded by ESA to the Integrated Forecasting Cycle (IFS) model cycle CY46R1, which was the cycle used at the beginning of the project (October 2019) when starting migration. That represented very important step since it secured that all past development for the future. Without this, operational use of cloud radar and lidar observations could not be imagined and we would risk that hard work over many years would be lost. After successful moving of those developments to the model cycle CY46R1, some adjustments and minor corrections have been required based on experimentation done using the model cycle CY46R1.

In the next, the assimilation system was ported to the higher IFS model cycle, CY47R1, as part of preparations to include these developments in CY48R1 in order to appear passively in the operational code when releasing this cycle in 2022.

During the project, the extensive developments to the operational data assimilation at ECMWF have been done in preparation for the monitoring of EarthCARE-like observations other than cloud radar reflectivity and lidar backscatter. Specifically, observation operators for radar Doppler velocity, cloud extinction and Rayleigh backscatter have been included into the IFS, so that observations of these types can be monitored against the ECMWF model. This required the technical developments and modifications to the cloud radar and lidar assimilation system in order to account for the above additional observations in the IFS system, which are also reported in this part of the report.

Finally, modifications and corrections to the assimilation system for cloud radar and lidar assimilation in order to be able to use it in the most recent IFS cycle, CY48R1, officially released towards the end of October 2022, are summarized. This is the cycle which might still be ECMWF's operational cycle at the time of satellite launch. Even if the higher cycle is already operational at that time, majority of experimentations and adjustments for the EarthCARE type of observations as a part of the pre-operational preparations will be carried out using the cycle CY48R1.

Section 2 provides a brief description of 4D-Var system to be used for monitoring and assimilation of cloud radar and lidar observations. A summary of the work required to be done in order to migrate the assimilation system development for cloud radar and lidar to the IFS model cycle CY46R1 together is provided in Section 3 together with testing and debugging needed in order to make updated system with cloud radar and lidar observations functional. Section 4 summarizes the migration and testing process required to bring the assimilation system development for cloud radar and lidar observations from the model cycle CY46R1 to CY47R1. Section 5 describes work done to include additional observations, such as radar Doppler velocity, cloud extinction and Rayleigh backscatter, into the system and to check that they are properly seen by the system. Section 6 describes modifications and corrections to the assimilation system for cloud radar and lidar assimilation in order to be able to use it in the most recent IFS cycle, CY48R1, officially released towards the end of October 2022. It provides a summary of the work done to bring the cloud radar and lidar assimilation system to the up-to-date version based on testing performed still using the cycle CY46R1 and the work required because of running IFS 4D-Var data assimilation under the framework of the Object-Oriented Prediction System (OOPS) from the cycle CY48R1. A brief summary of the whole work package is provided in Section 7.

2 4D-Var system used for cloud radar and lidar observations

2.1 A brief description of 4D-Var system

At ECMWF, the current operational data assimilation system is an incremental version of 4D-Var (Courtier et al., 1994; Rabier et al., 2000). The principle of 4D-Var is to search for an optimal balance between observations and the model by finding a model trajectory $\mathbf{x}(t)$ that is closer, in a least-square sense, to the observations available during a given time period $[t_0, t_n]$. The model trajectory $\mathbf{x}(t)$ is completely defined by the initial state $\mathbf{x}_0$.

The mis-match to a given set observations $\mathbf{y}^o$ and to an *a-priori* model state $\mathbf{x}^b$ called the background (usually provided by a short-range forecast), is measured by an objective (cost) function. The cost function effectively penalises both differences between the state of the model $\mathbf{x}_0$ and the background $\mathbf{x}_0^b$, and differences between the observations and model-equivalent observations. An additional constraint to the cost function $\mathscr{J}^c$ is used in 4D-Var to control fast gravity waves using the digital filter approach developed by Gauthier and Thépaut (2001).

Using the incremental approach, 4D-Var can be approximated to the first order by finding the analysis increment $\delta\mathbf{x}_0$ at initial time t_0 which minimizes the following cost function $\mathscr{J}$:

$$\mathscr{J}(\delta\mathbf{x}_0) = \underbrace{\frac{1}{2}(\delta\mathbf{x}_0)^T\mathbf{B}^{-1}(\delta\mathbf{x}_0)}_{\mathscr{J}^b} + \underbrace{\frac{1}{2}\sum_{i=0}^{n}\left(H_i'\delta\mathbf{x}_i - \mathbf{d}_i\right)^T\mathbf{R}_i^{-1}\left(H_i'\delta\mathbf{x}_i - \mathbf{d}_i\right)}_{\mathscr{J}^o} + \mathscr{J}^c \tag{2.1}$$

where at any time t_i,

- $\mathbf{R}_i$ is the observation error covariance matrix (including measurement and representativeness errors);
- $\mathbf{B}$ is the background error covariance matrix of the state $\mathbf{x}^b$ and is based on a wavelet formulation (Fisher, 2004) to introduce regime-dependent error statistics.
- H_i' is the linearized observation operator providing the model equivalent to the observations and it also includes the spatial interpolations to observation locations as well as the propagation of the initial state to each observation time using the forecast model;
- $\delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_i^b$ is the analysis increment and represents the departure of the model state ($\mathbf{x}$) with respect to the background ($\mathbf{x}^b$) which consists of temperature, humidity, vorticity, divergence and surface pressure in the current 4D-Var system;
- $\mathbf{d}_i = \mathbf{y}_i^o - H_i(\mathbf{x}^b{}_i)$ is the so-called innovation vector providing the departure of the model background equivalent from the observation ($\mathbf{y}_i^o$);

The incremental approach is a cost effective formulation of 4D-Var since it allows to use a lower-resolution model (with its adjoint and tangent linear versions) for the iterative and relatively costly computation of analysis increments. A high resolution version of the model is only used for the computation of the model trajectory, and for calculating the departures between observations and model. At ECMWF, the lower-resolution iterations (the inner-loops) are nested within a set of outer-loop iterations at full resolution. The inner-loop resolution is increased with each iteration of the outer-loop using 'multi-resolution' extension to the incremental method (Veerse and Thépaut, 1998). The cost of the inner-loops depends not only on the resolution, but also on the complexity of the inner-loop model, e.g. the use of simpler or more complete representations of the physical processes (Janisková and Lopez, 2013).

2.2 Data flow in 4D-Var system

Figure 2.1 illustrates the flow of data in 4D-Var system when using cloud radar and lidar observations. Starting from the model background represented by temperature, humidity, vorticity, divergence and surface pressure and using the linearized model dynamics and “dry” physical parametrization represented by vertical diffusion, radiation and orographic and non-orographic gravity wave drag, updated fields of temperature, humidity and wind components together with their tendencies are obtained. The linearized cloud scheme with input from the convection scheme provides then the perturbations in cloud liquid and ice water content which are passed to the linearized observation operator for cloud radar reflectivity and lidar backscatter. Using information provided by observations and their model equivalents, the observation cost function $\mathscr{J}^o$ is constructed. To minimize the cost function, the gradient of the cost function needs to be computed. In the backward calculation, the gradient of the cost function with respect to the control variables is calculated using first the adjoint of of observation operator routine for cloud radar reflectivity and lidar backscatter to obtain the gradient with respect to the cloud liquid and ice water contents. The latter is then passed to the adjoint of the cloud and convection schemes and used to compute the cloud contribution to the gradient with respect to temperature and specific humidity. Through the adjoint of dry physical parametrization and model dynamics, also the gradient with respect to wind is computed together with updating the gradient with respect to temperature and specific humidity. The final gradient of the observation cost function with respect to the model state variables is transformed to the control–vector variables and passed together with the gradient of the background cost function to the minimization algorithm. Minimization provides the analysis increments $\delta\mathbf{x}_0^a$ to be added to the background $\mathbf{x}_0^b$ in order to obtain the model analysis.

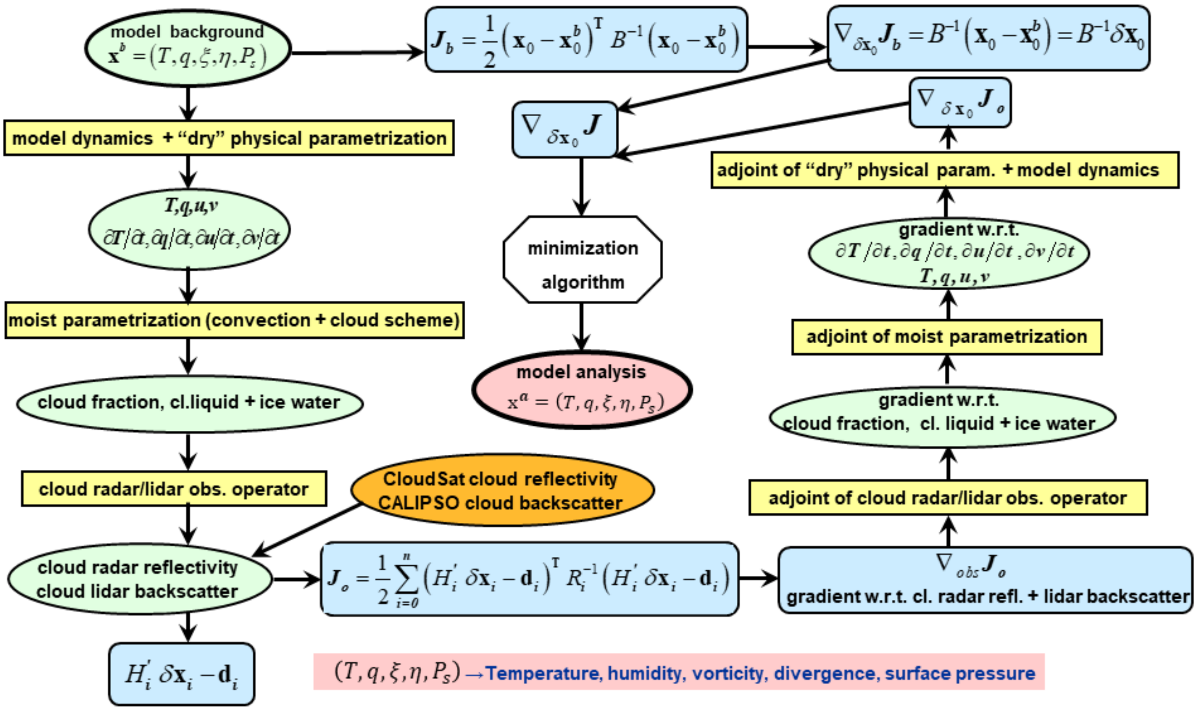

model background
$\mathbf{x}^b = (T, q, \xi, \eta, P_s)$
$J_b = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T B^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b)$
$\nabla_{\delta\mathbf{x}_0} J_b = B^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) = B^{-1}\delta\mathbf{x}_0$
$\nabla_{\delta\mathbf{x}_0} J_o$
$\nabla_{\delta\mathbf{x}_0} J$
model dynamics + “dry” physical parametrization
adjoint of “dry” physical param. + model dynamics
T, q, u, v
$\partial T/\partial t, \partial q/\partial t, \partial u/\partial t, \partial v/\partial t$
minimization algorithm
gradient w.r.t.
$\partial T/\partial t, \partial q/\partial t, \partial u/\partial t, \partial v/\partial t$
T, q, u, v
moist parametrization (convection + cloud scheme)
adjoint of moist parametrization
model analysis
$\mathbf{x}^a = (T, q, \xi, \eta, P_s)$
cloud fraction, cl.liquid + ice water
gradient w.r.t.
cloud fraction, cl. liquid + ice water
cloud radar/lidar obs. operator
CloudSat cloud reflectivity
CALIPSO cloud backscatter
adjoint of cloud radar/lidar obs. operator
cloud radar reflectivity
cloud lidar backscatter
$J_o = \frac{1}{2}\sum_{i=0}^{n}(H_i' \delta\mathbf{x}_i - \mathbf{d}_i)^T R_i^{-1}(H_i' \delta\mathbf{x}_i - \mathbf{d}_i)$
$\nabla_{obs} J_o$
gradient w.r.t. cl. radar refl. + lidar backscatter
$H_i' \delta\mathbf{x}_i - \mathbf{d}_i$
$(T, q, \xi, \eta, P_s) \rightarrow$ Temperature, humidity, vorticity, divergence, surface pressure

Figure 2.1: Diagram displaying data flow in 4D-Var system from the model background to the model analysis when using cloud radar and lidar observations.

As it can be seen from the flow diagram of Fig. 2.1, the control variables in the 4D–Var are temperature, humidity, vorticity, divergence and surface pressure. The cloud gradients computed by the adjoint of the moist parametrizations contribute directly to the gradients in temperature and humidity, and indirectly to the gradients in the other control variables, i.e. wind, through the coupling which takes place in the 4D–Var. At the moment, the cloud variables are not used as the control variables of 4D-Var system used at ECMWF. As a consequence the cloud–related observations can only impact the model indirectly as described above. Figure 2.2 then shows how the departures of the model equivalent from cloud radar and lidar observations are propagated by 4D-Var system in the steps displayed in Fig. 2.1 to analysis increments of the control variables.

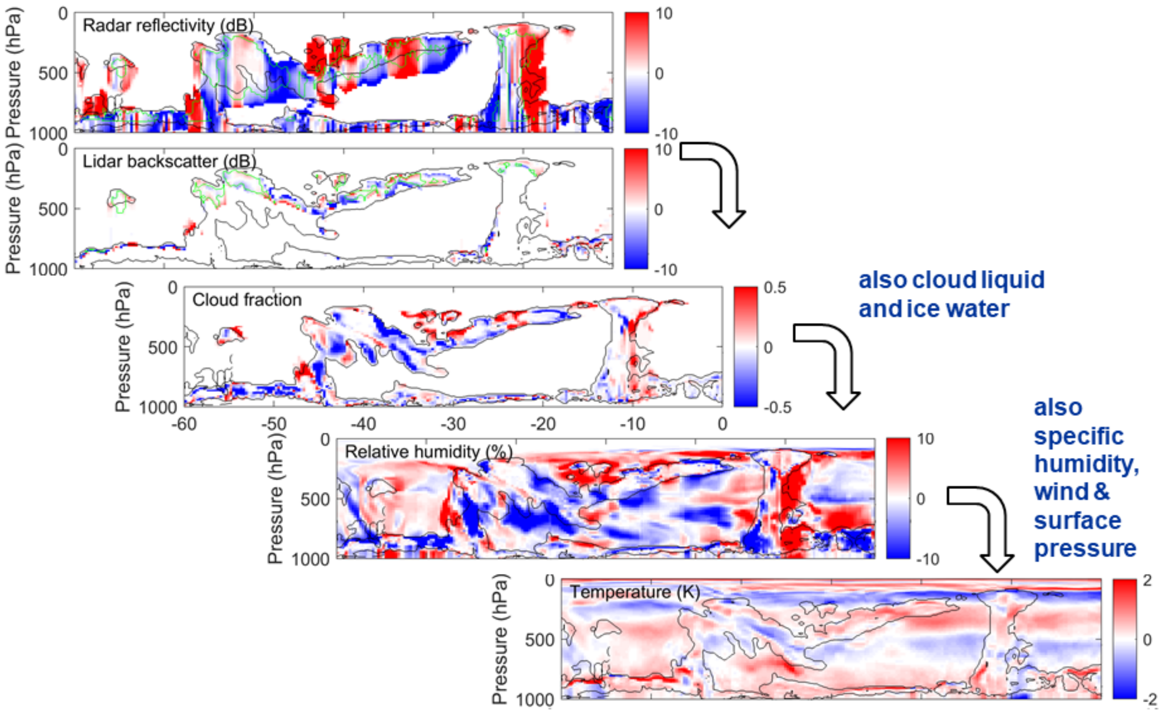


Figure 2.2: Diagram displaying on sample of crossections how increments (the departures of the model equivalent from observations) from cloud radar reflectivity and lidar backscatter are propagated by 4D-Var system to analysis increments of the control variables through the steps displayed in Fig. 2.1.

2.3 Observation pre-processing and handling

For any observational data to be assimilated in the ECMWF operational system it must first be converted to Binary Universal Form (BUFR). A framework for converting radar and lidar data to BUFR was established in Fielding and Janisková (2017), but requires modifications to account for the additional observation types considered in this project. The conversion from the data's source format to BUFR is designed to be as close to a one-to-one mapping as possible and any differences in the context of the data should be limited to initial quality checks and reductions in precision to limit file sizes. In addition to developments related to BUFR, updates to the Observation Data Base (ODB) are also required. Whereas BUFR is optimised for the efficient storage of data, the role of the ODB is to provide fast I/O to the assimilation system on all observation related data. In this section, a brief overview of BUFR and ODB formats is provided before the new developments for additional observations are presented.

2.3.1 BUFR definitions

BUFR is a World Meteorological Organization (WMO) standard for transmitting and storing observations of all kinds of meteorological data. Its flexibility lies with its use of 'data descriptors', which are used to access data values. Metadata is stored in external table files. Variables stored within BUFR must be selected from a finite pre-existing list of observation types. New variable types must be approved by WMO. The list of data descriptors for each observation type is known as a 'BUFR sequence'. Many of the descriptors are generic, such as the time and geolocation descriptors, however the more specific descriptors, such as different radar and lidar descriptors, will need to be approved by WMO for operational use.

Each BUFR descriptor has a unique 6 digit code in the format FXXYYY, where F determines the type of descriptor, X determines the class of the descriptor and Y determines the name of the descriptor within its class. There are four types of descriptors: the most common are element descriptors (F=0), which are used to convey either meta-data or numerical data. The other types of descriptors are used to manipulate either data or the BUFR sequence itself. Replication descriptors (F=1) create loop-like structures in the BUFR template to allow descriptors to be repeated without the need for repeating the element descriptor explicitly. If Y=0, the replication descriptors are "delayed" and the number of replications needs to be provided by a subsequent elemental descriptor. Operator descriptors (F=2) perform actions on elemental descriptors, such as adding additional bits. Finally sequence descriptors (F=3) correspond to sequences of element descriptors that are often repeated. Sequence descriptors are not necessary, but can significantly reduce the size of the BUFR file if used wisely.

Table 2.1 shows the BUFR sequence created for space-borne radar observations with new descriptors for Doppler velocity and Doppler velocity uncertainty included on top of descriptors for Cloud radar reflectivity, Cloud radar uncertainty already defined by the previous assimilation project (Janisková and Fielding, 2018). In addition to their code, each descriptor has a *scale*, *reference* and *width* value that describes how the data is encoded to binary and vica versa. The *scale* determines the precision of the stored data, while the *width* and *reference* determine the range of the data. Specifically, the data is encoded by allocating a range of integers, **I**, from 1 to 2^{width} to data values, **R**, so that:

$$\mathbf{R} = (\mathbf{I} + reference)10^{-scale} \tag{2.2}$$

The corresponding BUFR sequence for space-borne HSRL observations is shown in Table 2.2. There are many similarities to the cloud radar BUFR sequence, with the main differences being the observation variables, such as lidar backscatter. For cloud extinction, we re-use the 'Extinction coefficient' descriptor originally included for aerosol observations. New descriptors include the Rayleigh backscatter and Mie co-polar backscatter.

2.3.2 ODB definitions

While BUFR is extremely efficient for storing data, a different format with fast I/O is required for operational data assimilation. At ECMWF, the ODB is 'in-house' data storage software to allow the 4D-Var system within IFS to store and access data. ODB is formulated on fast and efficient Structured Query Language (SQL) to define and retrieve observational data. There is no unique centralized ODB database: a new ODB is created each analysis time a 4D-Var analysis is made. Each ODB is stored locally in the ECFS (ECMWF's File Storage System) for post-processing and evaluation. Any new ODB variable name codes must be approved by the 'ODB Governance' to ensure consistency between ECMWF,

F-X-Y	Description	Scale	Ref.	Width	Units	Comment
0 01 007	Satellite identifier	0	0	10		satID=TBD
0 02 019	Satellite instruments	0	0	11		instrumentID=TBD
3 01 011	Year, month, day					‘profileTime’
3 01 013	Hour, minute, second					‘profileTime’
3 01 021	Latitude / Longitude (high accuracy)					‘longitude’ and ‘latitude’
0 10 033	Altitude (Platform to Ellipsoid)	1	0	27	m	
0 02 153	Satellite Channel Centre Frequency	-8	0	26	Hz	94 GHz
0 25 182	L1 processing flag					
0 25 181	L2 processing flag					
0 21 197	Height	0	-1000	17	m	
0 21 192	Cloud radar reflectivity	2	-9000	15	dBZ	‘radarReflectivityFactor’
0 21 193	Cloud radar reflectivity uncertainty	2	0	9	dB	see text
0 21 198	Doppler velocity	2	-20	11	m s^{-1}	
0 21 199	Doppler velocity uncertainty	2	-20	11	m s^{-1}	
0 21 194	Data classification type	0	0	4	CODE TABLE	0 Surface 1 Cloud likely 2 Cloud probable 3 Cloud possible 4 Unclassified 15 Missing value
0 33 003	Quality information				CODE TABLE	
0 08 049	Number of observations					
0 21 195	Cloud fraction	3	0	11		

Table 2.1: Proposed BUFR sequence for cloud radar observations.

Code	Description	Scale	Ref.	Width	Units	Comment
0 01 007	Satellite identifier	0	0	10		satID=787
0 02 019	Satellite instruments	0	0	11		instrumentID=303
0 02 153	Satellite Channel wavelength	9	0	16	m	
3 01 011	Year, month, day					
3 01 013	Hour, minute, second					
3 01 021	Latitude / Longitude (high accuracy)					
0 10 033	Altitude (Platform to Ellipsoid)	1	0	27	m	
0 25 182	L1 processing flag					
0 25 181	L2 processing flag					
0 21 197	Height	0	-1000	17	m	
0 21 206	Total attenuated backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 21 207	Uncertainty in total attenuated backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 15 067	Extinction coefficient	9	0	30	m^{-1}	Range: 0 to 0.1 m^{-1}
0 15 068	Uncertainty in extinction coefficient	9	0	30	m^{-1}	Range: 0 to 0.1 m^{-1}
0 21 204	Rayleigh Attenuated Backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 21 205	Uncertainty in Rayleigh attenuated backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 21 202	Mie Copolar Attenuated Backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 21 203	Uncertainty in Mie Copolar Attenuated Backscatter	2	-9000	15	$m^{-1}\,sr^{-1}$	Range: 0 to 0.1 $m^{-1}\,sr^{-1}$
0 21 194	Data classification type	0	0	4	CODE TABLE	0 Surface 1 Cloud 2 Aerosol 3 Unclassified 15 Missing value
0 33 003	Quality information				CODE TABLE	
0 08 049	Number of observations					
0 21 195	Cloud fraction	3	0	11		

Table 2.2: Proposed BUFR sequence for space-borne HSRL observations.

member states and collaborators.

The ODB contains all the input data that is needed by the data assimilation system. Most data is stored in either a header (hdr) entry or a body entry (see left-hand side of Fig. 2.3). Header data typically consists of geolocation and instrument meta-data, while body data are observations or their meta-data. All body entries are linked to a header entry. For radar and lidar observations, the hdr contains the lat, lon and time the observations were taken. The observation value to be assimilated is stored within the 'obsvalue' variable within the body, alongside information needed for assimilation such as the observation error, screening status and bias correction. To identify the observation, each observation type has a unique variable number, 'varno', which tells the 4D-Var system which observation operator must be used to produce the model-equivalent.

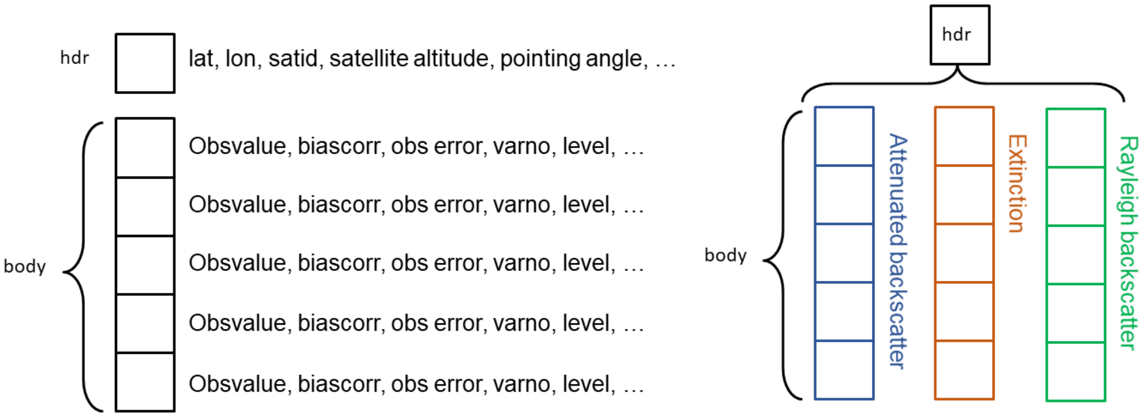

hdr
lat, lon, satid, satellite altitude, pointing angle, …
body
Obsvalue, biascorr, obs error, varno, level, …
Obsvalue, biascorr, obs error, varno, level, …
Obsvalue, biascorr, obs error, varno, level, …
Obsvalue, biascorr, obs error, varno, level, …
Obsvalue, biascorr, obs error, varno, level, …
hdr
body
Attenuated backscatter
Extinction
Rayleigh backscatter

Figure 2.3: Schematic of hdr and body storage within the ODB.

The total observation error is stored in 'final_obs_error', but a breakdown of the error into representativity error, measurement error and forward model error is made possible by the inclusion of 'obs_error' and 'repres_error', which will be useful for the analysis of experiments. Also useful for analysis are the 'standard_deviation','n_obs' and 'cloud_fraction' variables. Note that much of the ODB variable definitions framework for CALIPSO observations (Table 2.4) is identical to that for CloudSat (Table 2.3), the interpretation of some variables differs, such as the 'datum_status' variable, which will contain different flags.

2.3.3 Screening criteria

Preventing observations that may degrade the analysis from being assimilated is an important component of the observation pre-processing system. Known as 'screening', the selection of which observations to enter the minimization is achieved by checking each observation against a set of screening flags. The screening flags are represented as individual bits in an integer variable that is stored in the odb as the 'datum_status' and archived for diagnostic purposes.

Table 2.5 shows the various screening flags and associated bitfields for both radar reflectivity and Doppler velocity observations. The first 9 flags are generic for all observation types, for example if the observation is missing or if the observation is out of expected bounds. The last three are specific to cloud radar observations: the 'low_CF' flag is used when either the model or superob cloud fraction is below a threshold, the 'mscat' flag is used when multiple-scattering is suspected using the criteria in Fielding and Janisková (2017) and the 'FG_low' flag is used when the forward modelled value is below the sensitivity of the instrument. The Doppler velocity sensitivity is obtained by checking the corresponding radar

Varname	Parent	Dimension	Description
obsvalue	body	model_levels (vertco_type 5)	Radar reflectivity averaged to model grid and level
biascorr	body	model_levels (vertco_type=5)	offline bias correction to be applied to radar reflectivity
datum_status	body	model_levels (vertco_type=5)	flag for storing screening/blacklisting/quality control information
clreflvalue	body	model_levels (vertco_type=5)	Model equivalent radar reflectivity
clreflvaluetl	body	model_levels (vertco_type=5)	Tangent linear variable for radar reflectivity
clreflvaluead	body	model_levels (vertco_type=5)	Adjoint variable for radar reflectivity
report_clreflflag	body	model_levels (vertco_type=5)	Flag for storing quality control information related to model equilavent
standard_deviation	superobs	model_levels (vertco_type=5)	Standard deviation of radar reflectivity within model grid and level
n_obs	superobs	model_levels (vertco_type=5)	number of samples used to compute obsvalue and standard_deviation
cloud_fraction	superobs	model_levels (vertco_type=5)	number of cloudy points defined by cloud mask divided by number of samples
repres_error	errstat	model_levels (vertco_type=5)	Flow dependent representativity error
obs_error	errstat	model_levels (vertco_type=5)	Measurement error
final_obs_error	errstat	model_levels (vertco_type=5)	Combination of meaurement, representativity and forward model error
surface_pressure	modsurf	scalar	background surface pressure from model
lat	hdr	scalar	Average latitude of observations
lon	hdr	scalar	Average longitude of observations
stalt	hdr	scalar	Height of satellite above sea level

Table 2.3: Selected ODB variable definitions for CloudSat observations.

Varname	Parent	Dimension	Description
obsvalue	body	model_levels (vertco_type=5)	Lidar backscatter averaged to model grid and level
biascorr	body	model_levels (vertco_type=5)	offline bias correction to be applied to lidar backscatter
datum_status	body	model_levels (vertco_type=5)	flag for storing screening/blacklisting/quality control information
clbscvalue	body	model_levels (vertco_type=5)	Model equivalent lidar backscatter
clbscvaluetl	body	model_levels (vertco_type=5)	Tangent linear variable for lidar backscatter
clbscvaluead	body	model_levels (vertco_type=5)	Adjoint variable for lidar backscatter
report_clbscflag	body	model_levels (vertco_type=5)	Flag for storing quality control information related to model equilavent
standard_deviation	superobs	model_levels (vertco_type=5)	Standard deviation of lidar backscatter within model grid and level
n_obs	superobs	model_levels (vertco_type=5)	number of samples used to compute obsvalue and standard_deviation
cloud_fraction	superobs	model_levels (vertco_type=5)	number of cloudy points defined by cloud mask divided by number of samples
repres_error	errstat	model_levels (vertco_type=5)	Flow dependent representativity error
obs_error	errstat	model_levels (vertco_type=5)	Measurement error
final_obs_error	errstat	model_levels (vertco_type=5)	Combination of meaurement, representativity and forward model error
surface_pressure	modsurf	scalar	background surface pressure from model
lat	hdr	scalar	Average latitude of observations
lon	hdr	scalar	Average longitude of observations
stalt	hdr	scalar	Height of satellite above sea level

Table 2.4: Selected ODB variable definitions for CALIPSO observations.

reflectivity value minimum sensitivity.

Bitfield	**Key**	**Description**
	Radar reflectivity (CLREF)	
0	NRAD_CLREF_ACTIVE	Observation active if no other bits set
1	NRAD_CLREF_OBS	Observed clref out of bounds
2	NRAD_CLREF_NEGATIVE_Q	Negative Q in model profile
3	NRAD_CLREF_NO_OBS	No obs for this grid point (GP-space only)
4	NRAD_CLREF_NO_GRID	No grid point for this obs
5	NRAD_CLREF_PASSIVE	Passive observation
6	NRAD_CLREF_MISSING	Observation value is missing (=RMDI)
7	NRAD_CLREF_LAST_TIMESTEP	Last timestep,when TL/AD gp model doesn't run
8	NRAD_CLREF_FG_DEPARTURE	FG departure outside limit
9	NRAD_CLREF_LOW_CF	Low model or obs cloud fraction
10	NRAD_CLREF_MSCAT	Multiple-scattering
11	NRAD_CLREF_FG_LOW	Low FG value (below sensitivity)
	Radar Doppler velocity (CLDOP)	
0	NRAD_CLDOP_ACTIVE	Observation active if no other bits set
1	NRAD_CLDOP_OBS	Observed cldop out of bounds
2	NRAD_CLDOP_NEGATIVE_Q	Negative Q in model profile
3	NRAD_CLDOP_NO_OBS	No obs for this grid point (GP-space only)
4	NRAD_CLDOP_NO_GRID	No grid point for this obs
5	NRAD_CLDOP_PASSIVE	Passive observation
6	NRAD_CLDOP_MISSING	Observation value is missing (=RMDI)
7	NRAD_CLDOP_LAST_TIMESTEP	Last timestep,when TL/AD gp model doesn't run
8	NRAD_CLDOP_FG_DEPARTURE	FG departure outside limit
9	NRAD_CLDOP_LOW_CF	Low model or obs cloud fraction
10	NRAD_CLDOP_MSCAT	Multiple-scattering
11	NRAD_CLDOP_FG_LOW	Low FG value (below sensitivity)

Table 2.5: Screening flags for cloud radar observations.

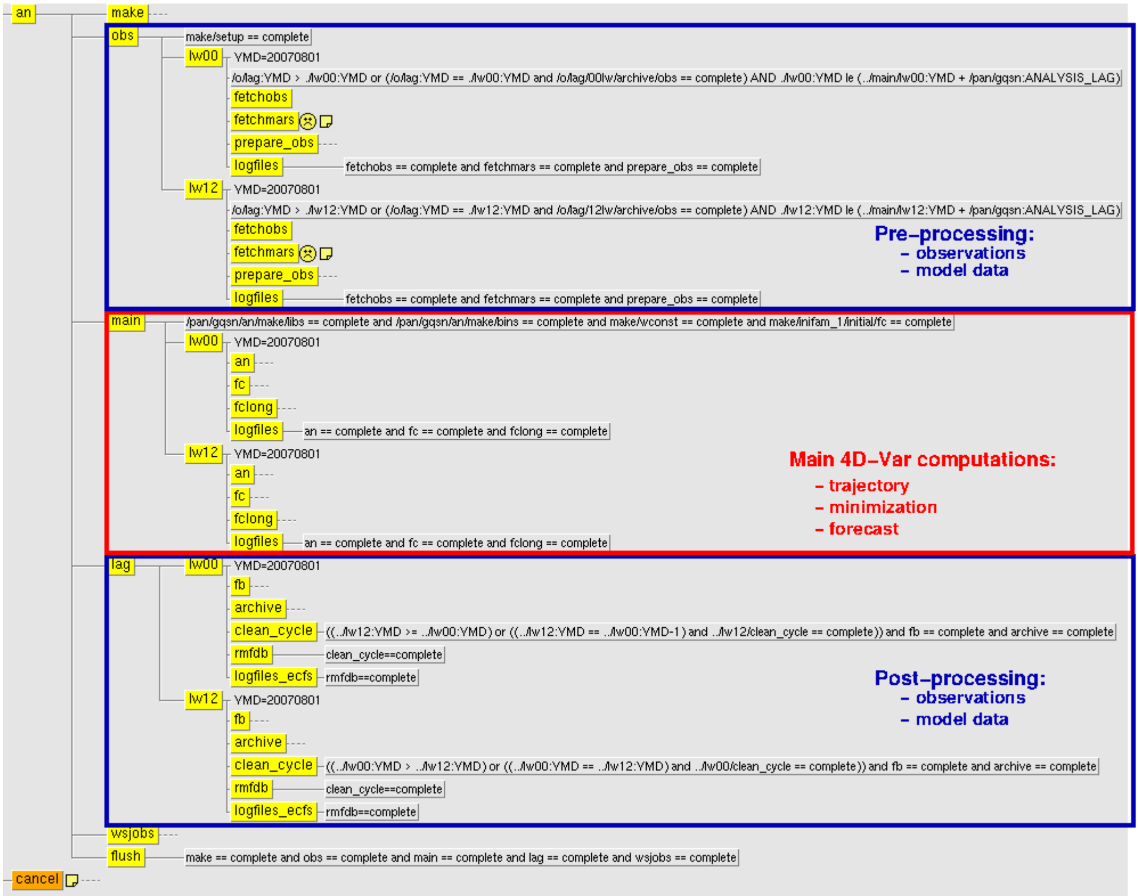
For the lidar screening flags (Table 2.6), the first 10 flags are the same as for the radar observations (although the criteria for applying them might be different). As for the radar Doppler velocity observations, the sensitivity flag for the cloud extinction observation relies on a threshold in total attenuated backscatter. For the Rayleigh backscatter there are two additional flags for indicating if there are either no clouds in the observations or no clouds in the model. These flags could be useful for preventing spurious increments in temperature or pressure from assimilating Rayleigh backscatter when the model and observations disagree on the presence of clouds.

Bitfield	Key	Description
	Lidar backscatter (CLBSC)	
0	NLID_CLBSC_ACTIVE	Observation active if no other bits set
1	NLID_CLBSC_OBS	Observed CLBSC out of bounds
2	NLID_CLBSC_NEGATIVE_Q	Negative Q in model profile
3	NLID_CLBSC_NO_OBS	No obs for this grid point (GP-space only)
4	NLID_CLBSC_NO_GRID	No grid point for this obs
5	NLID_CLBSC_PASSIVE	Passive observation
6	NLID_CLBSC_MISSING	Observation value is missing (=RMDI)
7	NLID_CLBSC_LAST_TIMESTEP	Last timestep,when TL/AD gp model doesn't run
8	NLID_CLBSC_FG_DEPARTURE	FG departure outside limit
9	NLID_CLBSC_LOW_CF	Low model or obs cloud fraction
10	NLID_CLBSC_FG_LOW	Low FG value (below sensitivity)
11	NLID_CLBSC_ATT_HIGH	Suspected excessive attenuation
	Lidar cloud extinction (CLEXT)	
0	NLID_CLEXT_ACTIVE	Observation active if no other bits set
1	NLID_CLEXT_OBS	Observed CLEXT out of bounds
2	NLID_CLEXT_NEGATIVE_Q	Negative Q in model profile
3	NLID_CLEXT_NO_OBS	No obs for this grid point (GP-space only)
4	NLID_CLEXT_NO_GRID	No grid point for this obs
5	NLID_CLEXT_PASSIVE	Passive observation
6	NLID_CLEXT_MISSING	Observation value is missing (=RMDI)
7	NLID_CLEXT_LAST_TIMESTEP	Last timestep,when TL/AD gp model doesn't run
8	NLID_CLEXT_FG_DEPARTURE	FG departure outside limit
9	NLID_CLEXT_LOW_CF	Low model or obs cloud fraction
10	NLID_CLEXT_BSC_LOW	Corresponding backscatter below sensitivity
11	NLID_CLEXT_FG_LOW	Low FG value (below sensitivity)
	Lidar rayleigh backscatter (CLRBSC)	
0	NLID_CLRBSC_ACTIVE	Observation active if no other bits set
1	NLID_CLRBSC_OBS	Observed CLRBSC out of bounds
2	NLID_CLRBSC_NEGATIVE_Q	Negative Q in model profile
3	NLID_CLRBSC_NO_OBS	No obs for this grid point (GP-space only)
4	NLID_CLRBSC_NO_GRID	No grid point for this obs
5	NLID_CLRBSC_PASSIVE	Passive observation
6	NLID_CLRBSC_MISSING	Observation value is missing (=RMDI)
7	NLID_CLRBSC_LAST_TIMESTEP	Last timestep,when TL/AD gp model doesn't run
8	NLID_CLRBSC_FG_DEPARTURE	FG departure outside limit
9	NLID_CLRBSC_LOW_CF	Low model or obs cloud fraction
10	NLID_CLRBSC_OBS_NOCLOUD	Observation cloud-free
11	NLID_CLRBSC_MOD_NOCLOUD	Model cloud-free
12	NLID_CLRBSC_FG_LOW	Low FG value (below sensitivity)

Table 2.6: Screening flags for cloud lidar observations.

2.4 Overview of 4D-Var assimilation tasks and observation processing

In order to summarize the work required to move all the development for cloud radar and lidar assimilation done during the previous ESA project (Operational Assimilation of Space-borne Radar and Lidar Cloud Profile Observations for Numerical Weather Prediction, Janisková and Fielding, 2018) from the IFS model cycle CY43R1 to higher cycle CY46R1, the structure of 4D-Var assimilation tasks is provided in Fig. 2.4. Call tree of tasks used by the 4D-Var assimilation system there and in the following subsections comes from ecFlowview - a graphical user interface used at ECMWF to display graphically the status of tasks within experiments that are in-progress. The running of a large number of programs, as required for data assimilation, is enabled by ECMWF's work-flow manager (ecFlow).



```
an
    make ----
    obs          make/setup == complete
        lw00     YMD=20070801
                 /o/lag:YMD > ./lw00:YMD or (/o/lag:YMD == ./lw00:YMD and /o/lag/00lw/archive/obs == complete) AND ./lw00:YMD le (../main/lw00:YMD + /pan/gqsn:ANALYSIS_LAG)
            fetchobs
            fetchmars
            prepare_obs ----
            logfiles     fetchobs == complete and fetchmars == complete and prepare_obs == complete
        lw12     YMD=20070801
                 /o/lag:YMD > ./lw12:YMD or (/o/lag:YMD == ./lw12:YMD and /o/lag/12lw/archive/obs == complete) AND ./lw12:YMD le (../main/lw12:YMD + /pan/gqsn:ANALYSIS_LAG)
            fetchobs
            fetchmars
            prepare_obs ----
            logfiles     fetchobs == complete and fetchmars == complete and prepare_obs == complete
    main         /pan/gqsn/an/make/libs == complete and /pan/gqsn/an/make/bins == complete and make/wconst == complete and make/inifam_1/initial/fc == complete
        lw00     YMD=20070801
            an ----
            fc ----
            fclong ----
            logfiles     an == complete and fc == complete and fclong == complete
        lw12     YMD=20070801
            an ----
            fc ----
            fclong ----
            logfiles     an == complete and fc == complete and fclong == complete
    lag
        lw00     YMD=20070801
            fb ----
            archive ----
            clean_cycle  ((../lw12:YMD >= ../lw00:YMD) or ((../lw12:YMD == ../lw00:YMD-1) and ../lw12/clean_cycle == complete)) and fb == complete and archive == complete
            rmfdb        clean_cycle==complete
            logfiles_ecfs  rmfdb==complete
        lw12     YMD=20070801
            fb ----
            archive ----
            clean_cycle  ((../lw00:YMD > ../lw12:YMD) or ((../lw00:YMD == ../lw12:YMD) and ../lw00/clean_cycle == complete)) and fb == complete and archive == complete
            rmfdb        clean_cycle==complete
            logfiles_ecfs  rmfdb==complete
    wsjobs ----
    flush        make == complete and obs == complete and main == complete and lag == complete and wsjobs == complete
cancel ----
```

Figure 2.4: Structure of the ECWMF 4D-Var assimilation computation split to three task groups: **obs** *group - tasks for observation pre-processing;* **main** *group - tasks performing main 4D-Var computations such as high resolution trajectory run, minimization and forecast run;* **lag** *group - tasks for post-processing and archiving observations and model data. Tasks groups* **lw00** *and* **lw12** *are for the 00 UTC and 12 UTC analysis production, respectively.*

The current length of the 4D-Var assimilation window at ECMWF is 12 hours, running from 21 UTC to 09 UTC to produce the 00 UTC analysis and forecast products (**lw00** task groups in Fig. 2.4), and from 09 UTC to 21 UTC for the 12 UTC production (**lw12** task groups).

The first group of tasks under **obs** is devoted to different pre-processing computations for the subsequent assimilation calculations. At this stage, observations and model data are fetched from the ECMWF archive. Observations are then pre-processed and converted from BUFR (Binary Universal Form for the Representation of meteorological data) to ODB (Observation Data Base). The main 4D-Var computations are performed under the second group of tasks, **main**, where high resolution trajectory, minimization and forecast runs are done. In the last group, **lag**, post-processing and archiving of observations and model data are done.

The handling of observations is rather complex part of the IFS data assimilation system and it happens at many different stages. Observation processing involves (i) ingestion of raw data (as BUFR files), i.e. observations themselves, to the archiving of ODBs, (ii) running the observation operators in the data assimilation, (iii) screening, (iv) quality control, (v) assigning observation errors and (vi) applying bias corrections.

Figure 2.5 illustrates how some of the data formats are used in the IFS. In the first, preparation stage, all the parameters needed to run the observation operator in the assimilation system are generated. This stage also includes format conversions (e.g. BUFR to ODB). It may further perform simple thinning (such as discarding some observations) or averaging observations to the model horizontal resolution and/or vertical matching of observations. This stage also sets up observation errors and other parameters as the settings for quality control.

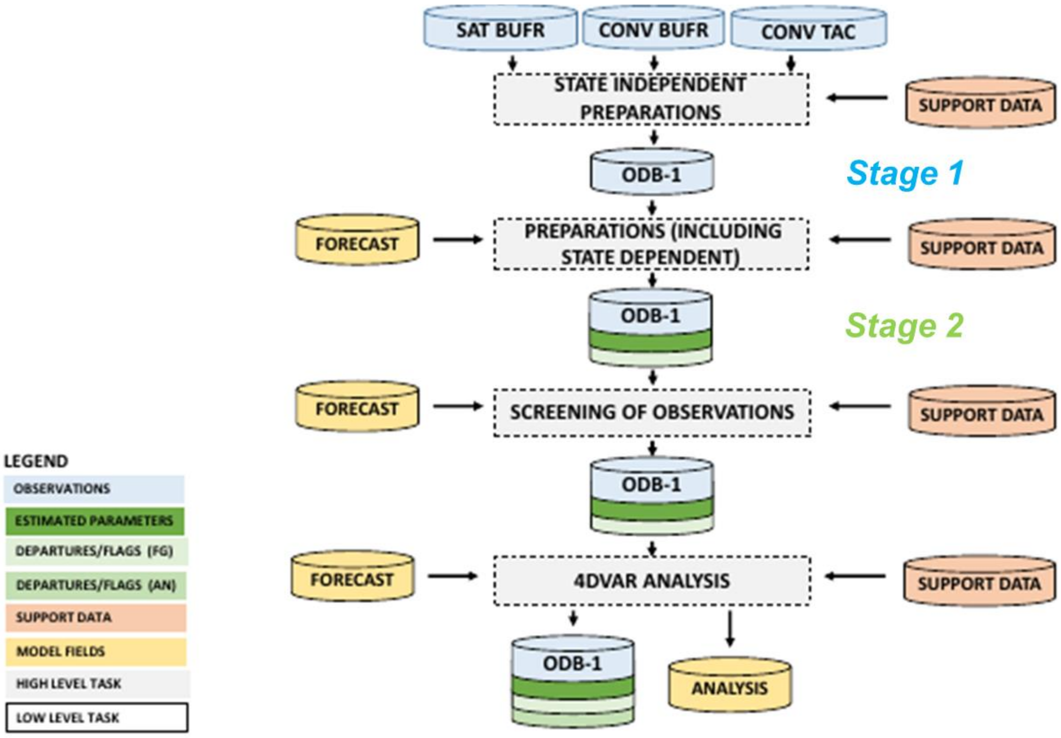


Figure 2.5: Simplified IFS observation processing flow diagram, with the disc icons representing data stores and the rectangles representing processing stages.

Screening decisions whether or not to use a certain data are made throughout the observation processing chain. They includes blacklisting and other control decisions that can only be made when a model first guess is available, or in the presence of all observations. Parameters that control all these decisions can be pre-set during pre-processing and stored in the ODB for use during the assimilation.

Observation errors are also created during observation processing. Errors, such as instrument ones, are set at the pre-processing stage as they do not depend on any model information. Other types of observation errors are created when the observation operator is called for the first time, particularly if the observation error is situation-dependent. The observation errors are then stored in the ODB for later use by the assimilation system.

Some bias correction can be performed as part of pre-processing, but most is now handled inside the 4D-Var minimization using variational bias correction (VarBC). For new observations from cloud radar and lidar, pre-computed biases will be applied during the 4D-Var minimization.

For computational reasons it can be helpful to reduce observation numbers early in processing chain, before too much computational time has been spent on unwanted data. However, much of screening needs to be left until later stages because it is often necessary to have access to: (i) the first guess (FG) model state, (ii) the first guess departures (i.e. difference between observations and their model equivalents) and/or (iii) the results of screening decisions from other observations. In the data assimilation system, there is a distinction between ’independent’ and ’dependent’ screening decisions. The dependent decisions are such as checking for redundant observations and thinning, where a selection is made among all the observations that passed quality control and blacklisting.

In general, observation preparations may take place before the main 4D-Var calculation (in the so called pre-processing stage), or inside the main calculations. The pre-processing stages are identified as ’state-independent’ (Stage 1 in Fig. 2.5) since the only place that has access to the atmospheric state is the main part. In this part, the sequence of jobs starts with the first (high resolution) trajectory run. During this run the model counterparts for all the observations are calculated through the non-linear observation operators, and the observation minus model differences (the departures) are calculated. At this stage (Stage 2), ’state-dependent’ observation processing is performed.

Technically, the final result of the observation processing is a pair of ODBs. Figure 2.5 shows how the ODB-1 data is progressively augmented with additional information as the observations flow through the system. After processing, the original ‘extended’ observation data base, ECMA (Extended CMA, Central Memory Array) ODB, contains observations complemented by the background departures, together with quality control information for most of the observations. This ECMA remains on disc for later use in feedback creation at the end of the whole 4D-Var calculation. The compressed ODB, the CCMA (Compressed CMA), contains a subset of the original observations, and is passed for the subsequent minimization job. It contains only those observations, so called active observations, that are to be used in the minimization.

2.5 Modification and development of IFS code

For including the new observations into the system, computation of the model equivalent to these observations needs to be updated to account for them. Such computation is done in the master routine, **hop**, dealing with all different types of observations in the IFS (Fig. 2.6). Currently it is assumed that each observation equivalent can be computed from a single vertical profile of model data. After obtaining these equivalents, their departures to observations and the observation cost function are computed Bias correction is also carried out at this point by subtracting the bias estimate from the calculated departure. Finally the departure is divided by the observation error to form the normalized departure.

The routine for computation of observation cost function also stores the resulting effective departure in ODB, for reuse as the input to the adjoint. The effective departure is the normalized departure after the effects of (vertical) observation error correlation and quality control have been taken into account.

Under the **hop** routine, the subroutine for all computations related to cloud radar and lidar observations, **obsop_clradlid**, is called. At this point, having access to the model state information, computation of situation dependent observation errors, screening for high observation errors as well as departures and assigning bias correction are performed. Since some of those requires knowledge of the model equivalent to the observations, the above observation processing is only done once such equivalent is

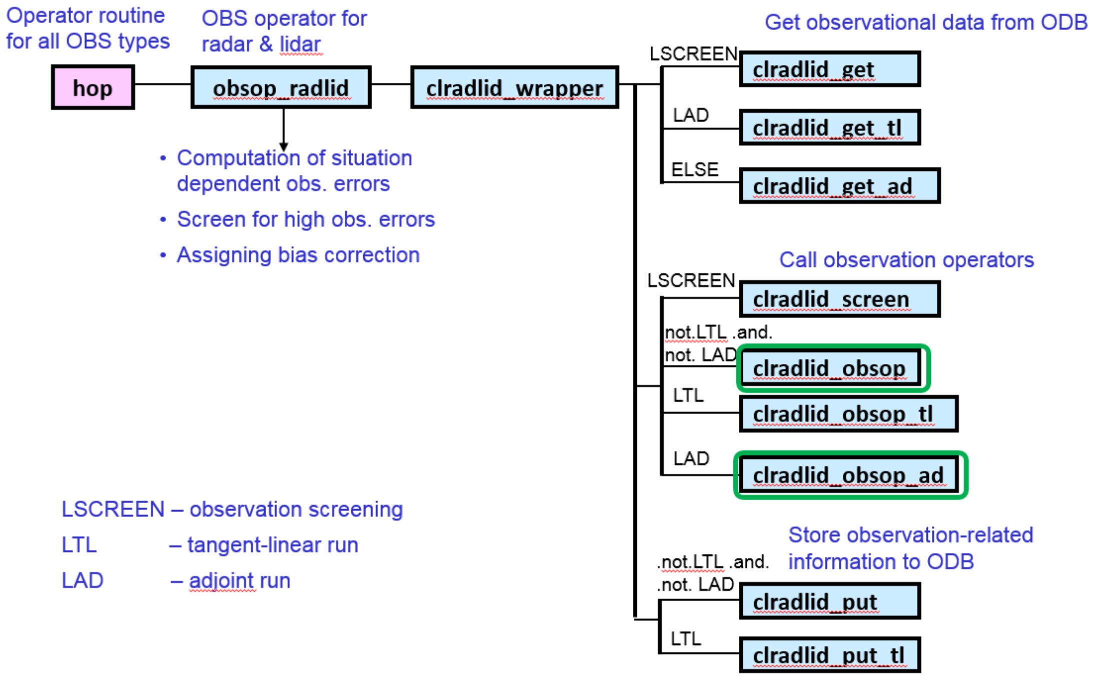

Operator routine for all OBS types
OBS operator for radar & lidar
hop
obsop_radlid
clradlid_wrapper
• Computation of situation dependent obs. errors
• Screen for high obs. errors
• Assigning bias correction
Get observational data from ODB
LSCREEN
clradlid_get
LAD
clradlid_get_tl
ELSE
clradlid_get_ad
Call observation operators
LSCREEN
clradlid_screen
not.LTL .and. not. LAD
clradlid_obsop
LTL
clradlid_obsop_tl
LAD
clradlid_obsop_ad
Store observation-related information to ODB
.not.LTL .and. .not. LAD
clradlid_put
LTL
clradlid_put_tl
LSCREEN – observation screening
LTL – tangent-linear run
LAD – adjoint run

Figure 2.6: Schematic diagram describing inclusion of the observation operators for computing model equivalent to cloud radar and lidar observations, as well as getting/storing these observational data in ODB.

available after calling **clradlid_wrapper** routine. In this routine, firstly observational data are obtained from ODB using **clradl_get** routines (suffix **_tl** and **_ad** are used for the tangent-linear, TL, and adjoint, AD, versions of those routines). Observation screening depending on model situation is done in routine **clradlid_screen**. The model equivalents to cloud radar and lidar observations are obtained by performing calculations in the routines **clradlid_obsop** (suffix **_tl** and **_ad** for TL and AD versions, respectively). More detailed structure of the subroutines for the observation operators is below (Fig. 2.7). At the end of **clradlid_wrapper** routine, observation related information is stored to ODB in the **clradlid_put** and **clradlid_put_tl** routines.

Figure 2.7 provide the detailed structure of the observation operator subroutines for the direct non-linear and adjoint integrations, respectively.

A brief description of the contents of all the observation operator subroutines for cloud radar and lidar observations follows:

clradar_driver and **cllidar_driver** are drivers to all routines for computation of overlap (cloud and precipitation related quantities), cloud radar reflectivity or lidar backscatter.

clradlid_cp_overlap is used for the computation of overlap for cloud and precipitation related quantities.

clradlid_radars is used to compute of radar reflectivity and extinction from the input values (temperature, specific humidity, rain, snow, cloud, ice) using a look-up table of radar reflectivity and extinction values. In this routine, cloudy sky attenuation is also given by the sum of clear sky attenuation and the hydrometeor extinction. **clradlid_lidars** computes lidar backscatter and extinction from the input values (temperature, specific humidity, rain, snow, cloud, ice) using a look-up table of lidar backscatter and extinction values.

clradlid_lut_radars serves for localization of temperature and hydrometeor intervals and searching the

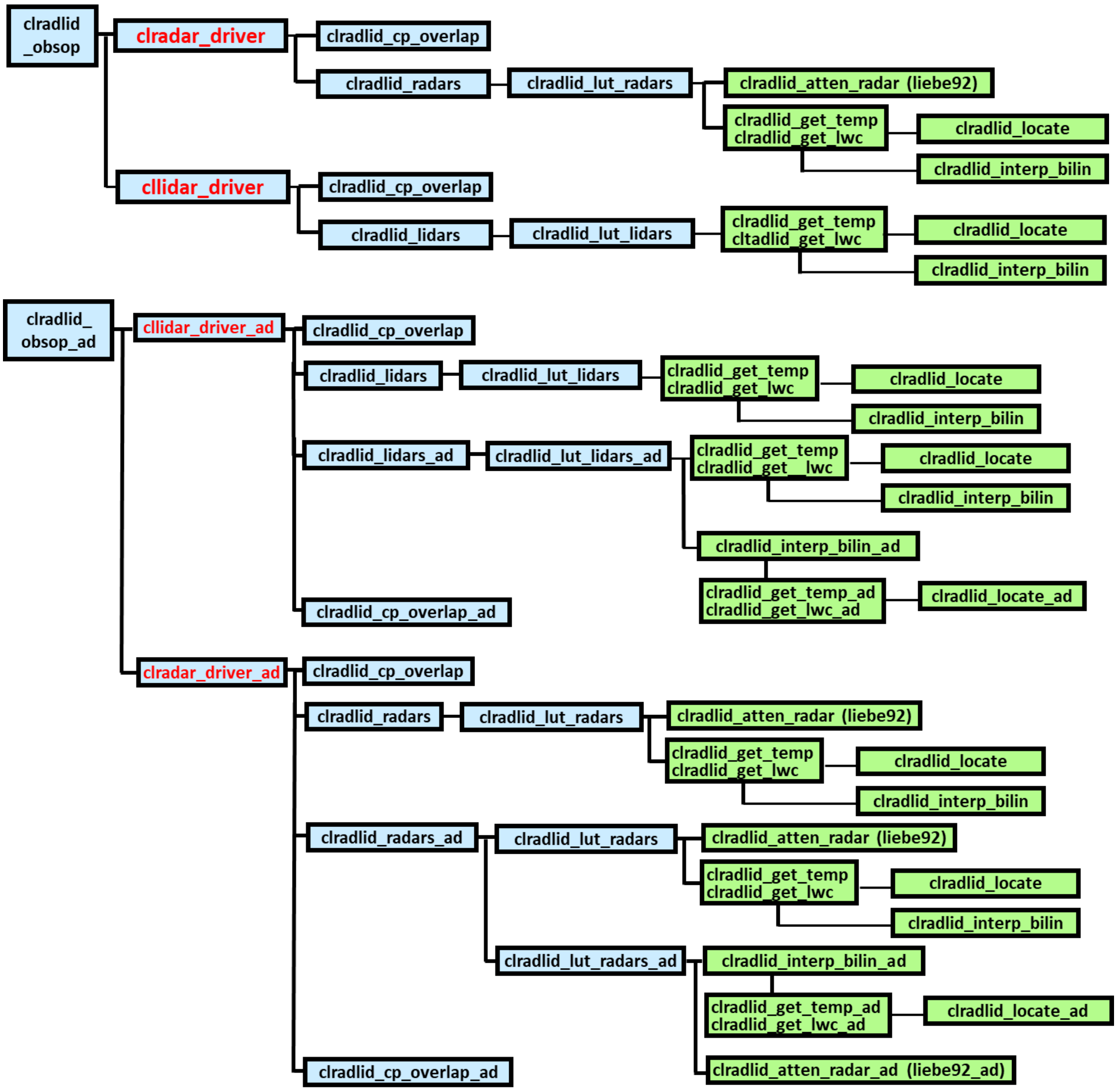


Figure 2.7: Call tree of routines for direct observation operators (top) and adjoint versions of observation operators (bottom) used for cloud radar and lidar observations. Two groups of operators are called under **clradlid_obsop***:* **clradar_driver** *- cloud radar operator;* **cllidar_driver** *- cloud lidar operator.*

reflectivity and extinction tables. It then calls routine for interpolation according to the temperature and the water content. The scaling of extinction and reflectivity to obtain in cloud values is also performed there. **clradlid_lut_lidars** provides the same as **clradlid_lut_radars**, but for the lidar backscatter and extinction.

clradlid_lut_lidars provides the same as **clradlid_lut_radars**, but for the lidar backscatter and extinction.

clradlid_get_temp performs searching to obtain specified model temperature values from the lookup tables.

clradlid_get_lwc performs searching to obtain specified model water content values from the lookup tables.

clradlid_locate is used for locating position in lookup table based on prescribed temperature or water content intervals and model values.

clradlid_interp_bilin performs bilinear interpolation using the model temperature and in-cloud water content to obtain unattenuated radar reflectivity, lidar backscatter or extinction for each hydrometeor type from the pre-computed lookup-tables.

clradlid_atten_radar (subroutine name in the code is **liebe92**) computes attenuation of the signal along the radar beam. Clear sky attenuation due to gases is calculated using the model of Liebe (1985); Liebe et al. (1992).

A call tree for adjoint version of the routines for observation operators is at the bottom of Fig. 2.7. Adjoint routines have the same name as nonlinear, but suffix **_ad** is added.

3 Migration of the assimilation system development for cloud radar and lidar - from CY43R1 to 46R1

3.1 A brief analysis of required work

Although, the system enabling assimilation of cloud radar and lidar observations was developed and tested in the previous project, its update to the newer model cycle requires to move all different steps of that development to account for these observations. The particular steps of 4D-Var solution, where modifications and developments have been done (with indication *in italic*) and needed to be migrated, are summarized here:

i. Comparison of the observations with the background is done at high resolution to compute the innovation vector providing the departure of the model background equivalent to the observation. This procedure involves using appropriate observation operators which need to be properly included in the 4D-Var system. *The direct, non-linear, versions of cloud radar and lidar observation operators had to be included for that.*

The background departures must then be stored in the ODB for later use in the minimization. This job step also performs screening (i.e. blacklisting, thinning and quality control against the background) of observations. The screening determines which observations will be passed to the main minimization. Very large volumes of data are present during the screening run only, for the purpose of data monitoring. *These steps had to be implemented for the new observations as well.*

The model trajectory is subsequently interpolated to the resolution of the next job step and written out.

ii. The first minimization at low resolution is used to produce preliminary low-resolution analysis increments, using simplified tangent-linear and adjoint physics, and tangent-linear and adjoint observation operators. *Therefore the tangent-linear and adjoint versions of cloud radar and lidar observations were included to the system.*

iii. First update of the high-resolution trajectory to take non-linear effects partly into account applies the analysis increments obtained in the first minimization and performs another forecast integration with comparison to observations. Departures from this new atmospheric state are stored in the ODB. *This procedure also involves the new observations.*

The analysis problem is then re-linearized around the updated model state which provides a new linearization state for the next minimization. The updated model trajectory is then interpolated to the resolution of the next job step and written out.

iv. The second main minimization is performed at increased resolution with a more complete representation of tangent-linear and adjoint physics. *The tangent-linear and adjoint versions of cloud radar and lidar observation operators are also used in this minimization.*

Steps (*iii*) and (*iv*) are then repeated again in the current operational data assimilation system.

v. The production of the high-resolution analysis is finally carried out by adding the low-resolution increments to the background (at initial time), and integrating to the analysis times. The comparison of the analysis with all observations (including those not used by the analysis, but also those used for diagnostic purposes) is performed. *This step had also to be implemented for the new observations.*

3.2 Work carried to move cloud radar and lidar assimilation to the model cycle 46R1

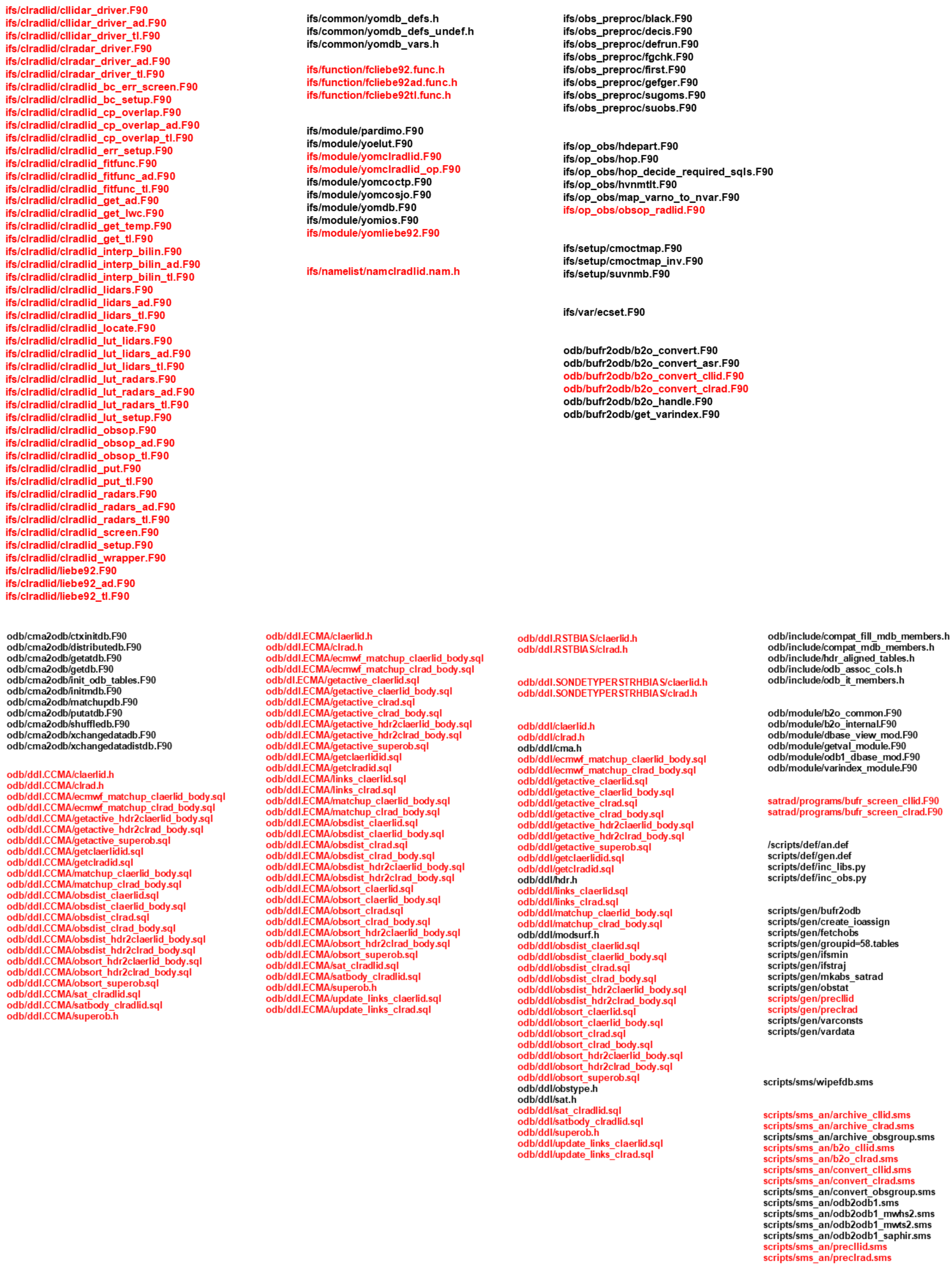
As described in the previous section 3.1, the amount of migration work from the model cycle CY43R1, where all required modifications and/or developments to account for the new observation types of cloud radar and lidar were originally made, to the newer cycle CY46R1 was rather large.

Figure 3.1 shows the list of over 300 new and modified subroutines and scripts to be included in the model cycle CY46R1 for 4D-Var assimilation of cloud radar and lidar observations. Since the development had to be moved over cycles (CY43R1 → CY46R1), there were many changes to the files containing original development, which often led to conflicts that needed to be resolved. Unfortunately, all the migration had to be done manually, without using any developing/version controlling software. This was the case because the versioning software used by CY43R1, Perforce, was replaced by Git in CY46R1.

For an illustration, Fig. 3.2 shows an example of differences to be dealt with when moving the development for cloud radar and lidar assimilation over so many model cycles. Figure contains only on part of differences for the existing routine, hop.F90, master routine dealing with all different types of observations in the IFS. However, on the right side of figure, the red and blue colour bars provide some indication of how many changes between the model cycles had to be checked in order to assure that our modifications were properly included in the newer cycle.

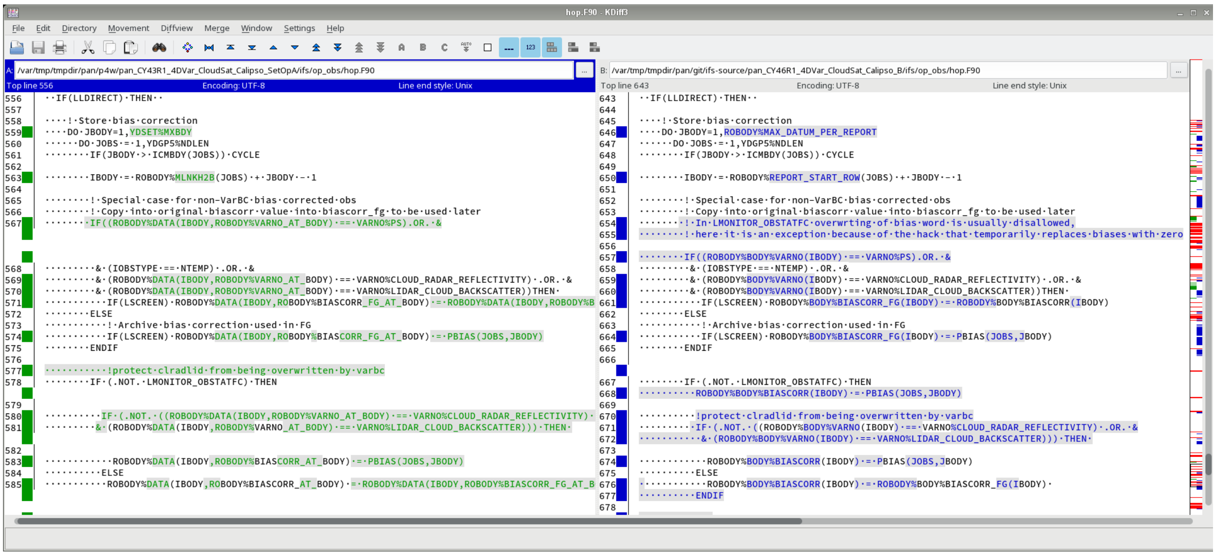
Unfortunately, even many the new routines originally developed for cloud radar and lidar assimilation could not be just simply copied from one cycle to another. As Fig. 3.3 shows, there were many changes to be made even to the new routines. This is clearly illustrated on the new routine used to get cloud radar reflectivity and lidar backscatter observations from ODB to the model grid-point-space.

The analysis of the files to be migrated clearly showed that it was necessary to carefully check not only existing IFS routines, which needed to be modified in order to take into account cloud radar and lidar observations, but also the new files developed for these observations. With the extent of differences clearly illustrated in Figs. 3.2 and 3.3, it was obvious to expect that making all development for cloud radar and lidar assimilation working in the cycle CY46R1 will require a lot of debugging (i.e. searching for errors introduced by migration). Indeed we spent several weeks to make cloud radar and lidar assimilation working in CY46R1. We had not only needed to resolve coding errors, but also to find out extra modifications required for these observations by the new cycle. Ironically ☺, all this was even more challenging since the whole debugging process was carried remotely due to office closure because of the coronavirus outbreak. Figure 3.4 illustrates how we spent many hours online debugging, searching for the reasons of frustrating job failures (red boxes in ecFlowview displaying the status of experiment tasks). To resolve some problems, we needed to discus and get some advice and help from other teams at ECMWF, which was inevitably more difficult when working remotely. However, all the hard work paid off and **we managed to make system working in the cycle CY46R1**.



```
ifs/clradlid/cllidar_driver.F90
ifs/clradlid/cllidar_driver_ad.F90
ifs/clradlid/cllidar_driver_tl.F90
ifs/clradlid/clradar_driver.F90
ifs/clradlid/clradar_driver_ad.F90
ifs/clradlid/clradar_driver_tl.F90
ifs/clradlid/clradlid_bc_err_screen.F90
ifs/clradlid/clradlid_bc_setup.F90
ifs/clradlid/clradlid_cp_overlap.F90
ifs/clradlid/clradlid_cp_overlap_ad.F90
ifs/clradlid/clradlid_cp_overlap_tl.F90
ifs/clradlid/clradlid_err_setup.F90
ifs/clradlid/clradlid_fitfunc.F90
ifs/clradlid/clradlid_fitfunc_ad.F90
ifs/clradlid/clradlid_fitfunc_tl.F90
ifs/clradlid/clradlid_get_ad.F90
ifs/clradlid/clradlid_get_lwc.F90
ifs/clradlid/clradlid_get_temp.F90
ifs/clradlid/clradlid_get_tl.F90
ifs/clradlid/clradlid_interp_bilin.F90
ifs/clradlid/clradlid_interp_bilin_ad.F90
ifs/clradlid/clradlid_interp_bilin_tl.F90
ifs/clradlid/clradlid_lidars.F90
ifs/clradlid/clradlid_lidars_ad.F90
ifs/clradlid/clradlid_lidars_tl.F90
ifs/clradlid/clradlid_locate.F90
ifs/clradlid/clradlid_lut_lidars.F90
ifs/clradlid/clradlid_lut_lidars_ad.F90
ifs/clradlid/clradlid_lut_lidars_tl.F90
ifs/clradlid/clradlid_lut_radars.F90
ifs/clradlid/clradlid_lut_radars_ad.F90
ifs/clradlid/clradlid_lut_radars_tl.F90
ifs/clradlid/clradlid_lut_setup.F90
ifs/clradlid/clradlid_obsop.F90
ifs/clradlid/clradlid_obsop_ad.F90
ifs/clradlid/clradlid_obsop_tl.F90
ifs/clradlid/clradlid_put.F90
ifs/clradlid/clradlid_put_tl.F90
ifs/clradlid/clradlid_radars.F90
ifs/clradlid/clradlid_radars_ad.F90
ifs/clradlid/clradlid_radars_tl.F90
ifs/clradlid/clradlid_screen.F90
ifs/clradlid/clradlid_setup.F90
ifs/clradlid/clradlid_wrapper.F90
ifs/clradlid/liebe92.F90
ifs/clradlid/liebe92_ad.F90
ifs/clradlid/liebe92_tl.F90

ifs/common/yomdb_defs.h
ifs/common/yomdb_defs_undef.h
ifs/common/yomdb_vars.h

ifs/function/fcliebe92.func.h
ifs/function/fcliebe92ad.func.h
ifs/function/fcliebe92tl.func.h

ifs/module/pardimo.F90
ifs/module/yoelut.F90
ifs/module/yomclradlid.F90
ifs/module/yomclradlid_op.F90
ifs/module/yomcoctp.F90
ifs/module/yomcosjo.F90
ifs/module/yomdb.F90
ifs/module/yomios.F90
ifs/module/yomliebe92.F90

ifs/namelist/namclradlid.nam.h

ifs/obs_preproc/black.F90
ifs/obs_preproc/decis.F90
ifs/obs_preproc/defrun.F90
ifs/obs_preproc/fgchk.F90
ifs/obs_preproc/first.F90
ifs/obs_preproc/gefger.F90
ifs/obs_preproc/sugoms.F90
ifs/obs_preproc/suobs.F90

ifs/op_obs/hdepart.F90
ifs/op_obs/hop.F90
ifs/op_obs/hop_decide_required_sqls.F90
ifs/op_obs/hvnmtlt.F90
ifs/op_obs/map_varno_to_nvar.F90
ifs/op_obs/obsop_radlid.F90

ifs/setup/cmoctmap.F90
ifs/setup/cmoctmap_inv.F90
ifs/setup/suvnmb.F90

ifs/var/ecset.F90

odb/bufr2odb/b2o_convert.F90
odb/bufr2odb/b2o_convert_asr.F90
odb/bufr2odb/b2o_convert_cllid.F90
odb/bufr2odb/b2o_convert_clrad.F90
odb/bufr2odb/b2o_handle.F90
odb/bufr2odb/get_varindex.F90

odb/cma2odb/ctxinitdb.F90
odb/cma2odb/distributedb.F90
odb/cma2odb/getatdb.F90
odb/cma2odb/getdb.F90
odb/cma2odb/init_odb_tables.F90
odb/cma2odb/initmdb.F90
odb/cma2odb/matchupdb.F90
odb/cma2odb/putatdb.F90
odb/cma2odb/shuffledb.F90
odb/cma2odb/xchangedatadb.F90
odb/cma2odb/xchangedatadistdb.F90

odb/ddl.CCMA/claerlid.h
odb/ddl.CCMA/clrad.h
odb/ddl.CCMA/ecmwf_matchup_claerlid_body.sql
odb/ddl.CCMA/ecmwf_matchup_clrad_body.sql
odb/ddl.CCMA/getactive_hdr2claerlid_body.sql
odb/ddl.CCMA/getactive_hdr2clrad_body.sql
odb/ddl.CCMA/getactive_superob.sql
odb/ddl.CCMA/getclaerlidid.sql
odb/ddl.CCMA/getclradid.sql
odb/ddl.CCMA/matchup_claerlid_body.sql
odb/ddl.CCMA/matchup_clrad_body.sql
odb/ddl.CCMA/obsdist_claerlid.sql
odb/ddl.CCMA/obsdist_claerlid_body.sql
odb/ddl.CCMA/obsdist_clrad.sql
odb/ddl.CCMA/obsdist_clrad_body.sql
odb/ddl.CCMA/obsdist_hdr2claerlid_body.sql
odb/ddl.CCMA/obsdist_hdr2clrad_body.sql
odb/ddl.CCMA/obsort_hdr2claerlid_body.sql
odb/ddl.CCMA/obsort_hdr2clrad_body.sql
odb/ddl.CCMA/obsort_superob.sql
odb/ddl.CCMA/sat_clradlid.sql
odb/ddl.CCMA/satbody_clradlid.sql
odb/ddl.CCMA/superob.h

odb/ddl.ECMA/claerlid.h
odb/ddl.ECMA/clrad.h
odb/ddl.ECMA/ecmwf_matchup_claerlid_body.sql
odb/ddl.ECMA/ecmwf_matchup_clrad_body.sql
odb/dl.ECMA/getactive_claerlid.sql
odb/ddl.ECMA/getactive_claerlid_body.sql
odb/ddl.ECMA/getactive_clrad.sql
odb/ddl.ECMA/getactive_clrad_body.sql
odb/ddl.ECMA/getactive_hdr2claerlid_body.sql
odb/ddl.ECMA/getactive_hdr2clrad_body.sql
odb/ddl.ECMA/getactive_superob.sql
odb/ddl.ECMA/getclaerlidid.sql
odb/ddl.ECMA/getclradid.sql
odb/ddl.ECMA/links_claerlid.sql
odb/ddl.ECMA/links_clrad.sql
odb/ddl.ECMA/matchup_claerlid_body.sql
odb/ddl.ECMA/matchup_clrad_body.sql
odb/ddl.ECMA/obsdist_claerlid.sql
odb/ddl.ECMA/obsdist_claerlid_body.sql
odb/ddl.ECMA/obsdist_clrad.sql
odb/ddl.ECMA/obsdist_clrad_body.sql
odb/ddl.ECMA/obsdist_hdr2claerlid_body.sql
odb/ddl.ECMA/obsdist_hdr2clrad_body.sql
odb/ddl.ECMA/obsort_claerlid.sql
odb/ddl.ECMA/obsort_claerlid_body.sql
odb/ddl.ECMA/obsort_clrad.sql
odb/ddl.ECMA/obsort_clrad_body.sql
odb/ddl.ECMA/obsort_hdr2claerlid_body.sql
odb/ddl.ECMA/obsort_hdr2clrad_body.sql
odb/ddl.ECMA/obsort_superob.sql
odb/ddl.ECMA/sat_clradlid.sql
odb/ddl.ECMA/satbody_clradlid.sql
odb/ddl.ECMA/superob.h
odb/ddl.ECMA/update_links_claerlid.sql
odb/ddl.ECMA/update_links_clrad.sql

odb/ddl.RSTBIAS/claerlid.h
odb/ddl.RSTBIAS/clrad.h

odb/ddl.SONDETYPERSTRHBIAS/claerlid.h
odb/ddl.SONDETYPERSTRHBIAS/clrad.h

odb/ddl/claerlid.h
odb/ddl/clrad.h
odb/ddl/cma.h
odb/ddl/ecmwf_matchup_claerlid_body.sql
odb/ddl/ecmwf_matchup_clrad_body.sql
odb/ddl/getactive_claerlid.sql
odb/ddl/getactive_claerlid_body.sql
odb/ddl/getactive_clrad.sql
odb/ddl/getactive_clrad_body.sql
odb/ddl/getactive_hdr2claerlid_body.sql
odb/ddl/getactive_hdr2clrad_body.sql
odb/ddl/getactive_superob.sql
odb/ddl/getclaerlidid.sql
odb/ddl/getclradid.sql
odb/ddl/hdr.h
odb/ddl/links_claerlid.sql
odb/ddl/links_clrad.sql
odb/ddl/matchup_claerlid_body.sql
odb/ddl/matchup_clrad_body.sql
odb/ddl/modsurf.h
odb/ddl/obsdist_claerlid.sql
odb/ddl/obsdist_claerlid_body.sql
odb/ddl/obsdist_clrad.sql
odb/ddl/obsdist_clrad_body.sql
odb/ddl/obsdist_hdr2claerlid_body.sql
odb/ddl/obsdist_hdr2clrad_body.sql
odb/ddl/obsort_claerlid.sql
odb/ddl/obsort_claerlid_body.sql
odb/ddl/obsort_clrad.sql
odb/ddl/obsort_clrad_body.sql
odb/ddl/obsort_hdr2claerlid_body.sql
odb/ddl/obsort_hdr2clrad_body.sql
odb/ddl/obsort_superob.sql
odb/ddl/obstype.h
odb/ddl/sat.h
odb/ddl/sat_clradlid.sql
odb/ddl/satbody_clradlid.sql
odb/ddl/superob.h
odb/ddl/update_links_claerlid.sql
odb/ddl/update_links_clrad.sql

odb/include/compat_fill_mdb_members.h
odb/include/compat_mdb_members.h
odb/include/hdr_aligned_tables.h
odb/include/odb_assoc_cols.h
odb/include/odb_it_members.h

odb/module/b2o_common.F90
odb/module/b2o_internal.F90
odb/module/dbase_view_mod.F90
odb/module/getval_module.F90
odb/module/odb1_dbase_mod.F90
odb/module/varindex_module.F90

satrad/programs/bufr_screen_cllid.F90
satrad/programs/bufr_screen_clrad.F90

/scripts/def/an.def
scripts/def/gen.def
scripts/def/inc_libs.py
scripts/def/inc_obs.py

scripts/gen/bufr2odb
scripts/gen/create_ioassign
scripts/gen/fetchobs
scripts/gen/groupid=58.tables
scripts/gen/ifsmin
scripts/gen/ifstraj
scripts/gen/mkabs_satrad
scripts/gen/obstat
scripts/gen/precllid
scripts/gen/preclrad
scripts/gen/varconsts
scripts/gen/vardata

scripts/sms/wipefdb.sms

scripts/sms_an/archive_cllid.sms
scripts/sms_an/archive_clrad.sms
scripts/sms_an/archive_obsgroup.sms
scripts/sms_an/b2o_cllid.sms
scripts/sms_an/b2o_clrad.sms
scripts/sms_an/convert_cllid.sms
scripts/sms_an/convert_clrad.sms
scripts/sms_an/convert_obsgroup.sms
scripts/sms_an/odb2odb1.sms
scripts/sms_an/odb2odb1_mwhs2.sms
scripts/sms_an/odb2odb1_mwts2.sms
scripts/sms_an/odb2odb1_saphir.sms
scripts/sms_an/precllid.sms
scripts/sms_an/preclrad.sms
```

Figure 3.1: The list of new (in red) and modified (in black) files required to be included in the model cycle CY46R1 for 4D-Var assimilation of cloud radar and lidar observations.



A: /var/tmp/tmpdir/pan/p4w/pan_CY43R1_4DVar_CloudSat_Calipso_SetOpA/ifs/op_obs/hop.F90

Top line 556 Encoding: UTF-8 Line end style: Unix

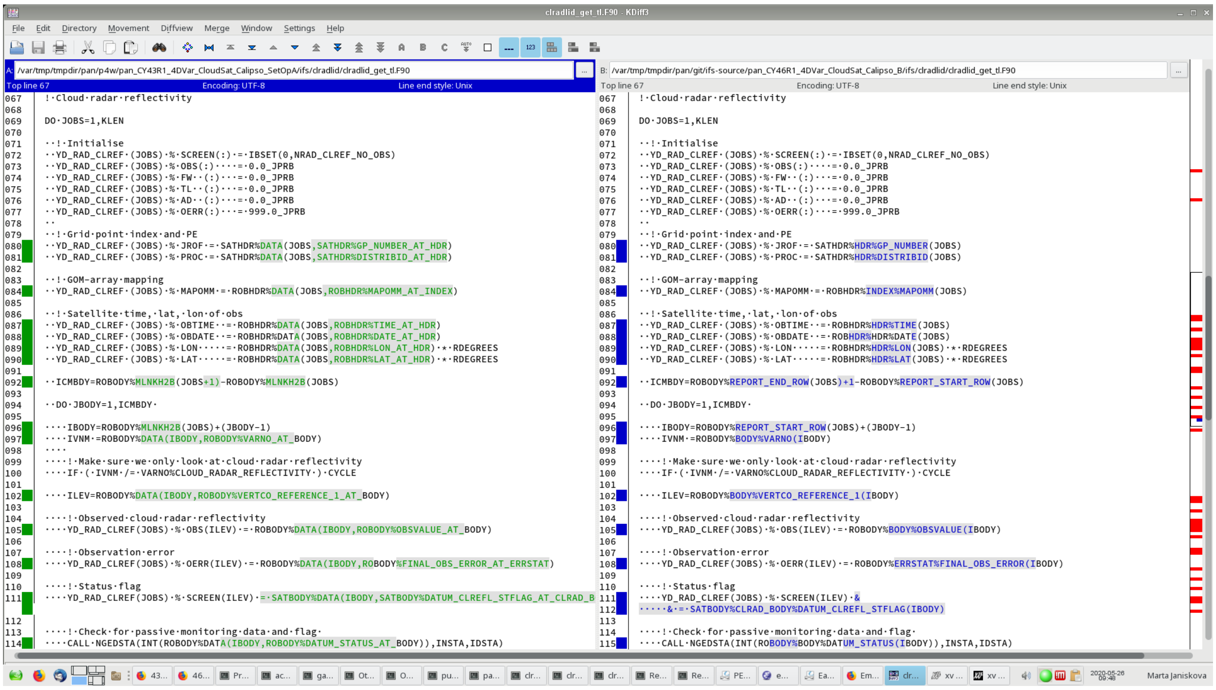
```
  IF(LLDIRECT) THEN  

    ! Store bias correction
    DO JBODY=1,YDSET%MXBDY
      DO JOBS = 1,YDGP5%NDLEN
        IF(JBODY > ICMBDY(JOBS)) CYCLE

        IBODY = ROBODY%MLNKH2B(JOBS) + JBODY - 1

        ! Special case for non-VarBC bias corrected obs
        ! Copy into original biascorr value into biascorr_fg to be used later
        IF((ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY) == VARNO%PS).OR. &


          & (IOBSTYPE == NTEMP) .OR. &
          & (ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
          & (ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY) == VARNO%LIDAR_CLOUD_BACKSCATTER))THEN 
            IF(LSCREEN) ROBODY%DATA(IBODY,ROBODY%BIASCORR_FG_AT_BODY) = ROBODY%DATA(IBODY,ROBODY%B
        ELSE
            ! Archive bias correction used in FG
            IF(LSCREEN) ROBODY%DATA(IBODY,ROBODY%BIASCORR_FG_AT_BODY) = PBIAS(JOBS,JBODY)
        ENDIF

            !protect clradlid from being overwritten by varbc
        IF (.NOT. LMONITOR_OBSTATFC) THEN


          IF (.NOT. ((ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY) == VARNO%CLOUD_RADAR_REFLECTIVITY) 
          & (ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY) == VARNO%LIDAR_CLOUD_BACKSCATTER))) THEN 


             ROBODY%DATA(IBODY,ROBODY%BIASCORR_AT_BODY) = PBIAS(JOBS,JBODY)
            ELSE
            ROBODY%DATA(IBODY,ROBODY%BIASCORR_AT_BODY) = ROBODY%DATA(IBODY,ROBODY%BIASCORR_FG_AT_B
```

B: /var/tmp/tmpdir/pan/git/ifs-source/pan_CY46R1_4DVar_CloudSat_Calipso_B/ifs/op_obs/hop.F90

Top line 643 Encoding: UTF-8 Line end style: Unix

```
  IF(LLDIRECT) THEN  

    ! Store bias correction
    DO JBODY=1,ROBODY%MAX_DATUM_PER_REPORT
      DO JOBS = 1,YDGP5%NDLEN
        IF(JBODY > ICMBDY(JOBS)) CYCLE

        IBODY = ROBODY%REPORT_START_ROW(JOBS) + JBODY - 1

        ! Special case for non-VarBC bias corrected obs
        ! Copy into original biascorr value into biascorr_fg to be used later
        ! In LMONITOR_OBSTATFC overwrting of bias word is usually disallowed,
        ! here it is an exception because of the hack that temporarily replaces biases with zero

        IF((ROBODY%BODY%VARNO(IBODY) == VARNO%PS).OR. &
          & (IOBSTYPE == NTEMP) .OR. &
          & (ROBODY%BODY%VARNO(IBODY) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
          & (ROBODY%BODY%VARNO(IBODY) == VARNO%LIDAR_CLOUD_BACKSCATTER))THEN 
            IF(LSCREEN) ROBODY%BODY%BIASCORR_FG(IBODY) = ROBODY%BODY%BIASCORR(IBODY)
        ELSE
            ! Archive bias correction used in FG
            IF(LSCREEN) ROBODY%BODY%BIASCORR_FG(IBODY) = PBIAS(JOBS,JBODY)
        ENDIF


        IF (.NOT. LMONITOR_OBSTATFC) THEN
          ROBODY%BODY%BIASCORR(IBODY) = PBIAS(JOBS,JBODY)

          !protect clradlid from being overwritten by varbc
         IF (.NOT. ((ROBODY%BODY%VARNO(IBODY) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
          & (ROBODY%BODY%VARNO(IBODY) == VARNO%LIDAR_CLOUD_BACKSCATTER))) THEN 

            ROBODY%BODY%BIASCORR(IBODY) = PBIAS(JOBS,JBODY)
          ELSE
             ROBODY%BODY%BIASCORR(IBODY) = ROBODY%BODY%BIASCORR_FG(IBODY) 
          ENDIF

```

Figure 3.2: An example of differences to be dealt with when moving the development for cloud radar and lidar assimilation from the model cycle CY43R1 to more recent cycle 46R1. Differences are shown for the existing routine, hop.F90, master routine dealing with all different types of observations in the IFS.



A: /var/tmp/tmpdir/pan/p4w/pan_CY43R1_4DVar_CloudSat_Calipso_SetOpA/ifs/clradlid/clradlid_get_tl.F90

Top line 67 Encoding: UTF-8 Line end style: Unix

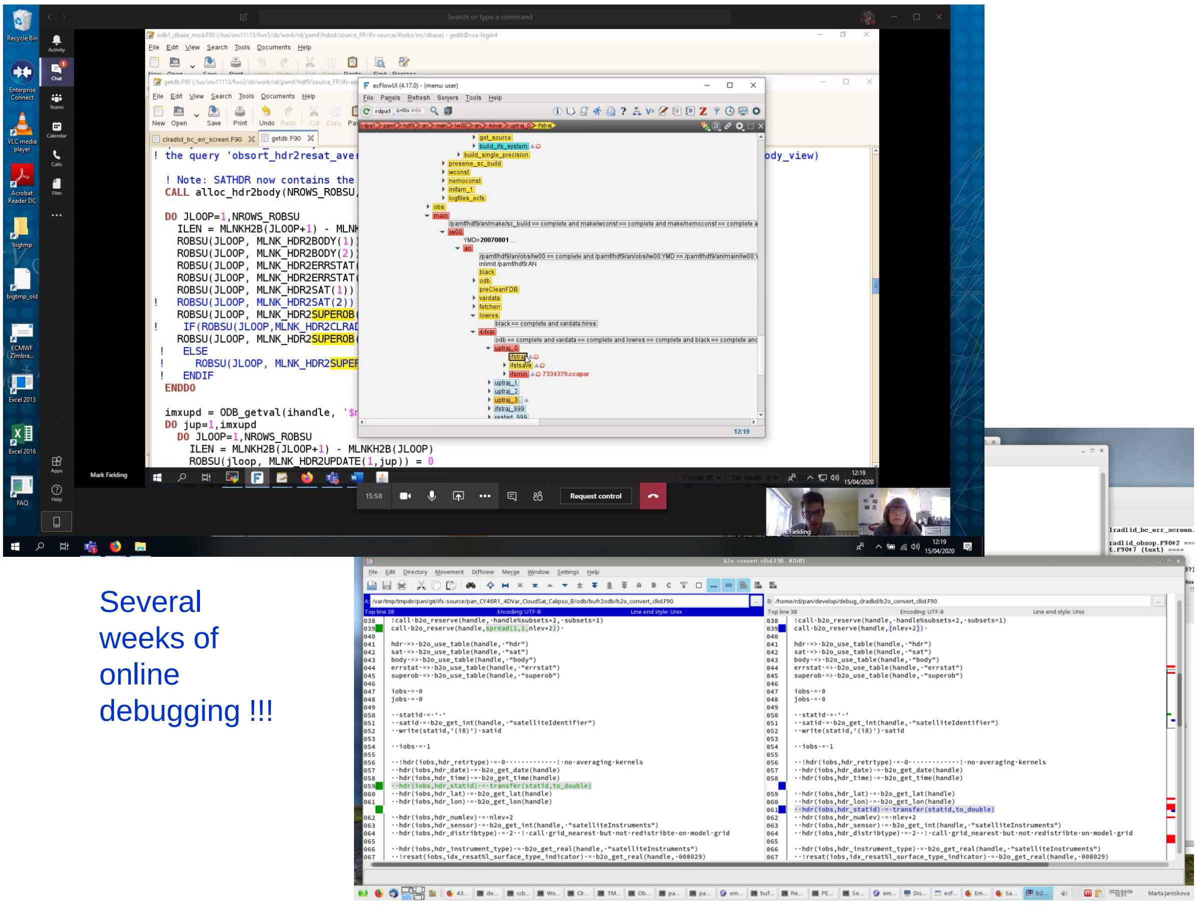
```
! Cloud radar reflectivity

DO JOBS=1,KLEN

  ! Initialise
  YD_RAD_CLREF (JOBS) % SCREEN(:) = IBSET(0,NRAD_CLREF_NO_OBS)
  YD_RAD_CLREF (JOBS) % OBS(:)    = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % FW  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % TL  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % AD  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % OERR(:)   = 999.0_JPRB
  
  ! Grid point index and PE
  YD_RAD_CLREF (JOBS) % JROF = SATHDR%DATA(JOBS,SATHDR%GP_NUMBER_AT_HDR)
  YD_RAD_CLREF (JOBS) % PROC = SATHDR%DATA(JOBS,SATHDR%DISTRIBID_AT_HDR)

  ! GOM-array mapping
  YD_RAD_CLREF (JOBS) % MAPOMM = ROBHDR%DATA(JOBS,ROBHDR%MAPOMM_AT_INDEX)

  ! Satellite time, lat, lon of obs
  YD_RAD_CLREF (JOBS) % OBTIME  = ROBHDR%DATA(JOBS,ROBHDR%TIME_AT_HDR)
  YD_RAD_CLREF (JOBS) % OBDATE  = ROBHDR%DATA(JOBS,ROBHDR%DATE_AT_HDR)
  YD_RAD_CLREF (JOBS) % LON     = ROBHDR%DATA(JOBS,ROBHDR%LON_AT_HDR) * RDEGREES
  YD_RAD_CLREF (JOBS) % LAT     = ROBHDR%DATA(JOBS,ROBHDR%LAT_AT_HDR) * RDEGREES

  ICMBDY=ROBODY%MLNKH2B(JOBS+1)-ROBODY%MLNKH2B(JOBS)

  DO JBODY=1,ICMBDY 

    IBODY=ROBODY%MLNKH2B(JOBS)+(JBODY-1)
    IVNM =ROBODY%DATA(IBODY,ROBODY%VARNO_AT_BODY)
    
    ! Make sure we only look at cloud radar reflectivity
    IF ( IVNM /= VARNO%CLOUD_RADAR_REFLECTIVITY ) CYCLE

    ILEV=ROBODY%DATA(IBODY,ROBODY%VERTCO_REFERENCE_1_AT_BODY)

    ! Observed cloud radar reflectivity
    YD_RAD_CLREF(JOBS) % OBS(ILEV) = ROBODY%DATA(IBODY,ROBODY%OBSVALUE_AT_BODY)

    ! Observation error
    YD_RAD_CLREF(JOBS) % OERR(ILEV) = ROBODY%DATA(IBODY,ROBODY%FINAL_OBS_ERROR_AT_ERRSTAT)

    ! Status flag
    YD_RAD_CLREF(JOBS) % SCREEN(ILEV) = SATBODY%DATA(IBODY,SATBODY%DATUM_CLREFL_STFLAG_AT_CLRAD_B


    ! Check for passive monitoring data and flag 
    CALL NGEDSTA(INT(ROBODY%DATA(IBODY,ROBODY%DATUM_STATUS_AT_BODY)),INSTA,IDSTA)
```

B: /var/tmp/tmpdir/pan/git/ifs-source/pan_CY46R1_4DVar_CloudSat_Calipso_B/ifs/clradlid/clradlid_get_tl.F90

Top line 67 Encoding: UTF-8 Line end style: Unix

```
! Cloud radar reflectivity

DO JOBS=1,KLEN

  ! Initialise
  YD_RAD_CLREF (JOBS) % SCREEN(:) = IBSET(0,NRAD_CLREF_NO_OBS)
  YD_RAD_CLREF (JOBS) % OBS(:)    = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % FW  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % TL  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % AD  (:)   = 0.0_JPRB
  YD_RAD_CLREF (JOBS) % OERR(:)   = 999.0_JPRB
  
  ! Grid point index and PE
  YD_RAD_CLREF (JOBS) % JROF = SATHDR%HDR%GP_NUMBER(JOBS)
  YD_RAD_CLREF (JOBS) % PROC = SATHDR%HDR%DISTRIBID(JOBS)

  ! GOM-array mapping
  YD_RAD_CLREF (JOBS) % MAPOMM = ROBHDR%INDEX%MAPOMM(JOBS)

  ! Satellite time, lat, lon of obs
  YD_RAD_CLREF (JOBS) % OBTIME  = ROBHDR%HDR%TIME(JOBS)
  YD_RAD_CLREF (JOBS) % OBDATE  = ROBHDR%HDR%DATE(JOBS)
  YD_RAD_CLREF (JOBS) % LON     = ROBHDR%HDR%LON(JOBS) * RDEGREES
  YD_RAD_CLREF (JOBS) % LAT     = ROBHDR%HDR%LAT(JOBS) * RDEGREES

  ICMBDY=ROBODY%REPORT_END_ROW(JOBS)+1-ROBODY%REPORT_START_ROW(JOBS)

  DO JBODY=1,ICMBDY 

    IBODY=ROBODY%REPORT_START_ROW(JOBS)+(JBODY-1)
    IVNM =ROBODY%BODY%VARNO(IBODY)

    ! Make sure we only look at cloud radar reflectivity
    IF ( IVNM /= VARNO%CLOUD_RADAR_REFLECTIVITY ) CYCLE

    ILEV=ROBODY%BODY%VERTCO_REFERENCE_1(IBODY)

    ! Observed cloud radar reflectivity
    YD_RAD_CLREF(JOBS) % OBS(ILEV) = ROBODY%BODY%OBSVALUE(IBODY)

    ! Observation error
    YD_RAD_CLREF(JOBS) % OERR(ILEV) = ROBODY%ERRSTAT%FINAL_OBS_ERROR(IBODY)

    ! Status flag
    YD_RAD_CLREF(JOBS) % SCREEN(ILEV) &
     & = SATBODY%CLRAD_BODY%DATUM_CLREFL_STFLAG(IBODY)

    ! Check for passive monitoring data and flag 
    CALL NGEDSTA(INT(ROBODY%BODY%DATUM_STATUS(IBODY)),INSTA,IDSTA)
```

Figure 3.3: Same as for Fig. 3.2, but for the new routine, clradlid_gettl.F90, used to get cloud radar (reflectivity) and lidar (backscatter) observations from ODB to grid-point-space.



```
! the query 'obsort_hdr2resat_ave
  ! Note: SATHDR now contains the
  CALL alloc_hdr2body(NROWS_ROBSU,

  DO JLOOP=1,NROWS_ROBSU
    ILEN = MLNKH2B(JLOOP+1) - MLNK
    ROBSU(JLOOP, MLNK_HDR2BODY(1))
    ROBSU(JLOOP, MLNK_HDR2BODY(2))
    ROBSU(JLOOP, MLNK_HDR2ERRSTAT(
    ROBSU(JLOOP, MLNK_HDR2ERRSTAT(
    ROBSU(JLOOP, MLNK_HDR2SAT(1))
!   ROBSU(JLOOP, MLNK_HDR2SAT(2))
    ROBSU(JLOOP, MLNK_HDR2SUPEROB(
!    IF(ROBSU(JLOOP,MLNK_HDR2CLRAD
    ROBSU(JLOOP, MLNK_HDR2SUPEROB(
 !   ELSE
 !     ROBSU(JLOOP, MLNK_HDR2SUPER
 !   ENDIF
  ENDDO

  imxupd = ODB_getval(ihandle, '$r
  DO jup=1,imxupd
    DO JLOOP=1,NROWS_ROBSU
      ILEN = MLNKH2B(JLOOP+1) - MLNKH2B(JLOOP)
      ROBSU(jloop, MLNK_HDR2UPDATE(1,jup)) = 0
```

A: /var/tmp/tmpdir/pan/git/ifs-source/pan_CY46R1_4DVar_CloudSat_Calipso_B/odb/bufr2odb/b2o_convert_cllid.F90

```
!call b2o_reserve(handle, handle%subsets*2, subsets=1)
call b2o_reserve(handle,spread(1,1,nlev*2))

hdr => b2o_use_table(handle, "hdr")
sat => b2o_use_table(handle, "sat")
body => b2o_use_table(handle, "body")
errstat => b2o_use_table(handle, "errstat")
superob => b2o_use_table(handle, "superob")

iobs = 0
jobs = 0

  statid = ' '
  satid = b2o_get_int(handle, "satelliteIdentifier")
  write(statid,'(i8)') satid

  iobs = 1

  !hdr(iobs,hdr_retrtype) = 0             ! no averaging kernels
  hdr(iobs,hdr_date) = b2o_get_date(handle)
  hdr(iobs,hdr_time) = b2o_get_time(handle)
  hdr(iobs,hdr_statid) = transfer(statid,to_double)
  hdr(iobs,hdr_lat) = b2o_get_lat(handle)
  hdr(iobs,hdr_lon) = b2o_get_lon(handle)

  hdr(iobs,hdr_numlev) = nlev*2
  hdr(iobs,hdr_sensor) = b2o_get_int(handle, "satelliteInstruments")
  hdr(iobs,hdr_distribtype) = 2  ! call grid_nearest but not redistribte on model grid

  hdr(iobs,hdr_instrument_type) = b2o_get_real(handle, "satelliteInstruments")
  !resat(iobs,idx_resat%l_surface_type_indicator) = b2o_get_real(handle, 008029)
```

Several weeks of online debugging !!!

Figure 3.4: New way of working - challenging several weeks of online debugging of the development moved from CY43R1 to CY46R1.

3.3 Testing and debugging of the development in CY46R1

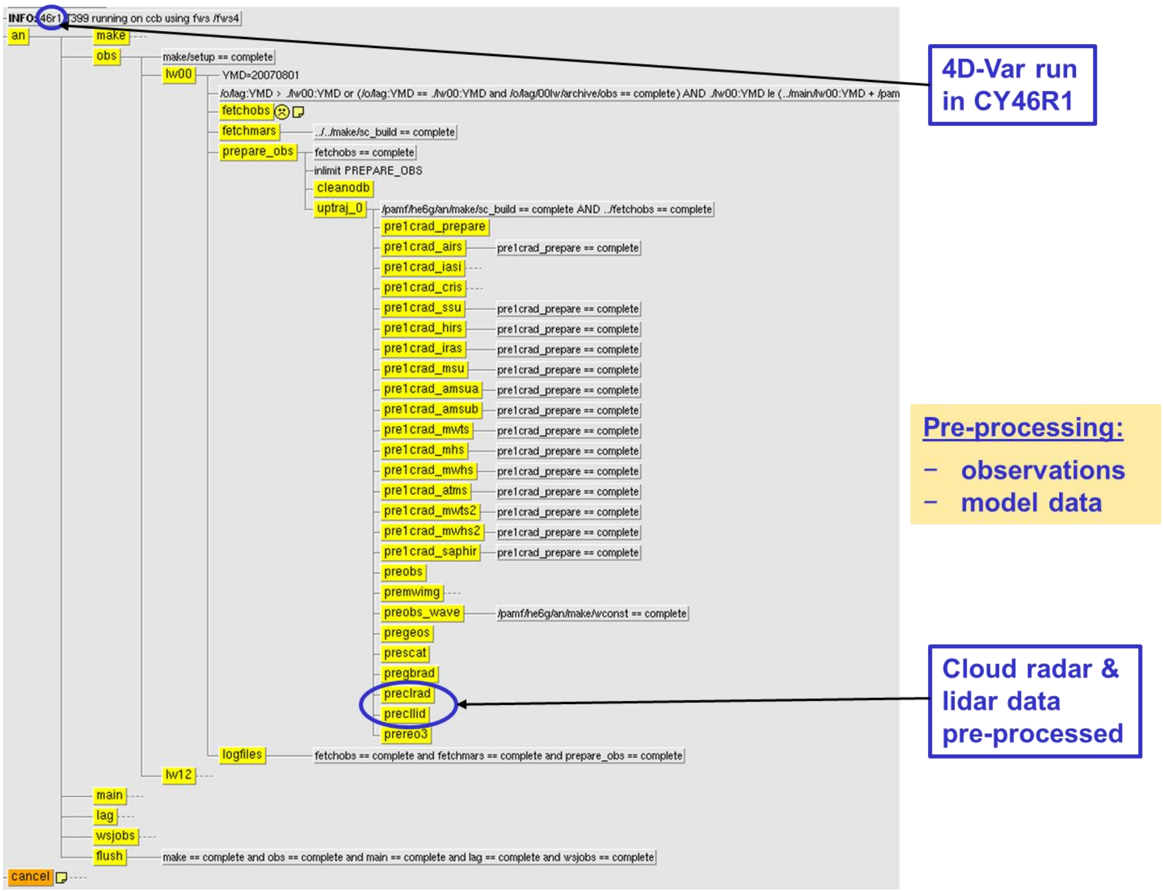
In this section we detail the testing of all the developments for cloud radar and lidar assimilation after updating it to the IFS model cycle CY46R1.

3.3.1 Technical tests

Technical testing is required to ensure that the new observations are recognized correctly and are fed through to the ODB properly. The observation equivalents generated by the observation operators and their passing to the ODB must also be verified. Only once all the data stored in the ODB are known to be correct, a basic test of the 4D-Var minimization can be made, where the convergence of the system with the new observations and operators can be checked.

3.3.1.1 Pre-processing observations

The first phase for preparation of observations to be used in the data assimilation system is the processing required before a normalised first guess departure can be computed. All satellite data goes through the long-established path of BUFR preprocessing (the **prepare_obs** tasks, Fig. 3.5) and BUFR to ODB conversion (the **bufr2odb**, Fig. 3.6 tasks). In this area, data undergo some elementary quality controls, such as a check for the observation format and position, and for the climatological limits.


INFO: 46r1 T399 running on ccb using fws /fws4
an
make
obs
make/setup == complete
lw00
YMD=20070801
/o/lag:YMD > ./lw00:YMD or (/o/lag:YMD == ./lw00:YMD and /o/lag/00lw/archive/obs == complete) AND ./lw00:YMD le (../main/lw00:YMD + /pam
fetchobs
fetchmars
../../make/sc_build == complete
prepare_obs
fetchobs == complete
inlimit PREPARE_OBS
cleanodb
uptraj_0
/pamf/he6g/an/make/sc_build == complete AND ../fetchobs == complete
pre1crad_prepare
pre1crad_airs
pre1crad_prepare == complete
pre1crad_iasi
pre1crad_cris
pre1crad_ssu
pre1crad_prepare == complete
pre1crad_hirs
pre1crad_prepare == complete
pre1crad_iras
pre1crad_prepare == complete
pre1crad_msu
pre1crad_prepare == complete
pre1crad_amsua
pre1crad_prepare == complete
pre1crad_amsub
pre1crad_prepare == complete
pre1crad_mwts
pre1crad_prepare == complete
pre1crad_mhs
pre1crad_prepare == complete
pre1crad_mwhs
pre1crad_prepare == complete
pre1crad_atms
pre1crad_prepare == complete
pre1crad_mwts2
pre1crad_prepare == complete
pre1crad_mwhs2
pre1crad_prepare == complete
pre1crad_saphir
pre1crad_prepare == complete
preobs
premwimg
preobs_wave
/pamf/he6g/an/make/wconst == complete
pregeos
prescat
pregbrad
preclrad
precllid
prereo3
logfiles
fetchobs == complete and fetchmars == complete and prepare_obs == complete
lw12
main
lag
wsjobs
flush
make == complete and obs == complete and main == complete and lag == complete and wsjobs == complete
cancel

4D-Var run in CY46R1

Pre-processing:
- observations
- model data

Cloud radar & lidar data pre-processed

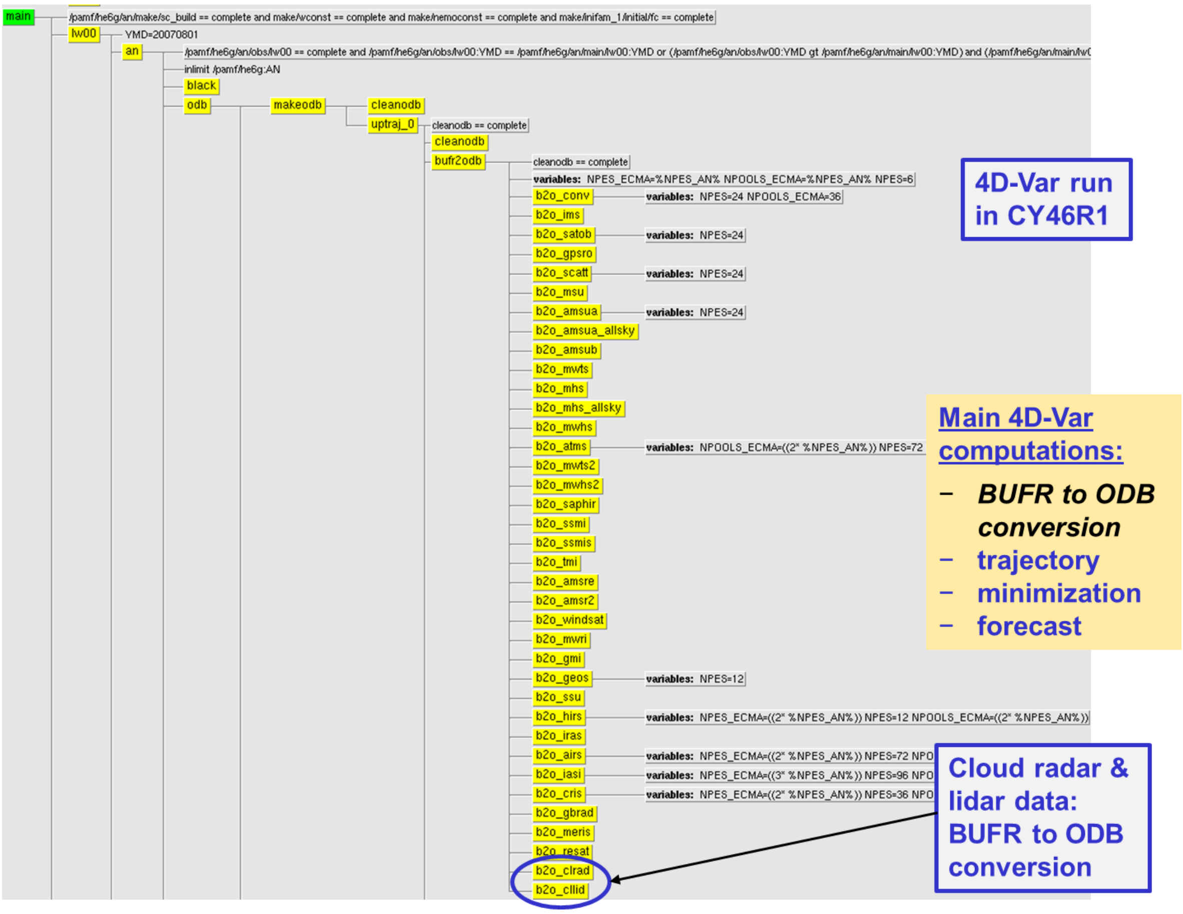
Figure 3.5: Observation pre-processing tasks in 4D-Var cycling of the IFS - tasks **preclrad** *and* **precllid** *(marked in blue) perform some pre-screening for cloud radar and lidar observations, respectively.*

A pre-screening process (tasks **preclrad** and **precllid**), which the new cloud radar and lidar observation must go through, has been successfully moved to CY46R1. This is demonstrated by yellow boxes in Fig. 3.5 indicating that those tasks has been successfully completed (note: failed task would be red). The performed pre-screening is used to reduce the data volume (thus the computational cost) of the main screening and also to reject observations that fail to contain crucial information, such as missing observed observed value, observation error or vertical coordinate of observations. The so-called superobbing is also performed at this stage, when observations are placed onto some grid with resolution depending on the level of desired smoothing. At this point, vertical matching with the model vertical resolution is done as well.

After creating the superobs, additional pre-screening tasks are carried out. They include threshold checks on: the number and the standard deviation of the observations forming the superob, the mean observation value itself and the observed cloud fraction. For radar, a check for multiple scattering is also done by applying a threshold check on the integrated radar reflectivity.

The pre-processed data for cloud radar and lidar observations, still in BUFR format, are converted to ODB in the **b2o_clrad** and **b2ocllid** tasks, respectively (Fig. 3.6). This involves one-to-one mapping of superobs contained within the temporary BUFR file to the ODB variables.

To complete the whole observation pre-screening requires successful completion of the above described tasks. This was achieved for the CY46R1 model cycle.



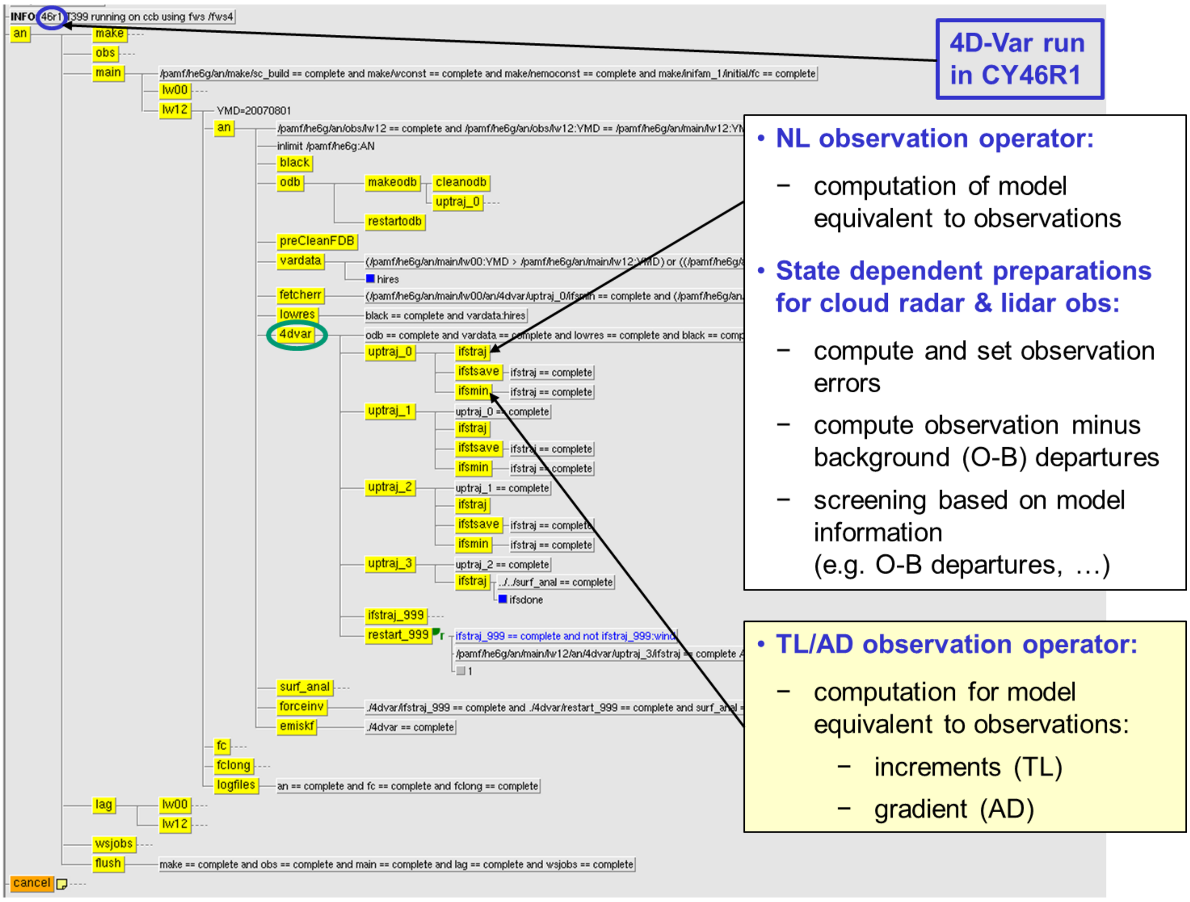
```
main ── /pamf/he6g/an/make/sc_build == complete and make/wconst == complete and make/nemoconst == complete and make/inifam_1/initial/fc == complete
  lw00 ── YMD=20070801
    an ── /pamf/he6g/an/obs/lw00 == complete and /pamf/he6g/an/obs/lw00:YMD == /pamf/he6g/an/main/lw00:YMD or (/pamf/he6g/an/obs/lw00:YMD gt /pamf/he6g/an/main/lw00:YMD) and (/pamf/he6g/an/main/lw0
       inlimit /pamf/he6g:AN
       black
       odb ── makeodb ── cleanodb
                         uptraj_0 ── cleanodb == complete
                                     cleanodb
                                     bufr2odb ── cleanodb == complete
                                                 variables: NPES_ECMA=%NPES_AN% NPOOLS_ECMA=%NPES_AN% NPES=6
                                                 b2o_conv ──── variables: NPES=24 NPOOLS_ECMA=36
                                                 b2o_ims
                                                 b2o_satob ─── variables: NPES=24
                                                 b2o_gpsro
                                                 b2o_scatt ─── variables: NPES=24
                                                 b2o_msu
                                                 b2o_amsua ─── variables: NPES=24
                                                 b2o_amsua_allsky
                                                 b2o_amsub
                                                 b2o_mwts
                                                 b2o_mhs
                                                 b2o_mhs_allsky
                                                 b2o_mwhs
                                                 b2o_atms ──── variables: NPOOLS_ECMA=((2* %NPES_AN%)) NPES=72
                                                 b2o_mwts2
                                                 b2o_mwhs2
                                                 b2o_saphir
                                                 b2o_ssmi
                                                 b2o_ssmis
                                                 b2o_tmi
                                                 b2o_amsre
                                                 b2o_amsr2
                                                 b2o_windsat
                                                 b2o_mwri
                                                 b2o_gmi
                                                 b2o_geos ──── variables: NPES=12
                                                 b2o_ssu
                                                 b2o_hirs ──── variables: NPES_ECMA=((2* %NPES_AN%)) NPES=12 NPOOLS_ECMA=((2* %NPES_AN%))
                                                 b2o_iras
                                                 b2o_airs ──── variables: NPES_ECMA=((2* %NPES_AN%)) NPES=72 NPO
                                                 b2o_iasi ──── variables: NPES_ECMA=((3* %NPES_AN%)) NPES=96 NPO
                                                 b2o_cris ──── variables: NPES_ECMA=((2* %NPES_AN%)) NPES=36 NPO
                                                 b2o_gbrad
                                                 b2o_meris
                                                 b2o_resat
                                                 b2o_clrad
                                                 b2o_cllid
```

Figure 3.6: Conversion of BUFR to ODB in the pre-processing stage of 4D-Var cycling tasks - in **b2o_clrad** *and* **b2o_cllid** *(marked in blue), the pre-processed data are converted to ODB for cloud radar and lidar observations, respectively.*

3.3.1.2 Main 4D-Var computations

The group of main 4D-Var tasks (Fig. 3.7) starts with reading observations into 4D-Var from the Observation Data Base (ODB). This is done by the **odb** tasks of the **main** task group. The sequence of jobs involved in the minimization system starts with the first (high resolution) trajectory run, **ifstraj**. During this run the model counterparts for all observations are calculated through the non-linear observation operators. The observation minus model differences (departures) are then computed and stored in the ODB. Since quality control decisions depend on the magnitude of the departure, these computed departures are an important input to the data selection procedures. During such screening procedure, the best quality observations are selected, duplicates are detected and data redundancy is reduced through thinning. At this stage, any observations with too large departures (i.e. exceeding the predefined limits) are rejected.

In the next stage, blacklisting procedure selects which observation types, variables, vertical ranges etc.



4D-Var run in CY46R1

- **NL observation operator:**
  - computation of model equivalent to observations
- **State dependent preparations for cloud radar & lidar obs:**
  - compute and set observation errors
  - compute observation minus background (O-B) departures
  - screening based on model information (e.g. O-B departures, …)

- **TL/AD observation operator:**
  - computation for model equivalent to observations:
    - increments (TL)
    - gradient (AD)

Figure 3.7: Tasks of 4D-Var cycling as used by the IFS at ECMWF - main 4D-Var computations.

will be used for the assimilation. The first type of decisions made to use or not use a particular observations is an a priori type decision which takes no account of the actual value of the observations. The second type of test is based on the observed values (or observation departure from the background). Blacklist-rejected data are subsequently excluded and will not be present in the minimisation job steps.

After performing screening, the original ODB is complemented by the background departures together with quality information for observations. The compressed ODB, the CCMA, contains only those observations that passed screening and that are to be used in the minimization (**ifsmin** tasks).

To complete successfully the 4D-Var minimization process in the model cycle 46R1 required substantial effort and debugging since it required to complete many rather complex tasks in the **4dvar** task group, such as:

- The so called ‘screening run’ described above is done in the first trajectory run (**ifstraj** in **uptraj_0**) which includes a model integration, comparison to observations, and observation screening (quality control and data selection).
- The first minimization (**ifsmin** in **uptraj_0**) typically runs at low resolution. This step includes estimation of analysis and forecast error variances, and calculation of Hessian eigenvectors for pre-conditioning of subsequent minimisation(s).
- The first trajectory update (**ifstraj** in **uptraj_1**) applies the analysis increments obtained in the first

minimisations and performs another forecast integration with comparison to observations. This provides a new linearisation state for the next minimisation.

- The second (**ifsmin** in **uptraj_1** and subsequent in **uptraj_2**) minimization uses higher resolution increments.
- The final trajectory runs (**ifstraj** in **uptraj_3**) carries out verification screening, which involves a comparison between the observations and the final analysis. In the final trajectory the final analysis is formed by adding the low-resolution increment to the background (at initial time), and integrating to the analysis times.

Yellow boxes in Fig. 3.7 indicate that we were able successfully to accomplished all the tasks in the **4dvar** task group.

Figure 3.8 provides a summary of how observation cost function for cloud radar (CRFL) and cloud lidar (CBSC) observations is decreased by progressing iteratively through the set of outer and inner loops. It demonstrates that we managed to move all developments from the model cycle CY43R1 used at the previous ESA project (Janisková and Fielding, 2018) and succeeded to make it working in more recent model cycle 46R1. For debugging and first technical testing we used a lower model resolution of T399 for trajectory computation (outer loops) and the model forecast. The resolution of inner loops is increased progressively: T95 in **uptraj_0**, T159 in **uptraj_1** and T255 in **uptraj_2**. In the provided output, the observation cost function (when looking at value of observation cost function, Jo, per number of observations, n, i.e. Jo/n) is decreased from the value of 0.84 in the first **ifstraj** run in **uptraj_0** to the value of 0.41 in the final trajectory run in **uptraj_3**. One can also see that the observation cost function is decreased more for cloud radar reflectivity than for cloud lidar backscatter. However, overall the results in Fig. 3.8 indicates that the analysis is getting closer to cloud radar and lidar observations as the minimization is progressing. That is the important technical sanity check before we can progress with more scientific evaluation and subsequent improvements of cloud radar and lidar assimilation.

3.3.1.3 Post-processing of observations

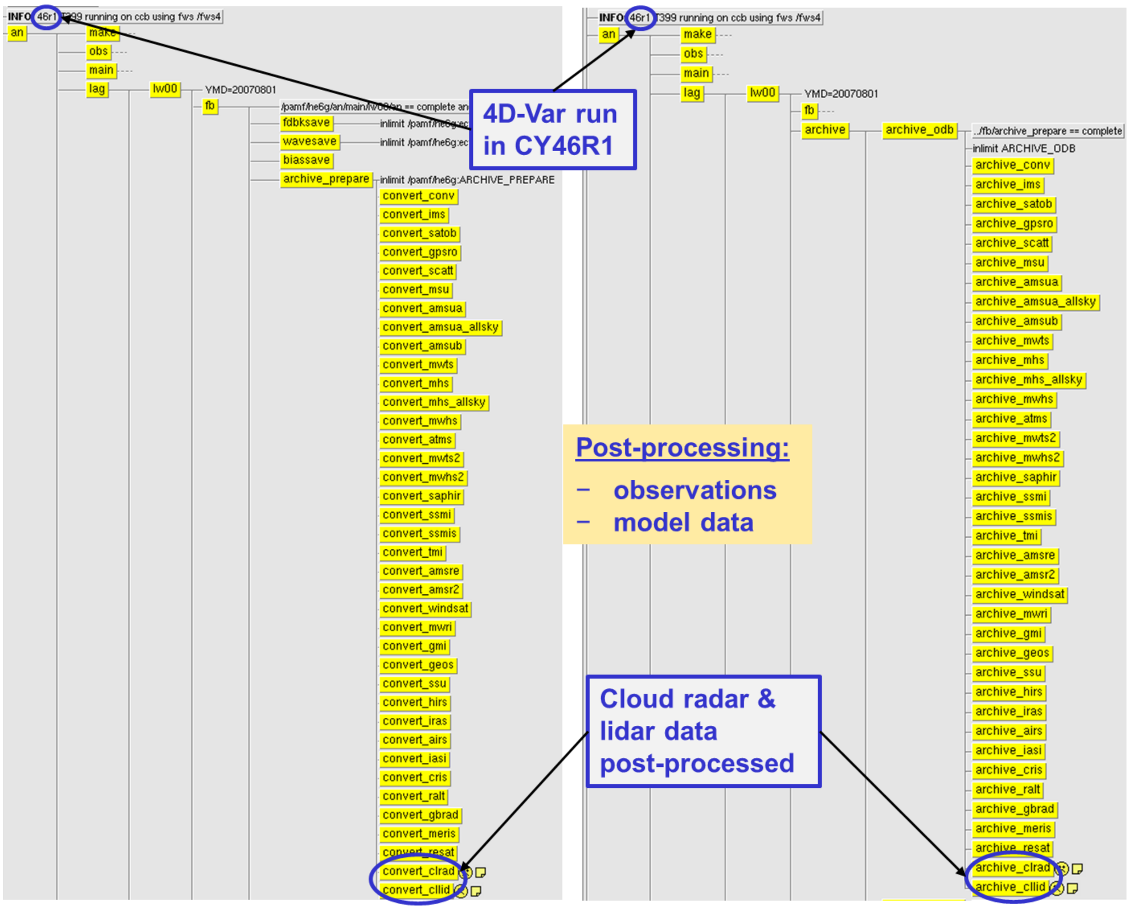
Post-processing of observations is done in two stages (Fig. 3.9) after completing all the tasks in the **main** 4D-Var calculations, when the high-resolution analysis is produced. First, an ODB file is created from ECMA containing information about all observations, not only the active ones. This is done in the task group

archive_prepare by the new tasks **convert_clrad** and **convert_cllid** for cloud radar and lidar observations, respectively. Created ODB files are then archived in MARS (Meteorological Archiving and Retrieval System of ECMWF) by the tasks **archive_clrad** and **archive_cllid**. All these task have been moved to cycle CY46R1 and ODB files were properly created.

The successful post-processing for cloud radar and lidar observations in CY46R1 is also displayed in Fig. 3.10. The listing of output files from 4D-Var calculation (Fig. 3.10 left) shows that databases ECMA for cloud radar and lidar observations were created. Obviously, since we managed to complete the full data assimilation cycle, ECMAs had to be properly recognized and updated in the different stages of data assimilation process. Figure 3.10 (right) illustrates that the created ECMAs are recognized by the trajectory run (**ifstraj**).

```
uptraj_0 - ifstraj
  Diagnostic JO-table (JOT) SCREENING JOB    T0399 NCONF=     1 NSIM4D=     0 NUPTRA=     0
  =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              78754            15645.07045080          0.20     0.179E+02   0.000E+00
            CRFL             168744            192881.4860162          1.14     0.109E+02   0.000E+00
                          ----------   ---------------------------   --------
     ObsType 19 Total:       247498            208526.5564670          0.84

uptraj_0 - ifsmin
  Diagnostic JO-table (JOT) MINIMISATION JOB T0095 NCONF=   131 NSIM4D=     0 NUPTRA=     0
  =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              51336            7328.207881755          0.14     0.133E+02   0.100E-02
            CRFL             119352            76921.64561061          0.64     0.103E+02   0.300E+01
                           ----------   ---------------------------   --------
     ObsType 19 Total:       170688            84249.85349236          0.49

uptraj_1 - ifstraj
  Diagnostic JO-table (JOT) TRAJECTORY JOB   T0399 NCONF=     1 NSIM4D=     0 NUPTRA=     1
  =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              45803            6426.823595215          0.14     0.129E+02   0.000E+00
            CRFL             116019            69893.36911001          0.60     0.104E+02   0.000E+00
                           ----------   ---------------------------   --------
     ObsType 19 Total:       161822            76320.19270523          0.47

uptraj_1 - ifsmin
  Diagnostic JO-table (JOT) MINIMISATION JOB T0159 NCONF=   131 NSIM4D=     0 NUPTRA=     1
  =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              45803            6430.957912019          0.14     0.129E+02   0.100E-02
            CRFL             116019            69835.08107902          0.60     0.104E+02   0.300E+01
                           ----------   ---------------------------   --------
     ObsType 19 Total:       161822            76266.03899104          0.47

uptraj_2 - ifstraj
  Diagnostic JO-table (JOT) TRAJECTORY JOB   T0399 NCONF=     1 NSIM4D=     0 NUPTRA=     2
  =========================================================================================

     Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              41450            5843.686899308          0.14     0.124E+02   0.000E+00
            CRFL             112023            62869.21872256          0.56     0.104E+02   0.000E+00
                           ----------   ---------------------------   --------
     ObsType 19 Total:       153473            68712.90562187          0.45

uptraj_2 - ifsmin
  Diagnostic JO-table (JOT) MINIMISATION JOB T0255 NCONF=   131 NSIM4D=     0 NUPTRA=     2
   =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              41450            5847.698420813          0.14     0.124E+02   0.100E-02
            CRFL             112023            63152.50061440          0.56     0.104E+02   0.300E+01
                           ----------   ---------------------------   --------
     ObsType 19 Total:       153473            69000.19903522          0.45

uptraj_3 - ifstraj
  Diagnostic JO-table (JOT) TRAJECTORY JOB   T0399 NCONF=     1 NSIM4D=     0 NUPTRA=     3
   =========================================================================================

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      --------------------------------------------------
        Codetype   197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction         JO/n       ObsErr      BgErr
            CBSC              38690            5575.855335052          0.14     0.122E+02   0.000E+00
            CRFL             108517            54765.63456560          0.50     0.104E+02   0.000E+00
                           ----------   ---------------------------   --------
     ObsType 19 Total:       147207            60341.48990065          0.41
```

Figure 3.8: Observation cost function Jo for cloud radar reflectivity (CRFL) and cloud lidar backscatter (CBSC) for one of 4D-Var assimilation cycle. The output provides information on number of observations used (DataCount), the value of cost function (Jo_Costfunction), the value of cost function per observation (JO/n), as well as observation (ObsErr) and background (BgErr) errors for trajectory **ifstraj** *and minimization* **ifsmin** *computations for all 4D-Var loops* **uptraj**. *The model resolution is T399 for outer loops and varies for inner loops: T95 in the 1st loop, T159 in the 2nd and T255 in the 3rd loop.*



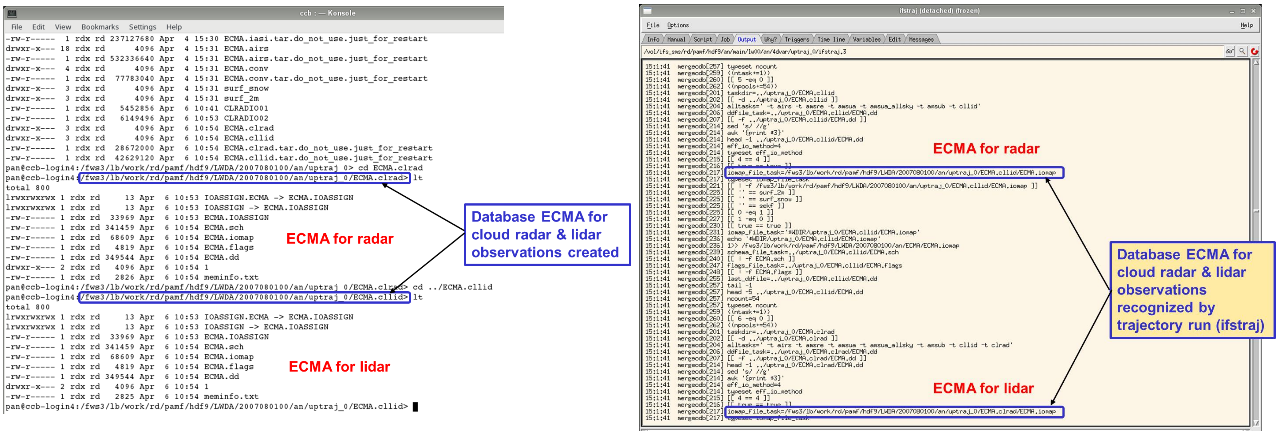
4D-Var run in CY46R1

Post-processing:

- observations
- model data

Cloud radar & lidar data post-processed

Figure 3.9: Post-processing tasks in 4D-Var cycling: (left) creation of ODB database with direct column access under **archive_prepare** *task group - tasks* **convert_clrad** *and* **convertcllid** *(marked in blue) used for the new cloud radar and lidar observations; (right) archiving ODB database under* **archive_odb** *task group - tasks* **archive_clrad** *and* **archive_cllid** *(marked in blue) for the new observations.*



```
-rw-r-----  1 rdx rd 237127680 Apr  4 15:30 ECMA.iasi.tar.do_not_use.just_for_restart
drwxr-x--- 18 rdx rd      4096 Apr  4 15:31 ECMA.airs
-rw-r-----  1 rdx rd 532336640 Apr  4 15:31 ECMA.airs.tar.do_not_use.just_for_restart
drwxr-x---  4 rdx rd      4096 Apr  4 15:31 ECMA.conv
-rw-r-----  1 rdx rd  77783040 Apr  4 15:31 ECMA.conv.tar.do_not_use.just_for_restart
drwxr-x---  3 rdx rd      4096 Apr  4 15:31 surf_snow
drwxr-x---  3 rdx rd      4096 Apr  4 15:31 surf_2m
-rw-r-----  1 rdx rd   5452856 Apr  6 10:41 CLRADIO01
-rw-r-----  1 rdx rd   6149496 Apr  6 10:53 CLRADIO02
drwxr-x---  3 rdx rd      4096 Apr  6 10:54 ECMA.clrad
drwxr-x---  3 rdx rd      4096 Apr  6 10:54 ECMA.cllid
-rw-r-----  1 rdx rd  28672000 Apr  6 10:54 ECMA.clrad.tar.do_not_use.just_for_restart
-rw-r-----  1 rdx rd  42629120 Apr  6 10:54 ECMA.cllid.tar.do_not_use.just_for_restart
pan@ccb-login4:/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0> cd ECMA.clrad
pan@ccb-login4:/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.clrad> lt
total 800
lrwxrwxrwx 1 rdx rd     13 Apr  6 10:53 IOASSIGN.ECMA -> ECMA.IOASSIGN
lrwxrwxrwx 1 rdx rd     13 Apr  6 10:53 IOASSIGN -> ECMA.IOASSIGN
-rw-r----- 1 rdx rd  33969 Apr  6 10:53 ECMA.IOASSIGN
-rw-r----- 1 rdx rd 341459 Apr  6 10:54 ECMA.sch
-rw-r----- 1 rdx rd  68609 Apr  6 10:54 ECMA.iomap
-rw-r----- 1 rdx rd   4819 Apr  6 10:54 ECMA.flags
-rw-r----- 1 rdx rd 349544 Apr  6 10:54 ECMA.dd
drwxr-x--- 2 rdx rd   4096 Apr  6 10:54 1
-rw-r----- 1 rdx rd   2826 Apr  6 10:54 meminfo.txt
pan@ccb-login4:/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.clrad> cd ../ECMA.cllid
pan@ccb-login4:/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.cllid> lt
total 800
lrwxrwxrwx 1 rdx rd     13 Apr  6 10:53 IOASSIGN.ECMA -> ECMA.IOASSIGN
lrwxrwxrwx 1 rdx rd     13 Apr  6 10:53 IOASSIGN -> ECMA.IOASSIGN
-rw-r----- 1 rdx rd  33969 Apr  6 10:53 ECMA.IOASSIGN
-rw-r----- 1 rdx rd 341459 Apr  6 10:54 ECMA.sch
-rw-r----- 1 rdx rd  68609 Apr  6 10:54 ECMA.iomap
-rw-r----- 1 rdx rd   4819 Apr  6 10:54 ECMA.flags
-rw-r----- 1 rdx rd 349544 Apr  6 10:54 ECMA.dd
drwxr-x--- 2 rdx rd   4096 Apr  6 10:54 1
-rw-r----- 1 rdx rd   2825 Apr  6 10:54 meminfo.txt
pan@ccb-login4:/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.cllid> █
```

ECMA for radar

Database ECMA for cloud radar & lidar observations created

ECMA for lidar

ifstraj (detached) (frozen)
File Options Help
Info Manual Script Job Output Why? Triggers Time line Variables Edit Messages
/vol/ifs_sms/rd/pamf/hdf9/an/main/lw00/an/4dvar/uptraj_0/ifstraj.3

```
15:1:41  mergeodb[257] typeset ncount
15:1:41  mergeodb[259] ((ntask+=1))
15:1:41  mergeodb[260] [[ 5 -eq 0 ]]
15:1:41  mergeodb[262] ((npools+=54))
15:1:41  mergeodb[201] taskdir=../uptraj_0/ECMA.cllid
15:1:41  mergeodb[202] [[ -d ../uptraj_0/ECMA.cllid ]]
15:1:41  mergeodb[204] alltasks=' -t airs -t amsre -t amsua -t amsua_allsky -t amsub -t cllid'
15:1:41  mergeodb[206] ddfile_task=../uptraj_0/ECMA.cllid/ECMA.dd
15:1:41  mergeodb[207] [[ -f ../uptraj_0/ECMA.cllid/ECMA.dd ]]
15:1:41  mergeodb[214] sed 's/ //g'
15:1:41  mergeodb[214] awk '{print $3}'
15:1:41  mergeodb[214] head -1 ../uptraj_0/ECMA.cllid/ECMA.dd
15:1:41  mergeodb[214] eff_io_method=4
15:1:41  mergeodb[214] typeset eff_io_method
15:1:41  mergeodb[215] [[ 4 == 4 ]]
15:1:41  mergeodb[216] [[ true == true ]]
15:1:41  mergeodb[217] iomap_file_task=/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.cllid/ECMA.iomap
15:1:41  mergeodb[217] typeset iomap_file_task
15:1:41  mergeodb[221] [[ ! -f /fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.cllid/ECMA.iomap ]]
15:1:41  mergeodb[225] [[ '' == surf_2m ]]
15:1:41  mergeodb[225] [[ '' == surf_snow ]]
15:1:41  mergeodb[225] [[ '' == sekf ]]
15:1:41  mergeodb[225] [[ 0 -eq 1 ]]
15:1:41  mergeodb[227] [[ 1 -eq 0 ]]
15:1:41  mergeodb[230] [[ true == true ]]
15:1:41  mergeodb[231] iomap_file_task='$WDIR/uptraj_0/ECMA.cllid/ECMA.iomap'
15:1:41  mergeodb[236] echo '$WDIR/uptraj_0/ECMA.cllid/ECMA.iomap'
15:1:41  mergeodb[236] 1>> /fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/ECMA/ECMA.iomap
15:1:41  mergeodb[239] schema_file_task=../uptraj_0/ECMA.cllid/ECMA.sch
15:1:41  mergeodb[240] [[ ! -f ECMA.sch ]]
15:1:41  mergeodb[247] flags_file_task=../uptraj_0/ECMA.cllid/ECMA.flags
15:1:41  mergeodb[248] [[ ! -f ECMA.flags ]]
15:1:41  mergeodb[255] last_ddfile=../uptraj_0/ECMA.cllid/ECMA.dd
15:1:41  mergeodb[257] tail -1
15:1:41  mergeodb[257] head -5 ../uptraj_0/ECMA.cllid/ECMA.dd
15:1:41  mergeodb[257] ncount=54
15:1:41  mergeodb[257] typeset ncount
15:1:41  mergeodb[259] ((ntask+=1))
15:1:41  mergeodb[260] [[ 6 -eq 0 ]]
15:1:41  mergeodb[262] ((npools+=54))
15:1:41  mergeodb[201] taskdir=../uptraj_0/ECMA.clrad
15:1:41  mergeodb[202] [[ -d ../uptraj_0/ECMA.clrad ]]
15:1:41  mergeodb[204] alltasks=' -t airs -t amsre -t amsua -t amsua_allsky -t amsub -t cllid -t clrad'
15:1:41  mergeodb[206] ddfile_task=../uptraj_0/ECMA.clrad/ECMA.dd
15:1:41  mergeodb[207] [[ -f ../uptraj_0/ECMA.clrad/ECMA.dd ]]
15:1:41  mergeodb[214] head -1 ../uptraj_0/ECMA.clrad/ECMA.dd
15:1:41  mergeodb[214] sed 's/ //g'
15:1:41  mergeodb[214] awk '{print $3}'
15:1:41  mergeodb[214] eff_io_method=4
15:1:41  mergeodb[214] typeset eff_io_method
15:1:41  mergeodb[215] [[ 4 == 4 ]]
15:1:41  mergeodb[216] [[ true == true ]]
15:1:41  mergeodb[217] iomap_file_task=/fws3/lb/work/rd/pamf/hdf9/LWDA/2007080100/an/uptraj_0/ECMA.clrad/ECMA.iomap
15:1:41  mergeodb[217] typeset iomap_file_task
```

ECMA for radar

Database ECMA for cloud radar & lidar observations recognized by trajectory run (ifstraj)

ECMA for lidar

Figure 3.10: Database ECMA for cloud radar and lidar observations: (left) successful creation and (right) recognition by the 4D-Var trajectory run **ifstraj***.*

3.3.2 First testing results of 4D-Var with cloud radar and lidar observations in CY46R1

After the technical tests showing that the cloud radar and lidar assimilation developed in the previous project (Janisková and Fielding, 2018), where it worked under the model cycle 43R1, was successfully moved to the CY46R1, the first basic performance evaluation was carried out. This was done by looking at along-track performance starting with just one assimilation cycle. An investigation of the system performance is demonstrated on the portion of orbital track representing several model profiles containing radar and lidar observations assimilated in the data assimilation window between 21:00 UTC 31 July 2007 and 09:00 UTC 1 August 2007 (Figs. 3.11 - 3.13).

Figure 3.11 indicates that the first-guess radar reflectivity generated from the model background is remarkably similar for CY43R1 and CY46R1. In both cycles it corresponds rather well to the superobbed CloudSat radar reflectivity. Observation errors, which take into account instrument, observation operator and representativity errors, are also similar for the both model cycles. As is expected, the analysis radar reflectivity provides a closer fit to the observations. Overall, there are a lot of similarities between analysed reflectivity values obtained in CY43R1 and CY46R1. The larger difference in performance for these two model cycles is observed in the convective regions, where it seems that slightly more observation data are used in CY46R1 than CY43R1. That could indicate that in CY46R1 we could expect that the analyses might be driven closer to the observations in such regions. This could be due to improvements in the parametrization of moist processes in both, the nonlinear and the linearized model (used for the minimization calculations in 4D-Var) present in CY46R1. 4D-Var experiments performed for longer period later in the project could show whether that is generally indeed the case.

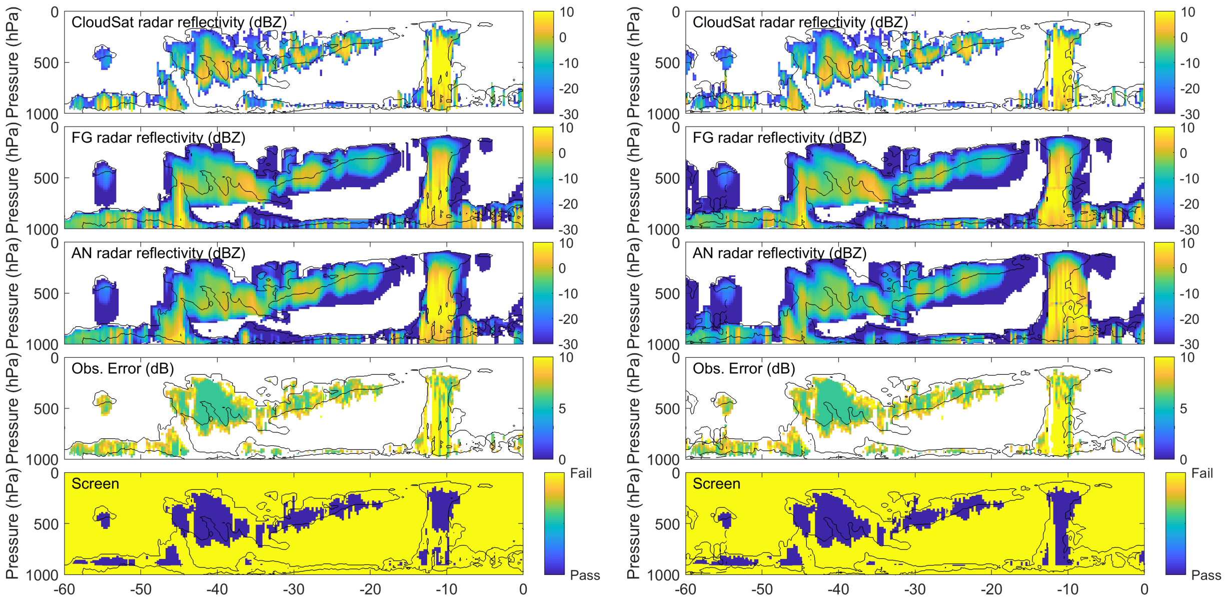


Figure 3.11: Cross-sections of cloud radar reflectivity related variables corresponding to the portion of orbital track on 31 July 2007 at 21:00 UTC. Panels show from top to bottom: observed CloudSat radar reflectivity (dBZ), model equivalent (FG) radar reflectivity using the model background (dBZ), model equivalent (AN) radar reflectivity using the model analysis from the assimilation experiment using all observations and radar and lidar, the observation error (dB) assigned to the observations and screening value (blue colour = observations passed). The cross-sections on the left panels correspond to outputs from the model cycle CY43R1, while the right panels are for the cycle CY46R1.

A visualisation of the cross-section for lidar backscatter is shown in Fig. 3.12. The initial fit of the first guess values to the observations is similar for the model cycles CY43R1 and CY46R1, comparably to radar reflectivity. The observation usage and analysed values of the attenuated lidar backscatter are very similar for both model cycles.

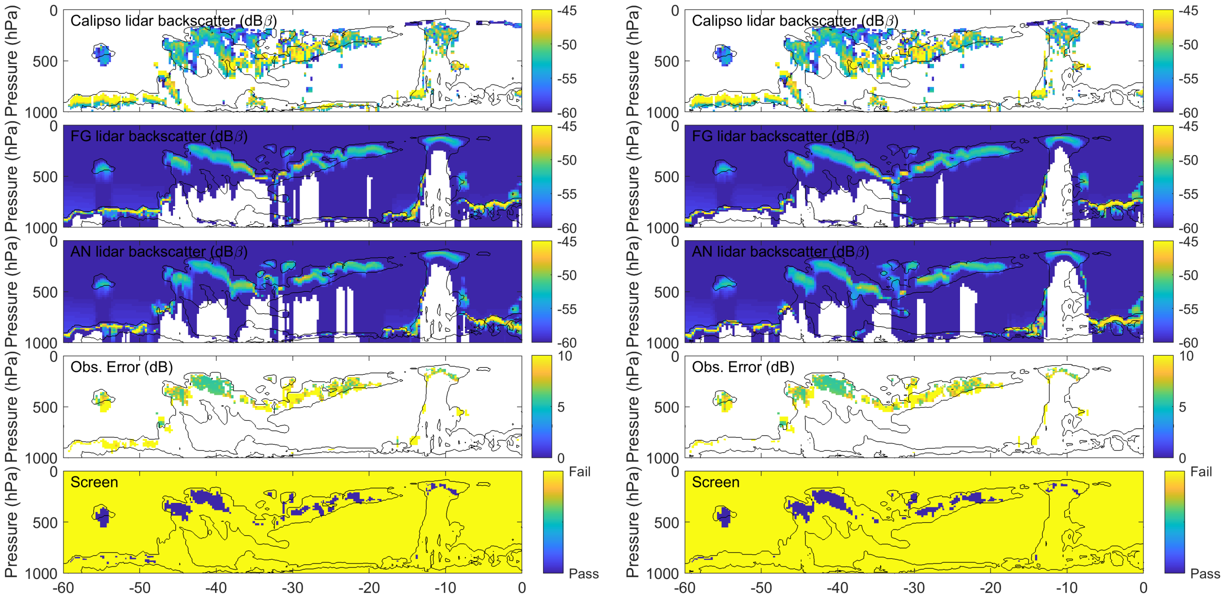


Figure 3.12: Same as Fig. 3.11, but for cloud lidar backscatter and CALIPSO attenuated backscatter (dBβ).

To show how the information from radar reflectivity and lidar backscatter gets converted to more tangible cloud related variables such as temperature and relative humidity, Fig. 3.13 compares analysis increments for the cycles CY43R1 and CY46R1. Comparing these two cycles, it is possible to see that there are a lot of similarities in obtained increments. For cloud radar reflectivity, large differences are obvious in the convective regions where more observations were used in CY46R1 than in CY43R1 as indicated in Fig. 3.11. The largest differences in the size of analysis increments are observed for relative humidity, when the relative humidity increments are generally large in CY46R1 that in CY43R1. This will be further investigated whether it comes from some changes in the model background statistics for humidity or it is related to the different observation usage for these model cycles.

Overall, the performance of 4D-Var system assimilating cloud radar and lidar observations on the top of regularly used observations is rather similar for the originally used model cycle 43R1 and the cycle 46R1, where the cloud radar and lidar development was moved. These first test results are very encouraging as they indicate that cloud radar and lidar assimilation was successfully ported to CY46R1.

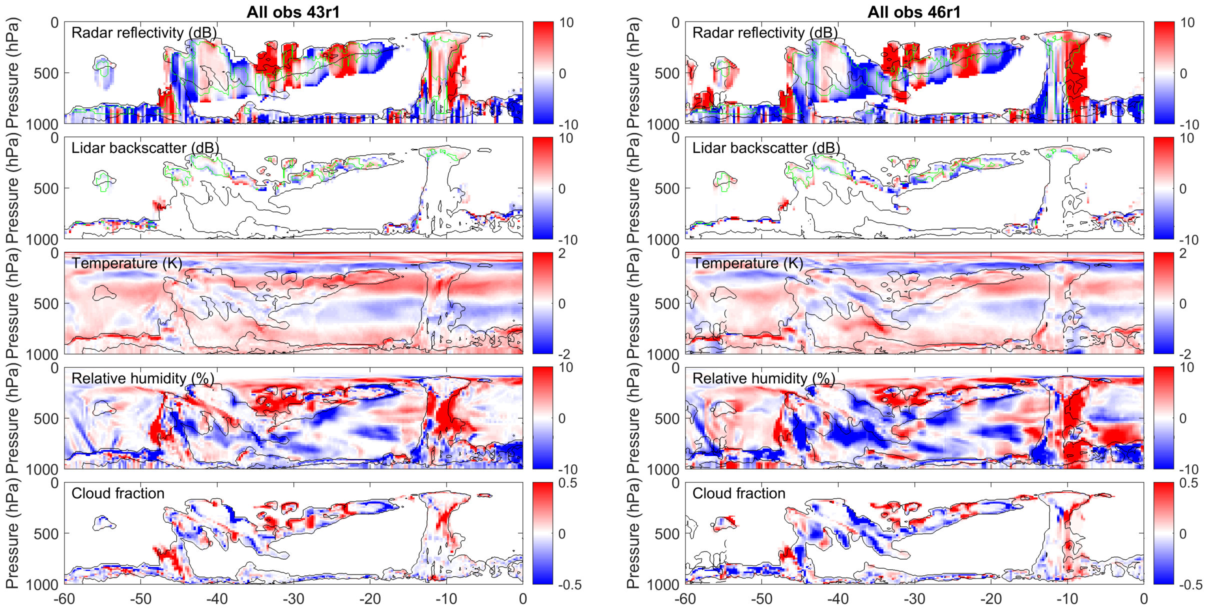


Figure 3.13: Analysis increments for the the portion of orbital track on 31 July 2007 at 21:00 UTC as in Fig. 3.11 and Fig. 3.12 for the model cycle CY43R1 (left column) and CY46R1 (right column). Panels from top to bottom show model-level increments of: radar reflectivity (dB), lidar backscatter (dB), temperature (K), relative humidity (%) and cloud fraction.

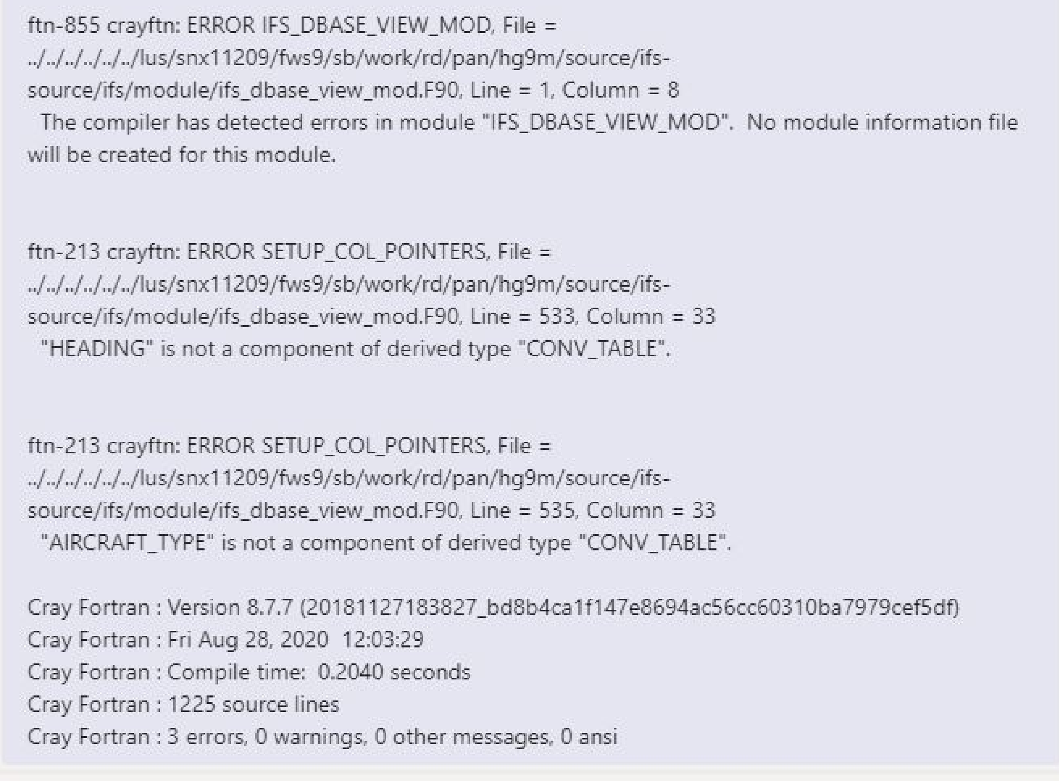
4 Migration of the assimilation system development for cloud radar and lidar observations - from CY46R1 to CY47R1

To keep track with the ECMWF model upgrades, the satellite cloud radar and lidar capacity of the 4D-Var assimilation system needs also to be regularly updated and tested to ensure its proper functionality. The first step to maintain that developed assimilation was to bring developments built under IFS model cycle CY43R1 at the previous project (Janisková and Fielding, 2018) to the more up-to-date model cycle CY46R1. That was achieved in the first 7-8 months of the project and reported in Section 3. Since then the IFS model cycle was father updated. Thus following those updates and as a preparation to move cloud radar and lidar assimilation system in the future operational code in 2022, our developments were migrated to the higher cycle, CY47R1. Obviously these developments will be passive at that stage, but such move will save us from porting them ourselves to the new model cycles which might be few per year. Although, the satellite cloud radar and lidar capacity of the 4D-Var assimilation system will need to be tested for each new cycle.

4.1 Porting cloud radar and lidar assimilation to the model cycle CY47R1

Section 3 summarized the challenging effort of moving over 300 routines and scripts across several model cycles, i.e. to bring the assimilation system from the model cycle CY43R1 to CY46R1. There were not only many changes in the routines which needed to be modified to take into account cloud radar and lidar observations, but also the new routines for these observations needed to be adapted to CY46R1. This huge task was even complicated by the fact that developing/version controlling software used by CY43R1, Perforce, was replaced by Git in CY46R1. Therefore all the porting had to be done manually, routine by routine. Successfully updating the cloud radar and lidar assimilation developments to CY46R1 was a big step securing all past development for the future.

In the next step, all that development was ported to the higher cycle, CY47R1, to bring the developments up to date with the current operational IFS model cycle (which became operational in June 2020). Although still rather complex, this porting was simplified by possibility to use Git software. Using the software, vast majority of routines and scripts could be merged automatically. Although there were few routines with conflicts that were required to be resolved manually. On top of that, due to the model and structural changes in CY47R1, additional routines and problems required to be fixed in order to make assimilation of cloud radar and lidar observations functional in CY47R1. There were several issues to be solved related to the definition of observation tables and observation types due to structural changes in observation handling. That lead to several compilation failures which required support of people from different ECMWF teams to be resolved. Figure 4.1 shows an example of such failure for routine defining observation types. Once all conflicting files were fixed, updated cycle CY47R1 was successfully compiled and ready for basic testing as described in Section 4.3.



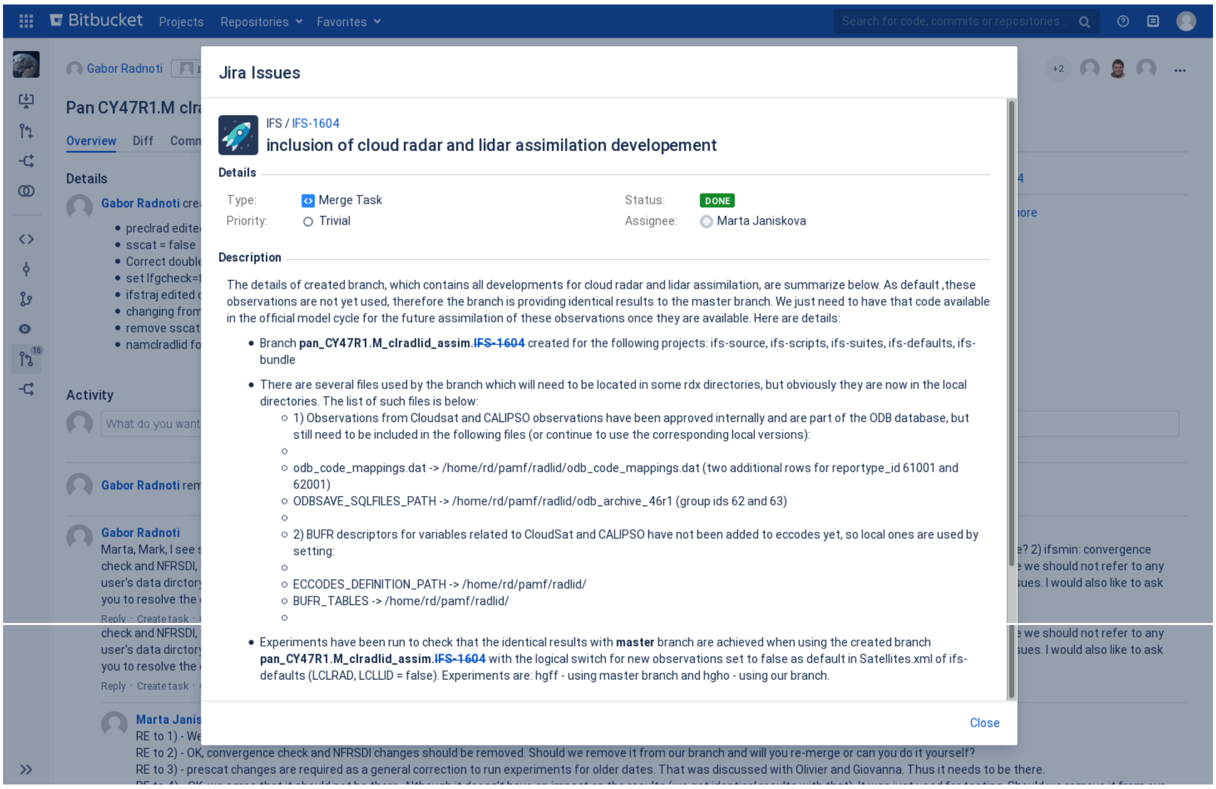
```
ftn-855 crayftn: ERROR IFS_DBASE_VIEW_MOD, File =
../../../../../../lus/snx11209/fws9/sb/work/rd/pan/hg9m/source/ifs-
source/ifs/module/ifs_dbase_view_mod.F90, Line = 1, Column = 8
  The compiler has detected errors in module "IFS_DBASE_VIEW_MOD".  No module information file
will be created for this module.


ftn-213 crayftn: ERROR SETUP_COL_POINTERS, File =
../../../../../../lus/snx11209/fws9/sb/work/rd/pan/hg9m/source/ifs-
source/ifs/module/ifs_dbase_view_mod.F90, Line = 533, Column = 33
  "HEADING" is not a component of derived type "CONV_TABLE".


ftn-213 crayftn: ERROR SETUP_COL_POINTERS, File =
../../../../../../lus/snx11209/fws9/sb/work/rd/pan/hg9m/source/ifs-
source/ifs/module/ifs_dbase_view_mod.F90, Line = 535, Column = 33
  "AIRCRAFT_TYPE" is not a component of derived type "CONV_TABLE".

Cray Fortran : Version 8.7.7 (20181127183827_bd8b4ca1f147e8694ac56cc60310ba7979cef5df)
Cray Fortran : Fri Aug 28, 2020  12:03:29
Cray Fortran : Compile time:  0.2040 seconds
Cray Fortran : 1225 source lines
Cray Fortran : 3 errors, 0 warnings, 0 other messages, 0 ansi
```

Figure 4.1: Example of failing compilation due to some merging conflicts shown for routine ifs_dbase_view_mod.F90 defining observation types.

4.2 Passing CY47R1 branch to be merged into the official model cycle

After basic testing assuring that the CY47R1 branch containing all new files and modifications required for cloud radar and lidar assimilation works, an official request for pulling it into official branch stream (Fig. 4.2) had to be created in order to start a complex preparation process to include our developments into the future official model cycle.

Our branch contains several so called projects:



Jira Issues

IFS / IFS-1604

inclusion of cloud radar and lidar assimilation developement

Details

Type: Merge Task
Priority: Trivial
Status: DONE
Assignee: Marta Janiskova

Description

The details of created branch, which contains all developments for cloud radar and lidar assimilation, are summarize below. As default ,these observations are not yet used, therefore the branch is providing identical results to the master branch. We just need to have that code available in the official model cycle for the future assimilation of these observations once they are available. Here are details:

- Branch **pan_CY47R1.M_clradlid_assim.~~IFS-1604~~** created for the following projects: ifs-source, ifs-scripts, ifs-suites, ifs-defaults, ifs-bundle
- There are several files used by the branch which will need to be located in some rdx directories, but obviously they are now in the local directories. The list of such files is below:
    - 1) Observations from Cloudsat and CALIPSO observations have been approved internally and are part of the ODB database, but still need to be included in the following files (or continue to use the corresponding local versions):
    -
    - odb_code_mappings.dat -> /home/rd/pamf/radlid/odb_code_mappings.dat (two additional rows for reportype_id 61001 and 62001)
    - ODBSAVE_SQLFILES_PATH -> /home/rd/pamf/radlid/odb_archive_46r1 (group ids 62 and 63)
    -
    - 2) BUFR descriptors for variables related to CloudSat and CALIPSO have not been added to eccodes yet, so local ones are used by setting:
    -
    - ECCODES_DEFINITION_PATH -> /home/rd/pamf/radlid/
    - BUFR_TABLES -> /home/rd/pamf/radlid/
    -
- Experiments have been run to check that the identical results with **master** branch are achieved when using the created branch **pan_CY47R1.M_clradlid_assim.~~IFS-1604~~** with the logical switch for new observations set to false as default in Satellites.xml of ifs-defaults (LCLRAD, LCLLID = false). Experiments are: hgff - using master branch and hgho - using our branch.

Figure 4.2: Request for merging local model branch containing routines for cloud radar and lidar assimilation into the official branch stream in order to appear in the future official model cycle.

- **ifs-source** - containing all type of routines, source code, to perform model integration;
- **ifs-scripts** - containing many scripts to perform different types of model integration (e.g. data processing, data assimilation, forecast, ...);
- **ifs-suites** - scripts for ecFlow system (ECMWF's work-flow manager enabling to run large number of programs) and scripts for acceptance tests for the new branches;
- **ifs-defaults** - different files for default setup and creating structure of experiments used by PrepIFS tool which ECMWF and Member State scientists use to configure their model experiments with ECMWF's IFS;
- **ifs-bundle** - scripts for loading different software and make files for compilation.

All these projects were a part of the merging request.

Creating request for merging triggered the whole machinery of pulling our branch and checking its consistency and correctness with all requirements for official model cycle. Several people were involved to review our contribution. We had to resolve several issues related to the scripts and testing environment for acceptance. Thanks to support of colleagues from IFS section, the raised problems were resolved.

4.3 Testing and debugging of the development in CY47R1

In this section we detail the testing of all the developments for cloud radar and lidar assimilation after updating it to the IFS model cycle CY47R1.

4.3.1 Technical testing

Technical testing is the first step always needed after updating any development to the higher model cycle. In the case of our assimilation developments, firstly such test is performed to ensure that the cloud radar and lidar observations are recognized correctly and are passed through to the Observation Data Base (ODB) properly. Secondly, after getting correct data in the ODB, a basic test of the 4D-Var minimization is made in order to check the convergence of the system with the new observations and operators.

The following steps of the whole data assimilation process (all of them described in more details in Section 3) were checked and successfully completed:

1. Observation pre-screening processing - This is used to reduce data volume of the main screening and to reject observations that fail to contain crucial information (such as missing observed value, observation error or vertical coordinate of observations), as well as to perform the so-called superobbing (i.e. placing observations onto a grid with resolution depending on the level of required smoothing) and/or vertical matching with the model vertical resolution. After creating the superobs, additional pre-screening is performed based on the number and the standard deviation of the observations forming the superob, the mean observation value itself and the observed cloud fraction.

2. Data conversion - The pre-processed cloud radar and lidar observations, which are at that point in BUFR (Binary Universal Form for the Representation of meteorological data) format, are converted to ODB at this stage of observation pre-processing.

3. Reading observations into the main 4D-Var assimilation tasks from ODB.

4. High-resolution trajectory run - During this run the model counterparts for all observations, cloud radar and lidar data included, are computed through the non-linear observation operators and the observation minus model differences (departures) are calculated and subsequently stored in the ODB. At this stage, additional screening is also performed based on the magnitude of departures, i.e. rejecting any observations with too large departures exceeding the predefined limits.

5. Blacklisting procedure - This processing firstly selects which observation types, variables, vertical ranges etc. will be used for the assimilation. Secondly, based on the observed values (or observation departure from the background) additional observations are rejected.

6. Updating ODB - The original ODB is complemented by the background departures together with quality information for observations after performing screening. A new reduced ODB is then created from the subset of the original ODB that contains only those observations that passed screening and that are to be used in the minimization.

7. 4D-Var minimization process - This represents the sequence of the high-resolution trajectory runs and lower resolution minimization integrations in order to obtain analysis increments. At the end of the whole process, the low resolution increments are added to the background (at initial time) and integrated to the analysis time to provide the final analysis. In this process, observation cost functions for cloud radar and lidar observations are checked whether they are decreased by progressing iteratively through the set of outer and inner loops (as described in Section 2) since such decrease is an indication that all developments brought from the previous cycle, CY46R1, works also in cycle CY47R1.

8. Post-processing of observations - This is done after completing all the tasks in the main 4D-Var calculations. The ODB files containing information about all observations, not only the active ones, created by the experiment are then archived in MARS (Meteorological Archiving and Retrieval System of ECMWF).

By checking that each of the tasks listed above of 4D-Var process were correctly performed we could conclude that porting of cloud radar and lidar assimilation to CY47R1 was successful. Obviously, to accomplish all the technical testing is a rather complex process as it often involves debugging parts of the system that are not working properly because of merging issues or changes in the higher model cycle which need to be adapted for the new observations. The technical test indicated that the observation cost function for cloud radar and lidar observations is progressively decreased through 4D-Var iteration process as expected and required, as well as that these new observations are properly stored in ODB and MARS. This is important sanity check before we can progress with more scientific evaluation and improvements to the system based on CY47R1. The first testing results of 4D-Var with cloud radar and lidar observations in the model cycle CY47R1 are summarized in Section 4.3.2.

The created and tested CY47R1 branch was subsequently merged into the official model cycle as a part of the preparations required in order to move these developments to the model cycle CY48R1 when that cycle enters its testing phase, so that they are in the operational code (albeit passively) when the cycle is released (due in 2021). An additional requirement was testing of our CY47R1 branch for the the bit reproducibility of the system when cloud radar and lidar observations are switched off (i.e. to secure that passive observations do not modify any integration of the operational code). This checking was successfully completed and therefore our development could enter into the acceptance process as described in Section 4.2.

4.3.2 First testing results of 4D-Var with cloud radar and lidar observations in CY47R1

As a first test of the updated assimilation system, a single-cycle analysis experiment is performed where observations of CloudSat radar reflectivity and CALIPSO lidar backscatter are fed to the system in addition to all the regularly assimilated observations. The output of this experiment is compared to an experiment using the previous model version that is otherwise identical. A visual assessment of this test is demonstrated on the portion of orbital track representing several model profiles containing radar and lidar observations assimilated in the data assimilation window between 21:00 UTC 31 July 2007 and 09:00 UTC 1 August 2007 (Figs. 4.3 - 4.5).

To test that the observations have entered the ODB correctly, the superobbed CloudSat radar reflectivity for cycles CY46R1 and CY47R1 are compared against each other. To pass the test the observations should be bit-identical as their processing is the same in each cycle. A similar approach is followed for testing the superobbed CALIPSO lidar backscatter (Figure 3.2) and is shown to be bit-identical. Whereas the observations should be identical, the corresponding observation error is not necessarily the same between cycles because it partially depends on the model first guess, which will be different due to the model changes between CY47R1 and CY46R1. However, it should be broadly similar, which is demonstrated in both Fig. 4.3 and Fig. 4.4. The screening for both observation types is also similar, but not necessarily identical, between cycles.

Next, we assess the difference between the first guess and the analysis of the model equivalent for the observations. For radar reflectivity, the analysis radar reflectivity is qualitatively the same in both cycles; the fit of the analysis to the observations is improved compared to the first guess. Similarly, for lidar backscatter, the analysis model equivalent is almost identical between cycles.

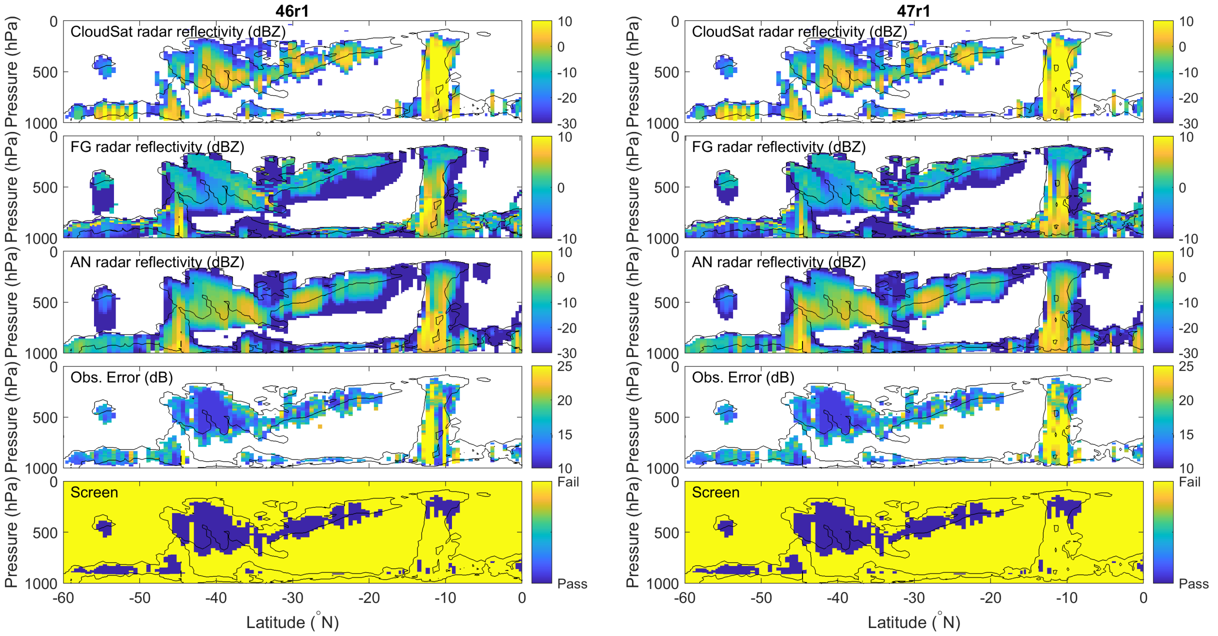


Figure 4.3: Cross-sections of cloud radar reflectivity related variables corresponding to the portion of orbital track on 31 July 2007 at 21:00 UTC. Panels show from top to bottom: observed CloudSat radar reflectivity (dBZ), model equivalent (FG) radar reflectivity using the model background (dBZ), model equivalent (AN) radar reflectivity using the model analysis from the assimilation experiment using all observations and radar and lidar, the observation error (dB) assigned to the observations and screening value (blue colour = observations passed). The cross-sections on the left panels correspond to outputs from the model cycle CY46R1, while the right panels are for the cycle CY47R1.

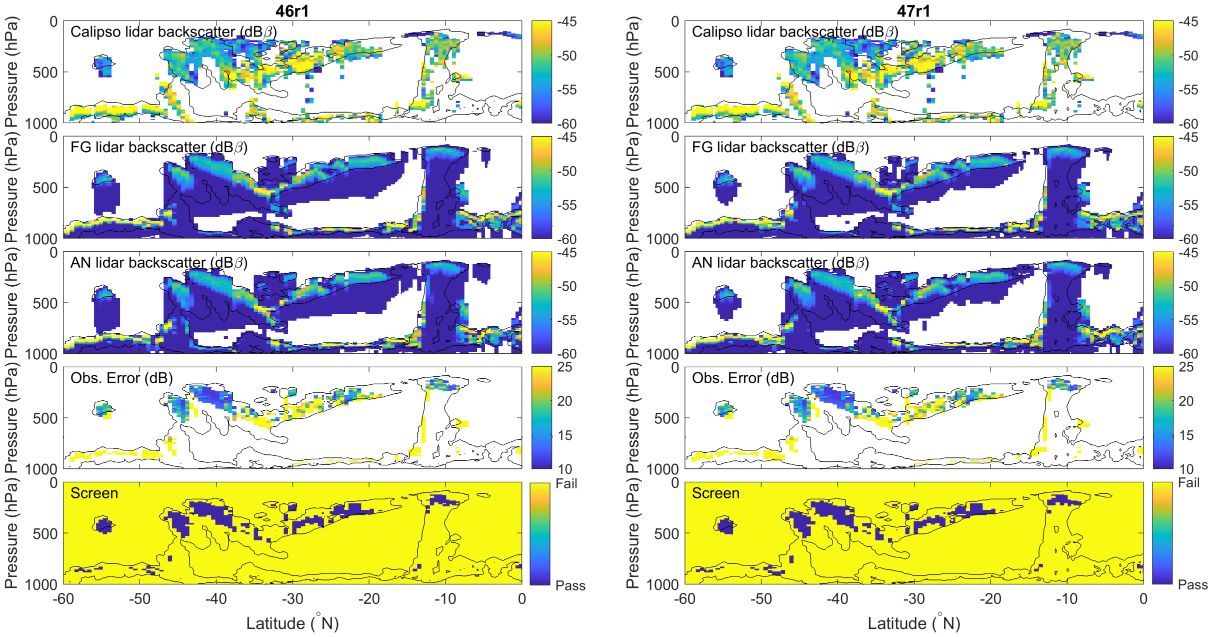


Figure 4.4: Same as Fig. 3.11, but for cloud lidar backscatter and CALIPSO attenuated backscatter (dBβ).

As a final check of the consistency of the analysis between cycles, Fig 4.5 shows the analysis increments of both radar and lidar observations and more tangible cloud related variables such as temperature and relative humidity. Comparing these two cycles, it is clear to see that there are a lot of similarities in obtained increments. For cloud radar reflectivity, we do not see the large differences in convective regions that were apparent between CY46R1 than in CY43R1. Also, the size of analysis increments of relative humidity are broadly similar between CY46R1 and CY47R1, whereas there were greater differences between CY43R1 and CY46R1, because of the greater number of differences to the model code between cycles.

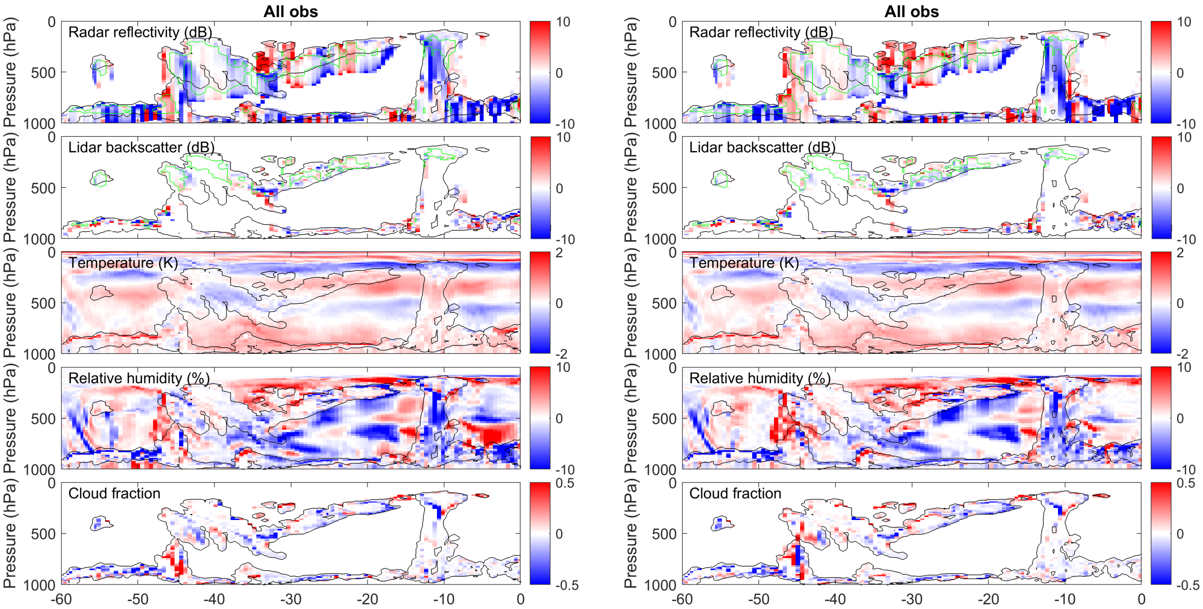


Figure 4.5: Analysis increments for the the portion of orbital track on 31 July 2007 at 21:00 UTC as in Fig. 3.11 and Fig. 3.12 for the model cycle CY46R1 (left column) and CY47R1 (right column). Panels from top to bottom show model-level increments of: radar reflectivity (dB), lidar backscatter (dB), temperature (K), relative humidity (%) and cloud fraction.

Overall, the performance of 4D-Var system assimilating cloud radar and lidar observations on the top of regularly used observations is similar for CY46R1 and for CY47R1. These first test results are reassuring as they indicate that cloud radar and lidar assimilation was successfully ported to CY47R1.

5 Technical developments and modifications for additional observations

Additional developments to the ECMWF assimilation system were done in preparation for the inclusion of EarthCARE observations other than cloud radar reflectivity and lidar backscatter, such as cloud extinction, Rayleigh backscatter and radar Doppler velocity. The WP-4000 report (Fielding and Janisková, 2021b) provides the scientific basis for the observation operators of these additional observations and summary of initial testing of the updated system. The technical modifications and developments to the cloud radar and lidar assimilation system in order to account for the above additional observations in the IFS system are described in this section.

5.1 Modifications to observation database

As described in Section2.3.2, to store and access data used by the 4D-Var system at ECMWF, the in-house data storage system, ODB, is used. In ODB, data is stored in either a header (**hdr**) entry or a **body** entry. Structure Query Language (SQL) is used by ODB for fast and efficient storage and retrieval of observational data.

For radar and lidar observations, the **hdr** contains the latitude, longitude and time when the observations were taken. In the **body**, the observation value to be monitored or assimilated is stored within the *obsvalue* variable, together with additional information such as the observation error, screening status and bias correction. Each observation type, in order to be identified by the system, has a unique variable number, *varno* and variable names as summarized in Table 5.1. Those identificators tell: (i) the ODB where and which data to store, and (ii) the 4D-Var system which observation operator must be used to produce the model-equivalent.

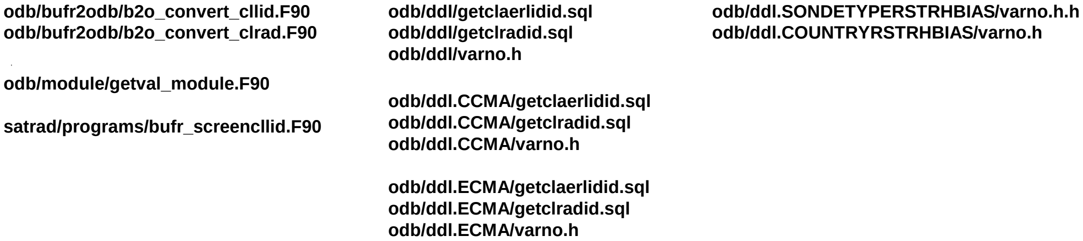
Variable name	short name	NVAR	varno
LIDAR_CLOUD_BACKSCATTER	CBSC	110	237
LIDAR_CLOUD_EXTINCTION	CEXT	112	238
LIDAR_RAYLEIGH_BACKSCATTER	RBSC	113	280
CLOUD_RADAR_REFLECTIVITY	CREF	111	239
CLOUD_DOPPLER_VELOCITY	CDOP	114	281

Table 5.1: List of variable names and varno numbers used within the ODB and IFS systems. The additional observations included to the system are in red.

When including new observations into the system, any new ODB variable names must be approved by the ODB Governance to ensure consistency between ECMWF, member states and collaborators. That needs to be done for all additional observations using the names as appeared in Table 5.1.

Each additional observation included should have a corresponding radar reflectivity or lidar backscatter value. Based on practice used for a passive satellite remote sensing instruments, where additional channels are just added to the same **body** belonging to them by extending the number of body rows, the same approach was applied for the additional cloud radar and lidar observations. They share the same **hdr** and just **body** is extended by adding CLOUD_DOPPLER_VELOCITY for radar observations, and by adding LIDAR_CLOUD_EXTINCTION and LIDAR_RAYLEIGH_BACKSCATTER for lidar observations.

To accommodate the new observation types, the above described modifications to the ODB system had to be applied. Additional changes were required to the the bufr2odb code and also to the SQL requests used to communicate between the IFS and the ODB. The list of modified routines accounting for those changes is in Fig. 5.1.



odb/bufr2odb/b2o_convert_cllid.F90
odb/bufr2odb/b2o_convert_clrad.F90

odb/module/getval_module.F90

satrad/programs/bufr_screencllid.F90

odb/ddl/getclaerlidid.sql
odb/ddl/getclradid.sql
odb/ddl/varno.h

odb/ddl.CCMA/getclaerlidid.sql
odb/ddl.CCMA/getclradid.sql
odb/ddl.CCMA/varno.h

odb/ddl.ECMA/getclaerlidid.sql
odb/ddl.ECMA/getclradid.sql
odb/ddl.ECMA/varno.h

odb/ddl.SONDETYPERSTRHBIAS/varno.h.h
odb/ddl.COUNTRYRSTRHBIAS/varno.h

Figure 5.1: The list of modified files required to account for the additional observations in the ODB system used by 4D-Var assimilation.

5.2 Modifications to the model code

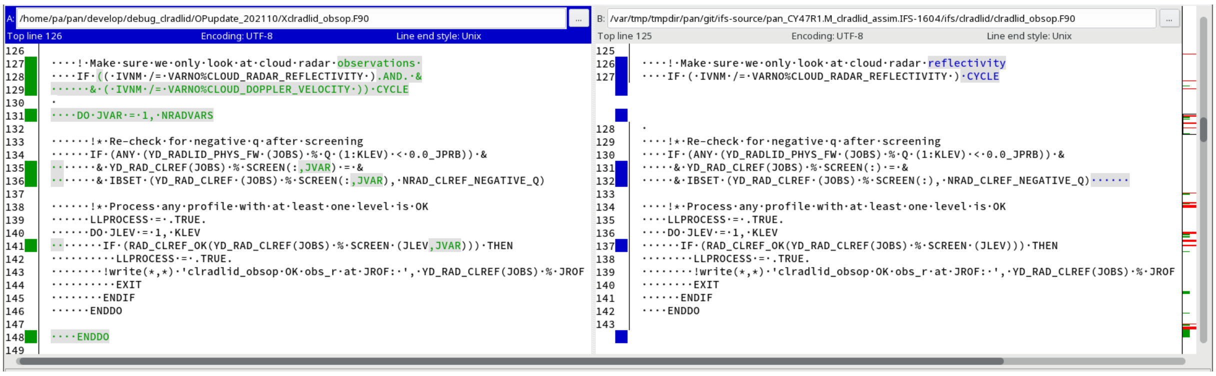
Accounting for additional observations required several modifications in the model code which is schematically described in Fig. 2.6 - 2.7.

For including the new observations into the system, computation of the model equivalent to these observations needs to be updated to account for them. Such computation is done in the master routine, **hop**, dealing with all different types of observations in the IFS (Fig. 2.6). Under the **hop** routine, the subroutine for all computations related to cloud radar and lidar observations, **obsop_clradlid**, is called. The interface between **hop** and the suite of different **clradlid** routines, which is necessary when **clradlid** is called in the observation operator, is provided by **clradlid_wrapper** routine. This interface routine serving for direct, tangent-linear (TL) and adjoint (AD) purposes has been modified for including new observations. At this point, having access to the model state information, computation of situation dependent observation errors, screening for high observation errors as well as departures and assigning bias correction are performed. In this routine, observational data are obtained from ODB using **clradl_get** routines (suffix **_tl** and **_ad** are used for the TL and AD versions of those routines) and all observation related information is then stored to ODB in the **clradl_put** and **clradl_put_tl** routines at the end of **clradlid_wrapper** routine. All these routines for getting and storing data related to observations from/to ODB had to be updated. The modifications had also to be applied to the routine **clradlid_screen** providing observation screening depending on model situation. The routines computing the model equivalents to cloud radar and lidar observations, **clradlid_obsop** (suffix **_tl** and **_ad** for TL and AD versions, respectively) were modified to account for all additional observations. As shown in Fig. 5.2, showing a comparison between the original (on the right) and the updated (on the left) versions of **clradlid_obsop**, CLOUD_DOPPLER_VELOCITY was added to the part for radar observations and LIDAR_CLOUD_EXTINCTION together with LIDAR_RAYLEIGH_BACKSCATTER to the part used for lidar observations. Having more types of observations for radar and lidar, the loop for number of observation types was included, NRADVARS and NLIDVARS, respectively.

More detailed structure of the subroutines for the observation operators is in Fig. 2.7 which shows the call tree of subroutines for the direct, non-linear and adjoint integrations called under **clradlid_obsop** and **clradlid_obsop_ad**. A brief description of the contents of all the observation operator subroutines for cloud radar and lidar observations is provided in Section 2.5. Here, only the subroutines which required modifications in order to account for the additional observations are described. Both drivers routines, **clradar_driver** and **cllidar_driver**, were originally used for computations of cloud radar reflectivity and lidar backscatter only. Computations related to the new observations had to be included there. That had to be done for the TL and AD versions of those routines as well.

For radar, the subroutine **clradlid_radars** together with **clradlid_radars_tl** and **clradlid_radars_ad** were updated to compute Doppler velocity on top of the already existing computation of radar reflectivity and extinction from the input values of temperature, specific humidity, rain, snow, cloud and ice using a precomputed look-up table of radar reflectivity and extinction values, extended for Doppler velocity. This also required modifications to **clradlid_lut_radars**, to search in the table for Doppler velocity values as it was done for reflectivity and extinction based on localized temperature and hydrometeors intervals. The scaling of extinction is also performed there to obtain grid-box mean values. The TL routine, **clradlid_lut_radars_tl**, and the adjoint routine, **clradlid_lut_radars_ad** had to be updated accordingly.

For lidar, similarly as for radar, the subroutine **clradlid_lidars** and their TL and AD versions (**clradlid_lidars_tl** and **clradlid_lidars_ad**, respectively) required to include computation of Rayleigh backscatter and some adjustments to the computation of extinction.



A: /home/pa/pan/develop/debug_clradlid/OPupdate_202110/Xclradlid_obsop.F90

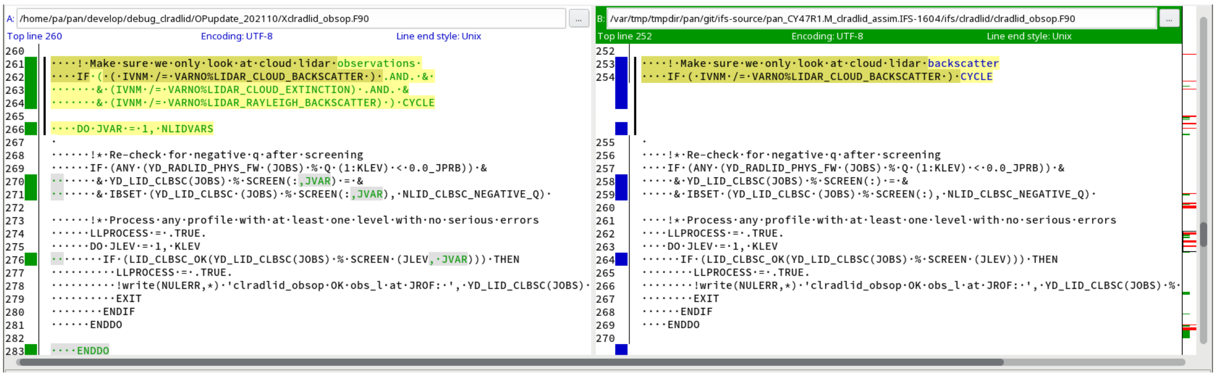
Top line 126 Encoding: UTF-8 Line end style: Unix

```

    ! Make sure we only look at cloud radar observations 
    IF (( IVNM /= VARNO%CLOUD_RADAR_REFLECTIVITY ).AND. &
      & ( IVNM /= VARNO%CLOUD_DOPPLER_VELOCITY )) CYCLE
 
    DO JVAR = 1, NRADVARS

      !* Re-check for negative q after screening
      IF (ANY (YD_RADLID_PHYS_FW (JOBS) % Q (1:KLEV) < 0.0_JPRB)) &
        & YD_RAD_CLREF(JOBS) % SCREEN(:,JVAR) = &
        & IBSET (YD_RAD_CLREF (JOBS) % SCREEN(:,JVAR), NRAD_CLREF_NEGATIVE_Q)

      !* Process any profile with at least one level is OK
      LLPROCESS = .TRUE.
      DO JLEV = 1, KLEV
        IF (RAD_CLREF_OK(YD_RAD_CLREF(JOBS) % SCREEN (JLEV,JVAR))) THEN
          LLPROCESS = .TRUE.
        !write(*,*) 'clradlid_obsop OK obs_r at JROF: ', YD_RAD_CLREF(JOBS) % JROF
          EXIT
        ENDIF
      ENDDO

    ENDDO

```

B: /var/tmp/tmpdir/pan/git/ifs-source/pan_CY47R1.M_clradlid_assim.IFS-1604/ifs/clradlid/clradlid_obsop.F90

Top line 125 Encoding: UTF-8 Line end style: Unix

```

    ! Make sure we only look at cloud radar reflectivity
    IF ( IVNM /= VARNO%CLOUD_RADAR_REFLECTIVITY ) CYCLE
 
    !* Re-check for negative q after screening
    IF (ANY (YD_RADLID_PHYS_FW (JOBS) % Q (1:KLEV) < 0.0_JPRB)) &
     & YD_RAD_CLREF(JOBS) % SCREEN(:) = &
     & IBSET (YD_RAD_CLREF (JOBS) % SCREEN(:), NRAD_CLREF_NEGATIVE_Q)      

    !* Process any profile with at least one level is OK
    LLPROCESS = .TRUE.
    DO JLEV = 1, KLEV
      IF (RAD_CLREF_OK(YD_RAD_CLREF(JOBS) % SCREEN (JLEV))) THEN
        LLPROCESS = .TRUE.
        !write(*,*) 'clradlid_obsop OK obs_r at JROF: ', YD_RAD_CLREF(JOBS) % JROF
        EXIT
      ENDIF
    ENDDO

```



A: /home/pa/pan/develop/debug_clradlid/OPupdate_202110/Xclradlid_obsop.F90

Top line 260 Encoding: UTF-8 Line end style: Unix

```

    ! Make sure we only look at cloud lidar observations 
    IF ( ( IVNM /= VARNO%LIDAR_CLOUD_BACKSCATTER ) .AND. & 
       & (IVNM /= VARNO%LIDAR_CLOUD_EXTINCTION) .AND. &
       & (IVNM /= VARNO%LIDAR_RAYLEIGH_BACKSCATTER) ) CYCLE

    DO JVAR = 1, NLIDVARS
 
      !* Re-check for negative q after screening
      IF (ANY (YD_RADLID_PHYS_FW (JOBS) % Q (1:KLEV) < 0.0_JPRB)) &
        & YD_LID_CLBSC(JOBS) % SCREEN(:,JVAR) = &
        & IBSET (YD_LID_CLBSC (JOBS) % SCREEN(:,JVAR), NLID_CLBSC_NEGATIVE_Q) 

      !* Process any profile with at least one level with no serious errors
      LLPROCESS = .TRUE.
      DO JLEV = 1, KLEV
        IF (LID_CLBSC_OK(YD_LID_CLBSC(JOBS) % SCREEN (JLEV, JVAR))) THEN
          LLPROCESS = .TRUE.
          !write(NULERR,*) 'clradlid_obsop OK obs_l at JROF: ', YD_LID_CLBSC(JOBS) 
          EXIT
        ENDIF
      ENDDO

    ENDDO
```

B: /var/tmp/tmpdir/pan/git/ifs-source/pan_CY47R1.M_clradlid_assim.IFS-1604/ifs/clradlid/clradlid_obsop.F90

Top line 252 Encoding: UTF-8 Line end style: Unix

```

    ! Make sure we only look at cloud lidar backscatter
    IF ( IVNM /= VARNO%LIDAR_CLOUD_BACKSCATTER ) CYCLE
 
    !* Re-check for negative q after screening
    IF (ANY (YD_RADLID_PHYS_FW (JOBS) % Q (1:KLEV) < 0.0_JPRB)) &
     & YD_LID_CLBSC(JOBS) % SCREEN(:) = &
     & IBSET (YD_LID_CLBSC (JOBS) % SCREEN(:), NLID_CLBSC_NEGATIVE_Q) 

    !* Process any profile with at least one level with no serious errors
    LLPROCESS = .TRUE.
    DO JLEV = 1, KLEV
      IF (LID_CLBSC_OK(YD_LID_CLBSC(JOBS) % SCREEN (JLEV))) THEN
        LLPROCESS = .TRUE.
        !write(NULERR,*) 'clradlid_obsop OK obs_l at JROF: ', YD_LID_CLBSC(JOBS) % 
        EXIT
      ENDIF
    ENDDO

```

Figure 5.2: Comparison between original (on the right) and updated (on the left) versions of **clradlid_obsop** *subroutine indicating modifications required to account for all additional observations.*

The list of all modified files in the model required to account for the additional observations in 4D-Var assimilation system is provided in Fig. 5.3.

ifs/clradlid/cllidar_driver.F90
ifs/clradlid/cllidar_driver_ad.F90
ifs/clradlid/cllidar_driver_tl.F90

ifs/clradlid/clradar_driver.F90
ifs/clradlid/clradar_driver_ad.F90
ifs/clradlid/clradar_driver_tl.F90

ifs/clradlid/clradlid_bc_err_screen.F90

ifs/clradlid/clradlid_get.F90
ifs/clradlid/clradlid_get_ad.F90
ifs/clradlid/clradlid_get_tl.F90

ifs/clradlid/clradlid_lidars.F90
ifs/clradlid/clradlid_lidars_ad.F90
ifs/clradlid/clradlid_lidars_tl.F90

ifs/clradlid/clradlid_lut_radars.F90
ifs/clradlid/clradlid_lut_radars_ad.F90
ifs/clradlid/clradlid_lut_radars_tl.F90

ifs/clradlid/clradlid_lut_setup.F90

ifs/clradlid/clradlid_obsop.F90
ifs/clradlid/clradlid_obsop_ad.F90
ifs/clradlid/clradlid_obsop_tl.F90

ifs/clradlid/clradlid_put.F90
ifs/clradlid/clradlid_put_tl.F90

ifs/clradlid/clradlid_radars.F90
ifs/clradlid/clradlid_radars_ad.F90
ifs/clradlid/clradlid_radars_tl.F90
ifs/clradlid/clradlid_screen.F90

ifs/clradlid/clradlid_setup.F90
ifs/clradlid/clradlid_wrapper.F90

Ifs/module/parcma.F90
ifs/module/pardimo.F90
Ifs/module/varno_module.F90
ifs/module/yomclradlid.F90
ifs/module/yomclradlid_op.F90
ifs/module/yomcosjo.F90

ifs/obs_preproc/black.F90
ifs/obs_preproc/defrun.F90
ifs/obs_preproc/fgchk.F90
ifs/obs_preproc/first.F90
ifs/obs_preproc/gefger.F90

ifs/op_obs/hjo.F90
ifs/op_obs/hop.F90
ifs/op_obs/hop_decide_required_sqls.F90
Ifs/op_obs/hretrv_conv.F90
ifs/op_obs/intervert_obserr_corF90
ifs/op_obs/map_varno_to_nvar.F90

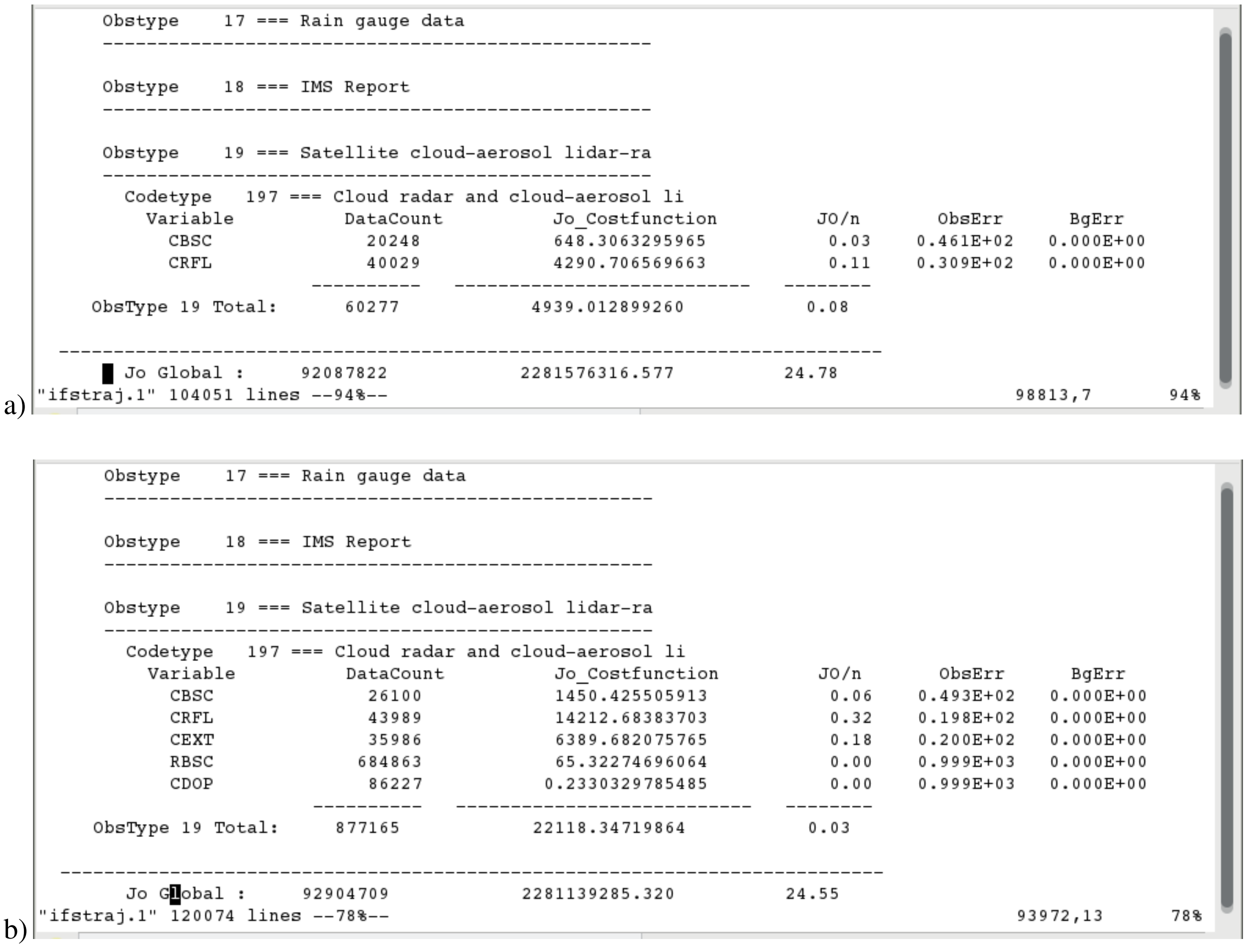
Figure 5.3: The list of modified files in the model required to account for the additional observations in 4D-Var assimilation system.

5.3 Technical testing

This section provides information on the technical testing performed to evaluate the correctness of our developments for including additional observations to the IFS system.

5.3.1 Test for nonlinear computation

Once the new observations are correctly recognized and are properly fed through to the ODB, the next step is to check whether the model changes required for these observations are correct. This can be done by using the 4D-Var assimilation system, concentrating on outputs from the job **ifstraj** (Fig. 5.4), i.e. high resolution trajectory (nonlinear) run. The details of all 4D-Var tasks can be found in Sections 2.4 and 3.3.1, as well as in Figs: 3.5 - 3.7 and 3.9.



a)

```
      Obstype    17 === Rain gauge data
      ---------------------------------------------------

      Obstype    18 === IMS Report
      ---------------------------------------------------

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      ---------------------------------------------------
        Codetype    197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction          JO/n       ObsErr      BgErr
            CBSC              20248            648.3063295965           0.03     0.461E+02   0.000E+00
            CRFL              40029            4290.706569663           0.11     0.309E+02   0.000E+00
                          ----------   ---------------------------   --------
     ObsType 19 Total:       60277           4939.012899260           0.08

  ---------------------------------------------------------------------------
        Jo Global :      92087822          2281576316.577          24.78
"ifstraj.1" 104051 lines --94%--                                                  98813,7       94%
```

b)

```
      Obstype    17 === Rain gauge data
      ---------------------------------------------------

      Obstype    18 === IMS Report
      ---------------------------------------------------

      Obstype    19 === Satellite cloud-aerosol lidar-ra
      ---------------------------------------------------
        Codetype    197 === Cloud radar and cloud-aerosol li
          Variable          DataCount          Jo_Costfunction          JO/n       ObsErr      BgErr
            CBSC              26100            1450.425505913           0.06     0.493E+02   0.000E+00
            CRFL              43989            14212.68383703           0.32     0.198E+02   0.000E+00
            CEXT              35986            6389.682075765           0.18     0.200E+02   0.000E+00
            RBSC             684863            65.32274696064           0.00     0.999E+03   0.000E+00
            CDOP              86227           0.2330329785485           0.00     0.999E+03   0.000E+00
                          ----------   ---------------------------   --------
     ObsType 19 Total:      877165           22118.34719864           0.03

  ---------------------------------------------------------------------------
        Jo Global :      92904709          2281139285.320          24.55
"ifstraj.1" 120074 lines --78%--                                                  93972,13      78%
```

Figure 5.4: Screen-shots of ifstraj outputs showing the cost function containing: (a) cloud radar reflectivity (CRFL) and lidar cloud backscatter (CBSC) only as used in the original system for cloud radar and lidar assimilation, and (b) containing also the additional observations of lidar cloud extinction (CEXT), lidar Rayleigh backscatter (RBSC) and cloud Doppler velocity (CDOP) as prepared for the updated system.

During **ifstraj** run, the model counterparts for all observations are calculated through the non-linear observation operators. The observation minus model differences (departures) are then computed and stored in the ODB. These computed departures are an important input to the data selection procedures and they are used for the computation of observation cost function. The output from **ifstraj** provides information about the usage of observations, such the number of observations (DataCount) in the different observation code type groups (Codetype 197 === Cloud radar and cloud-aerosol lidar), the contribution of particular observations to the observation cost function (Jo_Costfunction), the value of cost function per observation (Jo/n), as well as the value of observation errors (ObsErr). In the original system, only observations of cloud radar reflectivity (CRFL) and lidar cloud backscatter (CBSC) were used (Fig. 5.4a). The updated system also contains the additional observations of lidar cloud extinction (CEXT), lidar Rayleigh backscatter (RBSC) and cloud Doppler velocity (CDOP) (Fig. 5.4b).

The output from **ifstraj** (Fig. 5.4b) indicates that the updated system succeeded to compute contributions from the additional observations and to include them to the observation cost function. For the observations, such as Rayleigh backscatter and Doppler velocity, for which the observation errors have been set to be very large, the cost function contribution is small. This is because of the cost function is given by the sum of the square of the first guess departures divided by the observation errors. One can also notice that the Rayleigh backscatter observations have the greatest number of observations because a first-guess departure is recorded in both clear and cloudy conditions.

5.3.2 Test of tangent-linear and adjoint correctness

Once the non-linear (NL) version of the observation operator was developed and tested, efforts were first devoted to the development of the tangent-linear code and then of adjoint code. The TL and AD versions of the observation operator for the additional included observations, such as Rayleigh backscatter and radar Doppler velocity, are needed in order to be able to perform any feasibility studies for assimilation of these observation types directly in 4D-Var.

At ECMWF, manual line-by-line TL and AD coding was applied for the linearization of the model code, and such approach was also used for cloud radar and lidar observation operators. Once the TL and AD versions of the NL code are derived, it is imperative to check the correctness of the derived codes. Without that, a proper convergence of 4D-Var system to the optimal value is not guaranteed. Even a small error can lead to failure of the 4D-Var calculations.

Tangent-linear test

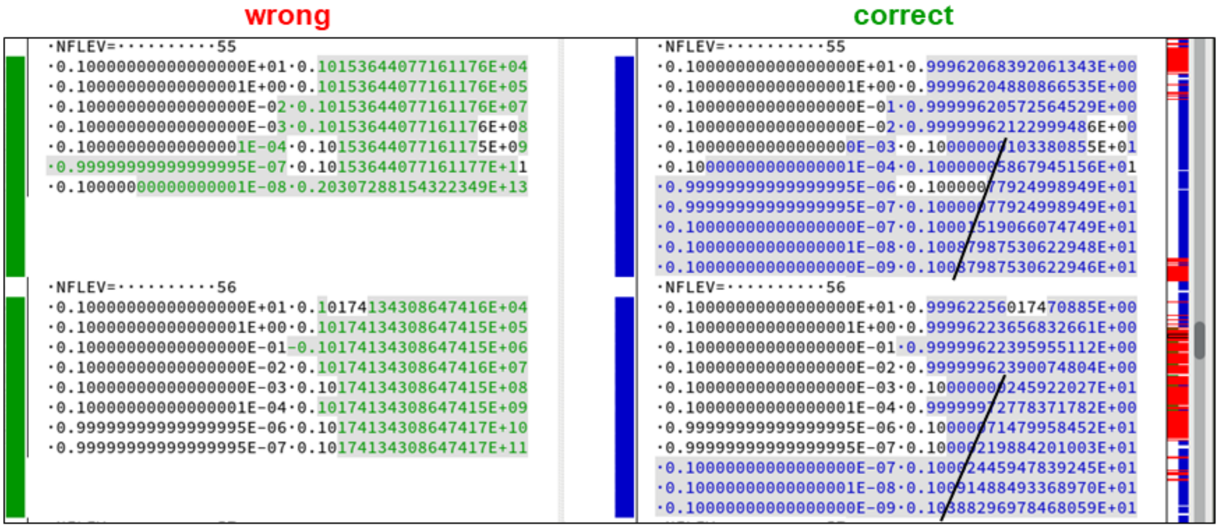
To build the TL model, the linearization is performed with respect to the local tangent of the model trajectory.

Test of tangent linear is used to check the numerical correctness of TL version of the model through the classical Taylor formula which provides an examination of the asymptotic behaviour, using perturbations the size of which becomes infinitesimally small:

$$\lim_{\lambda \to 0} \frac{M(\mathbf{x}+\lambda\delta\mathbf{x})-M(\mathbf{x})}{\mathbf{M}(\lambda\delta\mathbf{x})} = 1, \tag{5.1}$$

where M is the model describing the time evolution of the model state, $\mathbf{x}$, and $\delta\mathbf{x}$ is the model state perturbation.

Figure 5.5 shows an example of the TL test result for an output value of the model equivalent to observations at the different model levels. The TL code is correct when there is a clear convergence to unity based on Eq. 5.1.



```
·NFLEV=···········55
·0.10000000000000000E+01·0.10153644077161176E+04
·0.10000000000000001E+00·0.10153644077161176E+05
·0.10000000000000000E-02·0.10153644077161176E+07
·0.10000000000000000E-03·0.10153644077161176E+08
·0.10000000000000001E-04·0.10153644077161175E+09
·0.99999999999999995E-07·0.10153644077161177E+11
·0.10000000000000001E-08·0.20307288154322349E+13


·NFLEV=···········56
·0.10000000000000000E+01·0.10174134308647416E+04
·0.10000000000000001E+00·0.10174134308647415E+05
·0.10000000000000000E-01-0.10174134308647415E+06
·0.10000000000000000E-02·0.10174134308647416E+07
·0.10000000000000000E-03·0.10174134308647415E+08
·0.10000000000000001E-04·0.10174134308647415E+09
·0.99999999999999995E-06·0.10174134308647417E+10
·0.99999999999999995E-07·0.10174134308647417E+11
```

```
·NFLEV=···········55
·0.10000000000000000E+01·0.99962068392061343E+00
·0.10000000000000001E+00·0.99996204880866535E+00
·0.10000000000000000E-01·0.99999620572564529E+00
·0.10000000000000000E-02·0.99999962122999486E+00
·0.10000000000000000E-03·0.10000000103380855E+01
·0.10000000000000001E-04·0.10000005867945156E+01
·0.99999999999999995E-06·0.10000007924998949E+01
·0.99999999999999995E-07·0.10000007924998949E+01
·0.10000000000000000E-07·0.10001519066074749E+01
·0.10000000000000001E-08·0.10087987530622948E+01
·0.10000000000000000E-09·0.10087987530622946E+01
·NFLEV=···········56
·0.10000000000000000E+01·0.99962256017470885E+00
·0.10000000000000001E+00·0.99996223656832661E+00
·0.10000000000000000E-01·0.99999622395955112E+00
·0.10000000000000000E-02·0.99999962390074804E+00
·0.10000000000000000E-03·0.10000000245922027E+01
·0.10000000000000001E-04·0.99999972778371782E+00
·0.99999999999999995E-06·0.10000071479958452E+01
·0.99999999999999995E-07·0.10000219884201003E+01
·0.10000000000000000E-07·0.10002445947839245E+01
·0.10000000000000001E-08·0.10091488493368970E+01
·0.10000000000000000E-09·0.10388296978468059E+01
```

Figure 5.5: Example of the wrong (on the left) and the correct (on the right) TL test result for an output value of the model equivalent to observations at the different model levels (NFLEV).

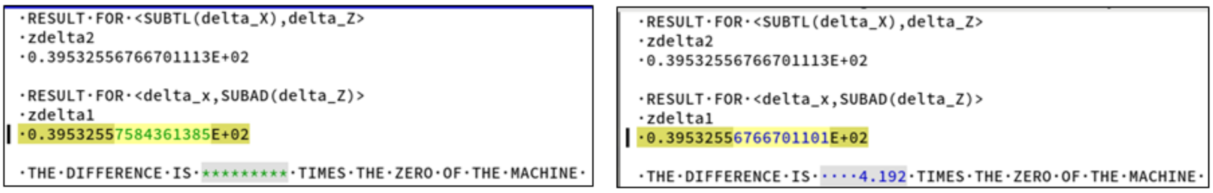
Adjoint test

The aim of this test is to check the correctness of the adjoint code with respect to its tangent linear counterpart, i.e. to provide the so-called test of adjoint identity. This is based on the definition of adjoint. The adjoint of a linearized operator, **M**, is the linear operator, $\mathbf{M}^*$, such that:

$$\forall \mathbf{x}, \forall \mathbf{y} \qquad < \mathbf{M}.\mathbf{x}, \mathbf{y} > = < \mathbf{x}, \mathbf{M}^*.\mathbf{y} > \tag{5.2}$$

where $<,>$ denotes the inner product and **x** and **y** are input vectors.

An example of the AD test result is shown in Fig. 5.6. The adjoint is considered to be correct when the TL inner product agrees with the adjoint inner product on around 10-12 digits at least to provide a difference between these products which is ideally not more than few hundred times the zero of the machine used for the test.



```
·RESULT·FOR·<SUBTL(delta_X),delta_Z>
·zdelta2
·0.39532556766701113E+02

·RESULT·FOR·<delta_x,SUBAD(delta_Z)>
·zdelta1
·0.39532557584361385E+02

·THE·DIFFERENCE·IS·*********·TIMES·THE·ZERO·OF·THE·MACHINE·
```

```
·RESULT·FOR·<SUBTL(delta_X),delta_Z>
·zdelta2
·0.39532556766701113E+02

·RESULT·FOR·<delta_x,SUBAD(delta_Z)>
·zdelta1
·0.39532556766701101E+02

·THE·DIFFERENCE·IS·····4.192·TIMES·THE·ZERO·OF·THE·MACHINE·
```

Figure 5.6: Example of the wrong (on the left) and the correct (on the right) AD test result for all outputs from the radar and lidar observation operators.

It should be emphasized that it is absolutely essential to ensure that the TL and AD codes satisfy the test to the level of machine precision, even when vectors are global 3D atmospheric states and even for time integrations up to 12 or 24 hours. Thus getting the correct TL and AD is often very time consuming.

6 Updating the cloud radar and lidar assimilation in CY48R1

The IFS model cycle CY48R1 was officially released for testing and experimentation towards the end of October 2022. The operational implementation of cycle CY48R1 is currently planned for Q2 2023. Although it will be the first new scientific model upgrade to run on Atos at ECWMF's new Data Centre in Bologna, it is the cycle CY47R3, previously running on Cray, which was the first cycle implemented for the operational use on the Atos system on 18 October 2022. Based on the plans, the next new cycle, CY49R1, is planned to be declared in June 2023, with the aim for operational implementation by Q1 2024. But the dates might be reviewed based on when the cycle CY48R1 will actually become operational. Therefore majority of experimentations and adjustments for the EarthCARE observations, as a part of the pre-operational preparations, will be carried out using the cycle CY48R1. In the case of some delays with cycle upgrades, the cycle CY48R1 might even still be operational at the EarthCARE launch.

6.1 A brief summary of required work

Up to now, all major experimentations for cloud radar and lidar assimilation were carried out using the cycle CY46R1 as described in the work package reports WP-2000, WP-3000, WP-4000 and WP-5000 (Fielding and Janisková, 2020, 2021a,b, 2022). As part of preparations to include the assimilation system developments for cloud radar and lidar observation to the official release of CY48R1, it was necessary to port them first from the cycle CY46R1 to CY47R1 (Section 4).

Once CY48R1 was declared, it was necessary to check whether the cloud radar and lidar assimilation system was functional and up-to date in this new cycle. It was discovered that several updates must be done based on:

- testing and experimentation using the cycle CY46R1 (i.e. developments, modifications and corrections which were not yet ready or known when porting to an intermediate cycle CY47R1);
- requirements to make the system working when running IFS 4D-Var data assimilation under the framework of the Object-Oriented Prediction System (OOPS, English et al., 2017) to be used from the cycle CY48R1.

The lists of updated routines in CY48R1 are in Tables 6.1 and 6.2 for the source code routines and scripts, respectively.

The majority of the required updates were related to porting missing modifications and corrections from the cycle CY46R1 to the cycle CY48R1. That applies to all routines in the directories *arpifs/clradlid*, *odb/bufr2odb* and *satrad/programs* as listed in Tab. 6.1. The remaining source code routines (Tab. 6.1) and scripts (Tab. 6.2) had to be updated to allow the cloud radar and lidar assimilation to run under OOPS.

A lot of modifications to allow to run our development in the cycle CY48R1 (i.e. required changes between CY47R1, where we ported our developments, and the new cycle CY48R1) using either the original IFS or OOPS-IFS systems were done by our colleagues dealing with the model cycle updates. Since the cloud radar and lidar assimilation is switched off as default in CY48R1, checking that everything was updated for using these observations and is functional in the new cycle, had to be done as part of of the project after official cycle release. Many changes in some routines can appear even between two consecutive cycles as illustrated in Fig. 6.1 for the routine *hop.F90* which is the master routine calling different

arpifs/clradlid/cllidar_driver.F90
arpifs/clradlid/cllidar_driver_tl.F90
arpifs/clradlid/cllidar_driver_ad.F90
arpifs/clradlid/clradar_driver.F90
arpifs/clradlid/clradar_driver_tl.F90
arpifs/clradlid/clradar_driver_ad.F90
arpifs/clradlid/clradlid_cp_overlap.F90
arpifs/clradlid/clradlid_cp_overlap_tl.F90
arpifs/clradlid/clradlid_cp_overlap_ad.F90
arpifs/clradlid/clradlid_lidars.F90
arpifs/clradlid/clradlid_lidars_ad.F90
arpifs/clradlid/clradlid_lut_lidars_tl.F90
arpifs/clradlid/clradlid_lut_lidars_ad.F90
arpifs/clradlid/clradlid_lut_radars_tl.F90
arpifs/clradlid/clradlid_lut_radars_ad.F90
arpifs/clradlid/clradlid_radars_ad.F90
arpifs/clradlid/clradlid_get_ad.F90
arpifs/clradlid/clradlid_put.F90
arpifs/clradlid/clradlid_bc_err_screen.F90
arpifs/clradlid/clradlid_setup.F90

arpifs/op_obs/hdepart.F90
arpifs/op_obs/hop.F90

odb/bufr2odb/b2o_convert_cllid.F90
odb/bufr2odb/b2o_convert_clrad.F90

satrad/programs/bufr_screen_cllid.F90
satrad/programs/bufr_screen_clrad.F90

Table 6.1: Source code modifications for cloud radar and lidar observations required in the cycle CY48R1.

gen/bufr2odb
gen/ifsmin
gen/link_satrad_data
gen/model
gen/oopsvar

gen/precllid
gen/preclrad
gen/prescat
gen/sstana

Table 6.2: Script modifications for cloud radar and lidar observations required in the cycle CY48R1.

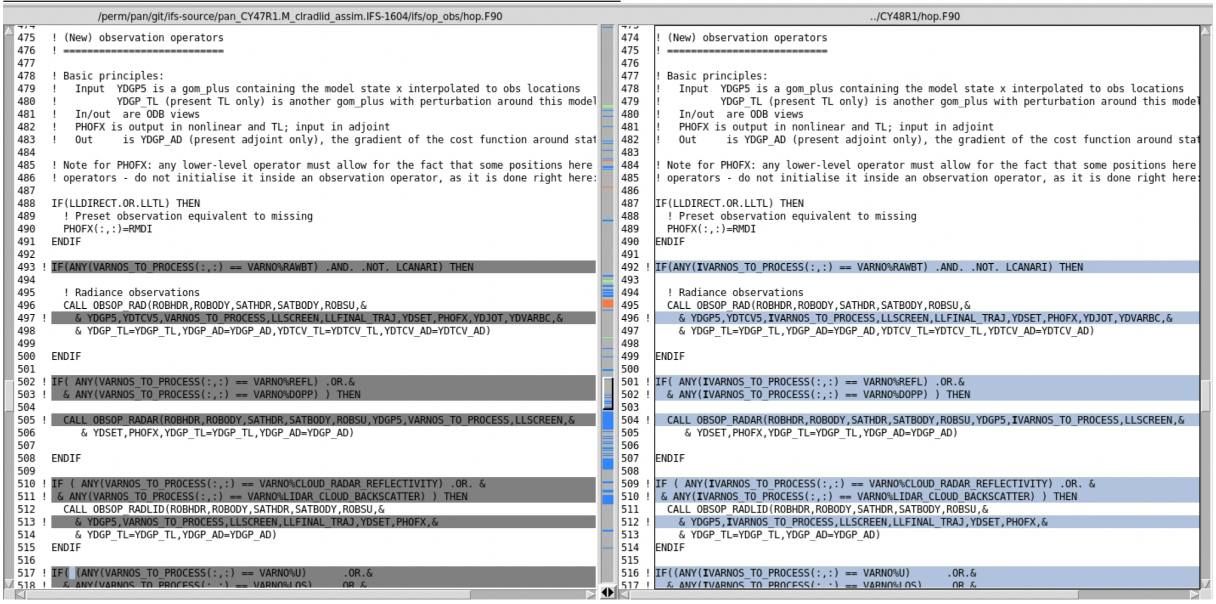
observation operators for all types of observations. The top of Fig. 6.1 shows changes done between the cycle CY47R1 and CY48R1, including modifications for cloud radar and lidar observations, which were done during a general update of the model cycle, and thus appeared directly in the official release of CY48R1. However, our testing revealed that there were still some problems for our type of observations to be resolved. The most difficult problem to solve appeared to be finding out how to account properly for the bias correction of cloud radar and lidar observations when using the OOPS-IFS system. Although, at the end it required rather small, but important, correction to be applied in the *hop.F90* routine as illustrated at the bottom of Fig. 6.1.

Once modifications of the routines listed in Tables 6.1 and 6.2 have been done, the first tests of assimilating cloud radar and lidar observations in the cycle CY48R1 could be performed.

6.2 Testing and debugging of the development in CY48R1

In this section we detail the testing of all the developments for cloud radar and lidar assimilation after they appeared in the official release of the IFS model cycle CY48R1.

Update of hop.f90 routine from CY47R1 to CY48R1



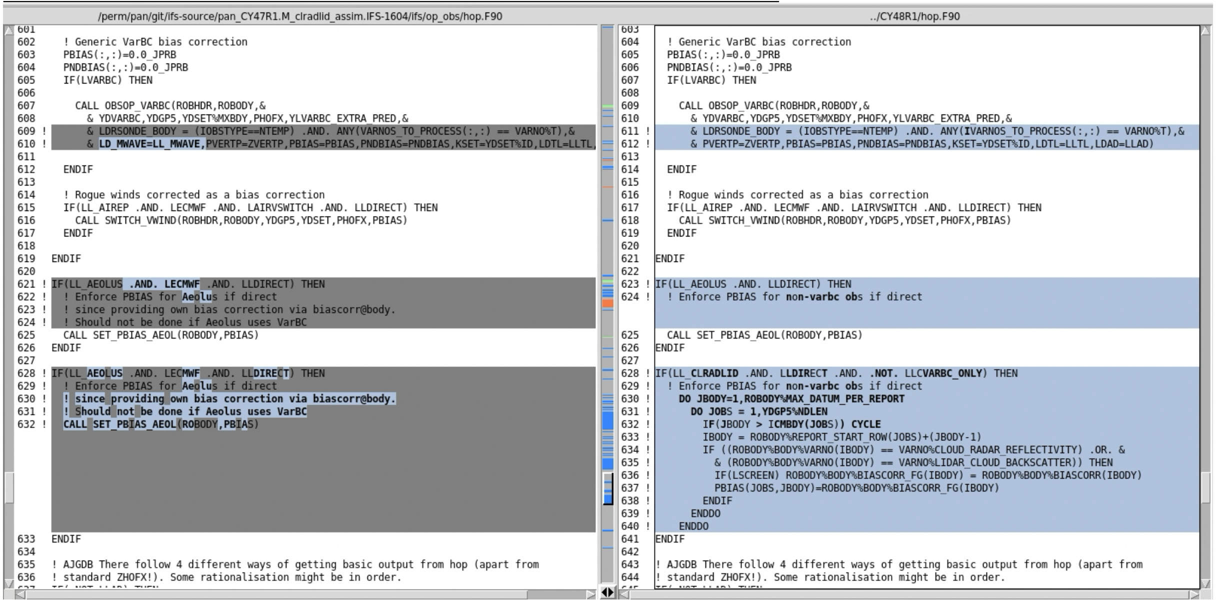
/perm/pan/git/ifs-source/pan_CY47R1.M_clradlid_assim.IFS-1604/ifs/op_obs/hop.F90

```
  ! (New) observation operators
  ! ===========================

  ! Basic principles:
  !   Input  YDGP5 is a gom_plus containing the model state x interpolated to obs locations
  !          YDGP_TL (present TL only) is another gom_plus with perturbation around this model
  !   In/out  are ODB views
  !   PHOFX is output in nonlinear and TL; input in adjoint
  !   Out     is YDGP_AD (present adjoint only), the gradient of the cost function around stat

  ! Note for PHOFX: any lower-level operator must allow for the fact that some positions here
  ! operators - do not initialise it inside an observation operator, as it is done right here:

  IF(LLDIRECT.OR.LLTL) THEN
    ! Preset observation equivalent to missing
    PHOFX(:,:)=RMDI
  ENDIF

! IF(ANY(VARNOS_TO_PROCESS(:,:) == VARNO%RAWBT) .AND. .NOT. LCANARI) THEN

    ! Radiance observations
    CALL OBSOP_RAD(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,&
!     & YDGP5,YDTCV5,VARNOS_TO_PROCESS,LLSCREEN,LLFINAL_TRAJ,YDSET,PHOFX,YDJOT,YDVARBC,&
      & YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD,YDTCV_TL=YDTCV_TL,YDTCV_AD=YDTCV_AD)

  ENDIF

! IF( ANY(VARNOS_TO_PROCESS(:,:) == VARNO%REFL) .OR.&
!   & ANY(VARNOS_TO_PROCESS(:,:) == VARNO%DOPP) ) THEN

!   CALL OBSOP_RADAR(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,YDGP5,VARNOS_TO_PROCESS,LLSCREEN,&
       & YDSET,PHOFX,YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD)

  ENDIF

! IF ( ANY(VARNOS_TO_PROCESS(:,:) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
!  & ANY(VARNOS_TO_PROCESS(:,:) == VARNO%LIDAR_CLOUD_BACKSCATTER) ) THEN
    CALL OBSOP_RADLID(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,&
!     & YDGP5,VARNOS_TO_PROCESS,LLSCREEN,LLFINAL_TRAJ,YDSET,PHOFX,&
      & YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD)
  ENDIF

! IF( (ANY(VARNOS_TO_PROCESS(:,:) == VARNO%U)      .OR.&
!   & ANY(VARNOS_TO_PROCESS(:,:) == VARNO%LOS)      .OR.&
```

../CY48R1/hop.F90

```
 ! (New) observation operators
 ! ===========================

 ! Basic principles:
 !   Input  YDGP5 is a gom_plus containing the model state x interpolated to obs locations
 !          YDGP_TL (present TL only) is another gom_plus with perturbation around this model
 !   In/out  are ODB views
 !   PHOFX is output in nonlinear and TL; input in adjoint
 !   Out     is YDGP_AD (present adjoint only), the gradient of the cost function around stat

 ! Note for PHOFX: any lower-level operator must allow for the fact that some positions here
 ! operators - do not initialise it inside an observation operator, as it is done right here:

 IF(LLDIRECT.OR.LLTL) THEN
   ! Preset observation equivalent to missing
   PHOFX(:,:)=RMDI
 ENDIF

! IF(ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%RAWBT) .AND. .NOT. LCANARI) THEN

   ! Radiance observations
   CALL OBSOP_RAD(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,&
!    & YDGP5,YDTCV5,IVARNOS_TO_PROCESS,LLSCREEN,LLFINAL_TRAJ,YDSET,PHOFX,YDJOT,YDVARBC,&
     & YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD,YDTCV_TL=YDTCV_TL,YDTCV_AD=YDTCV_AD)

 ENDIF

! IF( ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%REFL) .OR.&
!   & ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%DOPP) ) THEN

!  CALL OBSOP_RADAR(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,YDGP5,IVARNOS_TO_PROCESS,LLSCREEN,&
      & YDSET,PHOFX,YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD)

 ENDIF

! IF ( ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
!  & ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%LIDAR_CLOUD_BACKSCATTER) ) THEN
   CALL OBSOP_RADLID(ROBHDR,ROBODY,SATHDR,SATBODY,ROBSU,&
!    & YDGP5,IVARNOS_TO_PROCESS,LLSCREEN,LLFINAL_TRAJ,YDSET,PHOFX,&
     & YDGP_TL=YDGP_TL,YDGP_AD=YDGP_AD)
 ENDIF

! IF((ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%U)      .OR.&
!   & ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%LOS)      .OR.&
```

Correction/modification of hop.F90 for bias correction in CY48R1



/perm/pan/git/ifs-source/pan_CY47R1.M_clradlid_assim.IFS-1604/ifs/op_obs/hop.F90

```
    ! Generic VarBC bias correction
    PBIAS(:,:)=0.0_JPRB
    PNDBIAS(:,:)=0.0_JPRB
    IF(LVARBC) THEN

      CALL OBSOP_VARBC(ROBHDR,ROBODY,&
        & YDVARBC,YDGP5,YDSET%MXBDY,PHOFX,YLVARBC_EXTRA_PRED,&
!       & LDRSONDE_BODY = (IOBSTYPE==NTEMP) .AND. ANY(VARNOS_TO_PROCESS(:,:) == VARNO%T),&
!       & LD_MWAVE=LL_MWAVE,PVERTP=ZVERTP,PBIAS=PBIAS,PNDBIAS=PNDBIAS,KSET=YDSET%ID,LDTL=LLTL,

    ENDIF

    ! Rogue winds corrected as a bias correction
    IF(LL_AIREP .AND. LECMWF .AND. LAIRVSWITCH .AND. LLDIRECT) THEN
      CALL SWITCH_VWIND(ROBHDR,ROBODY,YDGP5,YDSET,PHOFX,PBIAS)
    ENDIF

  ENDIF

! IF(LL_AEOLUS .AND. LECMWF .AND. LLDIRECT) THEN
!   ! Enforce PBIAS for Aeolus if direct
!   ! since providing own bias correction via biascorr@body.
!   ! Should not be done if Aeolus uses VarBC
    CALL SET_PBIAS_AEOL(ROBODY,PBIAS)
  ENDIF

! IF(LL_AEOLUS .AND. LECMWF .AND. LLDIRECT) THEN
!   ! Enforce PBIAS for Aeolus if direct
!   ! since providing own bias correction via biascorr@body.
!   ! Should not be done if Aeolus uses VarBC
!   CALL SET_PBIAS_AEOL(ROBODY,PBIAS)
  ENDIF

  ! AJGDB There follow 4 different ways of getting basic output from hop (apart from
  ! standard ZHOFX!). Some rationalisation might be in order.
```

../CY48R1/hop.F90

```
   ! Generic VarBC bias correction
   PBIAS(:,:)=0.0_JPRB
   PNDBIAS(:,:)=0.0_JPRB
   IF(LVARBC) THEN

     CALL OBSOP_VARBC(ROBHDR,ROBODY,&
       & YDVARBC,YDGP5,YDSET%MXBDY,PHOFX,YLVARBC_EXTRA_PRED,&
!      & LDRSONDE_BODY = (IOBSTYPE==NTEMP) .AND. ANY(IVARNOS_TO_PROCESS(:,:) == VARNO%T),&
!      & PVERTP=ZVERTP,PBIAS=PBIAS,PNDBIAS=PNDBIAS,KSET=YDSET%ID,LDTL=LLTL,LDAD=LLAD)

   ENDIF

   ! Rogue winds corrected as a bias correction
   IF(LL_AIREP .AND. LECMWF .AND. LAIRVSWITCH .AND. LLDIRECT) THEN
     CALL SWITCH_VWIND(ROBHDR,ROBODY,YDGP5,YDSET,PHOFX,PBIAS)
   ENDIF

 ENDIF

! IF(LL_AEOLUS .AND. LLDIRECT) THEN
!   ! Enforce PBIAS for non-varbc obs if direct


   CALL SET_PBIAS_AEOL(ROBODY,PBIAS)
 ENDIF

! IF(LL_CLRADLID .AND. LLDIRECT .AND. .NOT. LLCVARBC_ONLY) THEN
!   ! Enforce PBIAS for non-varbc obs if direct
!     DO JBODY=1,ROBODY%MAX_DATUM_PER_REPORT
!       DO JOBS = 1,YDGP5%NDLEN
!         IF(JBODY > ICMBDY(JOBS)) CYCLE
!         IBODY = ROBODY%REPORT_START_ROW(JOBS)+(JBODY-1)
!         IF ((ROBODY%BODY%VARNO(IBODY) == VARNO%CLOUD_RADAR_REFLECTIVITY) .OR. &
!           & (ROBODY%BODY%VARNO(IBODY) == VARNO%LIDAR_CLOUD_BACKSCATTER)) THEN
!           IF(LSCREEN) ROBODY%BODY%BIASCORR_FG(IBODY) = ROBODY%BODY%BIASCORR(IBODY)
!           PBIAS(JOBS,JBODY)=ROBODY%BODY%BIASCORR_FG(IBODY)
!         ENDIF
!       ENDDO
!     ENDDO
 ENDIF

 ! AJGDB There follow 4 different ways of getting basic output from hop (apart from
 ! standard ZHOFX!). Some rationalisation might be in order.
```

Figure 6.1: Example of changes in CY48R1: (top) update of hop.F90 routine, operator routine for all types of observations, from the cycle CY47R1 to the cycle CY48R1 and (bottom) correction/modification in hop.F90 to account for bias correction of cloud radar and lidar observations when using OOPS-IFS system in CY48R1.

6.2.1 Technical testing

Technical testing is the first step always needed after updating any development to the higher model cycle. In the case of our assimilation developments, firstly such test is performed to ensure that the cloud radar and lidar observations are recognized correctly and are passed through to the Observation Data Base (ODB) properly. Secondly, after getting correct data in the ODB, a basic test of the 4D-Var minimization is made in order to check the convergence of the system with the new observations and operators.

The following steps of the whole data assimilation process, all of them described in more details in the previous sections, Section 3 and 4, were checked and successfully completed:

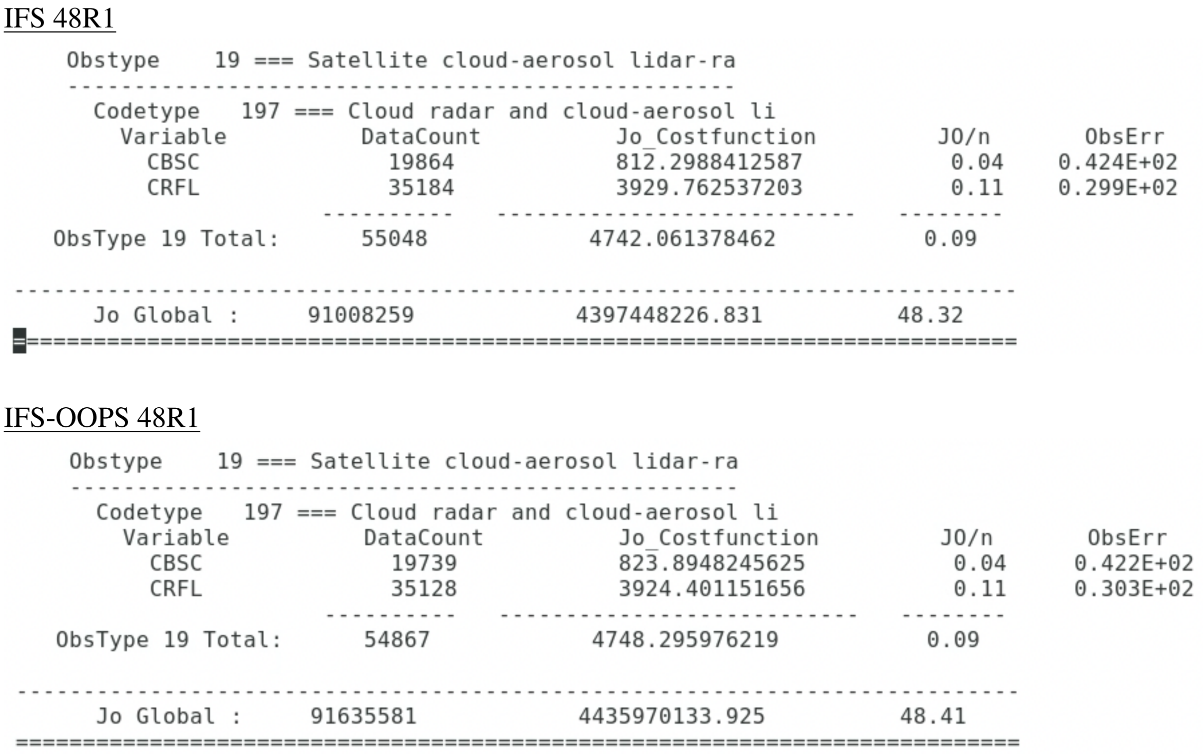
1. observation pre-screening processing,
2. data conversion from BUFR (Binary Universal Form for the Representation of meteorological data) format to ODB,
3. reading observations into the main 4D-Var assimilation tasks from ODB,
4. high-resolution trajectory run,
5. blacklisting procedure to reject observations based on the defined criteria,
6. updating ODB by the background departures together with quality information for observations after performing screening,
7. 4D-Var minimization process,
8. post-processing of observations and archiving them in MARS (Meteorological Archiving and Retrieval System of ECMWF).

Obviously, to accomplish all the technical testing is a rather complex process as it often involves debugging parts of the system that are not working properly because of merging issues or changes in the higher model cycle which need to be adapted for the new observations. For the cycle CY48R1, this process was even more complex, because the testing had to be performed not only for the original IFS system, but also for the new 4D-Var assimilation framework to be used for ECMWF's IFS, OOPS. As mentioned in Section 6.1 quite few changes and some corrections to the code were required to make it performing properly using the cloud radar and lidar observations in CY48R1.

The technical test indicated that the observation cost function for cloud radar and lidar observations is similar for the original IFS and the OOPS-IFS systems as expected (Fig. 6.2). This function is also progressively decreased through 4D-Var iteration process (using the both systems) as required (not shown here) and these new observations are properly stored in ODB and MARS. This is important sanity check before we can progress with more scientific evaluation based on CY48R1. The first testing results of 4D-Var with cloud radar and lidar observations in the model cycle CY48R1 are summarized in Section 6.2.2.

6.2.2 *First testing results of 4D-Var with cloud radar and lidar observations in CY48R1*

As a first test of the ported assimilation system to CY48R1, a single-cycle analysis experiment is performed at that cycle where observations of CloudSat radar reflectivity and CALIPSO lidar backscatter are fed to the system in addition to all the regularly assimilated observations. The output of this experiment is compared to an experiment using the model cycle CY46R1 used in our extended experimentations for optimisation of observation impact on analysis (Fielding and Janisková, 2020) and on forecast (Fielding and Janisková, 2021a). Two sets of experiments have to be run in CY48R1, one experiment using the original IFS system and another experiment using the new OOPS-IFS system. A visual assessment of this test is demonstrated on the portion of orbital track representing several model profiles containing radar and lidar observations assimilated in the data assimilation window between 21:00 UTC 31 July 2007 and 09:00 UTC 1 August 2007 (Figs. 6.3 - 6.6).



IFS 48R1

```
Obstype    19 === Satellite cloud-aerosol lidar-ra
--------------------------------------------------
  Codetype   197 === Cloud radar and cloud-aerosol li
    Variable          DataCount          Jo_Costfunction          JO/n       ObsErr
      CBSC              19864          812.2988412587           0.04    0.424E+02
      CRFL              35184          3929.762537203           0.11    0.299E+02
                   ----------    ---------------------------    --------
 ObsType 19 Total:       55048           4742.061378462         0.09

----------------------------------------------------------------------------
      Jo Global :     91008259          4397448226.831          48.32
============================================================================
```

IFS-OOPS 48R1

```
Obstype    19 === Satellite cloud-aerosol lidar-ra
--------------------------------------------------
  Codetype   197 === Cloud radar and cloud-aerosol li
    Variable          DataCount          Jo_Costfunction          JO/n       ObsErr
      CBSC              19739          823.8948245625           0.04    0.422E+02
      CRFL              35128          3924.401151656           0.11    0.303E+02
                   ----------    ---------------------------    --------
 ObsType 19 Total:       54867           4748.295976219         0.09

----------------------------------------------------------------------------
      Jo Global :     91635581          4435970133.925          48.41
============================================================================
```

Figure 6.2: Screen-shots of ifstraj outputs showing the cost function containing cloud radar reflectivity (CRFL) and lidar cloud backscatter (CBSC) as used in: (top) the original IFS 4D-Var system and (bottom) IFS-OOPS system in the cycle CY48R1.

To test that the observations have entered the ODB correctly, the superobbed CloudSat radar reflectivity (Fig. 6.3) and CALIPSO lidar backscatter (Fig. 6.5) for cycles CY46R1 and CY48R1 are compared against each other. Note that even though observations are very similar, they are not identical as one would expect, because both CloudSat and CALIPSO data were updated to the latest versions since the previous cycle (CY46R1) testing: to R05 version from R04 version for CloudSat and to V4.2 version from V3.0 version for CALIPSO. Both, CloudSat and CALIPSO, had substantial differences in their processing, particularly the cloud mask for CloudSat. Obviously, observations are bit-identical in CY48R1 (Fig. 6.4 and 6.6) whether using the original IFS or OOPS-IFS system as their processing is the same in that case.

The first-guess (FG) values, model equivalent to observations, as well as the corresponding observation errors are not the same for the cycles CY46R1 and CY48R1, since the model, providing input to observation operators, is changed between these cycles. However, the FG values and also observation errors are broadly similar (Fig. 6.3, 6.5). The screening for both observation types is also similar. All these values for the run using either original IFS system or the run using OOPS-IFS system in CY48R1 (Fig. 6.4 and 6.6) are very close to each other, although not identical. Whilst both runs are fed with the same input data, using the different frameworks under which 4D-Var runs, leads to small differences which, however, in the longer integrations should not be statistically significant, i.e. neither better nor worse than the other.

Similarities in the model equivalent to observations between the model cycle CY46R1 and cycle CY48R1 is also demonstrated on Fig. 6.7 displaying the probability density function (PDF) distribution of the FG departures of the model from both, the CloudSat cloud radar reflectivity and the CALIPSO cloud lidar backscatter. Especially, after applying the bias correction and screening of observations, PDFs are quite

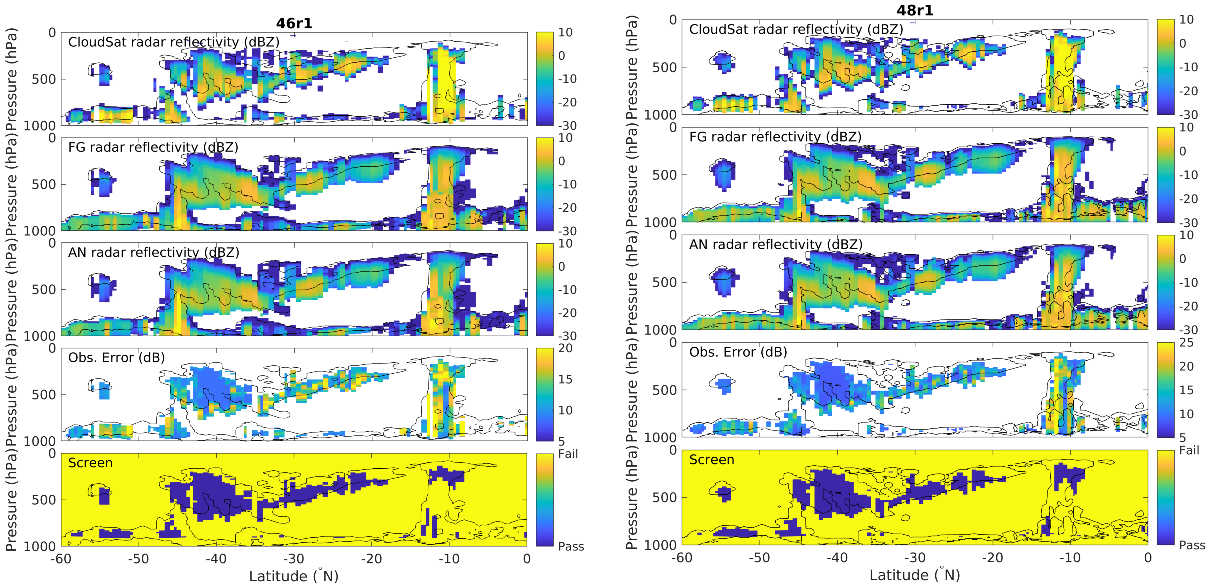


Figure 6.3: Cross-sections of cloud radar reflectivity related variables corresponding to the portion of orbital track on 31 July 2007 at 21:00 UTC. Panels show from top to bottom: observed CloudSat radar reflectivity (dBZ), model equivalent (FG) radar reflectivity using the model background (dBZ), model equivalent (AN) radar reflectivity using the model analysis from the assimilation experiment using all observations and radar and lidar, the observation error (dB) assigned to the observations and screening value (blue colour = observations passed). The cross-sections on the left panels correspond to outputs from the model cycle CY46R1, while the right panels are for the cycle CY48R1, both using the original IFS system.

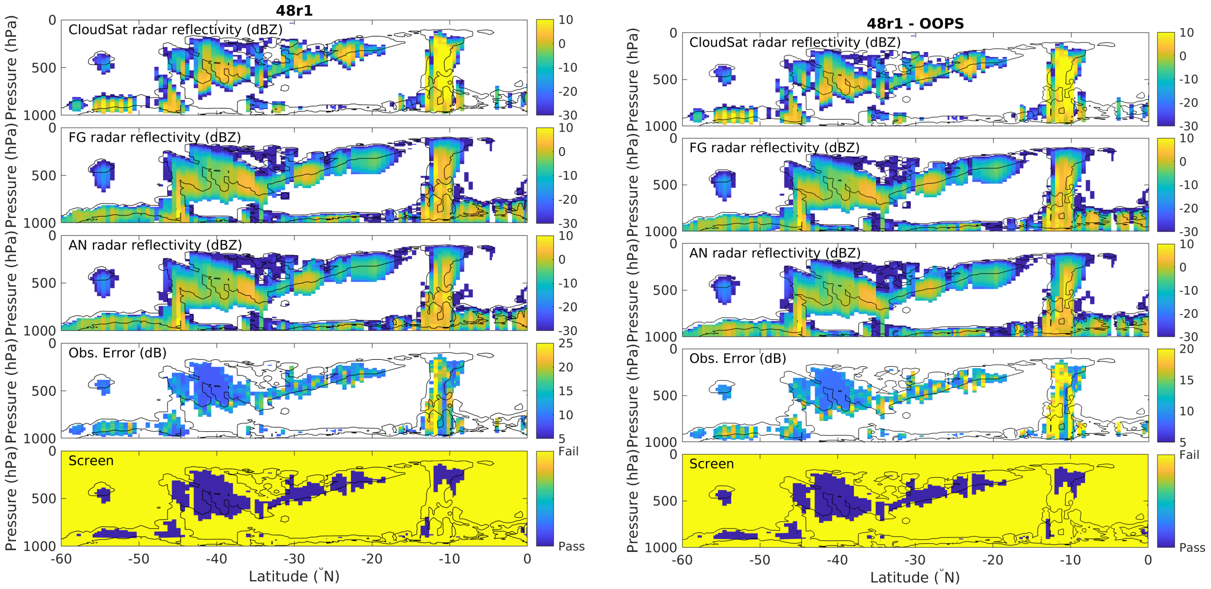


Figure 6.4: Same as Fig. 6.3, but the cross-sections on the left panels correspond to outputs from the model cycle CY48R1 using the original IFS system, while the right panels are from the cycle CY48R1 using the OOPS-IFS system.

similar. The PDF distributions from the model cycle CY48R1 using either the original IFS system or the OOPS IFS system are nearly identical (Fig. 6.8).

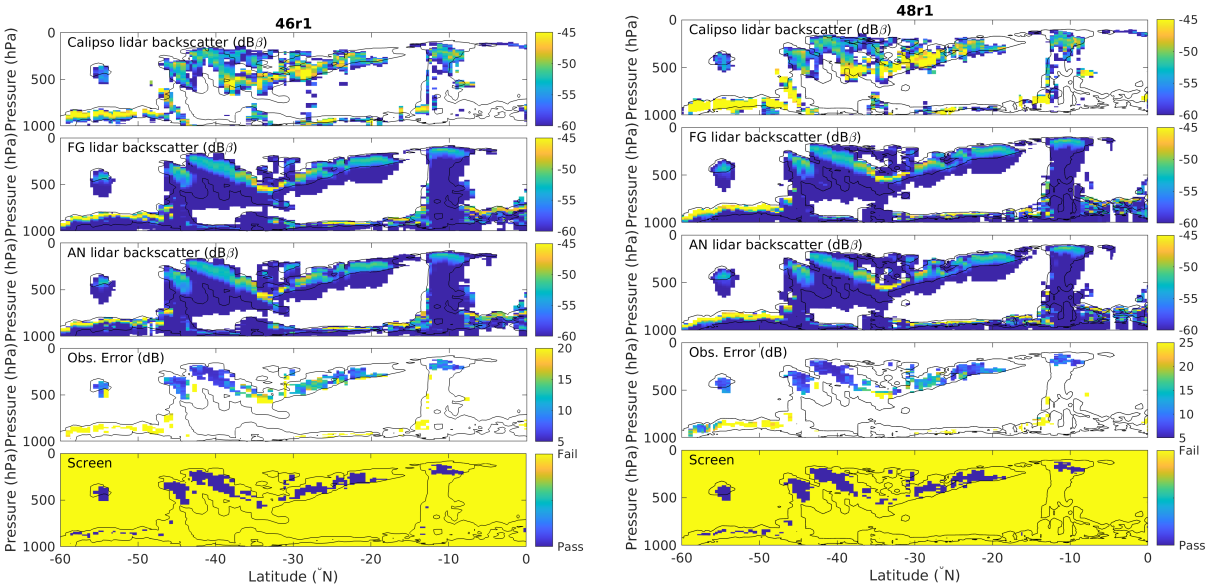


Figure 6.5: Same as Fig. 6.3, but for cloud lidar backscatter and CALIPSO attenuated backscatter (dBβ).

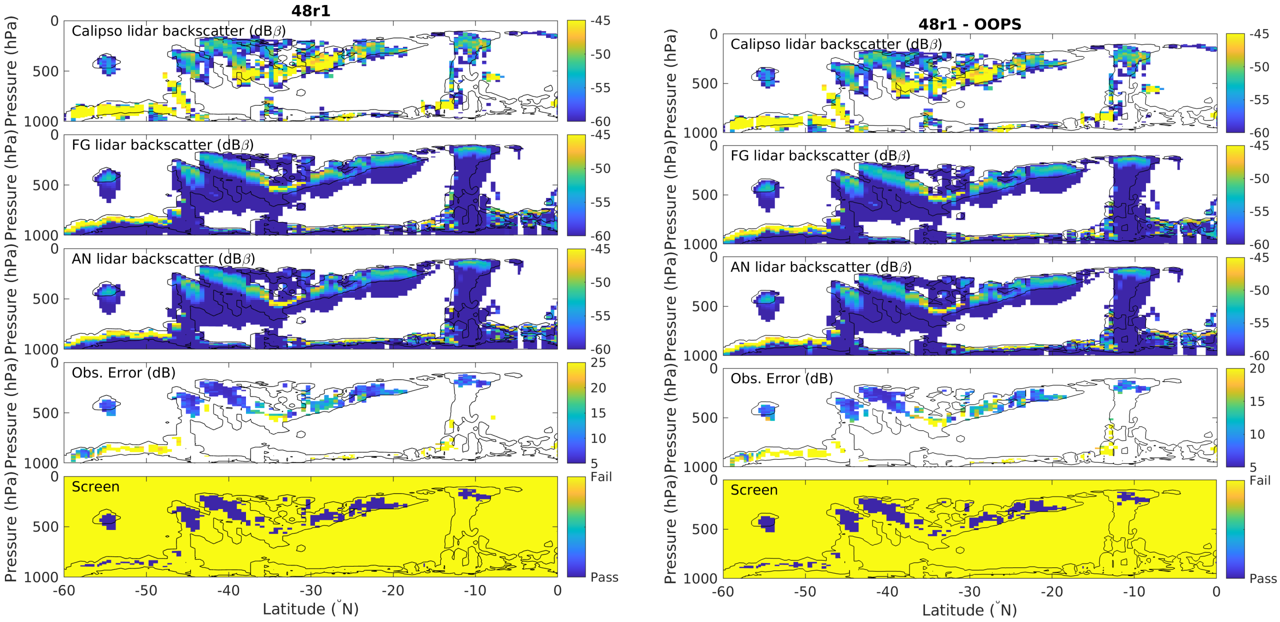


Figure 6.6: Same as Fig. 6.5, but the cross-sections on the left panels correspond to outputs from the model cycle CY48R1 using the original IFS system, while the right panels are from the cycle CY48R1 using the OOPS-IFS system.

In the next step, the difference between the first guess and the analysis of the model equivalent for the observations was assessed (Fig. 6.3 - 6.4). For both, radar reflectivity and lidar backscatter, the fit of the analysis to the observations is improved compared to the first guess in the new cycle, CY48R1, as it was also the case in CY46R1. The small differences in the FG values, observation error and screening is obviously leading to the differences in analysed values consequently.

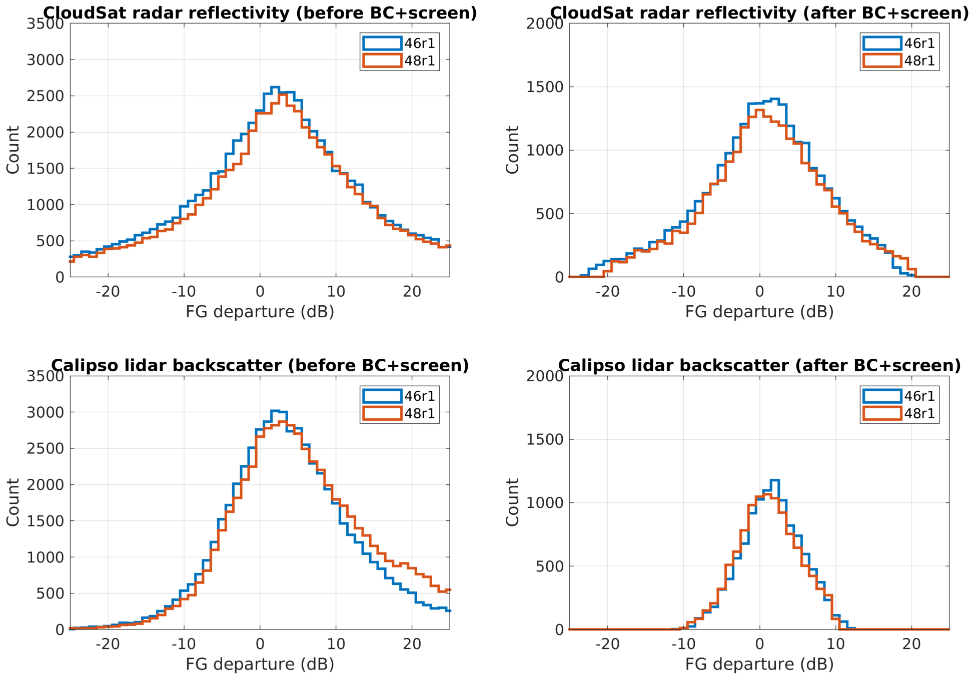


Figure 6.7: Probability density function (PDF) distribution of the first guess departures of the model from: (on top) CloudSat radar reflectivity (left) before and (right) after bias correction and screening; (on bottom) CALIPSO lidar backscatter (left) before and (right) after bias correction and screening. The PDF distributions in blue are from the model cycle CY46R1, while in red from the cycle CY48R1, both using the original IFS system.

As a final check of the consistency of the analysis between the cycles CY46R1 and CY48R1, Fig. 6.9 shows the analysis increments of both radar and lidar observations, as well as cloud related variables such as temperature and relative humidity. Comparing these two cycles, one can clearly see that there are a lot of similarities in obtained increments, except in temperature increments, where differences are large. Small differences in increments between these two cycles for radar reflectivity, lidar backscatter, relative humidity and cloud fraction can be explained by differences in the model code between the cycles. However, the differences in temperature increments seemed to be alarming at the time we obtained them. More detailed investigation and discussions with colleagues from the data assimilation team led to an explanation of such different increments. It was realized that they are a consequence of using different background error correlations in the vertical for temperature, whose climatological contribution was significantly changed between cycles as illustrated in Fig. 6.11. The correlation is almost positive only and the distribution is rather narrow in the cycle CY48R1, contrary to the much broader structure, with positive and negative values used at the cycle CY46R1.

The differences in analysis increments between the runs using different framework for 4D-Var in the cycle CY48R1 are small, but in general very similar, as illustrated in Fig. 6.10.

Overall, the performance of 4D-Var system assimilating cloud radar and lidar observations on the top of regularly used observations is not very different for the cycle CY48R1 compared to CY46R1. The differences can be attributed to the model changes over two cycles. These first test results are reassuring as they indicate that cloud radar and lidar assimilation was successfully ported to CY48R1.

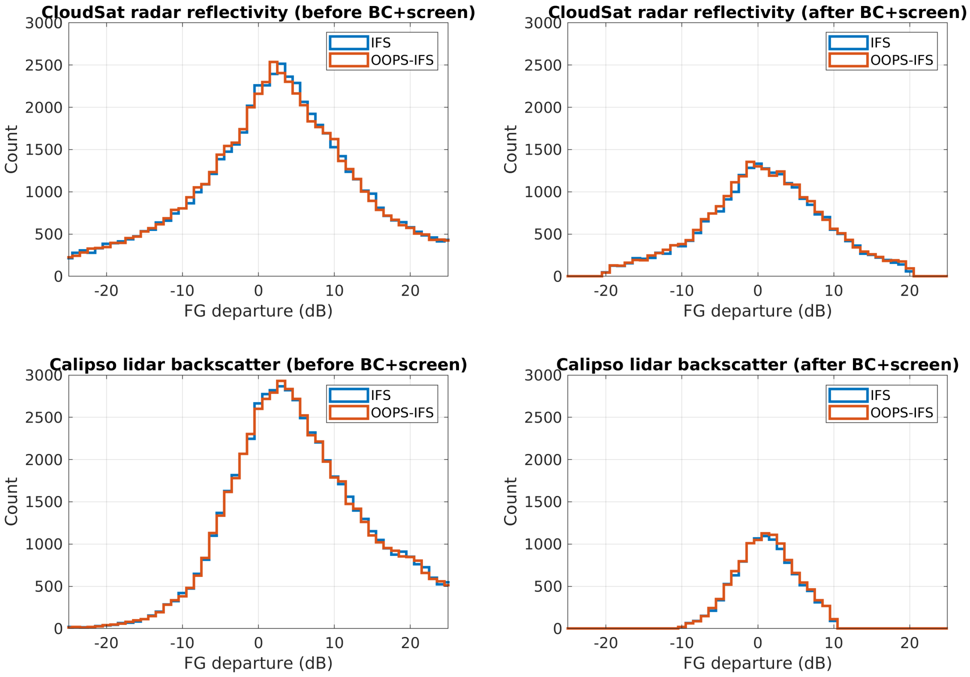


Figure 6.8: Same as Fig. 6.7, but the PDF distributions in blue are from the model cycle CY48R1 using the original IFS system, while in red from the model cycle CY48R1 using the OOPS IFS system.

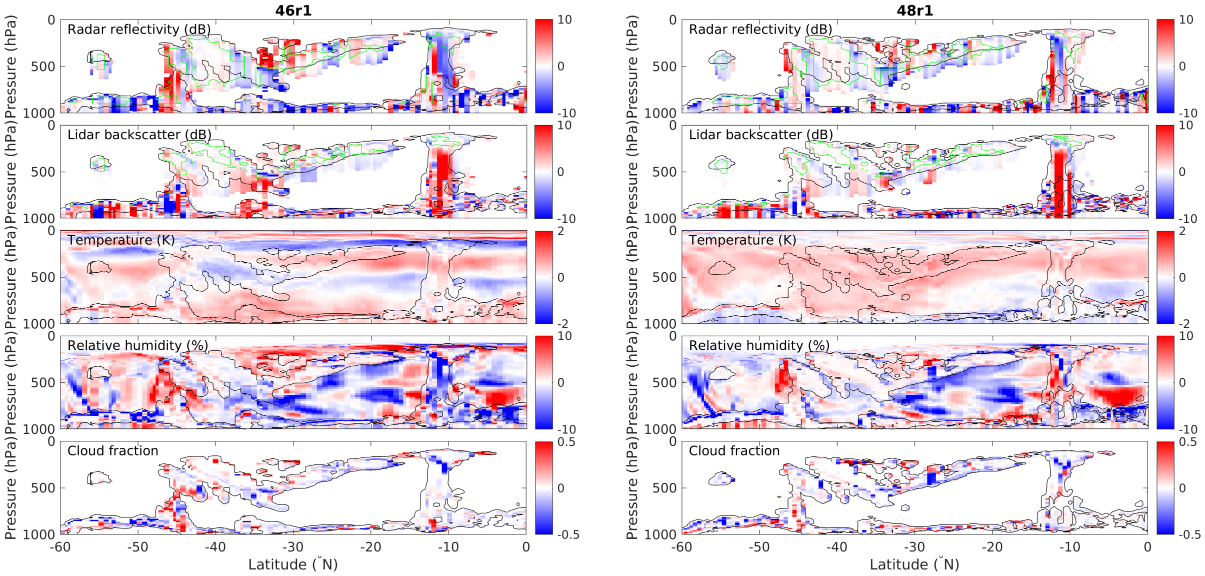


Figure 6.9: Analysis increments for the the portion of orbital track on 31 July 2007 at 21:00 UTC as in Fig. 6.3 and Fig. 6.5 for the model cycle CY46R1 (left column) and CY48R1 (right column), both using the original IFS system. Panels from top to bottom show model-level increments of: radar reflectivity (dB), lidar backscatter (dB), temperature (K), relative humidity (%) and cloud fraction.

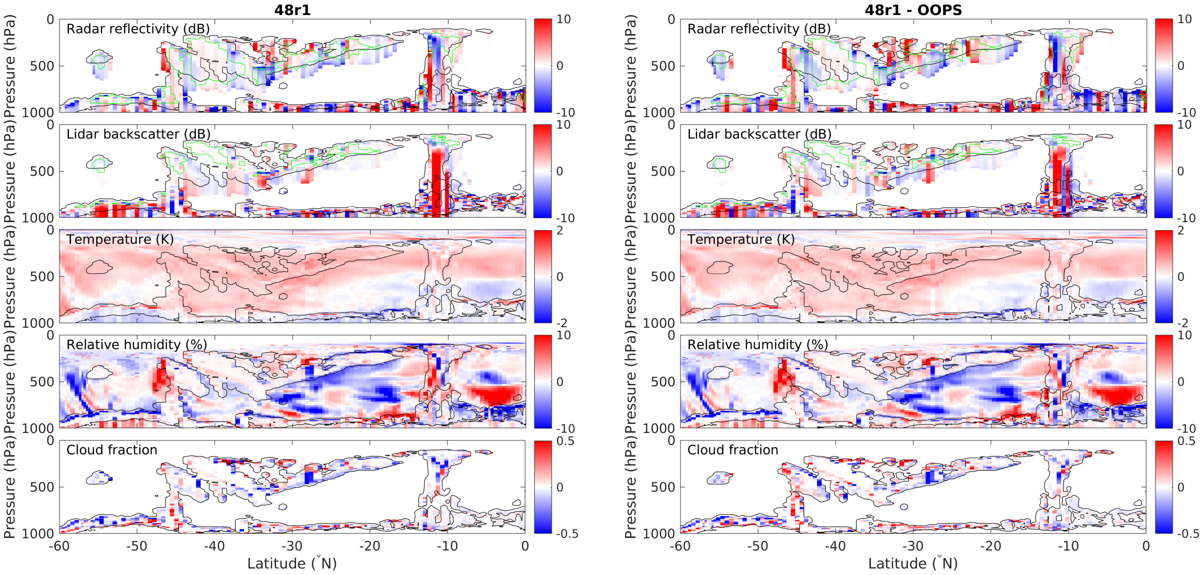


Figure 6.10: Same as Fig. 6.9, but the cross-sections on the left panels correspond to outputs from the model cycle CY48R1 using the original IFS system, while the right panels are from the cycle CY48R1 using the OOPS-IFS system.

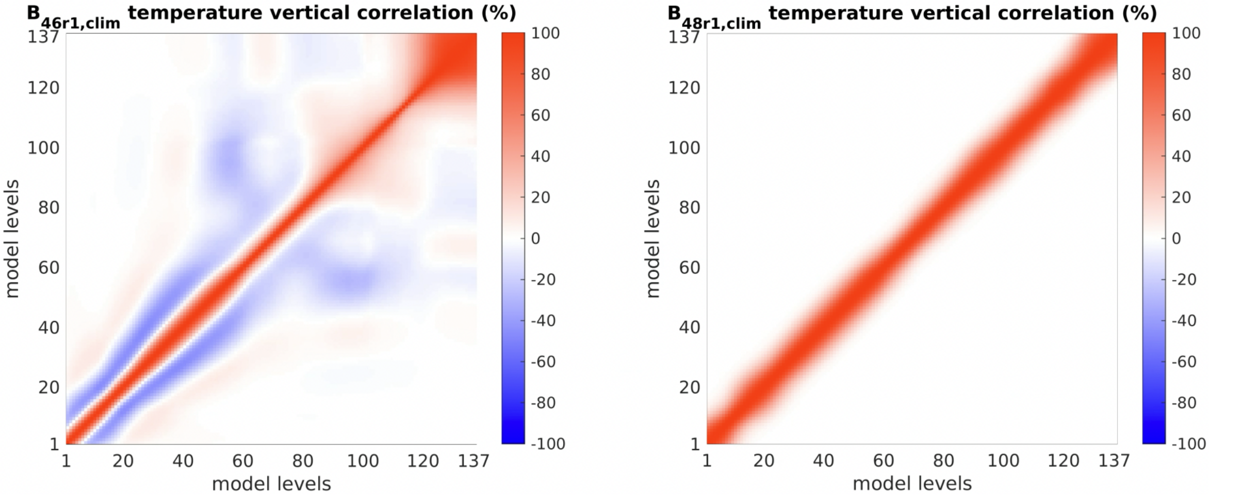


Figure 6.11: Correlation between background temperature error at different model levels based on the cycle (left) CY46R1 and (right) CY48R1 climatologies.

7 Summary

Adaptation and maintenance of our assimilation system for cloud radar and lidar observations were carried out in several stages during the whole project.

Stage 1:

During the first period, the assimilation system developed in the previous project (Janisková and Fielding, 2018) using CY43R1 was brought to CY46R1, which was the current cycle used at the beginning of project in October 2019. This represented a big effort to move and adopt over 300 routines and scripts over few cycles. There were not only many changes in the routines which needed to be modified to take into account cloud radar and lidar observations, but also the new routines for these observations needed to be adapted to CY46R1. Unfortunately all the porting had to be done manually, routine by routine, without using some developing/version controlling software. This was the case because the software used by CY43R1, Perforce, was replaced by Git in CY46R1.

All the subroutines and scripts required for cloud radar and lidar assimilation have been adapted during that period. After successful compilation and a lot of debugging needed to make all previous developments working in CY46R1, first, mainly technical tests were done. The results of these tests were presented in this report demonstrating that the updated system works. Successful update of cloud radar and lidar assimilation to CY46R1 was very important step since it helps to secure all past development for the future. Without this, operational use of cloud radar and lidar observations could not be imagined and we would risk that hard work over many years would be lost.

Stage 2:

During the second period of WP-1000, the assimilation system development for cloud radar and lidar observations was migrated from the model cycle CY46R1 to the cycle CY47R1. Regular updating of the satellite cloud radar and lidar capability of the 4D-Var assimilation system and regular testing is crucial to ensure its proper functionality and to keep track with the ECMWF model upgrades.

Basic testing of the developments in CY47R1 has been done. This involved a single-cycle analysis experiment where observations of CloudSat radar reflectivity and CALIPSO lidar backscatter were included to the assimilation system in addition to all the regularly assimilated observations. A comparison of the output of this experiment with an experiment using the previous model version indicated that the performance the of 4D-Var system assimilating these new observations is similar for CY46R1 and CY47R1. These first test results were reassuring as they indicated that cloud radar and lidar assimilation was successfully ported to CY47R1.

The updated and tested CY47R1 branch was subsequently merged into the official model cycle as part of the preparations to move these developments to the model cycle CY48R1 in order to appear passively in the operational code in 2022. Therefore, the CY47R1 branch also had to be tested for the bit reproducibility of the system when cloud radar and lidar observations are switched off (i.e. to insure that passive observations do not modify any integration of the operational code).

Stage 3:

During the third period, extensive assimilation experimentation was carried out with the aim of optimizing impact of cloud radar reflectivity and lidar backscatter on analysis and the subsequent forecast. Any issues discovered as part of such testing were addressed and the cloud radar and lidar assimilation system was updated based on them. The system was also updated to include the new developments since the project started: (i) a novel triple-column forward model taking into account subgrid condensate variability in the computation of model equivalent to the observations together with its tangent-linear and adjoint versions, (ii) optional specification of vertical correlations in the observation error, (iii) updated bias correction scheme. The developments and modifications to the cloud radar and lidar assimilation system required in order to account for additional observations, such as Doppler velocity, cloud extinction and Rayleigh backscatter, have also been included in the IFS system. Those included (i) observation operators for the above additional observations, (ii) tangent linear and adjoint versions of the new operators, (iii) observation handling of the new observation types into BUFR format, (iv) necessary ODB developments.

Stage 4:

During the fourth period of WP-1000, the assimilation system for cloud radar and lidar assimilation was tested in the most recent IFS model cycle, CY48R1. To ensure the correct functioning of the system and make its performance comparable with the cycle CY46R1, where our extensive experimentations and adjustments were done during this project as described in WP-2000, WP-3000, WP-4000 and WP-5000 (Fielding and Janisková, 2020, 2021a,b, 2022), some modifications and corrections were required. Some updates were necessary to account for developments, modifications and corrections from the cycle CY46R1 which were not yet ready or known when porting the system for the new observations to an intermediate cycle CY47R1. These modifications were rather straightforward and easy to adapt comparing to some changes required to make the system working when running IFS 4D-Var data assimilation under the framework of the OOPS.

Basic testing of the developments in CY48R1 have been done. This involved a single-cycle analysis experiment where observations of CloudSat radar reflectivity and CALIPSO lidar backscatter were included to the assimilation system in addition to all the regularly assimilated observations. A comparison of the output of this experiment with an experiment using older model version indicated that cloud radar and lidar assimilation was successfully ported to CY48R1. Time permitting during this project, 4D-Var experiments using these new observations will be extended for the longer period to re-check impact of these observations on analysis and the subsequent forecast in this more recent cycle.

Acknowledgments

The authors would like to thank all those who helped during debugging process which led to successful migration of the cloud radar and lidar assimilation first to the model cycle CY46R1, in particular to Peter Lean, Tomáš Král, Philippe Lopez, Giovanna Di Chiara, Oliver Marsden and Paul Creswell and later to the model cycle CY47R1, in particular to Tomáš Král, Gabor Radnoti, Peter Lean and Andrew Dawson. Peter Lean is also thanked or his advice on making modifications to the ODB, Marcin Chrust for help with OOPS modifications, Elias Holm for understanding differences in background errors between cycles and Philip Browne for help in prepIFS settings for the surface analysis for older dates in CY48R1.

List of Acronyms

4D-Var	Four-Dimensional Variational Assimilation
AD	ADjoint
AN	Analysis
ATLID	ATmospheric LIDar
BUFR	Binary Universal Form for the Representation of meteorological data
CALIOP	Cloud-Aerosol Lidar with Orthogonal Polarization
CALIPSO	Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation
CloudSat	NASA's cloud radar mission
CCMA	Compressed CMA (used for active observations after screening)
CMA	Central Memory Array (used for observations at ECMWF)
CPR	Cloud Profiling Radar
EarthCARE	Earth, Clouds, Aerosols and Radiation Explorer
ECFS	ECMWF's File Storage system
ecFlow	ECMWF's work-flow manager enabling to run large number of programs
ecFlowview	graphical user interface to display the status of experiment tasks
ECMA	Extended CMA (used for all observations before screening)
ECMWF	European Centre for Medium Range Weather Forecasts
ESA	European Space Agency
FG	First Guess
HSRL	High-Spectral Resolution Radar
IFS	Integrated Forecasting System of ECMWF
I/O	Input / Output
MARS	Meteorological Archiving and Retrieval System
NWP	Numerical Weather Prediction
OBS	OBServations
ODB	Observation Data Base
OOPS	Object-Oriented Prediction System
PDF	Probability Density Function
prepIFS	user unterface to prepare IFS experiment
SQL	Structured Query Language
T95,T159,T399	Model grid with spectral truncation T95, T159, T399
TL	Tangent Linear
UTC	Universal Time Coordinated
VarBC	Variational Bias Correction
WMO	World Meteorological Organization

References

Courtier, P., J.-N. Thépaut, and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-Var, using an incremental approach, *Q. J. R. Meteorol. Soc.*, **120**, 1367–1387.

English, S., D. Salmon, M. Chrust, A. Geer, E. Holm, S. Massart, M. Hamrud, R. Stappers, and R. El Khatib, 2017: Progress with running IFS 4D-Var under OOPS, ECMWF Newsletter **153**, 13–14.

Fielding, M. and M. Janisková, 2020: Optimisation of observation impact on analysis, WP-2000 report for the project Preparations for EarthCARE Assimilation - Radar and Lidar Cloud Observations (PEARL Cloud) to ESA ESTEC, contract 4000128669/19/NL/CT, 44 pp.

Fielding, M. and M. Janisková, 2021a: Optimisation of observation impact on forecast, WP-3000 report for the project Preparations for EarthCARE Assimilation - Radar and Lidar Cloud Observations (PEARL Cloud) to ESA ESTEC, contract 4000128669/19/NL/CT, 34 pp.

Fielding, M. and M. Janisková, 2021b: Assimilation system development for additional observations, WP-4000 report for the project Preparations for EarthCARE Assimilation - Radar and Lidar Cloud Observations (PEARL Cloud) to ESA ESTEC, contract 4000128669/19/NL/CT, 30 pp.

Fielding, M. and M. Janisková, 2022: Additional observations preliminary assessment, WP-5000 report for the project Preparations for EarthCARE Assimilation - Radar and Lidar Cloud Observations (PEARL Cloud) to ESA ESTEC, contract 4000128669/19/NL/CT, 32 pp.

Fielding, M. D. and M. Janisková, 2017: Observation quality monitoring and pre-processing, WP-2000 report for the project Operational Assimilation of Space-borne Radar and Lidar Cloud Profile Observations for Numerical Weather Prediction, 4000116891/16/NL/LvH, pp.

Fisher, M., 2004: Generalized frames on the sphere, with application to the background error covariance modelling, *Proc. Seminar on Recent Developments in Numerical Methods for Atmospheric and Ocean Modelling, Reading, UK, ECMWF*, pp. 87–102.

Gauthier, P. and J.-N. Thépaut, 2001: Impact of the digital filters as a weak constraint in the preoperational 4D-Var assimilation system of Météo-France, *Mon. Weather Rev.*, **129**, 2089–2102.

Janisková, M., S. Di Michele, and E. Martins, 2014: Support-to-Science-Elements (STSE) Study - EarthCARE Assimilation, ESA Contract Report on Project 4000102816/11/NL/CT, 225 pp.

Janisková, M. and M. Fielding, 2018: Operational Assimilation of Space-borne radar and Lidar Cloud Profile Observations for Numerical Weather Prediction, Contract report to the European Space Agency 4000116891/16/NL/LvH, 164 pp.

Janisková, M. and P. Lopez, 2013: Linearized physics for data assimilation at ECMWF, in *S.K. Park and L. Xu (Eds), Data Assimilation for Atmospheric, Ocean and Hydrological Applications (Vol II), Springer-Verlag Berline Heidelberg*, pp. 251–286, doi:10.1007/978–3–642–35088–7–11.

Janisková, M., O. Stiller, S. Di Michele, R. Forbes, J.-J. Morcrette, M. Ahlgrimm, P. Bauer, and L. Jones, 2010: QuARL - Quantitative Assessment of the Operational Value of Space-Borne Radar and Lidar Measurements of Cloud and Aerosol Profiles, ESA Contract Report on Project 21613/08/NL/CB, 329 pp.

Liebe, H. J., 1985: An updated model for milimeter wave propagation in moist air, *Radio Science*, **20(5)**, 1069–1089.

Liebe, H. J., P. Rosenkranz, and G. A. Hufford, 1992: Atmospheric 60-GHz oxygen spectrum - new laboratory measurements and line parameters, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **48**, 629–643.

Rabier, H., F.and Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons, 2000: The ECMWF operational implementation of four-dimensional variational assimilation. Part I: Experimental results with simplified physics, *Q. J. R. Meteorol. Soc.*, **126**, 1143–1170.

Stephens, G., D. Vane, R. Boain, G. Mace, K. Sassen, Z. Wang, A. Illingwort, E. O'Connor, W. Rossow, and S. Durden, 2002: The CloudSat mission and the A-train, *Bull. Am. Meteorol. Soc.*, **83(12)**, 1771–1790.

Veerse, F. and J. N. Thépaut, 1998: Multiple-truncation incremental approach for four-dimensional variational data assimilation., *Q. J. R. Meteorol. Soc.*, **124**, 1889–1908.

Winker, D., M. Vaughan, A. Omar, Y. Hu, K. Powell, Z. Liu, W. Hunt, and S. Young, 2009: Overview of the CALIPSO mission and CALIOP data processing algorithms, *J. Atmos. and Ocean. Tech.*, **26(7)**, 2310–2323.