IFS DOCUMENTATION – Cy48r1
Operational implementation 27 June 2023

PART II: DATA ASSIMILATION

REVISION HISTORY

Changes since Cy47r3

- Chapter 2, comprehensive revision due the roll out of the Object Oriented Prediction System (OOPS).
- Chapter 6, update to section: Calculation of eigenvalues and eigenvectors of the Hessian.
- Chapter 6, update to section: The preconditioner.

Changes since Cy45r1

- Chapter 9, update to section: SEKF soil anlaysis based on the EDA to compute the Jacobians, and SMOS soil moisture assimilation.

Changes since Cy43r1

- Chapter 6, minor revision to sections on Calculation of forecast-error variances, Diagnosis of background error variances through the EDA.

Changes since Cy41r2

- Chapter 5, minor revisions to all sections.
- Chapter 6, minor revisions to section: Diagnosis of background error variances through the EDA; Diagnosis of online background error covariances.
- Chapter 8, update to section: The 24-hr forecast error contribution to observations.
- Chapter 9, minor revisions to sections: 2D Optimal Interpolation (screen-level analysis and snow analysis).

Chapter 1

Overview

Table of contents

1.1   INTRODUCTION

This documentation of 4D-Var serves as a scientific guide to the data assimilation code within ECMWF's Integrated Forecasting System (IFS). The documentation is divided into a number of chapters. This first chapter provides an overview of ECMWF's data assimilation system, a brief history of its evolution and a description of the basic scientific formulation.

The second chapter describes the practical implementation of the multi-resolution incremental method for 4D-Var data assimilation (the multi-resolution version has been used operationally since January 2003, Cy25r3), and the solution algorithm including minimization and preconditioning. The tangent linear physics that is an important part of the variational algorithm is described in Chapter 3. Thereafter follows a description of the background term (Chapter 4) and a chapter on observation-related parts of the data assimilation algorithm (Chapter 5) though the main documentation of observations is to be found in Part 1. Chapter 6 deals with the computation and cycling of background and analysis errors including the recent use of variances and covariances based on the Ensemble of Data Assimilations (EDA). Chapter 7 is on control of gravity waves in the minimization cycle. Diagnostic tools for investigation of the performance of the assimilation system are described in Chapter 8. Chapter 9 describes the land surface analysis, including the screen level parameters, the soil analysis and the snow analysis, Chapter 10 describes the sea surface temperature and sea-ice analysis. Finally Chapter 11 provides summary information about the main unix-scripts and important files, and a schematic of the data flow between the various jobs steps that constitute a data assimilation cycle.

1.2   SCIENTIFIC PUBLICATIONS

The scientific description of 3D/4D-Var has been published in a series of papers in the *Quarterly Journal of the Royal Meteorological Society* (QJRMS), in ECMWF workshop proceedings and Technical Memoranda over the years. The incremental formulation was introduced by Courtier *et al.* (1994). The ECMWF implementation of 3D-Var was published in a three-part paper by Courtier *et al.* (1998), Rabier *et al.* (1998) and Andersson *et al.* (1998). The original observation operators for conventional data can be found in Vasiljevic *et al.* (1992). The pre-operational experimentation with 4D-Var has been documented in three papers by Rabier *et al.* (2000), Mahfouf and Rabier (2000) and Klinker *et al.* (2000). The background term has been published by Derber and Bouttier (1999), with more recent developments described by Fisher (2003). EDA-based background error variances and covariances were introduced in 2011 and 2013, as discussed in Section 1.3. The implementation of 4D-Var in the Object Oriented Prediction System (OOPS), the algorithmic layer of the ECMWF data assimilation system, is described in Chapter 2.

1.3 BRIEF HISTORY OF 3D- AND 4D-VAR IN ECMWF OPERATIONS

3D-Var was implemented in ECMWF operations on 30 January 1996, and was replaced on 25 November 1997 by 4D-Var. The two three-part series of QJRMS papers mentioned above basically describe the two schemes as they were at the time of their implementation. However, there have been very significant developments of the variational data assimilation system during its time in operations. Major code-upgrades have taken place in connection with migrations to new computers. 3D-Var was first implemented on a CRAY C90 shared-memory parallel computer system and was migrated to a distributed-memory Fujitsu VPP700 machine in September 1996 (Dent *et al.*, 1997). Further migrations of the codes took place in May 2000 to a Fujitsu VPP5000, and in March 2003 to an IBM P690, and later in 2004 to an IBM P690+. A history of the evolution of ECMWF super-computing is available at http://www.ecmwf.int/services/computing/overview/supercomputer_history.html. The current IBM implementation of the assimilation system utilizes both shared-memory (OpenMP) and distributed-memory (MPI) parallelism. A general list of changes to the operational forecasting system is available at http://www.ecmwf.int/products/data/technical/model_id/index.html.

The observation handling and data screening modules have also developed rapidly to keep pace with the changing computer environment, and the arrival of new observational data sets. The codes inherited from OI were replaced with new codes at the time of the migration to Fujitsu in 1996. The need for improved parallelism and rapidly increasing data volumes lead to the development of the Observation DataBase (ODB) software (see ODB documentation), facilitating in March 2003 the move to the IBM computers and the introduction of high-spectral resolution satellite data (first AIRS and then IASI in 2009) as well as the use of radio occultation data (GPSRO) in 2008. The quality control, data selection and screening algorithms are described in the paper by Järvinen and Undén (1997), and variational quality control of observations in Andersson and Järvinen (1999). Since 2009 the variational quality control of conventional observations uses the Huber norm (Tavolato and Isaksen, 2010).

One of the most important aspects of a variational assimilation is the specification of background errors. The original formulation of Courtier *et al.* (1998) was replaced in May 1997 by that of Derber and Bouttier (1999). The latter formulation is still used as described in Chapter 4, including improvements such as non-linear balance (January 2003, Cy25r3, Fisher (2003)), wavelet-Jb (April 2005, Cy29r1, Fisher (2006)), introduction of ozone as an analysis variable (Dethof and Holm, 2004) in October 1999 (Cy21r4), the new humidity analysis (Hólm *et al.*, 2002) in October 2003 (Cy26r3), and Jb for GEMS variables (Benedetti and Fisher, 2006). The cycling algorithms for analysis and background errors (Fisher and Courtier, 1995, and Chapter 6) were introduced in 1996. The calibration of background error statistics is since October 1999 based on an ensemble of 3D-Var data assimilations, updated in January 2003 to statistics based on a 4D-Var ensemble (Fisher, 2003). Since May 2011 the EDA (Isaksen *et al.*, 2010) provides flow-dependent background error standard deviations for vorticity on each cycle (Bonavita *et al.*, 2012). Since November 2013 (Cy40r1) the EDA is also used to compute online estimates of the background error correlation model (wavelet-Jb), as described in Chapter 6.

6-hourly 4D-Var was introduced operationally in 1997, at resolution T213L31, with two iterations of the outer loop: the first at T63L31 with 50 iterations (simplified physics) and the second with 20 iterations (with tangent-linear physics, at same resolution). In April 1998 the resolution was at T_L319. In June 1998 the radiosonde/pilot usage was revised (i.e. use of significant levels and temperature instead of geopotential) and we started using time-sequences of data (Järvinen *et al.*, 1999), so-called 4D-screening. The data assimilation scheme was extended higher into the atmosphere on 10 March 1999, when the $T_L319L50$ model was introduced, which in turn enabled the introduction in May 1999 of ATOVS radiance data (McNally *et al.*, 1999). In October 1999 the vertical resolution of the boundary layer was enhanced taking the number of model levels to a total of L60. In summer 2000 the 4D-Var period was extended from 6 to 12 hours (Bouttier, 2001), whereas the ERA-40 configuration (Uppala *et al.*, 2005) was built as an FGAT (first guess at the appropriate time) of 3D-Var with a period of 6 hours. In November 2000, the horizontal resolution of 4D-Var was increased to $T_L511L60$, with inner loop resolution enhanced from T63L60 to $T_L159L60$ using the linearized semi-Lagrangian scheme. We moved to three inner loops in June 2007.

In January 2003 the 4D-Var solution algorithm was comprehensively revised (in Cy25r4, Andersson *et al.*, 2003) to take advantage of the efficiency of conjugate gradient minimisation

(with pre-conditioning, as introduced one year earlier) and a multi-resolution incremental technique (Veerse and Thépaut, 1998; Trémolet, 2005). This is the 4D-Var solution algorithm that is presented here in Chapter 2. The early delivery suite was introduced in June 2004 (in Cy28r2, Haseler, 2004).

In February 2006, the outer loop resolution of 4D-Var was increased to $T_L799L91$, with inner loop resolution enhanced to $T_L255L91$. During 2007 and 2008 the moist linear physics was improved in three IFS cycles; the details are explained in Chapter 3. Then in January 2010 the outer loop resolution was increased to $T_L1279L91$, with unchanged inner loop resolution. The ERA-Interim reanalysis also used 12 hour 4D-Var at $T_L255L60$ outer loop resolution, with $T_L159L60$ inner loop resolution (Dee *et al.*, 2011). The outer loop resolution is the same as that of the forecast model.

Weak constraint 4D-Var (Trémolet, 2003) was introduced in September 2009, with EDA variances in May 2011 and EDA-based covariances in November 2013.

1.4 INCREMENTAL FORMULATION OF VARIATIONAL DATA ASSIMILATION

In 3D/4D-Var an objective function J is minimized. The cost function consists of three terms:

$$J = J_b + J_o + J_q + J_c \tag{1.1}$$

measuring, respectively, the discrepancy with the background (a short-range forecast started from the previous analysis), J_b, with the observations, J_o, with the model error, J_q, and with the slow manifold of the atmosphere, J_c. The J_q term is under active development and is not described further here, but see Laloyaux *et al.* (2020). The J_c-term controls the amplitude of fast waves in the analysis and is described in Chapter 2. It is omitted from the subsequent derivations in this section.

In its incremental formulation (Courtier *et al.*, 1994), we write

$$J(\delta\mathbf{x}) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}(\mathbf{H}\delta\mathbf{x} - \mathbf{d})^T\mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x} - \mathbf{d}) \tag{1.2}$$

$\delta\mathbf{x}$ is the increment and at the minimum the resulting analysis increment $\delta\mathbf{x}^a$ is added to the background $\mathbf{x}^b$ in order to provide the analysis $\mathbf{x}^a$ given by

$$\mathbf{x}^a = \mathbf{x}^b + \delta\mathbf{x}^a \tag{1.3}$$

$\mathbf{B}$ is the covariance matrix of background error while $\mathbf{d}$ is the innovation vector

$$\mathbf{d} = \mathbf{y}^o - H\mathbf{x}^b \tag{1.4}$$

where $\mathbf{y}^o$ is the observation vector. $\mathbf{H}$ is a suitable low-resolution linear approximation of the observation operator H in the vicinity of $\mathbf{x}^b$, and $\mathbf{R}$ is the covariance matrix of observation errors. The incremental formulation of 3D/4D-Var consists therefore of solving for $\delta\mathbf{x}$ the inverse problem defined by the (direct) observation operator $\mathbf{H}$, given the innovation vector $\mathbf{d}$ and the background constraint. The gradient of J is obtained by differentiating (1.2) with respect to $\delta\mathbf{x}$ to give

$$\nabla J = (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\delta\mathbf{x} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d} \tag{1.5}$$

At the minimum, the gradient of the objective function vanishes, thus from (1.5) we obtain the classical result that minimizing the objective function defined by (1.2) is a way of computing the equivalent matrix-vector products given by

$$\delta\mathbf{x}^a = (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{d} \tag{1.6}$$

where $\mathbf{B}$ and $\mathbf{R}$ are positive definite, see e.g. Lorenc (1986) for this standard result. $\mathbf{H}\mathbf{B}\mathbf{H}^T$ may be interpreted as the square matrix of the covariances of background errors in observation space while $\mathbf{B}\mathbf{H}^T$ is the rectangular matrix of the covariances between the background errors in model space and

the background errors in observation space. 3D-Var uses the observation operator $\mathbf{H}$ explicitly and statistical models ($\mathbf{B}$ and $\mathbf{R}$) are required only for describing the statistics of the background errors in model space and the observation error in observation space. Consequently, in 3D/4D-Var it is easy, from an algorithmic point of view, to make use of observations such as TOVS radiances, which have quite a complex dependence on the basic analysis variables. The background term and background error covariance modelling (i.e. $\mathbf{B}$) are described in Chapter 4 and observation operators $\mathbf{H}$ are described in Chapter 5.

Chapter 2

4D variational assimilation

Table of contents

2.1   INTRODUCTION

This part of the documentation provides a detailed description of the 4D-Var algorithm. It also provides an overview of the Object Oriented Prediction System (OOPS), the top most algorithmic software layer designed to decouple the implementation of variational algorithms from model specific details. OOPS drives the assimilation algorithm consisting of nested iterations, called inner and outer loops. The data flow is documented in Chapter 11. The interpolation of model fields to observation points (OBSHOR) and the organisation of the data in memory (yomsp, yommvo) is described in Chapter 5. The structure of the computation of the observation cost function and its gradient, managed by the routines OBSV and TASKOB can also be found in Chapter 5. The background term and the 'change-of-variable' operators are explained in Chapter 4.

2.2   SOLUTION ALGORITHM

2.2.1   The incremental method

The adopted solution algorithm is incremental (Courtier *et al.*, 1994), which allows for considerable flexibility with respect to the computer expense of the various steps of the 4D-Var minimisation. In the incremental approach the highest possible resolution is used for the computation of the model trajectory, and for calculating the departures between observations and model, whereas a lower-resolution model (its adjoint and tangent linear) are used for the iterative and relatively costly computation of analysis increments (Trémolet, 2004; Radnóti *et al.*, 2005). The lower-resolution iterations (the inner-loops) can optionally be nested within a set of outer-loop iterations at full resolution (Trémolet, 2005). Apart from the resolution, the cost of the inner-loops will depend also upon the complexity of the inner-loop model, e.g. the use of simpler or more complete representations of the physical processes (Janisková *et al.*, 2002; Tompkins and Janisková, 2004).

In a further 'multi-resolution' extension to the incremental method (Veerse and Thépaut, 1998) the inner-loop resolution is increased with each iteration of the outer-loop. A schematic is shown in Fig. 2.1. In particular, the information about the shape of the cost-function obtained during the early low-resolution iterations provides a very effective pre-conditioner (Chapter 6) for subsequent iterations at higher-resolution, thus reducing the number of costly iterations. The inner-loops can be particularly efficiently minimised using the conjugate gradient method, provided the cost-function is purely quadratic (Fisher,
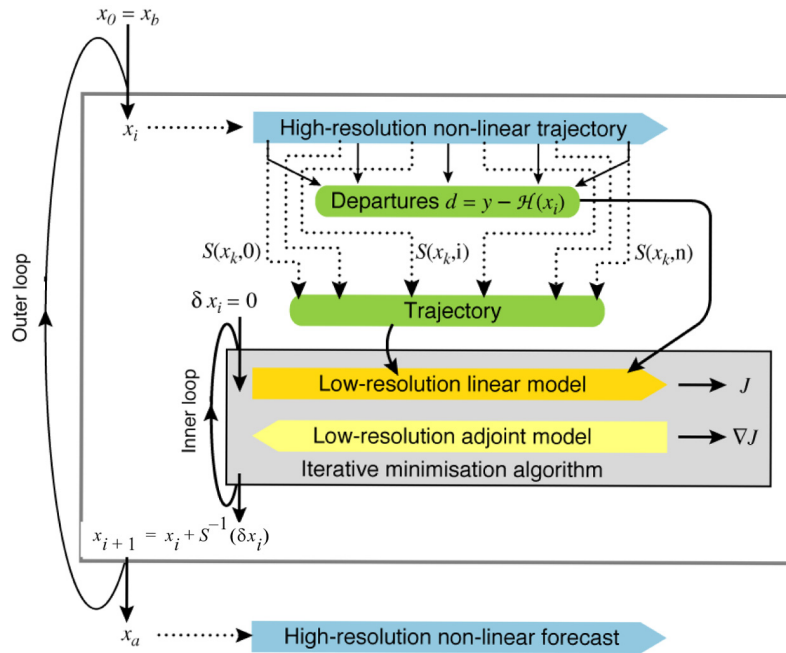
Figure 2.1 *Schematic of the revised 4D-Var solution algorithm implemented in January 2003 (Cy25r4). Outer loops are performed at high resolution (currently Tco1279) using the full non-linear model. Inner iterations are performed at lower resolution (first T255, then T319, T399 and T511) using the tangent-linear forecast model, linearized around a 12-hour succession of model states ('the trajectory') obtained through interpolation from high resolution (S denotes the truncation operator, J the cost function and x the atmospheric state vector).*

1998), i.e. the operators involved in the definition of the cost function (the model and the observation operators) are purely linear. For this reason, the inner-loops have been made entirely linear, with the non-linear effects gathered at the outer-loop level.

2.2.2   The data assimilation steps

In the CY48R1 the operational data assimilation cycle is 12-hours long, running from 09–21 UTC to produce the 12 UTC analysis and forecast products, and from 21–09UTC for the 00 UTC production (Haseler, 2004). Four outer loops are performed, where each outer loop computation is carried out by a dedicated task. An outer loop consists of a non-linear high resolution trajectory followed by minimisation of the quadratic cost function. The last outer loop task includes additionally the Late 4D-start mechanism, detailed further below, for computing analyses (type=an) at the main synoptic hours (00, 06, 12, 18). The following steps are performed in outer loop computations within a single executable.

(i)   Comparison of the observations with the background at high resolution to compute the departures. These are stored in the NCMIFC1-word of the ODB (the observation database) The first outer loop additionally performs **screening** (i.e. blacklisting, thinning and quality control against the background) of observations (see Part 1). The screening determines which observations will be used to evaluate the quadratic cost function gradient at the first guess (refer to Chapter 2.4.1 for its precise definition). Very large volumes of data are present during the screening run only, for the purpose of data monitoring. In all but the first outer loop, the variational quality control is computed, and the resulting QC weights are used in subsequent minimisations.

(ii)   The linearization trajectory of the linear model and observation operator is computed using a low resolution version of the non-linear model. This model is forced using the high resolution non-linear trajectory fields, recorded in memory during the non-linear model evaluation, at each linear model time step.

(iii) In order to set up the minimisation, the first guess gradient is computed (see Chapter 2.4.1). Optionally, adjoint tests for the linear model and observation operators are evaluated prior to the minimisation. The minimisation is performed using the Lanczos algorithm and followed by computation, in case of the first minimisation or update otherwise, of the spectral limited memory preconditioner (Fisher, 1998). The preconditioner is formed using Ritz vectors diagnosed by the Lanczos algorithm during the minimisation (see Chapter 6.5 for further details).

(iv) The control vector is transformed to the model space and the increment is interpolated to the outer loop resolution and added to the first guess.

(v) In case of the last outer loop, following the minimisation, the Late 4D-start mechanism is used to compute analysis fields (type=an) at t+3h and t+9h from the start of the assimilation window. The analysis fields are computed by adding the increment (propagated by the linear model) to the first guess fields valid at the required synoptic times. The first guess fields are recorded in memory during the non-linear trajectory prior to the minimisation. The analysis fields are produced subject to sanity tests performed on increments of temperature and horizontal wind components. In case the tests detect abnormally large increments, the analysis fields are produced by a dedicated task called restart_999. Triggering restart_999 puts the evaluation of the final non-linear trajectory in the critical path. The restart_999 task converts the fields (type=4v) output by the non-linear trajectory task into analysis fields (type=an) required by the downstream short and long forecast tasks. See also the schematic in Chapter 11.

Each outer loop consists of a non-linear trajectory and a minimisation and is carried out by a dedicated oopsvar task with NCONF = 131. In case of the first outer loop the non-linear trajectory is followed by observation screening. Observations can be optionally re-screened in the remaining outer loops depending on the configuration. The last outer loop includes calling the Late 4D-start mechanism for producing analysis fields (type=an) at the required synoptic times. The final non-linear trajectory run that carries out verification screening - that is a comparison between observations and final analysis trajectory - is performed by an ifstraj task with NUPTRA = 999, NCONF = 1.

2.2.3 Interpolation of trajectory and increments

A truncation operator (shown as S in the schematic, Fig. 2.1 above) is required to take the trajectory fields from the outer loop to the inner loop resolution. This is done using conserving (area averaging) grid-point interpolations for the upper-air and surface fields, whereas temperature, logarithm of the surface pressure, horizontal components of the wind are truncated spectrally. Initial time surface fields are interpolated in a dedicated task called lowres using the so-called 'full-pos' configuration of IFS. Trajectory cloud and surface fields, and trajectory of physical tendencies are generated in a (low resolution) non-linear model integration in which the model state is replaced by the interpolated trajectory at each time instance for which it is available. The computation of the trajectory is handled in OOPS by the TrajectorySaver class. The TrajectorySaver is a post-processor that is attached to the non-linear model trajectory computation where it records required high resolution fields. These recorded fields are subsequently used in the TrajectorySaver::recordTrajectory() method, called after the non-linear trajectory computation (which includes screening of observations if enabled), to force to low resolution linearization trajectory model. The interpolation of model fields is handled by the State::forceWith() method. This method of the State takes a Variables object which can be used to control fields that need to be interpolated. The State::forceWith() method is implemented by the FIELDS_FORCE_WITH procedure of the FIELDS type in the IFS. It includes regularisation of the multi-level snow model fields. The linearization trajectory memory allocation handling is managed by the module TRAJECTORY_MOD_OOPS. The non-linear model is represented by the Model class that is implemented by MODEL derived type in the IFS. The time-stepping of the model is controlled by the Model::forecast() method. A single model time step is performed by the Model::step() method which is implemented by MODEL_STEP procedure of the MODEL derived type, while the linearization trajectory generation step is performed by its MODEL_STEP_TRAJ procedure.

Increments and states in OOPS are represented by the Increment and State interface class, respectively. Increment and State interface classes are implemented by FIELDS_BASE and FIELDS derived types in the IFS. FIELDS type inherits from the FIELDS_BASE type to avoid duplication of code. Both the first

guess/background states and the analysis increment fields produced by the minimisation are defined in grid point space. Increment fields are added to the high resolution first guess fields via an overloaded assignment operator of the State class. This operator is implemented by the FIELDS_ADD_INCR procedure of the FIELDS_BASE derived type. Spectrally fitted variables, temperature, logarithm of surface pressure, horizontal components of wind, are transformed to spectral space and padded with zeros for the wave numbers not represented in the inner-loops before being added to the high resolution fields in spectral space. Humidity and ozone increments are interpolated in grid point space using bi-linear interpolation. The FIELDS_ADD_INCR procedure ensures that the resulting fields are positive. First guess humidity field is regularised after adding the increment in FIELDS_LIMIT_Q procedure of the FIELDS derived type to ensure that it takes on physical values.

2.2.4 Minimisation, preconditioning and control variable

In practice, it is necessary to pre-condition the minimisation problem in order to obtain a quick convergence. As the Hessian (the second derivative) of the objective function is not accessible, Lorenc (1988) suggested the use of the Hessian of the background term J_b. The Hessian of J_b is the matrix $\mathbf{B}$. Such a preconditioning may be implemented either by a change of metric (i.e. a modified inner product) in the space of the control variable, or by a change of control variable. As the minimisation algorithms have generally to evaluate several inner products, it was found more efficient to implement a change of variable. Algebraically, this requires the introduction of a variable χ such that

$$J_b = \chi^T \chi \tag{2.1}$$

Comparing (1.2) and (2.1) shows that $\chi = \mathbf{B}^{-1/2}\delta\mathbf{x}$ satisfies the requirement. χ thus becomes the *control variable* of the preconditioned problem (see Section 4.2). A single-observation analysis with such preconditioning converges in one iteration. The transformation from the model space to control space, $\delta\mathbf{x} = \mathbf{B}^{1/2}\chi$, and its adjoint, are required in practice. The increment $\delta\mathbf{x}$ is implemented by the ControlIncrement class. This class has a composite structure, as illustrated in Fig. 2.2, where each component of the increment: the model space, the auxiliary observation space and the auxiliary model space components are objects of Increment4D, ObsAuxIncrements ModelAuxIncrement classes, respectively. Increment4D contains a vector of objects of the Increment class, which in case of 4D-Var has only a single element. The Increment class is implemented by the FIELDS_BASE derived type in the IFS. The ModelAuxIncrement class represents the increment of the model error (see Chapter 4.4) and is implemented by the ERRMOD_FIELDS derived type in the IFS. ObsAuxIncrements is a composite class which contains a map of objects of the ObsAuxIncrement class providing flexibility for including an arbitrary number of auxiliary observation space increment components. The components corresponding to VarBC and TOVSCV are implemented in the IFS by VARBC_INCREMENT and TOVSCV derived types, respectively. The control vector χ has a similar composite structure and is implemented by the ControlVector class. It contains IncrCtlVec4D, ModelAuxCtlVec and ObsAuxCtlVecs components corresponding to those found in the ControlIncrement class. The IncrCtlVec4D class contains a vector of objects of the IncrCtlVec class that themselves wrap the components corresponding to the modelled (IncrModCtlVec) and ensemble parts (IncrEnsCtlVec) of the B matrix. The IncrModCtlVec class is implemented by the JB_CONTROL_VECTOR derived type. The B matrix is implemented by the CostJbTotal class and contains components corresponding to the structure of the ControlIncrement and the ControlVector. The model space component is represented by JbState base class which is inherited by CostJb3D in case of 4D-Var. The CostJb3D class wraps around a base class ModelSpaceCovarianceBase which is inherited by the interface class ErrorCovariance. The ErrorCovariance class is implemented by the ERROR_COVARIANCE_3D derived type in the IFS. The auxiliary model space component in CostJbTotal is represented by the interface class ModelAuxCovariance that is implemented by the ERRMOD_COVARIANCE_3D derived type in the IFS. Finally the auxiliary observation space components are implemented via a composite class ObsAuxCovariances that holds a map of ObsAuxCovariance interface class objects. The VarBC and TOVSCV covariance is implemented by VARBC_COV and TOVSCV_COV dervied type in the IFS, respectively. The CostJbTotal class exposes CostJbTotal::multiplyBsqrt() and CostJbTotal::multiplyBsqrtTrans() methods that implement the multiplication by the square root of the B matrix and its transpose, respectively.
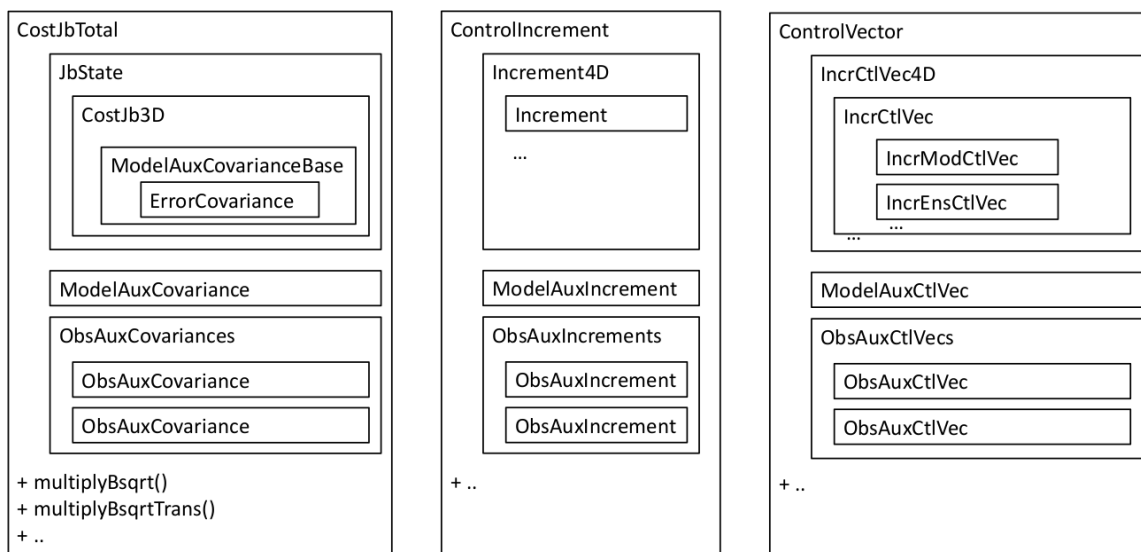
Figure 2.2 *Structure of the CostJbTotal, ControlIncrement and ControlVector classes implementing* **B***, δ***x** *and χ in equation 2.1.*

The J_b-based pre-conditioning is sufficient when the B-term dominates over the observation term of the Hessian. With increasing amounts of observational information, and in cases with locally dense observation coverage, the observation term can be dominant over the background term in defining the shape of the cost-function (its second derivative). The combined Lanczos/conjugate gradient method allows computation of the leading eigenvectors and eigenvalues of the Hessian while solving for the analysis, essentially at no extra cost. This method is fully described in Chapter 6, and it is used for all inner-loop iterations. The Hessian eigenvector information obtained at low resolution is used as pre-conditioner at subsequent inner-loop iterations at higher resolution. This has been shown to be a very effective way of reducing the number of iterations required at higher inner-loop resolutions.

2.3 TOP-LEVEL CONTROLS

Variational application implements the incremental 4D-Var algorithm in OOPS. The flow diagram of Variational is shown in Fig. 2.3. The Variational::execute() method starts with the set up of an outer loop resolution Geometry and nonlinear Model objects, followed by the setup of the CostFunction. The latter involves a call to CostFunction::setupTerms() method that constructs the terms of the cost function and reads in the background state. Next an IncrementalAssimilation object is created and its method IncrementalAssimilation::assimilate() is called. This is where the main body of the incremental 4D-Var algorithm is implemented.

The first step in the IncrementalAssimilation::assimilate() method is the set up of the first guess ControlVariable that is subsequently initialized with the background (set up earlier and read in in the CostFunction::setupTerms() method). It is followed by the construction of the Minimizer that is responsible for the minimisation of the quadratic cost function and Restarter which allows to start the execution from an arbitrary outer loop. The Restarter checks if the execution directory contains a file called Logbook.json. This file contains restart information required to determine which components of the system need restarting. If Logbook.json is not found, the calls to the Restarter::restart() method have no effect. If the restart mechanism is active, the first guess ControlVariable and IncrementalAssimilation are restarted. In case the execution is not restarted, the first guess ControlVariable can be optionally warm started if it is different from the background and also optionally re-centred via a call to the IncrementalAssimilation::recentreFirstGuess() method.

The loop over the outer loops, where the number of outer loops performed within a single executable is configurable, starts with an optional set up of a StateSaver post-processor. It is tasked with recording first guess states during the non-linear trajectory that are required by the Late 4D-start mechanism to

produce analysis fields (an-type). The CostFunction is next optionally restarted, again depending on the availability and content of the Logbook.json file.

The non-linear cost function is evaluated and the quadratic cost function is set up in the CostFunction::linearize() method. This method sets up first the inner loop resolution Geometry. Subsequently, a TrajectorySaver object is set up. It is attached to the non-linear model trajectory to record the interpolated high resolution states at each linear model time step. The J_o and J_c terms of the cost function, implemented by CostJo and CostJcDFI classes, respectively, call their initialize() methods that return post-processors (classes derived from the PostBase class) that are enrolled into a PostProcessor object attached to the forecast. The role of these post-processors is to collect required information during the time stepping of the trajectory model performed by CostFunction::runNL(). This method resolves to calling CostFct4DVar::runNL() that in turn calls the Model::forecast() method. In order to illustrate the idea of post-processors, consider CostJo::initialize() that returns a post-processor called Observer. Its task is to compute model equivalents at observation locations. The Model::forecast() method calls the PostProcessor::initialize(), PostProcessor::process() and PostProcessor::finalize() methods before, during and after the time stepping loop, respectively. As a PostProcessor class contains a collection of post-processors (recall that these are classes derived from the PostBase class), these PostProcessor methods call corresponding methods of individual PostBase derived post-processors. In case of the Observer, the call to PostProcessor::process() method entails performing interpolation of model fields to observation locations at each time step. Subsequent call to the PostProcessor::finalize() method results in calling the observation operator to compute model equivalents in observation space. Once the trajectory evaluation is completed the J_o and J_c terms of the cost function call their finalize() methods, where further processing of the gathered information takes place. In case of the CostJo::finalize() method, this step includes computation of the departures implemented by Departures class, observation screening (a method of the ObsSpaces class) and the set up of the observation error covariance matrix implemented by ObsErrorCovariance. Next, J_b, implemented by CostJbTotal, J_o and J_c terms call their initializeTraj() methods that return post-processors that are enrolled into the TrajectorySaver. These post-processors are required to setup the terms of the quadratic cost function. The computation of the linearization trajectory for the model and observation operator is performed by the TrajectorySaver::recordTrajectory() method. Subsequently J_b, J_o and J_c terms of the cost function call their finalizeTraj() method to finish setting up the terms of the quadratic cost function. Such design is required as the computation of the linearization trajectory of the observation operator cannot be performed during the non-linear model evaluation prior to screening.

Following the call to CostFunction::linearize(), the Minimizer::warmStart() method is called, if required, only in case when the execution is not restarted. In case when the execution is restarted, the Restarter::restart() method is called to initialize the Minimizer (read in the initial and preconditioning control vectors, and auxiliary minimisation diagnostics). The Minimizer::minimize() method starts with optional calls to Minimizer::tlmTests() and Minimizer::adjTests() methods that carry out diagnostic tests of the tangent linear model and linear observation operator and their adjoints. The main minimisation takes place in the Minimizer::doMinimize() method which resolves to the actual minimiser (configurable) being used, that is SQRTMinimizer. First, various minimisation set ups take place in SQRTMinimizer::setupMinimization() followed by the computation of the first guess gradient in CostFunction::computeGradientFG(). The system of linear equations is solved using the Lanczos algorithm in SQRTPLacznos::solve(). The obtained solution in control vector space is subsequently transformed to the model space to produce an increment. Afterwards, the SQRTMinimizer::enforceObsAuxFormulation() is called to enforce an incremental or non-incremental formulation for the auxiliary observation space components of the ControlIncrement. The Minimizer::doMinimize() method finishes with book-keeping of the quadratic cost function diagnostics in SQRTMinimizer::updateJbCostFunction().

Back in the IncrementalAssimilation::assimilate() method, the increment is added to the first guess in CostFunction::addIncrement(). The outer loop finishes by calling the Restarter::checkpoint() method to output restart files required by the subsequent outer loop. Finally, going beyond the loop over outer loops, the IncrementalAssimilation::late4Dstart() method is called. The Late 4D-start mechanism propagates the increment valid at the beginning of the window to required synoptic times. Subject to sanity tests on the magnitude of the temperature and wind increments, analysis fields are obtained by
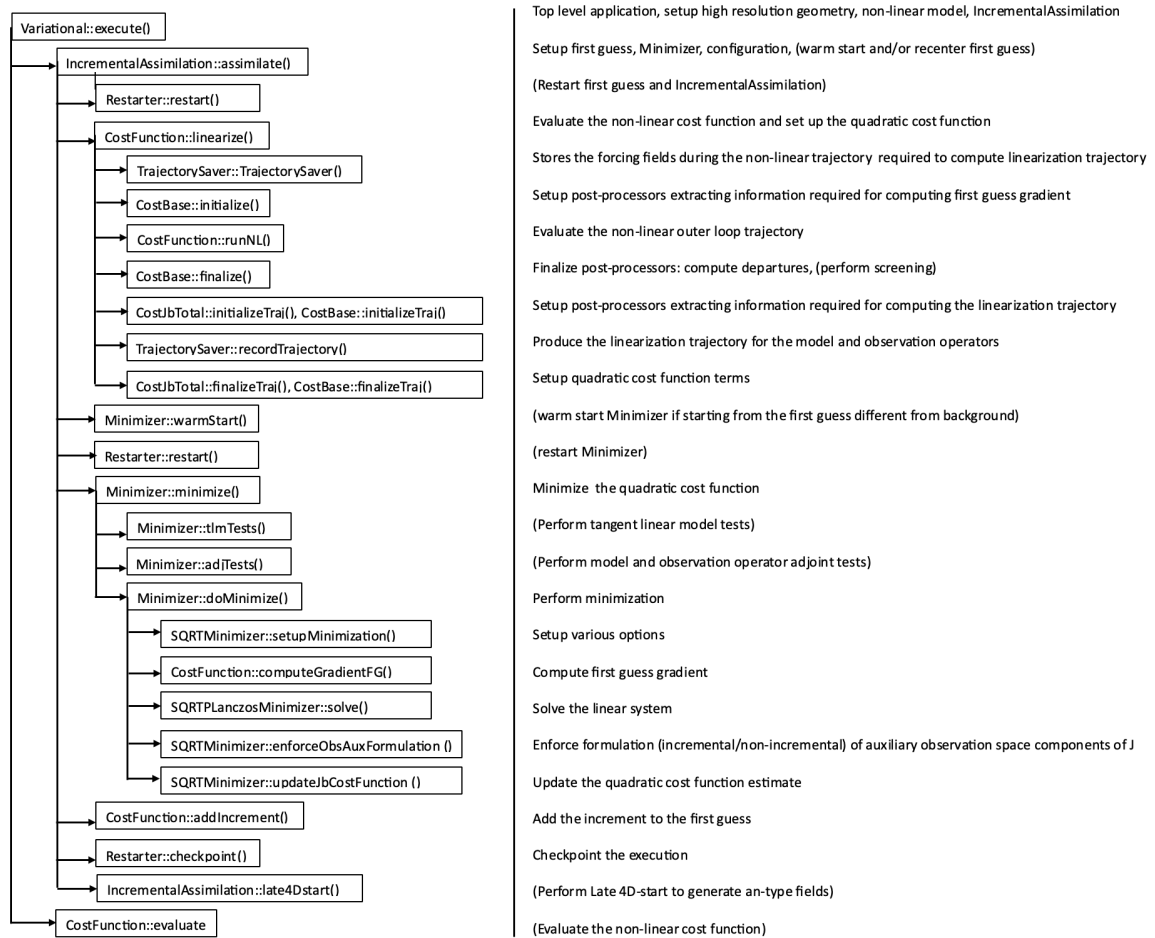
Figure 2.3 *Flow diagram for OOPS Variational application driving 4D-Var.*

adding the propagated increments to corresponding first guess fields stored away during the non-linear trajectory by the StateSaver mentioned earlier. In case the sanity test do not pass, only the analysis at the start of the window is written out (type 4V) and the remaining required analysis fields are generated by a dedicated task restart_999. Variational::execute() finishes with an optional diagnostic evaluation of the non-linear cost function, which is typically skipped. The final non-linear cost function is evaluated in a dedicated ifstraj task instead.

2.4 MINIMISATION

2.4.1 The cost function and its gradient

The non-linear cost function is evaluated and linearized in the CostFunction::linearize() method. The minimisation of the quadratic cost function J amounts to finding a solution for which its gradient ∇J vanishes. The quadratic cost function is preconditioned via a change of variable

$$dx = U\chi + x_b - x_{fg} \tag{2.2}$$

where U is the square root of the B matrix, χ is a control vector, x_b and x_{fg} are background and first guess, respectively. The first guess gradient is computed prior to the minimisation by calling CostFunction::computeGradientFG() from SQRTMinimizer::doMinimize(). The first guess gradient takes the following general form:

$$U^T H^T R^{-1}[d + H(dx_{fg} - U\chi_0)] - J_b\chi_0 \tag{2.3}$$

where the term $H^T R^{-1}[d + H(dx_{fg} - U\chi_0)]$ is computed by CostFunction::computeGradientFG(), χ_0 is the initial control vector, dx_{fg} is the first guess increment, d denotes the observation departures and J_b is an identity matrix except for the VarBC component of the control vector. The Hessian of the quadratic cost function takes the form:

$$J_b + U^T H^T R^{-1} U \tag{2.4}$$

where the operator $U^T H^T R^{-1} H U \chi$ is implemented by UtHtRinvHUMatrix. The linear system of equations is solved iteratively using the Lanczos algorithm implemented by the SQRTPLanczos class.

2.5 THE ASSIMILATION WINDOW

In 4D-Var, observations are organised in time-slots (currently half-hourly) as described in the ODB documentation. The cost-function measures the distance between a model trajectory and the available information (background, observations) over an assimilation interval or window. For a 12-hour window, it is either (09 UTC–21 UTC) or (21 UTC–09 UTC), see Haseler (2004). Equation (1.2) (see Chapter 1) is replaced by

$$J(\delta\mathbf{x}) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}\sum_{i=0}^{n}(\mathbf{H}_i\delta\mathbf{x}(t_i) - \mathbf{d}_i)^T\mathbf{R}_i^{-1}(\mathbf{H}_i\delta\mathbf{x}(t_i) - \mathbf{d}_i) \tag{2.5}$$

with subscript i the time index. Each i corresponds to a half-hour time slot. $\delta\mathbf{x}$ is as before the increment at low resolution at initial time, and $\delta\mathbf{x}(t_i)$ the increment evolved according to the tangent linear model from the initial time to time index i. $\mathbf{R}_i$ and $\mathbf{B}$ are the covariance matrices of observation errors at time index i and of background errors respectively. $\mathbf{H}_i$ is a suitable linear approximation at time index i of the observation operator H_i. The innovation vector is given at each time step by $\mathbf{d}_i = \mathbf{y}_i^o - H_i\mathbf{x}^b(t_i)$, where $\mathbf{x}^b(t_i)$ is the background propagated in time using the full non-linear model and $\mathbf{y}_i^o$ is the observation vector at time index i.

Chapter 3

Tangent-linear physics

Table of contents

3.1   INTRODUCTION

The linearized physics describes six processes: vertical diffusion, subgrid-scale orographic effects, non-rographyc gravity wave drag, radiation, large-scale condensation/precipitation and convection. In order to prevent spurious unstable perturbations from growing, a number of simplifications have been defined for these schemes with respect to the non-linear physical parametrization schemes (described in Part IV: Physical processes) used in the forecast model. All simplified parametrizations are called in each minimization of 4D-Var.

This text is focused on brief scientific description of the linearized physical parametrization schemes together with some technical aspects. More detailed scientific issues can be found in the mentioned literature references.

3.2 SET-UP

The full linearized physics package currently in operational use is activated by setting the following switches in namelist NAMTRAJP:

- LEVDIF2 (vertical diffusion),

- LEGWDG2 (subgrid-scale orographic gravity wave drag),

- LEGWWMS2 (non-orographic gravity wave drag),

- LERADI2, LERADS, LERADN2 and LERADSW2 (radiation),

- LENCLD2 (large-scale condensation/precipitation),

- LECUMF2 and LECUMFS2 (in namelist NAMCUMFS) (moist convection),

In addition, the following switches must be set to .TRUE.: LEPHYS and LAGPHY (also necessary to activate the ECMWF non-linear physics) in namelist NAEPHY and LETRAJP (to activate storage of the trajectory at $t - \Delta t$) in namelist NAMTRAJP.

Tunable parameters of the improved physics (which should not in principle be modified) are defined in SUPHLI. The logical LPHYLIN (namelist NAMTLEVOL)is used to activate the simplifications and/or modifications associated with the linear package in the non-linear physics. This variable is set to .FALSE. by default, but is forced to .TRUE. before starting tangent-linear (TL) and adjoint (AD) computations in CNT3TL and CNT3AD. Thus this switch remains .TRUE. for the linearized physics called in EC_PHYS_TL and EC_PHYS_AD.

3.3 MIXED-PHASE THERMODYNAMICS

The thermodynamical properties of the water mixed phase are represented by a differentiable weighting function between $T_0 = 0°C$ and $T_{ice} = -23°C$ given by

$$\alpha(T) = \frac{1}{2}[1 + \tanh\{\mu(T - T_{crit})\}] \tag{3.1}$$

with $\mu = 0.15$ (RLPALP1) and $T_{crit} = T_{ice} + \frac{T_0 - T_{ice}}{\sqrt{2}}$ (RLPTRC).

The tuning parameter μ controls the intensity of the smoothing, and the temperature T_{crit} has been chosen to give $\alpha = 0.5$ for the same temperature as in the operational quadratic formulation (see function FCTTRE).

This weighting function is used by the large-scale condensation and moist-convection routines.

3.4 VERTICAL DIFFUSION

Vertical diffusion applies to wind components, dry static energy and specific humidity. The exchange coefficients in the planetary boundary layer and the drag coefficients in the surface layer are expressed as functions of the local Richardson number (Louis *et al.*, 1982). Analytical expressions are generalized to the situation of different roughness lengths for heat and momentum transfer.

For any conservative variable ψ (wind components u, v; dry static energy s; specific humidity q), the tendency of its perturbation ψ' produced by vertical diffusion is

$$\frac{\partial \psi'}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial z}\left(K(Ri)\frac{\partial \psi'}{\partial z}\right) \tag{3.2}$$

where ρ is the air density and Ri is the Richardson number.

In the planetary boundary layer, the exchange coefficient K is given by

$$K = l^2 \left\| \frac{\partial \mathbf{V}}{\partial z} \right\| f(Ri) \tag{3.3}$$

where $f(Ri)$ is a function of the Richardson number and l is the local mixing length (see below).

3.4.1 Upper-air

<u>Stable conditions</u>

In stable conditions ($Ri > 0$), the respective values of $f(Ri)$ for heat/moisture and momentum are computed using the Louis *et al.* (1982) formulation, which reads

$$f(Ri) = \begin{cases} \dfrac{1}{1 + 2bRi(1 + dRi)^{-0.5}} & \text{for momentum} \\ \dfrac{1}{1 + 2bRi(1 + dRi)^{0.5}} & \text{for heat/moisture} \end{cases} \tag{3.4}$$

with $b = 5$ and $d = 1$.

The vertical profile of the mixing length is computed using Blackadar (1962) as

$$l = \frac{\kappa(z + z_0)}{1 + \kappa \frac{(z + z_0)}{\lambda}} \tag{3.5}$$

where z_0 is the roughness length and $\kappa = 0.4$ is von Kármán's constant. The asymptotic mixing length, λ, is set to 120 m for momentum inside the PBL, 60 m otherwise.

<u>Unstable conditions</u>

In unstable situations ($Ri < 0$), the Monin-Obukhov theory is applied and the stability functions are derived from (Dyer, 1974; Hogström, 1988)

$$f(Ri) = \begin{cases} (1 - 16Ri)^{0.5} & \text{for momentum} \\ (1 - 16Ri)^{0.75} & \text{for heat/moisture} \end{cases} \tag{3.6}$$

The vertical profile of the mixing length in the unstable case also follows Blackadar (1962) (see 3.5), with λ set to 150 m for both momentum and heat/moisture.

3.4.2 Surface

To parametrize the turbulent fluxes at the surface, the drag coefficients (i.e. the exchange coefficients between the surface and the lowest model level) are computed. The neutral coefficients C_{MN} and C_{HN} are written as

$$C_{MN} = \frac{\kappa^2}{\left[\ln\left(\frac{z + z_{0M}}{z_{0M}} \right) \right]^2} \quad \text{and} \quad C_{HN} = \frac{\kappa^2}{\left[\ln\left(\frac{z + z_{0M}}{z_{0M}} \right) \ln\left(\frac{z + z_{0M}}{z_{0H}} \right) \right]} \tag{3.7}$$

The drag coefficients for momentum are computed as

$$C_M = \begin{cases} \dfrac{1}{1 + \frac{2bRi}{\sqrt{1 + dRi}}} C_{MN} & \text{if } Ri > 0 \\ \left(1 - \dfrac{2bRi}{1 + 3bc C_{MN} \sqrt{\frac{z + z_{0M}}{z_{0M}} |Ri|}} \right) C_{MN} & \text{if } Ri < 0 \end{cases} \tag{3.8}$$

The surface exchange coefficients for dry static energy and specific humidity are

$$
C_{\mathrm{H}} = \begin{cases} \dfrac{1}{1 + 3bRi\sqrt{1 + dRi}}C_{\mathrm{HN}} & \text{if } Ri > 0 \\[3ex] \left(1 - \dfrac{3bRi}{1 + 3bcC_{\mathrm{HN}}\sqrt{\frac{z+z_{0\mathrm{M}}}{z_{0\mathrm{M}}}|Ri|}}\right)C_{\mathrm{HN}} & \text{if } Ri < 0 \end{cases}
\tag{3.9}
$$

The empirical coefficients b (RLPBB), c (RLPCC) and d (RLPDD) are set to 5 in SUPHLI.

The same mixed layer parametrization as in the fully-fledged non-linear vertical diffusion scheme is also included (for details, see Part IV: Physical processes - 3.3 The exchange coefficients above the surface and mixed layer).

3.4.3 Regularization

Until cycle Cy33r1, perturbations of the exchange coefficients were neglected ($K' = 0$), in order to prevent spurious unstable perturbations from growing in the linearized version of the scheme (Mahfouf, 1999). From Cy33r1 onwards, regularization was introduced for exchange coefficients between upper model levels to include partial perturbations of these coefficients.

3.4.4 Flow chart

The simplified diffusion scheme VDFMAINS can be called from CALLPAR when the logical switch for linearized physics LPHYLIN is set to .TRUE..

The linear versions of the vertical diffusion scheme are called from the drivers VDFMAINSTL and VDFMAINSAD. The ensemble of routines that prepare the surface exchange coefficients and associated surface quantities needed for the solution of the vertical diffusion equations are part of the routine SURFEXCDRIVERSTL and SURFEXCDRIVERSTL. At this moment the surface arrays are not evolved in time, though the routines are coded for computing perturbations of: the roughness length (VDFUPDZ0TL, VDFUPDZ0AD), the surface boundary conditions for T and q (VDFSURFTL, VDFSURFAD), the surface exchange coefficients (VDFEXCSTL, VDFEXCSAD) and the equivalent evapotranspiration efficiency (VDFEVAPTL, VDFEVAPAD). This can be activated in the future by setting the logical switch LENOPERT set to .FALSE. in NAMTRAJP. The current default version is .TRUE. (i.e. no perturbation is required for the surface arrays).

Other computations performed in VDFMAINSTL and VDFMAINSAD involve the calculation of exchange coefficients above the surface layer (VDFEXCUSTL , VDFEXCUSAD), the solving of the diffusion equations for momentum (VDFDIFMSTL, VDFDIFMSAD) and dry static energy and moisture (VDFDIFHSTL, VDFDIFHSAD, as well as the incrementation of tendencies for wind, temperature and specific humidity (VDFINCRSTL, VDFINCRSTL.

The logical LEKPERT in NAMTRAJP controls the perturbations of the exchange and drag coefficients. When set to .FALSE. (default), the perturbations would be set to 0. In the current 4D-Var computation, LEKPERT is set to .TRUE., which means that the regularized exchanged coefficients (as described above) are used.

3.5 SUBGRID SCALE OROGRAPHIC EFFECTS

Only the low-level blocking part of the operational non-linear scheme developed by Lott and Miller (1997) (see documentation on ECMWF non-linear physics) is taken into account in the tangent-linear and adjoint calculations. The deflection of the low-level flow around orographic obstacles is supposed to occur below an altitude Z_{blk} such that

$$
\int_{Z_{\mathrm{blk}}}^{2\mu} \frac{N}{|\mathbf{U}|}\,\mathrm{d}z \geq H_{n_{\mathrm{crit}}}
\tag{3.10}
$$

where $H_{n_{\mathrm{crit}}}$ is a critical non-dimensional mountain height (GFRCRIT = 0.5), μ is the standard deviation of subgrid-scale orography, N is the vertical stability and $\mathbf{U}$ is the wind vector.

The deceleration of the wind due to the low-level blocking is given by

$$
\begin{aligned}
\left(\frac{\partial \mathbf{U}}{\partial t}\right)_{\text{blk}} &= -C_{\text{d}} \max\left(2 - \frac{1}{r}, 0\right) \frac{\sigma}{2\mu} \left(\frac{Z_{\text{blk}} - z}{z + \mu}\right)^{0.5} (B\cos^2\psi + C\sin^2\psi)\frac{\mathbf{U}|\mathbf{U}|}{2} \\
&= A(\mathbf{U}|\mathbf{U}|)
\end{aligned}
\tag{3.11}
$$

where

C_{d} is the low-level drag coefficient (GWAKE $= 1$)
σ is the mean slope of the subgrid-scale orography
γ is the anisotropy of the subgrid-scale orograph
ψ is the angle between the low-level wind and the principal axis of topography
$B = 1 - 0.18\gamma - 0.04\gamma^2$
$C = 0.48\gamma + 0.3\gamma^2$
$r = (\cos^2\psi + \gamma\sin^2\psi)/(\gamma\cos^2\psi + \sin^2\psi)$

Since the final wind tendency produced by the low-level drag parametrization is obtained from the following partially implicit discretization of (3.11)

$$
\left(\frac{\partial \mathbf{U}}{\partial t}\right)_{\text{blk}} = -\frac{\beta}{\beta + 1}\frac{\mathbf{U}^{n-1}}{2\Delta t}
\tag{3.12}
$$

where $\beta = A|\mathbf{U}^{n-1}|2\Delta t$, the corresponding perturbation is

$$
\left(\frac{\partial \mathbf{U}}{\partial t}\right)_{\text{blk}} = -\frac{\beta'}{(\beta + 1)^2}\frac{\mathbf{U}^{n-1}}{2\Delta t} - \frac{\beta}{\beta + 1}\frac{\mathbf{U}'^{n-1}}{2\Delta t}
\tag{3.13}
$$

Finally, the local dissipation heating is computed in the same way as in the non-linear scheme, together with the associated perturbation.

The main tangent-linear and adjoint routines, GWDRAGTL and GWDRAGAD, are called from CALLPARTL and CALLPARAD respectively. The depth of the low-level blocking layer is determined in GWSETUPTL and GWSETUPAD, while the low-level blocking effect described by (3.12) is computed at the end of GWDRAGTL and GWDRAGAD. As mentioned above, the representation of wave breaking is currently not activated in the linearized code by setting the constant RLPDRAG to zero in SUPHLI. Note that RLPDRAG is only used when logical LPHYLIN is .TRUE., otherwise GKDRAG is used (set to 0.3 in SUGWD).

3.6 NON-OROGRAPHIC GRAVITY WAVE DRAG

The non-linear scheme for non-orographic gravity wave (nonorog-gw) (Orr *et al.* (2010)) has been used operationally in the forecast model since September 2009 (cycle 35R3). Tangent-linear and adjoint versions were developed in order to reduce discrepancies between the full non-linear and linearized versions of the model. This parametrization scheme is based on Scinocca (2003). The latter is a spectral scheme derived from the Warner and McIntyre (1996) scheme which represents the three basic mechanisms of conservative propagation, critical level filtering, and non-linear dissipation. Since the full non-hydrostatic and rotational wave dynamics considered by Warner and McIntyre (1996) is too costly for operational models, only hydrostatic and non-rotational wave dynamics are employed.

The dispersion relation for a hydrostatic gravity wave in the absence of rotation is

$$
m^2 = \frac{k^2 N^2}{\tilde{\omega}^2} = \frac{N^2}{\tilde{c}^2}
\tag{3.14}
$$

where k, m are horizontal and vertical wavenumbers, while $\tilde{\omega} = \omega - kU$ and $\tilde{c} = c - U$ are the intrinsic frequency and phase speed (with c being the ground based phase speed and U the background wind speed in the direction of propagation), respectively.

The launch spectrum, which is globally uniform and constant, is given by the total wave energy per unit mass in each azimuth angle ϕ following Fritts and Nastrom (1993) as

$$\tilde{E}(m, \tilde{\omega}, \phi) = B\left(\frac{m}{m_*}\right)^s \frac{N^2 \tilde{\omega}^{-d}}{1 - \left(\frac{m}{m_*}\right)^{s+3}} \tag{3.15}$$

where B, s and d are constants, and $m_* = 2\pi L$ is a transitional wavelength. Instead of the total wave energy $\tilde{E}(m, \tilde{\omega}, \phi)$, the momentum flux spectral density $\rho \tilde{F}(m, \tilde{\omega}, \phi)$ is required. This is obtained through the group velocity rule. In order to have the momentum flux conserved in the absence of dissipative processes as the spectrum propagates vertically through height-varying background wind and buoyancy frequency, the coordinate framework (k, ω) is used instead of $(m, \tilde{\omega})$ as shown in Scinocca (2003).

The dissipative mechanisms applied to the wave field in each azimuthal direction and on each model level are critical level filtering and non-linear dissipation. After application of them, the resulting momentum flux profiles are used to derive the net eastward $\rho \bar{F}_E$ and northward $\rho \bar{F}_N$ fluxes. The tendencies for the (u, v) wind components are then given by the divergence of those fluxes obtained through summation of the total momentum flux (i.e. integrated over all phase speed bins) in each azimuth ϕ_i projected onto the east and north directions, respectively:

$$\frac{\partial u}{\partial t} = g \frac{\partial (\rho \bar{F}_E)}{\partial p} \tag{3.16}$$

$$\frac{\partial v}{\partial t} = g \frac{\partial (\rho \bar{F}_N)}{\partial p} \tag{3.17}$$

where g is the gravitational acceleration and p is the pressure.

In order to eliminate the artificial noise in TL computations caused by the introduction of this scheme, it was necessary to implement some regularizations. These included rewriting buoyancy frequency computations in the non-linear scheme in height coordinates instead of pressure coordinates and setting increments for momentum flux in the last three spectral elements (high phase speed) of the launch spectrum to zero.

The simplified non-linear version of the non-orographic gravity wave drag scheme can be found in routine GWDRAG_WMSS. TL and AD versions of the non-orographic gravity wave drag scheme (routines GWDRAG_WMSSTL and GWDRAG_WMSSAD) are activated by setting LEGWWMS2 to .TRUE. in namelist NAMTRAJP. The schemes would then be used with the prescribed time frequency GTPHYGWWMS. The necessary regularizations are activated by setting logical switch LREGWWMS in namelist NAMTRAJP.

3.7 RADIATION

The radiation scheme solves the radiative transfer equation in two distinct spectral regions. The computations for the longwave (LW) radiation are performed over the spectrum from 0 to 2820 cm^{-1}. The shortwave (SW) part of the scheme integrates the fluxes over the whole shortwave spectrum between 0.2 and 4.0 μm. The scheme used for data assimilation purposes must be computationally efficient to be called at each time step and at the full spatial resolution for an improved description of the cloud-radiation interactions during the assimilation period (Janisková *et al.*, 2002).

3.7.1 The short-wave radiation scheme

The linearized code for the shortwave radiation scheme has been derived from the ECMWF original non-linear scheme developed by Fouquart and Bonnel (1980) and revised by Morcrette (1991), and which was previously used in the operational forecast model. In this scheme, the photon-path-distribution method is used to separate the parametrization of the scattering processes from that of molecular absorption. Upward $F_{sw}^{\uparrow}$ and downward $F_{sw}^{\downarrow}$ fluxes at a given level j are obtained from the

reflectance and transmittance of the atmospheric layers as

$$F_{\text{sw}}^{\downarrow}(j) = F_0 \prod_{k=j}^{N} T_{\text{bot}}(k) \tag{3.18}$$

$$F_{\text{sw}}^{\uparrow}(j) = F_{\text{sw}}^{\downarrow}(j) R_{\text{top}}(j-1) \tag{3.19}$$

Computations of the transmittance at the bottom of a layer T_{bot} start at the top of atmosphere and work downward. Those of the reflectance at the top of the same layer R_{top} start at the surface and work upward. In the presence of cloud in the layer, the final fluxes are computed as a weighted average of the fluxes in the clear sky and in the cloudy fractions of the column as

$$R_{\text{top}} = C_{\text{cloud}} R_{\text{cloud}} + (1 - C_{\text{cloud}}) R_{\text{clear}} \tag{3.20}$$

$$T_{\text{top}} = C_{\text{cloud}} T_{\text{cloud}} + (1 - C_{\text{cloud}}) T_{\text{clear}} \tag{3.21}$$

In the previous equations, C_{cloud} is the cloud fractional coverage of the layer within the cloudy fraction of the column (depending on the cloud-overlap assumption).

The non-linear version of the shortwave radiation scheme (described in details in section IV: Physical processes - 2.2. The pre-Cy32r2 shortwave radiation scheme) used six spectral intervals with transmission functions derived from a line-by-line code (Dubuisson *et al.*, 1996).

The non-linear scheme is reasonably fast for application in 4D-Var and has, therefore, been linearized without a priori modifications. The only modification with respect to the non-linear model is using two spectral intervals with transmission functions (instead of six intervals used in the former operational non-linear model) in order to reduce the computational cost.

3.7.2 The longwave radiation scheme

The longwave radiation scheme, operational in the ECMWF forecast model until June 2000, was a band emissivity type scheme (Morcrette, 1989). This scheme has been replaced by the Rapid Radiation Transfer Model (RRTM: (Mlawer *et al.*, 1997) in June 2000. The complexity of the RRTM scheme for the longwave part of the spectrum makes accurate computations expensive. In the variational assimilation framework, the former operational scheme (Morcrette, 1989) has been linearized. In this scheme, the longwave spectrum from 0 to 2820 cm^{-1} is divided into six spectral regions. The transmission functions for water vapour and carbon dioxide over those spectral intervals are fitted using Padé approximations on narrow-band transmissions obtained with statistical band models (Morcrette *et al.*, 1986). Integration of the radiation transfer equation over wavenumber ν within the particular spectral regions gives the upward and downward fluxes.

The incorporation of the effects of clouds on the longwave fluxes follows the treatment discussed by (Washington and Williamson, 1977). The scheme calculates first upward and downward fluxes ($F_0^{\uparrow}(i)$ and $F_0^{\downarrow}(i)$) corresponding to a clear-sky atmosphere. In any cloudy layer, the scheme evaluates the fluxes assuming a unique overcast cloud of emissivity unity, i.e. $F_n^{\uparrow}(i)$ and $F_n^{\downarrow}(i)$ for a cloud present in the nth layer of the atmosphere. The fluxes for the actual atmosphere are derived from a linear combination of the fluxes calculated in the previous steps with some cloud overlap assumption in the case of clouds present in several layers. Let N be the number of model layers starting from the top of atmosphere to the bottom, C_i the fractional cloud cover in layer i, the cloudy upward $F_{lw}^{\uparrow}$ and downward $F_{lw}^{\downarrow}$ fluxes are expressed as:

$$F_{lw}^{\uparrow}(i) = (1 - CC_{N,i}) F_0^{\uparrow}(i) + \sum_{k=i}^{N} (CC_{i,k+1} - CC_{i,k}) F_k^{\uparrow}(i) \tag{3.22}$$

$$F_{lw}^{\downarrow}(i) = (1 - CC_{i-1,0}) F_0^{\downarrow}(i) + \sum_{k=1}^{i-1} (CC_{i,k+1} - CC_{i,k}) F_k^{\downarrow}(i) \tag{3.23}$$

where $CC_{i,j}$ is the cloudiness encountered between any two levels i and j in the atmosphere computed using a certain overlap assumption.

In the case of semi-transparent clouds, the fractional cloudiness entering the calculations is an effective cloud cover equal to the product of the emissivity (ε_{cld}) due to condensed water and gases in the layer by the horizontal coverage of the cloud cover. This is the so called effective emissivity approach. The detailed description of this longwave radiation scheme can be found in Section IV - Physical processes (2.2. The pre-Cy22r3 longwave radiation scheme).

To reduce a computational cost of the linearized longwave radiation for data assimilation, the transmission functions are only computed for H_2O and CO_2 absorbers (though the version taking into account the whole spectrum of absorbers is also coded). The cloud effects on LW radiation are only computed to the level defined by the current cloud top height.

3.7.3 Cloud overlap assumptions

Cloud overlap assumptions must be made in atmospheric models in order to organize the cloud distribution used for radiation and precipitation/evaporation computations. A cloud overlap assumption of some sort is necessary to account for the fact that clouds often do not fill the whole grid box. The maximum-random overlap assumption (originally introduced in Geleyn and Hollingsworth, 1997) is used operationally in the ECMWF model (Morcrette, 2000). Adjacent layers containing cloud are combined by using maximum overlap to form a contiguous cloud and discrete layers separated by clear-sky are combined randomly as

$$CC_{i,j} = 1 - (1 - C_i) \prod_{k=i+1}^{j-1} \left[\frac{1 - \max(C_k, C_{k-1})}{1 - C_{k-1}} \right] \tag{3.24}$$

where $CC_{i,j}$ is cloudiness encountered between any levels i and j in the atmosphere and C_k is the cloud fraction of the layer k located between levels k and $k + 1$.

3.7.4 Cloud optical properties

Considering the cloud-radiation interactions, it is not only the cloud fraction or cloud volume, but also cloud optical properties that matter. In the case of shortwave radiation, the cloud radiative properties depend on three different parameters: the optical thickness δ_c, the asymmetry factor g_c and the single scattering albedo $\overline{\omega_c}$. They are derived from Fouquart (1987) for the water clouds, and Ebert and Curry (1992) for the ice clouds. The optical thickness δ_c is related to the cloud liquid/ice water amount u_{LWP} by :

$$\delta_c = u_{LWP} \left(a_i + \frac{b_i}{r_e} \right) \tag{3.25}$$

where a_i and b_i are defined from Ebert and Curry (1992) for ice particles and are set to respectively 0 and 3/2 for water particles. The mean effective radius of the size distribution of the cloud water droplets is defined by r_e. For water clouds, r_e is set to 13 μm over oceans and to 10 μm over continents. When ice cloud optical properties were initially introduced in the radiation code, the effective radius was set to 40 μm. However, observations indicate that the effective radius of ice crystals increases with temperature, usually attributed to accretion from falling crystals. In the current scheme, account is taken of this by using the diagnostic formulation of Ou and Liou (1995):

$$r_e = 326.3 + 12.42 \times T_i + 0.197 \times T_i^2 + 0.0012 \times T_i^3 \tag{3.26}$$

where $T_i = \min(T, -23^{\circ}C)$. The effective radius is then limited within the interval 30–60 μm.

In the two spectral intervals of the shortwave scheme, the asymmetry factor g_c is fixed to 0.865 and 0.910, respectively and ω_c is given as a function of δ_c following Fouquart (1987):

$$\overline{\omega}_{c1} = 0.9999 - 5 \times 10^{-4} \exp(-0.5\delta_c) \tag{3.27}$$

$$\overline{\omega}_{c2} = 0.9988 - 2.5 \times 10^{-3} \exp(-0.05\delta_c) \tag{3.28}$$

These cloud shortwave radiative parameters have been fitted to in-situ measurements of stratocumulus clouds (Bonnel et al., 1983).

The optical properties of ice clouds are expressed as:

$$\overline{\omega}_i = c_i - d_i r_e \tag{3.29}$$

$$g_i = e_i + f_i r_e \tag{3.30}$$

where the coefficients are derived from Ebert and Curry (1992).

Cloud longwave optical properties are represented by the emissivity ε_{cld} related to the condensed water amount and by the condensed water mass absorption coefficient k_{abs}. The emissivity ε_{cld} is related to the condensed water amount by:

$$\varepsilon_{cld} = 1 - \exp(-k_{abs}u_{LWP}) \tag{3.31}$$

where k_{abs} is the condensed water mass absorption coefficient obtained following Smith and Shi (1992) for the water clouds and Ebert and Curry (1992) for the ice clouds. u_{LWP} is the condensed water path. k_{abs} depends upon the water phase (ice or water) and upon temperature. A spectral dependency can also be included.

Linearized shortwave and longwave radiation schemes are activated using the logical switch LERADSN2=.TRUE. in the namelist NAMTRAJP.

To use the tangent-linear (TL) and adjoint (AD) versions of the shortwave radiation scheme, LERADSW2 should be set to .TRUE. in NAMTRAJP. The default value is set to .FALSE.. To decrease a computational cost of the shortwave radiation in data assimilation, the number of spectral intervals is reduced to two (six intervals are used in the forecast model) during minimization. The number of spectral intervals NSW is changed in CVA1. This change also requires to read several parameters for two spectral intervals. This is done in set-up routines SUSWN, SUCLOPN and SUAERSN.

The TL/AD longwave radiation scheme can be activated by setting LERADLW2=.TRUE. in the namelist NAMTRAJP (the default value is .FALSE.). To use a time/memory optimized version of the code, additional set-up (requirement for the shorter loops inside of NPROMA by setting LOPTLWPR=.TRUE. and definition of the number of loops inside of NPROMA, currently NLOOPLW=3) should also be included in NAMTRAJP. For further optimization, cloud effects on the longwave radiation are only computed to the certain level determined from the cloud top height. This is activated by LWLCLHR=.TRUE. in NAMTRAJP. Using this optimization, computational cost is decreased, but TL and AD results are not bit reproducible. If reproducibility is required in 4D-Var (LREPRO4DVAR=.TRUE.) the switch LWLCLHR is automatically set to .FALSE.. The same can be achieved by using default value for LWLCLHR, which is .FALSE..

The linearized radiation schemes are called from RADINATL and RADINAAD, where the computation of radiation fluxes is performed. Tendencies produced by the linearized longwave and shortwave radiation are computed in RADHEATTL and RADHEATAD. All those routines are called from CALLPARTL and CALLPARAD, respectively.

RADLSWTL and RADLSWAD are the drivers for the computation of the solar and thermal fluxes by calling specialized routines for shortwave radiation (SWTL and SWAD) and for longwave radiation (LWTL and LWAD).

3.8 LARGE-SCALE CONDENSATION AND PRECIPITATION

The original version of the simplified diagnostic large-scale cloud and precipitation scheme currently used in the minimization of 4D-Var is described in Tompkins and Janisková (2004). This scheme replaced the much simpler large-scale precipitation parametrization of Mahfouf (1999), which was used in operations until Cy33r1.

The physical tendencies of temperature and specific humidity produced by moist processes on the large-scale can be written as

$$\frac{\partial q}{\partial t} = -\mathcal{C} + E_{prec} + \mathcal{D}_{conv} \tag{3.32}$$

$$\frac{\partial T}{\partial t} = L(\mathcal{C} - E_{prec} - \mathcal{D}_{conv}) + L_f(\mathcal{F} - \mathcal{M}) \tag{3.33}$$

where $\mathcal{C}$ denotes large-scale condensation (negative if evaporation), E_{prec} is the moistening due to the evaporation of precipitation and $\mathcal{D}_{\text{conv}}$ is the detrainment of cloud water from convective clouds. $\mathcal{F}$ and $\mathcal{M}$ correspond to the freezing of rain and melting of snow, respectively. L and L_{f} are the latent heats of vaporisation/sublimation and fusion, respectively.

3.8.1 Stratiform condensation

The subgrid-scale variability of humidity is assumed to be represented by a uniform distribution with half width W. Condensation inside the model gridbox occurs whenever gridbox mean relative humidity, RH, exceeds a critical threshold, RH_{crit}. The half width of the distribution is given by

$$W = q_{\text{sat}}\left\{1 - RH_{\text{crit}} - \kappa(RH - RH_{\text{crit}})\right\} \tag{3.34}$$

The critical relative humidity threshold, RH_{crit}, is assumed to be dependent on reduced pressure $\sigma = p/p_{\text{surf}}$, through

$$RH_{\text{crit}} = 0.7\sigma(1 - \sigma)(1.85 + 0.95(\sigma - 0.5)) \tag{3.35}$$

κ is a coefficient that reduces the width of the uniform distribution when RH increases and it also depends on σ as

$$\kappa = \max(0, 0.9(\sigma - 0.2)^{0.2}) \tag{3.36}$$

Equation (3.34) together with the assumption of a uniform distribution yields the following relationships for stratiform cloud cover, C_{strat}, and cloud condensate specific ratio, $q_{\text{c}}^{\text{strat}}$:

$$C_{\text{strat}} = 1 - \sqrt{\frac{1 - RH}{1 - RH_{\text{crit}} - \kappa(RH - RH_{\text{crit}})}} \tag{3.37}$$

$$q_{\text{c}}^{\text{strat}} = q_{\text{sat}}C_{\text{strat}}^2\left\{\kappa(1 - RH) + (1 - \kappa)(1 - RH_{\text{crit}})\right\} \tag{3.38}$$

where q_{sat} is the saturation specific humidity.

Figure Fig. 3.1 displays the variations of stratiform cloud cover, C_{strat}, as a function of relative humidity for various settings of parameter κ.
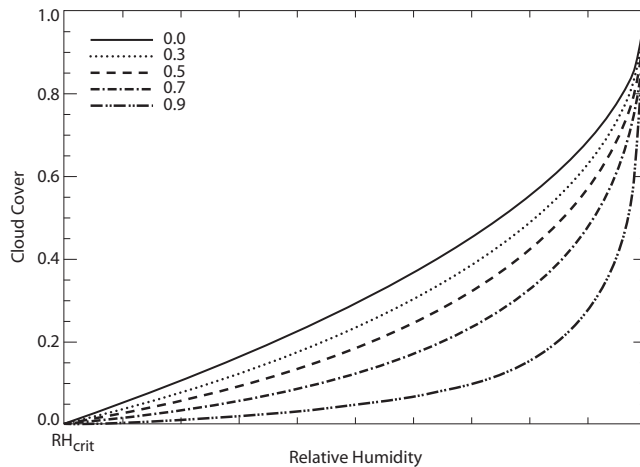

Figure 3.1 *Stratiform cloud cover, C_{strat}, as a function of grid-box mean relative humidity for various values of parameter κ (see top left legend). Relative humidity ranges between RH_{crit} and unity (i.e. grid box is entirely saturated).*

3.8.2 Convective contribution

The impact of convective activity on large-scale clouds, which is particularly important in the tropics and mid-latitude summers, is accounted for through the detrainment term coming out of the convection scheme (see Section 3.9). The additional cloud cover, C_{conv}, and cloud condensate, q_c^{conv}, resulting from convection are computed as

$$C_{\text{conv}} = (1 - C_{\text{strat}})\left\{1 - \exp\left(\frac{-\delta_u M_u \Delta t}{\rho \Delta z}\right)\right\} \tag{3.39}$$

$$q_c^{\text{conv}} = \delta_u M_u q_l^u \frac{\Delta t}{\rho \Delta z} \tag{3.40}$$

where δ_u (unitless), M_u (in kg m^{-2} s^{-1}) and q_l^u (in kg kg^{-1}) are the detrainment rate, mass flux and cloud liquid water content in the convective updraught, respectively. Δt and Δz denote the model time step and model layer depth, while ρ is the air density.

3.8.3 Precipitation formation

The formation of precipitation from cloud condensate, q_c, is parameterized according to Sundqvist *et al.* (1989). The corresponding tendency writes

$$\left(\frac{\partial q_c}{\partial t}\right)_{\text{prec}} = -C_0\, q_c \left\{1 - \exp\left[-\left(\frac{q_c}{Cq_c^{\text{crit}}}\right)^2\right]\right\} \tag{3.41}$$

where $C = C_{\text{strat}} + C_{\text{conv}}$ is used to obtain in-cloud condensate amounts. The critical cloud water threshold, q_c^{crit} is set to 3×10^{-4} kg kg^{-1} if precipitation evaporation is activated, 6×10^{-4} kg kg^{-1} otherwise. The conversion factor C_0 is set to 3.33×10^{-4} s^{-1}. Note that the Bergeron-Findeisen mechanism and collection processes are disregarded in the current version of the simplified scheme.

In the code, the new cloud water content after precipitation formation is calculated as

$$q_c^{\text{new}} = q_c\, \exp(-D) \tag{3.42}$$

where the quantity D is computed as

$$D = C_0\, \Delta t \left\{1 - \exp\left[-\left(\frac{q_c}{Cq_c^{\text{crit}}}\right)^2\right]\right\} \tag{3.43}$$

It is worth noting that precipitation formed from cloud liquid water at temperatures below the freezing point is assumed to freeze instantly, which corresponds to term $\mathcal{F}$ in (3.33).

3.8.4 Precipitation evaporation

The scheme partially accounts for the overlap of precipitation with the subgrid clear-sky distribution of humidity fluctuations (uniformly distributed). Precipitation evaporation computations are based on the mean specific humidity, $\overline{q_{\text{prec}}^{\text{clr}}}$, in the clear-sky portion through which precipitation is falling, denoted $f_{\text{prec}}^{\text{clr}}$. Consistent with the maximum overlap assumption made in the calculations of total cloud cover and precipitation fraction, it is hypothesized that $f_{\text{prec}}^{\text{clr}}$ corresponds to the moistest part of the uniform distribution, as illustrated in Fig. 3.2.

From this, one can show that

$$\overline{q_{\text{prec}}^{\text{clr}}} = q_{\text{sat}} - \frac{f_{\text{prec}}^{\text{clr}}(q_{\text{sat}} - \overline{q})}{(1 - C)^2} \tag{3.44}$$

where $\overline{q}$ denotes the gridbox mean specific humidity.

Precipitation evaporation is parameterized following Kessler (1969):

$$\left(\frac{\partial q}{\partial t}\right)_{\text{evap}} = f_{\text{prec}}^{\text{clr}}\beta(q_{\text{sat}} - \overline{q_{\text{prec}}^{\text{clr}}}) \tag{3.45}$$
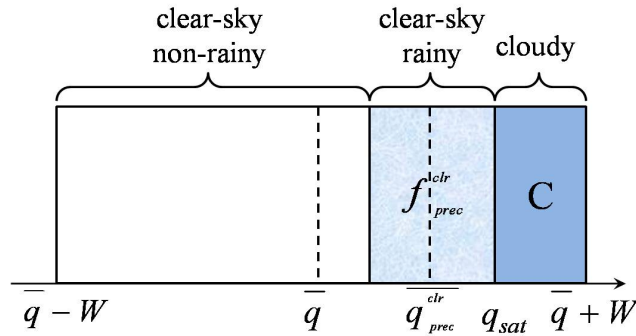
Figure 3.2 *Illustration of the uniform distribution of specific humidity over the model grid box with various quantities referred to in the text.*

where β is given by

$$\beta = 5.44 \times 10^{-4} \left\{ \left(\frac{p}{p_{\text{surf}}} \right)^{\frac{1}{2}} \frac{P_{\text{clr}}}{5.9 \times 10^{-3} f_{\text{prec}}^{\text{clr}}} \right\}^{0.577} \tag{3.46}$$

where P_{clr} denotes precipitation in the clear-sky fraction of the grid box. Equation (3.45) is solved implicitly to take into account the reduction of q_{sat} due to evaporative cooling, which yields the moistening associated to precipitation evaporation:

$$E_{\text{prec}} = \frac{f_{\text{prec}}^{\text{clr}} \beta \Delta t (q_{\text{sat}} - \overline{q_{\text{prec}}^{\text{clr}}})}{1 + \beta \Delta t \left[1 + \frac{L_{\text{v}}}{c_{\text{p}}} \frac{\partial q_{\text{sat}}}{\partial T} \right]} \tag{3.47}$$

Finally, the loss of precipitation through evaporation (in kg m^{-2} s^{-1}) is simply expressed as $\Delta P_{\text{evap}} = -E_{\text{prec}} \Delta p / (g \Delta t)$.

3.8.5 Phase partitioning

A simple diagnostic partitioning based on temperature is used to separate cloud condensate into liquid and ice. The liquid water fraction, α_l, is calculated as

$$\alpha_l = \begin{cases} 0.545 \times \tanh\left[0.17(T - 266.41) + 1 \right] & \text{if } T < 273.15 \text{ K} \\ \\ 1 & \text{if } T \geq 273.15 \text{ K} \end{cases} \tag{3.48}$$

3.8.6 Code

The computations performed in the simplified large-scale cloud and precipitation scheme are performed in subroutine CLOUDST, which is called from the main physics routine CALLPAR. LENCLD2 (namelist NAMTRAJP) is the main switch that activates this simplified scheme in non-linear and linearized (tangent-linear and adjoint) integrations. The switch LEVAPLS2 (namelist NAMTRAJP) controls the activation of the precipitation evaporation computations (default setting is .FALSE.). The tangent-linear and adjoint versions of the scheme are CLOUDSTTL and CLOUDSTAD, respectively.

SET-UP OF PARAMETERS

Defined in subroutines SUCLDP and SU0PHY.

3.8.7 Regularization

Special care had to be taken to avoid the spurious growth of some perturbations in the tangent-linear and adjoint versions of the simplified large-scale cloud and precipitation code. Problematic perturbations are artificially reduced, as described below. Switch LREGCL should be set to .TRUE. in namelist NAMTRAJP to activate these regularizations in the linearized code. In the following, the prime symbol is employed to denote perturbations.

CLOUDSTTL/CLOUDSTAD:

- After (3.37): $C'_{\text{strat}} = C'_{\text{strat}} \times \begin{cases} -1.2\, C_{\text{strat}} + 0.94 & \text{if } 0.2 < C_{\text{strat}} < 0.7 \\ 0.1 & \text{if } 0.7 \le C_{\text{strat}} < 0.95 \\ 0.1\sqrt{(1 - C_{\text{strat}})/(1 - 0.95)} & \text{if } C_{\text{strat}} \ge 0.95 \end{cases}$

- D' is scaled by 0.01 in autoconversion of cloud water to precipitation (3.43).

3.9 MOIST CONVECTION

The original version of the simplified mass-flux convection scheme currently used in the minimization of 4D-Var is described in Lopez and Moreau (2005). It replaced the much simpler convective parametrization of Mahfouf (1999), which was used in operations until Cy33r1. Through time, the original scheme from Lopez and Moreau (2005) has been updated so as to gradually converge towards the full convection scheme used in high-resolution 10-day forecasts (see Chapter 6 of Part IV). The transport of tracers by convection has also been added.

The physical tendencies produced by convection on any conservative variable ψ (dry static energy, wind components, specific humidity, cloud liquid water) can be written in a mass-flux formulation as Betts (1997)

$$\frac{\partial \psi}{\partial t} = \frac{1}{\rho}\left[(M_{\text{u}} + M_{\text{d}})\frac{\partial \psi}{\partial z} + D_{\text{u}}(\psi_{\text{u}} - \psi) + D_{\text{d}}(\psi_{\text{d}} - \psi)\right] \tag{3.49}$$

The first term on the right hand side represents the compensating subsidence induced by cumulus convection on the environment through the mass flux, M. The other terms accounts for the detrainment of cloud properties in the environment with a detrainment rate, D. Subscripts u and d refer to the updraughts and downdraughts properties, respectively. Evaporation of cloud water and precipitation should also be added in (3.49) for dry static energy, $s = c_p T + gz$, and specific humidity, q.

3.9.1 Equations for updraught and downdraught

The equations describing the evolution with height of the convective updraught and downdraught mass fluxes, M_{u} and M_{d} (units kg m^2 s^{-1}), respectively, are

$$\frac{\partial M_{\text{u}}}{\partial z} = (\epsilon_{\text{u}} - \delta_{\text{u}})M_{\text{u}} \tag{3.50}$$

$$\frac{\partial M_{\text{d}}}{\partial z} = -(\epsilon_{\text{d}} - \delta_{\text{d}})M_{\text{d}} \tag{3.51}$$

where ϵ and δ respectively denote the entrainment and detrainment rates (in m^{-1}). A second set of equations is used to describe the evolution with height of any other characteristic, ψ, of the updraught or downdraught, namely

$$\frac{\partial \psi_{\text{u}}}{\partial z} = -\epsilon_{\text{u}}(\psi_{\text{u}} - \overline{\psi}) \tag{3.52}$$

$$\frac{\partial \psi_{\text{d}}}{\partial z} = \epsilon_{\text{d}}(\psi_{\text{d}} - \overline{\psi}) \tag{3.53}$$

where $\overline{\psi}$ is the value of ψ in the large-scale environment.

In practice, (3.50) and (3.51) are solved in terms of $\mu = M/M_{\text{u}}^{\text{base}}$, where $M_{\text{u}}^{\text{base}}$ is the mass flux at cloud base (determined from the closure assumptions as described further down). μ is equal to 1 at cloud base for the updraught and is set to -0.3 at the Level of Free Sinking (LFS) in the case of the downdraught. The LFS is assumed to be the highest model level (below the level of minimum moist static energy) where a mixture of equal parts of cloud and saturated environmental air at the wet-bulb temperature becomes negative buoyant with respect to the environmental air. In other words, LFS corresponds to the starting level of the downdraught. The solution profiles of μ values are eventually multiplied by M_u^{base} to obtain the final mass-flux profiles.

3.9.2 Triggering of moist convection

The determination of the occurrence of moist convection in the model is based on whether a positively buoyant test parcel starting at each model level (iteratively from the surface and upwards) can rise high enough to produce a convective cloud and possibly precipitation. Shallow convection is first tested for by considering a parcel rising from the lowest model level. New test parcels starting from model levels of increasing altitude are then considered to identify deep convection, this time. This procedure is repeated until deep convection is found or until the departure altitude of the parcel reaches about 15 km.

The initial characteristics of the test parcel originating from the lowest model level (to test for shallow convection) are derived from surface sensible and latent turbulent heat fluxes (J_s and J_q, respectively). The initial vertical velocity of the test parcel is assumed to be equal to the convective-scale vertical velocity, w_*, defined as

$$w_* \quad = \quad 1.2\left(u_*^3 - 1.5\frac{gz\kappa}{\overline{\rho T}}\left[\frac{J_s}{c_p} + 0.608\overline{T}\frac{J_q}{L_v}\right]\right) \tag{3.54}$$

where $\kappa=0.4$ is the von Karman constant and the friction velocity u_* is set to a constant value of 0.1 m s^{-1}.

Following Jakob and Siebesma (2003), the temperature excess, ΔT_u, and moisture excess, Δqu, of the test parcel with respect to the environment are prescribed as

$$\Delta T_u = -1.5\frac{J_s}{\overline{\rho}c_p w_*} \quad \text{and} \quad \Delta q_u = -1.5\frac{J_q}{\overline{\rho}L_v w_*} \tag{3.55}$$

For a test parcel initiated higher than the lowest model level (i.e. deep convection test), its initial vertical velocity is arbitrarily set to 1 m s^{-1}, while its temperature and moisture excesses are assumed to be

$$\Delta T_u = 0.2\text{ K} \quad \text{and} \quad \Delta q_u = 1 \times 10^{-4}\text{ kg kg}^{-1} \tag{3.56}$$

Furthermore, in the lowest 60 hPa of the atmosphere that typically correspond to the mixed-layer depth over oceanic regions, the updraught values of the dry static energy and moisture at the departure level k are initialized as $s_k^u = \tilde{s}_k + c_p\Delta T_u$, where the tilde symbol represents a 50 hPa layer average, instead of $s_k^u = \overline{s}_k + c_p\Delta T_u$ as for departure levels above the 60 hPa mixed-layer.

If convection is found for the parcel initiated from the lowest level, it is classified as shallow convection if cloud depth is lower than 200 hPa. If convection is found for a parcel initiated above the lowest model level, it is classified as deep convection provided cloud depth exceeds 200 hPa. A grid point affected by both shallow and deep convection is treated as deep convection. Note that mid-level convection is not distinguished from deep convection.

3.9.3 Entrainment and detrainment

Updraught

For deep convection, the entrainment rate to be used in (3.50) and (3.52) for the updraught is specified as:

$$\epsilon_u \quad = \quad \begin{cases} \text{ENTRORG2} \times \left[1.3 - \min(\overline{RH}, 1)\right]\min\left(1, \left(\frac{\overline{q_{sat}}}{q_{sat}^{base}}\right)^3\right) & \text{for } z > z_{base} \\ 0 & \text{otherwise} \end{cases} \tag{3.57}$$

where $\text{ENTRORG2} = 1.75 \times 10^{-4}$ m^{-1}. ϵ_u is therefore a function of the local relative humidity in the environment, $\overline{RH}$, and of the ratio between the local saturation specific humidity, $\overline{q_{sat}}$, and its value at the base of the convective cloud.

For shallow convection, the entrainment rate is set to twice (ENTSHALP2) the value computed in (3.57) for deep convection.

Turbulent detrainment in the updraught is specified as

$$\delta_u^* \quad = \quad \begin{cases} \text{DETRPEN2} \times \left[1.6 - \min(\overline{RH}, 1)\right] & \text{for deep convection} \\ \epsilon_u\left[1.6 - \min(\overline{RH}, 1)\right] & \text{for shallow convection} \end{cases} \tag{3.58}$$

where DETRPEN2 $= 0.75 \times 10^{-4} \, \text{m}^{-1}$.

Updraught detrainment is also assumed to occur inside the convective cloud wherever the vertical gradient of updraught kinetic energy ($\partial w_{\mathrm{u}}^2 / \partial z$) and buoyancy are negative (i.e. usually in the upper part of the convective cloud). Eventually, the total detrainment rate, δ_{u}, between model levels $k + 1$ and k is given by

$$\delta_{\mathrm{u}} \quad = \quad \max \left(\delta_{\mathrm{u}}^*, 1 - \sqrt{\frac{(w_{\mathrm{u}}^2)_k}{(w_{\mathrm{u}}^2)_{k+1}}} \right) \tag{3.59}$$

Any remaining updraught mass flux is supposed to be detrained at the level where w_{u} vanishes.

Downdraught

For all types of convection, the entrainment rate in the downdraught is set to

$$
\epsilon_{\mathrm{d}} = \begin{cases} \mathrm{ENTRDD2} + \epsilon_{\mathrm{org}} & \text{for } z_{\mathrm{topde}} \leq z \leq z_{\mathrm{LFS}} \\ 0 & \text{elsewhere} \end{cases}
\tag{3.60}
$$

where $\mathrm{ENTRDD2} = 3 \times 10^{-4}$ m^{-1}. Subscript *topde* denotes the model level corresponding to the top of the 60 hPa atmospheric layer just above the surface. Note that the value of $\epsilon_{\mathrm{d}}\Delta z$ is capped to a maximum of 0.3. The organized entrainment term, ϵ_{org}, is derived from (Nordeng, 1994) as

$$
\epsilon_{\mathrm{org}} = \frac{\left\{ g \frac{T_{\mathrm{v,d}} - T_{\mathrm{d}} r_{\mathrm{d}} - \bar{T}_{\mathrm{v}}}{\bar{T}_{\mathrm{v}}} \right\}}{(w_{\mathrm{d}}^{\mathrm{LFS}})^2 - \int_{z_{\mathrm{LFS}}}^{z} \left\{ g \frac{T_{\mathrm{v,d}} - T_{\mathrm{d}} r_{\mathrm{d}} - \bar{T}_{\mathrm{v}}}{\bar{T}_{\mathrm{v}}} \right\} \mathrm{d}z}
\tag{3.61}
$$

where $w_{\mathrm{d}}^{\mathrm{LFS}}$ is the vertical velocity in the downdraught at the LFS (set to -1 ms^{-1}). The evaporation rate in the downdraft at a given level corresponds to the downdraft precipitation rate and is simply given as

$$
e_{\mathrm{d}} = \frac{g}{\Delta p}(q_{\mathrm{d}} - \hat{q}_{\mathrm{d}})M_{\mathrm{d}}
\tag{3.62}
$$

where q_{d} is the value of the downdraft humidity before saturation adjustment and $\hat{q}_{\mathrm{d}}$ is the humidity after the saturation adjustment. The value of the rain water content in the downdraft used in (3.61) is estimated as $r_{\mathrm{d}} = e_{\mathrm{d}}\Delta p / (gM_{\mathrm{u}})$.

Detrainment in the downdraught is defined as

$$
\delta_{\mathrm{d}} = \begin{cases} \frac{1}{z_{\mathrm{topde}}} \frac{M_{\mathrm{d}}^{\mathrm{topde}}}{M_d} & \text{for } z < z_{\mathrm{topde}} \\ \mathrm{ENTRDD2} & \text{for } z_{\mathrm{topde}} \leq z \leq z_{\mathrm{LFS}} \\ 0 & \text{for } z > z_{\mathrm{LFS}} \end{cases}
\tag{3.63}
$$

This formulation ensures a downward linear decrease of downdraught mass flux to zero at the surface.

Special case of momentum

To describe momentum exchange between the updraught and the environment, the entrainment rate is specified as a linear combination of the entrainment and detrainment rates that are applied to all other variables (as given in (3.57) and (3.59))

$$
\epsilon_u^{\mathrm{mom}} = \begin{cases} \epsilon_{\mathrm{u}} + 2\delta_{\mathrm{u}} & \text{in all types of convection} \\ \epsilon_{\mathrm{u}} + 3\delta_{\mathrm{u}} & \text{at the top three levels of deep convective cloud} \\ 0 & \text{otherwise} \end{cases}
\tag{3.64}
$$

3.9.4 Precipitation formation

The formation of precipitation from the cloud water contained in the updraught (q_l^u) is parameterized according to Sundqvist *et al.* (1989). The corresponding tendency writes:

$$
\left(\frac{\partial q_l^{\mathrm{u}}}{\partial z} \right)_{\mathrm{prec}} = -C_0 \, C_{\mathrm{BF}} \, q_l^{\mathrm{u}} \left\{ 1 - \exp \left[-\left(\frac{q_l^{\mathrm{u}}}{q_l^{\mathrm{crit}}} \right)^2 \right] \right\}
\tag{3.65}
$$

where the critical cloud water threshold, q_l^{crit} is set to 5×10^{-4} (resp. 3×10^{-4}) kg kg^{-1} over land (resp. sea). Conversion factors C_0 and C_{BF} (Bergeron-Findeisen mechanism) are given by:

$$
C_0 = \frac{1.4 \times 10^{-3}}{0.75 w_{\mathrm{u}}}
\tag{3.66}
$$

$$
C_{\mathrm{BF}} = \begin{cases} 1 + 0.5\sqrt{\min(T_{\mathrm{BF}} - T_{\mathrm{ice}}, T_{\mathrm{BF}} - T_{\mathrm{u}})} & \text{if } T_{\mathrm{u}} < T_{\mathrm{BF}} \\ 1 & \text{if } T_{\mathrm{u}} \geq T_{\mathrm{BF}} \end{cases}
\tag{3.67}
$$

where w_u and T_u are the vertical velocity and the temperature of the updraught, and $T_{\text{ice}}=250.16$ K $(-23°C)$ and $T_{\text{BF}}=268.16$ K $(-5°C)$.

In the code, (3.65) is integrated over each model layer during the updraught ascent. The new cloud water content in the updraught at a given model level k after precipitation formation is expressed as a function of the initial upraught cloud water content at model level $k+1$ and of the additional amount of condensate (*Cond*) produced by convection during the ascent from levels $k+1$ to k (by convention, model level number increases downwards):

$$(q_l^{\text{u}})_k = (q_l^{\text{u}})_{k+1} \exp(-D) + Cond \frac{1-\exp(-D)}{D} \tag{3.68}$$

where the quantity D is computed as

$$D = C_0 C_{\text{BF}} \Delta z \left\{ 1 - \exp\left[-\left(\frac{(q_l^{\text{u}})_{k+1}}{q_l^{\text{crit}}}\right)^2 \right] \right\} \tag{3.69}$$

It is worth noting that precipitation formed from cloud liquid water at temperatures below the freezing point is assumed to freeze instantly.

3.9.5 Closure assumptions

One needs to formulate so-called closure assumptions to compute the convective updraught mass-flux at cloud base, $M_{\text{u}}^{\text{base}}$, from quantities that are explicitly resolved by the model.

Deep convection

The closure is based on the balance between the convective available potential energy (*CAPE*) in the subgrid-scale updraught and the total heat release (*HEAT*) in the resolved larger-scale environment. *CAPE* and *HEAT* are computed as vertical integrals over the depth of the convective cloud as

$$CAPE = \int_{\text{cloud}} g\left(\frac{T_{\text{v}}^{\text{u}} - \overline{T_{\text{v}}}}{\overline{T_{\text{v}}}} - q_l^{\text{u}}\right) dp \tag{3.70}$$

$$HEAT = \int_{\text{cloud}} g\left(\frac{1}{c_p \overline{T}}\frac{\partial \overline{s}}{\partial z} + 0.608\frac{\partial \overline{q}}{\partial z}\right)(\mu_{\text{u}} + \mu_{\text{d}}) dz \tag{3.71}$$

where T_v denotes virtual temperature.

To account for non-equilibrium convective situations, an extra term corresponding to the departure of *CAPE* from equilibrium is computed following Bechtold *et al.* (2014) as

$$CAPE_{\text{bl}} = -\tau_{\text{bl}}\frac{1}{T_\star}\int_{p_{\text{surf}}}^{p_{\text{base}}} \left.\frac{\partial \overline{T}_{\text{v}}}{\partial t}\right|_{\text{bl}} dp \tag{3.72}$$

In (3.72), the tendency $\partial \overline{T}_{\text{v}}/\partial t|_{\text{bl}}$ includes all boundary-layer processes other than convection, i.e. turbulent diffusion, radiation and advection; therefore the corresponding tendencies must be available before calling the convection scheme. The temperature scale $T_\star = c_p^{-1} gH$ is set to 1 K and different boundary-layer time-scales τ_{bl} are used over land and water (namely the convective turnover time-scale over land and the advective time-scale over water)

$$\tau_{\text{bl}} = \begin{cases} \dfrac{H_{\text{cld}}}{\overline{w}_{\text{u}}} & \text{over land} \\[2ex] \dfrac{H_{\text{base}}}{\overline{u}_{\text{bl}}} & \text{over water} \end{cases} \tag{3.73}$$

where $\overline{w}_{\text{u}}$ is the updraught vertical velocity averaged over the cloud depth, H_{cld}, while H_{base} is the cloud base height and $\overline{u}_{\text{bl}}$ the average horizontal wind speed in the subcloud layer.

With this definition CAPE$_{bl}$ can also be seen as an efficient sine filter on CAPE. CAPE$_{bl}$ is set to zero for convection originating above the boundary layer.

Finally, the cloud-base mass flux for deep convection situations is given by:

$$M_{\mathrm{u}}^{\mathrm{base}} = \frac{CAPE - CAPE_{bl}}{HEAT \times \tau} \tag{3.74}$$

where τ is an adjustment timescale (in seconds), which is expressed as

$$\tau = (1 + 264/n_{\mathrm{T}})H_{\mathrm{cld}}/\overline{w_{\mathrm{u}}} \tag{3.75}$$

where n_{T} denotes the spectral truncation.

Shallow convection

The closure assumption links the moist-energy excess at cloud base, δh_{base}, to the moist-energy convergence inside the sub-cloud layer (SCL), δh_{SCL}. The two latter quantities are defined as

$$\delta h_{\mathrm{base}} = \left[c_p(T_{\mathrm{u}} - \overline{T}) + L_{\mathrm{v}}(q^{\mathrm{u}} + q_l^{\mathrm{u}} - \overline{q})\right]_{\mathrm{base}} \tag{3.76}$$

$$\delta h_{\mathrm{SCL}} = \int_{SCL} \left(c_p\frac{\partial\overline{T}}{\partial t} + L_{\mathrm{v}}\frac{\partial\overline{q}}{\partial t}\right)\frac{dp}{g} \tag{3.77}$$

Eventually, the cloud-base mass flux for shallow convection is given by:

$$M_{\mathrm{u}}^{\mathrm{base}} = \frac{\delta h_{\mathrm{SCL}}}{\delta h_{\mathrm{base}}} \tag{3.78}$$

3.9.6 Flow chart

The computations performed in the simplified cumulus convection scheme are performed in the subroutines shown in Fig. 3.3. LECUMFS and LECUMFS2 are the two main switches to activate this simplified convection scheme in non-linear and linearized (tangent-linear and adjoint) integrations, respectively (namelist NAMCUMFS).

CUCALLN2: Provides interface of routines for cumulus parametrization. It is called from CALLPAR and returns updated tendencies of T, q, u, v and chemical tracers, as well as convective precipitation rates.

CUMASTRN2: Master routine for convection scheme. Also performs the convective closure and computes the momentum transport by convective draughts.

CUININ2: Initializes variables for convection scheme (including vertical interpolation to half model levels).

CUBASEN2: Triggering of convective updraught. Calculates condensation level and sets updraught base variables and first-guess cloud type.

CUPDRA: Computes preliminary updraught ascent to evaluate convective cloud top and base heights.

CUASCN2: Calculates actual ascent in updraught.

CUDDRAFN2: Calculates the downdraught descent.

CUFLX2: Calculates final convective fluxes and surface precipitation rates taking into account of melting/freezing and the evaporation of falling precipitation.

CUDTDQN2: Calculates the tendencies of T and q from convection.

CUDUDV2: Calculates the tendencies of u and v from convection.

CUCTRACER2: Calculates the tendencies of tracer fields due to the transport by convection.
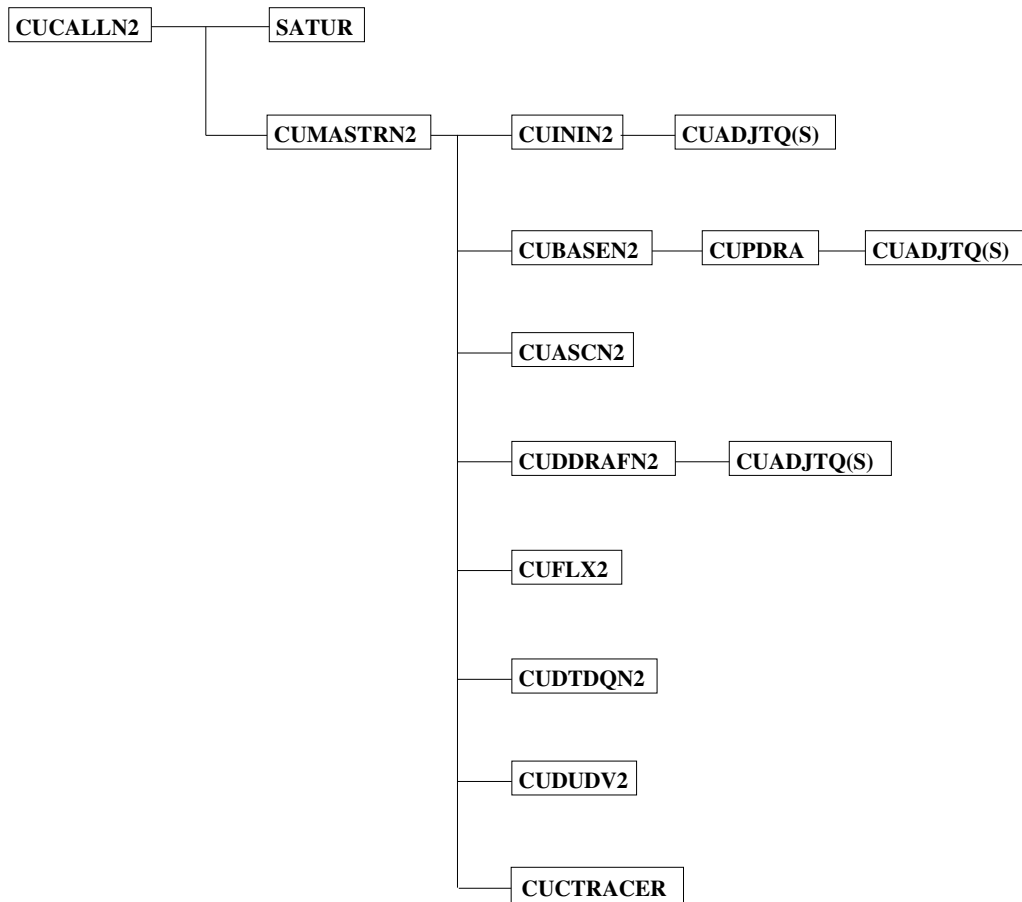
EXTERNALS

Figure 3.3 *Structure of the simplified convection scheme for data assimilation.*

SATUR: Computes saturation specific humidity.

CUADJTQS: Calculates super/sub-saturation and adjusts *T* and *q* accordingly.

All subroutines listed above have tangent-linear and adjoint versions with extension TL and AD, respectively. Of course, in the adjoint code, the calling order of the subroutines shown in Fig. 3.3 is reversed.

SET-UP OF PARAMETERS

Defined in subroutine SUCUM2 called from SUPHEC.

3.9.7 Regularization

Special care had to be taken to avoid the spurious growth of some perturbations in the tangent-linear and adjoint versions of the simplified convection code. In most cases, problematic perturbations are artificially reduced or even set to zero, as described below for each affected routine. Switch LREGCV should be set to .TRUE. in namelist NAMCUMFS to activate these regularizations in the linearized code. In the following, the prime symbol is employed to denote perturbations.

CUMASTRN2TL/AD:

- $(\delta h_{\text{base}})'$ is multiplied by 0.1 in computation of $M_{\text{u}}^{\text{base}}$ for shallow convection.

- $(M_{\text{u}}^{\text{base}})'$ is scaled by 0.2 in test whether $\delta h_{\text{SCL}}/(\delta h_{\text{base}})^2 > 10^3$.

- $(M_{\text{u}}^{\text{base}})'$ is scaled by 0.25 for both deep and shallow convection.

CUBASEN2TL/AD:

- $w'_* = 0$ if $w_* < 0.5$ m s^{-1} (convective-scale velocity).

- Buoyancy perturbation is scaled by 0.35 in updraught initialization.

CUASCN2TL/AD:

- $(\epsilon_{\mathrm{u}}^{\mathrm{org}})'$ is scaled by 0.1.

- $\left(\frac{w_{\mathrm{u}}^2(k)}{w_{\mathrm{u}}^2(k+1)}\right)'$ is set to 0 in organized detrainment computations.

- w'_{u} is set to $\begin{cases} 0 & \text{if } w_{\mathrm{u}} < 0.1 \text{ m s}^{-1} \\ \\ 0.1 \times w'_{\mathrm{u}} & \text{if } w_{\mathrm{u}} \geq 0.1 \text{ m s}^{-1} \end{cases}$

- $\left(\frac{q_{\mathrm{sat}}}{q_{\mathrm{sat}}^{\mathrm{base}}}\right)'$ is set to 0 in turbulent entrainment computations.

- Buoyancy perturbation is scaled by 0.33 in updraught computations.

CUPDRATL/AD:

- Buoyancy perturbation is scaled by 0.33 in updraught computations.

SUCUMF2:

- Mass-flux limiter, RMFCFL2, is set to 1 to avoid instabilities in implicit solver.

3.10 TRAJECTORY MANAGEMENT

The ECMWF physics uses the tendencies from the dynamics and variables at $t - \Delta t$ as input to compute the tendencies of a given process (represented by the operator P) for a prognostic variable ψ. Therefore

$$\frac{\psi^{n+1} - \psi_{\mathrm{u}}^{n-1}}{2\Delta t} = \mathrm{P}(\psi_{\mathrm{u}}^{n-1}) \tag{3.79}$$

where the variable ψ_{u} has already been updated by the dynamics and by the previous physical processes (which are called in the following order: radiation; vertical diffusion; subgrid-scale orographic effects; moist convection; large-scale condensation).

Thus

$$\psi_{\mathrm{u}}^{n-1} = \psi^{n-1} + \left(\frac{\partial \psi}{\partial t}\right)_{\mathrm{dyn}} + \left(\frac{\partial \psi}{\partial t}\right)_{\mathrm{phys}} \tag{3.80}$$

In (3.79), if the operator P is non-linear, its linearization around the basic state ψ_{u}^{n-1}, will require to store the model state at time step $n - 1$ (trajectory at $t - \Delta t$) as well as the tendencies produced by the dynamics $(\partial \psi / \partial t)_{\mathrm{dyn}}$. The physical tendencies from the previous processes $(\partial \psi / \partial t)_{\mathrm{phys}}$, require an additional call to the non-linear routines in the adjoint computations (CALLPARAD) and a local storage of the partial tendencies.

The storage of the trajectory at $t - \Delta t$ is performed in EC_PHYS by the routine STORE_TRAJ_PHYS called before the driver of the ECMWF physics CALLPAR. Fields are stored in grid-point space in an array TRAJ_PHYS. This array is allocated in the module TRAJ_PHYSICS, where also the number of the fields to be stored is defined.

The following three-dimensional fields are stored.

(i) **For the atmosphere:** the prognostic variables (wind components, temperature, specific humidity) and their tendencies produced by adiabatic processes, the vertical velocity, the long-wave fluxes and the solar transmissivity.

(ii) **For the soil:** the prognostic variables for temperature and moisture content (used to compute the surface fluxes from the trajectory in the linear vertical-diffusion scheme).

(iii) **For the tiles (i.e. vegetation (surface cover) types):** u- and v-stress, surface sensible heat flux, surface evaporation and skin temperature.

A number of two-dimensional fields used at time step $t - \Delta t$ need to be stored: surface pressure, surface fluxes, skin temperature, skin reservoir, snow reservoir, roughness lengths (mainly for the vertical diffusion).

The preliminary computations (pressure and geopotential at full and half model levels, astronomy parameters) are performed in EC_PHYS_TL and EC_PHYS_AD before calling the driver of the tangent-linear physics CALLPARTL or the driver of the adjoint physics CALLPARAD, and after reading the trajectory fields from GET_TRAJ_PHYS.

Chapter 4

Background terms

Table of contents

4.1   INTRODUCTION

The background term described in Courtier *et al.* (1998) was replaced in May 1997 by a new formulation by Bouttier *et al.* (1997), and replaced again in April 2005 by a wavelet-based covariance model (Fisher (2004), Fisher (2003)). The two older formulations are still part of the IFS but will not be described in this documentation. The model error background term used in weak-constraint 4D-Var is described in the last section.

4.2   DESCRIPTION OF THE ALGORITHM

We use the following notation.

(i)   **B** is the assumed background error covariance matrix.

(ii) $\delta\mathbf{x} = (\delta\zeta, \delta\eta, \delta(T, p_\mathrm{surf}), \delta q, \delta o3, \ldots)^\mathrm{T}$ is the low-resolution analysis increment (i.e. model field departures from the background) of vorticity, divergence, temperature and surface pressure, specific humidity, ozone mass mixing ratio and . . ., on model levels.

(iii)   A tilde $\tilde{\ }$ denotes a univariate transform of the increments with more Gaussian error statistics than the original increments.

(iv)   A subscript $\cdot_\mathrm{u}$ denotes a multivariate transform of the increments with less correlated error statistics than the original increments. Dynamic and other balance relationships are included here.

The incremental variational analysis problem, (1.2) of Chapter 1, is rewritten in the space defined by the change of variable $\delta\mathbf{x} = \mathbf{L}\chi$ (Section 1.4) where **L** satisfies $\mathbf{L}\mathbf{L}^\mathrm{T} = \mathbf{B}$ so that J_b takes a simple form. In operational practice, the initial point of the minimization is either the background (in which case $\delta\mathbf{x} = \chi = 0$) or the control vector saved from an earlier minimization. The minimization is carried out in the space of χ. At the end of the minimization, the analysis increments are reconstructed in model space by $\delta\mathbf{x} = \mathbf{L}\chi$. In order to compare with observations, **x** is reconstructed using (2.4), in each simulation. Thus the variational analysis can be performed using only transformations from minimization space to model space (CHAVARIN). The transformation from model space to minimization space is never

required. In particular, **L** is not required to be invertible, and may even be rectangular. This is the case in the "wavelet" J_b formulation described here, where the dimension of the control vector is considerably larger than the dimension of the model state vector.

The background-error covariance matrix **B** is implied by the design of **L**, which currently has the form

$$\mathbf{L} = \mathbf{K}\mathbf{L}_u \tag{4.1}$$

where **K** is a balance and variable transform operator going from the set of variables $\delta\widetilde{\mathbf{x}}_u$ to the model variables $\delta\mathbf{x}$.

The $\mathbf{L}_u$ operator defines the covariance matrix for $\delta\widetilde{\mathbf{x}}_u$ as

$$\mathbf{B}_u = \mathbf{L}_u\mathbf{L}_u^T \tag{4.2}$$

So far, the formulation is quite general. Now, we restrict $\mathbf{L}_u$ to a simple form and choose a particular balance and variable transform operator **K**.

The covariance matrix $\mathbf{B}_u$ is assumed to be block-diagonal, with no correlation between the parameters, so that

$$\mathbf{B}_u = \begin{bmatrix} \mathbf{C}_\zeta & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{C}_{\eta_u} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{C}_{(T,p_{\text{surf}})_u} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_{q_u} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{C}_{o3} & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots \end{bmatrix} \tag{4.3}$$

The matrix $\mathbf{L}_u$ is similarly block-diagonal, with diagonal blocks $\mathbf{L}_\zeta$, $\mathbf{L}_{\eta_u}$, $\mathbf{L}_{(T,p_{\text{surf}})_u}$, $\mathbf{L}_{q_u}$, $\mathbf{L}_{o3}$, (etc.). However, these sub-matrices are rectangular. Each sub-matrix is treated identically in the code, except for differences in the coefficients used to describe the covariances, so we will consider just one sub-matrix, $\mathbf{L}_\zeta$.

4.2.1 The wavelet correlation matrix and gridpoint variances

The "wavelet" J_b formulation was devised to allow both spatial and spectral variation of the horizontal and vertical covariances of background error. Only a brief description is given here. The reader is referred to Fisher (2004) for a mathematical justification of the method, and also to Fisher (2003).

Simultaneous spatial and spectral variation of horizontal and vertical covariances is achieved by dividing the control vector up into several parts, each of which corresponds to a band of total spherical wavenumbers, n. For each band, the elements of the control vector are arranged on a linear reduced Gaussian grid that is appropriate to the spectral truncation corresponding to the highest wavenumber in the band. The wavenumber bands overlap, with the result that exactly two bands use linear grids corresponding to the full model resolution. The cutoff wavenumber decreases by approximately a factor of $\sqrt{2}$ for each subsequent band and the number of gridpoints reduces by a factor of two, so that the total number of elements of the control vector is approximately three times larger than a gridpoint representation of the model fields.

Information about the vertical and horizontal correlations is stored in the form of sets of vertical covariance matrices. There is one set of matrices for each wavenumber band, and for each band the symmetric square-roots of these matrices are stored on a horizontal grid that is somewhat coarser than the grid used for the band's control vector elements.

The transformation of the control vector to model space, represented by $\mathbf{L}_\zeta$, consists of the following steps. First, for each wavenumber band, each vertical column of the grid is multiplied by the square-root of the covariance matrix (from the set corresponding to the wavenumber band) that is nearest to the grid column. Next, the control vector elements are transformed to spectral space. Since the wavenumber bands overlap, there are now more than one (in fact, exactly two) sets of spectral coefficients for each spherical wavenumber, n. These sets of coefficients are combined in a weighted sum to give a conventional spectral description of the model fields. For each wavenumber, the sums of squares of the

weights is equal to one. In fact, the weights are the square-roots of triangle functions that decay to zero towards the boundaries of each wavenumber band, and take the value one at the centre of the band. As a result, the vertical covariance associated with each wavenumber is effectively linearly interpolated between the covariance matrices defined for each band, and the horizontal covariance is defined by a piecewise-linear function of n.

The final step of the transformation L_ζ is to transform the spectral fields to the model grid, and to multiply by the assumed standard deviations of background error.

4.2.2 The multivariate error correlations between variables: 'balance'

The operator **K** first applies a multivariate 'balance' operator to obtain variables with less correlated errors, and then a univariate variable transform is applied to obtain variables with more Gaussian error statistics. The multivariate part of **K** currently accounts for dynamic balance in spectral space (nonlinear balance and quasi-geostrophic omega equation) and thermodynamic balance between humidity and temperature in gridpoint space (moist-adiabatic relationship in cloud covered fraction of gridbox). In general, any multivariate error correlations will be included here.

(*a*) *Dynamic balance in spectral space*

After the control variable has been multiplied by the background error standard deviations, the contribution due to dynamic balances is added back in spectral space, currently defined by the following transformations for temperature, surface pressure and divergence only:

$$
\begin{aligned}
\delta(T, p_{\text{surf}}) &= \mathbf{N}\delta\zeta + \mathbf{P}\delta\eta_{\text{u}} + \delta(T, p_{\text{surf}})_{\text{u}} \\
\delta\eta &= (\mathbf{M} + \mathbf{Q}_2)\delta\zeta + \delta\eta_{\text{u}} + \mathbf{Q}_1\delta(T, p_{\text{surf}})
\end{aligned}
\tag{4.4}
$$

The matrix blocks **M**, **N**, **P**, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are in general not invertible, however the balance operator **K** is. The **M**, **N** and **P** operators used to define balance have a restricted algebraic structure. **M** and **N** are both the product of a so-called horizontal balance operator H and vertical balance operators M, N such that

$$
\begin{aligned}
\mathbf{M} &= MH \\
\mathbf{N} &= NH
\end{aligned}
\tag{4.5}
$$

The H operator is a block–diagonal matrix of identical horizontal operators transforming the spectral coefficients of vorticity, independently at each level, into an intermediate variable P_{b} which is a kind of linearized mass variable. The horizontal operators in H are defined analytically as linearized versions of the non-linear balance equation:

$$
\nabla^2 P_{\text{b}} = (f + \zeta) \times v_\psi + \frac{1}{2}\nabla(v_\psi \cdot v_\psi)
\tag{4.6}
$$

where $v_\psi = \mathbf{k} \times \nabla\psi$ is the rotational wind.

This equation is simplified, by treating model levels as if they were pressure levels, and is linearised about the background state to provide a linear equation for increments in P_{b} as a function of vorticity increments.

The matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are also defined analytically, as simplified and linearised versions of the quasi-geostrophic omega equation:

$$
\left(\sigma\nabla^2 + f_0^2\frac{\partial^2}{\partial p^2}\right)\omega' = -2\nabla \cdot \mathbf{Q}
\tag{4.7}
$$

(Here, **Q** is the Hoskins' Q-vector: a function of temperature and rotational wind.)

Once again, model levels are treated as if they were pressure levels, and the stability parameter, σ is assumed to be a function of pressure. This allows the equation to be separated into a set of small tri-diagonal systems that can be solved non-iteratively to give divergence increments as a linear function of vorticity and temperature increments.

The M, N and $\mathbf{P}$ operators all have the same structure: block-diagonal, with one full vertical matrix per spectral component. The vertical matrices depend only on the total wavenumber n, and are calculated by linear regression. (For example, in the case of the operator N, the regression is between $\delta\mathbf{P}$ and $\delta(T, p_{\mathrm{surf}})$.)

Calibration of the statistically-derived operators of the balance operator is performed outside the IFS using programs developed by Francois Bouttier.

(b) Thermodynamic balance in gridpoint space

The contribution to humidity coming from the thermodynamic relation with temperature is now added back in gridpoint space,

$$\delta\widetilde{q} = \delta\widetilde{q}_{\mathrm{u}} + Q_{\mathrm{qT}}\frac{q^{\mathrm{b}}}{q_{\mathrm{s}}(T^{\mathrm{b}})}\frac{L}{R_{\mathrm{v}}(T^{\mathrm{b}})^2}\delta T \tag{4.8}$$

where $\delta\widetilde{q} = \delta q/q_{\mathrm{s}}(T^{\mathrm{b}})$ (see below), $q_{\mathrm{s}}(T^{\mathrm{b}})$ is the saturation specific humidity at the background temperature T^{b}, L is the latent heat for mixed phase and R_{v} is the gas constant for water vapour. This relationship between humidity and temperature changes is derived from the Clausius-Clapeyron equation which gives the heat generated by a given phase-change of water vapour in clouds. The coefficient Q_{qT} is determined by a statistical regression as function of the background relative humidity rh^{b} and model level. Q_{qT} starts from zero in cloud free grid boxes, estimated as rh^{b} below about 80% relative humidity, and goes to unity as rh^{b} approaches 1. The regression coefficient Q_{qT} does resemble a simplified statistical model of cloud cover. The humidity-temperature balance is put to zero in supersaturated (wrt mixed phase) areas and in the stratosphere (determined by a tropopause diagnosed from the background fields).

The statistical regression for obtaining Q_{qT} is performed outside the IFS using programs developed by Elías Hólm.

4.2.3 The univariate variable transform: 'gaussianization'

The last step is to transform any variables that have been 'gaussianized' back to the model variables. Currently only humidity has been transformed to another variable to obtain more Gaussian error statistics,

$$\delta q = q_{\mathrm{s}}(T^{\mathrm{b}})\delta\widetilde{q} \tag{4.9}$$

For the other variables $\delta T = \delta\widetilde{T}$, etc. It is important to note that this is a linear transform in the inner loops, whereas a non-linear transform may really be what is required to account for the non-Gaussianity at this stage. The non-linear aspect of the transform is dealt with at the outer loop level in the variational framework, see Hólm *et al.* (2002) for a discussion.

4.3 IMPLEMENTATION

4.3.1 The J_{b} information structure: SPJB_VARS_INFO

The J_{b} code is required to be sufficiently flexible to allow the incorporation of new variables, and to allow the choice of analysed variables to differ from the choice of model variables. This is achieved through a separation of the arrays used by J_{b} from the model arrays, and is controlled by a "structure" (Fortran derived type) containing information about the J_{b} variables: SPJB_VARS_INFO. This structure is ubiquitous throughout the J_{b} code.

SPJB_VARS_INFO is defined in YOMJG, and is a one-dimensional array with one element for each analysed variable. Analysed variables may be either three-dimensional (for example, vorticity) or two-dimensional (e.g. surface pressure).

Each element of SPJB_VARS_INFO is a structure with the following elements:

- IGRIBCODE
- IGRIBCODE_FCE

- IPT
- IPTJB
- IPTFCE
- IPTBG
- L_IN_SPA3
- L_IN_SPA2
- L_IN_SPGFL
- L_IN_GPGFL
- COR_STRING

The first two elements give the GRIB parameter codes used to identify the variable and the corresponding background error fields. These may be different since, for fields involved in the balance operator, IGRIBCODE_FCE refers to the unbalanced component of the field, whereas IGRIBCODE refers to the full (balanced plus unbalanced) field.

The elements IPT, IPTJB, IPTFCE and IPTBG are indexes into various arrays. IPT indicates which field in the model's SPA3, SPA2, SPGFL or GFL arrays corresponds to the analysed variable. IPTJB provides an index into the (spectral) J_b arrays, SPA3JB and SPA2JB. IPTFCE locates the variable in the forecast-error array FCEBUF, and IPTBG locates the corresponding background fields in SP7A3, GP7A3 or SP7A2.

Four logical flags, L_IN_SPA3, L_IN_SPA2, L_IN_SPGFL and L_IN_GPGFL, show where in the model's various arrays the variable is stored. These flags are used in conjunction with the element IPT to locate the model's equivalent of the analysed variable.

The final element of SPJB_VARS_INFO is a character string. This is used to match the variable with the covariance matrices in the input file "wavelet.cv".

4.3.2   User input

The initial setup of J_b is performed by SUJB. The user can input, via NAMJG, the number of analysed variables (N_SPJB_VARS). GRIB parameter codes for these variables (M_GRIBCODES) and for the corresponding background error fields (M_GRIBCODES_FCE), and also the character strings that identify the corresponding covariance matrices (C_COR_STRINGS), are input via NAMJBCODES.

These inputs are sufficient for SUJB to construct the information structure, SPJB_VARS_INFO. The correspondence between analysis fields and model fields is derived by looking up the grib code of the analysed variable in the model's GMV and GFL structures. If none of the inputs is specified, the analysis defaults to 6 variables: vorticity, divergence, temperature, surface log-pressure, humidity and ozone.

Background error statistics for wavelet J_b may be calculated by setting LJBWSTATS to TRUE. In this case, the code expects to read a large set of background states as spectral GRIB fields. The states are expected to be produced by an analysis ensemble with N_BGMEMBERS members and covering N_BGDATES dates. Inter-member differences are calculated for each date, and the statistics of background error are calculated from this set of differences. (Statistics may be calculated using the "NMC method" by regarding, for example, a set of 48-hour forecasts as member 1, and a set of 24-hour forecasts as member 2.) The filename for each state is constructed as a six-character prefix (CINBGSTATES) followed by a string of the form "mXXX_dYYY", where XXX is in the range 1...N_BGMEMBERS, and YYY is in the range 1...N_BGDATES.

Calculation of the statistics is performed in SUJBWAVGEN and associated subroutines. At the end of the calculation, a file called "wavelet.cv" is written. Note that calculation of the J_b statistics is largely single-threaded.

4.3.3   Input files

The IFS requires three input files to define J_b. The covariance matrices that define the vertical and horizontal correlations are read from "wavelet.cv". This file also contains power spectrum information which is used to determine global mean vertical profiles of standard deviations of background

error. Coefficients for the balance operator are read from "stabal96.bal", and standard deviations of background error are read from "errgrib".

4.3.4 Initial setup

SUJBWAVELET0 determines the number of wavenumber bands and their wavenumber boundaries by reading header information from the statistics file "wavelet.cv". Next, the spherical transforms and grid layouts corresponding to the cutoff wavenumbers for each band are defined by a call to SUJBWAVTRANS. This information is stored in the structure GRID_DEFINITION (YOMWAVELET).

Some allocation of arrays is carried out in SUALGES, and the control vector is allocated in SUALCTV. Coefficients for the humidity variable and for the balance operators are initialised by SUJBCHVAR and SUJBBAL, respectively.

Most of the setup for "wavelet J_b" is performed by SUJBWAVELET and SUJBWAVALLO. The main tasks are to read the input file, which contains the symmetric square-roots of the vertical covariance matrices. To reduce the size of the input file, each matrix has an associated "skyline", which identifies the first and last element in each row (or column, since the matrices are symmetric) that is significantly different from zero. Only the matrix elements between these limits are stored in the file.

In addition to the covariance matrices, the input file also contains power-spectrum information about the analysed variables. This is used to construct global-mean profiles of background error standard deviation.

The covariance matrices are stored in the structure WAVELET_VCORS, together with information about their grid layout.

The final part of the J_b setup is to read the background fields (SUECGES), and to initialise the standard deviations of background error (SUINFCE).

4.3.5 Change of variable

The change of variable, CHAVARIN, implicitly defines the background error covariance model. It consists of three steps: CVAR2IN implements multiplication of control vector by the matrix denoted **L** in Section 4.2. The transformed variables are inserted into the model's arrays by JBTOMODEL. This may involve spectral transforms, since the J_b variables are all spectral, whereas model variables may be held as gridpoint fields. Finally, the background state is added to the model fields by ADDBGS.

The control vector for the minimization contains several components besides those related to J_b. Within CVAR2IN all these components are handled, and the J_b components are dealt with by a call to CVAR3IN. In CVAR3IN a call to SQRTBIN applies the vertical and horizontal correlations (JGCORI) and multiplies by the standard deviations of background error (CVARGPTL). The next step in CVAR3IN is to apply the dynamic balance operator via calls to BALNONLINTL, BALVERT and BALOMEGATL. The final step in CVAR3IN is a call to JBCHVARI which includes the humidity related thermodynamic balance and 'gaussianization' change of variable.

Note that CVARGPTL calls TRANSINV_WAVELET and TRANSDIR_WAVELET to perform the spectral transforms required to convert variables between spectral and gridpoint representations. Here these routines implement *spectral* transforms using the information stored in GRID_DEFINITION. They do not implement wavelet transforms.

JGCORI calls JBVCOR_WAVELETIN once for each band of wavenumbers to perform the main part of the wavelet J_b change of variable. JBVCOR_WAVELETIN has three steps. First, each column of the control vector is multiplied by the square-root of a vertical covariance matrix. This is performed by JBVCORG. Next, the gridpoint fields are transformed to spectral space by a call to TRANSDIR_WAVELET. The coefficients are multiplied by the appropriate weights by WAVXFORM, and the weighted coefficients are added into the arrays PSPA3JB and PSPA2JB.

The calls to JBVCORG are the most expensive part of the wavelet J_b code. The calls are performed within an Open-MP loop over NPROMA blocks. For each grid column and each analysis variable, the nearest

covariance matrix is identified by a call to JBMATINTERP, and the column is multiplied by the square-root of the covariance matrix. A considerable saving of computational time is achieved by taking into account the fact that elements outside the matrix "skyline" (see above) are known to be zero.

4.4 MODEL ERROR BACKGROUND TERM

The weak-constraint 4D-Var formulation that has been implemented in the IFS introduces an additive model error forcing term to the model prognostic equations which is active in the data assimilation cycle only:

$$\mathbf{x}_k = \mathcal{M}_{k,k-1}(\mathbf{x}_{k-1}) + \eta \quad \text{for} \quad k = 1, \dots, N. \tag{4.10}$$

In the current implementation (Laloyaux *et al.* (2020)), the model error forcing is assumed to be additive and constant within the 12-hour assimilation window. The model error forcing is applied to temperature, vorticity, divergence and logarithm of surface pressure. The weak-constraint 4D-Var cost function can be written as

$$J_{WC}(\mathbf{x}_0, \mathbf{j}) = \frac{1}{2} \left(\mathbf{x}_0 - \mathbf{x}_0^b\right)^{\mathrm{T}} \mathbf{B}^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^b\right) + \frac{1}{2} \sum_{k=0}^{N} \left(\mathcal{H}_k(\mathbf{x}_k) - \mathbf{y}_k\right)^{\mathrm{T}} \mathbf{R}_k^{-1} \left(\mathcal{H}_k(\mathbf{x}_k) - \mathbf{y}_k\right)$$

$$+ \frac{1}{2} \left(\mathbf{j} - \mathbf{j}^b\right)^{\mathrm{T}} \mathbf{Q}^{-1} \left(\mathbf{j} - \mathbf{j}^b\right) \tag{4.11}$$

where $\mathbf{j}^b$ is the prior estimate of the model forcing estimated from the previous assimilation cycle. This "forcing" formulation of weak-constraint 4D-Var estimates simultaneously the initial state $\mathbf{x}_0$ and model forcing $\mathbf{j}$ that best fit the observations and the background information given their respective error covariances matrices.

In September 2009, weak constraint 4D-Var was introduced for the first time in the ECMWF operational system with its implementation in the IFS cycle 35R3. It was deactivated in June 2013 in cycle 38R2 when the vertical resolution was upgraded from 91 to 137 levels and reintroduced in July 2017 in cycle 43R3. The model error covariance matrix $\mathbf{Q}$ was estimated using an ensemble of forecasts perturbed by the SPPT scheme and was only activated above 40 hPa to prevent unacceptable degradation of the analysis quality and forecast skill. A new formulation of the model error covariance matrix was computed from the timeseries of the model error estimated by the 43R3 weak-constraint formulation based on the SPPT formulation. This new sample model error covariance matrix is given by

$$\mathbf{Q}_\eta = \frac{1}{N-1} \sum_{i=1}^{N} (\eta_i - \bar{\eta})(\eta_i - \bar{\eta})^T$$

where η_i is the model error estimated over a 12-hour assimilation window and $\bar{\eta}$ is the average model error estimated over the whole assimilation period. The weak-constraint 4D-Var with this model error covariance is active above 125 hPa in the HRES system starting with IFS cycle 47R1 and has been extended to the EDA starting with IFS cycle 47R3.

Chapter 5

Observation-related processing (J_o)

Table of contents

5.1 INTRODUCTION

The observation operators provide the link between the analysis variables and the observations (Lorenc, 1986; Pailleux, 1990). The observation operator is applied to components of the model state to obtain the model equivalent of the observation, so that the model and observation can be compared. The operator H in (1.4) signifies the ensemble of operators transforming the control variable $\mathbf{x}$ into the equivalent of each observed quantity, $\mathbf{y}^o$, at observation locations. The 3D/4D-Var implementation allows H to be (weakly) non-linear, which is an advantage for the use of satellite radiance data, scatterometer data with aliased wind direction, cloud and precipitation data, for example. In this chapter we define the aspects of observation processing that are integrated into the data assimilation method, particularly the calculation of departures and the J_o costfunction, along with VarQC, VarBC and the modelling of correlated observation error. See part 1 for more information on specific observing systems, observation operators, screening, thinning, blacklisting and other tasks that are not tied to a particular assimilation algorithm.

5.2 HORIZONTAL INTERPOLATION TO OBSERVATION POINTS

5.2.1 Overview

Currently it is assumed that each observation equivalent can be computed from a single vertical profile of model data. That is, it is assumed that each observation operator can be written as $H = H_v H_h(\mathbf{x})$ where H_h is horizontal interpolation of model data to the location of the observation. H_h is performed for all observations within a time-slot directly after the corresponding model time step has been performed. The output of H_h is a vertical profile of model data for each observation location. These profiles are stored in a data structure called the GOM-arrays, and used later in the vertical part of the observation operator H_v performed in HOP, called from OBSV. So-called two-dimensional GOM-arrays have been

developed, to permit horizontal integration, initially for limb-sounding and radio-occultation data. However, now they are also used for slant-path radiative transfer for nadir satellite observations. The orientation and locations of the two-dimensional GOM-arrays (e.g. one vertical plane per observation) is then determined by the satellite viewing geometry. The H_v operator for those data receive the vertical plane of model data as input, or in the case of slant path radiative transfer, just the slant path itself.

Interpolation is only one of the issues involved in the GOM arrays: as important is the mapping from one parallel representation to another. For the model fields, the globe is broken into contiguous regions that can be handled by separate MPI tasks. In contrast, observations are scattered among MPI tasks so that the observation processing can be load-balanced. A set of tables created by MKGLOBSTAB_MODEL and MKGLOBSTAB_OB describes the mapping between the tasks where the model fields are located and the tasks where the observations are stored. The main functions of the GOMs are:

(i) To assemble model fields including the haloes (e.g. model points owned by different MPI tasks) and carry out the horizontal interpolation to observation locations.

(ii) To do the message-passing that sends the interpolated model fields to the right tasks for the observations. The reverse of this process is done in the adjoint case, with special attention paid to keeping the results bit-reproducible. Message passing can take a lot of computer time, so it has to be carefully optimised, and the number of models fields included in the interpolation needs to be limited.

5.2.2 Implementation

For OOPS, all of the computations around the GOM arrays have been encapsulated in objects of CLASS_SUPERGOM. However, most of the GOM code is found in the lower-level module GOM_MOD. Only its public interface should be of concern to the rest of the IFS; it is designed so that its internal workings are private. In the minimisation there are two instances of the GOM object in use: YDGOM5 for the direct fields and YDGOM for the TL and adjoint fields. GOM variables have their own naming convention, but it is supposed to be as close as possible to the naming in the model itself, e.g. GFL and GMV arrays. GOM variables are indexed using a derived type called GID, defined in gom_mod. For example:

- GID%T = temperature
- GID%U = u-wind
- GID%L = cloud liquid water

To add a new GOM variable, follow the checklist in the header of GOM_MOD or see the GOM user guide, RD internal memorandum RD13-199.

Configuration is done in SUGOMS. In the minimisation, this is called twice, once for the settings of the direct model fields (i.e. GOM5) and once for the TL/adjoint fields (i.e. GOM). Inside this routine, the logical LDOBSTL indicates whether configuration is being done for the direct or the TL/adjoint.

There are two switches, each dimensioned by observation type and GOM variable ID. The first switch activates GOM interpolation. For example, to make the vertical profile of temperature available in the observation operator for clear-sky satellite observations:

GOM%LINTERP(NSATEM,GID%T) = .true.

Note that NSATEM=7 is the satellite obstype. The second switch determines how the horizontal interpolation should be done, e.g:

GOM%INTERP_TYPE(NSATEM,GID%T) = GINTERP%BILINEAR_4POINT

The main interpolation options are:

- ginterp%default - One of the following two, according to the value of the IFS configuration variable NOBSHOR:
- ginterp%bilinear_4point - Bilinear 4 point interpolation, as provided by laidliobs. Default when NOBSHOR=201.

- ginterp%bidimensional_12point - Bidimensional twelve point interpolation as provided by laiddiobs. Default when NOBSHOR=203.
- ginterp%nearest_neighbour - Nearest neighbour interpolation, as provided by laidlic

The default is currently driven by the setting of NOBSHOR: NOBSHOR = 203 (bi-cubic) for the trajectory jobs, but NOBSHOR = 201 (bi-linear) in the minimisation jobs. There are also a number of specific Meteo France interpolation options that take special account of the surface type, e,g, the presence of ocean or land in the model grid points used in the horizontal interpolation. Much of the interpolation code is shared with the semi-Lagrangian advection scheme of the dynamics.

In the model code, SCAN2M contains the entry point for the horizontal interpolation of model data to observation points. CLASS_SUPERGOM%MODEL_IN takes the model data. It requires model fields from both the beginning and end of the timestep; the majority from before the timestep and afterwards, a few derived quantities that are only available from the gridpoint model. An example is the vertical profile of precipitation fraction, which is used in the assimilation of microwave observation in all-sky conditions.

In the observation operator HOP receives the interpolated model fields in the form of a GOM_PLUS derived type. they have been unpacked from the SUPERGOM at the level of TASKOB_THREAD by a call to CLASS_SUPERGOM%MODEL_IN. The GOM_PLUS contains a number of derived model variables that are not supplied to MODEL_IN. For example, instead of wasting computer time interpolating the pressure profile horizontally, the GOM_PLUS recreates the pressure profile from the horizontally interpolated surface pressure and the a and b coefficients. This principle applies to other variables that can be derived from the core atmospheric state.

5.2.3 Diagnostics

The r.m.s. of the GOM arrays is printed if the switch LPRTGOM = .TRUE., (in YOMOBS). The default is that the print is switched on. It can be located in the log file by searching for 'Norm of interpolation operator (GOM)'. The printing is done in the minimisation (i) when the GOM arrays contain the background interpolated to the observation points, (ii) when it contains ∇J_o of the first simulation, (iii) when it contains first TL perturbations after the initial call to the minimizer and (iv) when it contains ∇J_o at the final simulation.

5.3 COST FUNCTION

5.3.1 Construction

The master routine controlling the calls to the individual observation operators is called HOP. This routine deals with all different types of observations - see part 1 for more information. After HOP, DEPARTURE_JO computes the departures and the costfunction. It calls HDEPART, to calculate the departure $\mathbf{z}$ as

$$\mathbf{z} = \mathbf{y}^o - H\mathbf{x} + (\mathbf{y}^o - H\mathbf{x}_{HR}^b) - (\mathbf{y}^o - H\mathbf{x}_{LR}^b) \tag{5.1}$$

where the two terms in brackets have been computed previously: the first one in the high resolution trajectory run and the second one in the LOBSREF call, described in Section 2.2.

If LOBSTL then $\mathbf{z}$ is

$$\mathbf{z} = \mathbf{y}^o - \mathbf{H}\delta\mathbf{x} + (\mathbf{y}^o - H\mathbf{x}_{HR}^b) - \mathbf{y}^o \tag{5.2}$$

which simplifies to what has been presented in Section 1.4.

Bias correction is also carried out at this point by subtracting the bias estimate (kept in the NCMTORB-word of ODB) from the calculated departure.

Finally the departure is divided by the observation error σ_o (NCMFOE in ODB) to form the *normalized departure*.

Departures of correlated data are multiplied by $\mathbf{R}^{-1}$, see Subsection 5.4. The division by σ_o has already taken place in HDEPART, so $\mathbf{R}$ at this point is in fact a correlation (not a covariance) matrix.

The cost function is computed in HJO, as

$$J_{\mathrm{o}} = \mathbf{z}^{\mathrm{T}}\mathbf{z} \tag{5.3}$$

for all data, even for SCAT data when LQSCAT = .TRUE., as in current 4D-Var with quadratic cost-function. When LQSCAT = .FALSE. (as in current 3D-Var configuration) the SCAT cost function combines the two ambiguous winds (subscripts 1 and 2) in the following way (also in HJO),

$$J_{\mathrm{SCAT}} = \left[\frac{J_1^4 J_2^4}{J_1^4 + J_2^4} \right]^{1/4} \tag{5.4}$$

These expressions for the cost function are modified by variational quality control, see Section 5.5. The cost-function values are store in two tables, as detailed in Subsection 5.3.2.

HJO, also stores the resulting *effective departure* in the NCMIOM0-word of 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, $\mathbf{z}_{\mathrm{eff}} = \mathbf{z}^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{QC}_{\mathrm{weight}}]$, where the QC-weight will be defined below, Section 5.5 .

In the adjoint routine DEPARTURE_JOAD the input to the adjoint observation operators (the effective departure) is read from the ODB. The expression for the gradient (with respect to the observed quantity) is then simply

$$\nabla_{\mathrm{obs}}J_{\mathrm{o}} = -2\mathbf{z}_{\mathrm{eff}}/\sigma_{\mathrm{o}} \tag{5.5}$$

which is calculated for all data. The adjoint code closely follows the structure of the direct code, with the adjoint operators applied in the reverse order.

5.3.2 J_{o} tables

There are two different tables for storing the J_{o} values. One is purely diagnostic (JOT, yomcosjo.h), and is used for producing the printed J_{o} tables in the log-file (PRTJO called rom EVCOST). JOT is a FORTRAN90 derived type with items containing cost function value, data count, observation error, background error and codetype, for each observation type and for each observed variable (as defined by NVAR above).

The actual J_{o}-table is called FJOS (yomcosjo). FJOS is indexed by the absolute observation number, iabnob = MABNOB(jobs, kset), so that the contributions from each individual observation can be summed up in a predetermined order (in EVCOST), to ensure reproducibility, irrespective of number of processors and tasks.

5.4 CORRELATION OF OBSERVATION ERROR

The observation error is assumed uncorrelated (i.e. the matrix $\mathbf{R}$ is diagonal) for all data. Optionally, by setting LTC=.true., time-sequences of SYNOP/DRIBU surface pressure and height data (Järvinen *et al.*, 1999) will be time correlated. For radiance observations, the option exists to take inter-channel error correlations into account, through the variable LIC_OBSERR_COR_RAD1C set in DEFRUN.

The serial correlation for SYNOP and DRIBU data is modelled by a continuous correlation function $a\mathrm{e}^{-b(t_1-t_2)^2}$ where a = RTCPART = 0.3 and b = RTCEFT = 6.0 hours, under the switch LTC (namjo). The remaining fraction $1 - a$ of the error variance is assumed uncorrelated (see COMTC).

When $\mathbf{R}$ is non-diagonal, the 'effective departure' $\mathbf{z}_{\mathrm{eff}}$ is calculated by solving the linear system of equations $\mathbf{z}_{\mathrm{eff}}\mathbf{R}$ for $\mathbf{z}_{\mathrm{eff}}$, using LAPACK routines SPOTRF (Choleski decomposition) and SPOTRS (backwards substitution), as is done in COMTC.

A similar approach is taken to account for inter-channel error correlations if LIC_OBSERR_COR_RAD1C is activated for a radiance datum. The relevant routine is INTERCHAN_OBSERR_COR, called in HJO. For inter-channel error correlations to be active, the correlation information has to be provided as part of an instrument-specific error file which is read in the routine DEFRUN.

5.5 VARIATIONAL QUALITY CONTROL

The variational quality control, VarQC, has been described by Andersson and Järvinen (1999). It is a quality control mechanism which is incorporated within the variational analysis itself. A modification of the observation cost function to take into account the non-Gaussian nature of gross errors, has the effect of reducing the analysis weight given to data with large departures from the current iterand (or preliminary analysis). Data are not irrevocably rejected, but can regain influence on the analysis during later iterations if supported by surrounding data. VarQC is a type of buddy check, in that it rejects those data that have not been fitted by the preliminary analysis (i.e. the current state vector), often because it conflicts with surrounding data.

5.5.1 Description of the method

The method is based on Bayesian formalism. First, an *a priori* estimate of the probability of gross error $P(G)_i$ is assigned to each datum, based on study of historical data. Then, at each iteration of the variational scheme, an *a posteriori* estimate of the probability of gross error $P(G)_f$ is calculated (Ingleby and Lorenc, 1993), given the current value of the iterand (the preliminary analysis). VarQC modifies the gradient (of the observation cost function with respect to the observed quantity) by the factor $1 - P(G)_f$ (the QC-weight),which means that data which are almost certainly wrong ($P(G)_f \approx 1$) are given near-zero weight in the analysis. Data with a $P(G)_f > 0.75$ are considered 'rejected' and are flagged accordingly, for the purpose of diagnostics and feedback statistics, etc.

The normal definition of a cost function is

$$J_o = - \ln p \tag{5.6}$$

where p is the probability density function. Instead of the normal assumption of Gaussian statistics, we assume that the error distribution can be modelled as a sum of two parts: one Gaussian, representing correct data and one flat distribution, representing data with gross errors. We write

$$p_i = N_i[1 - P(G_i)] + F_i P(G_i) \tag{5.7}$$

where subscript i refers to observation number i. N and F are the Gaussian and the flat distributions, respectively, given by

$$N_i = \frac{1}{\sqrt{2\pi}\sigma_o} \exp\left[-\frac{1}{2}\left(\frac{y_i - Hx}{\sigma_o}\right)^2\right] \tag{5.8}$$

$$F_i = \frac{1}{L_i} = \frac{1}{2l_i\sigma_o} \tag{5.9}$$

The flat distribution is defined over an interval L_i which in (5.9) has been written as a multiple of the observation error standard deviation σ_o. Substituting (5.7) to (5.9) into (5.6), we obtain after rearranging the terms, an expression for the QC-modified cost function J_o^{QC} and its gradient ∇J_o^{QC}, in terms of the normal cost function J_o^N

$$J_o^N = \frac{1}{2}\left(\frac{y_i - Hx}{\sigma_o}\right)^2 \tag{5.10}$$

$$J_o^{QC} = - \ln\left(\frac{\gamma_i + \exp[-J_o^N]}{\gamma_i + 1}\right) \tag{5.11}$$

$$\nabla J_o^{QC} = \nabla J_o^N\left(1 - \frac{\gamma_i}{\gamma_i + \exp[-J_o^N]}\right) \tag{5.12}$$

$$\nabla J_o^{QC} = \nabla J_o^N w_i \tag{5.13}$$

where

$$w_i = 1 - \frac{\gamma_i}{\gamma_i + \exp[-J_o^N]} \tag{5.14}$$

$$\gamma_i = \frac{P(G_i)/(2l_i)}{[1 - P(G_i)]/\sqrt{2\pi}} \tag{5.15}$$

5.5.2 Implementation

The *a priori* information, i.e. $P(G)_i$ and l_i, is set during the screening, in the routine HJO, and stored in the NCMFGC1 and NCMFGC2-words of the ODB. Default values are set in DEFRUN, and can be modified by the namelist namjo. VarQC can be switched on/off for each observation type and variable individually using LVARQC, or it can be switched off all together by setting the global switch LVARQCG = .FALSE.. Since an as good as possible 'preliminary analysis' is needed before VarQC starts, it is necessary to perform part of the minimization without VarQC, and then switch it on.

HJO computes $J_{\mathrm{o}}^{\mathrm{QC}}$ according to (5.11) and the QC-weight, w_i, according to (5.14).

The 3D-Var assimilation system (Chapter 3) can handle non-quadratic cost functions because it is using the M1QN3 minimization routine. This allows VarQC to be activated after NITERQC (40 by default) iterations during the minimization process. The 4D-Var assimilation system (Chapter 2) by default use a conjugate gradient minimization method that requires a strictly quadratic cost function. So in 4D-Var we use a quadratic formulation of VarQC, controlled by LQVARQC (default .TRUE.).

In the non-quadratic formulation the QC-weight, w_i, (5.14) is updated for each simulation based on the normalized departure values calculated from the latest model state. However, to obtain a strictly quadratic cost function the QC-weight is not allowed to change during the minimisation process, because the cost function shape would then change. Also, when LQVARQC = .TRUE. it is not possible to activate VarQC in the middle of the minimisation process, i.e. NITERQC MUST be zero, because this would otherwise introduce a sudden jump in the cost function. Therefore, in 4D-Var assimilations VarQC is not applied during the first outer-loop but only for the second (and possibly subsequent) outer-loop iteration(s).

The quadratic VarQC implementation satisfies the quadratic constraint by calculating the QC-weight, w_i, based on the high resolution trajectory fields and using this constant weight during the next minimisation step. The cost function value is increased/reduced for each simulation during the minimisation by a factor w_i^*(norm_dep_LR-norm_dep_HR), linearized in the vicinity of the high resolution cost function at norm_dep_HR. For the minimisation the most important input is the modification of the gradient by the weight w_i. During the trajectory run the high resolution departure is stored in ROBODY(..,MDBIFC1(NUPTRA+1)) in HDEPART. For each simulation during the next minimisation step the high resolution departure is read and normalized by the final observation error in HJO HJO loop 1.3.2. This is used as input to the VarQC weight calculations (loop 1.5.3–1.5.4 in HJO). It is assured that the same weight is used for each simulation in the minimisation if LLQVARQC_MIN = .TRUE., because ZFJOS_HR(JOBS,JBODY) is constant during the minimisation. The cost function value is calculated like Jo_varqc_LR = Jo_varqc_HR + w_i^*(norm_dep_LR-norm_dep_HR).

The *a posteriori* probability of gross error is stored in the ODB and passed to BUFR feedback. Storing in ODB is done at the final simulation of the last minimisation if LQVARQC = .F., but done during the final trajectory run if LQVARQC = .TRUE. in order to use the updated final trajectory for calculating the QC-weight based on final high-resolution analysis values. This is done in HJO – see Subsection 5.3.

Variational quality control for time correlated observations: The same method as above is (by default) applied for time correlated observations (LTC = .TRUE.). Here the high resolution departure values are required for all the time correlated observations. This is achieved by storing the normalized high resolution departures in the array RTCNDPHR in HJO and reading them in COMTC. They are then copied to local array ZTCNDPHR_W and then used instead of the low resolution departures in the calculation (overwriting ZOM1DEP_W by high resolution values).

5.6 VARIATIONAL BIAS CORRECTION

Variational bias correction (VarBC, Dee (2004)) of observations was first introduced into the IFS in Cy31r1. VarBC works by including additional degrees of freedom (bias parameters) in the observational term of the 4D-Var cost function to account for possible systematic errors in selected observations and/or observation operators. The systematic errors (or biases) are represented by linear predictor models, which can be formulated separately for different groups of observations.

Configuration of the VarBC system involves specifications of:

observation classes:	types of observations subject to VarBC (e.g. clear-sky radiances; aircraft temperatures)
data groups:	subsets within each class that share the same bias model (e.g. individual radiance channels; reports from specified aircraft types)
bias predictors:	information used to explain the bias in a given data group (e.g. layer thickness; scan position; solar elevation)

The bias correction b for an observation belonging to data group j is defined as

$$b = \sum_{i=0}^{N^j-1} \beta_i^j p_i^j \qquad (5.16)$$

where the p_i^j are the N^j bias predictors associated with the group. The bias parameters β_i^j are adjusted by the variational analysis. By convention $p_0 \equiv 1$, representing a globally constant component of bias, but the other predictors can be flexibly defined. They may depend on properties of the observed atmospheric column, on the state of the instrument, or on any other available information.

If N^g is the number of data groups, then the total number N^p of bias parameters to be estimated is

$$N^p = \sum_{j=1}^{N^g} N^j \qquad (5.17)$$

The number N^j of predictors used for each data group is typically between 1 and 10. As of Cy36r1, the number N^g of data groups subject to variational bias correction is approximately 10^3, and the total number N^p of bias parameters used for this purpose is slightly less than 10^4.

5.6.1 Variational estimation of the bias parameters

Including the bias parameter vector $\beta = \{\beta_i^j, i = 1, \ldots, N^j; j = 1, \ldots, N^g\}$ in the variational analysis replaces equation (1.2) by

$$J(\delta \mathbf{x}, \delta \beta) = \frac{1}{2}\delta \mathbf{x}^{\mathrm{T}}\mathbf{B}^{-1}\delta \mathbf{x} + \frac{1}{2}\delta \beta^{\mathrm{T}}\mathbf{B}_\beta^{-1}\delta \beta + \frac{1}{2}(\mathbf{H}\delta \mathbf{x} + \mathbf{b}(\mathbf{fi}) - \mathbf{d})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{H}\delta \mathbf{x} + \mathbf{b}(\mathbf{fi}) - \mathbf{d}) \qquad (5.18)$$

where $\delta \beta = \beta - \beta^b$ with β^b a prior estimate, usually the result of the previous analysis cycle. The matrix $\mathbf{B}_\beta$ controls the adaptivity of the bias parameter estimates. We use a diagonal $\mathbf{B}_\beta$ with elements

$$\sigma_{\beta_i^j}^2 = (\sigma^j)^2/M^j, \qquad i = 1, \ldots, N^j; \quad j = 1, \ldots, N^g \qquad (5.19)$$

where σ^j is an estimate of the error standard deviation for a typical observation in group j, and M^j is a positive integer. This formula effectively provides the same weight to the prior estimate for β_i^j as it would to M^j newly available observations.

The number M^j in (5.19) is a stiffness parameter for the bias estimates associated with a data group. It should be interpreted as the minimum sample size required to allow significant adjustment of the estimates by VarBC. For satellite observations, the number of data per analysis cycle is typically much larger than the number of bias parameters to be estimated. The background term in that case does not strongly affect the bias estimates unless M^j is extremely large. For data groups with typically small sample sizes, such as station data, the effect of the stiffness parameter M^j is mainly to control the time scale over which bias estimates are allowed to vary.

Preconditioning of the joint minimisation problem (5.18) is based on a separately defined change-of-variable for the bias control parameters, and is described in Dee (2004).

5.6.2 Modular implementation

VarBC is encapsulated in a Fortran object of class CLASS_VARBC that is instantiated during IFS setup by calling the method SETUP_TRAJ or SETUP_MIN. These routines read in the VARBC.cycle file which contains the latest VARBC coefficients and auxiliary data including the background errors. Then, the VarBC object can be used to perform all the necessary VarBC actions, such as the change-of-variable between bias parameter space and control vector space that represents $\mathbf{B}_\beta^{-1}$, or to generate the bias correction $\mathbf{b}(\mathbf{fi}, \mathbf{p})$, given some predictors $\mathbf{p}$ that are computed in the observation operator. This design will be refined further in the future to more clearly separate the change-of-variable and control vector parts from the rest.

VarBC design is modular, with separate code modules for different observation classes. All information needed by VarBC that is specific to the observations contained in a class is handled by the corresponding class module. This includes, for example, the exact definition of a data group, the choice of predictors for each group, the option for initialising bias parameters for new groups, specification of the stiffness parameters, and various other configuration choices. These are examples of class modules (more exist now):

varbc_rad	clear-sky radiance data
varbc_allsky	all-sky radiance data
varbc_to3	ozone layer concentrations (MACC ozone, greenhouse gases, and aerosol optical depth available in Cy36r4)
varbc_tcwv	total column water vapour data from MERIS
varbc_airep	temperature reports from aircraft (available in Cy36r4)

Several generic modules provide functions that are essentially independent of the type of data involved. These are:

varbc_setup	to configure VarBC, administer VarBC data structures, set up preconditioning, provide I/O functions
varbc_pred	to evaluate predictors
varbc_eval	to evaluate the bias models, including TL and AD

Communication between the various class-dependent code modules on the one hand and the generic modules on the other is accomplished by assigning a unique key to each data group within a class. This key is a character string created by the class module, which encodes into the key whatever information it needs in order to uniquely identify the observations for that group. In the case of satellite radiance data, for example, the key contains the satellite id, sensor number, and channel number associated with the group. Only the class module needs to be able to decode the key. The generic modules communicate with the class modules simply by reference to the keys. VarBC assigns a unique integer (index) to each key, and requests the class modules to store this index in the ODB at varbc_idx@body. From that point on, the IFS recognises observations that are subject to VarBC simply by the presence of a positive VarBC index, and handles all VarBC-related operations in essentially the same way, independently of the type of data involved, by reference to its VarBC index.

Each class module must have the following public interfaces to serve the generic code modules ("varbc_xxx" denotes the class module name):

varbc_xxx_config	to configure itself
varbc_xxx_groups	to identify the available data groups in its class
varbc_xxx_groupdescr	to provide a printable description for a data group
varbc_xxx_varbcix	to store the assigned varbc indices for all its data groups in the ODB
varbc_xxx_pred	to provide configuration parameters needed to define and/or initialise the bias predictor models
varbc_xxx_min	to provide configuration parameters for the minimisation

Chapter 6

Background errors

Table of contents

6.1   NOMENCLATURE

The calculation of standard deviations of background errors is unfortunately an area of confusing nomenclature. "Standard deviations of background error" is quite a mouthful, so they are generally referred to simply as 'background errors' (likewise for standard deviations of analysis errors and forecast errors). Although inaccurate, this nomenclature has been adopted in the following for the sake of brevity.

A second source of confusion is that the terms 'background error' and 'forecast error' are often used interchangeably. This confusion has even crept into the code, where the buffer which contains the standard deviations of background error is called FCEBUF. Such confusion is clearly unwise when discussing their calculation. In the following, we describe the processing of error variances during a single analysis cycle. The term 'background error' will be used exclusively to refer to the standard deviations of background error used in the background cost function. The background errors are an input to the analysis. The term 'forecast error' will refer to an estimate of the standard deviation of error in a short-term forecast made from the current analysis. The forecast errors are calculated by inflating an estimate of the standard deviation of analysis error, and are an output from the analysis system. They provide background errors for the *next* analysis cycle.

6.2   INPUT AND 'MASSAGING' OF BACKGROUND ERRORS

Background errors for use in J_b are initialised by a call to SUINFCE. This is part of the J_b set-up, which is described in Subsection 4.3.3.

As with most of the J_b code, the structure SPJB_VARS_INFO, described in Subsection 4.3.1, is important. At the beginning of SUINFCE, this structure is examined to determine which variables are present in J_b, and to locate the corresponding profiles of globally-averaged background error. Next, a call to IO_INQUIRE is made to determine which fields are present in the input file (filename **errgrib**), and to find details of their grid resolution, etc.

The fields of background error are read by a call to IO_GET, and interpolated to the model's horizontal grid using SUHIFCE.

Vertical interpolation requires knowledge of the background surface pressure. In addition, one method of constructing humidity background errors requires knowledge of background values of specific humidity and temperature. So, background fields in grid space are obtained either by a call to GET_TRAJ_GRID, or by transforming the corresponding spectral fields.

Before the vertical interpolation takes place, the locations of the different J_b variables within the background error array FCEBUF are found by a set of calls to the internal subroutine LOOKUP, which interrogates SPJB_VARS_INFO.

The large loop "VARIABLE_LOOP" in SUINFCE loops over the fields of the input file. Each field is interpolated onto model levels by a call to SUVIFCE (unless it is a surface field), and either stored in FCEBUF or used to construct a field in FCEBUF. (For example, if background errors for vorticity are not available in the input file, they are constructed by scaling zonal wind errors.)

Background errors for the humidity variable are treated as a special case, and generated after the main loop over variables by a call to SUSHFCE. The calculation of background errors for humidity is described in Subsection 4.3.4.

Next, one of two routines is called. SUMDFCE calculates a vertically averaged 'pattern' of background error. This is required if the background errors are to be represented as a product of a vertical profile of global mean error and a horizontal pattern. The pattern is stored in FGMWNE. (Note in particular that SUMDFCE is called if horizontally-constant background errors are requested by setting LCFCE. In this case, all elements of FGMWNE are set to one.)

Alternatively, SUPRFFCE is called to calculate global mean profiles of the input background errors. This is the default. The profiles are stored in FCEIMN.

The final step in processing the background errors is performed by SUSEPFCE. This modifies the background errors in one of two ways. If separable background errors have been requested, the contents of the background error buffer are replaced by the product of the vertical profile stored in FCEMN and the horizontal pattern stored in FGMWNE. Otherwise, the background errors for each variable at each level are multiplied by the ratio of the corresponding elements of FCEMN and FCEIMN. The result of this operation is to adjust the global mean profiles of background error to match those stored in FCEMN.

6.3   DIAGNOSIS OF BACKGROUND-ERROR VARIANCES

Background error variances for use in J_b and for observation first guess checks can be computed in two distinct ways. If an Ensemble of Data Assimilations (EDA) is available, background errors can be directly diagnosed from the EDA background forecasts - section 6.6. EDA-derived errors will be used if the logical switch LEDA_ERRORS_IN is true. Otherwise they can be estimated by subroutine BGVECS, which is called from FORECAST_ERROR. One of two methods may be employed, depending on whether NBGVECS is equal to, or greater than, zero. In either case, the estimated variances of background error are stored in the analysis error buffer, ANEBUF (in YOMANEB).

If NBGVECS is zero, then background errors for variables which are explicitly present in the background error buffer, FCEBUF, are copied into ANEBUF and squared. Errors for those variables whose background errors are defined implicitly through the change of variable are estimated using simple scaling of appropriate explicit errors. This scaling is performed by a call to ESTSIGA.

If NBGVECS is non-zero, then the variances of background error are estimated using randomization. This method assumes that the change of variable transforms the background error covariance matrix into the identity matrix. A sample of NBGVECS vectors drawn from a multi-dimensional Gaussian distribution with zero mean and identity covariance matrix is generated by calls to RANDOM_CTLVEC. These vectors are transformed to the space of physical variables by CHAVARIN. The transformed variables form a sample drawn from the distribution of background errors. A call to STEPOTL('0AA00A000') transforms each vector to gridpoint space and accumulates the sums of squares in ANEBUF. In addition, observation operators are applied to the random variables to generate samples of background error in terms of radiance, for example. The sums of squares of these observation-space variables are also accumulated to provide estimates of background error in observation space.

Finally, the sums of squares of random vectors are divided by the number of vectors by a call to SCALEAE to provide a somewhat noisy estimate of the variances of background error actually used in the analysis. Noise may be filtered by a call to FLTBGERR, which transforms the variances to spectral coefficients, multiplies each coefficient by $\cos^2(\min((n/\text{NBGTRUNC}), 1)\pi/2)$, and then transforms to

grid space. The default is to filter with a very large value of NBGTRUNC. Effectively, the background errors are simply spectrally truncated. It is highly recommended that the filtering is performed, since it prevents a grid-scale numerical instability which occurs when the error growth model introduces spatial features which cannot be resolved by the spectral control variable.

Flow-dependent background errors, valid at the end of the window may be generated by setting LBGM (namvar) to TRUE. In this case, the tangent linear model is used to propagate each of the random background vectors to the end of the analysis window. The eigenvectors of the analysis Hessian (see the next section) are also propagated in time, by a call to CNT3TL from XFORMEV. For the normal case, LBGM is false, and background and analysis errors are estimated at the beginning of the analysis window, and propagated using the simple error-growth model described in section 6.7.

The background errors diagnosed by BGVECS may be written out for diagnostic purposes by setting LWRISIGB. The errors are written by a call to WRITESD (called from FORECAST_ERROR).

6.4 CALCULATION OF EIGENVALUES AND EIGENVECTORS OF THE HESSIAN

The second stage in the calculation of analysis errors is to determine eigenvalues and eigenvectors of the Hessian of the cost function. This is done using a combined Lanczos and conjugate-gradient algorithm, implemented in the SQRTPLanczos class. The SQRTPLanczos class inherits from the SQRTMinimizer class that implements a collection of square root minimisers. The latter class, in turn, is derived from the Minimizer class that represents minimisers in the most broad sense. The reader is referred to Fisher (1998, ECMWF Seminar proceedings pp364-385) for a detailed description of the Lanczos algorithm. Note that the Lanczos algorithm requires that the cost function is strictly quadratic.

The SQRTPLancznos::solve() method starts with the normalisation of the first guess gradient (refer to Subsection 2.4 to recall how it is computed) that becomes the first Lanczos vector. Optionally, prior to the normalisation, the first guess gradient can be preconditioned via a call to the SpectralSqrtLMP::inverseMultiplySqrt() method. The Lanczos vectors are stored in a vector of ControlVector data structure called vvecs_.

Each iteration of the conjugate-gradient/Lanczos algorithm starts by calculating the product of the Hessian and the latest search direction. This is calculated as $J''d = [\nabla J(x_0 + d) - \nabla J(x_0)]$, where d" is a vector of unit length. This finite difference formula is exact, since the cost function is quadratic.

The main iteration loop calculates the sequence of gradients, and the sequence of coefficients according to the the Lanczos recurrence. The sequence of control vectors that partially minimise the cost function is not explicitly generated. Consecutive gradients do not, in general, decrease as the minimisation proceeds. The optimal control vector can be determined as a linear combination of the Lanczos vectors. Two convergence criteria are available with one based on monitoring the decrease of the norm of the gradient and the other based on monitoring the relative change in the norm of the control vector. The latter is used in practice with the first metric being still computed for the diagnostic purposes.

The Lanczos algorithm produces a sequence of coefficients, which are stored in the arrays alphas_ and betas_. These correspond to the diagonal and sub-diagonal of a symmetric tri-diagonal matrix. The calculation of the optimal point and the corresponding gradient require the solution of a linear system involving this matrix. This is performed by the TriDiagonalSolve() function, which is a simple interface to the LAPACK routine SPTSV. In addition, the eigenvalues of the tri-diagonal matrix are approximations to eigenvalues of the Hessian of the cost function. These approximate eigenvalues are calculated every iteration, together with bounds on their accuracy. As soon as the leading eigenvalue has converged sufficiently, it is monitored to check that it does not increase. This provides a sensitive test that the algorithm is behaving correctly. Any increase in the leading eigenvalue provides an early indication of failure (for example, due to a bad gradient) and the algorithm is immediately terminated. The calculation is not aborted, since the test detects the failure of the algorithm before the converged eigenvalues and eigenvectors become corrupted.

After the last iteration, the converged eigenvectors of the Hessian are calculated by calling SpectralSqrtLMP::update(). Note that the criterion used to decide which eigenvalues have converged is relaxed at this stage to $\|J''\mathbf{v} - \lambda\mathbf{v}\| < \varepsilon\|\mathbf{v}\|$, where ε is 0.1.

Finally, SQRTPLanczos calculates the optimal control vector as a linear combination of the Lanczos vectors.

6.5 THE PRECONDITIONER

The conjugate-gradient/Lanczos algorithm allows the use of a preconditioner. The preconditioner is a matrix which approximates the Hessian matrix of the cost function. The preconditioner used in the SQRTPLacznos class is a matrix of the form

$$\mathbf{I} + \sum_{i=1}^{L}(\mu_i - 1)\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} \tag{6.1}$$

where the vectors $\mathbf{w}_i$ are orthogonal. The pairs $\{\mu_i, \mathbf{w}_i\}$ are calculated via a call to the SpectralSqrtLMP::update() method at the end of the previous minimisation, and are intended to approximate some of the eigenpairs (i.e. eigenvalues and associated eigenvectors) of the Hessian matrix of the cost function. They are calculated as follows.

A set of L vectors, $\mathbf{u}_i$, is read in when SpectralSqrtLMP is restarted via call to SpectralSqrtLMP::restart(). These vectors are in the space of the control vector.

The vectors are assumed to satisfy

$$\mathbf{I} - \sum_{i=1}^{L}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}} \approx (J'')^{-1} \tag{6.2}$$

The input vectors are not necessarily orthogonal, whereas the preconditioner requires a set of orthogonal vectors. The orthogonalisation procedure described below is implemented in the SpectralSqrtLMP::postprocess() method. Let us denote by $\mathbf{U}$ the matrix whose columns are the vectors $\mathbf{u}_i$. A sequence of Householder transformations is now performed to transform $\mathbf{U}$ to upper triangular. Let us represent this sequence of Householder transformations by the matrix $\mathbf{Q}$. Then $\mathbf{QU}$ is upper triangular, which means that $(\mathbf{QU})(\mathbf{QU})^{\mathrm{T}}$ is zero except for an $L \times L$ block in the top left hand corner.

It is clear that $(\mathbf{QU})(\mathbf{QU})^{\mathrm{T}}$ has only L non-zero eigenvalues. Moreover, the non-zero eigenvalues are the eigenvalues of the $L \times L$ block matrix, and the eigenvectors of $(\mathbf{QU})(\mathbf{QU})^{\mathrm{T}}$ are the eigenvectors of the block matrix, appended by zeroes. These eigenvalues and eigenvectors are calculated by a call to the LAPACK routine SSYEV.

Now, since $\mathbf{Q}$ is an orthogonal matrix, we have $\mathbf{QQ}^{\mathrm{T}} = \mathrm{I}$. So, we may write (6.2) as

$$\mathbf{I} - \mathbf{Q}^{\mathrm{T}}(\mathbf{QU})(\mathbf{QU})^{\mathrm{T}}\mathbf{Q} \approx (J'')^{-1} \tag{6.3}$$

Let us denote the eigenpairs of $(\mathbf{QU})(\mathbf{QU})^{\mathrm{T}}$ by $\{\rho_i, \mathbf{v}_i\}$. Then we may write (6.3) as

$$\mathbf{I} - \sum_{i=1}^{L}\rho_i(\mathbf{Q}^{\mathrm{T}}\mathbf{v}_i)(\mathbf{Q}^{\mathrm{T}}\mathbf{v}_i)^{\mathrm{T}} \approx (J'')^{-1} \tag{6.4}$$

The orthogonality of $\mathbf{Q}$ and the orthonormality of the eigenvectors $\mathbf{v}_i$, means that the vectors $\mathbf{Q}^{\mathrm{T}}\mathbf{v}_i$ are orthonormal. They are, in fact, the required vectors, $\mathbf{w}_i$ of the preconditioner matrix.

Inverting (6.4) gives

$$\mathbf{I} - \sum_{i=1}^{L}\frac{1}{\rho_i}\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} \approx J'' \tag{6.5}$$

Defining $\mu_i = 1 - 1/\rho_i$ gives the required approximation to the Hessian matrix.

The preconditioner vectors are sorted in decreasing order of μ_i, and all vectors for which $\mu_i < 1$ are rejected. These vectors cannot be good approximations to eigenvectors of the Hessian matrix, since the eigenvalues of the Hessian matrix are all greater than or equal to one. A final refinement to the calculation is to reduce large values of μ_i to a maximum of typically 10. This was found to be necessary in practice to avoid ill-conditioning the minimisation.

Application of the preconditioner is straightforward, and is performed by the Spectral-SqrtLMP::inverseMultiplySqrt() method which applies an inverse of the symmetric square root of the preconditioner matrix. Application of the latter matrices relies on the observation that if

$$\mathbf{M} = \mathbf{I} + \sum_{i=1}^{L}(\mu_i - 1)\mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} \tag{6.6}$$

with orthonormal $\mathbf{w}_i$, then the expression for $\mathbf{M}^{-1/2}$ results from replacing μ_i in (6.6) by $1/(\sqrt{\mu_i})$.

6.6 DIAGNOSIS OF BACKGROUND ERROR VARIANCES THROUGH THE EDA

The EDA consists of an ensemble of 50 (as from CY46R1) independent lower-resolution 4DVar assimilations (TCo639L137 with two inner loop minimisations both at TL191) that differ by the explicit perturbation of the observations, the sea-surface temperature and the model physics, plus 1 control, unperturbed 4DVar. The control member is run for diagnostic purposes. The approximate Hessian eigenvectors produced as a by-product of the first minimisation of the control member are also used to accelerate convergence of the perturbed member 4DVars. This results in an approx. 20% reduction in the computational cost of the perturbed minimisations. The reason for running an EDA is that it can be shown (Isaksen *et al.*, 2010) that if the perturbations to the observations, model evolution and boundary conditions are properly chosen, the EDA analysis and background spread will provide realistic estimates of analysis and background errors standard deviations.

In the EDA, for each observation, perturbations are defined by randomly sampling a Gaussian distribution with zero mean and standard deviation equal to the estimate of the observation error standard deviation. Sea surface temperature fields are also perturbed, with correlated patterns derived from the HadISST2 dataset (Kennedy *et al.*, 2016) which has been developed for the ERA5 ECMWF reanalysis (Hirahara *et al.*, 2016). To simulate the impact of model uncertainty, the stochastically perturbed parametrization tendency (SPPT) scheme is used; this perturbs the total parametrized tendency of physical processes (more details can be found in part V of this document). From CY41R2 the EDA cycles its own background error and covariance estimates. It can thus be seen as a variational implementation of a perturbed observation EnKF. This behaviour is activated by setting both environment variables LEDA_ERRORS_IN and LEDA_ERRORS_OUT to true.

The EDA background mean and standard deviations are computed in the enda_pp family which runs after completion of the EDA background forecasts family. The first task of the enda_pp family is ens_stats_members, where the EDA background fields are retrieved from storage and necessary pre-processing operations are carried out. These include the computation of fields which are not archived from the EDA run (Geopotential Height, Relative Humidity, Radiances for a selection of sensors). The ens_stats_members task is split into a number of sub-tasks (ens_stats_mem) equal to the size of the EDA in order to parallelize the computations on the different EDA members.

Task ens_stats_gather computes the ensemble mean and standard deviations of the EDA background fields. These fields are stored as grib files of type 'em' and 'es' of stream 'elda', respectively. From IFS cycle 38r2, standard deviations of the unbalanced components of the control vector (see Section 4.2) are also computed in ens_stats_gather from EDA background perturbations. From IFS cycle 43r3 the EDA error estimates of relative humidity are also used in the 4DVar minimisation. Thus, EDA online estimates for the error fields for the full control vector are now used in 4DVar.

Task ens_fetch_fields gathers the operational analysis fields needed for the calibration step of the EDA standard deviations. This is performed in task ens_cal, where the calibration coefficients of the EDA background variances are computed by Spread_Skill_Time_Avg. Variance Calibration is computed

against the ECMWF operational analysis separately for each parameter, model level and geographical region (northern extra-tropics, tropics, southern extra-tropics).

Variance calibration can be optionally done to ensure that EDA background variance matches the mean squared difference of the EDA background mean and the corresponding operational analysis (as from Cy42R1 default is to perform no calibration of EDA background variances). In order to have a large enough statistical sample and to reduce time variability, the calibration is computed using the latest 10 sets of EDA backgrounds and operational analysis verifying at the same time. Note that as of IFS cycle 41r2, the calibration is only computed for diagnostic purposes but it is no longer applied to the EDA error estimates used in 4DVar.

The last job in the enda_pp family is ens_errors. If the logical switch LENS_CAL is true the EDA background variances computed in ens_stats are calibrated by Ens_Spread_Cal using the coefficients computed in ens_cal. The calibrated variances are then transformed to a wavelet representation and filtered in order to reduce sampling noise. The filtering procedure is based on Bonavita *et al.* (2012). Finally the filtered variances are converted to standard deviations and stored as grib files of type 'ses' of stream 'elda'.

Similar calculations are performed to estimate situation-dependent background errors in radiance space for selected instruments, as described in Bormann and M.Bonavita (2013). This is also performed under the enda_pp family, and it involves four steps:

(i)   Radiance fields are calculated in grid-point space for each ensemble member (using the program calc_radiance_fields from the satrad project in the ens_stats_mem tasks), and ensemble spread and mean are calculated in the ens_stats_gather task, analogous to the spread and mean calculations for model fields.

(ii) Coefficients for the radiance spread calibration are calculated in task ens_cal_rad (using program eda_rad_coefs from the satrad project). In contrast to the calibration of model fields, this calibration is based on observations, and uses past statistics on First Guess departure information and ensemble spread as described in Bormann and M.Bonavita (2013).

(iii) The calibration is applied to the ensemble spread fields in task ens_errors_rad using program eda_rad_scale from the satrad project.

(iv) To reduce sampling noise resulting from the use of a small number of ensemble members, the SES fields are also spatially filtered in task ens_errors_rad, using the Spectral_Filter program as for the model fields.

The result of the above process are GRIB files of type 'ses' of stream 'elda', providing situation-dependent estimates of background errors in radiance space. These fields are used only in the background quality control during the screening step in the IFS.

6.7   DIAGNOSIS OF ONLINE BACKGROUND-ERROR COVARIANCES

Starting with cycle 40R1 the use of the EDA has been extended to allow the online computation of the background-error covariances used in 4DVar in a wavelet representation. A new set of wavelet **B** matrices (see section 4.2) is computed at each assimilation cycle to represent flow-dependent error correlations. In the current implementation (as from cycle 41R2) hybrid wavelet **B** matrices are updated by a linear combination of a pre-existing climatological **B** and a flow-dependent **B** computed from the latest available EDA background forecasts Bonavita *et al.* (2016). This capability is activated by setting the NAMWAVELETJB namelist variable LHYBRID_JB to true. The relative weight given to the online estimate of **B** s specified by the NAMWAVELETJB namelist variable ALPHA_HYBRID_JB. The default value for the current 25 member EDA is 0.3.

The procedure described above introduces a degree of flow-dependency in the horizontal structure functions and in the vertical correlations. In combination with the use of EDA background errors, the use of online background error covariances allows the analysis algorithm to automatically and optimally adapt to changes in the configuration of the global observing system (McNally *et al.*, 2014).

The computation the "wavelet" **B** matrices is organised in two phases. In the first phase (fetch_jb_fields family), EDA forecast perturbations are retrieved from storage and interpolated as required. The second phase (jb_calc family) performs the actual computation of the "wavelet" **B**. Since error covariances are needed at all the spectral resolutions used in 4DVar outer loops, the jb_calc family is organised into separate ifsmin tasks which compute the required **B** matrices in parallel. The computation then proceeds as described in Subsection 4.3.2.

Chapter 7

Gravity-wave control

Table of contents

7.1   INTRODUCTION

In 3D-Var, gravity-wave control is achieved via a penalty term J_c based on the techniques of normal-mode initialization (NMI), in 4D-Var a weak constraint digital filter is used.

Section 7.2 provides a brief overview of NMI techniques, together with references to scientific papers in which further details can be found. Section 7.3 describes the computation of normal modes and related arrays. Section 7.4 documents the implementation of nonlinear NMI in 3D- and 4D-Var, while Section 7.5 describes the computation of J_c based on NMI. Section 7.6 gives an overview of digital filter initialization techniques while Section 7.7 describes its implementation as it is used in the 4D-Var assimilation system.

7.2   NORMAL-MODE INITIALIZATION

If the model equations are linearized about a state of rest, the solutions can (with a certain amount of arbitrariness) be classified into 'slow' (Rossby) and 'fast' (gravity) modes. This classification defines two mutually orthogonal subspaces of the finite-dimensional vector space containing the model state $\mathbf{x}$. Thus, the model state can be written as

$$\mathbf{x} = \mathbf{x}_R + \mathbf{x}_G \qquad (7.1)$$

where $\mathbf{x}_R$ is the 'slow' component and $\mathbf{x}_G$ the 'fast' component. *Linear* NMI consists of removing the fast component altogether ($\mathbf{x}_G = 0$). Since the model is nonlinear, a much better balance is obtained by setting the *tendency* of the fast component to zero ($\dot{\mathbf{x}}_G = 0$); it is this balance condition which *nonlinear* NMI seeks to impose.

Nonlinear NMI was first demonstrated by Machenhauer (1977), in the context of a spectral shallow-water model. For a multi-level model, the first stage in the modal decomposition is a vertical transform; each vertical mode then has its own set of horizontal slow and fast modes (for the shallower vertical modes, all the corresponding horizontal modes can be considered as 'slow'). In the case of a multi-level spectral model using the ECMWF hybrid vertical coordinate the details may be found in the report by Wergen (1987), which also describes techniques for taking into account forcing by physical (non-adiabatic) processes and the diurnal and semi-diurnal tidal signals. Although these options are still coded in the IFS, they are no longer used operationally at ECMWF and will not be described in this documentation.

Implicit normal mode initialization (Temperton, 1988) is based on the observation that, except at the largest horizontal scales, the results of NMI can be reproduced almost exactly without computing the horizontal normal modes at all. The calculation reduces to solving sets of elliptic equations. In the case of a spectral model (Temperton, 1989), these sets of equations are tridiagonal in spectral space. The IFS code includes the option of 'partially implicit NMI', in which the initialization increments are computed using the full 'explicit' NMI procedure for large horizontal scales while the remaining increments at smaller horizontal scales are computed using the simpler implicit procedure.

7.3 COMPUTATION OF NORMAL MODES

7.3.1 Vertical modes

The vertical normal modes depend on the number of levels in the model and on their vertical distribution. They also depend on the choice of reference temperature SITR (assumed isothermal) and reference surface pressure (SIPR). The vertical modes used by the initialization routines are also used in the semi-implicit scheme for the forward integration of the model. The computation of J_b and J_c also uses the vertical normal modes, but for these purposes different values of SITR and SIPR may be selected. Thus the vertical modes are computed both in SUDYN and SUSINMI, the latter being used especially in 4D-Var where it is necessary to alternate between applications using different choices of SITR and SIPR. The vertical modes are computed by first calling SUBMAT to set up a vertical structure matrix and then calling an eigenvalue/eigenvector routine EIGSOL (at the end of SUDYN, it calls routine RG in the auxiliary library). After reordering and normalization, the eigenvectors (vertical modes) are stored in the matrix SIMO, while the corresponding eigenvalues (equivalent depths) are stored in the array SIVP. The inverse of SIMO is computed and stored in SIMI.

7.3.2 Horizontal modes and help arrays

The horizontal normal modes depend on the equivalent depths (see above) and the chosen spectral truncation NXMAX. For 'explicit' NMI, NXMAX is equal to the model's spectral truncation NSMAX. Normally, 'partially implicit NMI' is chosen by setting the switch LRPIMP to .TRUE. In this case the explicit NMI increments are used only up to spectral truncation NLEX (21 by default) but in order to blend the explicit and implicit increments smoothly, explicit increments are computed up to a slightly higher resolution. By default, NXMAX = NLEX + 5.

For most applications of the NMI procedure in the operational suite, it is considered that the larger horizontal scales are best left uninitialized (they include, for example, atmospheric tidal signals and large-scale tropical circulations driven by diabatic processes). To cater for this option there is another logical switch, LASSI ('adiabatic small-scale initialization'), which sets to zero all the initialization increments for total wavenumbers up to NFILTM (= 19 by default). Since only the small-scale increments are used, the NMI can be completely implicit: NLEX is set to 0 and there is no need to calculate the 'explicit' horizontal normal modes.

All the horizontal-normal-mode computations are carried out only for the first NVMOD vertical modes. By default, NVMOD = 5.

The horizontal modes are computed by calling SUMODE3. In turn, SUMODE3E computes the explicit modes and their frequencies while SUMODE3I computes the 'help' arrays required to invert the tridiagonal systems encountered in implicit NMI.

7.4 IMPLEMENTATION OF NMI

Nonlinear NMI can be invoked by calling NNMI3. This is no longer applied by default in IFS, now all gravity-wave control during the assimilation process is done through the penalty term or digital filter. But signicant parts of the NMI code is still used for this, as described in Section 7.5. Model tendencies are computed by calling STEPO to perform one (forward) timestep. The tendencies are then supplied to MO3DPRJ which computes the required increments, using the 'explicit' (Machenhauer) or the 'implicit' scheme (or both, after which the results are merged). The increments are added to the original spectral fields and the process is iterated NITNMI (by default 2) times.

7.5 COMPUTATION OF J_c BASED ON NMI

In the notation of (7.1), the penalty term J_c is defined by

$$J_c = \varepsilon \|(\dot{\mathbf{x}} - \dot{\mathbf{x}}_b)_G\|^2 \tag{7.2}$$

where ε is an empirically chosen weighting factor, $\mathbf{x}$ is the current state of the control variable and $\mathbf{x}_b$ is the background. The norm $\| \ \|^2$ is based on a weighted sum of squares of spectral coefficients. Only the first NVMOD vertical modes are included in the evaluation of (7.2).

J_c is computed by calling the routine COSJC. Control passes through JCCOMP to NMIJCTL, where J_c is evaluated by calling STEPO twice, then projecting the differences in the tendencies on to the gravity modes via MO3DPRJ, and finally computing J_c in NMICOST.

7.6 DIGITAL FILTER INITIALIZATION

Digital filter initialization consists in removing high frequency oscillations from the temporal signal represented by the meteorological fields. A general description of digital filter initialization can be found in Lynch (1993). It can be implemented as a strong constraint by filtering the model fields at the beginning of each forecast or as a weak constraint as described in Gustafsson (1992) and Gauthier and Thépaut (2001).

Time oscillations exceeding a cut-off frequency $\omega_c = (2\pi)/T_c$ can be filtered by applying a digital filter to a time series $f_k = f(t_k)$ for $f_k = k\Delta t$, Δt being the timestep. This proceeds by doing a convolution of $f(t)$ with a step function $h(t)$ so that

$$f \bullet h(t_N) = \sum_{k=-\infty}^{\infty} h_k f_{N-k}$$

The step function h_k is found to be

$$h_k = \frac{\sin(\omega_c k \Delta t)}{k\pi}$$

In practice, the convolution is restricted to a finite time interval of time span T_s. We can write $T_s = 2M\Delta t$ and

$$f \bullet h(t_0) = \sum_{k=-M}^{M} \alpha_k f_k$$

with $\alpha_k = -h_{-k}$. This truncation introduces Gibbs oscillations which can be attenuated by introducing a Lanczos window which implies that the weights α_k are defined as $\alpha_k = -h_{-k} W_k$ with

$$W_k = \frac{\sin((k\pi)/(M+1))}{(k\pi)/(M+1)}$$

An alternative which is used at ECMWF has been proposed by Lynch (1997) to use a Dolph–Chebyshev window in which case

$$W_k = \frac{1}{2M+1} \left[1 + 2r \sum_{m=0}^{M} T_{2M}(x_0 \cos \theta_m/2) \cos m\theta_k \right] \tag{7.3}$$

where $1/x_0 = \cos(\pi\Delta t)/\tau_s$, $1/r = \cosh(2M a\cosh x_0)$, $\theta_k = (k2\pi)/M$ and T_{2M} is the Chebyshev polynomial of degree $2M$. The time span of the window is chosen so that $\tau_s = M\Delta t$.

7.7 IMPLEMENTATION OF DFI AS A WEAK CONSTRAINT IN 4D-VAR

In the context of 4D-Var data assimilation, the digital filter is used as a weak constraint. A penalty term is added to the cost function and replaces the NMI based penalty term. The implementation is based

on Gauthier and Thépaut (2001). The filtered increments are calculated as

$$\delta\bar{\mathbf{X}}(t_{N/2}) = \sum_{k=0}^{N} \alpha_k \delta\mathbf{X}(t_k)$$

where N is the number of time steps in the minimisation and delta represents increments.

During each integration of the tangent linear model in the inner loop of the 4D-Var, the digital filter is applied to the increments and the partial sum accumulated by EDIGFIL. This gives a filtered increment valid at the mid-point of the assimilation window (arrays RACCSPA2 and RACCSPA3). The value of the non-filtered increment valid at the same time is also stored by ECOPSP in arrays RSTOSPA2 and RSTOSPA3. These routines are called in CNT4TL each time-step.

The adjoint model integration calculates the gradient in EDIGFILAD. The gradient is obtained by a single backward integration of the adjoint model. The adjoint calculations associated with the digital filter is a virtually cost free addition to the adjoint observation cost function calculations.

The filtering weights used by the digital filter are calculated during the setup phase by SUEFW. The default is to use a Dolph–Chebyshev non-recursive filter (NTPDFI = 4, see equation (7.3))

The weak constraint term which is added to the cost function is the moist energy norm of the departure between those two states times a weight factor. All these computations are conducted in spectral space and applied to the spectral fields. The default mode at ECMWF is to redefine the norm so the digital filter is only applied to divergence (LDIVONLY = .TRUE.). A larger weight factor (ALPHAG = 100.) is in that case used in the weak constraint term.

The norm of the departure is computed in two steps. In EVJCDFI, the difference between RACCSPA2/RACCSPA3 and RSTOSPA2/RSTOSPA3 is computed and multiplied with ALPHAG for the subset of wave numbers and vertical levels associated with each processor. The cost contribution is calculated for each wavenumber and vertical level using the specified norm. The contributions for all wavenumbers, levels and variables are gathered on each processor by GATHERCOST2 and the total cost contribution is summed in array RSUMJCDFI for each variable and level. Finally, in EVCOST, the contributions from each variable and level are added to obtain the value of the penalty term.

Chapter 8

Diagnostics

Table of contents

8.1   INTRODUCTION

8.1.1   Influence matrix diagnostic in 4D-Var

The influence matrix is used in ordinary least-squares applications for monitoring statistical multiple-regression analyses. Concepts related to the influence matrix provide diagnostics on the influence of individual data on the analysis, the analysis change that would occur by leaving one observation out, and the effective information content (degrees of freedom for signal) in any sub-set of the analysed data. The corresponding concepts have been derived in the context of linear statistical data assimilation in numerical weather prediction. An approximate method to compute the diagonal elements of the influence matrix (the self-sensitivities or observation influence) has been developed for a large-dimension variational data assimilation system (the 4D-Var system of ECMWF).

8.1.2   How to compute the observation influence (OI)

In prepIFS, experiment type 'an' task 'forecast error handling', the logical switch LANOBS=on will allow the OI computation for all observation assimilated. OI is saved in ODB in an_sens_obs.

A Fortran program is available to compute the DFS (Degree of Freedom for Signal) and OI. The program output is a Table that can be imported in XL for graphical display. Geographical maps and time series for all the assimilated observations are produced by OBSTAT.

8.2   OBSERVATIONAL INFLUENCE FOR A DA SCHEME

8.2.1   Linear statistical estimation in numerical weather prediction

Data assimilation systems for NWP provide estimates of the atmospheric state $\mathbf{x}$ by combining meteorological observations $\mathbf{y}$ with prior (or background) information $\mathbf{x}_b$. A simple Bayesian Normal model provides the solution as the posterior expectation for $\mathbf{x}$, given $\mathbf{y}$ and $\mathbf{x}_b$. The same solution can be achieved from a classical *frequentist* approach, based on a statistical linear analysis scheme providing the best linear unbiased estimate (Talagrand, 1997) of $\mathbf{x}$, given $\mathbf{y}$ and $\mathbf{x}_b$. The optimal GLS solution to the analysis problem (see Lorenc, 1986) can be written

$$\mathbf{x}_a = \mathbf{K}\mathbf{y} + (\mathbf{I}_n - \mathbf{K}\mathbf{H})\mathbf{x}_b \tag{8.1}$$

The vector $\mathbf{x}_a$ is the 'analysis'. The gain matrix $\mathbf{K}(n \times p)$ takes into account the respective accuracies of the background vector $\mathbf{x}_b$ and the observation vector $\mathbf{y}$ as defined by the $n \times n$ covariance matrix $\mathbf{B}$ and

the $p \times p$ covariance matrix $\mathbf{R}$, with

$$\mathbf{K} = (\mathbf{R}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1} \tag{8.2}$$

Here, $\mathbf{H}$ is a $p \times n$ matrix interpolating the background fields to the observation locations, and transforming the model variables to observed quantities (e.g. radiative transfer calculations transforming the models temperature, humidity and ozone into brightness temperatures as observed by several satellite instruments). In the 4D-Var context introduced below, $\mathbf{H}$ is defined to include also the propagation in time of the atmospheric state vector to the observation times using a forecast model.

Substituting (8.2) into (8.1) and projecting the analysis estimate onto the observation space, the estimate becomes

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{x}_{\mathrm{a}} = \mathbf{H}\mathbf{K}\mathbf{y} + (\mathbf{I}_p - \mathbf{H}\mathbf{K})\mathbf{H}\mathbf{x}_{\mathrm{b}} \tag{8.3}$$

It can be seen that the analysis state in observation space ($\mathbf{H}\mathbf{x}_{\mathrm{a}}$) is defined as a sum of the background (in observation space, $\mathbf{H}\mathbf{x}_{\mathrm{b}}$) and the observations $\mathbf{y}$, weighted by the $p \times p$ square matrices $\mathbf{I} - \mathbf{H}\mathbf{K}$ and $\mathbf{H}\mathbf{K}$, respectively.

In this case, for each unknown component of $\mathbf{H}\mathbf{x}$, there are two data values: a real and a 'pseudo' observation. The additional term in (8.3) includes these pseudo-observations, representing prior knowledge provided by the observation-space background $\mathbf{H}\mathbf{x}_{\mathrm{b}}$. From (8.3), the analysis sensitivity with respect to the observations is obtained

$$\mathbf{S} = \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{y}} = \mathbf{K}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} \tag{8.4}$$

Similarly, the analysis sensitivity with respect to the background (in observation space) is given by

$$\frac{\partial \hat{y}}{\partial (\mathbf{H}\mathbf{x}_{\mathrm{b}})} = \mathbf{I} - \mathbf{K}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} = \mathbf{I}_p - \mathbf{S} \tag{8.5}$$

We focus here on the expressions (8.4) and (8.5). The influence matrix for the weighted regression DA scheme is actually more complex, but it obscures the dichotomy of the sensitivities between data and model in observation space.

The (projected) background influence is complementary to the observation influence. For example, if the self-sensitivity with respect to the ith observation is S_{ii}, the sensitivity with respect the background projected at the same variable, location and time will be simply $1 - S_{ii}$. It also follows that the complementary trace, $\mathrm{tr}(\mathbf{I} - \mathbf{S}) = p - \mathrm{tr}(\mathbf{S})$, is not the *df* for noise but for background, instead. That is the weight given to prior information, to be compared to the observational weight $\mathrm{tr}(\mathbf{S})$. These are the main differences with respect to standard LS regression. Note that the different observations can have different units, so that the units of the cross-sensitivities are the corresponding unit ratios. Self-sensitivities, however, are pure numbers (no units) as in standard regression. Finally, as long as $\mathbf{R}$ is diagonal, $0 \leq S_{ii} \leq 1$ is assured, but for more general non-diagonal $\mathbf{R}$-matrices it is easy to find counter-examples to that property.

Inserting (8.1) into (8.4), we obtain

$$\mathbf{S} = \mathbf{R}^{-1}\mathbf{H}(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}} \tag{8.6}$$

As $(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})$ is equal to the analysis error covariance matrix $\mathbf{A}$, we can also write $\mathbf{S} = \mathbf{R}^{-1}\mathbf{H}\mathbf{A}\mathbf{H}^{\mathrm{T}}$.

8.2.2   Approximate calculation of self-sensitivity in a large variational analysis system

In a optimal variational analysis scheme, the analysis error covariance matrix $\mathbf{A}$ is approximately the inverse of the matrix of second derivatives (the Hessian) of the cost function J, i.e. $\mathbf{A} = (\mathbf{J})^{-1}$ (Rabier and Courtier, 1992). Given the large dimension of the matrices involved, $\mathbf{J}$ and its inverse cannot be computed explicitly. Following Fisher and Courtier (1995) we use an approximate representation of the Hessian based on a truncated eigen-vector expansion with vectors obtained through the Lanczos algorithm. The calculations are performed in terms of a transformed variable P, $\mathbf{P} = \mathbf{L}^{-1}(\mathbf{x} -$

x_b), with $\mathbf{L}$ chosen such that $\mathbf{B}^{-1} = \mathbf{L}^T\mathbf{L}$. The transformation $\mathbf{L}$ thus reduces the covariance of the prior to the identity matrix. In variational assimilation $\mathbf{L}$ is referred to as the change-of-variable operator (Courtier *et al.*, 1998).

$$\mathbf{J}''^{-1} \simeq \mathbf{B} - \sum_{i=1}^{M} \frac{1-\lambda_i}{\lambda_i}(\mathbf{L}v_i)(\mathbf{L}v_i)^T \tag{8.7}$$

The summation in (8.7) approximates the variance reduction **B-A** due to the use of observations in the analysis. (λ_i, v_i) are the eigen-pairs of $\mathbf{A}$. In ECMWF's operational data assimilation system, the variances of analysis error are computed according to this method. The variances are inflated to provide estimates of short-term forecast (background) error variances to be used as background errors in the next analysis cycle (Fisher, 1996). The Hessian eigen-vectors are also used to precondition the minimization (Fisher and Andersson, 2001). The computed eigen-values are not used to minimize the cost function but only to estimate the analysis covariance matrix. It is well known, otherwise, that the minimization algorithm is analogous to the conjugate-gradient algorithm. Because the minimum is found within an iterative method, the operational number of iterations is sufficient to find the solution (with required accuracy) but does not provide a sufficient number of eigen-pairs to estimate the analysis-error variances.

The diagonal of the background error covariance matrix $\mathbf{B}$ in (8.7) is also computed approximately, using the randomisation method proposed by Fisher and Courtier (1995). From a sample of N random vectors $\mathbf{u}_i$ (in the space of the control-vector), drawn from a population with zero mean and unit Gaussian variance, a low-rank representation of $\mathbf{B}$ (in terms of the atmospheric state variables $\mathbf{x}$) is obtained by

$$\mathbf{B} = \frac{1}{N} \sum_{i=1}^{N}(\mathbf{L}u_i)(\mathbf{L}u_i)^T \tag{8.8}$$

This approximate representation of $\mathbf{B}$ has previously been used by Andersson *et al.* (2000) to diagnose background errors in terms of observable quantities, i.e. $\mathbf{HBH}^T$.

Inserting (8.7) and (8.8) into (8.6) we arrive at an approximate method for calculating $\mathbf{S}$, that is practical for a large dimension variational assimilation (both 3D and 4D-Var). This is given by

$$\mathbf{S} = \mathbf{R}^{-1}\mathbf{H}\left[\frac{1}{N}\sum_{i=1}^{N}(\mathbf{L}u_i)(\mathbf{L}u_i)^T + \sum_{i=1}^{M}\frac{1-\lambda_i}{\lambda_i}(\mathbf{L}v_i)(\mathbf{L}v_i)^T\right]\mathbf{H}^T \tag{8.9}$$

Only the diagonal elements of $\mathbf{S}$ are computed and stored in ODB – that is, the analysis sensitivities with respect to the observations, or self-sensitivities S_{ii}. The cross-sensitivity S_{ij} for $i \neq j$, that represents the influence of the jth observation to the analysis at the ith location, is not computed. Note that the approximation of the first term is unbiased, whereas the second term is truncated such that variances are underestimated. For small M the approximate S_{ii} will tend to be over-estimates. For the extreme case $M = 0$, (8.9) gives $\mathbf{S} = \mathbf{R}^{-1}\mathbf{HBH}^T$ which in particular can have diagonal elements larger than one if elements of $\mathbf{HBH}^T$ are larger than the corresponding elements of $\mathbf{R}$. The number of Hessian vectors operationally computed is $M = 40$ and the number of random $\mathbf{B}$ vectors is $N = 50$.

In general, in the operational system, 15% of the global influence is due to the assimilated observations in any one analysis, and the complementary 85% is the influence of the prior (background) information, a short-range forecast containing information from earlier assimilated observations. About 25% of the observational information is currently provided by surface-based observing systems, and 75% by satellite systems.

Low-influence data points usually occur in data-rich areas, while high-influence data points are in data-sparse areas or in dynamically active regions. Background error correlations also play an important role: High correlation diminishes the observation influence and amplifies the importance of the surrounding real and pseudo observations (prior information in observation space). Incorrect specifications of background and observation error covariance matrices can be identified, interpreted and better understood by the use of influence matrix diagnostics for the variety of observation types and observed variables used in the data assimilation system.

Self-sensitivities cannot be larger than one (they are bounded in the interval zero to one) but, because of the small number of eigenpair we can compute, S_{ii} can be greater than one. Approximations in both of the two terms of (8.9) contribute to the problem. In the second term the number of Hessian eigenvectors is truncated to M. The term is therefore underestimated, and S_{ii} will tend to be over-estimated. The degree of over-estimation depends on the structure of the covariance reduction matrix **B-A**.

For an analysis in which observations lead to strongly localised covariance reduction (such as the humidity analysis with its short co-variance length scales $\sim$180 km, and large observational impacts) a large M is required to approximate **B-A** accurately. The approximate computation is mostly affecting the self-sensitivities close to the upper bound leaving the self-sensitivities <0.7 almost unaffected.

To conclude, the self-sensitivity provides a quantitative measure of the observation influence in the analysis. In robust regression, it is expected that the data have similar self-sensitivity (sometimes called leverage) – that is, they exert similar influence in estimating the regression line. Disproportionate data influence on the regression estimate can have different reasons: First, there is the inevitable occurrence of incorrect data. Second, influential data points may be legitimately occurring extreme observations. However, even if such data often contain valuable information, it is constructive to determine to which extent the estimate depends on these data. Moreover, diagnostics may reveal other patterns e.g. that the estimates are based primarily on a specific sub-set of the data rather than on the majority of the data. In the context of 4D-Var there are many components that together determine the influence given to any one particular observation. First there is the specified observation error covariance **R**, which is usually well known and obtained simply from tabulated values. Second, there is the background error covariance **B**, which is specified in terms of transformed variables that are most suitable to describe a large proportion of the actual background error covariance. The implied covariance in terms of the observable quantities is not immediately available for inspection, but it determines the analysis weight given to the data. Third, the dynamics and the physics of the forecast model propagate the covariance in time, and modify it according to local error growth in the prediction. The influence is further modulated by data density. Low influence data points occur in data-rich areas while high influence data points are in data-sparse regions or in dynamically active areas. Background error correlations also play an important role. In fact, very high correlations drastically lessen the observation influence in favour of background influence and amplify the influence of the surrounding observations.

With the approximate method used here, out-of-bound self-sensitivities occur if the Hessian representation based on an eigen-vector expansion is truncated, especially when few eigen-vectors are used. However, this problem affects only a small percentage of the self-sensitivities computed in this study, and in particular those that are closer to one. Remaining values greater than one can be due to large background to observation error ratio, which is one factor that is known to contribute towards ill-conditioning and poor convergence of the 4D-Var algorithm.

8.3 GRADIENT TEST

If LTEST = .TRUE. a gradient test will be performed both before and after minimization. This is done by the routine GRTEST. In the gradient test a test value t_1 is computed as the ratio between a perturbation of the co-t-function and its first-order Taylor expansion using

$$t_1 = \lim_{\delta\chi \to 0} \frac{J(\chi + \delta\chi) - J(\chi)}{\langle \nabla J, \delta\chi \rangle} \tag{8.10}$$

with $\delta\chi = -\alpha\nabla J$. Repeatedly increasing α by one order of magnitude, printing t_1 at each step should show t_1 approaching one, by one order of magnitude at a time, provided $J(\chi)$ is approximately quadratic over the interval $[\chi, \chi + \delta\chi]$. The near linear increase in the number of 9's in the print of t_1 over a wide range of α (initially as well as after minimization) proves that the coded adjoint is the proper adjoint for the linearization around the given state χ.

The behaviour of the cost function in the vicinity of χ in the direction of the gradient ∇J is also diagnosed by several additional quantities for each α. The results are printed out on lines in the log-file starting with

the string 'GRTEST:'. To test the continuity of J, for example, a test value t_0 is computed with

$$t_0 = \frac{J(\chi + \delta\chi)}{J(\chi)} - 1 \tag{8.11}$$

and printed. For explanation of other printed quantities see the routine GRTEST itself. A range of additional test was introduced in Cy28r2.

8.4 THE 24-HOUR FORECAST ERROR CONTRIBUTION TO OBSERVATIONS (FEC)

The Forecast Sensitivity to Observations Impact (FSOI) is an adjoint-based tool to estimate the reduction of the forecast error due to the assimilation of the observations and is routinely used to monitor the observation performance on short-range forecasts (Baker and Daley, 2000; Cardinali, 2009).

To compute the FSOI for all observation types, ECMWF run an *fsobs* suite. This is a configuration of the IFS (NCONF=801) controlled by CFCSENS2OBS (called from the top-level routine CVA1) and by the namelist NAMVRTL.

The computation of the FSOI involves a scalar cost function, taken here to be the difference between a global dry energy measure of the 24-h and 36-h forecast errors using trajectories starting from an analysis at 00 UTC and from the corresponding background state at 12 UTC. The cost function and its gradient are calculated in COSTRA called from CNT3AD, when L801TL is switched on. After the sensitivity gradients are provided valid at the beginning of the assimilation window and after they are transformed to control variable space by applying CHAVARINAD, the linear system (eq. 2.10 from Cardinali (2009)) is solved using the CONGRAD algorithm. The forecast sensitivity to observations is then obtained after interpolating in the observation space and after normalizing with respect to the observation error covariance matrix. Finally, the contribution of each observation towards reducing the 24-hour forecast error is estimated using the innovation and the forecast sensitivity to observations vectors.

FSOI is saved in ODB in the fc_sens_obs@body column. A positive fc_sens_obs@body value is associated with forecast error increase and a negative fc_sens_obs@body contribution with forecast error decrease.

FSOI statistics are displayed on the web, as part of the monitoring of the observing system information:

http://www.ecmwf.int/en/forecasts/charts/obstat

Chapter 9

Land-surface analysis

Table of contents

9.1   INTRODUCTION

The land-surface analysis includes the screen-level parameters analysis, the snow depth analysis, the soil moisture analysis, the soil temperature and snow temperature analysis. The screen level parameters analysis and the snow analysis rely on a two-dimensional Optimal Interpolation (2D OI). The soil moisture analysis uses an EDA-based simplified Extended Kalman Filter (EKF). The soil and snow temperature analysis uses a one-dimensional OI (1D OI).

The surface-analysis module includes also the sea-surface temperature, sea-ice fraction and screen-level temperature analysis described separately in Chapter 10.

Snow water equivalent, soil temperature, snow temperature and soil water content are prognostic variables of the forecasting system and, as a consequence, they need to be initialised at each analysis cycle. The ECMWF soil moisture, soil temperature and snow temperature analyses rely on SYNOP relative humidity and temperature at screen-level (2 metres) available on the GTS (around 10,000 reports over the globe are assimilated every 6 hours). For the soil moisture analysis, the analysed relative humidity and temperatures fields are assimilated as pseudo-observations. The MetOp-A and MetOp-B Advanced Scatterometer (ASCAT) soil moisture products are also assimilated along with the Soil Moisture and Ocean Salinity (SMOS) neural network soil moisture data. The snow analysis relies on SYNOP and national ground observations of snow depth available on the GTS, as well as on the NOAA/NESDIS (National Oceanic and Atmospheric Administration - National Environmental Satellite, Data, and Information Service) Interactive Multi-sensor Snow and Ice Mapping System (IMS) snow cover information.

The structure of the land surface analysis components and their dependencies are shown in Fig. 9.1. Firstly, a screen-level analysis is performed for temperature and humidity. Secondly, the snow analysis and the soil moisture analysis are conducted. Then the soil temperature and snow temperature analysis is performed. Analysed screen-level temperature and relative humidity are used as input of the
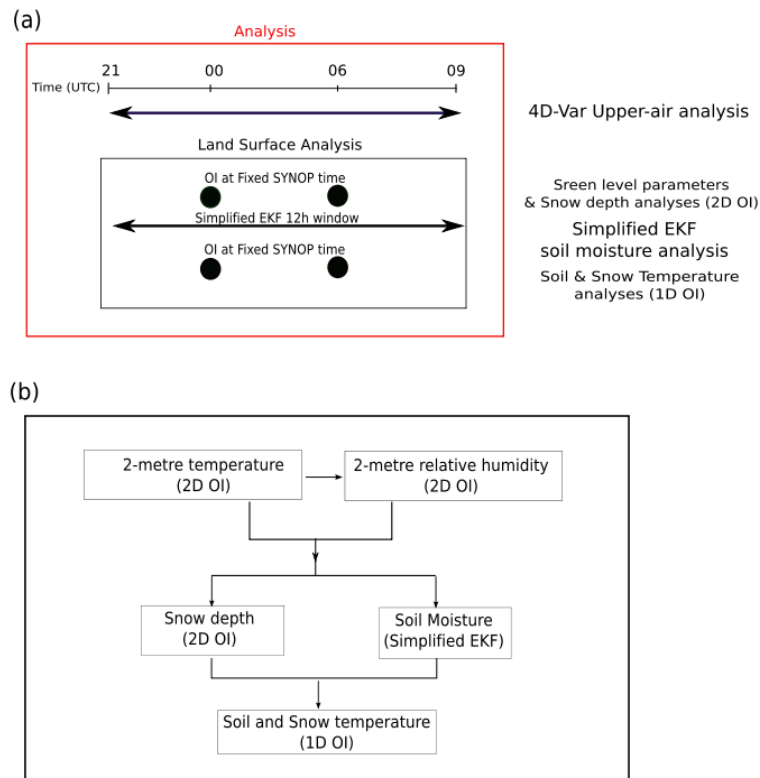
Figure 9.1 *Land surface analysis components and tasks organisation, for an example of a 12-hour (21:00 UTC to 09:00 UTC) data assimilation window.*

simplified EKF soil moisture analysis. while screen-level analysis temperature increments are used as inputs of the 1D OI soil and snow temperature analysis.

Land analysis is run separately from the upper-air analysis. It feedbacks to the upper-air analysis of the next cycle through its influence on the short forecast that propagates information from one cycle to the next. Reciprocally, the 4D-Var influences through the short forecasts the land surface analysis from one cycle to the next. The OI analyses of screen level parameters, snow depth, snow and soil temperature are performed at fixed times at 0000, 0600, 1200, and 1800 UTC. The simplified EKF analysis runs at the same time as the 4D-Var windows for both the delayed cut-off and the early delivery analyses.

9.2 2D OPTIMAL INTERPOLATION SCREEN-LEVEL ANALYSIS

9.2.1 Methodology

Two independent analyses are performed for 2-metre temperature and 2-metre relative humidity. The method used is a two-dimensional univariate Optimal Interpolation (2D OI). In a first step, the background field (6-hour or 12-hour forecast) is interpolated horizontally to the observation locations using a bilinear interpolation scheme and background increments ΔX_i are estimated at each observation location i.

The analysis increments ΔX_p^a at each model grid-point p are then expressed as a linear combination of the first-guess increments (up to N values) given by

$$\Delta X_p^a = \sum_{i=1}^{N} W_i \times \Delta X_i \tag{9.1}$$

where W_i are optimum weights given (in matrix form) by

$$(\mathbf{B} + \mathbf{O})\mathbf{W} = \mathbf{b} \tag{9.2}$$

The column vector $\mathbf{b}$ (dimension N) represents the background error covariance between the observation i and the model grid-point p. The ($N \times N$) matrix $\mathbf{B}$ describes the error covariances of background fields between pairs of observations. The correlation coefficients (structure functions) of $\mathbf{b}$ and $\mathbf{B}$ are assumed to have the form:

$$\mu_{ij} = \alpha_{slv}(r_{ij})\beta(\Delta z_{ij}) \tag{9.3}$$

where r_{ij} and z_{ij} are the horizontal and the vertical separation between points i and j, respectively. $\alpha_{slv}(r_{ij})$ and $\beta(\Delta z_{ij})$ are the horizontal and vertical structure functions respectively:

$$\alpha_{slv}(r_{ij}) = \exp\left(-\frac{1}{2}\left[\frac{r_{ij}}{d}\right]^2\right) \tag{9.4}$$

$$\beta(\Delta z_{ij}) = \exp\left(-\left[\frac{\Delta z_{ij}}{h}\right]^2\right) \tag{9.5}$$

d is the horizontal length parameter set to 300 km, corresponding to the e-folding distance taken of 420 km. h is the vertical length scale taken to 800 m.

The correlation matrix B is expressed as:

$$B(i,j) = \sigma_b^2 \times \mu(i,j) \tag{9.6}$$

with σ_b the standard deviation of background errors.

The covariance matrix of observation errors $\mathbf{O}$ is set to $\sigma_o^2 \times \mathbf{I}$ where σ_o is the standard deviation of observation errors and $\mathbf{I}$ the identity matrix.

The standard deviations of background and observation errors are set respectively to 1.5 K and 2 K for temperature and 5% and 10% for relative humidity. The number of observations closest to a given grid point that are considered for solving (9.1) is $N = 50$ (scanned within a radius of 1000 km). The analysis is performed over land and ocean but only land (ocean) observations are used for model land (ocean) grid points.

9.2.2 Quality controls

Gross quality checks are first applied to the observations such as $RH \in [2, 100]$ and $T > T^d$ where T^d is the dew point temperature. Redundant observations are also removed by keeping only the closest (and more recent) to the analysis time.

Observation points that differ by more than 300 m from the model orography are rejected.

For each datum a check is applied based on statistical interpolation methodology. An observation is rejected if it satisfies

$$|\Delta X_i| > \gamma\sqrt{\sigma_o^2 + \sigma_b^2} \tag{9.7}$$

where γ has been set to 3, both for temperature and humidity analyses.

The number of used observations every 6 hours varies between 4,000 and 6,000 corresponding to around 40% of the available observations.

The final relative humidity analysis is bounded between 2% and 100%. The final MARS archived product is dew-point temperature that uses, following Buck (1981), the 2-metre temperature analysis T_a to perform the conversion so that

$$T^d = \frac{17.502 \times 273.16 - 32.19 \times \Psi}{17.502 - \Psi} \tag{9.8}$$

with

$$\Psi = \log(RH_a) + 17.502 \times \frac{T_a - 273.16}{T_a - 32.19} \tag{9.9}$$

- SSA
  - INIT2M / INISNW
    - COORDINATES
    - GETFIELDS
    - FIELD2ARRAY
    - SCAN_CMA_ODB
      - INIT_SURFBODY_FEEDBACK
      - GET_DB
      - FETCH_VARIABLE
      - FILL_RAWTABLE
      - LAND_OBS
      - INITIAL_REJECTION
  - T2M_ANALYSIS / SNOW_ANALYSIS
    - FG2OBS
    - REDUNDANT_OBS
    - OIUPD
      - CALC_DISTANCE
      - OISET
      - OIINC
        - EQUSOLVE
        - DOT_PRODUCT
  - FDB_OUTPUT
  - PRINT_SUMMARY
  - FEEDBACK_ODB

Figure 9.2 *Program organisation when performing the 2D OI screen-level parameters and snow depth analyses. Red colour highlights differences between the screen-level and snow analyses.*
.

9.2.3 Technical aspects

The screen-level analysis software is implemented as a branch of the more comprehensive surface and screen-level analysis (SSA) package. The other branches currently include snow analysis and sea surface temperature and sea-ice fraction analyses. The program organisation when performing screen-level parameters analysis is roughly as shown in Fig. 9.2.

The main program is SSA. It starts with the setup and namelist handling. The routine INIT2M performs initialisation of the analysis of the actual screen-level parameters by generating latitudinal coordinates that stem from the model resolution and zeros of the Bessel function. It also reads input fields into memory in GETFIELDS and FIELD2ARRAY. Input fields consist of the 2-metre temperature, 2-metre dew point temperature, 2-metre relative humidity from the first-guess (6-hour or 12-hour forecasts), land/sea mask and the orography in a form of the geopotential. Then SCAN_CMA_ODB is called to read observations into memory and a quick validity check of the non-applicable observations for this analysis is performed.

Additional screening is done in INITIAL_REJECTION and later on in REDUNDANT_OBS. The first of these sets up an internal table where all the observations which survived from the quick screening are placed with a minimum amount of context information. This routine rejects some of the observations entered into the table due to inconsistencies.

The actual 2-metre temperature and 2-metre relative humidity analyses are performed under T2M_ANALYSIS. First the background field is interpolated to the observation location and the background increment is computed at each observation location in FG2OBS. In this routine, the first guess departure check are applied and the blacklist is read.
Then the routine REDUNDANT_OBS removes time duplicates and retains the observations of the considered station with the closest (and the most recent) to the analysis time. Slowly moving platform handling is present in the REDUNDANT_OBS for the 2-metre temperature and 2-metre relative

humidity analyses.

The analysis increments are computed in the subroutine OIUPD. Subroutines CALC_DISTANCE and OISET select and sort the N closest observations from a given grid-point. Subroutine OIINC provides the analysis increments from (9.1) and (9.2), by first computing $\mathbf{q} = (\mathbf{B} + \mathbf{O})^{-1}\Delta\mathbf{X}$ (in subroutine EQUSOLVE – inversion of a linear system), which does not depend upon the position of the analysis grid point, and then estimating $\mathbf{b}^{\mathrm{T}}\mathbf{q}$ (in subroutine DOT_PRODUCT).

Most of the control parameters of the screen-level analysis are defined in the namelist NAMSSA.

(i)   C_SSA_TYPE: 't2m' for temperature analysis and 'rh2m' for relative humidity analysis.
(ii)   L_OI : .TRUE. for statistical interpolation and .FALSE. for Cressman interpolation.
(iii) N_OISET: number of observations (parameter N).
(iv)   SIGMAB: standard deviation of background error (parameter σ_{b}).
(v)   SIGMAO: standard deviation of observation error (parameter σ_{o}).
(vi)   TOL_RH: Tolerance criteria for RH observations (parameter γ in (9.7)).
(vii) TOL_T: Tolerance criteria for T observations (parameter γ in (9.7)).
(viii) SCAN_RAD_2M(1): Scanning radius for available observations (set to 1000 km).

9.3   2D OPTIMUM INTERPOLATION SNOW ANALYSIS

9.3.1   Background and observations

The snow analysis is a two-dimensional Optimal Interpolation performed every 6 hours, at 00 UTC, 06 UTC, 12 UTC and 18 UTC. From IFS cycle 48R1, up to 5 snow layers allow to represent the snowpack vertical distribution of snow mass and density in the background forecast, described in detail in the IFS documentation Part IV: Physical processes Chapter 8.4. The snow analysis is performed on total (e.g. integrated) quantities and its methodology remains the same as in pre-48R1 cycles. Bulk analysis increments are then partitioned onto the different layers. The total snow-depth background $S_{\mathrm{s}}^{\mathrm{b}}$ (units: m) and equivalent snow density $\rho_{\mathrm{s}}^{\mathrm{b}}$ are estimated from the short-range forecast of snow water equivalent $SWE_{\mathrm{i}}^{\mathrm{b}}$ (units: kg m^{-2}) and snow density $\rho_{\mathrm{i}}^{\mathrm{b}}$ (units: kg m^{-3}) integrated over the $N_l = 5$ snow layers, given by

$$S_{\mathrm{s}}^{\mathrm{b}} = \sum_{i=1}^{N_l} \frac{SWE_{\mathrm{i}}^{\mathrm{b}}}{\rho_{\mathrm{i}}^{\mathrm{b}}}$$

$$\rho_{\mathrm{s}}^{\mathrm{b}} = \frac{\sum_{i=1}^{N_l} SWE_{\mathrm{i}}^{\mathrm{b}}}{S_{\mathrm{s}}^{\mathrm{b}}} \ . \tag{9.10}$$

The snow analysis is performed using snow-depth observations, the snow-depth background field, and the high resolution (4km) NOAA/NESDIS snow extent. Snow depth observations include conventional snow depth reports from SYNOP stations as well as additional national snow depth observations reported by several member states and available on the GTS (de Rosnay *et al.*, 2011b, 2014, 2015). If snow-depth observations are not available, the snow accumulation/melting is simulated from the model 6-hour forecast.

In the current default configuration, the satellite derived snow extent is used once per day, for the 00 UTC analysis. Snow extent is a binary information that indicates snow free or snow covered conditions of each pixel. It is converted in the Observation Data Base into a quantitative snow depth information. To this end the model relation between snow extent and snow depth is used as observation operator, with 5 cm of snow depth where binary snow cover is one and 0 cm snow depth where binary snow cover is zero. The latter observations enter the analysis whatever the first guess conditions are. In contrast the 5 cm snow depth derived from the snow cover observations enter the analysis only where the model first guess indicates snow free conditions. As an alternative, snow climate can be used to ensure the stability of the scheme and to give a seasonal snow trend in areas without any observations.
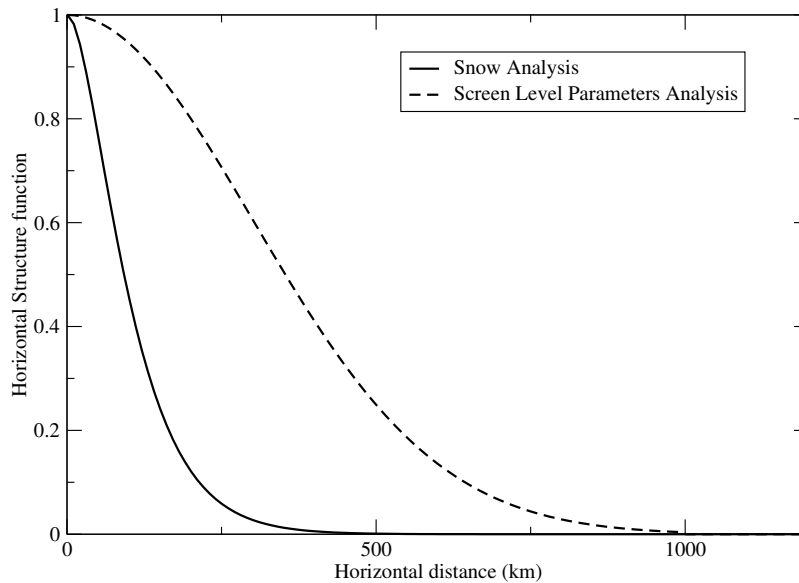
Figure 9.3 *Horizontal structure functions used in the Optimal Interpolation scheme for the screen level parameters and snow depth analyses.*

9.3.2 Methodology

The OI snow analysis is performed based on N observations from ground stations reports and satellite observations, which enter the analysis with a snow depth of 0 cm (snow free) or 5 cm (snow covered, used only if the model first guess is snow free). Following the same 2D OI method than for the screen level parameters analysis, the snow depth analysis increment is computed at each model grid point p:

$$\Delta S_p^{\mathrm{a}} = \sum_{i=1}^{N} W_i \times \Delta S_i \tag{9.11}$$

where ΔS_i is the background increment at each observation i location, W_i are optimum weights given (in matrix form) by:

$$(\mathbf{B} + \mathbf{O})\mathbf{W} = \mathbf{b} \tag{9.12}$$

The column vector $\mathbf{b}$ (dimension N) represents the background error covariance between the observation i and the model grid-point p. The $(N \times N)$ matrix $\mathbf{B}$ describes the error covariances of background fields between pairs of observations (i, j).

For the snow analysis the correlation coefficients (structure functions) of $\mathbf{b}$ and $\mathbf{B}$ follow the formulation proposed by Brasnett (1999):

$$\mu_{ij} = \alpha_{sno}(r_{ij})\beta(\Delta z_{ij}) \tag{9.13}$$

where r_{ij} and z_{ij} are the horizontal and the vertical separation between points i and j, respectively. $\alpha_{sno}(r_{ij})$ and $\beta(\Delta z_{ij})$ are the horizontal and vertical structure functions:

$$\alpha_{sno}(r_{ij}) = (1 + \frac{r_{ij}}{L}) \exp(-\frac{r_{ij}}{L}) \tag{9.14}$$

$\beta(\Delta z_{ij})$ is the same vertical structure function than that used for the screen level analysis (Equation 9.5). L is the horizontal length parameter taken to 55000m, corresponding to an e-folding distance taken to 120 km.

The structure function parameters L (d for the screen level analysis) and h are controlled in the namelist NAMSSA and in the current version their value is defined in the script SSAANA. The horizontal structure functions used in the snow analysis and in the screen level parameters analysis are represented in Fig. 9.3.

The correlation matrix B is expressed as:

$$B(i, j) = \sigma_b^2 \times \mu(i, j) \tag{9.15}$$

with σ_b the standard deviation of background errors.

The covariance matrix of observation errors $\mathbf{O}$ is set to $\sigma_o^2 \times \mathbf{I}$ where σ_o is the standard deviation of observation errors and $\mathbf{I}$ the identity matrix.

The standard deviations of background, in situ observation and snow extent observation errors are set to 3 cm, 4 cm and 8 cm, respectively. The number of observations closest to a given grid point that are considered is $N = 50$ (scanned within a radius of 250 km).

The total snow water equivalent product SWE_s^a, in m of water equivalent, is then calculated using the analysed snow depth S^a:

$$SWE_s^a = \frac{\rho_s^b \times S^a}{1000} \tag{9.16}$$

The snow density is updated to ensure the consistency between the analysis and the physical parameterisation:

- In the case of positive analysis increments on snow free areas in the first guess, the snow density of fresh snow is applied. It accounts for compaction effects of fresh snow due to meteorological conditions, following a function of near surface air temperature and wind speed as described in the IFS documentation Part IV: Physical processes Chapter 8.4.3.
- In the other cases, the snow density remains unchanged.

To map the single-layer increments to the multi-layer prognostic variables, the total SWE analysis increment, $\Delta SWE_s^a = SWE_s^a - SWE_s^b$, is partitioned into the layers so that the sum of SWE increments for all layers N_l is consistent with the total SWE increment of Eq. 9.16, thus satisfying the following relation:

$$\sum_{i=1}^{N_l} SWE_i^a = 10^3 \, SWE_s^a \,, \tag{9.17}$$

with the multiplying factor 1000 converting the units from m of water equivalent to kg m^{-2}. Depending on the sign of total increments, the following partitioning procedure is followed:

- $\Delta SWE_s^a \geq 0$: the analysis increment of SWE is added to the top snow layer; in case of increments over snow free areas in the first guess, the snow density of the top layer is modified accordingly using the density of fresh snow, as previously described;
- $\Delta SWE_s^a < 0$: the analysis increment of SWE are applied to all layers, proportionally to the depth of each layer.

Areas with permanent snow and ice (defined using the Global Land Cover Characterisation product) are set to an arbitrary high value at each analysis cycle ($SWE_s^a = 10$ m of water equivalent).

The snow temperature analysis is performed within the soil and snow temperature analysis as described in Section 9.5.

9.3.3 Quality controls

For each datum a check is applied based on statistical interpolation methodology. An observation is rejected if it satisfies

$$|\Delta S_i| > \gamma \sqrt{\sigma_o^2 + \sigma_b^2} \tag{9.18}$$

where γ is the tolerance, controlled in the namelist NAMSSA and set to 5 for the snow analysis in the SSAANA script.

Use of satellite data snow extent in mountainous area is switched off in the subroutine CALC_DISTANCE, based on an altitude threshold of 1500 m. This prevents from using relatively large scale information (4 km resolution of the NOAA/NESDIS product), for grid points where local conditions apply.

In addition to the preliminary quality control, the following checks are applied for each grid point when using SYNOP snow depth report.

(i)   If $T_{2m}^b < 8°C$ only snow-depth observations below 140 cm are accepted.
(ii) This limit is reduced to 70 cm if $T_{2m}^b > 8°C$.
(v)   Snow-depth analysis is limited to 140 cm.
(vi) Snow-depth increments are set to zero when larger than $(160 - 16T_{2m}^b)$ mm (where T_{2m}^b is expressed in Celsius).

Satellite-derived snow extent is used in the analysis to replace the role of snow depth climatology in correcting for the model bias. However, there is the option to weight the analysis of snow depth with the climatological value S^{clim} so that the final analysis is provided by

$$S^a = (1 - \alpha)S^a + \alpha S^{clim} \tag{9.19}$$

The relaxation coefficient α can be changed through the namelist. Its default value is set to 0.02, which corresponds to a time scale of 12.5 days at six-hourly cycling.

9.3.4 Technical aspects

The technical aspects are very similar to those of the screen-level parameters analysis (see Section 9.2.3). So, the program organisation when performing snow analysis is as shown in Fig. 9.2.

The routine INISNW performs initialisation of the actual snow analysis (instead of INIT2M for the screen-level parameters analysis). Input fields read into memory in GETFIELDS and FIELD2ARRAY consist of the snow water equivalent and snow density from the first-guess (6-hour forecast), 2-metre temperature first guess, snow-depth climate (varies monthly with a linear temporal interpolation), land/sea mask and finally the orography in a form of the geopotential. The actual snow analysis is performed under SNOW_ANALYSIS.

The analysis increments are computed in the subroutine OIUPD. Subroutines CALC_DISTANCE and OISET select and sort the N closest observations from a given grid-point. Subroutine OIINC provides the analysis increments computed from equations (9.11) and (9.12). Increments are finally added to the snow-depth fields at grid points producing the final snow-depth output field, which is output in routine FDB_OUTPUT. Statistics are summarised in PRINT_SUMMARY.

The main logicals of the namelist NAMSSA are as follows.

(i)   L_SNOW_ANALYSIS: When set to .TRUE., the snow analysis is performed.
(ii)   L_SNOW_DEPTH_ANA: When set to .TRUE., the snow analysis is performed in snow depth (in opposition to snow water equivalent assuming a constant value of $250\,kg\,m^{-2}$ for observed snow density).
(iii) L_USE_SNOW_CLIMATE: When set to .TRUE., a relaxation of the snow analysis towards a monthly climatology is performed with a time scale of 12.5 days (the relaxation coefficient is passed through NAMSSA).
(iv)   L_USE_FG_FIELD: When set to .TRUE. the snow analysis is set to the first-guess value (no use of observations) and there is no relaxation to climatology.
(v)   L_USE_SCOVER_NESDIS: When set to .TRUE., the NOAA/NESDIS satellite product is used.

Most of the control parameters of the screen-level analysis are defined in the namelist NAMSSA. In addition to those defined in Section 9.2.3

(vi)   TOL_SN: Tolerance criteria for snow depth observations (parameter γ in 9.18).

(viii) RSCALE_X: horizontal distance L used in the horizontal structure function in (9.14), set to 55000 m.

(ix) RSCALE_Z: vertical distance h used in the vertical structure function in (9.5), set to 800 m.

9.4 SIMPLIFIED EXTENDED KALMAN FILTER SOIL MOISTURE ANALYSIS

9.4.1 Methodology

The simplified EKF soil moisture analysis used at ECMWF is a point wise data assimilation scheme. As described in Drusch *et al.* (2009); de Rosnay *et al.* (2011c, 2013), the analysed soil moisture state vector $\mathbf{x}^a$ a is computed for each grid point as:

$$\mathbf{x^a} = \mathbf{x^b} + \mathbf{K} \left[\mathbf{y^o} - \mathcal{H}(\mathbf{x^b}) \right] \tag{9.20}$$

with superscripts a, b, o standing for background, analysis and observations, respectively, $\mathbf{x}$ the model state vector, $\mathbf{y}$ the observation vector and $\mathcal{H}$ the non-linear observation operator. The Kalman gain matrix $\mathbf{K}$ is expressed as:

$$\mathbf{K} = \left[\mathbf{P^{-1}} + \mathbf{H^T R^{-1} H} \right]^{-1} \mathbf{H^T R^{-1}} \tag{9.21}$$

where $\mathbf{H}$ is the linearised observation operator, $\mathbf{P}$ is the error covariance matrix associated with $\mathbf{x}$ and $\mathbf{R}$ is the observation errors covariance matrix.

The background error covariance matrix $\mathbf{P}$ and the observation error matrix $\mathbf{R}$ are diagonal with terms composed of error variances. These terms are based on static background errors, with soil moisture standard deviation $\sigma_b = 0.01 m^3 m^{-3}$, and screen levels parameters standard deviations of $\sigma_T = 1K$ for the 2-metre temperature and $\sigma_{RH} = 4\%$ for the relative humidity. The ASCAT soil moisture data observation error is also static and it is set to $\sigma_{ascat} = 0.05 m^3 \cdot m^{-3}$. For SMOS soil moisture, the product observation uncertainty ϵ_{smos} is accounted for in the SMOS observation error which is expressed as $\sigma_{smos} = 0.02 + 3 \cdot \epsilon_{smos} m^3 \cdot m^{-3}$.

In contrast to Drusch *et al.* (2009); de Rosnay *et al.* (2011c, 2013), in which the linearised observation operator was computed in finite differences using trajectories with model state vector perturbations, the current SEKF uses the EDA spread to compute the Jacobians. On each grid point, the EDA covariances between soil moisture in each analysed layer and model predicted values of 2-metre temperature relative humidity and surface soil moisture are computed and divided by the corresponding layer soil moisture variances. Using i to represent the ith element of the observations vector and j for the model state vector, corresponding to the soil layer index, the Jacobian elements $\mathbf{H_{ij}}$ of the observation operator are:

$$\mathbf{H_{ij}} = \frac{cov(\mathcal{H_i}(\mathbf{x^{eda}}), \mathbf{x_j^{eda}})}{var(\mathbf{x_j^{eda}})} \cdot c_j \tag{9.22}$$

with the superscript *eda* indicating that EDA fields are used, and with $c_j = 1/(1 + (j-1) \cdot \alpha_{sekf})$ and $\alpha_{sekf} = 0.6$ a tapering coefficient that reduces the Jacobians at depth.

The Jacobians elements are computed at the observation time. The analysis increments are applied at the analysis time t. The model state vector evolution from time t to time $t + \Delta t$ is then controlled by the equation:

$$\mathbf{x}^b(t + \Delta t) = \mathcal{M_t} \left[\mathbf{x^a(t)} \right] \tag{9.23}$$

with $\mathcal{M}$ the non-linear state forecast model.

In the current implementation, the state vector is soil moisture. It has dimension $j_{max} = 3$ since the first three layers of the HTESSEL LSM are analysed. The observations vector $\mathbf{y}$ includes 2-metre temperature and relative humidity analyses as well as ASCAT-A and ASCAT-B rescaled soil moisture de Rosnay *et al.* (2011a), and SMOS neural network soil moisture. For each grid point, a maximum of two ASCAT and two SMOS soil moisture observations are used. When 12-hour assimilation windows are used, $\mathbf{y}$ has dimension $i_{max} = 8$ since 2-meter temperature and relative humidity analyses are available twice per assimilation window, at synoptic times.

9.4.2 Quality controls

To avoid spurious corrections, observations are rejected if the Jacobians become larger than $50K/m^3m^{-3}$ for T_{2m}, $5/m^3m^{-3}$ for RH_{2m}, or $2m^3m^{-3}/m^3m^{-3}$ for ASCAT and SMOS soil moisture. In addition, the analysis is locally switched off if the soil moisture increment for any layer is larger than $0.1m^3m^{-3}$.

9.4.3 Technical aspects

The simplified EKF soil moisture analysis is a configuration of the IFS (NCONF=302). It is controlled by the namelist NAMSEKF. In the current IFS cycle, a number of control parameters of this namelist are defined in the SEKF_SM script:

- Number of control variables, N_SEKF_CV=3, (i.e. number of analysed soil layers)
- EDAH_TAPER=0.6, for $\alpha_{sekf} = 0.6$
- Background soil moisture error, BACK_ERR=0.0001 m^6m^{-6}, corresponds to $\sigma_b = 0.01m^3m^{-3}$
- Screen level analysis errors, T2M_ERR=1, RH2M_ERR=16, correspond to $\sigma_T = 1K$ and $\sigma_{RH} = 4\%$
- ASCAT soil moisture errors, ASCAT_ERR=0.0025, corresponds to $\sigma_{ASCAT} = 0.05m^3m^{-3}$.
- For the SMOS soil moisture error, SMOS_SM_COEF_ERR=3 and SMOS_SM_MIN_VOL_ERR=0.02 for $\sigma_{smos} = 0.02 + 3 \cdot \epsilon_{smos}m^3 \cdot m^{-3}$.
- A switch is present to activate the use of the ASCAT (Advanced SCATterometer) soil moisture data (LUSE_ASCAT). From IFS cycle 40r3 is set to true by default.
- A switch is present from IFS cycle 46r1 to activate the use of the SMOS soil moisture data (LUSE_SMOS_SM). It is set to true by default.

Fig. 9.4 shows the main steps of the simplified EKF soil moisture analysis. CNT0, SEKF1 and SEKF2 are the control programs. From CSEKF2, SUSEKF does the setup of the EKF analysis. CNT3 is used to run a single model trajectory by calling CNT4 which stores the model soil moisture that will enter equation 9.20.

OBSGEN and SEKF_PREP_ASCAT are the routine that prepare the observations to be used in the simplified EKF analysis. OBSGEN reads the gridded observations (screen level observations and SMOS neural network soil moisture) and SEKF_PREP_ASCAT reads the ASCAT ODB. Then CNT4 calls the simplified EKF programme SM_EKF_MAIN which runs the EKF on each grid point of the model. Here the simplified EKF vectors and matrices defined in Equations 9.20 to 9.22 are filled with model and observation values stored in the previous routines. SEKF_MAGN_RH computes the relative humidity from the dew point and air temperatures following the formulation of Buck (1981). SEKF_GAIN computes the Kalman Gain.

9.5 1D OI SOIL AND SNOW TEMPERATURE ANALYSIS

The temperature of the first layer of soil (0-7cm) and the first (topmost) snow layer are analysed using a "local" one-dimensional Optimal Interpolation (1D OI) technique as described in Mahfouf (1991) and Douville *et al.* (2001). The analysis increments from the screen-level temperature analysis are used to produce increments for the first layer soil and snow temperatures:

$$\Delta T = c \times (T_{\mathrm{a}} - T_{\mathrm{b}}) \tag{9.24}$$

with T_{a} and T_{b} the analysed and model first-guess temperatures, respectively. The coefficient c providing the analysis increments is:

$$c = (1 - F_1)F_3 \tag{9.25}$$

F_1 and F_3 are empirical functions. F_1 is function of the cosine of the mean solar zenith angle μ_{M}, averaged over the 6-hours before the analysis time given by:

$$F_1 = \frac{1}{2}\{1 + \tanh[\lambda(\mu_{\mathrm{M}} - 0.5)]\} \quad \lambda = 7 \tag{9.26}$$

- CNT0
  - SEKF1
    - SEKF2
      - SUSEKF
      - CNT3
        - CNT4
        - OBSGEN
        - SEKF_PREP_ASCAT
        - SEKF_GETEDA_JACOBIANS
        - SM_EKF_MAIN
          - SEKF_MAGN_RH
          - SEKF_GAIN
          - SEKF_COSTF
        - SM_WRITE

Figure 9.4 *Program organisation for the simplified EKF soil moisture analysis (NCONF=302).*
.

The empirical function F_3 reduces increments over mountainous areas so that

$$
F_3 = \begin{cases}
0 & Z > Z_{\max} \\
\left(\dfrac{Z - Z_{\max}}{Z_{\min} - Z_{\max}} \right)^2 & Z_{\min} < Z < Z_{\max} \\
1 & Z < Z_{\min}
\end{cases}
\tag{9.27}
$$

where Z is the model orography, $Z_{\min} = 500$ m and $Z_{\max} = 3000$ m.

The coefficient c is such that soil and snow temperatures are more effective during night and in winter, when the temperature errors are less likely to be related to soil moisture. In the 12-hour 4D-Var configuration, the snow and soil temperature analysis is performed twice during the assimilation window and the sum of the increments is added to the background values at analysis time. The other soil and snow layers are not analysed, maintaining their background values.

Chapter 10

Analysis of sea-ice concentration and sea-surface temperature

10.1 INTRODUCTION

The analyses of sea-ice concentration (CI) and sea-surface temperature (SST) are a combination of the OCEAN5 analysis of SST and CI, and interpolations to the model grid of daily global datasets provided by the UK Met Office. The OSTIA product is described by Donlon *et al.* (2012) and is available in the ECMWF archives form 20071020 onwards. Its sea ice component is taken from the Satellite Application Facility on Ocean and Sea Ice (OSI-SAF) product. The lake information in the OSTIA product is produced using a lake surface water temperature (LSWT) analysis described by Fiedler *et al.* (2014) Lake ice is added to the OSTIA ice field using a combination of NCEP (National Centers for Environmental Prediction) SSM/I ice concentration and a temperature threshold based on the LSWT analysis itself. The OCEAN5 analysis is described by Zuo *et al.* (2018).

10.2 ECMWF RE-SAMPLING OF OSTIA TO MODEL GRID

The OSTIA product comes oversampled (i. e. at higher resolution than there is real information) at 0.05 degree resolution, which is then area averaged over the boxes of the models reduced Gaussian grid. Product checks are made so that the values lie within realistic values. The minimum ice concentration that is accepted by the analysis is 20%. The SST in a grid box containing ice is now dependent on the ice concentration rather than just being set to the freezing temperature of salt water (T_{fzsw}=271.46K) The default option (T0220) is a smooth transition set by a weighting function W rather than threshold step change when ice was present, which is what was used before. The sea-surface temperature is adjusted at grid points containing sea ice weighted by the sea ice concentration such that

$$SST = W * T_{fzsw} + (1 - W) * \max(OSTIA\_SST, T_{fzsw}) \tag{10.1}$$

where

$$W = 0.5(\tanh((\text{sea ice conc} - A)B) + 1) \tag{10.2}$$

with the parameters $A = 0.2$ and $B = 20$.

Other W options coded are T0310 where W is the same as 10.2 but $A = 0.3$ and $B = 10$. The other options are that the SST weighting is linear for concentration up to 0.2 (L20) and 0.5(L50) or the SST is just changed to the freezing temperature of salt water whenever sea ice is present (orig).

10.3 INTERPOLATION OF OCEAN5 FIELDS TO ATMOSPHERIC GRID

The OCEAN5 fields are on a tri-polar grid, with a nominal $0.25°$ resolution, referred to as the ORCA_025 grid. A weight based interpolation scheme is used such that the field f at an atmospheric grid point x can be written as

$$f(x) = \sum_{i \in \text{loc}} w_i f(x_i), \tag{10.3}$$

where loc refers to some local neighbourhood of the atmospheric grid point x that contains the ocean grid points x_i a distance d_i from x. For a given length-scale ℓ the w_i are weights given by the formula

$$w_i = \frac{e^{-\frac{d_i^2}{2\ell}}}{\sum_{i \in \text{loc}} e^{-\frac{d_i^2}{2\ell}}}. \tag{10.4}$$

10.4 COMBINING OCEAN5 WITH PROCESSED OSTIA DATA

Let $\mathcal{A}$ denote either the SST or CI field as a result of the processing listed in Section 10.2. Let $\mathcal{I}$ denote either the SST of CI field as a result of the *interpolation* of OCEAN5 fields as detailed in 10.3.

OCEAN5 data is only available within the ORCA_025 domain that does not include most lakes. The OCEAN5 data is combined with the field coming from the atmospheric analysis such that

$$\mathcal{O} = \mathcal{A}(\text{land,lakes}) + \mathcal{I}(\text{ocean,seas}).$$

Then a mask is applied to choose which of the atmospheric or ocean fields to use in the final analysis, i.e. apply mask m so that final field $\mathcal{F}$ satisfies

$$\mathcal{F} = m\mathcal{A} + (1 - m)\mathcal{O}.$$

For sea ice the mask m is used to take OSTIA sea ice south of the extent of the ORCA_025 grid. Hence

$$m = \begin{cases} 0 & \text{lat} > -75.5 \\ 1 - 2(\text{lat} + 76) & -76 \leq \text{lat} \leq -75.5 \\ 1 & \text{lat} < -76 \end{cases}. \tag{10.5}$$

For SST the mask m is used to take OSTIA in the extra tropics and OCEAN5 in the tropics:

$$m = \begin{cases} 0 & |\text{lat}| < 20 \\ \frac{1}{5}(|\text{lat}| - 20) & 20 \leq |\text{lat}| \leq 25 \\ 1 & |\text{lat}| > 25 \end{cases}. \tag{10.6}$$

Chapter 11

Data flow

Table of contents

11.1 NOTATION

The following environment variables, which are used in the same way in the data assimilation scripts, are referred to in this chapter.

Table 11.1 *Definition of environment variables.*

Variable	Meaning	Default value
${DATA}	Data directory for invariant files	
${GTYPE}	Gaussian grid type	l_2 (ie linear reduced Gaussian grid)
${IFS_CYCLE}	IFS cycle name	CY36R1
${LEVELS}	Number of vertical model levels	91
${MM}	Month	
${RESOL}	Spectral truncation	1279
${starttime}	Start of 4D-Var window as yyyymmddhh	
${WDIR}	Work directory (1 for each cycle)	

In this chapter, the notation illustrated in Fig. 11.1 is used in diagrams to distinguish between computation steps and data sets.

Computation steps

Data sets
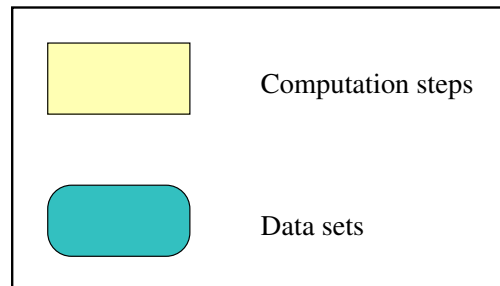
Figure 11.1 *Notation.*

11.2 DATA ASSIMILATION CYCLING

Fig. 11.2 gives an overview of the data flow through the data assimilation system with the operational early-delivery configuration. The 12-hour 4D-Var analyses are run with a delayed cut-off time, in order to use the maximum possible number of observations. The 0000 UTC analysis uses observations from the time window 2101–0900 UTC, while the 1200 UTC analysis uses observations in the window 0901–2100 UTC. The extraction tasks for observations in the periods 2101–0300 UTC and 0301–0900 UTC are run at 1345 and 1400 UTC respectively, while the extraction tasks for the observations in the periods 0901–1500 UTC and 1501–2100 UTC are run at 0145 and 0200 UTC. The 0000 UTC 12-hour 4D-Var analysis generates two sets of analysed fields, at 0000 and 0600 UTC. A separate surface analysis is run every 6 hours. The final analysis is a combination of the fields from 4D-Var and from the surface analysis. The first guess for the 0000 UTC 12-hour 4D-Var analysis is the three-hour forecast from the previous day's 1800 UTC delayed cut-off analysis. The first guess for the 1200 UTC 12-hour 4D-Var
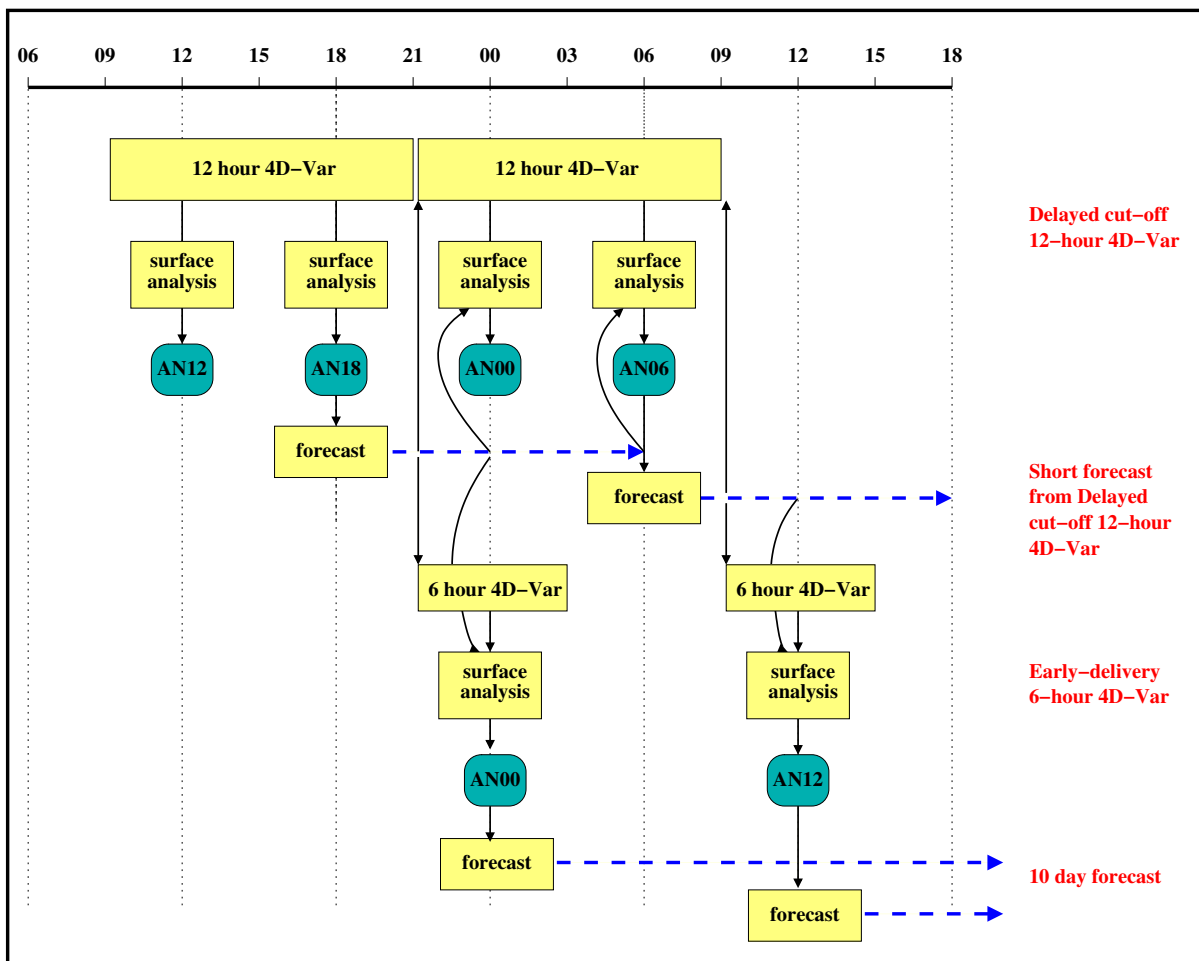
Figure 11.2 *Data assimilation cycling with the Early Delivery configuration.*

delayed cut-off analysis is the three-hour forecast from the 0600 UTC analysis. It is these 12-hour 4D-Var delayed cut-off analyses that propagate information forwards from day to day.

The early-delivery analyses do not propagate information from cycle to cycle. Each analysis is reinitialized with the best available model fields from the delayed cut-off assimilation. The 0000 UTC early-delivery analysis is a 6-hour 4D-Var analysis that uses observations in the time window 2101–0300 UTC. The cut-off time is 0400 UTC, and any observations which arrive after this time are not used by the early-delivery analysis. However, if they arrive by 1400 UTC, they can still be used by the delayed cut-off 12-hour 4D-Var 0000 UTC analysis. The first guess for the 0000 UTC early-delivery analysis is the three-hour forecast from the previous day's 1800 UTC delayed cut-off analysis.

The early-delivery 1200 UTC analysis is a 6-hour 4D-Var analysis that uses observations in the time window 0901–1500 UTC, with a cut-off time of 1600 UTC. Its first guess is the three-hour forecast from the 0600 UTC delayed cut-off analysis.

11.3 OVERVIEW OF 4D-VAR DATA FLOW

Fig. 11.3 gives an overview of the data input to and output from 4D-Var. There are three types of input data.

(i) GRIB fields from the Fields Data Base (FDB). GRIB is a World Meteorological Organisation (WMO) standard format for the representation of General Regularly-distributed Information in Binary.
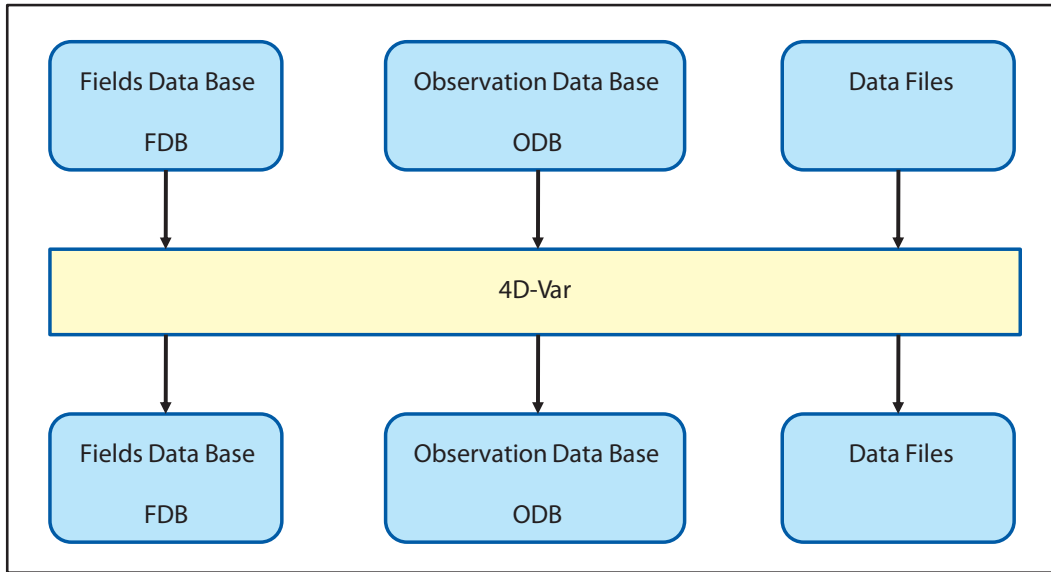
Figure 11.3 *4D-Var data flow.*

The GRIB code is described at http://www.ecmwf.int/products/data/software/grib.html. The background fields, forecast errors and model errors are read from the Fields Data Base.
(ii)   Observations from the Observation Data Base (ODB).
(iii)   Cycling variational bias correction and emissivity files, plus other data files.

Output data from 4D-Var is in three forms.

(i)   GRIB fields in the Fields Data Base, eg analysis fields, error fields.
(ii)   updates to the ODB, including departures of observations from the background and the analysis and quality control information.
(iii)   updated variational bias correction and emissivity files.

11.4   INPUT GRIB FIELDS

The following files in the work directory, $WDIR, contain GRIB format fields which have been extracted from the Fields Data Base. Fields of type 'fc' are taken from the forecast from the previous cycle's analysis, and are valid at the start of the 4D-Var window. Fields of type 'an' are taken from the previous analysis. The spectral orography is taken from the climatology file of the appropriate resolution, to ensure that the orography is not changed by encoding into and decoding from GRIB. Fields of type 'ef', forecast error in radiance space, and 'me', model error, were output from the previous cycle's 4D-Var analysis. Fields of type 'ses', ensmeble based filtered uppera air forecast errors, are output from the attached ensemble data assimilation run.

Files which are needed as input to the coupled wave model also have the stream defined. If stream = DA, then the fields were output from the atmospheric model, and the GRIB codes are defined in (ECMWF local table 2, Version 128). If stream = WAVE, then the fields were output from the wave model and the GRIB codes are defined in (ECMWF local table 2, Version 140).

11.4.1 reftrajshml

Table 11.2 *Background, spherical harmonics, model levels.*

Code	Name	Description	Units	Levels	Type
129	Z	Orography (geopotential)	$m^2 s^{-2}$	1	climate
130	T	Temperature	K	1-$LEVELS	fc
138	VO	Vorticity	s^{-1}	1-$LEVELS	fc
152	LNSP	Logarithm of surface pressure		1	fc
155	D	Divergence	s^{-1}	1-$LEVELS	fc

11.4.2 reftrajggml

Table 11.3 *Background, Gaussian grid, model levels.*

Code	Name	Description	Units	Levels	Type
133	Q	Specific humidity	$kg\,kg^{-1}$	1-$LEVELS	fc
203	O3	Ozone mass mixing ratio	$kg\,kg^{-1}$	1-$LEVELS	fc
75	CRWC	Cloud rain water content	$kg\,kg^{-1}$	1-$LEVELS	fc
76	CSWC	Cloud snow water content	$kg\,kg^{-1}$	1-$LEVELS	fc
246	CLWC	Cloud liquid water content	$kg\,kg^{-1}$	1-$LEVELS	fc
247	CIWC	Cloud ice water content	$kg\,kg^{-1}$	1-$LEVELS	fc
248	CC	Cloud cover	(0-1)	1-$LEVELS	fc

11.4.3 reftrajggsfc

Table 11.4 *Surface fields, Gaussian grid, background (variable fields) and analysis (invariant fields).*

Code	Name	Description	Units	Type
031	CI	Sea-ice cover	(0-1)	fc
032	ASN	Snow albedo	(0-1)	fc
033	RSN	Snow density	kg m^{-3}	fc
034	SST	Sea surface temperature	K	fc
035	ISTL1	Ice surface temperature, layer 1	K	fc
036	ISTL2	Ice surface temperature, layer 2	K	fc
037	ISTL3	Ice surface temperature, layer 3	K	fc
038	ISTL4	Ice surface temperature, layer 4	K	fc
039	SWVL1	Volumetric soil water, layer 1	$\text{m}^3 \text{ m}^{-3}$	fc
040	SWVL2	Volumetric soil water, layer 2	$\text{m}^3 \text{ m}^{-3}$	fc
041	SWVL3	Volumetric soil water, layer 3	$\text{m}^3 \text{ m}^{-3}$	fc
042	SWVL4	Volumetric soil water, layer 4	$\text{m}^3 \text{ m}^{-3}$	fc
139	STL1	Soil temperature level 1	K	fc
141	SD	Snow depth	m of water equivalent	fc
148	CHNK	Charnock parameter		fc
170	STL2	Soil temperature level 2	K	fc
183	STL3	Soil temperature level 3	K	fc
198	SRC	Skin reservoir content	m of water	fc
235	SKT	Skin temperature	K	fc
236	STL4	Soil temperature level 4	K	fc
238	TSN	Temperature of snow layer	K	fc
228131	U10N	Neutral wind u component at 10m	m s^{-1}	fc
228132	V10N	Neutral wind v component at 10m	m s^{-1}	fc
244	FSR	Surface roughness	m	fc
245	FLSR	Logarithm of surface roughness	m	fc
015	ALUVP	UV visible albedo for direct radiation	(0-1)	an
016	ALUVD	UV visible albedo for diffuse radiation	(0-1)	an
017	ALNIP	Near IR albedo for direct radiation	(0-1)	an
018	ALNIP	Near IR albedo for diffuse radiation	(0-1)	an
027	CVL	Low vegetation cover	(0-1)	an
028	CVH	High vegetation cover	(0-1)	an
029	TVL	Type of low vegetation	Table index	an
030	TVH	Type of high vegetation	Table index	an
043	SLT	Soil type		an
066	LAI_LV	Leaf area index, low vegetation	$\text{m}^2 \text{ m}^{-2}$	an
067	LAI_HV	Leaf area index, high vegetation	$\text{m}^2 \text{ m}^{-2}$	an
074	SDFOR	Standard deviation of filtered sub-gridscale orography		an
160	SDOR	Standard deviation of orography		an
161	ISOR	Anisotrophy of sub-gridscale orography		an
162	ANOR	Angle of sub-gridscale orography	rad	an
163	SLOR	Slope of sub-gridscale orography		an
172	LSM	Land-sea mask	(0, 1)	an
173	SR	Surface roughness	m	an
174	ALB	Albedo	(0-1)	an
234	LSRH	Logarithm of surface roughness		an

11.4.4 errgrib

Table 11.5 *Background errors, model levels, Gaussian grid.*

Code	Name	Description	Units	Levels	Type
130	T	Temperature	K	1-$LEVELS	ses
131	U	*u* velocity	m s^{-1}	1-$LEVELS	ses
132	V	*v* velocity	m s^{-1}	1-$LEVELS	ses
133	Q	Specific humidity	kg kg^{-1}	1-$LEVELS	ses
138	VO	Vorticity	s^{-1}	1-$LEVELS	ses
152	LNSP	Logarithm of surface pressure		1	ses
156	GH	Geopotential height	m	1-$LEVELS	ses
157	R	Relative humidity	%	1-$LEVELS	ses
194	BTMP	Brightness temperature	K	channels 1-54	ef
203	O3	Ozone mass mixing ratio	kg kg^{-1}	1-$LEVELS	ses

11.4.5 spmoderr_bg_01

Table 11.6 *Model error, spherical harmonics, model levels.*

Code	Name	Description	Units	Levels	Type
130	T	Temperature	K	1-$LEVELS	me
138	VO	Vorticity	s^{-1}	1-$LEVELS	me
152	LNSP	Logarithm of surface pressure		1	me
155	D	Divergence	s^{-1}	1-$LEVELS	me

11.4.6 wam_specwavein

Table 11.7 *Background, surface, regular latitude/longitude grid, input for wave model.*

Code	Name	Description	Units	Type	Stream
251	2DFD	2D wave spectra	m^2 s radian^{-1}	fc	WAVE

11.4.7 wam_cdwavein

Table 11.8 *Background, surface, regular latitude/longitude grid, input for wave model.*

Code	Name	Description	Units	Type	Stream
233	CDWW	Coefficient of drag with waves		fc	WAVE

11.4.8 wam_uwavein

Table 11.9 *Background, surface, regular latitude/longitude grid, input for wave model.*

Code	Name	Description	Units	Type	Stream
245	WIND	10 metre wind speed	m s^{-1}	fc	WAVE

11.4.9 wam_sfcwindin

Table 11.10 *Background, surface, Gaussian grid, input for wave model.*

Code	Name	Description	Units	Type	Stream
031	CI	Sea-ice cover	(0-1)	fc	DA
165	10U	10 metre U wind component	m s^{-1}	fc	DA
166	10V	10 metre U wind component	m s^{-1}	fc	DA

11.5 INPUT OBSERVATION DATA

Observations are read into 4D-Var from the Observation Data Base (ODB). The observation processing is described in more detail in the ODB documentation (file:///home/rd/mps/public/ugodb.pdf).

11.6 INPUT DATA FILES

Input data files can be split into two categories, invariant and date-dependent. For the invariant files, a single copy is used for the lifetime of the experiment or the operational suite. The file is copied or linked into the experiment's ${DATA} directory at start-up time in task datalinks.

Some files, such as blacklists and bias files, are date-dependent. In the operational suite, the blacklist can be changed at short notice if, for example, a satellite channel fails or a new data source arrives which has to be passively monitored to assess its quality before it can be used actively. Date-dependent files are copied to the ${WDIR} directory in task vardata at the beginning of each data assimilation cycle.

11.6.1 Invariant data files

- ${DATA}/an/cmod.table.ieee – scatterometer coefficients
- ${DATA}/an/external_bl_mon_monit.b – external blacklist file
- ${DATA}/an/moderr.cov – model error covariances for weak-constraint 4D-Var
- ${DATA}/an/neuroflux_l${LEVELS} – extended linearized longwave radiation
- ${DATA}/an/radjacobian_l${LEVELS} – extended linearized longwave radiation
- ${DATA}/an/rs_bias_T_table1 – radiosonde temperature bias correction coefficients
- ${DATA}/an/rs_bias_T_table2 – radiosonde temperature bias correction coefficients
- ${DATA}/an/rs_bias_T_table3 – radiosonde temperature bias correction coefficients
- ${DATA}/an/rszcoef_fmt – radiosonde height observation error correlation coefficients
- ${DATA}/an/ship_anemometer_heights – ship anemometer heights
- ${DATA}/an/stabal96.bal – background error balance parameters
- ${DATA}/an/stabal96.cv – background error correlations
- ${DATA}/an/stdev_of_climate/m${MM} – climatological standard deviations for use in error growth model
- ${DATA}/an/wavelet_T${RESOLINC_n}_L${LEVELS}.cv – wavelet Jb background error covariances
- ${DATA}/climate/${RESOL}${GTYPE}/O3CHEM${MM} – monthly ozone chemistry climate files
- ${DATA}/ifs/namelist_${IFS_CYCLE} – an empty copy of all the IFS namelists
- ${DATA}/ifs/rtable${GTYPE}${RESOL} – namelist NAMRGRI, defining the number of points on each row of the Gaussian grid
- ${DATA}/ifs/vtable_L${LEVELS} – namelist NAMVV1, defining the hybrid vertical coordinate level coefficients
- ${DATA}/sat/mwave_error_${platform}_${instrument}.dat – microwave error files
- ${DATA}/sat/AIRS_CLDDET.NL – list of channels to be used for AIRS with cloud detection
- ${DATA}/sat/amv_bias_info – atmospheric motion vector information
- ${DATA}/sat/bcor_reo3 – ozone bias correction file
- ${DATA}/sat/chanspec_noaa – noaa channel specification file

- ${DATA}/sat/cstlim_noaa – noaa cost limit file
- ${DATA}/sat/filbiaso_[ssmi|tmi] – SSMI/TMI 1D-Var bias file
- ${DATA}/sat/filcmix_[ssmi|tmi] – SSMI/TMI 1D-Var coefficient file
- ${DATA}/sat/filcovb – SSMI 1D-Var coefficient file
- ${DATA}/sat/filcovo_[ssmi|tmi] – SSMI/TMI 1D-Var coefficient file
- ${DATA}/sat/filcwat_[ssmi|tmi] – SSMI/TMI 1D-Var coefficient file
- ${DATA}/sat/iasichannels – list of channels to be used for IASI
- ${DATA}/sat/IASI_CLDDET.NL – list of channels to be used for IASI with cloud detection
- ${DATA}/sat/mask_asc – radiosonde mask
- ${DATA}/sat/mietable_dmsp_ssmi – optical properties of hydrometeors used in the scattering calculations for 1D-Var rain
- ${DATA}/sat/rmtberr_[noaa|airs|iasi] – measurement error files
- ${DATA}/sat/rttov/rtcoef_${platform}_${instrument}.dat – RTTOV radiative transfer coefficient files, for all current and historic satellite platforms and instruments
- ${DATA}/sat/scanbias.ssmi – SSMI scan bias coefficients
- ${DATA}/sat/[sigmab|correl] – background error files
- ${DATA}/sat/ssmi_tovs1c_buf – ATOVS BUFR template for conversion of SSMI data to 1c-radiances
- ${DATA}/sat/thin_reo3 – ozone thinning file
- ${DATA}/scat/mle_norm.dat – QuikSCAT look-up tables
- ${DATA}/scat/nscat2.noise – QuikSCAT noise look-up tables
- ${DATA}/scat/nscat2.table – QuikSCAT GMF look-up tables
- ${DATA}/scat/qscat1.table – QuikSCAT GMF look-up tables

11.6.2 Date-dependent data files

- ${WDIR}/bl_data_sel – data selection blacklist
- ${WDIR}/monthly_bl_mon_monit.b – monthly monitoring blacklist
- ${WDIR}/VARBC.cycle.prev – variational bias correction file from previous cycle
- ${WDIR}/emiskf.cycle.prev.tar – emissivity file from previous cycle
- ${WDIR}/vardir/erss0 – ERS1 scatterometer sigma0 bias correction
- ${WDIR}/vardir/erssp – ERS1 scatterometer speed bias correction
- ${WDIR}/vardir/ascats0 – ASCAT scatterometer sigma0 bias correction
- ${WDIR}/vardir/ascatsp – ASCAT scatterometer speed bias correction

11.7 OUTPUT GRIB FIELDS

Fields of type '4v' (4D-Var analysis), 'an' (analysis), 'ea' (analysis errors), 'ef' (forecast errors) and 'me' (model error) are written in GRIB code to the Fields Data Base from 4D-Var. Fig. 11.4 illustrates the difference between type '4v' and type 'an' analysis fields. For type '4v' fields, the analysis increment from the final minimization is interpolated back to high resolution and added to the penultimate high resolution trajectory at its starting point. Analysis fields output from the final high resolution non-linear trajectory are of type '4v', with a base time at the start of the trajectory and a step corresponding to the number of hours into the trajectory. So, for example, for the 1200 UTC 12-hour 4D-Var for date yyyymmdd, with an observation window from 0300 to 1500 UTC, the 4D-Var analysis at 1200 UTC is stored in the Fields Data Base and MARS with parameters:

$$\text{date} = \text{yyyymmdd, hour} = 03, \text{step} = 9, \text{type} = 4v$$

For type 'an' fields, the increment from the final minimization is added to the penultimate high resolution trajectory at the actual analysis time. Fields from the surface analysis are combined with fields from 4D-Var to give the full analysis. For the 1200 UTC 12-hour 4D-Var for date yyyymmdd, with an observation window from 0300 to 1500 UTC, the type 'an' analysis at 1200 UTC is stored in the Fields Data Base and MARS with parameters:

$$\text{date} = \text{yyyymmdd, hour} = 12, \text{step} = 0, \text{type} = \text{an}$$
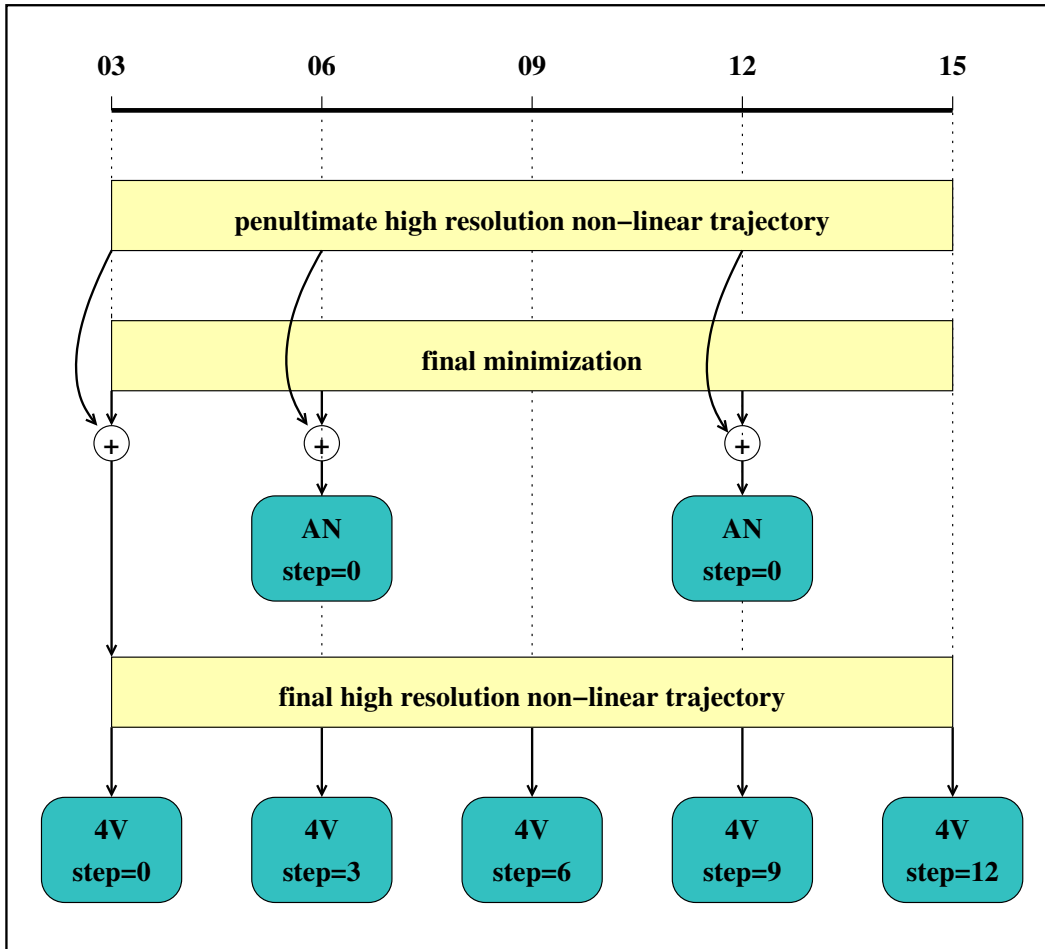
Figure 11.4 *Type 'an' and type '4v' fields written from 4D-Var to the Fields Data Base.*

Output analysis fields, of type '4v' and 'an', can be generated on model levels, pressure levels and isentropic surfaces. Namelist NAMFPC controls the content of the post-processing, and there is a wide selection of fields that can be produced. The IFS determines internally whether fields should be generated in spectral or grid-point form. Described below is only the list of fields that are needed as input for the next forecast in order to cycle the data assimilation forward in time. The forecast starts from fields of type 'an'. Some of the surface fields are generated by the surface analysis jobs, which run at the same time as 4D-Var. It is important that these fields are excluded from the type 'an' post-processing of 4D-Var, so that they cannot overwrite the surface analysis fields.

11.7.1   Output type 'an' model level spectral fields

Table 11.11 *Output type 'an' model level spectral fields.*

Code	Name	Description	Units	Levels
130	T	Temperature	K	1-$LEVELS
135	W	Vertical velocity	$m\,s^{-1}$	1-$LEVELS
138	VO	Vorticity	s^{-1}	1-$LEVELS
152	LNSP	Logarithm of surface pressure		1
155	D	Divergence	s^{-1}	1-$LEVELS

11.7.2 Output type 'an' model level Gaussian grid-point fields

Table 11.12 *Output type 'an' model level Gaussian grid-point fields.*

Code	Name	Description	Units	Levels
75	CRWC	Cloud rain water content	kg kg^{-1}	1-$LEVELS
76	CSWC	Cloud snow water content	kg kg^{-1}	1-$LEVELS
133	Q	Specific humidity	kg kg^{-1}	1-$LEVELS
203	O3	Ozone mass mixing ratio	kg kg^{-1}	1-$LEVELS
246	CLWC	Cloud liquid water content	kg kg^{-1}	1-$LEVELS
247	CIWC	Cloud ice water content	kg kg^{-1}	1-$LEVELS
248	CC	Cloud cover	(0-1)	1-$LEVELS

11.7.3 Output type 'an' surface Gaussian grid-point fields

Table 11.13 *Output type 'an' surface Gaussian grid-point fields.*

Code	Name	Description	Units
015	ALUVP	UV visible albedo for direct radiation	(0-1)
016	ALUVD	UV visible albedo for diffuse radiation	(0-1)
017	ALNIP	Near IR albedo for direct radiation	(0-1)
018	ALNIP	Near IR albedo for diffuse radiation	(0-1)
027	CVL	Low vegetation cover	(0-1)
028	CVH	High vegetation cover	(0-1)
029	TVL	Type of low vegetation	Table index
030	TVH	Type of high vegetation	Table index
032	ASN	Snow albedo	(0-1)
035	ISTL1	Ice surface temperature, layer 1	K
036	ISTL2	Ice surface temperature, layer 2	K
037	ISTL3	Ice surface temperature, layer 3	K
038	ISTL4	Ice surface temperature, layer 4	K
042	SWVL4	Volumetric soil water, layer 4	$m^3\,m^{-3}$
043	SLT	Soil type	
066	LAI-LV	Leaf area index, low vegetation	$m^2\,m^{-2}$
067	LAI-HV	Leaf area index, high vegetation	$m^2\,m^{-2}$
074	SDFOR	Standard deviation of filtered subgrid orography	
128	BV	Budget values	-
129	Z	Orography (geopotential)	$m^2\,s^{-2}$
134	SP	Surface pressure	Pa
136	TCW	Total column water	$kg\,m^{-2}$
137	TCWV	Total column water vapor	$kg\,m^{-2}$
148	CHNK	Charnock parameter	
151	MSL	Mean sea level pressure	Pa
160	SDOR	Standard deviation of orography	
161	ISOR	Anisotrophy of sub-gridscale orography	
162	ANOR	Angle of sub-gridscale orography	rad
163	SLOR	Slope of sub-gridscale orography	
164	TCC	Total cloud cover	(0-1)
165	10U	10 metre U wind component	$m\,s^{-1}$
166	10V	10 metre V wind component	$m\,s^{-1}$
172	LSM	Land-sea mask	(0,1)
173	SR	Surface roughness	m
174	ALB	Albedo	(0-1)
186	LCC	Low cloud cover	(0-1)
187	MCC	Medium cloud cover	(0-1)
188	HCC	High cloud cover	(0-1)
198	SRC	Skin reservoir content	m of water
206	TCO3	Total column ozone content	$kg\,m^{-2}$
234	LSRH	Logarithm of surface roughness	
235	SKT	Skin temperature	K
236	STL4	Soil temperature level 4	K
228021	FDIR	Total-sky direct solar radiation at surface	$J\,m^{-2}$
228022	CDIR	Clear-sky direct solar radiation at surface	$J\,m^{-2}$
228246	100U	Wind u component at 100 m	$m\,s^{-1}$
228247	100V	Wind v component at 100 m	$m\,s^{-1}$

11.7.4 Output type 'an' wave model fields

The output wave model fields are on a regular latitude/longitude grid. They are identified by stream 'WAVE' and are encoded with GRIB codes defined in ECMWF local table 2, Version 140.

Table 11.14 *Output type 'an' wave model fields.*

Code	Name	Description	Units
251	2DFD	2D wave spectra	m^2 s radian^{-1}
233	CDWW	Coefficient of drag with waves	
245	WIND	10 metre wind speed	m s^{-1}

11.7.5 Output error fields

The forecast errors output from one cycle are used as the background errors input to the next cycle, and their content is described in Section 11.4.4 above. The analysis errors contain similar fields, but are of type 'ea'. The analysis errors are used to calculate the perturbations for the Ensemble Prediction System.

11.7.6 Output model error fields

The model error fields which are output from the analysis are input to the next forecast. Their content is described in Section 11.4.5 above.

11.8 OUTPUT OBSERVATION DATA

Departures of observations from the background and the analysis, and quality information are written to the Observation Data Base (ODB). The observation processing is described in more detail in Part I.

11.9 OUTPUT DATA FILES

- ${WDIR}/VARBC.cycle – updated variational bias correction file
- ${WDIR}/emiskf.cycle.tar – updated emissivity file

11.10 SEA SURFACE TEMPERATURE ANALYSIS

The sea surface temperature analysis is done every 6 hours.

11.10.1 Input GRIB fields on model grid

These are extracted from the Fields Data Base. The background fields, of type 'fc', are taken from the forecast from the previous 4D-Var analysis time. The persistence analysis, of type 'an', is taken from the previous sea surface temperature analysis time, 6 hours earlier (which is not necessarily a 4D-Var analysis time). In the table below, 'T' is used to denote the analysis time, and 'T-6' is used to denote the persistence analysis time.

Table 11.15 *Input GRIB fields on model grid.*

Code	Name	Description	Units	Type	Time
031	CI	Sea-ice cover	(0-1)	fc	T
034	SST	Sea surface temperature	K	fc	T
034	SST	Sea surface temperature	K	an	T-6

11.10.2   Input OSTIA field

- $WDIR/surf_anal/ostia – OSTIA sea surface temperature and sea ice concentration fields, encoded as BUFR data

11.10.3   Input data files

- $DATA/sst/sst_clim – surface air temperature monthly climatology, reduced to mean-sea level (0.5 × 0.5 degree regular latitude/longitude grid, 720 × 361 points)
- $DATA/sst/ice_clim – ice monthly climatology
- ${DATA}/climate/${RESOL}${GTYPE}/lsmoro – land/sea mask and orography on model Gaussian grid
- ${DATA}/climate/${RESOL}${GTYPE}/clake – lake mask

11.10.4   Output GRIB fields on model grid

The following fields are written to the Fields Data Base:

Table 11.16 *Output GRIB fields on model grid.*

Code	Name	Description	Units	Type
031	CI	Sea-ice cover	(0-1)	an
034	SST	Sea surface temperature	K	an

11.11   2 METRE TEMPERATURE ANALYSIS

The 2 metre temperature analysis is done every 6 hours.

11.11.1   Input GRIB fields on model grid

These are extracted from the Fields Data Base. The background fields, of type 'fc', are taken from the forecast from the previous 4D-Var analysis time. The invariant fields, of type 'an', are taken from the previous 4D-Var analysis.

Table 11.17 *Input GRIB fields on model grid.*

Code	Name	Description	Units	Type
129	Z	Orography	$m^2\,s^{-2}$	an
172	LSM	Land/sea mask	(0-1)	an
139	STL1	Soil temperature level 1	K	fc
167	2T	2 metre temperature	K	fc
168	2D	2 metre dewpoint temperature	K	fc

11.11.2   Input observation data

Observations are read from the Observation Data Base.

11.11.3   Output GRIB field on model grid

The analysed 2 metre temperature field is written to the Fields Data Base.

Table 11.18 *Output GRIB field on model grid.*

Code	Name	Description	Units	Type
167	2T	2 metre temperature	K	an

11.12 2 METRE RELATIVE HUMIDITY ANALYSIS

The 2 metre relative humidity analysis is done every 6 hours. Although the analysed field is 2 metre relative humidity, the final output product is 2 metre dewpoint temperature. The 2 metre relative humidity analysis cannot start until the 2 metre temperature analysis has completed, since the output from the 2 metre temperature analysis is needed in the computation of the 2 metre dewpoint temperature.

11.12.1 Input GRIB fields on model grid

These are extracted from the Fields Data Base. The background fields, of type 'fc', are taken from the forecast from the previous 4D-Var analysis time. The invariant fields, of type 'an', are taken from the previous 4D-Var analysis.

Table 11.19 *Input GRIB fields on model grid.*

Code	Name	Description	Units	Type
129	Z	Orography	$m^2\,s^{-2}$	an
172	LSM	Land/sea mask	(0-1)	an
139	STL1	Soil temperature level 1	K	fc
167	2T	2 metre temperature	K	fc
168	2D	2 metre dewpoint temperature	K	fc

11.12.2 Input observation data

Observations are read from the Observation Data Base.

11.12.3 Output GRIB field on model grid

The derived 2 metre dewpoint temperature field is written to the Fields Data Base.

Table 11.20 *Output GRIB field on model grid.*

Code	Name	Description	Units	Type
168	2D	2 metre dewpoint temperature	K	an

11.13 SNOW ANALYSIS

The snow analysis is done every 6 hours. It cannot start until the 2 metre temperature analysis has completed, since the 2 metre temperature analysis field is one of the inputs to the snow analysis.

11.13.1 Input GRIB fields on model grid

These are extracted from the Fields Data Base. The background fields, of type 'fc', are taken from the forecast from the previous 4D-Var analysis time. The invariant fields of type 'an', orography and land/sea mask, are taken from the previous 4D-Var analysis. This is denoted T4V in the table below. The persistence snow depth analysis, of type 'an', is taken from the previous snow analysis time, 6 hours earlier (which is not necessarily a 4D-Var analysis time). In the table below, 'T' is used to denote the snow analysis time, and 'T-6' is used to denote the persistence snow analysis time; LevType is used to discriminate between single-layer ("sfc") and multi-layer ("sol") fields.

Table 11.21 *Input GRIB fields on model grid.*

Code	LevType	Name	Description	Units	Type	Time
129	sfc	Z	Orography	$m^2\,s^{-2}$	an	T4V
172	sfc	LSM	Land/sea mask	(0-1)	an	T4V
033	sfc	RSN	Snow density	$kg\,m^{-3}$	fc	T
33	sol	RSN	Multi-layer snow density	$kg\,m^{-3}$	fc	T
228141	sol	SD	Multi-layer snow depth	$kg\,m^{-2}$	fc	T
228141	sol	SD	Multi-layer snow depth	$kg\,m^{-2}$	an	T-6
167	sfc	2T	2 metre temperature	K	an	T

11.13.2 Input data files

- ${DATA}/climate/${RESOL}${GTYPE}/snow – snow depth climatology (m of water equivalent) on model Gaussian grid
- ${DATA}/climate/${RESOL}${GTYPE}/cicecap – on model Gaussian grid
- $WDIR/imssnow – NESDIS snow cover field (0,1) on polar stereographic grid of approximately 25 km resolution. The data is in BUFR format, with triplets of latitude/longitude/snow cover. The NESDIS snow cover field is only used once per day, for the 06Z snow analysis.

11.13.3 Input observation data

Observations are read from the Observation Data Base.

11.13.4 Output GRIB fields on model grid

The following fields are written to the Fields Data Base:

Table 11.22 *Output GRIB fields on model grid.*

Code	LevType	Name	Description	Units	Type
033	sfc	RSN	Snow density	$kg\,m^{-3}$	an
141	sfc	SD	Snow depth	m of water equivalent	an
33	sol	RSN	Multi-layer snow density	$kg\,m^{-3}$	an
228141	sol	SD	Multi-layer snow depth	$kg\,m^{-2}$	an

11.14 SOIL MOISTURE ANALYSIS

The soil moisture analysis is done every 6 hours. It cannot start until the sea surface temperature analysis and the snow analysis have completed.

11.14.1 Input GRIB fields on model grid

These are extracted from the Fields Data Base. The background fields, of type 'fc', are taken from the forecast from the previous 4D-Var analysis time. The analysed fields, of type 'an', are output from the current 4D-Var analysis, the sea surface temperature analysis, the 2 metre temperature analysis, the 2 metre relative humidity analysis or the snow analysis.

Table 11.23 *Input GRIB fields on model grid.*

Code	Name	Description	Units	Type	Origin
039	SWVL1	Volumetric soil water, layer 1	$m^3 m^{-3}$	fc	Forecast
040	SWVL2	Volumetric soil water, layer 2	$m^3 m^{-3}$	fc	Forecast
041	SWVL3	Volumetric soil water, layer 3	$m^3 m^{-3}$	fc	Forecast
133	Q	Specific humidity on lowest model level	$kg\,kg^{-1}$	fc	Forecast
139	STL1	Soil temperature, level 1	K	fc	Forecast
142	LSP	Large scale precipitation	m	fc	Forecast
143	CP	Convective precipitation	m	fc	Forecast
167	2T	2 metre temperature	K	fc	Forecast
168	2D	2 metre dewpoint temperature	K	fc	Forecast
170	STL2	Soil temperature, level 2	K	fc	Forecast
176	SSR	Surface solar radiation	$W\,m^{-2}\,s$	fc	Forecast
183	STL3	Soil temperature level 3	K	fc	Forecast
238	TSN	Temperature of snow layer	K	fc	Forecast
027	CVL	Low vegetation cover	(0-1)	an	4D-Var
028	CVH	High vegetation cover	(0-1)	an	4D-Var
029	TVL	Type of low vegetation	Table index	an	4D-Var
030	TVH	Type of high vegetation	Table index	an	4D-Var
129	Z	Orography	$m^2\,s^{-2}$	an	4D-Var
133	Q	Specific humidity on lowest model level	$kg\,kg^{-1}$	an	4D-Var
141	SD	Snow depth	m of water equivalent	an	Snow analysis
165	10U	10 metre U wind component	$m\,s^{-1}$	an	4D-Var
166	10V	10 metre V wind component	$m\,s^{-1}$	an	4D-Var
167	2T	2 metre temperature	K	an	2 metre temp. anal.
168	2D	2 metre dewpoint temperature	K	an	2 metre rel. hum. anal.
172	LSM	Land-sea mask	$(0, 1)$	an	4D-Var
174	AL	Albedo	(0-1)	an	4D-Var

11.14.2 Output GRIB fields on model grid

The following fields are output from the soil moisture analysis and written to the Fields Data Base. Before being written, the *STL1* (soil temperature level 1) field is manipulated as follows:

(i)   land values are unchanged
(ii) over sea,

$$STL1 = SST \times (1 - CI) + ISTL1 \times CI$$

where

SST = analysed sea surface temperature

CI = analysed sea ice field, which varies between 0 (open water) and 1 (full ice cover)

$ISTL1$ = background soil temperature level 1

Table 11.24 *Output GRIB fields on model grid.*

Code	Name	Description	Units
039	SWVL1	Volumetric soil water, layer 1	$m^3\,m^{-3}$
040	SWVL2	Volumetric soil water, layer 2	$m^3\,m^{-3}$
041	SWVL3	Volumetric soil water, layer 3	$m^3\,m^{-3}$
042	SWVL4	Volumetric soil water, layer 4	$m^3\,m^{-3}$
139	STL1	Soil temperature, level 1	K
170	STL2	Soil temperature, level 2	K
183	STL3	Soil temperature, level 3	K
238	TSN	Temperature of snow layer	K

11.14.3 Invariant climatological fields

The final step of the soil moisture analysis task is to copy the invariant fields from the climatology files to the analysis, after first manipulating the GRIB headers to give values appropriate for the current data assimilation cycle. In this way, it is ensured that invariant fields remain unchanged, without any loss of precision due to repeatedly encoding and decoding GRIB fields.

Table 11.25 *Invariant climatological fields.*

Code	Name	Description	Units
015	ALUVP	UV visible albedo for direct radiation	(0-1)
016	ALUVD	UV visible albedo for diffuse radiation	(0-1)
017	ALNIP	Near IR albedo for direct radiation	(0-1)
018	ALNIP	Near IR albedo for diffuse radiation	(0-1)
027	CVL	Low vegetation cover	(0-1)
028	CVH	High vegetation cover	(0-1)
029	TVL	Type of low vegetation	Table index
030	TVH	Type of high vegetation	Table index
043	SLT	Soil type	
129	Z	Orography	$m^2\,s^{-2}$
160	SDOR	Standard deviation of orography	
161	ISOR	Anisotrophy of sub-gridscale orography	
162	ANOR	Angle of sub-gridscale orography	rad
163	SLOR	Slope of sub-gridscale orography	
172	LSM	Land-sea mask	(0,1)
173	SR	Surface roughness	m
174	ALB	Albedo	(0-1)
234	LSRH	Logarithm of surface roughness	

References

Andersson, E., Beljaars, A., Bidlot, J., Miller, M., Simmons, A. and Thépaut, J. N. (2003). A major new cycle of the IFS: Cycle 25r4. *ECMWF Newsletter No. 97*, pp. 12–20.

Andersson, E., Fisher, M., Munro, R. and McNally, A. (2000). Diagnosis of background errors for radiances and other observable quantities in a variational data assimilation scheme, and the explanation of a case of poor convergence. *Q. J. R. Meteorol. Soc.*, **126**, 1455–1472.

Andersson, E., Haseler, J., Undén, P., Courtier, P., Kelly, G., Vasiljevic, D., Brankovic, C., Cardinali, C., Gaffard, C., Hollingsworth, A., Jakob, C., Janssen, P., Klinker, E., Lanzinger, A., Miller, M., Rabier, F., Simmons, A., Strauss, B., Thépaut, J.-N. and Viterbo, P. (1998). The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part III: Experimental results. *Q. J. R. Meteorol. Soc.*, **124**, 1831–1860.

Andersson, E. and Järvinen, H. (1999). Variational quality control. *Q. J. R. Meteorol. Soc.*, **125**, 697–722.

Baker, N. L. and Daley, R. (2000). Observation and background adjoint sensitivity in the adaptive observation-targeting problem. *Q. J. R. Meteorol. Soc.*, **126**, 1431–1454.

Bechtold, P., Semane, N., Lopez, P., Chaboureau, J.-P., Beljaars, A. and Bormann, N. (2014). Representing equilibrium and non-equilibrium convection in large-scale models. *J. Atmos. Sci.*, **71**, doi 10.1175/JAS–D–13–0163.1.

Benedetti, A. and Fisher, M. (2006). Background error statistics for aerosols. *ECMWF Tech. Memo. No. 489.*

Betts, A. (1997). The parametrization of deep convection: A review. In *Proc. ECMWF Workshop on New Insights and Approaches to Convective Parametrization*, pp. 166–188, Reading, 4–7 November 1996.

Blackadar, A. K. (1962). The vertical distribution of wind and turbulent exchange in a neutral atmosphere. *J. Geophys. Res.*, **67**, 3095–3102.

Bonavita, M., Holm, E., Isaksen, L. and Fisher, M. (2016). The evolution of the ECMWF hybrid data assimilation system. *Q. J. R. Meteorol. Soc.*, **142**, 287–303.

Bonavita, M., Isaksen, L. and Holm, E. (2012). On the use of eda background error variances in the ecmwf 4d-var. *Q. J. R. Meteorol. Soc.*, **138**, 1540–1559.

Bonnel, B., Fouquart, Y., Vanhoutte, J.-C., Fravalo, C. and Rosset, R. (1983). Radiative properties of some african and mid-latitude statocumulus. *Beitr. Phys. Atmos*, **56**, 409–428.

Bormann, N. and M.Bonavita (2013). Spread of the ensemble of data assimilations in radiance space. *ECMWF Tech. Memo. No. 708.*

Bouttier, F. (2001). The development of 12-hourly 4D-Var. *ECMWF Tech. Memo. No. 348.*

Bouttier, F., Derber, J. and Fisher, M. (1997). The 1997 revision of the Jb term in 3D/4D-Var. *ECMWF Tech. Memo. No. 238.*

Brasnett, B. (1999). A global analysis of snow depth for numerical weather prediction. *J. Appl. Meteorol.*, **38**, 726–740.

Buck, A. L. (1981). New equations for computing vapor pressure and enhancement factor. *J. Appl. Meteorol.*, **20**, 1527–1532.

Cardinali, C. (2009). Monitoring the observation impact on the short-range forecast. *Q. J. R. Meteorol. Soc.*, **135**, 239–250.

Courtier, P., Andersson, E., Heckley, W., Pailleux, J., Vasiljevic, D., Hamrud, M., Hollingsworth, A., Rabier, F. and Fisher, M. (1998). The ECMWF implementation of three dimensional variational assimilation (3D-Var). I: Formulation. *Q. J. R. Meteorol. Soc.*, **124**, 1783–1807.

Courtier, P., Thépaut, J.-N. and Hollingsworth, A. (1994). A strategy for operational implementation of 4D-Var, using an incremental approach. *Q. J. R. Meteorol. Soc.*, **120**, 1367–1388.

de Rosnay, P., Balsamo, G., Albergel, C., noz Sabater, J. M. and Isaksen, L. (2014). Initialisation of land surface variables for numerical weather prediction. *Surveys in Geophysics*, **35(3)**, doi: 10.1007/s10712–012–9207–x.

de Rosnay, P., Chiara, G. D. and Mallas, I. (2011a). Use of ascat soil moisture: revised bias correction and test of improved ascat product in ifs cycle 37r2. *ECMWF reasearch Memorandum R43.8/PdR/11100, August 2011*.

de Rosnay, P., Dragosavac, M., Isaksen, L., Andersson, E. and Haseler, J. (2011b). Use of new snow data from sweden in ifs cycle 36r4. *ECMWF Res. Dept. Memo. No. 1139*.

de Rosnay, P., Drusch, M., Balsamo, G., Albergel, C. and Isaksen, L. (2011c). Extended kalman filter soil-moisture analysis in the ifs. *ECMWF No. 127*, pp. 12–16.

de Rosnay, P., drusch, M., Vasiljevic, D., Balsamo, G., Albergel, C. and Isaksen, L. (2013). A simplified extended kalman filter for the global operational soil moisture analysis at ecmwf. *Q. J. R. Meteorol. Soc., doi: 10.1002/qj.2023, 139(674)*, pp. 1199–1213.

de Rosnay, P., Isaksen, L. and M., D. (2015). Snow data assimilation at ecmwf. *ECMWF Newsletter article*, **143**, 26–31.

Dee, D. (2004). Variational bias correction of radiance data in the ECMWF system. In *Proc. of the ECMWF Workshop on Assimilation of High Spectral Resolution Sounders in NWP*, pp. 97–112, Reading, 28 June–1 July, 2004.

Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P. *et al.* (2011). The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Q. J. R. Meteorol. Soc.*, **137**, 553–597.

Dent, D., Hoffmann, G.-R., Janssen, P. A. E. M. and Simmons, A. J. (1997). Use of the Fujitsu VPP700 for weather forecasting at ECMWF. *ECMWF Tech. Memo. No. 235*.

Derber, J. C. and Bouttier, F. (1999). A reformulation of the background error covariance in the ECMWF global data assimilation system. *Tellus*, **51A**, 195–221.

Dethof, A. and Holm, E. (2004). Ozone assimilation in the ERA-40 reanalysis project. *Q. J. R. Meteorol. Soc.*, **130**, 2851–2872.

Donlon, C. J., Martin, M., Stark, J., Roberts-Jones, J., Fiedler, E. and Wimmer, W. (2012). The operational sea surface temperature and sea ice analysis (ostia) system. *Remote Sensing of Environment*, **116**, 140–158.

Douville, H., Viterbo, P., Mahfouf, J.-F. and Beljaars, A. C. M. (2001). Evaluation of the optimum interpolation and nudging techniques for soil moisture analysis using FIFE data. *Mon. Wea. Rev.*, **128**, 1733–1756.

Drusch, M., Scipal, K., de Rosnay, P., Balsamo, G., Andersson, E., Bougeault, P. and Viterbo, P. (2009). Towards a Kalman Filter based soil moisture analysis system for the operational ECMWF Integrated Forecast System. *G. Res. Lett.*, **36**, L10401, doi:10.1029/2009GL037716.

Dubuisson, P. J., Buriez, J. and Fouquart, Y. (1996). High spectral solar radiative transfer in absorbing and scattering media: Application to the satellite simulation. *J. Quant. Spectrosc. Radiat. Transfer*, **55**, 103–126.

Dyer, A. J. (1974). A review of flux-profile relationships. *Boundary-Layer Meteorol.*, **7**, 363–372.

Ebert, E. E. and Curry, J. A. (1992). A parametrization of ice optical properties for climate models. *J. Geophys. Res.*, **97D**, 3831–3836.

Fiedler, E., Martin, M. and Roberts-Jones, J. (2014). An operational analysis of lake surface water temperature. *Tellus A*, **66**(0).

Fisher, M. (1996). The specification of background error variances in the ECMWF variational analysis system. In *Proc. ECMWF Workshop on Non-linear Aspects of Data Assimilation*, pp. 645–652, Reading, 9–11 September 1996.

Fisher, M. (1998). Minimization algorithms for variational data assimilation. In *Proc. ECMWF Seminar on Recent Developments in Numerical Methods for Atmospheric Modelling*, pp. 364–385, Reading, 7–11 September 1998.

Fisher, M. (2003). Background error covariance modelling. In *Proc. ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean*, pp. 45–64, Reading, 8–12 Sept 2003.

Fisher, M. (2004). Generalized frames on the sphere, with application to the background error covariance modelling. In *Proc. ECMWF Seminar on Recent Developments in Numerical Methods for Atmosphereic and Ocean Modelling*, pp. 87–102, Reading, 6–10 September 2004.

Fisher, M. (2006). Wavelet jb - a new way to model the statistics of background errors. *ECMWF Newsletter No. 106*, pp. 23–28.

Fisher, M. and Andersson, E. (2001). Developments in 4D-Var and Kalman Filtering. *ECMWF Tech. Memo. No. 347*.

Fisher, M. and Courtier, P. (1995). Estimating the covariance matrices of analysis and forecast error in variational data assimilation. *ECMWF Tech. Memo. No. 220*.

Fouquart, Y. (1987). Radiative transfer in climate models. In M. E. Schlesinger (Ed.), *Physically Based Modelling and Simulation of Climate and Climate Changes*, pp. 223–284, Kluwer Acad. Publ.

Fouquart, Y. and Bonnel, B. (1980). Computations of solar heating of the earth's atmosphere: A new parametrization. *Beitr. Phys. Atmos.*, **53**, 35–62.

Fritts, D. C. and Nastrom, D. D. (1993). Spectral estimates of gravity wave energy and momentum fluxes. Part I: Energy dissipation, acceleration, and constraints. *J. Atmos. Sci.*, **50**, 3685–3694.

Gauthier, P. and Thépaut, J.-N. (2001). Impact of the digital filter as a weak constraint in the preoperational 4DVAR assimilation system of Meteo-France. *Mon. Wea. Rev.*, **129**, 2089–2102.

Geleyn, J.-F. and Hollingsworth, A. (1997). An economical and analytical method for the interactions between scattering and line absorption of radiation. *Contrib. to Atmos. Phys.*, **52**, 1–16.

Gustafsson, N. (1992). Use of a digital filter as a weak constraint in variational data assimilation. In *Proc. ECMWF Workshop on Variational Data Assimilation with Special Emphasis on 3-dimensional Aspects*, pp. 327–338, Reading, 9–12 November, 1992.

Haseler, J. (2004). Early-delivery suite. *ECMWF Tech. Memo. No. 454*.

Hirahara, S., , Balmaseda, M., de Boisseson, E. and Hersbach, H. (2016). Sea surface temperature and sea ice concentration for era5. *ERA Report Series No. 26*.

Hogström, U. (1988). Non-dimensional wind and temperature profiles in the atmospheric surface layer: A re-evaluation. *Boundary-Layer Meteorol.*, **42**, 55–78.

Hólm, E., Andersson, E., Beljaars, A., Lopez, P., Mahfouf, J.-F., Simmons, A. J. and Thépaut, J.-N. (2002). Assimilation and modeling of the hydrological cycle: ECMWF's status and plans. *ECMWF Tech. Memo. No. 383*.

Ingleby, N. B. and Lorenc, A. C. (1993). Bayesian quality control using multivariate normal distributions. *Q. J. R. Meteorol. Soc.*, **119**, 1195–1225.

Isaksen, L., Bonavita, M., Buizza, R., Fisher, M., Haseler, J. and Raynaud, L. (2010). Ensemble of data assimilations at ECMWF. *ECMWF Tech. Memo. No. 636.*

Jakob, C. and Siebesma, A. P. (2003). A new subcloud model for mass flux convection schemes. influence on triggering, updraught properties and model climate. *Mon. Wea. Rev.*, **131**, 2765–2778.

Janisková, M., Mahfouf, J.-F., Morcrette, J.-J. and Chevallier, F. (2002). Linearized radiation and cloud schemes in the ECMWF model: Development and evaluation. *Q. J. R. Meteorol. Soc.*, **128**, 1505–1527.

Järvinen, H., Andersson, E. and Bouttier, F. (1999). Variational assimilation of time sequences of surface observations with serially correlated errors. *ECMWF Tech. Memo. No. 266*.

Järvinen, H. and Undén, P. (1997). Observation screening and first guess quality control in the ECMWF 3D-Var data assimilation system. *ECMWF Tech. Memo. No. 236*.

Kennedy, J., Reynerand, N., Millington, S. C. and Saunby, M. (2016). The met office hadley centre sea ice and sea-surface temperature data set, version 2, part 2: sea surface temperature analysis. Available online: http://www.metoffice.gov.uk/hadobs/hadisst2/.

Kessler, E. (1969). *On the distribution and continuity of water substance in atmopsheric circulation*, *Meteorological Monographs*, Vol. 10. Am. Meteorol. Soc., Boston, MA.

Klinker, E., Rabier, F., Kelly, G. and Mahfouf, J. F. (2000). The ECMWF operational implementation of four-dimensional variational assimilation. Part I: Experimental results and diagnostics with operational configuration. *Q. J. R. Meteorol. Soc.*, **126**, 1191–1215.

Laloyaux, P., Bonavita, M., Dahoui, M., Farnan, J., Healy, S., Holm, E. and Lang, S. T. K. (2020). Towards an unbiased stratospheric analysis. *Quarterly Journal of the Royal Meteorological Society*, **146**(730), 2392–2409.

Lopez, P. and Moreau, E. (2005). A convection scheme for data assimilation: Description and initial tests. *Q. J. R. Meteorol. Soc.*, **131**, 409–436.

Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Q. J. R. Meteorol. Soc.*, **112**, 1177–1194.

Lorenc, A. C. (1988). Optimal nonlinear objective analysis. *Q. J. R. Meteorol. Soc.*, **114**, 205–240.

Lott, F. and Miller, M. J. (1997). A new subgrid-scale orographic drag parametrization: Its formulation and testing. *Q. J. R. Meteorol. Soc.*, **123**, 101–127.

Louis, J.-F., Tiedtke, M. and Geleyn, J.-F. (1982). A short history of the PBL parametrization at ECMWF. In *Proc. ECMWF Workshop on Planetary Boundary Layer Parameterization*, pp. 59–80, Reading, 25–27 November, 1981.

Lynch, P. (1993). Digital Filters for Numerical Weather Prediction. *HIRLAM Technical Report No. 10*.

Lynch, P. (1997). The Dolph-Chebyshev Window: A simple optimal filter. *Mon. Wea. Rev.*, **125**, 655–660.

Machenhauer, B. (1977). On the dynamics of gravity oscilations in a shallow water model, with application to normal mode initialization. *Contrib. Atmos. Phys.*, **50**, 253–271.

Mahfouf, J.-F. (1991). Analysis of soil moisture from near surface parameters: A feasibility study. *J. Appl. Meteorol.*, **30**, 1534–1547.

Mahfouf, J.-F. (1999). Influence of physical processes on the tangent-linear approximation. *Tellus*, **51A**, 147–166.

Mahfouf, J. F. and Rabier, F. (2000). The ECMWF operational implementation of four-dimensional variational assimilation. Part I: Experimental results with improved physics. *Q. J. R. Meteorol. Soc.*, **126**, 1171–1190.

McNally, A. P., Andersson, E., Kelly, G. and Saunders, R. W. (1999). The use of raw TOVS/ATOVS radiances in the ECMWF 4D-Var assimilation system. *ECMWF Newsletter No. 83*, pp. 2–7.

McNally, A. P., Bonavita, M. and Thépaut, J.-N. (2014). The role of satellite data in the forecasting of hurricane sandy. *Mon. Wea. Rev.*, **142**, 634–646.

Mlawer, E., Taubman, S. J., Brown, P. D., Ianoco, M. and Clough, S. A. (1997). Radiative transfer for inhomogeneous atmospheres: RRTM a validated correlated-k model for the longwave. *J. Geophys. Res.*, **102**, 16663–16682.

Morcrette, J.-J. (1989). Description of the radiation scheme in the ECMWF operational weather forecast model. *ECMWF Tech. Memo. No. 165*.

Morcrette, J.-J. (1991). Radiation and cloud radiative properties in the ECMWF operational forecast model. *J. Geophys. Res.*, **96D**, 9121–9132.

Morcrette, J.-J. (2000). On the effects of the temporal and spatial sampling of radiation fields on the ECMWF forecasts and analyses. *Mon .Wea. Rev.*, **128**, 876–887.

Morcrette, J.-J., Smith, L. and Fouquart, Y. (1986). Pressure and temperature dependence of absorption in longwave radiation parametrizations. *Beitr. Phys. Atmos.*, **59**, 455–469.

Nordeng, T.-E. (1994). Extended versions of the convection parametrization scheme at ECMWF and their impact upon the mean climate and transient activity of the model in the tropics. *ECMWF Tech. Memo. No. 206*.

Orr, A., Bechtold, P., Scinoccia, J., Ern, M. and Janisková, M. (2010). Improved middle atmosphere climate and analysis in the ECMWF forecasting system through a non-orographic gravity wave parametrization. *J. Climate*, **23**, 5905–5926.

Ou, S. and Liou, K.-N. (1995). Ice microphysics and climate temperature feedback. *Atmosph. Research*, **35**, 127–138.

Pailleux, J. (1990). A global variational assimilation scheme and its application for using TOVS radiances. In *Proc. WMO International Symposium on Assimilation of Observations in Meteorology and Oceanography*, pp. 325–328, Clermont-Ferrand, France.

Rabier, F. and Courtier, P. (1992). Four-dimensional assimilation in the presence of baroclinic instability. *Q. J. R. Meteorol. Soc.*, **118**, 649–672.

Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F. and Simmons, A. (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.

Rabier, F., McNally, A., Andersson, E., Courtier, P., Undén, P., Eyre, J., Hollingsworth, A. and Bouttier, F. (1998). The ECMWF implementation of three dimensional variational assimilation (3D-Var). II: Structure funtions. *Q. J. R. Meteorol. Soc.*, **124**, 1809–1829.

Radnóti, G., Trémolet, Y., Andersson, E., Isaksen, L., Hólm, E. and Janisková, M. (2005). Diagnostics of linear and incremental approximations in 4d-var revisited for higher resolution. *ECMWF Tech. Memo. No. 467*.

Scinocca, J. F. (2003). An accurate spectral nonorographic gravity wave drag parameterization for general circulation models. *J. Atmos. Sci.*, **60**, 667–682.

Smith, E. A. and Shi, L. (1992). Surface forcing of the infrared cooling profile over the Tibetan plateau. Part I: Influence of relative longwave radiative heating at high altitude. *J. Atmos. Sci.*, **49**, 805–822.

Sundqvist, H., Berge, E. and Kristjánsson, J. E. (1989). Condensation and cloud parameterization studies with a mesoscale numerical weather prediction model. *Mon. Wea. Rev.*, **117**, 1641–1657.

Talagrand, O. (1997). Assimilation of observations, an Introduction. *J. Meteorol. Soc. Japan*, **75**, N.1B,191–209.

Tavolato, C. and Isaksen, L. (2010). Huber norm quality control in the IFS. *ECMWF Newsletter No. 122*, pp. 27–31.

Temperton, C. (1988). Implicit normal mode initialization. *Mon. Wea. Rev.*, **116**, 1013–1031.

Temperton, C. (1989). Implicit normal mode initialization for spectral models. *Mon. Weather Rev.*, **117**, 436–451.

Tompkins, A. and Janisková, M. (2004). A cloud scheme for data assimilation: Description and initial tests. *Q. J. R. Meteorol. Soc.*, **130**, 2495–2518.

Trémolet, Y. (2003). Model error in variational data assimilation. In *Proc. ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean*, pp. 361–367, Reading, 8–12 Sept 2003.

Trémolet, Y. (2004). Diagnostics of linear and incremental approximations in 4D-Var. *Q. J. R. Meteorol. Soc.*, **130**, 2233–2251.

Trémolet, Y. (2005). Incremental 4d-var convergence study. *ECMWF Tech. Memo. No. 469*.

Uppala, S. M. *et al.* (2005). The ERA-40 re-analysis. *Q. J. R. Meteorol. Soc.*, **131**, 2961–3012.

Vasiljevic, D., Cardinali, C. and Undén, P. (1992). ECMWF 3D-Variational assimilation of conventional observations. In *Proc. ECMWF Workshop on Variational Assimilation with Emphasis on Three-dimensional Aspects*, Reading, 9–12 November 1992.

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

Warner, C. D. and McIntyre, M. E. (1996). An ultra-simple spectral parametrization for non-orographic gravity waves. *J. Atmos. Sci.*, **58**, 1837–1857.

Washington, W. M. and Williamson, D. L. (1977). A description of the NCAR GCMs. GCMs of the atmosphere. j. chang, ed. *Methods in Computational Physics*, **17**, 111–172, Academic Press.

Wergen, W. (1987). Diabatic nonlinear normal mode initialisation for a spectral model with a hybrid vertical coordinate. *ECMWF Tech. Report No. 59*.

Zuo, H., Balmaseda, M. A., Mogensen, K. and Tietsche, S. (2018). OCEAN5: the ECMWF Ocean Reanalysis System and its Real-Time analysis component. *ECMWF Tech. Memo. No. 823*.