Chapter 1

Overview

Table of contents

1.1 Introduction
1.2 Scientific publications
1.3 Brief history of 3D- and 4D-Var in ECMWF operations
1.4 Incremental formulation of variational data assimilation

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 methods for assimilation of TOVS radiance data and ERS scatterometer data were developed by Andersson et al. (1994) and Stoffelen and Anderson (1997), respectively. 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.

Papers of data usage and impact include a study of commercial aircraft data (Cardinali et al., 2003), scatterometer impact (Isaksen, 1997; Isaksen and Stoffelen, 2000; Isaksen and Janssen, 2004), conventional and satellite humidity data (Andersson et al., 2004), ozone analysis (Dethof and Holm, 2004), time series of frequent data (Andersson et al., 2001), wind profiler data (Bouttier, 2001b;
Andersson and Garcia-Mendez, 2002), TOVS radiance data (McNally et al., 1999), water-vapour radiances from geostationary satellites (Muñoz et al., 2004; Köpken et al., 2004), and the use of high-volume satellite data in general (Thépaut and Andersson, 2003). An more complete overview of data usage at ECMWF is included in the reanalysis review paper by Dee et al. (2011).

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 (Hörm 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., 2012a). 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 T1319. 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 T1319L50 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, 2001a), 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 \( T_{511L60} \), with inner loop resolution enhanced from \( T_{63L60} \) to \( T_{159L60} \) 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 (Veersse 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 \( T_{799L91} \), with inner loop resolution enhanced to \( T_{255L91} \). 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 \( T_{1279L91} \), with unchanged inner loop resolution. The ERA-Interim reanalysis also used 12 hour 4D-Var at \( T_{255L60} \) outer loop resolution, with \( T_{159L60} \) 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:

\[
J = J_b + J_o + J_q + J_c
\]  

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 Fisher et al. (2011). 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

\[
J(\delta x) = \frac{1}{2} \delta x^T B^{-1} \delta x + \frac{1}{2} (H \delta x - d)^T R^{-1} (H \delta x - d)
\]  

\( \delta x \) is the increment and at the minimum the resulting analysis increment \( \delta x^a \) is added to the background \( x^b \) in order to provide the analysis \( x^a \) given by

\[
x^a = x^b + \delta x^a
\]

\( B \) is the covariance matrix of background error while \( d \) is the innovation vector

\[
d = y^o - H x^b
\]

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

\[
\nabla J = (B^{-1} + H^T R^{-1} H) \delta x - H^T R^{-1} d
\]  

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

\[
\delta x^a = (B^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1} d = BH^T (HBH^T + R)^{-1} d
\]
where $B$ and $R$ are positive definite, see e.g. Lorenc (1986) for this standard result. $HBHT$ may be interpreted as the square matrix of the covariances of background errors in observation space while $BH^T$ 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 $H$ explicitly and statistical models ($B$ and $R$) 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. $B$) are described in Chapter 4 and observation operators $H$ are described in Chapter 5.
Chapter 2
4D variational assimilation

Table of contents
2.1 Introduction
2.2 Solution algorithm
  2.2.1 The incremental method
  2.2.2 The job-steps
  2.2.3 Interpolation of trajectory and increments
  2.2.4 Minimization, preconditioning and control variable
2.3 Top-level controls
2.4 A simulation
  2.4.1 The cost function and its gradient
  2.4.2 Interface between control variable and model arrays
2.5 The assimilation window
2.6 3D variational assimilation (with the first-guess at appropriate time)
  2.6.1 Introduction
  2.6.2 Solution algorithm
  2.6.3 Main differences with respect to 4D-Var
  2.6.4 Data selection

2.1 INTRODUCTION
This part of the documentation covers the top level controls of 4D-Var. It gives a detailed description of
the solution algorithm, and the various steps that are performed during a 4D-Var simulation (SIM4D).
The procedure consists of nested iterations, called inner and outer loops, which communicate through
files that either contain meteorological fields, observations, or parameters. The data flow is documented
in Chapter 11. The interpolation of model fields to observation points (OBSHOR) and the organization
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 job-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).
Figure 2.1 Schematic of the revised 4D-Var solution algorithm implemented in January 2003 (Cy25r4). Outer loops are performed at high resolution (currently T1279) using the full non-linear model. Inner iterations are performed at lower resolution (first T159, then T255) using the tangent-linear forecast model, linearised 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).

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, 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. The convergence properties of the outer-loop iterations have been investigated by Trémolet (2005).

2.2.2 The job-steps

In the CY37R2 operational configurations the assimilation window is 12-hours long, running from 09–21 UTC to produce the 12 UTC analysis and forecast products, and from 21–09 UTC for the 00 UTC production (Haseler, 2004). Several different job steps are performed.

(i) Comparison of the observations with the background at high resolution to compute the innovation vectors. These are stored in the NCMIFC1-word of the ODB (the observation database) for later use in the minimization. This job step also performs screening (i.e. blacklisting, thinning and quality control against the background) of observations (see Part 1). The screening determines which observations will be passed for use in the main minimisation. Very large volumes of data are present during the screening run only, for the purpose of data monitoring. The model trajectory is interpolated to the resolution of the next job step and written out.

(ii) First minimization at low resolution to produce preliminary low-resolution analysis increments, using simplified tangent-linear physics, and tangent-linear observation operators. The eigenvectors of the analysis Hessian are computed and will be used to precondition subsequent inner-loop iterations.
(iii) Update of the high-resolution trajectory to take non-linear effects partly into account. Observed departures from this new atmospheric state are stored in the ODB and the analysis problem is re-linearized around the updated model state. Variational quality control is computed, and the resulting QC weights will be used in the subsequent minimisation. The updated model trajectory is interpolated to the resolution of the next job step and written out.

(iv) Second main minimization at increased resolution with more complete representation of tangent-linear physics. Step (iii) and (iv) are repeated if necessary.

(v) Formation of the high-resolution analysis (described below) and comparison of the analysis with all observations (including those not used by the analysis, for diagnostic purposes).

Each of the job steps is carried out by a different configuration of IFS. They are commonly named as follows.

(i) The first trajectory run: includes a model integration, comparison to observations, and observation screening (quality control and data selection) and is sometimes called the screening run. NCONF = 2, LSCREEN = .TRUE.

(ii) The first minimization: uses simplified physics, typically run at low resolution. This job step includes estimation of analysis and forecast error variances, and calculation of Hessian eigenvectors for pre-conditioning of subsequent minimisation(s). NCONF = 131, LSPHLC = .TRUE., LAVCGL = .TRUE.

(iii) The first trajectory update: applies the analysis increments obtained in the first minimisations and performs another forecast integration with comparison to observations. This provides a new linearisation state for the next minimisation. NCONF = 1, LOBS = .TRUE.

(iv) The second (and subsequent) minimization: uses more complete tangent-linear physics, typically higher resolution increments. NCONF = 131, LSPHLC = .FALSE.

(v) Late 4D-start runs: the analyses (type = an) at the main synoptic hours (00, 06, 12, 18), that fall within the assimilation window, are formed in separate quick job-steps by adding the low-resolution increment to the penultimate high-resolution trajectory of the corresponding time (no forecast integration). See also schematic in Chapter 11.

(vi) The final trajectory runs: carries out verification screening – that is comparison between observations and final analysis. In the final trajectory job-steps with NUPTRA = 999 the final analysis (type = 4v) is formed (Bouttier, 2001a), by adding the low-resolution increment to the background (at initial time), and integrating to the analysis times. NCONF = 1, LOBS = .TRUE., NUPTRA = NRESUPD

The steps (iii) and (iv) are referred to as the second iteration of the outer loop, and these can optionally be iterated further to incorporate additional non-linear effects. The trajectory update is not normally done in 3D-Var. The inner loop takes place within the main minimization, job steps (ii) and (iv).

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 high to low resolution. This is done using appropriate grid-point interpolations for the surface grid-point fields, whereas upper-air fields are truncated spectrally. Humidity and ozone (and any other grid-point fields) are interpolated in grid-point space to the resolution of the inner loops. Initial time model cloud fields are interpolated to lower resolution using the so-called ‘full-pos’ configuration of IFS. Trajectory cloud fields and trajectory of physical tendencies are generated in a (low resolution) model integration in which the model state is replaced by the interpolated trajectory at each time instance for which it is available. The trajectory handling is managed by the module TRAJECTORY.

The spectral and grid-point analysis increments produced by the minimisation are read in by the routine RDFPINC. The spectral fields are padded with zeroes for the wave numbers not represented in the inner-loops. The increments for the spectral model variables (vorticity, divergence and logarithm of surface pressure) are added to the background fields. The temperature increments are added to the temperature of the background (after it has been converted from virtual temperature) and the resulting temperature analysis is converted back to virtual temperature. The humidity and ozone increments are read in as
grid-point fields and interpolated to the outer-loop resolution and added to the background. Checks for negative values are applied to humidity and ozone (and trace gasses).

It was found that the TL model was unstable in the mesosphere in certain situations. This was resolved by smoothing the trajectory fields if the inner-loop resolution is T255 or higher. It is implemented as a fourth-order diffusion filter applied to the spectral trajectory fields, reducing the amplitude of the shortest wave by a factor five. This is done in TRAJ_MAIN just before the fields are written to files in ifstraj. The filtering is controlled by the variables FILTERFACTOR, FILTEREXPO, FILTERRESOL in NAMVAR. In addition, it was found necessary to reduce the amplitude of the analysis increments near the top of the model. They are tapered to zero applying the factor REDINC, over the topmost NLEV,REDINC levels (NAMJG), in RDFPINC.

2.2.4 Minimization, preconditioning and control variable

In practice, it is necessary to pre-condition the minimization 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 $B$. 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 minimization algorithms have generally to evaluate several inner products, it was found more efficient to implement a change of variable (under CHAVAR, CHAVARN, etc.). Algebraically, this requires the introduction of a variable $\chi$ such that

$$J_b = \chi^T \chi \quad (2.1)$$

Comparing (1.2) and (2.1) shows that $\chi = B^{-1/2} \delta x$ satisfies the requirement. $\chi$ 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 $J_b$-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

The routines CVA1, CVA2 and FORECAST_ERROR control the variational configurations of IFS. The flow diagram of CVA1 and CVA2 is shown in Fig. 2.2. The spectral and grid-point first guess fields (FG) have been read in to the SP7-arrays and GP7-arrays (in YOMSP) by SUECGES, called from SUJBSTD within the $J_b$ setup, see Subsection 4.3.3. At the start of CVA1 additional setups for the variational configurations are done (SU1YOM). The SP3-arrays and GFL-arrays, i.e. the current model state, (in YOMSP) are filled by a call to the setup routine SUINIF in the routine SUVAZX, and only for the first minimisation job-step. For subsequent minimisations the initial state for analysed variables must come from the previous minimisation, i.e. a warm start. This is achieved by getting the control vector YVAXZ by calling GETMINI and performing an inverse change of variables (calling CHAVARIN) to transform the control vector to model variables (SP3/SP2-arrays and GFL-arrays). Non-analysed fields (e.g. cloud fields) are read by a call to SUINIF earlier in SUVAZX.

After minimisation is complete one final simulation is performed in CVA2 or FORECAST_ERROR. This simulation is diagnostic, and characterized by the simulation counter being set to 999, NSIM4D = NSIM4DL, yomvar. The observation departure from the low-resolution analysis, $y' = Hx_a^{LR}$, is computed and stored in the NCMIOU-word of the ODB. Finally at the end of CVA2 and FORECAST_ERROR, respectively, the updated ODB is written to disk, using the routine WRITEOBA.
2.4 A SIMULATION

2.4.1 The cost function and its gradient

A simulation consists of the computation of $J$ and $\nabla J$. This is the task of the routine SIM4D, see Fig. 2.3 for the flow diagram. The input is the latest value of the control variable $\chi$ in the array VAZX, computed by M1QN3, or CONGRAD. First $J_b$ and its gradient are computed (see Section 4.2) using

$$J_b = \chi^T \chi$$
$$\nabla \chi J_b = 2 \chi$$

(2.2)

The gradient of $J_b$ with respect to the control variable is stored in the array VAZG (YOMCVA).

(i) Copy $\chi$ from VAZX to SP3-arrays (YOMSP) using the routine CAIN.
(ii) Compute $x$, the physical model variables, using CHAVARIN so that
\begin{equation}
  x = \delta x + x^b = L\chi + x^b
\end{equation}

(iii) SUBFGS computes $\delta x = x - x^g$, where $x^g$ is the guess (not necessarily equal to the background).

(iv) Perform the direct integration of the linear model, using the routine CNT3TL, and compare with observations. See Section 2.5.

(v) Calculate $J_o$ for which OBSV is the master routine.

(vi) Perform the adjoint model integration using CNT3AD, and observation operators’ adjoint.

(vii) Calculate $\nabla_x J_o$, and store it in SP3.

(viii) $J_c$ and its gradient are calculated in COSJC called from CNT3AD, if LJC is switched on (default) in namvar.

(ix) Transform $\nabla_x J_o + \nabla_x J_c$ to control variable space by applying CHAVARINAD.

(x) Copy $\nabla_x J_o + \nabla_x J_c$ from SP3 and add to $\nabla_x J_o$, already in the array VAZG, using CAIN.

(xi) Add the various contributions to the cost function together, in EVCOST, and print to log file using prtjo.

(xii) Increase the simulation counter NSIM4D by one.

Figure 2.3 Flow diagram for the subroutine sim4d.
The new $J$ and $\nabla \chi J$ are passed to the minimization algorithm to calculate the $\chi$ of the next iteration, and so on until convergence (or the maximum number of iterations) has been reached.

### 2.4.2 Interface between control variable and model arrays

The purpose of the routine CAIN (the canonical injection) is to identify those parts of the model state that should be included in the variational control variable. This is controlled by on/off switches such as NVA2D and NVA3D (yomcva) initialized in SUALCTV. The scalar product used is the one defined by the array SCALP (in yomcva, set up in the routine SCALJGS called from SUSCAL), which is 1 if $m = 0$, and 2 otherwise. This allows the compression of arrays of the type VAZX while using the $L^2$ norm on the sphere with real fields in spectral space.

The control vector currently consists of four parts, as defined in the module CONTROL_VECTORS residing in the ifsaux library. Any of the following four parts can be absent.

(i) The initial-condition control-variables that define the original 4D-Var problem.
(ii) A representation of model error (for weak-constraint 4D-Var), see Trémolet (2003).
(iii) The TOVS control vector (LTOVSCV), used to estimate surface skin temperature and simplified cloud variables (pressure and fraction) at the radiance field-of-view locations.
(iv) Components for the limited-area model ALADIN.

The organisation of elements of the control-vector is now quite flexible and allows for future planned extensions, such as observation bias coefficients (e.g. for radiance data, and wavelet-$J_b$ components).

### 2.5 THE ASSIMILATION WINDOW

In 4D-Var, observations are organized 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

$$
J(\delta x) = \frac{1}{2} \delta x^T B^{-1} \delta x + \frac{1}{2} \sum_{i=0}^{n} (H_i \delta x(t_i) - d_i)^T R_i^{-1} (H_i \delta x(t_i) - d_i) \tag{2.4}
$$

with subscript $i$ the time index. Each $i$ corresponds to a half-hour time slot. $\delta x$ is as before the increment at low resolution at initial time, and $\delta x(t_i)$ the increment evolved according to the tangent linear model from the initial time to time index $i$. $R_i$ and $B$ are the covariance matrices of observation errors at time index $i$ and of background errors respectively. $H_i$ 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 $d_i = y^o_i - H_i x^b(t_i)$, where $x^b(t_i)$ is the background propagated in time using the full non-linear model and $y^o_i$ is the observation vector at time index $i$.

### 2.6 3D VARIATIONAL ASSIMILATION (WITH THE FIRST-GUESS AT APPROPRIATE TIME)

#### 2.6.1 Introduction

3D-Var is a temporal simplification of 4D-Var. The simplest way to implement a 3D-Var within the context of an existing 4D-Var scheme is to replace the tangent-linear (and its adjoint) model integration within the inner loops (as defined in the previous section) by the identity operator (LIDMODEL = .TRUE. in yomtnewt.F90). This approach has indeed been adopted, as it saves on maintenance of scripts and codes. It is this 3D-Var version that was used for the production of the ERA-40 re-analysis (Uppala et al., 2005) and it was used until 14 March 2006 in ECMWF operations within the BC-suite to generate timely boundary conditions for the participating member states.

In this version of 3D-Var as much as possible of the 4D-Var structure is maintained. The available observations, typically over the period of a 6-hour assimilation window, are compared with a model
integration at high resolution. The comparison between observation and model is thus performed at the appropriate time: therefore the abbreviation 3D-FGAT (first-guess at appropriate time). This configuration is activated via the switch LFGAT = .TRUE. in namvar.h.

2.6.2 Solution algorithm

The tangent-linear model in 4D-Var evolves the analysis increment over time, within the assimilation window. In 3D-FGAT no such evolution takes place; one static increment in produced. The valid time of this increment is not at the initial time of the window as in 4D-Var, but at the central time. The increment is added to the background field, which forms the analysis at the central time. This has the consequence that a trajectory update from the start of the assimilation window cannot be performed. The 3D-FGAT configuration is therefore based on one single outer-loop without iteration. There are nevertheless two minimisation steps. The first provides analysis error and background error estimates required for cycling, and also pre-conditioning vectors (Hessian eigenvectors) that are used in the second minimisation job-step. The resolution of the first minimisation is typically lower (T42) than for the main one (e.g. T159).

Scatterometer de-aliasing is active and variational quality control is activated approximately halfway into the minimisation, rendering the problem non-linear (and the cost function non-quadratic). The use of conjugate-gradient minimisation is thus prevented. The M1QN3 algorithm is used instead.

2.6.3 Main differences with respect to 4D-Var

The main differences with respect to the standard operational 4D-Var configuration are as follows.

(i) Variational quality control is carried out within the inner-loop iterations (LQVARQC = .FALSE.).
(ii) The scatterometer de-aliasing is performed within the inner-loop iterations (LQSCAT = .FALSE.).
(iii) Use of M1QN3 instead of conjugate gradient minimisation, due to the cost-function being quadratic.
(iv) The \( J_c \) term based on a digital filter is not used (LJCDFI = .FALSE.).
(v) No iteration at outer-loop level.
(vi) LTRAJHR = .FALSE., which means that the observation operators are linearized around the low-resolution trajectory, which is generated through an integration by the full non-linear model (at the resolution of the minimisation) at the beginning of the minimisation job-step. CVA1 calls CNT2 to do this.

2.6.4 Data selection

The observational data are stored in time-slots within the ODB (just as in 4D-Var). The data selection criteria are applied in the same way as in 4D-Var, allowing time sequences of data to be used also in 3D-FGAT. The 3D scheme is however unable to extract temporal information from the data, and produces an analysis of the temporal average of the FGAT departures. Optionally the screening data selection rules can be applied once for the entire (6-hour) assimilation window, which would pick only the data closes to the analysis centre time. This is called ‘3D-screening’ and was the practice in the original operational 3D-Var configuration.
Chapter 3
Tangent-linear physics

Table of contents
3.1 Introduction
3.2 Set-up
3.3 Mixed-phase thermodynamics
3.4 Vertical diffusion
   3.4.1 Upper-air
   3.4.2 Surface
   3.4.3 Regularization
   3.4.4 Flow chart
3.5 Subgrid scale orographic effects
3.6 Non-orographic gravity wave drag
3.7 Radiation
   3.7.1 The short-wave radiation scheme
   3.7.2 The longwave radiation scheme
   3.7.3 Cloud overlap assumptions
   3.7.4 Cloud optical properties
3.8 Large-scale condensation and precipitation
   3.8.1 Stratiform condensation
   3.8.2 Convective contribution
   3.8.3 Precipitation formation
   3.8.4 Precipitation evaporation
   3.8.5 Phase partitioning
   3.8.6 Code
   3.8.7 Regularization
3.9 Moist convection
   3.9.1 Equations for updraught and downdraught
   3.9.2 Triggering of moist convection
   3.9.3 Entrainment and detrainment
   3.9.4 Precipitation formation
   3.9.5 Closure assumptions
   3.9.6 Flow chart
   3.9.7 Regularization
3.10 Trajectory management

3.1 INTRODUCTION
The linearized physics describes six processes: vertical diffusion, subgrid-scale orographic effects, non- orographic 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.: LE PHYS 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 $T_0 = 0^\circ C$ and $T_{\text{ice}} = -23^\circ C$ given by

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

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

The tuning parameter $\mu$ controls the intensity of the smoothing, and the temperature $T_{\text{crit}}$ 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 $\psi$ (wind components $u, v$; dry static energy $s$; specific humidity $q$), the tendency of its perturbation $\psi'$ produced by vertical diffusion is

$$\frac{\partial \psi'}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial z} \left( K(Ri) \frac{\partial \psi'}{\partial z} \right) \quad (3.2)$$
where $\rho$ is the air density and $Ri$ is the Richardson number.

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

$$K = l \left\| \frac{\partial V}{\partial z} \right\| f(Ri)$$

(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

#### Stable conditions

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

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

(3.4)

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

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

$$l = \frac{\kappa(z + z_0)}{1 + \kappa \left[ \frac{z + z_0}{\lambda} \right]}$$

(3.5)

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

#### Unstable conditions

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

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

(3.6)

The vertical profile of the mixing length in the unstable case also follows Blackadar (1962) (see 3.5), with $\lambda$ 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

$$C_{MN} = \frac{\kappa^2}{\ln \left( \frac{z + z_0}{z_0} \right)^2}$$

and

$$C_{HN} = \frac{\kappa^2}{\ln \left( \frac{z + z_0}{z_0} \right) \ln \left( \frac{z + z_0}{z_0} \right)}$$

(3.7)

The drag coefficients for momentum are computed as

$$C_M = \begin{cases} 
\frac{1}{1 + \frac{2bRi}{\sqrt{1 + dRi}}} C_{MN} & \text{if } Ri > 0 \\
1 - \frac{2bRi}{1 + 3bcC_{MN} \sqrt{\frac{z + z_0}{z_0}} |Ri|} C_{MN} & \text{if } Ri < 0
\end{cases}$$

(3.8)

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

$$C_H = \begin{cases} 
\frac{1}{1 + 3bcC_{HN} \sqrt{\frac{z + z_0}{z_0}} |Ri|} C_{HN} & \text{if } Ri > 0 \\
1 - \frac{3bRi}{1 + 3bcC_{HN} \sqrt{\frac{z + z_0}{z_0}} |Ri|} C_{HN} & \text{if } Ri < 0
\end{cases}$$

(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 $q$ (VDFSURF TL, VDFSURFAD), the surface exchange coefficients (VDFEXCSTL, VDFEXCSAD) and the equivalent evapotranspiration efficiency (VDFEVA PTL, VDFEVA PAD). 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 (VDFEXCUS TL, VDFEXCUSAD), the solving of the diffusion equations for momentum (VDFDIFMSTL, VDFDIFMSAD) and dry static energy and moisture (VDFDIFHS TL, 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

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

(3.10)

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

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

$$\left( \frac{\partial U}{\partial t} \right)_{blk} = -C_d \max \left( 2 - \frac{1}{r} , 0 \right) \frac{\sigma}{2\mu} \left( \frac{Z_{blk} - z}{z + \mu} \right)^{0.5} \left( B \cos^2 \psi + C \sin^2 \psi \right) \frac{|U||U|}{2}$$

(3.11)

where
Part II: Data Assimilation

\[ C_d \] is the low-level drag coefficient \((GWAKE = 1)\)
\[ \sigma \] is the mean slope of the subgrid-scale orography
\[ \gamma \] is the anisotropy of the subgrid-scale orography
\[ \psi \] 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)

\[
\left( \frac{\partial U}{\partial t} \right)_{blk} = -\beta \frac{U^{n-1}}{(\beta + 1)2\Delta t}
\] (3.12)

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

\[
\left( \frac{\partial U}{\partial t} \right)_{blk} = -\frac{U^{n-1}}{(\beta + 1)2\Delta t} - \frac{\beta U^{m-1}}{\beta + 1} \] (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

\[
m^2 = \frac{k^2}{\tilde{\omega}^2} = \frac{N^2}{\tilde{c}^2}
\] (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 \( \phi \) following Fritts and Nastrom (1993) as

\[
\tilde{E}(m, \tilde{\omega}, \phi) = B \left( \frac{m}{m_*} \right)^s \frac{N^2\tilde{\omega}^{-d}}{1 - \left( \frac{m}{m_*} \right)^{s+d}}
\] (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 \( \rho \tilde{F}(m, \omega, \phi) \) 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 \((k, \omega)\) 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 \(\bar{\rho}F_E\) and northward \(\bar{\rho}F_N\) 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 \(\phi_i\) 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 LEGGWMS2 to .TRUE. in namelist NAMTRAJP. The schemes would then be used with the prescribed time frequency GTPHYGWMS. The necessary regularizations are activated by setting logical switch LREGGWMS 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 \(\mu m\). 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

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

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

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

\[ R_{\text{top}} = C_{\text{cloud}} R_{\text{cloud}} + (1 - C_{\text{cloud}}) R_{\text{clear}} \]  
\[ T_{\text{top}} = C_{\text{cloud}} T_{\text{cloud}} + (1 - C_{\text{cloud}}) T_{\text{clear}} \]  

(3.20) \hspace{1cm} (3.21)

In the previous equations, \( C_{\text{cloud}} \) 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 \( \nu \) 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 \( F_{\text{lw}}^u(i) \) and \( F_{\text{lw}}^d(i) \) 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. \( F_{\text{lw}}^u(i) \) and \( F_{\text{lw}}^d(i) \) for a cloud present in the \( n \)th 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 \( F_{\text{lw}}^u(i) \) and downward \( F_{\text{lw}}^d(i) \) fluxes are expressed as:

\[
F_{\text{lw}}^u(i) = (1 - CC_{N,i}) F_{0}^u(i) + \sum_{k=1}^{N} (CC_{i,k+1} - CC_{i,k}) F_{k}^u(i) \]  
\[
F_{\text{lw}}^d(i) = (1 - CC_{i-1,0}) F_{0}^d(i) + \sum_{k=1}^{i-1} (CC_{i,k+1} - CC_{i,k}) F_{k}^d(i) \]  

(3.22) \hspace{1cm} (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 \( (\varepsilon_{\text{cloud}}) \) 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 \( \text{H}_2\text{O} \) and \( \text{CO}_2 \) 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

\[ CC_{i,j} = 1 - (1 - C_i) \prod_{k=i+1}^{j-1} \left[ 1 - \frac{\text{max}(C_k, C_{k-1})}{1 - C_{k-1}} \right] \] (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 \( \delta_c \), the asymmetry factor \( g_c \) and the single scattering albedo \( \omega_c \). They are derived from Fouquart (1987) for the water clouds, and Ebert and Curry (1992) for the ice clouds. The optical thickness \( \delta_c \) is related to the cloud liquid/ice water amount \( u_{LWP} \) by:

\[ \delta_c = u_{LWP} \left( a_i + \frac{b_i}{r_e} \right) \] (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 \( \mu m \) over oceans and to 10 \( \mu m \) over continents. When ice cloud optical properties were initially introduced in the radiation code, the effective radius was set to 40 \( \mu m \). 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):

\[ r_e = 326.3 + 12.42 \times T_i + 0.197 \times T_i^2 + 0.0012 \times T_i^3 \] (3.26)

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

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

\[ \bar{\omega}_{c1} = 0.9999 - 5 \times 10^{-4} \exp(-0.5\delta_c) \] (3.27)
\[ \bar{\omega}_{c2} = 0.9988 - 2.5 \times 10^{-3} \exp(-0.05\delta_c) \] (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:

\[ \bar{\omega}_i = c_i - d_i r_e \] (3.29)
\[ g_i = e_i + f_i r_e \] (3.30)

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

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

\[ \varepsilon_{cld} = 1 - \exp(-k_{abs} u_{LWP}) \] (3.31)
where $k_{\text{abs}}$ 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. $u_{LH} \rho$ is the condensed water path. $k_{\text{abs}}$ 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 CVAI. 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 LWCLHR=.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 LWCLHR is automatically set to .FALSE.. The same can be achieved by using default value for LWCLHR, 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

\[
\begin{align*}
\frac{\partial q}{\partial t} &= -C + E_{\text{prec}} + D_{\text{conv}} \\
\frac{\partial T}{\partial t} &= L(C - E_{\text{prec}} - D_{\text{conv}}) + L_f(F - M)
\end{align*}
\]

where $C$ denotes large-scale condensation (negative if evaporation), $E_{\text{prec}}$ is the moistening due to the evaporation of precipitation and $D_{\text{conv}}$ is the detrainment of cloud water from convective clouds. $F$ and $M$ 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, \( RH_{\text{crit}} \). The half width of the distribution is given by

\[
W = q_{\text{sat}} \left\{ 1 - RH_{\text{crit}} - \kappa(RH - RH_{\text{crit}}) \right\}
\]

(3.34)

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

\[
RH_{\text{crit}} = 0.7\sigma(1 - \sigma)(1.85 + 0.95(\sigma - 0.5))
\]

(3.35)

\( \kappa \) is a coefficient that reduces the width of the uniform distribution when \( RH \) increases and it also depends on \( \sigma \) as

\[
\kappa = \max(0, 0.9(\sigma - 0.2)^{0.2})
\]

(3.36)

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

\[
C_{\text{strat}} = 1 - \sqrt{1 - RH - \kappa(RH - RH_{\text{crit}})}
\]

(3.37)

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

(3.38)

where \( q_{\text{sat}} \) is the saturation specific humidity.

Figure 3.1 displays the variations of stratiform cloud cover, \( C_{\text{strat}} \), as a function of relative humidity for various settings of parameter \( \kappa \).

![Figure 3.1](image)

Stratiform cloud cover, \( C_{\text{strat}} \), as a function of grid-box mean relative humidity for various values of parameter \( \kappa \) (see top left legend). Relative humidity ranges between \( RH_{\text{crit}} \) 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, \( C_{\text{conv}} \), and cloud condensate, \( q_{\text{conv}}^c \), resulting from convection are computed as

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

(3.39)

\[
q_{\text{conv}}^c = \delta_u M_u q_l^0 \frac{\Delta t}{\rho \Delta z}
\]

(3.40)

where \( \delta_u \) (unitless), \( M_u \) (in kg m\(^{-2}\) s\(^{-1}\)) and \( q_l^0 \) (in kg kg\(^{-1}\)) are the detrainment rate, mass flux and cloud liquid water content in the convective updraught, respectively. \( \Delta t \) and \( \Delta z \) denote the model time step and model layer depth, while \( \rho \) 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

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

(3.41)

where $C = C_{\text{strat}} + C_{\text{conv}}$ is used to obtain in-cloud condensate amounts. The critical cloud water threshold, $q_c^{\text{crit}}$, is set to $3 \times 10^{-4}$ kg kg$^{-1}$ if precipitation evaporation is activated, $6 \times 10^{-4}$ kg kg$^{-1}$ otherwise. The conversion factor $C_0$ is set to $3.33 \times 10^{-4}$ 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

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

(3.42)

where the quantity $D$ is computed as

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

(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 $F$ 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, $\overline{q_{\text{prec}}} = q_{\text{sat}} - f_{\text{clr}}^{\text{prec}}(q_{\text{sat}} - \overline{q})$, in the clear-sky portion through which precipitation is falling, denoted $f_{\text{clr}}^{\text{prec}}$. Consistent with the maximum overlap assumption made in the calculations of total cloud cover and precipitation fraction, it is hypothesized that $f_{\text{clr}}^{\text{prec}}$ corresponds to the moistest part of the uniform distribution, as illustrated in Fig. 3.2.

![Figure 3.2](image)

Figure 3.2 Illustration of the uniform distribution of specific humidity over the model grid box with various quantities referred to in the text.

From this, one can show that

$$q_{\text{prec}}^{\text{clir}} = q_{\text{sat}} - f_{\text{clr}}^{\text{prec}}(q_{\text{sat}} - \overline{q})$$

(3.44)

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

Precipitation evaporation is parameterized following Kessler (1969):

$$\left(\frac{\partial q}{\partial t}\right)_{\text{evap}} = f_{\text{clr}}^{\text{prec}} \overline{\beta}(q_{\text{sat}} - \overline{q}_{\text{prec}})$$

(3.45)
where $\beta$ is given by
\[
\beta = 5.44 \times 10^{-4} \left\{ \left( \frac{p}{p_{\text{surf}}} \right)^{\frac{1}{4}} \frac{P_{\text{clr}}}{5.9 \times 10^{-3} f_{\text{prec}}} \right\}^{0.577} \tag{3.46}
\]
where $P_{\text{clr}}$ denotes precipitation in the clear-sky fraction of the grid box. Equation (3.45) is solved implicitly to take into account the reduction of $q_{\text{sat}}$ due to evaporative cooling, which yields the moistening associated to precipitation evaporation:
\[
E_{\text{prec}} = \frac{f_{\text{clr}} P_{\text{clr}} \beta \Delta t (q_{\text{sat}} - q_{\text{clr}})}{1 + \beta \Delta t \left[ 1 + \frac{L_v c_p}{c_p} \frac{\partial q_{\text{sat}}}{\partial T} \right]} \tag{3.47}
\]
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, $\alpha_l$, 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 \leq C_{\text{strat}} < 0.95 \\
0.1 \sqrt{(1 - C_{\text{strat}})/(1 - 0.95)} & \text{if } C_{\text{strat}} \geq 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 \( \psi \) (dry static energy, wind components, specific humidity, cloud liquid water) can be written in a mass-flux formulation as Betts (1997)

\[
\frac{\partial \psi}{\partial t} = \frac{1}{\rho} \left[ (M_u + M_d) \frac{\partial \psi}{\partial z} + D_u(\psi_u - \psi) + D_d(\psi_d - \psi) \right]
\]

(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, \( s = c_p T + g z \), 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

\[
\frac{\partial M_u}{\partial z} = (\epsilon_u - \delta_u)M_u
\]

(3.50)

\[
\frac{\partial M_d}{\partial z} = -(\epsilon_d - \delta_d)M_d
\]

(3.51)

where \( \epsilon \) and \( \delta \) 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, \( \psi \), of the updraught or downdraught, namely

\[
\frac{\partial \psi_u}{\partial z} = -\epsilon_u(\psi_u - \overline{\psi})
\]

(3.52)

\[
\frac{\partial \psi_d}{\partial z} = \epsilon_d(\psi_d - \overline{\psi})
\]

(3.53)

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

In practice, (3.50) and (3.51) are solved in terms of \( \mu = M/M^\text{base}_u \), where \( M^\text{base}_u \) is the mass flux at cloud base (determined from the closure assumptions as described further down). \( \mu \) 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 \( \mu \) values are eventually multiplied by \( M^\text{base}_u \) 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

\[
w_* = 1.2 \left( a^3 - 1.5 g z \kappa \left[ \frac{J_s}{\rho c_p T} + 0.608 \frac{J_q}{E_v T} \right] \right)
\]

(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, $\Delta T_u$, and moisture excess, $\Delta q_u$, of the test parcel with respect to the environment are prescribed as

\[ \Delta T_u = -1.5 \frac{J_s}{\rho c_p u_*} \quad \text{and} \quad \Delta q_u = -1.5 \frac{J_q}{\rho L_v u_*} \]  

(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

\[ \Delta T_u = 0.2 \, \text{K} \quad \text{and} \quad \Delta q_u = 1 \times 10^{-4} \, \text{kg kg}^{-1} \]  

(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

\[ s_u^k = \bar{s}_k + c_p \Delta T_u, \]

where the tilde symbol represents a 50 hPa layer average, instead of $s_u^k = \bar{s}_k + c_p \Delta T_u$ 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:

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

(3.57)

where ENTRORG2 = 1.75 $\times$ 10$^{-4}$ m$^{-1}$. $\epsilon_u$ is therefore a function of the local relative humidity in the environment, $RH$, and of the ratio between the local saturation specific humidity, $q_{sat}$, 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

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

(3.58)

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

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

\[ \delta_u = \max \left( \delta_u^*, 1 - \sqrt{\frac{(w_u^2)_k}{(w_u^2)_{k+1}}} \right) \]  

(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

\[ \epsilon_d = \begin{cases} \text{ENTRDD2} + \epsilon_{\text{org}} & \text{for } z_{\text{topde}} \leq z \leq z_{\text{LFS}} \\ 0 & \text{elsewhere} \end{cases} \]  

(3.60)

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

\[ \epsilon_{\text{org}} = \frac{g T_{v,a} - T_d r_d - \bar{T}_{v,a} \bar{T}_{v,a}}{(w_{LFS}^d)^2 - \int_{z_{\text{LFS}}}^{z} \frac{g T_{v,a} - T_d r_d - \bar{T}_{v,a}}{T_v} \, dz} \]  

(3.61)

where \(w_{LFS}^d\) 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

\[ e_d = \frac{g \Delta p (q_d - \hat{q}_d)}{M_d} \]  

(3.62)

where \(q_d\) is the value of the downdraft humidity before saturation adjustment and \(\hat{q}_d\) is the humidity after the saturation adjustment. The value of the rain water content in the downdraft used in (3.61) is estimated as \(r_d = e_d \Delta p / (g M_d)\).

Detrainment in the downdraught is defined as

\[ \delta_d = \begin{cases} \frac{1}{z_{\text{topde}}} \frac{M_d^{\text{topde}}}{M_d} & \text{for } z < z_{\text{topde}} \\ \text{ENTRDD2} & \text{for } z_{\text{topde}} \leq z \leq z_{\text{LFS}} \\ 0 & \text{for } z > z_{\text{LFS}} \end{cases} \]  

(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))

\[ \epsilon_u^{\text{mom}} = \begin{cases} \epsilon_u + 2 \delta_u & \text{in all types of convection} \\ \epsilon_u + 3 \delta_u & \text{at the top three levels of deep convective cloud} \\ 0 & \text{otherwise} \end{cases} \]  

(3.64)

3.9.4 Precipitation formation

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

\[ \left( \frac{\partial q_u^l}{\partial z} \right)_{\text{prec}} = -C_0 C_{\text{BF}} q_u^l \left\{1 - \exp \left[-\left(\frac{q_u^l}{q_u^{\text{crit}}}ight)^2\right]\right\} \]  

(3.65)

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

\[ C_0 = \begin{cases} 1.4 \times 10^{-3} \\ 0.75 w_u \end{cases} \]  

(3.66)

\[ C_{\text{BF}} = \begin{cases} 1 + 0.5 \sqrt{\min(T_{BF} - T_{\text{ice}}, T_{BF} - T_u)} & \text{if } T_u < T_{BF} \\ 1 & \text{if } T_u \geq T_{BF} \end{cases} \]  

(3.67)
where \( w_u \) and \( T_u \) are the vertical velocity and the temperature of the updraught, and \( T_{\text{ice}} = 250.16 \, \text{K} \) \((-23^\circ \text{C})\) and \( T_{\text{BF}} = 268.16 \, \text{K} \) \((-5^\circ \text{C})\).

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 \((\text{Cond})\) produced by convection during the ascent from levels \( k+1 \) to \( k \) (by convention, model level number increases downwards):

\[
(q^u_{t})_k = (q^u_{t})_{k+1} \exp(-D) + \text{Cond} \frac{1 - \exp(-D)}{D}
\]

where the quantity \( D \) is computed as

\[
D = \frac{C_0 \, C_{\text{BF}} \Delta z}{1 - \exp \left( - \left( \frac{(q^u_{t})_{k+1}}{q^u_{\text{crit}}^t} \right)^2 \right)}
\]

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, \( M^u_{\text{base}} \), 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

\[
\text{CAPE} = \int_{\text{cloud}} g \left( \frac{T^u - T_v}{T_v} - q^u \right) \, dp
\]

\[
\text{HEAT} = \int_{\text{cloud}} g \left( \frac{1}{c_p} \frac{\partial s}{\partial z} + 0.608 \frac{\partial \bar{p}}{\partial z} \right) (\mu_u + \mu_d) \, dz
\]

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

\[
\text{CAPE}_{\text{bl}} = -\tau_{\text{bl}} \frac{1}{T_v} \int_{P_{\text{part}}} \frac{\partial T_v}{\partial t} \bigg|_{\text{bl}} \, dp
\]

In (3.72), the tendency \( \frac{\partial T_v}{\partial t} \bigg|_{\text{bl}} \) 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 \( T_v = c_p^{-1} g H \) is set to 1 K and different boundary-layer time-scales \( \tau_{\text{bl}} \) are used over land and water (namely the convective turnover time-scale over land and the advective time-scale over water)

\[
\tau_{\text{bl}} = \begin{cases} \frac{H_{\text{land}}}{w_{\text{up}}} & \text{over land} \\ \frac{H_{\text{base}}}{\bar{u}_{\text{bl}}} & \text{over water} \end{cases}
\]

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

With this definition CAPE\(_{\text{bl}}\) can also be seen as an efficient sine filter on CAPE. CAPE\(_{\text{bl}}\) is set to zero for convection originating above the boundary layer.
Finally, the cloud-base mass flux for deep convection situations is given by:

$$M_{u}^{\text{base}} = \frac{CAPE - CAPE_{bl}}{HEAT \times \tau}$$  \hspace{1cm} (3.74)

where $\tau$ is an adjustment timescale (in seconds), which is expressed as

$$\tau = (1 + 264/nT)H_{cld}/\overline{w_{u}}$$  \hspace{1cm} (3.75)

where $nT$ denotes the spectral truncation.

Shallow convection

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

$$\delta h_{\text{base}} = \left[ c_p(T_u - \overline{T}) + L_v(q_u + q_{ul} - q_{l}) \right]_{\text{base}}$$  \hspace{1cm} (3.76)

$$\delta h_{\text{SCL}} = \int_{\text{SCL}} \left( c_p \frac{\partial T}{\partial t} + L_v \frac{\partial q}{\partial t} \right) \frac{dp}{g}$$  \hspace{1cm} (3.77)

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

$$M_{u}^{\text{base}} = \frac{\delta h_{\text{SCL}}}{\delta h_{\text{base}}}$$  \hspace{1cm} (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

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

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 \( M_{u}^{\text{base}} \) for shallow convection.
- \( (M_{u}^{\text{base}})' \) is scaled by 0.2 in test whether \( \delta h_{\text{SCL}}/(\delta h_{\text{base}})^2 > 10^3 \).
- \( (M_{u}^{\text{base}})' \) is scaled by 0.25 for both deep and shallow convection.

CUBASEN2TL/AD:
- \( w*' = 0 \) if \( w_* < 0.5 \text{ m s}^{-1} \) (convective-scale velocity).
- Buoyancy perturbation is scaled by 0.35 in updraught initialization.

CUASCN2TL/AD:
• \((e_{org}^*)^'\) is scaled by 0.1.

• \((w_{org}^*)^'\) is set to 0 in organized detrainment computations.

• \(w'_u\) is set to

\[
\begin{cases} 
0 & \text{if } w_u < 0.1 \text{ m s}^{-1} \\
0.1 \times w'_u & \text{if } w_u \geq 0.1 \text{ m s}^{-1}
\end{cases}
\]

• \((q_{sat}^*)^'\) is set to 0 in turbulent entrainment computations.

• Buoyancy perturbation is scaled by 0.33 in updraught computations.

\textbf{CUPDRATL/AD:}

• Buoyancy perturbation is scaled by 0.33 in updraught computations.

\textbf{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 \(\psi\). Therefore

\[
\frac{\psi^{n+1} - \psi^{n-1}}{2\Delta t} = P(\psi^{n-1}) \tag{3.79}
\]

where the variable \(\psi\) 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

\[
\psi^{n-1}_u = \psi^{n-1} + \left(\frac{\partial \psi}{\partial t}\right)_{dyn} + \left(\frac{\partial \psi}{\partial t}\right)_{phys} \tag{3.80}
\]

In (3.79), if the operator \(P\) is non-linear, its linearization around the basic state \(\psi^{n-1}_u\), 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 \(\left(\frac{\partial \psi}{\partial t}\right)_{dyn}\). The physical tendencies from the previous processes \(\left(\frac{\partial \psi}{\partial t}\right)_{phys}\), require an additional call to the non-linear routines in the adjoint computations (\textsc{Callparad}) and a local storage of the partial tendencies.

The storage of the trajectory at \(t - \Delta t\) is performed in \textsc{ECPhys} by the routine \textsc{StoreTrajPhys} called before the driver of the ECMWF physics \textsc{Callpar}. Fields are stored in grid-point space in an array \textsc{TrajPhys}. This array is allocated in the module \textsc{TrajPhysics}, 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 \textsc{ECPhys_Tl} and \textsc{ECPhys_Ad} before calling the driver of the
tangent-linear physics \texttt{CALLPARTL} or the driver of the adjoint physics \texttt{CALLPARAD}, and after reading the trajectory fields from \texttt{GET\_TRAJ\_PHYS}.
Chapter 4
Background term

Table of contents
4.1 Introduction
4.2 Description of the algorithm
  4.2.1 The wavelet correlation matrix and gridpoint variances
  4.2.2 The multivariate error correlations between variables: ‘balance’
  4.2.3 The univariate variable transform: ‘gaussianization’
4.3 Implementation
  4.3.1 The $J_b$ information structure: SPJB_VARS_INFO
  4.3.2 User input
  4.3.3 Input files
  4.3.4 Initial setup
  4.3.5 Change of variable

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.

4.2 DESCRIPTION OF THE ALGORITHM

We use the following notation.

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

(ii) $\delta x = (\delta \zeta, \delta \eta, \delta (T, p_{surf}), \delta q, \delta \omega, \ldots)^T$ 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 $	ilde{\cdot}$ denotes a univariate transform of the increments with more Gaussian error statistics than the original increments.

(iv) A subscript $\cdot_0$ 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 $\delta x = L \chi$ (Section 1.4) where $L$ satisfies $LL^T = B$ so that $J_b$ takes a simple form. In operational practice, the initial point of the minimization is either the background (in which case $\delta x = \chi = 0$) or the control vector saved from an earlier minimization. The minimization is carried out in the space of $\chi$. At the end of the minimization, the analysis increments are reconstructed in model space by $\delta x = L \chi$. 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.
Chapter 4: Background term

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

$$L = KL_u$$  (4.1)

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

The $L_u$ operator defines the covariance matrix for $\delta \tilde{x}_u$ as

$$B_u = L_u L_u^T$$  (4.2)

So far, the formulation is quite general. Now, we restrict $L_u$ to a simple form and choose a particular balance and variable transform operator $K$.

The covariance matrix $B_u$ is assumed to be block-diagonal, with no correlation between the parameters, so that

$$B_u = \begin{bmatrix} C_\zeta & 0 & 0 & 0 & 0 \\ 0 & C_{qp} & 0 & 0 & 0 \\ 0 & 0 & C_{(T,p_{surf})u} & 0 & 0 \\ 0 & 0 & 0 & C_{q_u} & 0 \\ 0 & 0 & 0 & 0 & C_{o3} \end{bmatrix}$$  (4.3)

The matrix $L_u$ is similarly block-diagonal, with diagonal blocks $L_\zeta, L_{qp}, L_{(T,p_{surf})}u, L_{q_u}, L_{o3}$, (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, $L_\zeta$.

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 $L_\zeta$, 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 \( L_\zeta \) 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:

\[
\delta(T, p_{\text{surf}}) = N \delta \zeta + P \delta \eta + \delta(T, p_{\text{surf}}) u
\]

\[
\delta \eta = (M + Q_2) \delta \zeta + \delta \eta_a + Q_1 \delta(T, p_{\text{surf}}) \tag{4.4}
\]

The matrix blocks \( M, N, P, Q_1 \) and \( Q_2 \) 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

\[
M = MH \\
N = NH \tag{4.5}
\]

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:

\[
\nabla^2 P_b = (f + \zeta) \times v_\psi + \frac{1}{2} \nabla (v_\psi \cdot v_\psi) \tag{4.6}
\]

where \( v_\psi = k \times \nabla \psi \) 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 \( Q_1 \) and \( Q_2 \) are also defined analytically, as simplified and linearised versions of the quasi-geostrophic omega equation:

\[
(\sigma \nabla^2 + f_0^2 \frac{\partial^2}{\partial p^2}) \omega' = -2 \nabla \cdot Q \tag{4.7}
\]

(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, \( \sigma \) 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 \( P \) 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 \( \delta P \) and \( \delta(T, p_{\text{surf}}) \).)
Calibration of the statistically-derived operators of the balance operator is performed outside the IFS using programs developed by François 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,

$$
\delta \tilde{q} = \delta \tilde{q}_u + Q_{qT} \frac{q_b}{q_s(T_b)} \frac{L}{R_v(T_b)^2} \delta T
$$

(4.8)

where \( \delta \tilde{q} = \delta q/q_s(T_b) \) (see below), \( q_s(T_b) \) 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 \( r_h \) and model level. \( Q_{qT} \) starts from zero in cloud free grid boxes, estimated as \( r_h < \) about 80% relative humidity, and goes to unity as \( r_h \) 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,

$$
\delta q = q_s(T_b) \delta \tilde{q}
$$

(4.9)

For the other variables \( \delta T = \delta \tilde{T} \), 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
Part II: Data Assimilation

- 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 (C_INBGSTATS) 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 Jb” 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 Jb 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 Jb 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 Jb. Within CVAR2IN all these components are handled, and the Jb 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 Jb 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 Jb 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.
Chapter 5
Observation-related processing ($J_0$)

Table of contents
5.1 Introduction
5.2 Horizontal interpolation to observation points
  5.2.1 Overview
  5.2.2 Implementation
  5.2.3 Diagnostics
5.3 Cost function
  5.3.1 Construction
  5.3.2 $J_0$ tables
5.4 Correlation of observation error
5.5 Variational quality control
  5.5.1 Description of the method
  5.5.2 Implementation
5.6 Variational bias correction
  5.6.1 Variational estimation of the bias parameters
  5.6.2 Modular implementation

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 $x$ into the equivalent of each observed quantity, $y^o$, 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 $J_0$ 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 $H = H_h H_v(x)$ 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. Recently, two-dimensional GOM-arrays have been developed, to
permit horizontal integration as required for limb-sounding and radio-occultation data. The orientation 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.

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 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

Most of the GOM code is found in the 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: GOM5 for the direct fields and GOM 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 gommod. 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.
Part II: Data Assimilation

- **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, **COBSALL** is the entry point for the horizontal interpolation of model data to observation points, called from **SCAN2M**. It is called twice, before and after the call to the gridpoint model. The first call is to get the majority of model fields at the beginning of the timestep; the second obtains some 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**, the **PREINT** routines supply the vertical profiles of model quantities at observation locations.

### 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 \( \nabla J_o \) of the first simulation, (iii) when it contains first TL perturbations after the initial call to the minimizer and (iv) when it contains \( \nabla J_o \) 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. Under **HOP**, **HDEPART**, calculates the departure \( z \) as

\[
z = y^o - Hx + (y^o - Hx^b_{HR}) - (y^o - Hx^b_{LR})
\]

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 \( z \) is

\[
z = y^o - H\delta x + (y^o - Hx^b_{HR}) - y^o
\]

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 \( \sigma_o \) (NCMFOE in ODB) to form the normalized departure.

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

The cost function is computed in **HJO**, as

\[
J_o = z^T z
\]

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**),

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

IFS Documentation – Cy41r1
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, \( \mathbf{z}_{\text{eff}} = \mathbf{z}^T \mathbf{R}^{-1} [\text{QC weight}] \), where the QC-weight will be defined below, Section 5.5.

In the adjoint routine HOPAD 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

\[
\nabla_{\text{obs}} J_{\text{O}} = -2 \frac{\mathbf{z}_{\text{eff}}}{\sigma_o} \quad (5.5)
\]

which is calculated in HOPAD for all data. The gradient of \( J_{\text{SCAT}} \) is much more complicated and is calculated in a separate section of HOPAD. The adjoint code closely follows the structure of the direct code, with the adjoint operators applied in the reverse order.

### 5.3.2 \( J_{\text{O}} \) tables

There are two different tables for storing the \( J_{\text{O}} \) values. One is purely diagnostic (JOT, yomcosjo.h), and is used for producing the printed \( J_{\text{O}} \) tables in the log-file (PRTJO called ron 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 \( J_{\text{O}} \)-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 \( ae^{-b(t_1-t_2)^2} \) where \( a = \text{RTCPART} = 0.3 \) and \( b = \text{RTCETF} = 6.0 \) 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’ \( \mathbf{z}_{\text{eff}} \) is calculated by solving the linear system of equations \( \mathbf{z}_{\text{eff}} \mathbf{R} \) for \( \mathbf{z}_{\text{eff}} \), 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 INTERCHANOBSERR_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 iterate (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

$$J_o = -\ln p$$

(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

$$p_i = N_i[1 - P(G)_i] + F_iP(G)_i$$

(5.7)

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

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

(5.8)

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

(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 $\sigma_o$. 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 $\nabla J_o^{QC}$, in terms of the normal cost function $J_o^N$

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

(5.10)

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

(5.11)

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

(5.12)

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

(5.13)

where

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

(5.14)

$$\gamma_i = \frac{P(G)_i/(2l_i)}{[1 - P(G)_i]/\sqrt{2\pi}}$$

(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 DEPART, 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.
Chapter 5: Observation-related processing \( (J_o) \)

HJO computes \( J_o^{QC} \) 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 \( w_i (\text{norm dep}_L - \text{norm dep}_HR) \), linearized in the vicinity of the high resolution cost function at \( \text{norm dep}_HR \). 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 ZFJOSHR(JOBS,JBODY) is constant during the minimisation. The cost function value is calculated like \( J_o^{varqc}_{LR} = J_o^{varqc}_{HR} + w_i (\text{norm dep}_L - \text{norm dep}_HR) \).

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 ZTCNDFHR_W and then used instead of the low resolution departures in the calculation (overwriting ZOM1DEPW 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

$$b = \sum_{i=0}^{N^j-1} \beta^j_i p^j_i$$

(5.16)

where the $p^j_i$ are the $N^j$ bias predictors associated with the group. The bias parameters $\beta^j_i$ are adjusted by the variational analysis. By convention $p^0_0 \equiv 1$, 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

$$N^p = \sum_{j=1}^{N^g} N^j$$

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

$$J(\delta x, \delta \beta) = \frac{1}{2} \delta x^T B^{-1} \delta x + \frac{1}{2} \delta \beta^T B_\beta^{-1} \delta \beta + \frac{1}{2} (H \delta x + b(\beta) - d)^T R^{-1} (H \delta x + b(\beta) - d)$$

(5.18)

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

$$\sigma^2_{\beta^j_i} = (\sigma^j)^2 / M^j, \quad i = 1, \ldots, N^j; \quad j = 1, \ldots, N^g$$

(5.19)

where $\sigma^j$ 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 $\beta^j_i$ 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 was originally designed to handle clear-sky radiance data only, with data groups consisting of individual channels for all available satellite sensors. The code was completely rewritten for Cy35r2 to facilitate implementation of bias corrections for other types of observations. The new 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.

The following class modules are included in Cy36r1:
Chapter 5: Observation-related processing (Jo)

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

---

**Variables**

- **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_airrep** temperature reports from aircraft (available in Cy36r4)
Chapter 6
Background, analysis and forecast errors

Table of contents
6.1 Nomenclature
6.2 Input and ‘massaging’ of background errors
6.3 Diagnosis of background-error variances
6.4 Calculation of eigenvalues and eigenvectors of the Hessian
6.5 The preconditioner
6.6 Calculation of analysis-error variances
6.7 Calculation of forecast-error variances
6.8 Diagnosis of background-error variances through the EDA
6.9 Diagnosis of background-error covariances through the EDA

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 6.8. 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.
Part II: Data Assimilation

Noise may be filtered by a call to FLTBGERR, which transforms the variances to spectral coefficients, multiplies each coefficient by $\cos^2(\min(n/NBGRUNC), 1)\pi/2)$, 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, CONGRAD, called from CVA1 under the control of LAVCGL. The reader is referred to Fisher (1998, ECMWF Seminar proceedings pp364-385) for a detailed description of the CONGRAD algorithm. Note that CONGRAD requires that the cost function is strictly quadratic.

CONGRAD starts by transforming the initial control variable and gradient to a space with Euclidian inner product. Typically, this transformation is simply a multiplication by YRSCALPSQRT, but may also involve preconditioning via calls to PRECOND. The transformed initial gradient is normalized to give the first Lanczos vector. The Lanczos vectors are stored in the array YL_ZCGLWK.

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 minimize the cost function is not explicitly generated. Thus, unlike e.g. MIQN3, the control vectors passed to SIM4D do not lie on a path towards the minimum, and the gradients returned by SIM4D do not, in general, decrease as the minimization proceeds. The optimal control vector and the corresponding gradient can, however, be determined as a linear combination of the Lanczos vectors.

In general, only the gradient is calculated at each iteration. It is required in order to monitor the convergence of the minimization. Optionally, by setting L_CHECK_GRADIENT, the optimal point is also calculated, and an additional call to SIM4D is made to evaluate the true gradient at the optimal point. This is useful as a means of checking the assumption of linearity on which the algorithm is based. However, it doubles the cost of the minimization.

The Lanczos algorithm produces a sequence of coefficients, which are stored in the arrays ZDELT and ZBETA. 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 internal subroutine PT3V, 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.

A call to PREDICT, RUNTIME is made every iteration. This subroutine attempts to predict how many iterations will be required to minimize the cost function, and how long this is likely to take. These predictions are written to STDERR.

After the last iteration, the converged eigenvectors of the Hessian are calculated by calling WREVECS.

Note that the criterion used to decide which eigenvalues have converged is relaxed at this stage to \( \| J''v - \lambda v \| < \varepsilon \| v \| \), where \( \varepsilon \) is given by EVBCGL. The default value for EVBCGL is 0.1.

Next, if required, upper and lower bounds for \( N - p^T(J'')^{-1}p \) (where \( N \) is the dimension of the control vector) and \( -p^T \log_2[(J'')^{-1}]p \) are calculated using the algorithm of Golub and Meurant (1994). These quantities may be used to evaluate the information content of the analysis using the switch L,INFO,CONTENT.

Finally, CONGRAD calculates the the optimal control vector and gradient as a linear combination of the Lanczos vectors, and transforms them from the Euclidian space used internally to the usual space of the control variable.

### 6.5 THE PRECONDITIONER

CONGRAD allows the use of a preconditioner. The preconditioner is a matrix which approximates the Hessian matrix of the cost function. The preconditioner used in CONGRAD is a matrix of the form

\[
I + \sum_{i=1}^{L} (\mu_i - 1) w_i w_i^T
\]  

(6.1)

where the vectors \( w_i \) are orthogonal. The pairs \( \{ \mu_i, w_i \} \) are calculated in PREPPCM, 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, \( u_i \), is read in using READVEC. These vectors may be in the space of the control vector (if LEVECCNTL is true), or in model space (if LEVECCNTL is false). In the latter case, the vectors are transformed to control space by calls to CHAVAR.

The vectors (transformed, if necessary to control space) are assumed to satisfy

\[
I - \sum_{i=1}^{L} u_i u_i^T \approx (J'')^{-1}
\]  

(6.2)

Vectors which meet this criterion can be written out from an earlier forecast error calculation by setting LWRIEVEC.

The input vectors are not necessarily orthogonal, whereas the preconditioner requires a set of orthogonal vectors. Let us denote by \( U \) the matrix whose columns are the vectors \( u_i \). A sequence of Householder transformations is now performed to transform \( U \) to upper triangular. Let us represent this sequence of Householder transformations by the matrix \( Q \). Then \( QU \) is upper triangular, which means that \( (QU)(QU)^T \) is zero except for an \( L \times L \) block in the top left hand corner.

It is clear that \( (QU)(QU)^T \) 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 \( (QU)(QU)^T \) 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 \( Q \) is an orthogonal matrix, we have \( QQ^T = I \). So, we may write (6.2) as

\[
I - Q^T(QU)(QU)^TQ \approx (J'')^{-1}
\]  

(6.3)
Let us denote the eigenpairs of \((\mathbf{QU})(\mathbf{QU})^T\) by \(\{\rho_i, \mathbf{v}_i\}\). Then we may write (6.3) as
\[
\mathbf{I} - \sum_{i=1}^{L} \rho_i (\mathbf{Q}^T \mathbf{v}_i)(\mathbf{Q}^T \mathbf{v}_i)^T \approx (J'')^{-1}
\] (6.4)
The orthogonality of \(\mathbf{Q}\) and the orthonormality of the eigenvectors \(\mathbf{v}_i\), means that the vectors \(\mathbf{Q}^T \mathbf{v}_i\) are orthonormal. They are, in fact, the required vectors, \(\mathbf{w}_i\) of the preconditioner matrix.

Inverting (6.4) gives
\[
\mathbf{I} - \sum_{i=1}^{L} \frac{1}{\rho_i} \mathbf{w}_i \mathbf{w}_i^T \approx J''
\] (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 \(\mu_i\), 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 \(\mu_i\) to a maximum of \(R\cdot\text{MAX}\cdot\text{CNUM}\cdot\text{PC}\) (typically 10). This was found to be necessary in practice to avoid ill-conditioning the minimization.

The numbers \(\mu_i\) are stored in RCGLPC. The vectors, \(\mathbf{w}_i\) are stored in YVCGLPC, and the total number of preconditioner vectors is stored in NVCGLPC.

Application of the preconditioner is straightforward, and is performed by subroutine PRECOND. This routine can also apply the inverse, the symmetric square root, or the inverse of the symmetric square root of the preconditioner matrix. Application of the latter matrices relies on the observation that if
\[
\mathbf{M} = \mathbf{I} + \sum_{i=1}^{L} (\mu_i - 1) \mathbf{w}_i \mathbf{w}_i^T
\] (6.6)
with orthonormal \(\mathbf{w}_i\), then the expressions for \(\mathbf{M}^{-1}\), \(\mathbf{M}^{1/2}\) and \(\mathbf{M}^{-1/2}\) result from replacing \(\mu_i\) in (6.6) by \(1/\mu_i\), \(\sqrt{\mu_i}\) and \(1/(\sqrt{\mu_i})\) respectively.

### 6.6 CALCULATION OF ANALYSIS-ERROR VARIANCES

The eigenvectors and eigenvalues of the Hessian matrix calculated by CONGRAD are passed to XFORMEV, which uses them to estimate the analysis-error variances.

The first step is to undo any preconditioning. If preconditioning has been employed, then the eigenvectors and eigenvalues produced by CONGRAD provide an approximation to the preconditioned Hessian, \(\mathbf{M}^{-1/2}J''\mathbf{M}^{-1/2}\), of the form
\[
\mathbf{M}^{-1/2}J''\mathbf{M}^{-1/2} \approx \mathbf{I} + \sum_{i=1}^{K} (\lambda_i - 1) \mathbf{v}_i \mathbf{v}_i^T
\] (6.7)
Multiplying to the left and right by \(\mathbf{M}^{1/2}\), gives
\[
J'' \approx \mathbf{M} + \sum_{i=1}^{K} (\lambda_i - 1) (\mathbf{M}^{1/2} \mathbf{v}_i)(\mathbf{M}^{1/2} \mathbf{v}_i)^T
\] (6.8)
Substituting for the preconditioner matrix from (6.6), gives the following
\[
J'' \approx \mathbf{I} + \sum_{i=1}^{L+K} \mathbf{s}_i \mathbf{s}_i^T
\] (6.9)
where
\[
\mathbf{s}_i = \begin{cases} 
(\mu_i - 1)^{1/2} \mathbf{w}_i & \text{for } i = 1 \ldots L \\
(\lambda_i - L)^{1/2} \mathbf{M}^{1/2} \mathbf{v}_{i-L} & \text{for } i = L + 1 \ldots L + K 
\end{cases}
\] (6.10)
Note that the resulting approximation to the Hessian is not expressed in terms of eigenvalues and eigenvectors. Consequently, inversion of the approximation must be performed using the Shermann–Morrison–Woodbury formula. Let \( S \) be the matrix whose columns are the vectors \( s_i \). Then, according to the Shermann–Morrison–Woodbury formula, the inverse of the approximate Hessian matrix is

\[
(J'')^{-1} \approx I - S(I + S^T S)^{-1} S^T
\]  

(6.11)

The matrix \((I + S^T S)\) has dimension \((L + K) \times (L + K)\). This matrix is constructed, and its Cholesky decomposition is calculated using the LAPACK routine SPOTRF. This gives a lower triangular matrix \( C \) such that

\[
(J'')^{-1} \approx I - (SC^{-1})(SC^{-1})^T
\]  

(6.12)

The final stage in the calculation of the analysis errors is to transform the columns of the matrix \((SC^{-1})\) to the space of model variables by applying the inverse change of variable, \( CHAVARIN \). This gives the required approximation to the analysis error covariance matrix

\[
P_a \approx B - VV^T
\]  

(6.13)

where \( V = L^{-1}SC^{-1} \), and where \( L^{-1} \) represents the inverse of the change of variable. The columns of \( V \) may be written out (e.g. for diagnostic purposes, or to form the preconditioner for a subsequent minimization) by setting LWRIEVEC. The columns of \( V \) are then transformed to gridpoint space, and their sums of squares (i.e. the diagonal elements of \( VV^T \) in gridpoint space) are subtracted from the variances of background error which were stored in ANEBUF before the minimization by BGVECS.

The analysis errors are calculated as the difference between the background errors and a correction derived from the eigenvectors of the Hessian. If the background errors are underestimated, there is a danger that the correction will be larger than the background error, giving negative variances of analysis error. This is unlikely to happen if the background errors are estimated using randomization, or for variables whose background errors are explicitly specified in the background cost function, but is possible for variables such as temperature whose background errors are not explicitly specified. To guard against this eventuality, if NBGVECS is zero, then the variances of analysis error for variables whose background errors are not explicit are estimated by applying a scaling to the explicit variables by a call to ESTSIGA from CV1.

The analysis errors are inflated according to the error growth model of Savijärvi (1995) to provide estimates of short-term forecast error. This is done by a call to ESTSIG.

The error growth model is

\[
\frac{d\sigma}{dt} = (a + b\sigma) \left( 1 - \frac{\sigma}{\sigma_{\infty}} \right)
\]  

(6.14)

Here, \( a \) represents growth due to model errors, \( b \) represents the exponential growth rate of small errors, and \( \sigma_{\infty} \) represents the standard deviation of saturated forecast errors. The growth due to model error is set to 0.1 times the global mean background error per day. The exponential growth rate, \( b \), is set to 0.4 per day.

The saturation standard deviations are calculated as \( \sqrt{2} \) times the standard deviation of each field. The standard deviations have been calculated for each month from the re-analysis dataset. ESTSIG reads these climatological error fields from file ‘stdev_of_climate’ by calling READGRIB, and interpolates them in the horizontal and vertical using SUHIFCE and SUVIFCE. The climatological errors may also be artificially increased in the tropics under the control of LFACHRO. If climate standard deviations are not available for any field, they are estimated as 10 times the global mean background error for the field.

The error growth model is integrated for a period of NFGFCLEN hours. The integration is done analytically using the expression given by Savijärvi (1995). Two precautions are taken in integrating the error growth model. First, negative analysis-error variances are set to zero. Second, the growth rate
due to model error is limited to a sensible value with respect to the saturation errors. This was found to be necessary to prevent numerical problems when calculating specific humidity errors for the upper levels of the model.

ESTSIG overwrites the contents of ANEBUF with the estimated variances of forecast error. The variances are converted to standard deviations and written out by WRITESD.

6.8 Diagnosis of Background-Error Variances Through the EDA

The EDA consists of an ensemble of 25 (as of Cy40r1) independent lower-resolution 4D-Var assimilations (T399L91 with two inner loops at T95 and T159) that differ by the explicit perturbation of the observations, the sea-surface temperature and the model physics, plus 1 control 4D-Var which runs without any perturbation. 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 as currently used in the Seasonal Forecasting System (Vialard et al., 2005). 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).

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 Cy38r2, 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. In this way EDA online estimates for the error fields for the full control vector are now available for use 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 performed against the ECMWF operational analysis separately for each parameter, model level and geographical region (northern extra-tropics, tropics, southern extra-tropics).

Variance calibration is done to ensure that EDA background variance matches the mean squared difference of the EDA background mean and the corresponding operational analysis. 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.

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 spectral representation and filtered by Spectral_Filter in order to reduce sampling noise. The filtering procedure is based on Bonavita et al. (2012b). Finally the filtered spectral variances are transformed back to a gridpoint representation, converted to standard deviations and stored as grib files of type ‘ses’ of stream ‘elda’.
Chapter 6: Background, analysis and forecast errors

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.

6.9 DIAGNOSIS OF BACKGROUND-ERROR COVARIANCES THROUGH THE EDA

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. A new set of "wavelet" B matrices (see section 4.2) is computed at each assimilation cycle. In the initial implementation of cycle 40R1 lagged EDA background perturbations over the most recent 12 days were used in the computation. In cycle 40R3 this has been modified: the training dataset of EDA background perturbations is constructed as a combination of perturbations from the latest EDA background forecast verifying at or near the nominal background time, and augmenting it with a climatology of EDA background forecasts. With the current ensemble size of the EDA (25 members), climatological perturbations make up two thirds of the training dataset: this percentage is expected to be reduced when larger ensembles will be affordable.

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 three separate ifsm_in 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
7.2 Normal-mode initialization
7.3 Computation of normal modes
7.3.1 Vertical modes
7.3.2 Horizontal modes and help arrays
7.4 Implementation of NMI
7.5 Computation of $J_c$ based on NMI
7.6 Digital filter initialization
7.7 Implementation of DFI as a weak constraint in 4D-Var

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 $x$. Thus, the model state can be written as

$$x = x_R + x_G$$  \hspace{1cm} (7.1)

where $x_R$ is the ‘slow’ component and $x_G$ the ‘fast’ component. Linear NMI consists of removing the fast component altogether ($x_G = 0$). Since the model is nonlinear, a much better balance is obtained by setting the tendency of the fast component to zero ($\dot{x}_G = 0$); 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 NFILTMI (= 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 significant 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

$$J_c = \varepsilon \| (\dot{x} - \dot{x}_b) G \|^2$$  \hspace{1cm} (7.2)

where $\varepsilon$ is an empirically chosen weighting factor, $x$ is the current state of the control variable and $x_b$ 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 NMJCTL, 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 $f_k = k\Delta t$, $\Delta t$ being the timestep. This proceeds by doing a convolution of $f(t)$ with a step function $h(t)$ so that

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

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

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

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

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

where $1/x_0 = \cos(\pi \Delta t)/\tau_s$, $1/r = \cosh(2M \cosh x_0)$, $\theta_k = (k\pi)/M$ 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

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

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
  8.1.2 How to compute the observation influence (OI)
8.2 Observational influence for a DA scheme
  8.2.1 Linear statistical estimation in numerical weather prediction
  8.2.2 Approximate calculation of self-sensitivity in a large variational analysis system
8.3 Gradient test
8.4 The 24-hour forecast error contribution to observations (FEC)

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
\[
\mathbf{x}_a = \mathbf{K} \mathbf{y} + (\mathbf{I}_n - \mathbf{KH})\mathbf{x}_b
\]  
(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 $R$, with

$$K = (R^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1}$$  \hspace{1cm} (8.2)$$

Here, $H$ 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 model’s temperature, humidity and ozone into brightness temperatures as observed by several satellite instruments). In the 4D-Var context introduced below, $H$ 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

$$\hat{y} = Hx_a = HKy + (I_p - HK)Hx_b$$  \hspace{1cm} (8.3)$$

It can be seen that the analysis state in observation space ($Hx_a$) is defined as a sum of the background (in observation space, $Hx_b$) and the observations $y$, weighted by the $p \times p$ square matrices $I - HK$ and $HK$, respectively.

In this case, for each unknown component of $Hx$, 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 $Hx_b$. From (8.3), the analysis sensitivity with respect to the observations is obtained

$$S = \frac{\partial \hat{y}}{\partial y} = K^T H^T$$  \hspace{1cm} (8.4)$$

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

$$\frac{\partial \hat{y}}{\partial Hx} = I - K^T H^T = I_p - S$$  \hspace{1cm} (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 $i$th observation is $S_{ii}$, the sensitivity with respect to the background projected at the same variable, location and time will be simply $I - S_{ii}$. It also follows that the complementary trace, $\text{tr}(I - S) = p - \text{tr}(S)$, 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 $\text{tr}(S)$. 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 $R$ is diagonal, $0 \leq S_{ii} \leq 1$ is assured, but for more general non-diagonal $R$-matrices it is easy to find counter-examples to that property.

Inserting (8.1) into (8.4), we obtain

$$S = R^{-1}H(B^{-1} + H^T R^{-1} H)^{-1}H^T$$  \hspace{1cm} (8.6)$$

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

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

In an optimal variational analysis scheme, the analysis error covariance matrix $A$ is approximately the inverse of the matrix of second derivatives (the Hessian) of the cost function $J$, i.e. $A = (J)^{-1}$ \cite{Rabier_and_Courtier_1992}. Given the large dimension of the matrices involved, $J$ 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 $P$, $P = L^{-1}(x - x_0)$, with $L$
chosen such that \( B^{-1} = L^T L \). The transformation \( L \) thus reduces the covariance of the prior to the identity matrix. In variational assimilation \( L \) is referred to as the change-of-variable operator (Courtier et al., 1998).

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

The summation in (8.7) approximates the variance reduction \( B-A \) due to the use of observations in the analysis. \((\lambda_i, v_i)\) are the eigen-pairs of \( A \). 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 \( B \) in (8.7) is also computed approximately, using the randomisation method proposed by Fisher and Courtier (1995). From a sample of \( N \) random vectors \( u_i \) (in the space of the control-vector), drawn from a population with zero mean and unit Gaussian variance, a low-rank representation of \( B \) (in terms of the atmospheric state variables \( x \)) is obtained by

\[
B = \frac{1}{N} \sum_{i=1}^{N} (L u_i)(L u_i)^T
\]  
(8.8)

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

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

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

Only the diagonal elements of \( S \) 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 \( j \)th observation to the analysis at the \( i \)th 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 \( S_{ij} \) will tend to be over-estimates. For the extreme case \( M = 0 \), (8.9) gives \( S = R^{-1} HBH^T \) which in particular can have diagonal elements larger than one if elements of \( HBH^T \) are larger than the corresponding elements of \( R \). The number of Hessian vectors operationally computed is \( M = 40 \) and the number of random \( B \) 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 eigen-vectors 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 $\text{LTEST} = \text{.TRUE.}$ a gradient test will be performed both before and after minimization. This is done by the routine $\text{GRTEST}$. 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

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

(8.10)

with $\delta \chi = -\alpha \nabla J$. Repeatedly increasing $\alpha$ 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 $\alpha$ (initially as well as after minimization) proves that the coded adjoint is the proper adjoint for the linearization around the given state $\chi$.

The behaviour of the cost function in the vicinity of $\chi$ in the direction of the gradient $\nabla J$ is also diagnosed by several additional quantities for each $\alpha$. 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

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

(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)

For information about the FEC computation see:

http://datasvc.ecmwf.int/twiki/bin/view/Main/DiagnosticsToolsDataAssimilation.
Chapter 9
Land-surface analysis

Table of contents

9.1 Introduction
9.2 2D Optimal Interpolation screen-level analysis
  9.2.1 Methodology
  9.2.2 Quality controls
  9.2.3 Technical aspects
9.3 2D Optimum Interpolation snow analysis
  9.3.1 Background and observations
  9.3.2 Methodology
  9.3.3 Quality controls
  9.3.4 Technical aspects
9.4 Simplified Extended Kalman Filter Soil moisture analysis
  9.4.1 Methodology
  9.4.2 Quality controls
  9.4.3 Technical aspects
9.5 1D OI soil and snow temperature analysis

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 is based on a 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 MetOp-A and MetOp-B Advanced Scatterometer (ASCAT) soil moisture data is also assimilated. 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 simplified EKF soil moisture analysis, while screen-level analysis temperature increments are used as inputs of the 1D OI soil and snow temperature analysis.
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.

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 $\Delta X_i$ are estimated at each observation location $i$.

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

$$\Delta X^a_p = \sum_{i=1}^{N} W_i \times \Delta X_i$$  \hspace{1cm} (9.1)

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

$$(B + O)W = b$$  \hspace{1cm} (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 horizontal correlation coefficients (structure functions) of $\mathbf{b}$ and $\mathbf{B}$ are assumed to have the form

$$
\mu(i, j) = \exp\left( -\frac{1}{2} \left( \frac{r_{ij}}{d} \right)^2 \right)
$$

(9.3)

where $r_{ij}$ is the horizontal separation between points $i$ and $j$, and $d$ is a parameter defined to be 300 km, corresponding to the e-folding distance taken of 420 km.

Therefore

$$
B(i, j) = \sigma_b^2 \times \mu(i, j)
$$

(9.4)

with $\sigma_b$ the standard deviation of background errors.

The covariance matrix of observation errors $\mathbf{O}$ is set to $\sigma_o^2 \times \mathbf{I}$ where $\sigma_o$ 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

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

(9.5)

where $\gamma$ 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 the 2-metre temperature analysis $T_a$ to perform the conversion so that

$$
T_d = \frac{17.502 \times 273.16 - 32.19 \times \Psi}{17.05 - \Psi}
$$

(9.6)

with

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

(9.7)

### 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 INT2M 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 \(q = (B + O)^{-1}\Delta X\) (in subroutine EQUISOLVE – inversion of a linear system), which does not depend upon the position of the analysis grid point, and then estimating \(b^T q\) (in subroutine DOT_PRODUCT).

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

(i) \(\text{C}_{\text{SSA}_{\text{TYPE}}}: \text{‘t2m’ for temperature analysis and ‘rh2m’ for relative humidity analysis.}\)
(ii) \(\text{L}_{\text{OI}}: \text{.TRUE. for statistical interpolation and .FALSE. for Cressman interpolation.}\)
(iii) \(\text{N}_{\text{OISET}}: \text{number of observations (parameter } N)\).
Part II: Data Assimilation

(iv) SIGMAB: standard deviation of background error (parameter $\sigma_b$).
(v) SIGMAO: standard deviation of observation error (parameter $\sigma_o$).
(vi) TOL$_{RH}$: Tolerance criteria for RH observations (parameter $\gamma$ in (9.5)).
(vii) TOL$_T$: Tolerance criteria for T observations (parameter $\gamma$ in (9.5)).
(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. The snow-depth background $S^b$ (units: m) is estimated from the short-range forecast of snow water equivalent $SWE^b$ (units: m of water equivalent) and snow density $\rho_b$ (units: kg m$^{-3}$). It is given by

$$S^b = \frac{1000 \times SWE^b}{\rho_b^b} \quad (9.8)$$

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). 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.

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 $p$:

$$\Delta S^a_p = \sum_{i=1}^{N} W_i \times \Delta S_i \quad (9.9)$$

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

$$(\mathbf{B} + \mathbf{O})\mathbf{W} = \mathbf{b} \quad (9.10)$$

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

$$\mu_{ij} = \alpha(r_{ij})\beta(\Delta z_{ij}) \quad (9.11)$$

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

$$\alpha(r_{ij}) = (1 + \frac{r_{ij}}{L}) \exp(-\frac{r_{ij}}{L}) \quad (9.12)$$
Chapter 9: Land-surface analysis

Figure 9.3 Horizontal structure functions used in the Optimal Interpolation scheme for the screen level parameters and snow depth analyses.

$L$ is the horizontal length parameter taken to 55000m, corresponding to an e-folding distance taken to 120 km.

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

(9.13)

$h$ is the vertical length scale taken to 800 m. The two parameters $L$ 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 is represented in Fig. 9.3.

The correlation matrix $B$ is expressed as:

$$B(i, j) = \sigma_b^2 \times \mu(i, j)$$

(9.14)

with $\sigma_b$ the standard deviation of background errors.

The covariance matrix of observation errors $O$ is set to $\sigma_o^2 \times I$ where $\sigma_o$ is the standard deviation of observation errors and $I$ 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 final snow water equivalent product $SWE_a^s$ is then calculated using the analysed snow depth $S^a$:

$$SWE_a^s = \frac{\rho_b \times S^a}{1000}$$

(9.15)

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 8.4.2.
- In the other cases, the snow density remains unchanged.

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 ($SWE_a^s = 10$ m).
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

\[ |\Delta S_i| > \gamma \sqrt{\sigma_s^2 + \sigma_b^2} \]  

(9.16)

where \( \gamma \) 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 \( T_{2m} < 8^\circ C \) only snow-depth observations below 140 cm are accepted.
(ii) This limit is reduced to 70 cm if \( T_{2m} > 8^\circ C \).
(v) Snow-depth analysis is limited to 140 cm.
(vi) Snow-depth increments are set to zero when larger than \( (160 - 16 T_{2m}) \) mm (where \( T_{2m} \) 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 \( S_{\text{clim}} \) so that the final analysis is provided by

\[ S^a = (1 - \alpha)S^a + \alpha S_{\text{clim}} \]  

(9.17)

The relaxation coefficient \( \alpha \) 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 (9.9) and (9.14). This increment is 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) \texttt{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 \texttt{NAMSSA}).

(iv) \texttt{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) \texttt{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 \texttt{NAMSSA}. In addition to those defined in Section 9.2.3

(vi) \texttt{TOL\_SN}: Tolerance criteria for snow depth observations (parameter $\gamma$ in 9.16).

(viii) \texttt{RSCALE\_X}: horizontal distance $L$ used in the horizontal structure function in (9.12), set to 55000 m.

(ix) \texttt{RSCALE\_Z}: vertical distance $h$ used in the vertical structure function in (9.13), 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 $x^a$ is computed at time $t_i$ for each grid point as:

$$x^a(t_i) = x^b(t_i) + K_i \left[ y^o(t_i) - H_i(x^b) \right]$$ (9.18)

with superscripts $a$, $b$, $o$ standing for background, analysis and observations, respectively, $x$ the model state vector, $y$ the observation vector and $H$ the non-linear observation operator. The Kalman gain matrix $K_i$ is computed at time $t_i$ as:

$$K_i = \left[ P^{-1} + H_i^T R_i^{-1} H_i \right]^{-1} H_i^T R_i^{-1}$$ (9.19)

where $H_i$ is the linearised observation operator, $P$ is the error covariance matrix associated with $x$ and $R$ is the observation errors covariance matrix.

The background error covariance matrix $P$ and the observation error matrix $R$ are static, with diagonal terms composed of error variances. These terms are based on 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 set to $\sigma_{ascat} = 0.05 m^3 m^{-3}$.

The linearisation of the observation operator is computed in finite differences, by using individual perturbations of the model state vector by a small amount $\delta x_n$ of the $n^{th}$ component of the model state vector. One perturbed simulation is required for each element of the control state vector. For each perturbed simulation, initial background state vector is perturbed by a vector $\delta x^b_n$ that contains $\delta x_n$ for the perturbed $n^{th}$ element and zero for all the others elements. Using index $m$ to represent the $m^{th}$ element of the observations vector, the Jacobian elements $H_{mn,i}$ of the observation operator at time $t_i$ is written as:

$$H_{mn,i} = \frac{H_{m,i}(x^b + \delta x^b_{n}) - H_{m,i}(x^b)}{\delta x_n}$$ (9.20)

The model state vector evolution from time $t_i$ to time $t_{i+1}$ is then controlled by the equation:

$$x^b(t_{i+1}) = M_i [x^a(t_i)]$$ (9.21)

with $M$ the non-linear state forecast model.

In the current implementation, the state vector is soil moisture. It has dimension $n_{max} = 3$ since the first three layers of the HTESSEL LSM are analysed. The observations vector $y$ includes 2-metre
temperature and relative humidity analyses as well as ASCAT-A and ASCAT-B rescaled soil moisture. When 12-hour assimilation windows are used, $y$ has dimension $m_{max} = 4$ 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 the soil moisture analysis is locally switched off if the Jacobians become larger than $50K/m^2m^{-3}$, $5/m^3m^{-3}$ and $2m^3m^{-3}/m^3m^{-3}$ for $T_{2m}$, $RH_{2m}$ and ASCAT soil moisture components, respectively. In addition, the analysis is 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)
- Soil moisture perturbation $\delta x_n$ for each layer, VSM_PERT_INC, set to $0.01 m^3m^{-3}$ in the current configuration,
- Background soil moisture error, BACK_ERR=0.0001 $m^6m^{-6}$, corresponds to $\sigma_b = 0.01m^3m^{-3}$
- Screen level analysis errors, $T_{2m}$, $RH_{2m}$ and ASCAT_ERR=0.0025, corresponds to $\sigma_{ASCAT} = 0.05m^3m^{-3}$.
- 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.

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 the main control program of the simplified EKF analysis. It defines the perturbed soil moisture initial conditions, by calling PERTSEKF_V2. Then it does a loop on the control and the perturbed runs with one CTN4 call for each of the control and perturbed cases. Resulting soil moisture and screen level parameters fields are stored from the routine STORE_SEKF.CV which is called at lower level from CALLPAR and EC_PHYS. After this loop on CTN4 call, all the model soil moisture and screen level conditions that will enter the equation 9.18 are stored.

OBSGEN and SEKF_PREP_ASCAT are the routine that prepare the observations to be used in the simplified EKF analysis. Then CTN4 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 9.18 to 9.21 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. 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 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 temperature and snow temperature:

$$\Delta T = c \times (T_a - T_b)$$  \hspace{1cm} (9.22)

with $T_a$ and $T_b$ the analysed and model first-guess temperatures, respectively. The coefficient $c$ providing the analysis increments is:

$$c = (1 - F_1)F_3$$  \hspace{1cm} (9.23)
Figure 9.4 Program organisation for the simplified EKF soil moisture analysis (NCONF=302).

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

$$F_1 = \frac{1}{2} \left\{ 1 + \tanh(\lambda(\mu_M - 0.5)) \right\} \quad \lambda = 7 \quad (9.24)$$

The empirical function $F_3$ reduces increments over mountainous areas so that

$$F_3 = \begin{cases} 0 & Z > Z_{\text{max}} \\ \left( \frac{Z - Z_{\text{max}}}{Z_{\text{min}} - Z_{\text{max}}} \right)^2 & Z_{\text{min}} < Z < Z_{\text{max}} \\ 1 & Z < Z_{\text{min}} \end{cases} \quad (9.25)$$

where $Z$ is the model orography, $Z_{\text{min}} = 500$ m and $Z_{\text{max}} = 3000$ 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.
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 interpolations to the model grid of daily global datasets provided by The Metoffice, with backup from NCEP in case OSTIA is not available. The Metoffice OSTIA product is described by Stark et al. (2007) 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 NCEP RTG real-time global SST product is described by http://polar.wwb.noaa.gov.sst and is available in the ECMWF archives from 20010130. The only lake information in these two products is for the Caspian Sea, with the addition of the Great Lakes in NCEP RTG. For other lakes, climatological values are used.

10.2 CHOICE OF PRODUCT

Both the OSTIA and the NCEP products are read in if available. If both are available, OSTIA is used for the oceans and the Caspian Sea and the Great Lakes are taken from NCEP. If NCEP is missing, no new information is used for the Great Lakes. If OSTIA is missing, NCEP data are used everywhere. If both are missing, yesterday’s fields will be used.

Since the OSI-SAF sea-ice product does not include ice coverage over the Caspian Sea and Sea of Azov, the NCEP sea-ice product is used over these areas between 15 September and 1 May (from 2011).

10.3 ECMWF RE-SAMPLING 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. The NCEP RTG product comes at 0.5 degree resolution and is bi-linearly interpolated to to the models reduced Gaussian grid. For both products checks are made that the values lie within realistic values. The temperature can not be lower than 271.46K when sea ice is below 20%, and the temperature is set to 273.16K when sea ice is above 20%. For temperatures above 274.26K, the sea ice concentration is set to zero.
Chapter 11

Data flow

Table of contents

11.1 Notation
11.2 Data assimilation cycling
11.3 Overview of 4D-Var data flow
11.4 Input GRIB fields
   11.4.1 reftrajshml
   11.4.2 reftrajgglm
   11.4.3 reftrajggsfc
   11.4.4 errgrib
   11.4.5 spmoderr
   11.4.6 wamspecwavin
   11.4.7 wamcdwavein
   11.4.8 wamnwavein
   11.4.9 wamsfcwindin
11.5 Input observation data
11.6 Input data files
   11.6.1 Invariant data files
   11.6.2 Date-dependent data files
11.7 Output GRIB fields
   11.7.1 Output type ‘an’ model level spectral fields
   11.7.2 Output type ‘an’ model level Gaussian grid-point fields
   11.7.3 Output type ‘an’ surface Gaussian grid-point fields
   11.7.4 Output type ‘an’ wave model fields
   11.7.5 Output error fields
   11.7.6 Output model error fields
11.8 Output observation data
11.9 Output data files
11.10 Sea surface temperature analysis
    11.10.1 Input GRIB fields on model grid
    11.10.2 Input OSTIA and NCEP fields
    11.10.3 Input data files
    11.10.4 Output GRIB fields on model grid
11.11 2 metre temperature analysis
    11.11.1 Input GRIB fields on model grid
    11.11.2 Input observation data
    11.11.3 Output GRIB field on model grid
11.12 2 metre relative humidity analysis
    11.12.1 Input GRIB fields on model grid
    11.12.2 Input observation data
    11.12.3 Output GRIB field on model grid
11.13 Snow analysis
    11.13.1 Input GRIB fields on model grid
    11.13.2 Input data files
11.13.3 Input observation data
11.13.4 Output GRIB fields on model grid

11.14 Soil moisture analysis
11.14.1 Input GRIB fields on model grid
11.14.2 Output GRIB fields on model grid
11.14.3 Invariant climatological fields

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>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>${DATA}$</td>
<td>Data directory for invariant files</td>
<td></td>
</tr>
<tr>
<td>${GTYPE}$</td>
<td>Gaussian grid type</td>
<td>L2 (ie linear reduced Gaussian grid)</td>
</tr>
<tr>
<td>${IFS_CYCLE}$</td>
<td>IFS cycle name</td>
<td>CY36R1</td>
</tr>
<tr>
<td>${LEVELS}$</td>
<td>Number of vertical model levels</td>
<td>91</td>
</tr>
<tr>
<td>${MM}$</td>
<td>Month</td>
<td></td>
</tr>
<tr>
<td>${RESOL}$</td>
<td>Spectral truncation</td>
<td>1279</td>
</tr>
<tr>
<td>${starttime}$</td>
<td>Start of 4D-Var window as yyyymmddhh</td>
<td></td>
</tr>
<tr>
<td>${WDIR}$</td>
<td>Work directory (1 for each cycle)</td>
<td></td>
</tr>
</tbody>
</table>

In this chapter, the notation illustrated in Fig. 11.1 is used in diagrams to distinguish between computation steps and 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 delayed cut-off
Part II: Data Assimilation

Figure 11.2 Data assimilation cycling with the Early Delivery configuration.

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
Figure 11.3 4D-Var data flow.

Binary. 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’, ensemble 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.

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

## 11.4.2 reftrajggml

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Levels</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>133</td>
<td>Q</td>
<td>Specific humidity</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>203</td>
<td>O3</td>
<td>Ozone mass mixing ratio</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>75</td>
<td>CRWC</td>
<td>Cloud rain water content</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>76</td>
<td>CSWC</td>
<td>Cloud snow water content</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>246</td>
<td>CLWC</td>
<td>Cloud liquid water content</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>247</td>
<td>CIWC</td>
<td>Cloud ice water content</td>
<td>kg kg$^{-1}$</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
<tr>
<td>248</td>
<td>CC</td>
<td>Cloud cover</td>
<td>(0-1)</td>
<td>1-LEVELS</td>
<td>fc</td>
</tr>
</tbody>
</table>
### 11.4.3 reftrajggsfc

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>CI</td>
<td>Sea-ice cover</td>
<td>(0-1)</td>
<td>fc</td>
</tr>
<tr>
<td>032</td>
<td>ASN</td>
<td>Snow albedo</td>
<td>(0-1)</td>
<td>fc</td>
</tr>
<tr>
<td>033</td>
<td>RSN</td>
<td>Snow density</td>
<td>kg m$^{-3}$</td>
<td>fc</td>
</tr>
<tr>
<td>034</td>
<td>SST</td>
<td>Sea surface temperature</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>035</td>
<td>ISTL1</td>
<td>Ice surface temperature, layer 1</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>036</td>
<td>ISTL2</td>
<td>Ice surface temperature, layer 2</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>037</td>
<td>ISTL3</td>
<td>Ice surface temperature, layer 3</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>038</td>
<td>ISTL4</td>
<td>Ice surface temperature, layer 4</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>039</td>
<td>SWVL1</td>
<td>Volumetric soil water, layer 1</td>
<td>m$^3$ m$^{-3}$</td>
<td>fc</td>
</tr>
<tr>
<td>040</td>
<td>SWVL2</td>
<td>Volumetric soil water, layer 2</td>
<td>m$^3$ m$^{-3}$</td>
<td>fc</td>
</tr>
<tr>
<td>041</td>
<td>SWVL3</td>
<td>Volumetric soil water, layer 3</td>
<td>m$^3$ m$^{-3}$</td>
<td>fc</td>
</tr>
<tr>
<td>042</td>
<td>SWVL4</td>
<td>Volumetric soil water, layer 4</td>
<td>m$^3$ m$^{-3}$</td>
<td>fc</td>
</tr>
<tr>
<td>139</td>
<td>STL1</td>
<td>Soil temperature level 1</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>141</td>
<td>SD</td>
<td>Snow depth</td>
<td>m of water equivalent</td>
<td>fc</td>
</tr>
<tr>
<td>148</td>
<td>CHNK</td>
<td>Charnock parameter</td>
<td></td>
<td>fc</td>
</tr>
<tr>
<td>170</td>
<td>STL2</td>
<td>Soil temperature level 2</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>183</td>
<td>STL3</td>
<td>Soil temperature level 3</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>198</td>
<td>SRC</td>
<td>Skin reservoir content</td>
<td>m of water</td>
<td>fc</td>
</tr>
<tr>
<td>235</td>
<td>SKT</td>
<td>Skin temperature</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>236</td>
<td>STL4</td>
<td>Soil temperature level 4</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>237</td>
<td>FSR</td>
<td>Surface roughness</td>
<td>m</td>
<td>fc</td>
</tr>
<tr>
<td>245</td>
<td>FLSR</td>
<td>Logarithm of surface roughness</td>
<td>m</td>
<td>fc</td>
</tr>
<tr>
<td>015</td>
<td>ALUVP</td>
<td>UV visible albedo for direct radiation</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>016</td>
<td>ALUVD</td>
<td>UV visible albedo for diffuse radiation</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>017</td>
<td>ALNIP</td>
<td>Near IR albedo for direct radiation</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>018</td>
<td>ALNIP</td>
<td>Near IR albedo for diffuse radiation</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>027</td>
<td>CVL</td>
<td>Low vegetation cover</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>028</td>
<td>CVH</td>
<td>High vegetation cover</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>029</td>
<td>TVL</td>
<td>Type of low vegetation</td>
<td>Table index</td>
<td>an</td>
</tr>
<tr>
<td>030</td>
<td>TVH</td>
<td>Type of high vegetation</td>
<td>Table index</td>
<td>an</td>
</tr>
<tr>
<td>043</td>
<td>SLT</td>
<td>Soil type</td>
<td></td>
<td>an</td>
</tr>
<tr>
<td>066</td>
<td>LAL$_{LV}$</td>
<td>Leaf area index, low vegetation</td>
<td>m$^2$ m$^{-2}$</td>
<td>an</td>
</tr>
<tr>
<td>067</td>
<td>LAL$_{HV}$</td>
<td>Leaf area index, high vegetation</td>
<td>m$^2$ m$^{-2}$</td>
<td>an</td>
</tr>
<tr>
<td>074</td>
<td>SDFOR</td>
<td>Standard deviation of filtered sub-gridscale orography</td>
<td></td>
<td>an</td>
</tr>
<tr>
<td>160</td>
<td>SDOR</td>
<td>Standard deviation of orography</td>
<td></td>
<td>an</td>
</tr>
<tr>
<td>161</td>
<td>ISOR</td>
<td>Anisotrophy of sub-gridscale orography</td>
<td></td>
<td>an</td>
</tr>
<tr>
<td>162</td>
<td>ANOR</td>
<td>Angle of sub-gridscale orography</td>
<td>rad</td>
<td>an</td>
</tr>
<tr>
<td>163</td>
<td>SLOR</td>
<td>Slope of sub-gridscale orography</td>
<td></td>
<td>an</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land-sea mask</td>
<td>(0, 1)</td>
<td>an</td>
</tr>
<tr>
<td>173</td>
<td>SR</td>
<td>Surface roughness</td>
<td>m</td>
<td>an</td>
</tr>
<tr>
<td>174</td>
<td>ALB</td>
<td>Albedo</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>234</td>
<td>LSRH</td>
<td>Logarithm of surface roughness</td>
<td></td>
<td>an</td>
</tr>
</tbody>
</table>
11.4.4 errgrib

Table 11.5 Background errors, model levels, Gaussian grid.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Levels</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>T</td>
<td>Temperature</td>
<td>K</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>131</td>
<td>U</td>
<td>u velocity</td>
<td>m s⁻¹</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>132</td>
<td>V</td>
<td>v velocity</td>
<td>m s⁻¹</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>133</td>
<td>Q</td>
<td>Specific humidity</td>
<td>kg kg⁻¹</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>138</td>
<td>VO</td>
<td>Vorticity</td>
<td>s⁻¹</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>152</td>
<td>LNSP</td>
<td>Logarithm of surface pressure</td>
<td></td>
<td>1</td>
<td>ses</td>
</tr>
<tr>
<td>156</td>
<td>GH</td>
<td>Geopotential height</td>
<td>m</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>157</td>
<td>R</td>
<td>Relative humidity</td>
<td>%</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
<tr>
<td>194</td>
<td>BTMP</td>
<td>Brightness temperature</td>
<td>K</td>
<td>channels 1-54</td>
<td>ef</td>
</tr>
<tr>
<td>203</td>
<td>O3</td>
<td>Ozone mass mixing ratio</td>
<td>kg kg⁻¹</td>
<td>1-LEVELS</td>
<td>ses</td>
</tr>
</tbody>
</table>

11.4.5 spmoderr_bg_01

Table 11.6 Model error, spherical harmonics, model levels.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Levels</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>T</td>
<td>Temperature</td>
<td>K</td>
<td>1-LEVELS</td>
<td>me</td>
</tr>
<tr>
<td>138</td>
<td>VO</td>
<td>Vorticity</td>
<td>s⁻¹</td>
<td>1-LEVELS</td>
<td>me</td>
</tr>
<tr>
<td>152</td>
<td>LNSP</td>
<td>Logarithm of surface pressure</td>
<td></td>
<td>1</td>
<td>me</td>
</tr>
<tr>
<td>155</td>
<td>D</td>
<td>Divergence</td>
<td>s⁻¹</td>
<td>1-LEVELS</td>
<td>me</td>
</tr>
</tbody>
</table>

11.4.6 wam_specwavein

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>251</td>
<td>2DFD</td>
<td>2D wave spectra</td>
<td>m² s radian⁻¹</td>
<td>fc</td>
<td>WAVE</td>
</tr>
</tbody>
</table>

11.4.7 wam_cdwavein

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>233</td>
<td>CDWW</td>
<td>Coefficient of drag with waves</td>
<td></td>
<td>fc</td>
<td>WAVE</td>
</tr>
</tbody>
</table>

11.4.8 wam_uwavein

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>245</td>
<td>WIND</td>
<td>10 metre wind speed</td>
<td>m s⁻¹</td>
<td>fc</td>
<td>WAVE</td>
</tr>
</tbody>
</table>
11.4.9 wam_sfcwindin

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>CI</td>
<td>Sea-ice cover (0-1)</td>
<td>fc</td>
<td>DA</td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>10U</td>
<td>10 metre U wind component</td>
<td>m s(^{-1})</td>
<td>fc</td>
<td>DA</td>
</tr>
<tr>
<td>166</td>
<td>10V</td>
<td>10 metre U wind component</td>
<td>m s(^{-1})</td>
<td>fc</td>
<td>DA</td>
</tr>
</tbody>
</table>

11.5 INPUT OBSERVATION DATA

Observations are read into 4D-Var from the Observation DataBase (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 $\{\text{DATA}\}$ 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 $\{\text{WDIR}\}$ directory in task vardata at the beginning of each data assimilation cycle.

11.6.1 Invariant data files

- $\{\text{DATA}\}/an/cmod.table.ieee – scatterometer coefficients
- $\{\text{DATA}\}/an/external\_mon\_monit.b – external blacklist file
- $\{\text{DATA}\}/an/moderr.cov – model error covariances for weak-constraint 4D-Var
- $\{\text{DATA}\}/an/neuroflux\_L\{\text{LEVELS}\} – extended linearized longwave radiation
- $\{\text{DATA}\}/an/radjacobian\_L\{\text{LEVELS}\} – extended linearized longwave radiation
- $\{\text{DATA}\}/an/r\_bias\_T\_table1 – radiosonde temperature bias correction coefficients
- $\{\text{DATA}\}/an/r\_bias\_T\_table2 – radiosonde temperature bias correction coefficients
- $\{\text{DATA}\}/an/r\_bias\_T\_table3 – radiosonde temperature bias correction coefficients
- $\{\text{DATA}\}/an/r\_sref\_fmt – radiosonde height observation error correlation coefficients
- $\{\text{DATA}\}/an/ship\_anemometer\_heights – ship anemometer heights
- $\{\text{DATA}\}/an/stabal96.bal – background error balance parameters
- $\{\text{DATA}\}/an/stabal96.cv – background error correlations
- $\{\text{DATA}\}/an/stdev\_of\_climate/m\$\{\text{MM}\} – climatological standard deviations for use in error growth model
- $\{\text{DATA}\}/an/wavelet\_T\{\text{RESOLINC}\$\{\text{LEVELS}\}\}.cv – wavelet Jb background error covariances
- $\{\text{DATA}\}/climate/$\{\text{RESOL}\}$/\{\text{GTYPE}\}/O3CHEM\$\{\text{MM}\} – monthly ozone chemistry climate files
- $\{\text{DATA}\}/ifs/namelist\_L\{\text{IFS\_CYCLE}\} – an empty copy of all the IFS namelists
- $\{\text{DATA}\}/ifs/vtable\$\{\text{GTYPE}\}\$\{\text{RESOL}\} – namelist NAMRGRI, defining the number of points on each row of the Gaussian grid
- $\{\text{DATA}\}/ifs/vtable\_L\{\text{LEVELS}\} – namelist NAMV1, defining the hybrid vertical coordinate level coefficients
- $\{\text{DATA}\}/sat/mwave\_error\_L\{\text{platform}\}\$\{\text{instrument}\}.dat – microwave error files
- $\{\text{DATA}\}/sat/AIRS\_CLDDET\_NL – list of channels to be used for AIRS with cloud detection
- $\{\text{DATA}\}/sat/amv\_bias\_info – atmospheric motion vector information
- $\{\text{DATA}\}/sat/bcor\_reo3 – ozone bias correction file
- $\{\text{DATA}\}/sat/chanspec\_noaa – noaa channel specification file
Part II: Data Assimilation

11.6.2 Date-dependent data files

- ${\text{DATA}}/sat/cstlim_noaa – noaa cost limit file
- ${\text{DATA}}/sat/fibiaso_{[\text{ssmi}|\text{tmi}]} – \text{SSMI/TMI 1D-Var bias file}
- ${\text{DATA}}/sat/filcfix_{[\text{ssmi}|\text{tmi}]} – \text{SSMI/TMI 1D-Var coefficient file}
- ${\text{DATA}}/sat/filcfixv – \text{SSMI/TMI 1D-Var coefficient file}
- ${\text{DATA}}/sat/iasichannels – list of channels to be used for IASI
- ${\text{DATA}}/sat/IASI\text{LCLDDET.NL} – list of channels to be used for IASI with cloud detection
- ${\text{DATA}}/sat/mask.asc – radiosonde mask
- ${\text{DATA}}/sat/mietable_{\text{dmsp|ssmi}} – \text{optical properties of hydrometeors used in the scattering calculations for 1D-Var rain}
- ${\text{DATA}}/sat/rmtberr_{[\text{noaa|airs|iasi}]} – \text{measurement error files}
- ${\text{DATA}}/sat/rttov/rtcoef_{\{\text{platform}\}_{\{\text{instrument}\}.dat} – \text{RTTOV radiative transfer coefficient files, for all current and historic satellite platforms and instruments}
- ${\text{DATA}}/sat/scanbias_{\text{ssmi}} – \text{SSMI scan bias coefficients}
- ${\text{DATA}}/sat/scanbias_{\text{tmi}} – \text{SSMI scan bias coefficients}
- ${\text{DATA}}/sat/scanbias_{\text{tmi}} – \text{SSMI scan bias coefficients}
- ${\text{DATA}}/sat/scanbias_{\text{tmi}} – \text{SSMI scan bias coefficients}
- ${\text{DATA}}/sat/scanbias_{\text{tmi}} – \text{SSMI scan bias coefficients}
- ${\text{DATA}}/sat/ssmi\_tovs1c\_buf – \text{ATOVS BUFR template for conversion of SSMI data to 1c-radiances}
- ${\text{DATA}}/sat/thin\_reo3 – \text{ozone thinning file}
- ${\text{DATA}}/scat/mle\_norm.dat – \text{QuikSCAT look-up tables}
- ${\text{DATA}}/scat/nscat2\_noise – \text{QuikSCAT noise look-up tables}
- ${\text{DATA}}/scat/nscat2\_table – \text{QuikSCAT GMF look-up tables}
- ${\text{DATA}}/scat/qscat1\_table – \text{QuikSCAT GMF look-up tables}

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 yyyyymmdd, 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:

\[
\text{date} = \text{yyyyymmdd}, \text{hour} = 03, \text{step} = 9, \text{type} = 4v
\]

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 yyyyymmdd, 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:

\[
\text{date} = \text{yyyyymmdd}, \text{hour} = 12, \text{step} = 0, \text{type} = an
\]
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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>130</td>
<td>T</td>
<td>Temperature</td>
<td>K</td>
<td>1-LEVELS</td>
</tr>
<tr>
<td>135</td>
<td>W</td>
<td>Vertical velocity</td>
<td>m s⁻¹</td>
<td>1-LEVELS</td>
</tr>
<tr>
<td>138</td>
<td>VO</td>
<td>Vorticity</td>
<td>s⁻¹</td>
<td>1-LEVELS</td>
</tr>
<tr>
<td>152</td>
<td>LNSP</td>
<td>Logarithm of surface pressure</td>
<td>1</td>
<td>1-LEVELS</td>
</tr>
<tr>
<td>155</td>
<td>D</td>
<td>Divergence</td>
<td>s⁻¹</td>
<td>1-LEVELS</td>
</tr>
</tbody>
</table>
### 11.7.2 Output type ‘an’ model level Gaussian grid-point fields

Table 11.12 Output type ‘an’ model level Gaussian grid-point fields.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>CRWC</td>
<td>Cloud rain water content</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
<tr>
<td>76</td>
<td>CSWC</td>
<td>Cloud snow water content</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
<tr>
<td>133</td>
<td>Q</td>
<td>Specific humidity</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
<tr>
<td>203</td>
<td>O3</td>
<td>Ozone mass mixing ratio</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
<tr>
<td>246</td>
<td>CLWC</td>
<td>Cloud liquid water content</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
<tr>
<td>247</td>
<td>CIWC</td>
<td>Cloud ice water content</td>
<td>kg kg$^{-1}$</td>
<td>1-$LEVELS</td>
</tr>
</tbody>
</table>
| 248  | CC     | Cloud cover                     | (0-1)     | 1-$LEVELS 
### Output type ‘an’ surface Gaussian grid-point fields

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>015</td>
<td>ALUVP</td>
<td>UV visible albedo for direct radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>016</td>
<td>ALUVD</td>
<td>UV visible albedo for diffuse radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>017</td>
<td>ALNIP</td>
<td>Near IR albedo for direct radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>018</td>
<td>ALNIP</td>
<td>Near IR albedo for diffuse radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>027</td>
<td>CVL</td>
<td>Low vegetation cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>028</td>
<td>CVH</td>
<td>High vegetation cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>029</td>
<td>TVL</td>
<td>Type of low vegetation</td>
<td>Table index</td>
</tr>
<tr>
<td>030</td>
<td>TVH</td>
<td>Type of high vegetation</td>
<td>Table index</td>
</tr>
<tr>
<td>032</td>
<td>ASN</td>
<td>Snow albedo</td>
<td>(0-1)</td>
</tr>
<tr>
<td>035</td>
<td>ISTL1</td>
<td>Ice surface temperature, layer 1</td>
<td>K</td>
</tr>
<tr>
<td>036</td>
<td>ISTL2</td>
<td>Ice surface temperature, layer 2</td>
<td>K</td>
</tr>
<tr>
<td>037</td>
<td>ISTL3</td>
<td>Ice surface temperature, layer 3</td>
<td>K</td>
</tr>
<tr>
<td>038</td>
<td>ISTL4</td>
<td>Ice surface temperature, layer 4</td>
<td>K</td>
</tr>
<tr>
<td>042</td>
<td>SWVL4</td>
<td>Volumetric soil water, layer 4</td>
<td>m³ m⁻³</td>
</tr>
<tr>
<td>043</td>
<td>SLT</td>
<td>Soil type</td>
<td></td>
</tr>
<tr>
<td>066</td>
<td>LAI-LV</td>
<td>Leaf area index, low vegetation</td>
<td>m² m⁻²</td>
</tr>
<tr>
<td>067</td>
<td>LAI-HV</td>
<td>Leaf area index, high vegetation</td>
<td>m² m⁻²</td>
</tr>
<tr>
<td>074</td>
<td>SDFOR</td>
<td>Standard deviation of filtered subgrid orography</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>BV</td>
<td>Budget values</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>Z</td>
<td>O rography (geopotential)</td>
<td>m² s⁻²</td>
</tr>
<tr>
<td>134</td>
<td>SP</td>
<td>Surface pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>136</td>
<td>TCW</td>
<td>Total column water</td>
<td>kg m⁻²</td>
</tr>
<tr>
<td>137</td>
<td>TCWV</td>
<td>Total column water vapor</td>
<td>kg m⁻²</td>
</tr>
<tr>
<td>148</td>
<td>CHNK</td>
<td>Charnock parameter</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>MSL</td>
<td>Mean sea level pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>160</td>
<td>SDOR</td>
<td>Standard deviation of orography</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>ISOR</td>
<td>Anisotropy of sub-gridscale orography</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>ANOR</td>
<td>Angle of sub-gridscale orography</td>
<td>rad</td>
</tr>
<tr>
<td>163</td>
<td>SLOR</td>
<td>Slope of sub-gridscale orography</td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>TCC</td>
<td>Total cloud cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>165</td>
<td>10U</td>
<td>10 metre U wind component</td>
<td>m s⁻¹</td>
</tr>
<tr>
<td>166</td>
<td>10V</td>
<td>10 metre V wind component</td>
<td>m s⁻¹</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land-sea mask</td>
<td>(0,1)</td>
</tr>
<tr>
<td>173</td>
<td>SR</td>
<td>Surface roughness</td>
<td>m</td>
</tr>
<tr>
<td>174</td>
<td>ALB</td>
<td>Albedo</td>
<td>(0-1)</td>
</tr>
<tr>
<td>186</td>
<td>LCC</td>
<td>Low cloud cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>187</td>
<td>MCC</td>
<td>Medium cloud cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>188</td>
<td>HCC</td>
<td>High cloud cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>198</td>
<td>SRC</td>
<td>Skin reservoir content</td>
<td>m of water</td>
</tr>
<tr>
<td>206</td>
<td>TCO3</td>
<td>Total column ozone content</td>
<td>kg m⁻²</td>
</tr>
<tr>
<td>234</td>
<td>LSRH</td>
<td>Logarithm of surface roughness</td>
<td></td>
</tr>
<tr>
<td>235</td>
<td>SKT</td>
<td>Skin temperature</td>
<td>K</td>
</tr>
<tr>
<td>236</td>
<td>STL4</td>
<td>Soil temperature level 4</td>
<td>K</td>
</tr>
<tr>
<td>228021</td>
<td>FDIR</td>
<td>Total-sky direct solar radiation at surface</td>
<td>J m⁻²</td>
</tr>
<tr>
<td>228022</td>
<td>CDIR</td>
<td>Clear-sky direct solar radiation at surface</td>
<td>J m⁻²</td>
</tr>
<tr>
<td>228246</td>
<td>100U</td>
<td>Wind u component at 100 m</td>
<td>m s⁻¹</td>
</tr>
<tr>
<td>228247</td>
<td>100V</td>
<td>Wind v component at 100 m</td>
<td>m s⁻¹</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>251</td>
<td>2DFD</td>
<td>2D wave spectra</td>
<td>m$^2$ s radian$^{-1}$</td>
</tr>
<tr>
<td>233</td>
<td>CDWW</td>
<td>Coefficient of drag with waves</td>
<td></td>
</tr>
<tr>
<td>245</td>
<td>WIND</td>
<td>10 metre wind speed</td>
<td>m s$^{-1}$</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>CI</td>
<td>Sea-ice cover</td>
<td>(0-1)</td>
<td>fc</td>
<td>T</td>
</tr>
<tr>
<td>034</td>
<td>SST</td>
<td>Sea surface temperature</td>
<td>K</td>
<td>fc</td>
<td>T</td>
</tr>
<tr>
<td>034</td>
<td>SST</td>
<td>Sea surface temperature</td>
<td>K</td>
<td>an</td>
<td>T-6</td>
</tr>
</tbody>
</table>
11.10.2 Input OSTIA and NCEP Fields

- $WDIR/surf/anal/ostia$ – OSTIA sea surface temperature and sea ice concentration fields, encoded as BUFR data
- $WDIR/surf/anal/cfmc$ – NCEP sea surface temperature analysis
- $WDIR/surf/anal/icenmc$ – NCEP sea ice analysis

The sea surface temperature analysis and sea ice analysis from NCEP (National Center for Environmental Prediction, Washington) are input on a $0.5 \times 0.5$ degree regular latitude/longitude grid. These fields are extracted from MARS with parameters:

\[ \text{expver} = 1, \text{origin} = \text{kwbc} \]

11.10.3 Input data files

- $DATA/sst/lsfil$ – land/sea mask for NCEP data, $0.5 \times 0.5$ degree regular latitude/longitude grid, 720 $\times$ 360 points
- $DATA/sst/sstclim$ – surface air temperature monthly climatology, reduced to mean-sea level ($0.5 \times 0.5$ degree regular latitude/longitude grid, 720 $\times$ 361 points)
- $DATA/sst/iceclim$ – 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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>031</td>
<td>CI</td>
<td>Sea-ice cover</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>034</td>
<td>SST</td>
<td>Sea surface temperature</td>
<td>K</td>
<td>an</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>129</td>
<td>Z</td>
<td>Orography</td>
<td>$m^2s^{-2}$</td>
<td>an</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land/sea mask</td>
<td>(0-1)</td>
<td>an</td>
</tr>
<tr>
<td>139</td>
<td>STL1</td>
<td>Soil temperature level 1</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
<td>2 metre temperature</td>
<td>K</td>
<td>fc</td>
</tr>
<tr>
<td>168</td>
<td>2D</td>
<td>2 metre dewpoint temperature</td>
<td>K</td>
<td>fc</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Table 11.18</th>
<th>Output GRIB field on model grid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Name</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table 11.19</th>
<th>Input GRIB fields on model grid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Name</td>
</tr>
<tr>
<td>129</td>
<td>Z</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
</tr>
<tr>
<td>139</td>
<td>STL1</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
</tr>
<tr>
<td>168</td>
<td>2D</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table 11.20</th>
<th>Output GRIB field on model grid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Name</td>
</tr>
<tr>
<td>168</td>
<td>2D</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>129</td>
<td>Z</td>
<td>Orography</td>
<td>m² s⁻²</td>
<td>an</td>
<td>T4V</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land/sea mask</td>
<td>(0-1)</td>
<td>an</td>
<td>T4V</td>
</tr>
<tr>
<td>033</td>
<td>RSN</td>
<td>Snow density</td>
<td>kg m⁻³</td>
<td>fc</td>
<td>T</td>
</tr>
<tr>
<td>141</td>
<td>SD</td>
<td>Snow depth</td>
<td>m of water equivalent</td>
<td>fc</td>
<td>T</td>
</tr>
<tr>
<td>141</td>
<td>SD</td>
<td>Snow depth</td>
<td>m of water equivalent</td>
<td>an</td>
<td>T-6</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
<td>2 metre temperature</td>
<td>K</td>
<td>an</td>
<td>T</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>033</td>
<td>RSN</td>
<td>Snow density</td>
<td>kg m⁻³</td>
<td>an</td>
</tr>
<tr>
<td>141</td>
<td>SD</td>
<td>Snow depth</td>
<td>m of water equivalent</td>
<td>an</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
<th>Type</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>039</td>
<td>SWVL1</td>
<td>Volumetric soil water, layer 1</td>
<td>m³ m⁻³</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>040</td>
<td>SWVL2</td>
<td>Volumetric soil water, layer 2</td>
<td>m³ m⁻³</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>041</td>
<td>SWVL3</td>
<td>Volumetric soil water, layer 3</td>
<td>m³ m⁻³</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>133</td>
<td>Q</td>
<td>Specific humidity on lowest model level</td>
<td>kg kg⁻¹</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>139</td>
<td>STL1</td>
<td>Soil temperature, level 1</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>140</td>
<td>LSP</td>
<td>Large scale precipitation</td>
<td>m</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>143</td>
<td>CP</td>
<td>Convective precipitation</td>
<td>m</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
<td>2 metre temperature</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>168</td>
<td>2D</td>
<td>2 metre dewpoint temperature</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>170</td>
<td>STL2</td>
<td>Soil temperature, level 2</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>176</td>
<td>SSR</td>
<td>Surface solar radiation</td>
<td>W m⁻² s⁻¹</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>183</td>
<td>STL3</td>
<td>Soil temperature level 3</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>238</td>
<td>TSN</td>
<td>Temperature of snow layer</td>
<td>K</td>
<td>fc</td>
<td>Forecast</td>
</tr>
<tr>
<td>027</td>
<td>CVL</td>
<td>Low vegetation cover</td>
<td>(0-1)</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>028</td>
<td>CVH</td>
<td>High vegetation cover</td>
<td>(0-1)</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>029</td>
<td>TVL</td>
<td>Type of low vegetation</td>
<td>Table index</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>030</td>
<td>TVH</td>
<td>Type of high vegetation</td>
<td>Table index</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>129</td>
<td>Z</td>
<td>Orography</td>
<td>m² s⁻²</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>133</td>
<td>Q</td>
<td>Specific humidity on lowest model level</td>
<td>kg kg⁻¹</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>141</td>
<td>SD</td>
<td>Snow depth</td>
<td>m of water equivalent</td>
<td>an</td>
<td>Snow analysis</td>
</tr>
<tr>
<td>165</td>
<td>10U</td>
<td>10 metre U wind component</td>
<td>m s⁻¹</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>166</td>
<td>10V</td>
<td>10 metre V wind component</td>
<td>m s⁻¹</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>167</td>
<td>2T</td>
<td>2 metre temperature</td>
<td>K</td>
<td>an</td>
<td>2 metre temp. anal.</td>
</tr>
<tr>
<td>168</td>
<td>2D</td>
<td>2 metre dewpoint temperature</td>
<td>K</td>
<td>an</td>
<td>2 metre rel. hum. anal.</td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land-sea mask</td>
<td>(0, 1)</td>
<td>an</td>
<td>4D-Var</td>
</tr>
<tr>
<td>174</td>
<td>AL</td>
<td>Albedo</td>
<td>(0-1)</td>
<td>an</td>
<td>4D-Var</td>
</tr>
</tbody>
</table>

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

\[ SST = \text{analysed sea surface temperature} \]
\[ CI = \text{analysed sea ice field, which varies between 0 (open water) and 1 (full ice cover)} \]
\[ ISTL1 = \text{background soil temperature level 1} \]
Table 11.24 Output GRIB fields on model grid.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>039</td>
<td>SWVL1</td>
<td>Volumetric soil water, layer 1</td>
<td>m³ m⁻³</td>
</tr>
<tr>
<td>040</td>
<td>SWVL2</td>
<td>Volumetric soil water, layer 2</td>
<td>m³ m⁻³</td>
</tr>
<tr>
<td>041</td>
<td>SWVL3</td>
<td>Volumetric soil water, layer 3</td>
<td>m³ m⁻³</td>
</tr>
<tr>
<td>042</td>
<td>SWVL4</td>
<td>Volumetric soil water, layer 4</td>
<td>m³ m⁻³</td>
</tr>
<tr>
<td>139</td>
<td>STL1</td>
<td>Soil temperature, level 1</td>
<td>K</td>
</tr>
<tr>
<td>170</td>
<td>STL2</td>
<td>Soil temperature, level 2</td>
<td>K</td>
</tr>
<tr>
<td>183</td>
<td>STL3</td>
<td>Soil temperature, level 3</td>
<td>K</td>
</tr>
<tr>
<td>238</td>
<td>TSN</td>
<td>Temperature of snow layer</td>
<td>K</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>015</td>
<td>ALUVP</td>
<td>UV visible albedo for direct radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>016</td>
<td>ALUVD</td>
<td>UV visible albedo for diffuse radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>017</td>
<td>ALNIP</td>
<td>Near IR albedo for direct radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>018</td>
<td>ALNIP</td>
<td>Near IR albedo for diffuse radiation</td>
<td>(0-1)</td>
</tr>
<tr>
<td>027</td>
<td>CVL</td>
<td>Low vegetation cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>028</td>
<td>CVH</td>
<td>High vegetation cover</td>
<td>(0-1)</td>
</tr>
<tr>
<td>029</td>
<td>TVL</td>
<td>Type of low vegetation</td>
<td>Table index</td>
</tr>
<tr>
<td>030</td>
<td>TVH</td>
<td>Type of high vegetation</td>
<td>Table index</td>
</tr>
<tr>
<td>043</td>
<td>SLT</td>
<td>Soil type</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>Z</td>
<td>Orography</td>
<td>m² s⁻²</td>
</tr>
<tr>
<td>160</td>
<td>SDOR</td>
<td>Standard deviation of orography</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>ISOR</td>
<td>Anisotrophy of sub-gridscale orography</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>ANOR</td>
<td>Angle of sub-gridscale orography</td>
<td>rad</td>
</tr>
<tr>
<td>163</td>
<td>SLOR</td>
<td>Slope of sub-gridscale orography</td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>LSM</td>
<td>Land-Sea mask</td>
<td>(0,1)</td>
</tr>
<tr>
<td>173</td>
<td>SR</td>
<td>Surface roughness</td>
<td>m</td>
</tr>
<tr>
<td>174</td>
<td>ALB</td>
<td>Albedo</td>
<td>(0-1)</td>
</tr>
<tr>
<td>234</td>
<td>LSRH</td>
<td>Logarithm of surface roughness</td>
<td></td>
</tr>
</tbody>
</table>
References


Part II: Data Assimilation


