Data assimilation concepts and methods
March 1999

By **F. Bouttier and P. Courtier**

Abstract

These training course lecture notes are an advanced and comprehensive presentation of most data assimilation methods that are considered useful in applied meteorology and oceanography today. Some are considered old-fashioned but they are still valuable for low cost applications. Others have never been implemented yet in realistic applications, but they are regarded as the future of data assimilation. A mathematical approach has been chosen, which allows a compact and rigorous presentation of the algorithms, though only some basic mathematical competence is required from the reader.

This document has been put together with the help of previous lecture notes, which are now superseded:

- Variational analysis: use of observations, example of clear radiances, Jean Pailleux, 1989.
- Inversion methods for satellite sounding data, J. Eyre, 1991. (part 2 only)
- Methods of data assimilation: optimum interpolation, P. Undén, 1993. (except section 5)
- Data assimilation methods: introduction to statistical estimation, J. Eyre and P. Courtier, 1994.
- Variational methods, P. Courtier, 1995. (except sections 3.2-3.6, 4.5, 4.6)
- Kalman filtering, F, Bouttier, 1997. (except the predictability parts)

Traditionally the lecture notes have been referring a lot to the assimilation and forecast system at ECMWF, rather than to more general algorithms. Sometimes ideas that had not even been tested found their way into the training course lecture notes. New notes had to be written every couple of years, with inconsistent notation.

In this new presentation it has been decided to stick to a description of the main assimilation methods used worldwide, without any reference to ECMWF specific features, and clear comparisons between the different algorithms. This should make it easier to adapt the methods to problems outside the global weather forecasting framework of ECMWF, e.g. ocean data assimilation, land surface analysis or inversion of remote-sensing data. It is hoped that the reader will manage to see the physical nature of the algorithms beyond the mathematical equations.

A first edition of these lecture notes was released in March 1998. In this second edition, some figures were added, and a few errors were corrected.

Thanks are due to J. Pailleux, J. Eyre, P. Undén and A. Hollingsworth for their contribution to the previous lecture notes, to A. Lorenc, R. Daley, M. Ghil and O. Talagrand for teaching the various forms of the statistical interpolation technique to the meteorological world, to D. Richardson for proof-reading the document, and to the attendees of training course who kindly provided constructive comments.

Table of contents

1. BASIC CONCEPTS OF DATA ASSIMILATION

Analysis. An analysis is the production of an accurate image of the true state of the atmosphere at a given time, represented in a model as a collection of numbers. An analysis can be useful in itself as a comprehensive and self-consistent diagnostic of the atmosphere. It can also be used as input data to another operation, notably as the initial state for a numerical weather forecast, or as a data retrieval to be used as a pseudo-observation. It can provide a reference against which to check the quality of observations.

The basic objective information that can be used to produce the analysis is a collection of observed values provided by observations of the true state. If the model state is overdetermined by the observations, then the analysis reduces to an interpolation problem. In most cases the analysis problem is under-determined[1] because data is sparse and only indirectly related to the model variables. In order to make it a well-posed problem it is necessary to rely on some *background* information in the form of an a priori estimate of the model state. Physical constraints on the analysis problem can also help. The background information can be a climatology or a trivial state; it can also be generated from the output of a previous analysis, using some assumptions of consistency in time of the model state, like stationarity (hypothesis of persistence) or the evolution predicted by a forecast model. In a well-behaved system, one expects that this allows the information to be accumulated in time into the model state, and to propagate to all variables of the model. This is the concept of data assimilation.

1. although it can be overdetermined locally in data-dense areas

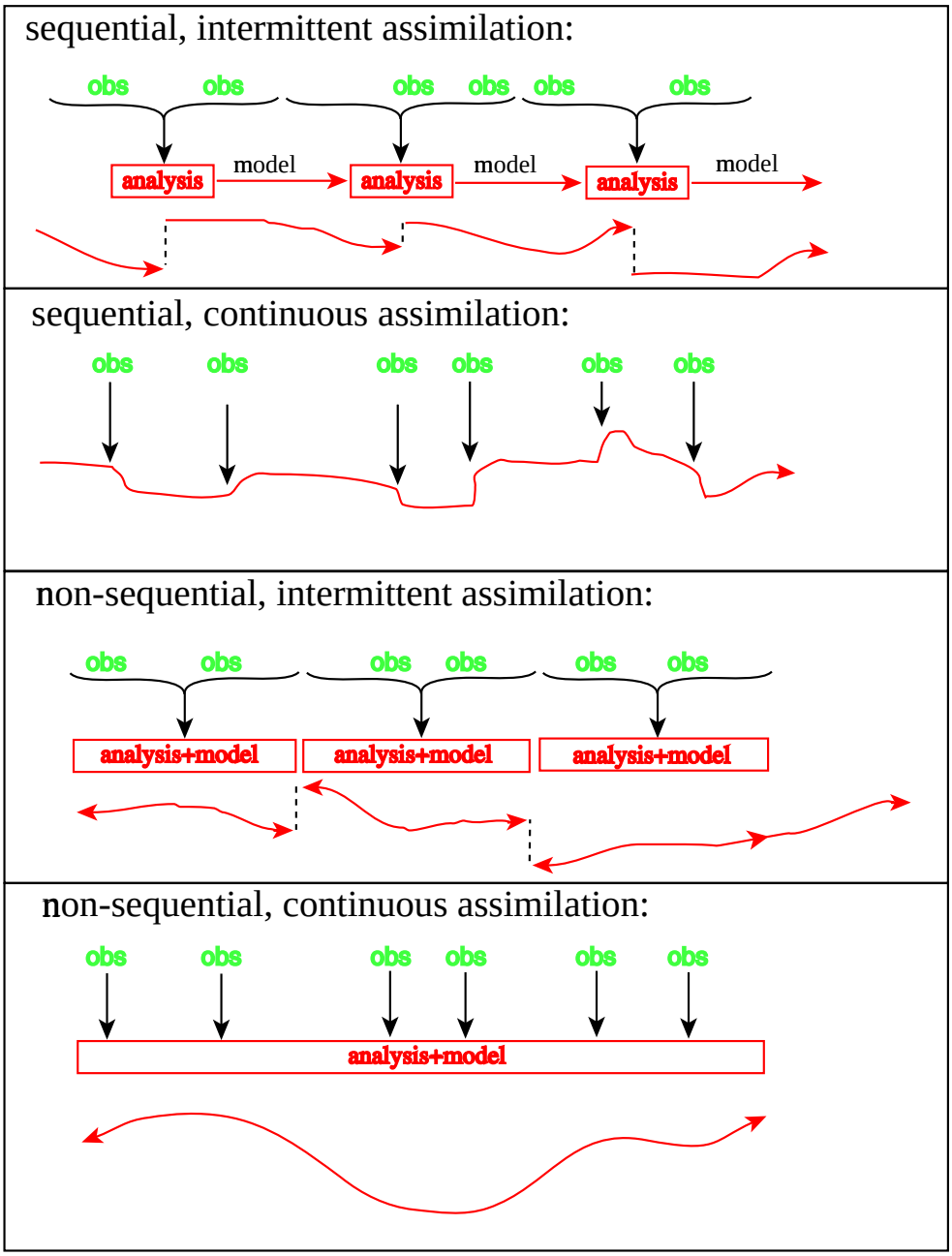

Figure 1. Representation of four basic strategies for data assimilation, as a function of time. The way the time distribution of observations ("obs") is processed to produce a time sequence of assimilated states (the lower curve in each panel) can be sequential and/or continuous.

Assimilation. Data assimilation is an analysis technique in which the observed information is accumulated into the model state by taking advantage of consistency constraints with laws of time evolution and physical properties.

There are two basic approaches to data assimilation: *sequential* assimilation, that only considers observation made in the past until the time of analysis, which is the case of real-time assimilation systems, and *non-sequential*, or *retrospective* assimilation, where observation from the future can be used, for instance in a reanalysis exercise. An-

other distinction can made between methods that are *intermittent* or *continuous* in time. In an intermittent method, observations can be processed in small batches, which is usually technically convenient. In a continuous method, observation batches over longer periods are considered, and the correction to the analysed state is smooth in time, which is physically more realistic. The four basic types of assimilation are depicted schematically in Fig. 1 . Compromises between these approaches are possible.

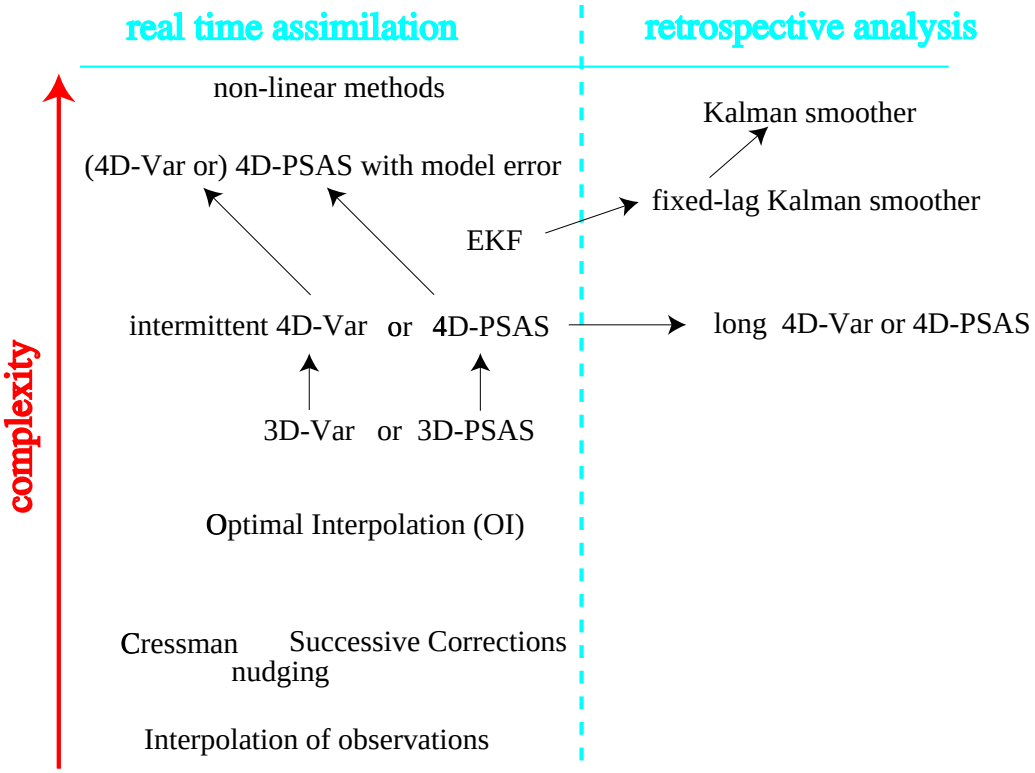

Figure 2. A summarized history of the main data assimilation algorithms used in meteorology and oceanography, roughly classified according to their complexity (and cost) of implementation, and their applicability to real-time problems. Currently, the most commonly used for operational applications are OI, 3D-Var and 4D-Var.

Many assimilation techniques have been developed for meteorology and oceanography (Fig. 2). They differ in their numerical cost, their optimality, and in their suitability for real-time data assimilation. Most of them are explained in this volume.

ref: Daley 1991; Lorenc 1986; Ghil 1989

1.1 On the choice of model

The concepts developed here are illustrated by examples in the ECMWF global meteorological model, but they can be (and they have been) applied equally well to limited area models, mesoscale models, ocean circulation models, wave models, two-dimensional models of sea surface temperature or land surface properties, or one-dimensional vertical column models of the atmosphere for satellite data retrieval, for example. This presentation could be made in the general framework of an infinite-dimensional model (i.e. without discretization) with a continuous time dimension. This would involve some sophisticated mathematical tools. For the sake of simplicity, only the discrete, finite-dimensional problem will be addressed here.

In meteorology there are often several equivalent ways of representing the model state. The fields themselves can be represented as grid-point values (i.e. averages of the fields inside grid boxes), spectral components, EOF values, finite-element decomposition, for instance, which can be projections on different basis vectors of the same state. The wind can be represented as components (u, v), vorticity and divergence (ζ, η), or streamfunction and velocity potential (ψ, χ), with a suitable definition of the integration constants. The humidity can be represented as specific or relative humidity or dew-point temperature, as long as temperature is known. In the vertical, under the assumption of hydrostatic balance, thicknesses or geopotential heights can be regarded as equivalent to the knowledge of temperature and surface pressure. All these transforms do not change the analysis problem, only its representation[2]. This may sound trivial, but it is important to realize that the analysis can be carried out in a representation that is not the same as the forecast model, as long as the transforms are invertible. The practical problems of finding the analysis, e.g. the modelling of error statistics, can be greatly simplified if the right representation is chosen.

Since the model has a lower resolution than reality, even the best possible analysis will never be completely realistic. In the presentation of analysis algorithms we will sometimes refer to the *true state* of the model. This is a phrase to refer to the best possible state represented by the model, which is what we are trying to approximate. Hence it is clear that, even if the observations do not have any instrumental error, and the analysis is equal to the true state, there will be some unavoidable discrepancies between the observed values and their equivalents in the analysis, because of *representativeness errors*. Although we will often treat these errors as a part of the *observation errors* in the mathematical equations below, one should keep in mind that they depend on the model discretization, not on instrumental problems.

1.2 Cressman analysis and related methods

One may like to design the analysis procedure as an algorithm in which the model state is set equal to the observed values in the vicinity of available observations, and to an arbitrary state (say, climatology or a previous forecast) otherwise. This formed the basis of the old *Cressman analysis scheme* (Fig. 3) which is still widely used for simple assimilation systems.

The model state is assumed to be univariate and represented as grid-point values. If we denote by $\mathbf{x}_\mathrm{b}$ a previous estimate of the model state (*background*) provided by climatology, persistence or a previous forecast, and by $\mathbf{y}(i)$, a set of $i = 1 \ldots\ldots n$ observations of the same parameter, a simple kind of Cressman analysis is provided by the model state $\mathbf{x}_\mathrm{a}$ defined at each grid point j according to the following *update* equation:

$$\mathbf{x}_\mathrm{a}(j) = \mathbf{x}_\mathrm{b}(j) + \frac{\sum_{i=1}^{n} w(i,j)\{\mathbf{y}(i) - \mathbf{x}_\mathrm{b}(i)\}}{\sum_{i=1}^{n} w(i,j)}$$

$$w(i,j) = \max\left(0, \frac{R^2 - d_{i,j}^2}{R^2 + d_{i,j}^2}\right)$$

where $d_{i,j}$ is a measure of the distance between points i and j. $\mathbf{x}_\mathrm{b}(i)$ is the background state interpolated to point i. The weight function $w(i,j)$ equals one if the grid point j is collocated with observation i. It is a

2. At ECMWF, the analysis problem is currently formulated in terms of the spectral components of vorticity, divergence, temperature, grid-point values of specific humidity, on surfaces defined by the hybrid coordinate, and logarithm of surface pressure, just like in the forecast model. In winter 1998 the model state dimension was about 6.10^6.

decreasing function of distance which is zero if $d_{i,j} > R$, where R is a user-defined constant (the "influence radius") beyond which the observations have no weight.

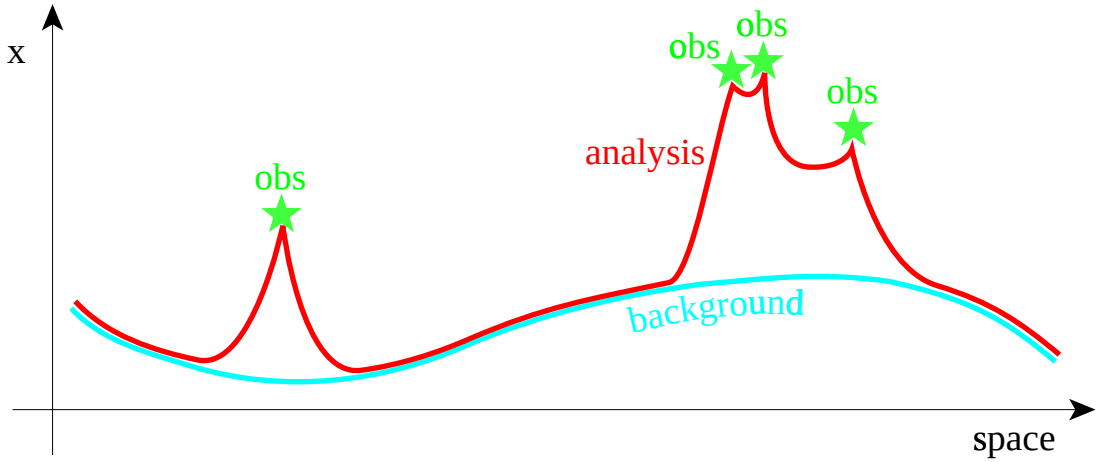

Figure 3. An example of Cressman analysis of a one-dimensional field. The background field $\mathbf{x}_b$ is represented as the blue function, and the observations in green. The analysis (black curve) is produced by interpolating between the background (grey curve) and the observed value, in the vicinity of each observation; the closer the observation, the larger its weight.

There are many variants of the Cressman method. One can redefine the weight function, e.g. as $\exp(-d_{i,j}^2 / 2R^2)$. A more general algorithm is the *successive correction method (SCM)*[3]. One of its features is that the weights can be less than one for $i = j$, which means that a weighted average between the background and the observation is performed. Another one is that the updates can be performed several times, either as several iterations at a single time in order to enhance the smoothness of corrections, or as several corrections distributed in time. With enough sophistication the successive correction method can be as good as any other assimilation method, however there is no direct method for specifying the optimal weights.

ref: Daley 1991

1.3 The need for a statistical approach

The Cressman method is not satisfactory in practice for the following reasons:

- if we have a preliminary estimate of the analysis with a good quality, we do not want to replace it by values provided from poor quality observations.
- when going away from an observation, it is not clear how to relax the analysis toward the arbitrary state, i.e. how to decide on the shape of the function w.
- an analysis should respect some basic known properties of the true system, like smoothness of the fields, or relationship between the variables (e.g. hydrostatic balance, or saturation constraints). This is not guaranteed by the Cressman method: random observation errors could generate unphysical features in the analysis.

Because of its simplicity, the Cressman method can be a useful starting tool. But it is impossible to get rid of the

3. In the recent literature this name is often replaced by *observation nudging* which is more or less the same thing. The *model nudging* is a model forcing technique in which the model state is relaxed toward another predefined state.

above problems and to produce a good-quality analysis without a better method. The ingredients of a good analysis are actually well known by anyone who has experience with manual analysis:

1) one should start from a good-quality first guess, i.e. a previous analysis or forecast that gives an overview of the situation,

2) if observations are dense, then one assumes that the truth probably lies near their average. One must make a compromise between the first guess and the observed values. The analysis should be closest to the data we trust most, whereas suspicious data will be given little weight.

3) the analysis should be smooth, because we know that the true field is. When going away from an observation, the analysis will relax smoothly to the first guess on scales known to be typical of the usual physical phenomena.

4) the analysis should also try to respect the known physical features of the system. Of course, it is possible in exceptional cases that unusual scales and imbalances happen, and a good analyst must be able to recognize this, because exceptional cases are usually important too.

Loosely speaking, the data that can go into the analysis system comprises the observations, the first guess and the known physical properties of the system. One sees that the most important feature to represent in the analysis system is the fact that all pieces of data are important sources of information, but at the same time we do not trust any of them completely, so we must make compromises when necessary. There are errors in the model and in the observations, so we can never be sure which one to trust. However we can look for a strategy that minimizes on average the difference between the analysis and the truth.

To design an algorithm that does this automatically, it is necessary to represent mathematically the uncertainty of the data. This uncertainty can be measured by calibrating (or by assuming) their error statistics, and modelled using probabilistic concepts. Then the analysis algorithm can be designed on a formal requirement that in the average the analysis errors must be minimal in a sense that is meaningful to the user. This will allow us to write the analysis as an optimization problem.

ref: Lorenc 1986

2. THE STATE VECTOR, CONTROL SPACE AND OBSERVATIONS

2.1 State vector

The first step in the mathematical formalisation of the analysis problem is the definition of the work space. As in a forecast model, the collection of numbers needed to represent the atmospheric state of the model is collected as a column matrix called the *state vector* $\mathbf{x}$. How the vector components relate to the real state depend on the choice of discretization, which is mathematically equivalent to a choice of basis.

As explained earlier, one must distinguish between reality itself (which is more complex than what can be represented as a state vector) and the best possible representation of reality as a state vector, which we shall denote $\mathbf{x}_t$, the *true* state at the time of the analysis. Another important value of the state vector is $\mathbf{x}_b$, the a priori or *background* estimate of the true state before the analysis is carried out, valid at the same time[4]. Finally, the *analysis* is denoted $\mathbf{x}_a$, which is what we are looking for.

4. It is sometimes called the *first guess*, but the recommended word is *background*, for reasons explained later.

2.2 Control variable

In practice it is often convenient not to solve the analysis problem for all components of the model state. Perhaps we do not know how to perform a consistent analysis of all components[5], or we have to reduce the resolution or domain of analysis because of insufficient computer power. This is difficult to avoid as the resolution and sophistication of forecast models tend to be as high as the computing power allows, i.e. too high for the analysis which is more expensive because the observations have to be processed on top of the management of the model state itself. In these cases the work space of the analysis is not the model space, but the space allowed for the corrections to the background, called *control variable space*. Then the analysis problem is to find a correction $\delta\mathbf{x}$ (or *analysis increment*) such that

$$\mathbf{x}_a = \mathbf{x}_b + \delta\mathbf{x}$$

is as close as possible to $\mathbf{x}_t$. Formally the analysis problem can be presented exactly like before by a simple translation: instead of looking for $\mathbf{x}_a$, we look for $(\mathbf{x}_a - \mathbf{x}_b)$ in a suitable subspace[6].

2.3 Observations

For a given analysis we use a number of observed values. They are gathered into an *observation vector* $\mathbf{y}$. To use them in the analysis procedure it is necessary to be able to compare them with the state vector. It would be nice if each degree of freedom were observed directly, so $\mathbf{y}$ could be regarded as a particular value of the state vector. In practice there are fewer observations than variables in the model and they are irregularly disposed, so that the only correct way to compare observations with the state vector is through the use of a function from model state space to observation space called an *observation operator*[7] that we will denote by H. This operator generates the values $H(\mathbf{x})$ that the observations would take if both they and the state vector were perfect, in the absence of any modelling error[8]. In practice H is a collection of interpolation operators from the model discretization to the observation points, and conversions from model variables to the observed parameters. For each scalar observation there is a corresponding line of H. The number of observations, i.e. the dimension of vector $\mathbf{y}$ and the number of lines in H, is varying if the observing network is not exactly periodic in time. There are usually many fewer observations than variables in the model.

2.4 Departures

The key to data analysis is the use of the discrepancies between observations and state vector. According to the previous paragraph, this is given by the vector of departures at the observation points:

$$\mathbf{y} - \mathbf{H}(\mathbf{x})$$

When calculated with the background $\mathbf{x}_b$ it is called *innovations*, and with the analysis $\mathbf{x}_a$, *analysis residuals*. Their study provides important information about the quality of the assimilation procedure.

5. This is often the case with surface or cloud-related variables, or the boundary conditions in limited-area models.
6. Mathematically speaking, we constrain $\mathbf{x}_a$ to belong to the affine manifold spanned by $\mathbf{x}_b$ plus the control variable vector subspace.
7. also called *forward operator*
8. the values $H(\mathbf{x})$ are also called *model equivalents of the observations*.

3. THE MODELLING OF ERRORS

To represent the fact that there is some uncertainty in the background, the observations and in the analysis we will assume some model of the errors between these vectors and their true counterparts. The correct way to do this is to assume some *probability density function*, or pdf, for each kind of error. There is a sophisticated and rigorous mathematical theory of probabilities to which the reader may refer. For the more practical minds we present a simplified (and mathematically loose) explanation of pdfs in the paragraph below, using the example of background errors.

3.1 Using pdfs to represent uncertainty

Given a background field $\mathbf{x}_b$ just before doing an analysis, there is one and only one vector of errors that separates it from the true state:

$$\boldsymbol{\varepsilon}_b = \mathbf{x}_b - \mathbf{x}_t$$

If we were able to repeat each analysis experiment a large number of times, under exactly the same conditions, but with different realizations of errors generated by unknown causes, $\boldsymbol{\varepsilon}_b$ would be different each time. We can calculate statistics such as averages, variances and histograms of frequencies of $\boldsymbol{\varepsilon}_b$. In the limit of a very large number of realizations, we expect the statistics to converge to values which depend only on the physical processes responsible for the errors, not on any particular realization of these errors. When we do another analysis under the same conditions, we do not expect to know what will be the error $\boldsymbol{\varepsilon}_b$, but at least we will know its statistics. The best information about the distribution of $\boldsymbol{\varepsilon}_b$ is given by the limit of the histogram when the classes are infinitely small, which is a scalar function of integral 1 called the *probability density function* of $\boldsymbol{\varepsilon}_b$. From this function one can derive all statistics, including the average (or expectation) $\bar{\boldsymbol{\varepsilon}}_b$ and the variances[9]. A popular model of scalar pdf is the Gaussian function, which can be generalized to a multivariate pdf.

3.2 Error variables

The errors in the background and in the observations[10] are modelled as follows:

- **background errors**: $\boldsymbol{\varepsilon}_b = \mathbf{x}_b - \mathbf{x}_t$, of average $\bar{\boldsymbol{\varepsilon}}_b$ and covariances $\mathbf{B} = \overline{(\boldsymbol{\varepsilon}_b - \bar{\boldsymbol{\varepsilon}}_b)(\boldsymbol{\varepsilon}_b - \bar{\boldsymbol{\varepsilon}}_b)^T}$. They are the estimation errors of the background state, i.e. the difference between the background state vector and its true value. They do not include discretization errors.
- **observation errors**: $\boldsymbol{\varepsilon}_o = \mathbf{y} - H(\mathbf{x}_t)$, of average $\bar{\boldsymbol{\varepsilon}}_o$ and covariances $\mathbf{R} = \overline{(\boldsymbol{\varepsilon}_o - \bar{\boldsymbol{\varepsilon}}_o)(\boldsymbol{\varepsilon}_o - \bar{\boldsymbol{\varepsilon}}_o)^T}$. They contain errors in the observation process (instrumental errors, because the reported value is not a perfect image of reality), errors in the design of the operator H, and representativeness errors i.e. discretization errors which prevent $\mathbf{x}_t$ from being a perfect image of the true state[11].
- **analysis errors**: $\boldsymbol{\varepsilon}_a = \mathbf{x}_a - \mathbf{x}_t$, of average $\bar{\boldsymbol{\varepsilon}}_a$. A measure $\overline{\|\boldsymbol{\varepsilon}_a - \bar{\boldsymbol{\varepsilon}}_a\|}$ of these errors is given by the trace of the analysis error covariance matrix $\mathbf{A}$,

9. Mathematically speaking, a pdf may not have an average or variances, but in the usual geophysical problems all pdfs do, and we will assume this throughout this presentation.

10. One could model forecast errors and balance properties in a similar way, although this is outside the scope of this discussion. See the section on the Kalman filter.

11. An example is sharp temperature inversions in the vertical. They can be fairly well observed using a radiosonde, but it is impossible to represent them precisely with the current vertical resolution of atmospheric models. On the other hand, temperature soundings obtained from satellite cannot themselves observe sharp inversions.

$$\mathrm{Tr}(\mathbf{A}) = \overline{\left\|\varepsilon_a - \bar{\varepsilon}_a\right\|^2}.$$

They are the estimation errors of the analysis state, which is what we want to minimize.

The averages of errors are called *biases* and they are the sign of a systematic problem in the assimilating system: a model drift, or a bias in the observations, or a systematic error in the way they are used.

It is important to understand the algebraic nature of the statistics. Biases are vectors of the same kind as the model state or observation vectors, so their interpretation is straightforward. Linear transforms that are applied to model state or observation vectors (such as spectral transforms) can be applied to bias vectors.

3.3 Using error covariances

Error covariances are more subtle and we will illustrate this with the background errors (all remarks apply to observation errors too). In a scalar system, the background error covariance is simply the variance, i.e. the root-mean-square (or *r.m.s.*, or *quadratic*) average of departures from the mean:

$$\mathbf{B} = \mathrm{var}(\varepsilon_b) = \overline{\mathrm{var}(\varepsilon_b - \bar{\varepsilon}_b)^2}$$

In a multidimensional system, the covariances are a square symmetric matrix. If the model state vector has dimension n, then the covariances are an $n \times n$ matrix. The diagonal of the matrix contain variances[12], for each variable of the model; the off-diagonal terms are cross-covariances between each pair of variables of the model. The matrix is positive[13]. Unless some variances are zero, which happens only in the rather special case where one believes some features are perfect in the background, the error covariance matrix is positive definite. For instance if the model state is tri-dimensional, and the background errors (minus their average) are denoted (e_1, e_2, e_3), then

$$\mathbf{B} = \begin{bmatrix} \mathrm{var}(e_1) & \mathrm{cov}(e_1, e_2) & \mathrm{cov}(e_1, e_3) \\ \mathrm{cov}(e_1, e_2) & \mathrm{var}(e_2) & \mathrm{cov}(e_2, e_3) \\ \mathrm{cov}(e_1, e_3) & \mathrm{cov}(e_2, e_3) & \mathrm{var}(e_3) \end{bmatrix}$$

The off-diagonal terms can be transformed into error correlations (if the corresponding variances are non zero):

$$\rho(e_i, e_j) = \frac{\mathrm{cov}(e_i, e_j)}{\sqrt{\mathrm{var}(e_i)\mathrm{var}(e_j)}}$$

Finally, linear transformations of the model state vector can only be applied to covariances as full matrix transforms. In particular, it is not possible to directly transform the fields of variances or standard deviations. If one defines a linear transformation by a matrix P (i.e. a matrix whose lines are the coordinates of the new basis vectors in terms of the old ones, so that the new coordinates of the transform of $\mathbf{x}$ are $P\mathbf{x}$), then the covariance matrix in terms of the new variables is $P\mathbf{B}P^{\mathrm{T}}$.

12. The square roots of variances are called *standard deviations*, or *standard errors*.

13. This does not mean that all the matrix elements are positive; the definition of a positive definite matrix is given in Appendix A. The positiveness can be proven by remarking that the eigenvalues of the matrix are the variances in the direction of the eigenvectors, and thus are positive.

3.4 Estimating statistics in practice

The error statistics (biases and covariances) are functions of the physical processes governing the meteorological situation and the observing network. They also depend on our a priori *knowledge* of the errors. Error variances in particular reflect our uncertainty in features of the background or the observations. In general, the only way to estimate statistics is to assume that they are stationary over a period of time and uniform over a domain[14] so that one can take a number of error realizations and make empirical statistics. This is in a sense a climatology of errors. Another empirical way to specify error statistics is to take them to be a fraction of the climatological statistics of the fields themselves.

When setting up an assimilation system in practice, such approximations are unavoidable because it is very difficult to gather accurate data to calibrate statistics: estimation errors cannot be observed directly. Some useful information on the average values of the statistics can be gathered from diagnostics of an existing data assimilation system using the *observational method* (see its description below) and the NMC method (use of forecast differences as surrogates to short-range forecast errors). More detailed, flow-dependent forecast error covariances can be estimated directly from a Kalman filter (described below), although this algorithm raises other problems. Finally, meteorological common sense can be used to specify error statistics, to the extent that they reflect our a priori knowledge of the physical processes responsible for the errors[15].

ref: Hollingsworth *et al.* 1986; Parrish and Derber 1992

4. STATISTICAL INTERPOLATION WITH LEAST-SQUARES ESTIMATION

In this section we present the fundamental equation for linear analysis in a general algebraic form: the *least squares estimation*, also called *Best Linear Unbiased Estimator (BLUE)*. The following sections will provide more explanations and illustrations, and we shall see how the least-squares estimation can be simplified to yield the most common algorithms used nowadays in meteorology and oceanography.

4.1 Notation and hypotheses

The dimension of the model state is n and the dimension of the observation vector is p. We will denote:

$\mathbf{x}_t$ true model state (dimension n)
$\mathbf{x}_b$ background model state (dimension n)
$\mathbf{x}_a$ analysis model state (dimension n)
$\mathbf{y}$ vector of observations (dimension p)
H observation operator (from dimension n to p)
$\mathbf{B}$ covariance matrix of the background errors $(\mathbf{x}_b - \mathbf{x}_t)$ (dimension $n \times n$)
$\mathbf{R}$ covariance matrix of observation errors $(\mathbf{y} - H[\mathbf{x}_t])$ (dimension $p \times p$)
$\mathbf{A}$ covariance matrix of the analysis errors $(\mathbf{x}_a - \mathbf{x}_t)$ (dimension $n \times n$)

The following hypotheses are assumed:

- **Linearized observation operator**: the variations of the observation operator in the vicinity of the background state are linear: for any $\mathbf{x}$ close enough to $\mathbf{x}_b$, $H(\mathbf{x}) - H(\mathbf{x}_b) = \mathbf{H}(\mathbf{x} - \mathbf{x}_b)$ where $\mathbf{H}$ is a linear operator.

14. It is called an assumption of *ergodicity.*

15. It is obvious that e.g. forecast errors in a tropical meteorological assimilation shall be increased in the vicinity of reported tropical cyclones, for instance, or that observation operators for satellite radiances have more errors in cloudy areas.

- **Non-trivial errors**: $\mathbf{B}$ and $\mathbf{R}$ are positive definite matrices.
- **Unbiased errors**: the expectation of the background and observation errors is zero i.e. $\overline{\mathbf{x}_b - \mathbf{x}_t} = \overline{\mathbf{y} - H(\mathbf{x}_t)} = 0$
- **Uncorrelated errors**: observation and background errors are mutually uncorrelated i.e. $\overline{(\mathbf{x}_b - \mathbf{x}_t)(\mathbf{y} - H[\mathbf{y}_t])^T} = 0$
- **Linear analysis**: we look for an analysis defined by corrections to the background which depend linearly on background observation departures.
- **Optimal analysis**: we look for an analysis state which is as close as possible to the true state in an r.m.s. sense (i.e. it is a minimum variance estimate).

ref: Daley 1991; Lorenc 1986; Ghil 1989

4.2 Theorem: least-squares analysis equations

(a) The *optimal least-squares estimator*, or *BLUE analysis*, is defined by the following interpolation equations:

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - H[\mathbf{x}_b]) \tag{A1}$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{A2}$$

where the linear operator $\mathbf{K}$ is called the *gain*, or *weight matrix*, of the analysis.

(a) The *analysis error covariance matrix* is, for any $\mathbf{K}$:

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B}(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \tag{A3}$$

If $\mathbf{K}$ is the optimal least-squares gain, the expression becomes

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B} \tag{A4}$$

(a) The BLUE analysis is equivalently obtained as a solution to the *variational optimization problem*:

$$\begin{aligned} \mathbf{x}_a &= \text{Arg min} J \\ J(\mathbf{x}) &= (\mathbf{x} - \mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + (\mathbf{y} - H[\mathbf{x}])^T\mathbf{R}^{-1}(\mathbf{y} - H[\mathbf{x}]) \\ &= J_b(\mathbf{x}) + J_o(\mathbf{x}) \end{aligned} \tag{A5}$$

where J is called the *cost function* of the analysis (or *misfit*, or *penalty function*), J_b is the *background term*, J_o is the *observation term*.

(a) The analysis $\mathbf{x}_a$ is *optimal*: it is closest in an r.m.s. sense to the true state $\mathbf{x}_t$.

(b) If the background and observation error pdfs are Gaussian, then $\mathbf{x}_a$ is also the *maximum likelihood estimator* of $\mathbf{x}_t$.

Proof:

With a translation of $\mathbf{x}$ by $\mathbf{x}_b$, we can assume that $H = \mathbf{H}$ so the observation operator is linear for our purposes. The equation (A1) is simply a mathematical expression of the fact that we want the analysis to depend linearly on the observation departures. The expression of $\mathbf{K}$ in (A2) is well-defined because $\mathbf{R}$ is a positive definite matrix, and $\mathbf{H}\mathbf{B}\mathbf{H}^T$ is positive.

The minimization problem (A5) is well-defined because J_o is a convex function and J_b is a strictly convex function (it is a quadratic form).
The equivalence between items (a) and (c) of the theorem stems from the requirement that the gradient of J is zero at the optimum $\mathbf{x}_a$:

$$\begin{aligned}\nabla J(\mathbf{x}_a) = 0 &= 2\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) - 2\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - H[\mathbf{x}_a]) \\ 0 &= \mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) - \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - H[\mathbf{x}_b]) - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}_a - \mathbf{x}_b) \\ (\mathbf{x}_a - \mathbf{x}_b) &= (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - H[\mathbf{x}_b])\end{aligned}$$

The identity with (A2) is straightforward to prove (all inverse matrices considered are positive definite):

$$\begin{aligned}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}) &= (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{B}\mathbf{H}^T \\ &= \mathbf{H}^T + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\end{aligned}$$

hence

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

The expressions (A3) and (A4) for $\mathbf{A}$ are obtained by rewriting the analysis equation (A1) in terms of the background, analysis and observation errors:

$$\begin{aligned}\boldsymbol{\varepsilon}_b &= \mathbf{x}_b - \mathbf{x}_t \\ \boldsymbol{\varepsilon}_a &= \mathbf{x}_a - \mathbf{x}_t \\ \boldsymbol{\varepsilon}_o &= \mathbf{y} - H[\mathbf{x}_t] \\ \boldsymbol{\varepsilon}_a - \boldsymbol{\varepsilon}_b &= \mathbf{K}(\boldsymbol{\varepsilon}_o - \mathbf{H}\boldsymbol{\varepsilon}_b) \\ \boldsymbol{\varepsilon}_a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\boldsymbol{\varepsilon}_b + \mathbf{K}\boldsymbol{\varepsilon}_o\end{aligned}$$

By developing the expression of $\boldsymbol{\varepsilon}_a\boldsymbol{\varepsilon}_a^T$ and taking its expectation, by linearity of the expectation operator one finds the general expression (A3) (remember that $\boldsymbol{\varepsilon}_b$ and $\boldsymbol{\varepsilon}_o$ being uncorrelated, their cross-covariance is zero). The simpler form (A4) is easy to derive by substituting the expression for the optimal $\mathbf{K}$ and simplifying the terms that cancel.

Finally to prove (A2) itself we take the analysis error covariance matrix given by (A3) and we minimize its trace, i.e. the total error variance: (note that $\mathbf{B}^T = \mathbf{B}$ and $\mathbf{R}^T = \mathbf{R}$)

$$\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{B}) + \mathrm{Tr}(\mathbf{K}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{K}^T) - 2\mathrm{Tr}(\mathbf{B}\mathbf{H}^T\mathbf{K}^T) + \mathrm{Tr}(\mathbf{K}\mathbf{R}\mathbf{K}^T)$$

This is a continuous differentiable scalar function of the coefficients of $\mathbf{K}$, so we can express its derivative $d_{\mathbf{K}}$ as the first-order terms in $\mathbf{K}$ of the difference $\mathrm{Tr}(\mathbf{A})(\mathbf{K} + \mathbf{L}) - \mathrm{Tr}(\mathbf{A})(\mathbf{K})$, $\mathbf{L}$ being an arbitrary test matrix:

$$\begin{aligned}d_{\mathbf{K}}[\mathrm{Tr}(\mathbf{A})]\mathbf{L} &= 2\mathrm{Tr}(\mathbf{K}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{L}^T) - 2\mathrm{Tr}(\mathbf{B}\mathbf{H}^T\mathbf{L}^T) + 2\mathrm{Tr}(\mathbf{K}\mathbf{R}\mathbf{L}^T) \\ &= 2\mathrm{Tr}(\mathbf{K}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{L}^T - \mathbf{B}\mathbf{H}^T\mathbf{L}^T + \mathbf{K}\mathbf{R}\mathbf{L}^T) \\ &= 2\mathrm{Tr}\{[\mathbf{K}(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}) - \mathbf{B}\mathbf{H}^T]\mathbf{L}^T\}\end{aligned}$$

The last line shows that the derivative is zero for any choice of $\mathbf{L}$ if and only if $(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})\mathbf{K}^T - \mathbf{H}\mathbf{B} = 0$, which is equivalent to

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

because $(\mathbf{HBH}^{\mathrm{T}} + \mathbf{R})$ is assumed to be invertible.

In the case of Gaussian pdfs, one can model the background, observation and analysis pdfs as follows, respectively: (b, o and a are normalization factors.)

$$\mathcal{P}_{\mathrm{b}}(\mathbf{x}) = b \exp\left[\frac{1}{2}(\mathbf{x}-\mathbf{x}_{\mathrm{b}})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_{\mathrm{b}})\right]$$

$$\mathcal{P}_{\mathrm{o}}(\mathbf{x}) = o \exp\left[\frac{1}{2}(\mathbf{y}-H[\mathbf{x}])^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}_{\mathrm{b}}])\right]$$

$$\mathcal{P}_{\mathrm{a}}(\mathbf{x}) = \mathcal{P}_{\mathrm{b}}(\mathbf{x})\mathcal{P}_{\mathrm{o}}(\mathbf{x})$$

which yields the right averages and covariances for the background and observations errors, and the analysis error pdf is simply defined as the Bayesian product of the two known sources of information, the background and the observation pdfs (this can be derived rigorously by using Bayes' theorem to write $\mathcal{P}_{\mathrm{a}}$ as a conditional probability of $\mathbf{x}$ given the observations and the a priori pdf of the background). Then, by taking minus the logarithm of $\mathcal{P}_{\mathrm{a}}(\mathbf{x})$, one finds that the model state with the maximum probability (or *likelihood*) is the one that minimizes the cost function $J(\mathbf{x})$ expressed in the theorem.

4.3 Comments

The hypotheses of non-triviality can always been made in well-posed analysis problems: if $\mathbf{B}$ is non-positive, one can restrict the control space to the orthogonal of the kernel of $\mathbf{B}$ (the analysis will not make any correction to background variables that are perfectly known). If $\mathbf{H}$ is not a surjection, then some observations are redundant and the observing network shall be restricted to the image of $\mathbf{H}$. If $\mathbf{R}$ is non-positive, the expression (A2) for $\mathbf{K}$ still holds (then the analysis will be equal to the observed value at the observation points $H(\mathbf{x}_{\mathrm{a}})$), but the variational version of the least-squares analysis cannot be used. It is even possible (with some algebraic precautions) to have some infinite eigenvalues in $\mathbf{R}$, i.e. a non-positive $\mathbf{R}^{-1}$, which means that some observations are not used because their errors are infinite.

The hypothesis of *unbiased errors* is a difficult one in practice because there often are significant biases in the background fields (caused by biases in the forecast model) and in the observations (or in the observation operators). If the biases are known, they can be subtracted from the background and observation values, and the above algebra applies to the debiased quantities. If the biases are left in, the analysis will not be optimal, even though it will seem to reduce the biases by interpolating between the background and observations. It is important to monitor the biases in an assimilation system, e.g. by looking at averages of background departures, but it is not trivial to decide which part of these are model or observation biases. The problem of bias monitoring and removal is the subject of ongoing research.

The hypothesis of *uncorrelated errors* is usually justified because the causes of errors in the background and in the observations are supposed to be completely independent. However, one must be careful about observation preprocessing practices (such as satellite retrieval procedures) that use the background field in a way that biases the observations toward the background. It might reduce the apparent background departures, but it will cause the analysis to be suboptimal (too close to the background, a condition nicknamed as the *incest problem*).

The *tangent linear hypothesis* is not trivial and it is commented in the next section.

It is possible to rewrite the least-squares analysis equations in terms of the *inverses* of the error covariance matrices, called *information* matrices. It makes the algebra a bit more complicated, but it allows one to see clearly that the *information* contained in the analysis is the sum, in a simple sense, of the observations provided by the background and by the observations. This is illustrated in the section on the estimation of analysis quality below.

It will be shown in the section on dual algorithms (PSAS analysis) that the equations, and in particular the cost function J, can be rewritten in the space of the observations $\mathbf{y}$. Also, it is easy to that least-squares analysis is closely related to a linear regression between model state and observations.

4.4 On the tangent linear hypothesis

The hypothesis of *linearized observation operator* is needed in order to derive a rigorous algebraic expression for the optimal $\mathbf{K}$. In practice, H may not be linear, but it usually makes physical sense to linearize it in the vicinity of the background state:

$$H(\mathbf{x}) - H(\mathbf{x}_b) \approx \mathbf{H}(\mathbf{x} - \mathbf{x}_b)$$

Then, $\mathbf{K}$ being a continuous function of $\mathbf{H}$, the least-squares equations for the analysis should intuitively yield a nearly optimal $\mathbf{x}_a$.

More generally, the *tangent linear hypothesis* on H can be written as the first-order Taylor–Young formula in the vicinity of an arbitrary state $\mathbf{x}$ and for a perturbation h:

$$H(\mathbf{x} + h) = H(\mathbf{x}) + \mathbf{H}h + \mathrm{O}(\|h\|^2),$$

with $\lim_{h \to 0} \mathrm{O}(\|h\|^2)h^{-2} = 0$. This hypothesis, called the *tangent linear hypothesis* is only acceptable if the higher-order variations of H can be neglected (in particular there should be no discontinuities) for all perturbations of the model state which have the same order of magnitude as the background errors. The operator $\mathbf{H}$ is called the *differential*, or *first derivative*, or *tangent linear (TL)*[16] function of H at point $\mathbf{x}$. Although this is a desirable mathematical property of H, it is not enough for practical purposes, because the approximation

$$H(\mathbf{x} + h) - H(\mathbf{x}) \approx \mathbf{H}h$$

must be satisfactory, in user-defined terms, for finite values of h that depend on the application considered. In the least-squares analysis problem, we need

$$\mathbf{y} - H(\mathbf{x}) \approx \mathbf{y} - \mathbf{H}(\mathbf{x} - \mathbf{x}_b) + H(\mathbf{x}_b)$$

for all values of $\mathbf{x}$ that will be encountered in the analysis procedure, notably $\mathbf{x} = \mathbf{x}_a$, $\mathbf{x} = \mathbf{x}_t$, and also all trial values used in the minimization of $J(\mathbf{x})$ if a variational analysis is performed[17]. Thus the important requirement is that the difference between $H(\mathbf{x}) - H(\mathbf{x}_b)$ and $\mathbf{H}(\mathbf{x} - \mathbf{x}_b)$ should be much smaller than the typical observation errors (defined by $\mathbf{R}$), for *all* model state perturbations $\mathbf{x} - \mathbf{x}_b$ of size and structure consistent with typical background errors, and also with the amplitude of the analysis increments $\mathbf{x}_a - \mathbf{x}_b$.

Thus the problem of linearizing H is not just related to the observation errors themselves. It must be appreciated in terms of the errors in the background $\mathbf{x}_b$ too, which in a sequential assimilation system are the previous forecast errors, which depend on the forecast range and the quality of the model. Ultimately the correctness of the linearization must be appreciated in the context of the fully integrated assimilation system. It will be easier to apply the

16. Both qualifiers *tangent* and *linear* are needed: obviously $\mathbf{H}$ could be linear without satisfying the Taylor formula. A function can also be tangent to another without being linear, if the difference between them is an $\mathrm{O}(\|h\|^2)$, e.g. x^2 and x^3 are tangent to each other for $x = 0$.

17. Qualitatively speaking they all belong to a neighbourhood of $\mathbf{x}_b$ having a shape and size which is consistent with the $\mathbf{B}$ and $\mathbf{R}$ error covariances.

linearization to a good system because the departures $\mathbf{x} - \mathbf{x}_b$ will be smaller. Conversely, the linearization may be inapplicable to difficult data assimilation problems. This is often the case with ocean models or satellite data, which means that it can be wrong to use sophisticated analysis algorithms that rely too much on the linearity of the problem.

The linearization problem can be even more acute for the linearization of the model forecast operator M which is needed in 4D-Var and in the Kalman filter described below. As with the linearization of H, it may or may not be licit depending on the quality of all components of the assimilation system: data coverage, observation quality, model resolution and physics, and forecast range. The user requirements and the physical properties of the system must be considered.

The non-linear analysis problem

The assumption of linear analysis is a strong one. Linear algebra is needed to derive the optimal analysis equations. One can rely on the linearization of a weakly non-linear observation operator, at the expense of optimality. The *incremental method* (described below for the variational analysis) performs this procedure iteratively in an empirical attempt to make the analysis more optimal. For strongly non-linear problems, there is no general and simple way to calculate the optimal analysis. The *simulated annealing method* can be useful; specific methods, such as the *simplex*, deal with variables with bounded definition domains. Finally, it is sometimes possible to make a problem more linear simply by a clever definition of model and observation variables (see the section on minimization methods).

4.5 The point of view of conditional probabilities

It is interesting to formalize the analysis problem using the conditional, or *Bayesian*, probabilities. Let us denote $P(x)$ the a priori pdf (probability density function) of the model state before the observations are considered, i.e. the background pdf. Let us denote $P(y)$ the pdf of the observations. The aim of the analysis is to find the maximum of $P(x|y)$, the conditional probability of the model state given the observations. The *joint pdf* of x and y (i.e. the probability that x and y occur together) is

$$P(x \wedge y) = P(x|y)P(y) = P(y|x)P(x)$$

i.e. it is the probability that x occurs when y occurs, and vice versa. The above expression is the Bayes theorem. In the analysis procedure we know that a measurement has been made and we know its value y, so $P(y) = 1$ and we obtain

$$P(x|y) = P(y|x)P(x)$$

which means that the analysis pdf is equal to the background pdf times the observation pdf $P(y|x)$. The latter peaks at $y = H(x)$ but it is not a Dirac distribution because the observations are not error-free.

The virtue of the probabilistic derivation of the analysis problem is that it can be extended to non-Gaussian probabilities (although this spoils the equivalence with the (A2) equation for $\mathbf{K}$). A practical application is done in the framework of variational quality control, where it is assumed that observation errors are not Gaussian but they contain some amount of “gross errors”, i.e. there is a probability that the error is not generated by the usual Gaussian physical processes but by some more serious problem, like coding or instrumental failure. The gross errors might be modelled using a uniform pdf over a predefined interval of admissible gross errors, leading to a non-Gaussian observation pdf. When the opposite of the logarithm of this pdf is taken, the resulting observation cost function is not quadratic, but gives less weight to the observation (i.e. there is less slope) for model states that disagree strongly with the observed value.

ref: Lorenc 1986

4.6 Numerical cost of least-squares analysis

In current operational meteorological models, the dimension of the model state (or, more precisely, of the control variable space) $\mathbf{x}$ is of the order of $n = 10^7$, and the dimension of the observation vector (the number of observed scalars) is of the order of $p = 10^5$ per analysis[18]. Therefore the analysis problem is mathematically underdetermined (although in some regions it might be overdetermined if the density of the observations is larger than the resolution of the model). In any practical application it is essential to keep in mind the size of the matrix operators involved in computing the analysis (Fig. 4). The least-squares analysis method requires in principle the specification of covariance matrices $\mathbf{B}$ and $\mathbf{R}$ (or their inverses in the variational form of the algorithm) which respectively contain of the order of $n^2/2$ and $p^2/2$ distinct coefficients, which are statistics to estimate (the estimation of a variance or covariance statistic converges like the square root of the number of realizations). The explicit determination of $\mathbf{K}$ requires the inversion of a matrix of size $p \times p$, which has an asymptotic complexity of the order of $p^2 \log(n)$. The exact minimization of the cost function J requires, in principle, $n + 1$ evaluations of the cost function and its gradient, assuming J is quadratic and there are no numerical errors (e.g. using a conjugate gradient method).

18. At ECMWF in winter 1998 the control variable dimension was 512000, the number of active observations (per 6-hour interval) was about 150000

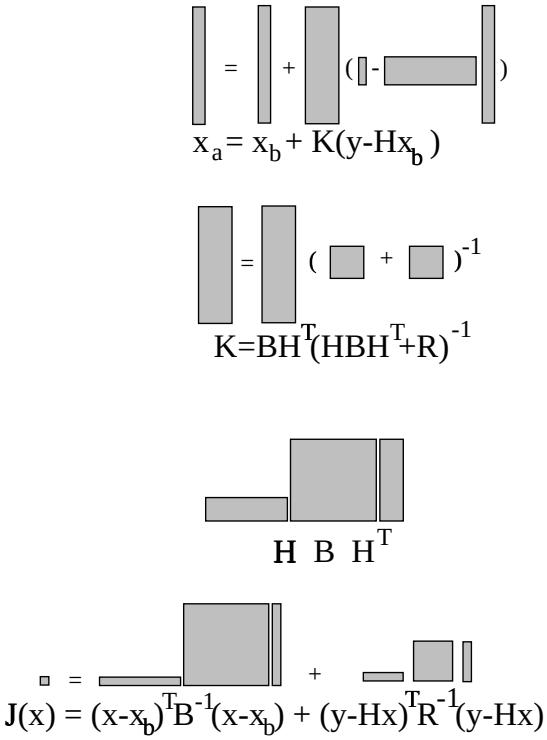

Figure 4. Sketches of the shapes of the matrices and vector dimensions involved in an usual analysis problem where there are many fewer observations than degrees of freedom in the model: from top to bottom, in the equations of the linear analysis, the computation of $\mathbf{K}$, of the $\mathbf{HBH}^T$ term, and the computation of the cost function J.

It is obvious that, except in analysis problems of very small dimension (like one-dimensional retrievals), it is impossible to compute exactly the least-squares analysis. Some approximations are necessary, they are the subject of the following sections.

4.7 Conclusion

We have seen that there are two main ways of defining the statistical analysis problem:

- either assume that the background and error covariances are known, and derive the analysis equations by requiring that the total analysis error variances are minimum,
- or assume that the background and observation error pdfs are Gaussian, and derive the analysis equations by looking for the state with the maximum probability.

Both approaches lead to two mathematically equivalent algorithms:

- the direct determination of the analysis gain matrix $\mathbf{K}$,
- the minimization of a quadratic cost function.

These algorithms have very different numerical properties, and their equivalence stops as soon as some underlying hypotheses are not verified, like the linearization of the observation operator, for instance.

5. A SIMPLE SCALAR ILLUSTRATION OF LEAST-SQUARES ESTIMATION

Let us assume that we need to estimate the temperature T_t of a room.

We have a thermometer of known accuracy σ_o (the standard deviation of measurement error) and we observe T_o, which is considered to have expectation T_t (i.e. we assume that the observation is unbiased) and variance σ_o^2. In the absence of any other information the best estimate we can provide of the temperature is T_o, with accuracy σ_o.

However we may have some additional information about the temperature of the room. We may have a reading from another, independent thermometer, perhaps with a different accuracy. We may notice that everyone in the room is wearing a jumper—another timely piece of information from which we can derive an estimate, although with a rather large associated error. We may have an accurate observation from an earlier date, which can be treated as an estimate for the current time, with an error suitably inflated to account for the separation in time. Any of these observations could be treated as *a priori* or *background* information, to be used with T_o in estimating the room temperature. Let our background estimate be T_b, of expectation T_t (i.e. it is unbiased) and of accuracy σ_b. Intuitively T_o and T_b can be combined to provide a better estimate (or *analysis*) of T_t than any of these taken alone. We are going to look for a linear weighted average of the form:

$$T_a = kT_o + (1-k)T_b$$

which can be rewritten as $T_a = T_b + k(T_o - T_b)$, i.e. we look for a correction to the background which is a linear function of the difference between the observation and the background.

The error variance of the estimate is:

$$\sigma_a^2 = (1-k)^2\sigma_b^2 + k^2\sigma_o^2$$

where we have assumed that the observation and background errors are uncorrelated. We choose the optimal value of k that minimizes the analysis error variance:

$$k = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2}$$

which is equivalent to minimizing (Fig. 5)

$$J(T) = J_b(T) + J_o(T) = \frac{(T-T_b)^2}{\sigma_b^2} + \frac{(T-T_o)^2}{\sigma_b^2}$$

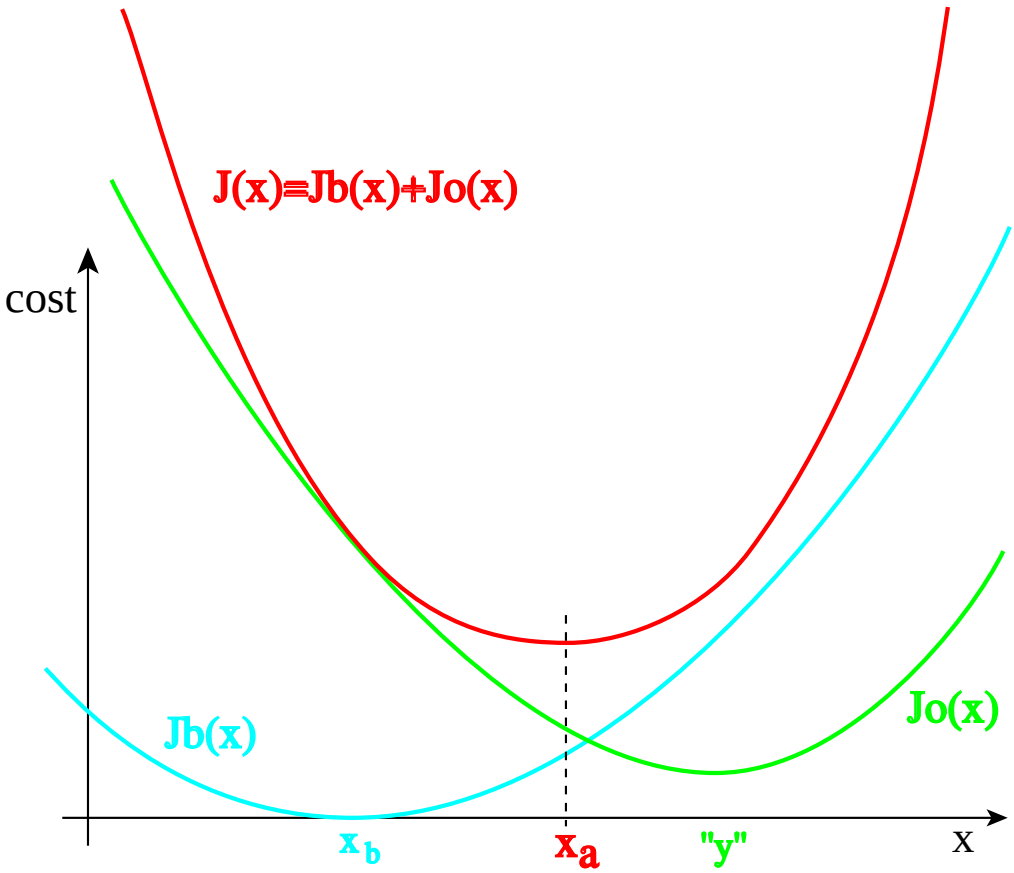

Figure 5. Schematic representation of the variational form of the least-squares analysis, in a scalar system where the observation y is in the same space as the model x: the cost-function terms J_b and J_o are both convex and tend to "pull" the analysis towards the background x_b and the observation y, respectively. The minimum of their sum is somewhere between x_b and y, and is the optimal least-squares analysis.

- In the limiting case of a very low quality measurement ($\sigma_o >> \sigma_b$), $k = 0$ and the analysis remains equal to the background.
- On the other hand, if the observation has a very high quality ($\sigma_o << \sigma_b$), $k = 1$ and the analysis is equal to the observation.
- If both have the same accuracy, $\sigma_o = \sigma_b$, $k = 1/2$ and the analysis is simply the average of T_o and T_b, which reflects the fact that we trust as much the observation as the background, so we make a compromise.
- In all cases, $0 \le k \le 1$, which means that the analysis is a weighted average of the background and the observation.

These situations are sketched in Fig. 6 .

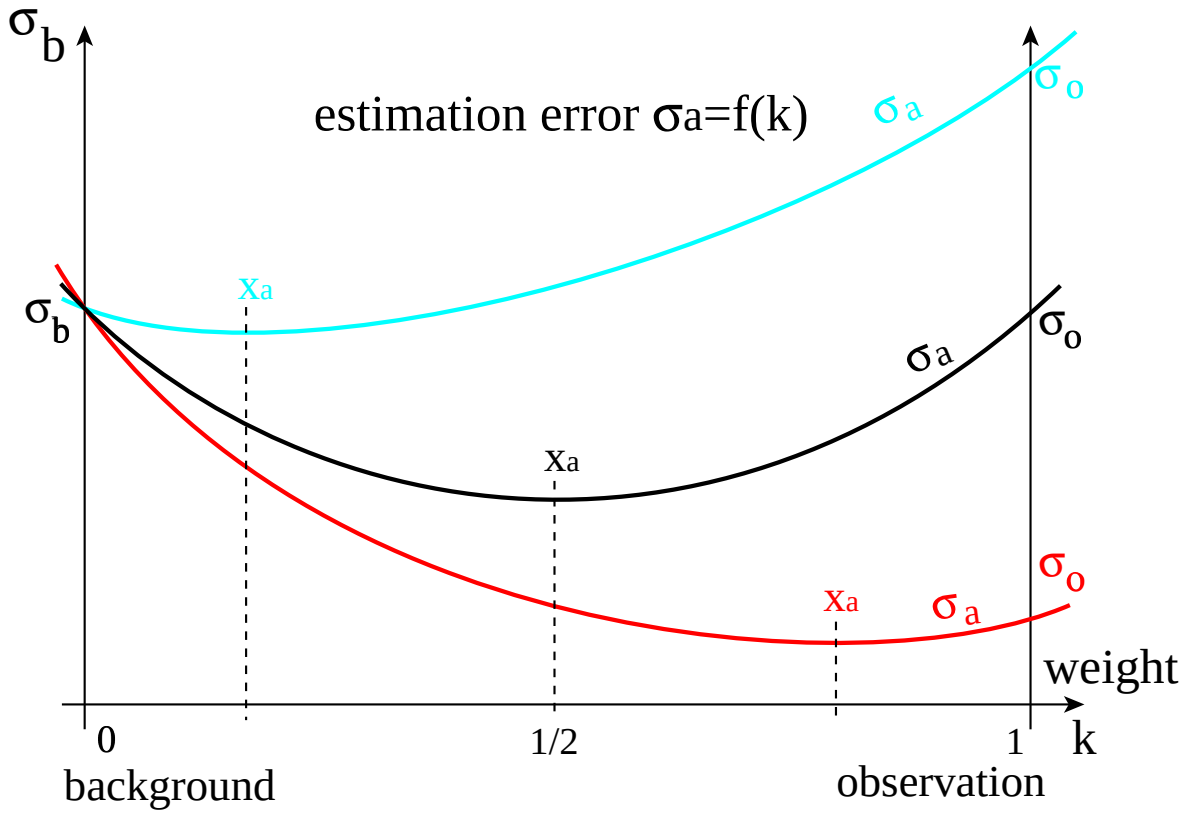

Figure 6. Schematic representation of the variations of the estimation error σ_a, and of the optimal weight k that determines the analysis $\mathbf{x}_a$, for various relative amplitudes of the background and observation standard errors (σ_b, σ_o).

It is interesting to look at the variance of analysis error for the optimal k:

$$\frac{1}{\sigma_a^2} = \frac{1}{\sigma_o^2} + \frac{1}{\sigma_b^2}$$

or

$$\sigma_a^2 = \frac{\sigma_o^2}{1 + (\sigma_o/\sigma_b)^2} = \frac{\sigma_b^2}{1 + (\sigma_b/\sigma_o)^2} = (1-k)\sigma_b^2$$

which shows that the analysis error variance is always smaller than both the background and observation error variances, and it is smallest if both are equal, in which case the analysis error variance is half the background error variance.

6. MODELS OF ERROR COVARIANCES

A correct specification of observation and background error covariances is crucial to the quality of the analysis, because they determine to what extent the background fields will be corrected to match the observations. The es-

sential parameters are the variances, but the correlations are also very important because they specify how the observed information will be smoothed in model space if there is a mismatch between the resolution of the model and the density of the observations. In the framework of Kalman filtering and 4D assimilation with model as a weak constraint, a third kind of covariances to specify is $\mathbf{Q}$, the model error covariances (see the relevant section below).

6.1 Observation error variances

They are mainly specified according to the knowledge of instrumental characteristics, which can be estimated using collocated observations, for instance. As explained before, they should also include the variance of representativeness errors which is not negligible when analysing phenomena which cannot be well represented in model space. It is wrong to leave observation biases as a contribution to the observation error variances because it will produce biases in the analysis increments; whenever observation biases can be identified, they should be removed from the observed value or from the background fields, depending on whether one thinks they are caused by problems in the model or in the observation procedure (unfortunately we do not always know what to decide).

6.2 Observation error correlations

They are often assumed to be zero, i.e. one believes that distinct measurements are affected by physically independent errors. This sounds reasonable for pairs of observations carried out by distinct instruments. This may not be true for sets of observations performed by the same platform, like radiosonde, aircraft or satellite measurements, or when several successive reports from the same station are used in 4D-Var. Intuitively there will be a significant observation error correlation for reports close to one another. If there is a bias it will show up as a permanent observation error correlation. The observation preprocessing can generate artificial correlations between the transformed observations e.g. when temperature profiles are converted to geopotential, or when there is a conversion between relative and specific humidity (correlation with temperature), or when a retrieval procedure is applied to satellite data. If the background is used in the observation preprocessing, this will introduce artificial correlations between observations and background errors which are difficult to account for: moving the observation closer to the background may make the observation and background errors look smaller, but it will unrealistically reduce the weight of the originally observed information. Finally, representativeness errors are correlated by nature: interpolation errors are correlated whenever observations are dense compared to the resolution of the model. Errors in the design of the observation operator, like forecast model errors in 4D-Var, are correlated on the same scales as the modelling problems.

The presence of (positive) observation error correlations can be shown to reduce the weight given to the average of the observations, and thus give more relative importance to differences between observed values, like gradients or tendencies. Unfortunately observation error correlations are difficult to estimate and can create problems in the numerics of the analysis and quality control algorithms. In practice it often makes sense to try to minimize them by working on a bias correction scheme, by avoiding unnecessary observation preprocessing, by thinning dense data and by improving the design of the model and observation operators. Most models of $\mathbf{R}$ covariances used in practice are diagonal or almost.

6.3 Background error variances

They are usually estimates of the error variances in the forecast used to produce $\mathbf{x}_b$. In the Kalman filter they are estimated automatically using the tangent-linear model, so they do not need to be specified (although this means that the problem is moved to the specification of the model error $\mathbf{Q}$ and the tuning of approximated algorithms that are less costly than the complete Kalman filter). A crude estimate can be obtained by taking an arbitrary fraction of climatological variance of the fields themselves. If the analysis is of good quality (i.e. if there are a lot of observations) a better average estimate is provided by the variance of the differences between the forecast and a verifying

analysis. If the observations can be assumed to be uncorrelated, much better averaged background error variances can be obtained by using the *observational* method explained below. However, in a system like the atmosphere the actual background errors are expected to depend a lot on the weather situation, and ideally the background errors should be flow-dependent. This can be achieved by the Kalman filter, by 4D-Var to some extent, or by some empirical laws of error growth based on physical grounds. If background error variances are badly specified, it will lead to too large or too small analysis increments. In least-squares analysis algorithms, only the relative magnitude of the background and observation error variances is important. However, the absolute values may be important if they are used to make quality-control decisions on the observations (it is usually desirable to accept more easily the observations with a large background departure if the background error is likely to be large).

6.4 Background error correlations

They are essential for several reasons:

Information spreading. In data-sparse areas, the shape of the analysis increment is completely determined by the covariance structures (for a single observation it is given by $\mathbf{BH}^{\mathrm{T}}$). Hence the correlations in $\mathbf{B}$ will perform the spatial spreading of information from the observation points (real observations are usually local) to a finite domain surrounding it.

Information smoothing. In data-dense areas, one can show that in the presence of discrete observations (which is the usual case) the amount of smoothing[19] of the observed information is governed by the correlations in $\mathbf{B}$, which can be understood by remarking that the leftmost term in $\mathbf{K}$ is $\mathbf{B}$. The smoothing of the increments is important in ensuring that the analysis contains scales which are statistically compatible with the smoothness properties of the physical fields. For instance, when analysing stratospheric or anticyclonic air masses, it is desirable to smooth the increments a lot in the horizontal in order to average and spread efficiently the measurements. When doing a low-level analysis in frontal, coastal or mountainous areas, or near temperature inversions, it is desirable on the contrary to limit the extent of the increments so as not to produce an unphysically smooth analysis. This has to be reflected in the specification of background error correlations.

Balance properties. There are often more degrees of freedom in a model than in reality. For instance, the large-scale atmosphere is usually hydrostatic. It is almost geostrophic, at least there is always a large amount of geostrophy in the extratropics. These *balance* properties could be regarded as annoying constraints on the analysis problem, and enforced brutally e.g. using an a posteriori normal-mode initialization. On the other hand, they are statistical properties that link the different model variables. In other words, they show up as correlations in the background errors because the existence of a balance in the reality and in the model state will imply that there is a (linearized) version of the balance that exists in the background error covariances, too. This is interesting for the use of observed information: observing one model variable yields information about all variables that are balanced with it, e.g. a low-level wind observation allows one to correct the surface pressure field by assuming some amount of geostrophy. When combined with the spatial smoothing of increments this can lead to a considerable impact on the quality of the analysis, e.g. a temperature observation at one point can be smoothed to produce a correction to geopotential height around it, and then produce a complete three-dimensional correction of the geostrophic wind field (Fig. 7). The relative amplitude of the increments in terms of the various model fields will

19. There is an equivalence between statistical analysis and the theory of interpolation by splines.

depend directly on the specified amount of correlation as well as on the assumed error variance in all the concerned parameters.

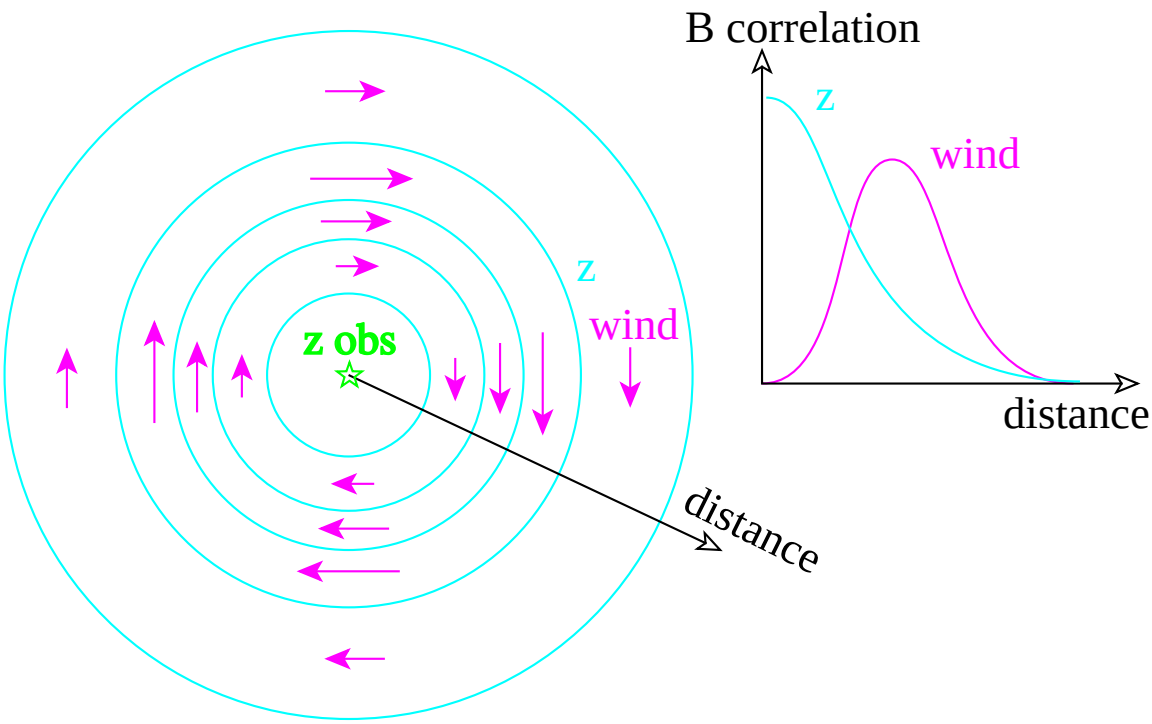

Figure 7. Example of horizontal structure functions commonly used in meteorology: the horizontal autocorrelation of height (or pressure) has an isotropic, gaussian-like shape as a function of distance (right panel). In turn, geostrophy implies that wind will be cross-correlated with height at distances where the gradient of height correlation is maximum. Hence, an isolated height observation will generate an isotropic height "bump" with a rotating wind increment in the shape of a ring.

Ill-conditioning of the assimilation. It is possible to include into the control variables some additional parameters which are not directly observed, like model tuning parameters or bias estimates. This can be an efficient indirect parameter estimation technique if there is a realistic coupling with the observed data, usually through the design of the observation operator or of the model (in a 4-D assimilation). It may not be possible or sensible to specify explicit correlations with the rest of the model state in **B** . However, one must be careful to specify a sensible background error for all parameters of the control variable, unless it is certain that the problem is over-determined by the observations. A too small error variance will obviously prevent any correction to the additional parameters. A too large variance may on the other hand make the additional parameters act like a sink of noise, exhibiting variations whenever it improves the fit of the analysis to observations, even if no such correction of the additional parameters is physically justified. This can create genuine problems because some implicit analysis coupling is often created by variable dependencies in the observation operators or in the model (in 4D-Var). Then, the specification of background errors for additional parameters will have an impact on the analysis of the main model state. They should reflect the acceptable amplitude of the analysis corrections.

Flow-dependent structure functions. If enough is known about the dynamics of the problem, one can make **B** depend on the uncertainty of the previous analysis and forecast, not only in terms of background error variances, but also in the correlations. In geophysical fluids there is not just a loss of predictability during the forecast, there are waves that follow specific patterns, and these patterns are expected to be found in the background errors. For instance, in an area prone to cyclogenesis, one expects the most likely background errors to have the shape (or *structure function*) of the most unstable structures, perhaps with a baroclinic wave tilt, and anticorrelations between the errors in the warm and in the cold air masses. This is equivalent to a balance property, and again if the relevant information can be embedded into the correlations of **B** , then the observed information can be more accurately spread spatially and distributed among all model parameters involved. Such information can be provided in the framework of a Kalman filter or 4D-Var.

ref: Courtier *et al.* 1998

6.5 Estimation of error covariances

It is a difficult problem, because they are never observed directly, they can only be estimated in a statistical sense, so that one is forced to make some assumptions of homogeneity. The best source of information about the errors in an assimilation system is the study of the background departures ($\mathbf{y} - H[\mathbf{x}_b]$) and they can be used in a variety of ways. Other indications can be obtained from the analysis departures, or from the values of the cost functions in 3D/4D-Var. There are some more empirical methods based on the study of forecasts started from the analyses, like the NMC method or the adjoint sensitivity studies, but their theoretical foundation is rather unclear for the time being. A comprehensive and rigorous methodology is being developed under the framework of *adaptive filtering* which is too complex to explain in this volume. Probably the most simple yet reliable estimation method is the observational method explained below.

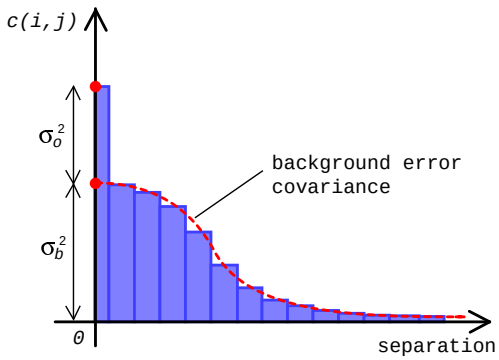

Figure 8. Schematic representation the observational method. The (observation – background) covariance statistics for a given assimilation system are stratified against distance, and the intercept at the origin of the histogram provides an estimate of the average background and observation error variances for these particular assimilation and observation systems.

The observational (or Hollingworth–Lonnberg) method. This method[20] relies on the use of background departures in an observing network that is dense and large enough to provide information on many scales, and that can

20. named after the authors that popularized it in meteorology, although it was known and used before in geophysics. The widespread *kriging method* is closely related.

be assumed to consist of uncorrelated and discrete observations. The principle (illustrated in Fig. 8) is to calculate an histogram (or *variogram*) of background departure covariances, stratified against separation (for instance). At zero separation the histogram provides averaged information about the background and observation errors, at non-zero separation it gives the averaged background error correlation: if i and j are two observation points, the background departure covariance $c(i, j)$ can be calculated empirically and it is equal to

$$\begin{aligned} c(i,j) &= \overline{(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_\mathrm{b})(\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_\mathrm{b})^\mathrm{T}} \\ &= \overline{[(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_\mathrm{t}) + (\mathbf{H}_i\mathbf{x}_\mathrm{t} - \mathbf{H}_i\mathbf{x}_\mathrm{b})][(\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_\mathrm{t}) + (\mathbf{H}_j\mathbf{x}_\mathrm{t} - \mathbf{H}_j\mathbf{x}_\mathrm{b})]^\mathrm{T}} \\ &= \overline{(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_\mathrm{t})(\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_\mathrm{t})^\mathrm{T}} + \mathbf{H}_i\overline{(\mathbf{x}_\mathrm{t} - \mathbf{x}_\mathrm{b})(\mathbf{x}_\mathrm{t} - \mathbf{x}_\mathrm{b})^\mathrm{T}}\mathbf{H}_j^\mathrm{T} + \\ &\quad + \overline{(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_\mathrm{t})(\mathbf{x}_\mathrm{t} - \mathbf{x}_\mathrm{b})^\mathrm{T}}\mathbf{H}_j^\mathrm{T} + \mathbf{H}_i\overline{(\mathbf{x}_\mathrm{t} - \mathbf{x}_\mathrm{b})(\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_\mathrm{t})^\mathrm{T}} \\ &= \mathbf{R}_{ij} + \mathbf{H}_i\mathbf{B}\mathbf{H}_j^\mathrm{T} + 0 + 0 \end{aligned}$$

If one assumes that there is no correlation between observation and background errors, the last two terms on the second line vanish. The first term is the observation error covariance between i and j, the second term is the background error covariance interpolated at these points, assuming both are homogeneous over the dataset used. In summary,

$$c(i,j) = \mathbf{R}_{ij} + \mathbf{H}_i\mathbf{B}\mathbf{H}_j^\mathrm{T}$$

- if $i = j$, $c(i,j) = \sigma_\mathrm{o}^2(i) + \sigma_\mathrm{b}^2(i)$, the sum of the observation and the background error variances,
- if $i \neq j$ and the observation errors are assumed to be uncorrelated, $c(i,j) = \mathrm{cov}_\mathrm{b}(i,j)$, the background error covariance between i and j. (If there are observation error correlations, it is impossible to disentangle the information about $\mathbf{R}$ and $\mathbf{B}$ without additional data)
- Under the same assumption, if i and j are very close to each other without being equal, then $\lim_{i \to j} c(i,j) = \sigma_\mathrm{b}^2(i)$, so that by determining the intercept for zero separation of $c(i,j)$, one can determine $\sigma_\mathrm{b}^2(i)$.
- Then, one gets $\sigma_\mathrm{o}^2(i) = c(i,j) - \sigma_\mathrm{b}^2(i)$ and the background error correlations are given by $(c(i,j))/\sigma_\mathrm{b}^2(i)$ (we have assumed that the background error variances are homogeneous over the considered dataset).

In most systems the background error covariances should go to zero for very large separations. If this is not the case, it is usually the sign of biases in the background and/or in the observations and the method may not work correctly (Hollingsworth and Lonnberg 1986.).

6.6 Modelling of background correlations

As explained above the full $\mathbf{B}$ matrix is usually too big to be specified explicitly. The variances are just the n diagonal terms of $\mathbf{B}$, which are usually specified completely. The off-diagonal terms are more difficult to specify. They must generate a symmetric positive definite matrix, so one must be careful about the assumptions made to specify them. Additionally $\mathbf{B}$ is often required to have some physical properties which are required to be reflected in the analysis:

- the correlations must be smooth in physical space, on sensible scales,
- the correlations should go to zero for very large separations if it is believed that observations should only have a local effect on the increments,

- the correlations should not exhibit physically unjustifiable variations according to direction or location,
- the most fundamental balance properties, like geostrophy, must be reasonably well enforced.
- the correlations should not lead to unreasonable effective background error variances for any parameter that is observed, used in the subsequent model forecast, or output to the users as an analysis product.

The complexity and subtlety of these requirements mean that the specification of background error covariances is a problem similar to physical parametrization. Physically sound hypotheses need to be made and tested carefully. Some of the more popular techniques are listed below, but more sophisticated ones remain to be invented.

- Correlation models can be specified independently from variance fields, under the condition that the scales of variation of the variances are much larger than the correlation scales, otherwise the shape of the covariances would differ a lot from the correlations, with unpredictable consequences on the balance properties.
- Vertical autocorrelation matrices for each parameter are usually small enough to be specified explicitly.
- Horizontal autocorrelations cannot be specified explicitly, but they can be reduced to sparse matrices by assuming that they are homogeneous and isotropic to some extent. It implies that they are diagonal in spectral space[21]. In grid-point space some low-pass digital filters can be applied to achieve a similar result.
- Three-dimensional multivariate correlation models can be built by carefully combining *separability*, *homogeneity* and *independency* hypotheses like: zero correlations in the vertical for distinct spectral wavenumbers, homogeneity of the vertical correlations in the horizontal and/or horizontal correlations in the vertical, property of the correlations being products of horizontal and vertical correlations. Numerically they imply that the correlation matrix is sparse because it is made of block matrices which are themselves block-diagonal[22]
- Balance constraints can be enforced by transforming the model variables into suitably defined complementary spaces of *balanced* and *unbalanced* variables. The latter are supposed to have smaller background error variances than the former, meaning that they will contribute less to the increment structures.
- The geostrophic balance constraint can be enforced using the classical f-plane or β-plane balance equations, or projections onto subspaces spanned by so-called Rossby and Gravity normal modes.
- More general kinds of balance properties can be expressed using linear regression operators calibrated on actual background error fields, if no analytical formulation is available.

Two last requirements which can be important for the numerical implementation of the analysis algorithm are the availability of the symmetric square root of $\mathbf{B}$ (a matrix L such that $LL^{\mathrm{T}} = \mathbf{B}$) and of its inverse. They can constrain notably the design of $\mathbf{B}$.

ref: Courtier *et al.* 1998

7. Optimal interpolation (OI) analysis

The OI is an algebraic simplification of the computation of the weight $\mathbf{K}$ in the analysis equations (A1) and (A2).

21. This is the Khinchine-Bochner theorem. The spectral coefficients are proportional to the spectral variance of the correlations for each total wavenumber. This is detailed on the sphere in Courtier *et al.* (1996).

22. It corresponds to the mathematical concept of *tensor product*.

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - H[\mathbf{x}_b]) \tag{A1}$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{A2}$$

The equation (A1) can be regarded as a list of scalar analysis equations, one per model variable in the vector $\mathbf{x}$. For each model variable the analysis increment is given by the corresponding line of $\mathbf{K}$ times the vector of background departures $(\mathbf{y} - H[\mathbf{x}_b])$. The fundamental hypothesis in OI is: *For each model variable, only a few observations are important in determining the analysis increment*. It is implemented as follows:

1) For each model variable $\mathbf{x}(i)$, select a small number p_i of observations using empirical selection criteria.

2) Form the corresponding list of p_i background departures $(\mathbf{y} - H[\mathbf{x}_b])_i$, the p_i background error covariances between the model variable $\mathbf{x}(i)$ and the model state interpolated at the p_i observation points (i.e. the relevant p_i coefficients of the i-th line of $\mathbf{B}\mathbf{H}^T$), and the $p_i \times p_i$ background and observation error covariance submatrices formed by the restrictions of $\mathbf{H}\mathbf{B}\mathbf{H}^T$ and $\mathbf{R}$ to the selected observations.

3) Invert the $p_i \times p_i$ positive definite matrix formed by the restriction of $(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})$ to the selected observations (e.g. by an *LU* or Choleski method),

4) Multiply it by the i-th line of $\mathbf{B}\mathbf{H}^T$ to get the necessary line of $\mathbf{K}$.

It is possible to save some computer time on the matrix inversion by solving directly a symmetric positive linear system, since we know in advance the vector of departures to which the inverse matrix will be applied. Also, if the same set of observations is used to analyse several model variables, then the same matrix inverse (or factorization) can be reused.

In the OI algorithm it is necessary to have the background error covariances $\mathbf{B}$ as a model which can easily be applied to pairs of model and observed variables, and to pairs of observed variables. This can be difficult to implement if the observation operators are complex. On the other hand, the $\mathbf{B}$ matrix needs not be specified globally, it can be specified in an ad hoc way for each model variable, as long as it remains locally positive definite. The specification of $\mathbf{B}$ usually relies on the design of empirical autocorrelation functions (e.g. Gaussian or Bessel functions and their derivatives), and on assumed amounts of balance constraints like hydrostatic balance or geostrophy.

The selection of observations should in principle provide all the observations which would have a significant weight in the optimal analysis, i.e. those which have significant background error covariances $\mathbf{B}\mathbf{H}^T$ with the variable considered. In practice, background error covariances are assumed to be small for large separation, so that only the observations in a limited geometrical domain around the model variable need to be selected. For computational reasons it may be desirable to ensure that only a limited number of observations are selected each time, in order to keep the matrix inversions cheap. Two common strategies for observation selection are pointwise selection (Fig. 9) and box selection (Fig. 10)

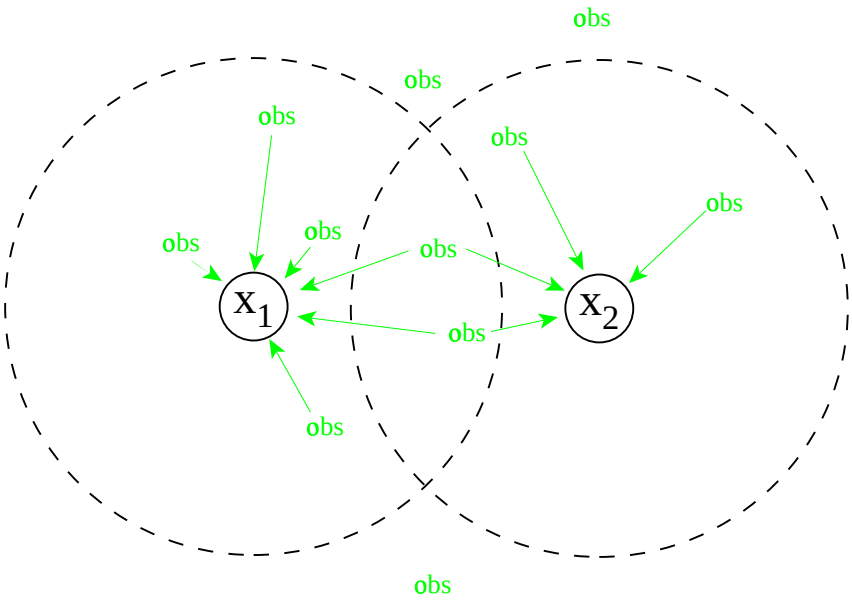

Figure 9. One OI data selection strategy is to assume that each analysis point is only sensitive to observations located in a small vicinity. Therefore, the observations used to perform the analysis at two neighbouring points x_1 or x_2 may be different, so that the analysis field will generally not be continuous in space. The cost of the analysis increases with the size of the selection domains.

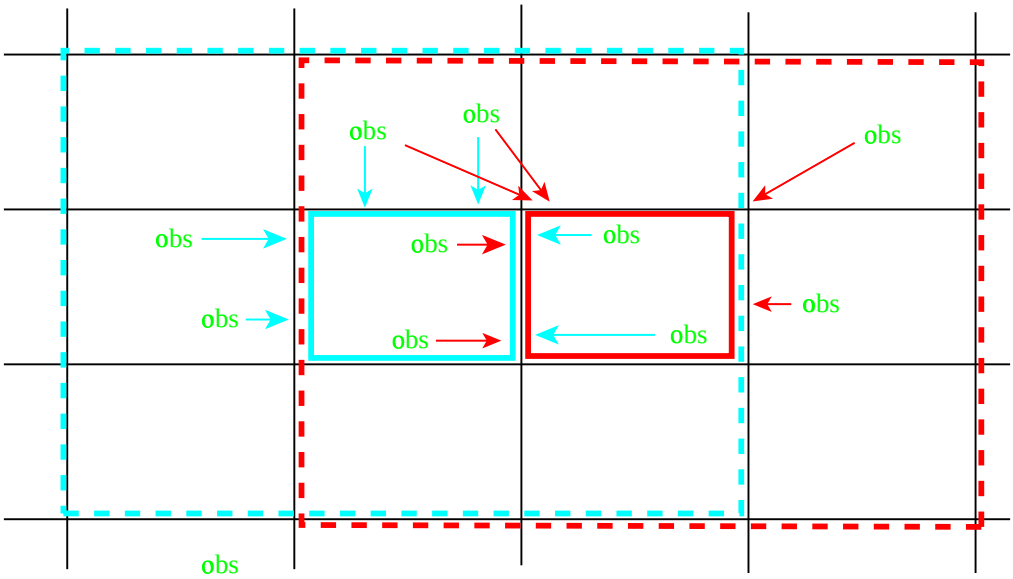

Figure 10. A slightly more sophisticated and more expensive OI data selection is to use, for all the points in an analysis box (black rectangle), all observations located in a bigger selection box (dashed rectangle), so that most of the observations selected in two neighbouring analysis boxes are identical.

The advantage of OI is its simplicity of implementation and its relatively small cost if the right assumptions can be made on the observation selection.

A drawback of OI is that spurious noise is produced in the analysis fields because different sets of observations (and possibly different background error models) are used on different parts of the model state. Also, it is impossible to guarantee the coherence between small and large scales of the analysis (Lorenc 1981).

8. THREE-DIMENSIONAL VARIATIONAL ANALYSIS (3D-VAR)

The principle of 3D-Var is to avoid the computation (A2) of the gain $\mathbf{K}$ completely by looking for the analysis as an approximate solution to the equivalent minimization problem defined by the cost function J in (A5). The solution is sought iteratively by performing several evaluations of the cost function

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) + (\mathbf{y}-H[\mathbf{x}])^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}])$$

and of its gradient

$$\nabla J(\mathbf{x}) = 2\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) - 2H^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}])$$

in order to approach the minimum using a suitable descent algorithm. The approximation lies in the fact that only a small number of iterations are performed. The minimization can be stopped by limiting artificially the number of iterations, or by requiring that the norm of the gradient $\|\nabla J(\mathbf{x})\|$ decreases by a predefined amount during the minimization, which is an intrinsic measure of how much the analysis is closer to the optimum than the initial point of the minimization. The geometry of the minimization is suggested in Fig. 11 .

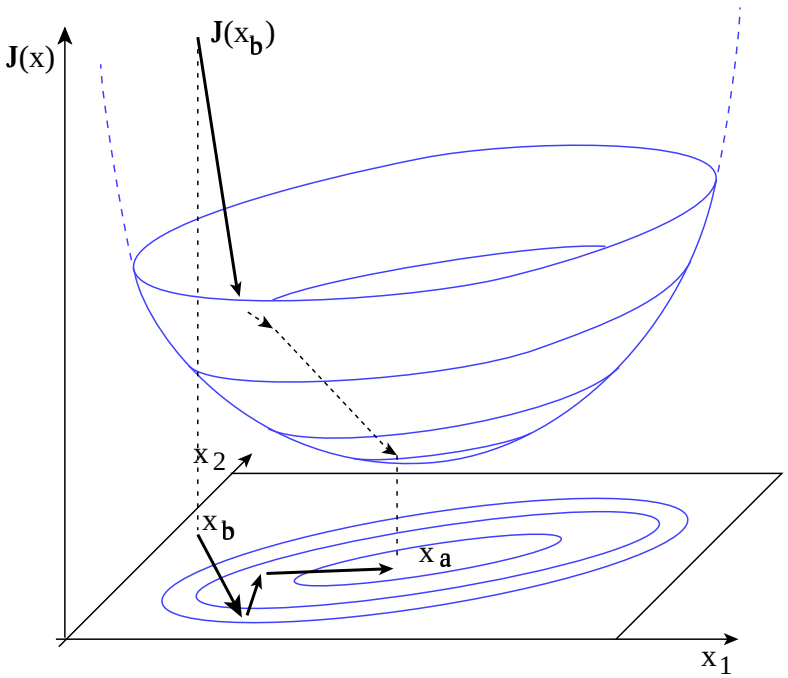

Figure 11. Schematic representation of the variational cost-function minimization (here in a two-variable model space): the quadratic cost-function has the shape of a paraboloid, or bowl, with the minimum at the optimal analysis $\mathbf{x}_a$. The minimization works by performing several line-searches to move the control variable $\mathbf{x}$ to areas where the cost-function is smaller, usually by looking at the local slope (the gradient) of the cost-function.

In practice, the initial point of the minimization, or *first guess*, is taken equal to the background $\mathbf{x}_\mathrm{b}$. This is not compulsory, however, so it is important to distinguish clearly between the terms *background* (which is used in the definition of the cost function) and *first guess* (which is used to initiate the minimization procedure). If the minimization is satisfactory, the analysis will not depend significantly on the choice of first guess, but it will always be sensitive to the background.

A significant difficulty with 3D-Var is the need to design a model for $\mathbf{B}$ that properly defines background error covariances for all pairs of model variables. In particular, it has to be symmetric positive definite, and the background error variances must be realistic when expressed in terms of observation parameters, because this is what will determine the weight of the observations in the analysis.

The popularity of 3D-Var stems from its conceptual simplicity and from the ease with which complex observation operators can be used, since only the operators and the adjoints of their tangent linear need to be provided[23]. Weakly non-linear observation operators can be used, with a small loss in the optimality of the result. As long as J is strictly convex, there is still one and only one analysis.

In most cases the observation error covariance matrix $\mathbf{R}$ is block-diagonal, or even diagonal, because there is no reason to assume observation error correlations between independent observing networks, observing platforms or stations, and instruments, except in some special cases. It is easy to see that a block-diagonal $\mathbf{R}$ implies that J_o is a sum of N scalar cost-functions $J_{\mathrm{o},i}$, each one defined by a submatrix $\mathbf{R}_i$ and the corresponding subsets H_i and $\mathbf{y}_i$ of the observation operators and values:

$$J_\mathrm{o} = \sum_{i=1}^{N} J_{\mathrm{o},i}(\mathbf{x})$$
$$J_{\mathrm{o},i} = (\mathbf{y}_i - H_i[\mathbf{x}])^\mathrm{T}\mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}])$$

The gradient ∇J_o can be similarly decomposed. The breakdown of J_o is a useful diagnostic tool of the behaviour of 3D-Var in terms of each observation type: the magnitude of each term measures the misfit between the state $\mathbf{x}$ and the corresponding subset of observations. It can also simplify the coding of the computations of J_o and its gradient[24].

Another advantage is the ability to enforce external weak (or *penalty*) constraints, such as balance properties, by putting additional terms into the cost function (usually denoted J_c). However, this can make the preconditioning of the minimization problem difficult.

ref: Parrish and Derber 1992, Courtier *et al.* 1998.

9. 1D-VAR AND OTHER VARIATIONAL ANALYSIS SYSTEMS

The essence of the 3D-Var algorithm is to rewrite a least-squares problem as the minimization of a cost-function. The method was introduced in order to remove the local data selection in the OI algorithm, thereby performing a global analysis of the 3-D meteorological fields, hence the name. Of course, the technique has been applied equally well to other problems in which the control variable is much smaller. A very successful example is the satellite data

23. whereas OI requires a background error covariance model between each observed variable and each model variable.

24. Actually the whole J_o can be decomposed into as many elementary cost functions as there are observed parameters, by redefining the observation space to be the eigenvectors of $\mathbf{R}$.

retrieval problem, in which the 1D-Var algorithm performs a local analysis of one atmospheric column (the model state) at the location of each satellite sounding such as TOVS radiances or microwave measurements. Similar variational techniques have been applied to the retrieval of surface wind fields from a collection of scatterometer ambiguous wind measurements or to the analysis of land surface properties in a numerical weather prediction model (in this case the control variable is more or less a column of the 3-D model, but the time dimension is taken into account as in 4D-Var). Except 1D-Var, these methods have no established name yet.

ref: Eyre 1987.

10. FOUR-DIMENSIONAL VARIATIONAL ASSIMILATION (4D-VAR)

4D-Var is a simple generalization of 3D-Var for observations that are distributed in time. The equations are the same, provided the observation operators are generalized to include a forecast model that will allow a comparison between the model state and the observations at the appropriate time.

Over a given time interval, the analysis being at the initial time, and the observations being distributed among n times in the interval, we denote by the subscript i the quantities at any given observation time i. Hence, $\mathbf{y}_i$, $\mathbf{x}_i$ and $\mathbf{x}_{ti}$ are the observations, the model and the true states at time i, and $\mathbf{R}_i$ is the error covariance matrix for the observation errors $\mathbf{y}_i - H_i(\mathbf{x}_{ti})$. The observation operator H_i at time i is linearized as $\mathbf{H}_i$. The background error covariance matrix $\mathbf{B}$ is only defined at initial time, the time of the background $\mathbf{x}_b$ and of the analysis $\mathbf{x}_a$.

10.1 The four-dimensional analysis problem

In its general form, it is defined as the minimization of the following cost function:

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_b) + \sum_{i=0}^{n}(\mathbf{y}_i - H_i[\mathbf{x}_i])^T\mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}_i])$$

which can be proven, like in the three-dimensional case detailed previously, to be equivalent to finding the maximum likelihood estimate of the analysis subject to the hypothesis of Gaussian errors.

The **4D-Var analysis**, or **four-dimensional variational assimilation problem**, is by convention defined as the above minimization problem subject to the strong constraint that the sequence of model states $\mathbf{x}_i$ must be a solution of the model equations:

$$\forall i, \quad \mathbf{x}_i = M_{0\rightarrow i}(\mathbf{x})$$

where $M_{0\rightarrow i}$ is a predefined model forecast operator from the initial time to i. 4D-Var is thus a nonlinear constrained optimization problem which is very difficult to solve in the general case. Fortunately it can be greatly simplified with two hypotheses:

> **Causality**. The forecast model can be expressed as the product of intermediate forecast steps, which reflects the causality of nature. Usually it is the integration of a numerical prediction model starting with $\mathbf{x}$ as the initial condition. If the times i are sorted, with $\mathbf{x}_0 = \mathbf{x}$ so that M_0 is the identity, then by denoting M_i the forecast step from $i-1$ to i we have $\mathbf{x}_i = M_i\mathbf{x}_{i-1}$ and by recurrence
>
> $$\mathbf{x}_i = M_iM_{i-1}\ldots M_1\mathbf{x}$$

Tangent linear hypothesis. The cost function can be made quadratic by assuming, on top of the linearization of H_i, that the M operator can be linearized, i.e.

$$\mathbf{y}_i - H_i M_{0\to i}(\mathbf{x}) \approx \mathbf{y}_i - H_i M_{0\to i}(\mathbf{x}_\mathrm{b}) - \mathbf{H}_i \mathbf{M}_{0\to i}(\mathbf{x}-\mathbf{x}_\mathrm{b})$$

where $\mathbf{M}$ is the *tangent linear (TL) model*, i.e. the differential of M. For a discussion of this hypothesis, refer to the section on the tangent linear hypothesis, in which the remarks made on H apply similarly to M. It explains that the realism of the TL hypothesis depends not only on the model, but also on the general characteristics of the assimilation system, including notably the length of the 4D-Var time interval.

The two hypotheses above simplify the general minimization problem to an unconstrained quadratic one which is numerically much easier to solve. The first term J_b of the cost function is no more complicated than in 3D-Var and it will be left out of this discussion. The evaluation of the second term J_o would seem to require n integrations of the forecast model from the analysis time to each of the observation times i, and even more for the computation of the gradient ∇J_o. We are going to show that the computations can in fact be arranged in a much more efficient way.

10.2 Theorem: minimization of the 4D-Var cost function

The evaluation of the 4D-Var observation cost function and its gradient, $J_\mathrm{o}(\mathbf{x})$ and $\nabla J_\mathrm{o}(\mathbf{x})$, requires one direct model integration from times 0 to n and one suitably modified *adjoint integration* made of transposes of the tangent linear model time-stepping operators $\mathbf{M}_i$.

Proof:

The first stage is the direct integration of the model from $\mathbf{x}$ to $\mathbf{x}_n$, computing successively at each observation time i:

1) the forecast state $\mathbf{x}_i = M_i M_{i-1} \ldots M_1 \mathbf{x}_1$,
2) the "normalized departures" $d_i = \mathbf{R}_i^{-1}(\mathbf{y}_i - H_i[\mathbf{x}_i])$ which are stored,
3) the contributions to the cost function $J_{\mathrm{o}i}(\mathbf{x}) = (\mathbf{y} - H_i[\mathbf{x}_i])^\mathrm{T} d_i$
4) And finally $J_\mathrm{o}(x) = \sum_{i=0}^{n} J_{\mathrm{o}i}(\mathbf{x})$.

To compute ∇J_o it is necessary to perform a slightly complex factorization:

$$\begin{aligned}
-\frac{1}{2}\nabla J_\mathrm{o} &= -\frac{1}{2}\sum_{i=0}^{n} \nabla J_{\mathrm{o}i} \\
&= \sum_{i=0}^{n} \mathbf{M}_1^\mathrm{T} \ldots \mathbf{M}_i^\mathrm{T} \mathbf{H}_i^\mathrm{T} d_i \\
&= \mathbf{H}_0^\mathrm{T} d_o + \mathbf{M}_1^\mathrm{T}[\mathbf{H}_1^\mathrm{T} d_1 + \mathbf{M}_2^\mathrm{T}[\mathbf{H}_2^\mathrm{T} d_2 + \ldots + \mathbf{M}_n^\mathrm{T}\mathbf{H}_n^\mathrm{T} d_n]\ldots]
\end{aligned}$$

and the last expression is easily evaluated from right to left using the following algorithm:

5) initialize the so-called *adjoint variable* $\tilde{\mathbf{x}}$ to zero at final time: $\tilde{\mathbf{x}} = 0$
6) for each time step $i-1$ the variable $\tilde{\mathbf{x}}_{i-1}$ is obtained by adding the *adjoint forcing* $\mathbf{H}_i^\mathrm{T} d_i$ to $\tilde{\mathbf{x}}_i$ and by performing the *adjoint integration* by multiplying the result by $\mathbf{M}_i^\mathrm{T}$, i.e. $\tilde{\mathbf{x}}_{i-1} = \mathbf{M}_i^\mathrm{T}(\tilde{\mathbf{x}}_i + \mathbf{H}_i^\mathrm{T} d_i)$
7) at the end of the recurrence, the value of the adjoint variable $\tilde{\mathbf{x}}_0 = -(1/2)J_\mathrm{o}(\mathbf{x})$ gives the required result.

The terminology employed in the algorithm reflects the fact that the computations look like the integration of an *adjoint model* backward in time with a time-stepping defined by the transpose time-stepping operators $\mathbf{M}_i^\mathrm{T}$ and an external forcing $\mathbf{H}_i^\mathrm{T} d_i$, which depends on the distance between the model trajectory and the observations. In this discrete presentation it is just a convenient way of evaluating an algebraic expression[25].

10.3 Properties of 4D-Var

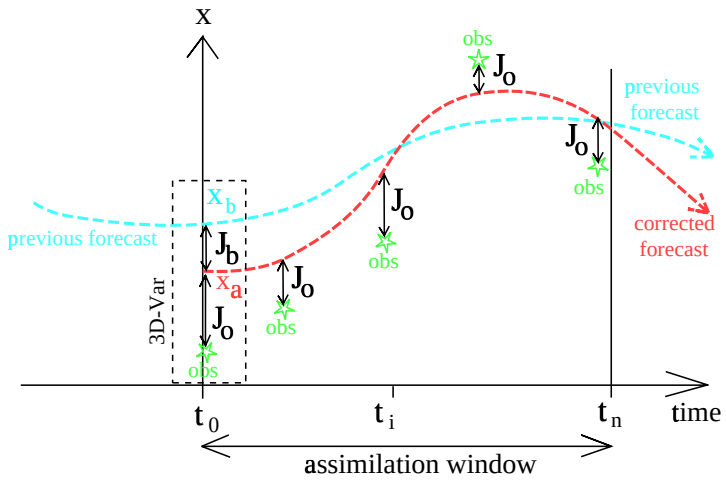

Figure 12. Example of 4D-Var intermittent assimilation in a numerical forecasting system. Every 6 hours a 4D-Var is performed to assimilate the most recent observations, using a segment of the previous forecast as background. This updates the initial model trajectory for the subsequent forecast.

When compared to a 3-D analysis algorithm in a sequential assimilation system, 4D-Var has the following characteristics:

- it works only under the assumption that the model is perfect. Problems can be expected if model error are large.
- it requires the implementation of the rather special $\mathbf{M}_i^{\mathrm{T}}$ operators, the so-called adjoint model. This can be a lot of work if the forecast model is complex.
- in a real-time system it requires the assimilation to wait for the observations over the whole 4D-Var time interval to be available before the analysis procedure can begin, whereas sequential systems can process observations shortly after they are available. This can delay[26] the availability of $\mathbf{x}_a$.
- $\mathbf{x}_a$ is used as the initial state for a forecast, then by construction of 4D-Var one is sure that the forecast will be completely consistent with the model equations and the four-dimensional distribution of observations until the end of the 4D-Var time interval n (the *cutoff time*). This makes intermittent 4D-Var a very suitable system for numerical forecasting (Fig. 12).
- 4D-Var is an optimal assimilation algorithm over its time period thanks to the following theorem. It means that it uses the observations as well as possible, even if $\mathbf{B}$ is not perfect, to provide $\mathbf{x}_a$ in a much less expensive way than the equivalent Kalman Filter. For instance, the coupling between advection and observed information in illustrated in Fig. 13 .

25. In a continuous (in time) presentation, the concept of adjoint model could be carried much further into the area of differential equations. However, this is not relevant to real models where the adjoint of the discretized model must be used, instead of the discretization of a continuous adjoint model. The only relevant case is if some continuous operators have a simple adjoint: then, with a careful discretization that preserves this property, the implementation of the discrete transpose operators can be simplified.

26. Some special implementations of 4D-Var can partly solve this problem.

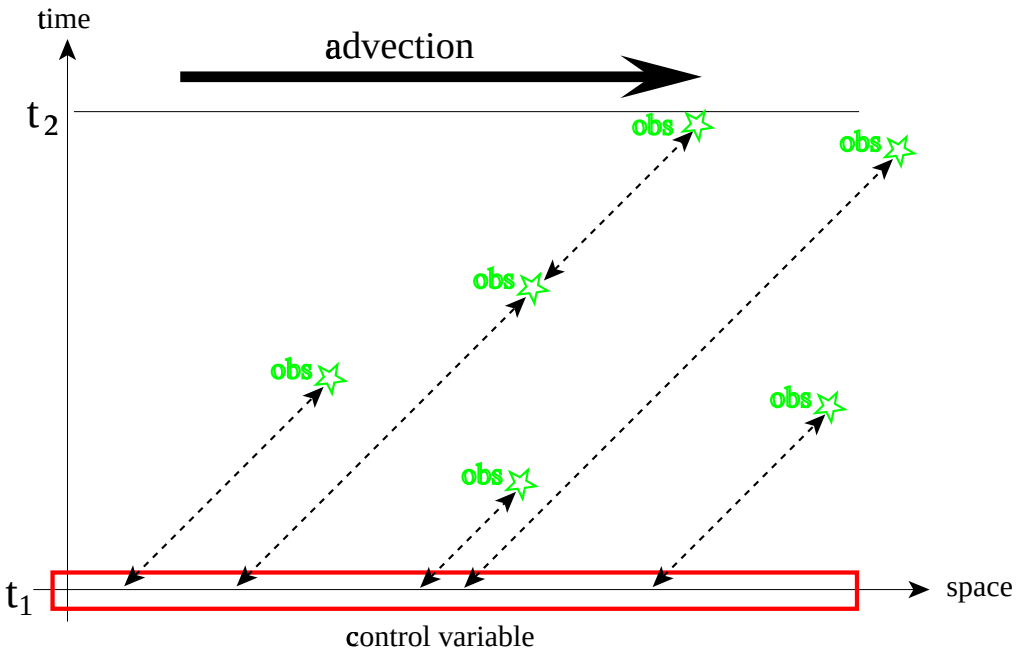

Figure 13. Example of propagation of the information by 4D-Var (or, equivalently, a Kalman filter) in a 1-D model with advection (i.e. transport) of a scalar quantity. All features observed at any point within the 4D-Var time window (t_1, t_2) will be related to the correct upstream point of the control variable by the tangent linear and adjoint model, along the characteristic lines of the flow (dashed).

10.4 Equivalence between 4D-Var and the Kalman Filter

Over a given time interval, under the assumption that the model is perfect, with the same input data (initial background and its covariances $\mathbf{B}$, distribution of observations and their covariances $\mathbf{R}_i$), the 4D-Var analysis at the end of the time interval is equal to the Kalman filter analysis at the same time.

This theorem is discussed in more details in the section about the Kalman filter algorithm, with a discussion of the pros and cons of using 4D-Var.

A special property of the 4D-Var analysis in the middle of the time interval is that it uses all the observations simultaneously, not just the ones before the analysis time. It is said that 4D-Var is a *smoothing* algorithm[27].

Ref: Talagrand and Courtier 1987, Thépaut and Courtier 1991, Rabier and Courtier 1992, Lacarra and Talagrand 1988, Errico *et al.* 1993.

11. ESTIMATING THE QUALITY OF THE ANALYSIS

It is usually an important property of an analysis algorithm that it should be able to provide an estimate of the quality of its output. If there is no observation the quality is obviously that of the background. In a sequential analysis system the knowledge of the analysis quality is useful because it helps in the specification of the background error

27. Equivalent to the *Kalman smoother* algorithm which is a generalization of the Kalman filter, but at a much smaller cost.

covariances for the next analysis, a problem called *cycling* the analysis. If the background is a forecast, then its errors are a combination of analysis and model errors, evolved in time according to the model dynamics. This is explicitly represented in the Kalman filter algorithm.

If the analysis gain $\mathbf{K}$ has been calculated, e.g. in an OI analysis, then the analysis error covariance matrix is provided by Eq. (A3)

$$\mathbf{A} = (\mathbf{I}-\mathbf{KH})\mathbf{B}(\mathbf{I}-\mathbf{KH})^{\mathrm{T}} + \mathbf{KRK}^{\mathrm{T}} \tag{A3}$$

which reduces to $\mathbf{A} = (\mathbf{I}-\mathbf{KH})\mathbf{B}$ (A4) in the unlikely case where $\mathbf{K}$ has been computed exactly.

In a variational analysis procedure, the error covariances of the analysis can be inferred from the matrix of second derivatives, or *Hessian*, of the cost function thanks to the following result:

11.1 Theorem: use of Hessian information

The Hessian of the cost function of the variational analysis is equal to twice the inverse of the analysis error covariance matrix:

$$\mathbf{A} = \left(\frac{1}{2}J''\right)^{-1}$$

Proof:

The Hessian is obtained by differentiating J twice with respect to the control variable $\mathbf{x}$:

$$\begin{aligned} J(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}_{\mathrm{b}})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_{\mathrm{b}}) + (\mathbf{y}-H[\mathbf{x}])^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}]) \\ \nabla J &= 2\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_{\mathrm{b}}) - 2\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}]) \\ J'' = \nabla\nabla J &= 2(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}) \end{aligned} \tag{A5}$$

Now we express the fact that $\nabla J(\mathbf{x}_{\mathrm{a}}) = 0$ and we insert the true model state $\mathbf{x}_{\mathrm{t}}$ into the equation:

$$\begin{aligned} 0 &= \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{a}}-\mathbf{x}_{\mathrm{b}}) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}_{\mathrm{a}}]) \\ &= \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{a}}-\mathbf{x}_{\mathrm{t}}+\mathbf{x}_{\mathrm{t}}-\mathbf{x}_{\mathrm{b}}) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}_{\mathrm{t}}]+\mathbf{H}[\mathbf{x}_{\mathrm{t}}-\mathbf{x}_{\mathrm{a}}]) \\ &= \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{a}}-\mathbf{x}_{\mathrm{t}}) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}(\mathbf{x}_{\mathrm{t}}-\mathbf{x}_{\mathrm{a}}) - \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{b}}-\mathbf{x}_{\mathrm{t}}) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}_{\mathrm{t}}]) \end{aligned}$$

Hence

$$(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})(\mathbf{x}_{\mathrm{a}}-\mathbf{x}_{\mathrm{t}}) = \mathbf{B}^{-1}(\mathbf{x}_{\mathrm{b}}-\mathbf{x}_{\mathrm{t}}) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-H[\mathbf{x}_{\mathrm{t}}])$$

When it is multiplied on the right by its transpose, and the expectation of the result is taken, the right-hand side then contains two terms that multiply

$$\overline{(\mathbf{x}_{\mathrm{b}}-\mathbf{x}_{\mathrm{t}})(\mathbf{y}-H[\mathbf{x}_{\mathrm{t}}])^{\mathrm{T}}}$$

which is zero because we assume background and observation errors are uncorrelated. The remaining terms lead to, successively:

$$
\begin{aligned}
(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\mathbf{A}(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}) &= \mathbf{B}^{-1}\mathbf{B}\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{H} \\
&= \mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} \\
\mathbf{A} &= (\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1}
\end{aligned}
$$

which proves the result.

11.2 Remarks

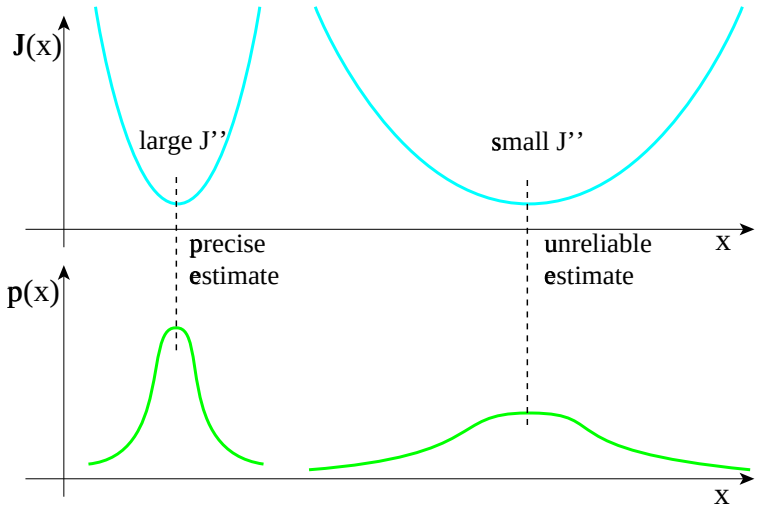

Figure 14. Illustration in a one-dimensional problem of the relationship between the Hessian and the quality of the analysis. In one dimension, the Hessian is the second derivative, or convexity, of the cost-function of the variational analysis: two examples of cost-functions are depicted in the upper panel, one with a strong convexity (on the left), the other with a weaker one (on the right). If the cost-function is consistent with the pdfs involved in the analysis problem, the Hessian is a measure of the sharpness of the pdf of the analysis (depicted in the lower panel). A sharper pdf (on the left) means that the analysis is more reliable, and that the probability of the estimated state to be the true one is higher.

A simple, geometrical illustration of the relationship between the Hessian and the quality of the analysis is provided in Fig. 14 . In a multidimensional problem, the same interpretation is valid along cross-sections of the cost-function.

If the linearization of the observation operator H can be performed exactly, the cost function J is exactly quadratic and J'' does not depend on the value of the analysis: $\mathbf{A}$ can be determined as soon as J is defined, even before the analysis is actually carried out[28]. If the linearization is not exact, $J''(\mathbf{x})$ is not constant. It may depend a lot on $\mathbf{x}$, even if J itself does not look very different from a quadratic function. For instance, if J is continuously differentiable but not strictly convex, there are points at which $J'' = 0$. If ∇J is not continuous, then there are points at which J'' is not defined at all. It means that H must be exactly linear in order to be able to calculate $\mathbf{A}$ using

28. Actually, neither $\mathbf{A}$ nor $\mathbf{K}$ depend on the values of the background or of the observations.

the Hessian. In practice J must be modified to use the tangent linear of H, which can be acceptable in a close vicinity of $\mathbf{x}_a$.

The identity $\mathbf{A}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ shows clearly how the observed data increases the inverse error covariances, also called *information matrices*.

Ref: Rabier and Courtier 1992.

12. IMPLEMENTATION TECHNIQUES

In most practical applications, numerical cost is an important issue. As shown above, there is a variety of analysis methods available. It does not imply that any of these is the best; they should be regarded as a choice of several compromises between numerical cost, statistical optimality and physical realism of the assimilation system. The sections below describe other features of the analysis algorithms which can be used to further cut down on the numerical cost, without sacrificing too much on the sophistication of the analysis method itself. They are discussed here in the framework of 3D-Var[29], but they can be applied equally well (with a few adaptations) to all related algorithms: 1D-Var, 4D-Var, PSAS or the Kalman filter.

12.1 Minimization algorithms and preconditioning

In a variational analysis system a cost function has to be minimized, usually using an iterative descent algorithm. The cost of the analysis is proportional to the number of evaluations of the cost function and its gradient[30], called the number of *simulations*. When the state itself $\mathbf{x}$ is updated, an *iteration* is performed. Each iteration may require one or more simulations, depending on the minimizing algorithm used. Hence the technical implementation of a variational analysis can be summarized as a *simulator* operator:

$$\mathbf{x} \Rightarrow J(\mathbf{x}), \nabla J(\mathbf{x})$$

How to use the simulator to minimize the cost function is a well-developed area of mathematics (called *optimization*, a part of *numerical analysis*). With the analysis methods described above, the cost function will be a scalar function of a real vector in a Euclidean space; in most applications it will be quadratic and $\mathbf{x}$ will be unconstrained. There are several ready-to-use algorithms that do the minimization, called *minimizers*. An obvious method, the *steepest descent* method, is to update $\mathbf{x}$ by adding a correction that is proportional to $-\nabla J(\mathbf{x})$. This is usually not very efficient, and more popular algorithms are the *conjugate gradient* and *quasi-Newton* methods. They are still being improved. There are more specialized algorithms for situations where J is not quadratic or $\mathbf{x}$ is bounded, e.g. *simulated annealing* or the *simplex*, although such methods can be very expensive. The *incremental* method described below can also be regarded as a particular minimizer. A detailed description of the main minimizing algorithms can be found in dedicated mathematical books. Among the important theoretical results are the optimality properties of the conjugate gradient method in the case of an exactly quadratic cost function, and its equivalence with a Lanczos method for determining eigenvectors of the Hessian matrix. Also, the quasi-Newton methods can be regarded as a preconditioning of the cost function using accumulated information about the second derivatives.

The main aspect of J that affects the performance of conventional minimizers (assuming J is quadratic or almost)

29. This reflects history. The main step in meteorological data assimilation methods was the move from OI to 3D-Var. It was a major technical challenge in terms of coding and numerical cost at the time, which required some major developments in the fields of adjoint coding, formulation of the incremental technique and design of the preconditioner.

30. Some minimization algorithms also use information about the second derivative of the cost function, which requires the coding of the *second-order adjoint* of its components.

is its *condition number*. This quantity measures the ellipticity of the iso-surfaces of J, and it describes the difficulty of minimization problem (or *ill-conditioning*) due to the gradient ∇J not pointing accurately toward the minimum (Fig. 15). In this case minimizers have trouble converging, a phenomenon called the *narrow valley* effect.

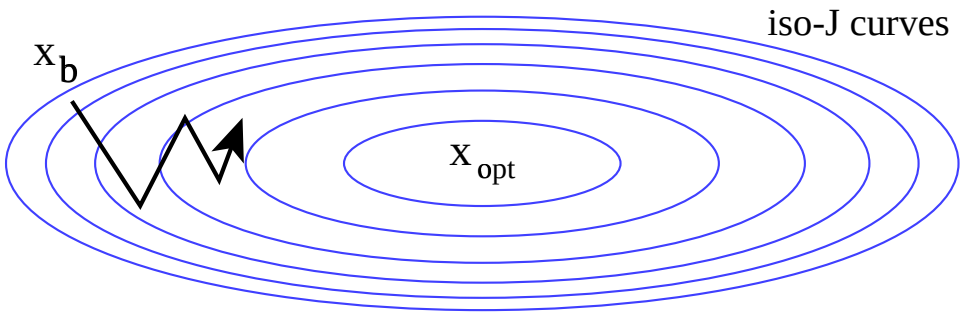

Figure 15. Illustration of the so-called narrow valley effect: in a plane of the control variable space where the convexity of the cost-function depends a lot on direction, the isolines are narrow ellipses, and in most places the gradient of the cost function is nearly orthogonal to the direction of the minimum $\mathbf{x}_{\text{opt}}$, which means that minimization algorithms will tend to waste many iterations zigzagging slowly towards the optimum.

Condition number. The condition number of J is defined to be the ratio between the largest and the smallest eigenvalue of J''. The larger the number, the more ill-conditioned the problem is.

If the condition number is equal to one, i.e. J'' is proportional to $\mathbf{I}$, the cost function is said to be *spherical* and the minimum can be found in one iteration because $-\nabla J(\mathbf{x}_{\text{b}})$ points directly toward the minimum.

In the general case, J is elliptic, but it is possible to define a change of minimization space called *preconditioning* that decreases the condition number. The idea is to present the minimizer with a problem that is not the minimization of $J(\mathbf{x})$, but another easier problem from which $\mathbf{x}_{\text{a}}$ can be obtained easily. The mapping between both problems is defined as follows using a *preconditioner* operator L:

12.2 Theorem: preconditioning of a variational analysis

If L is an invertible operator, an equivalent rewriting of the minimization problem:

$$\begin{aligned} \mathbf{x}_a &= \text{Arg min } J \\ J:\mathbf{x} &\Rightarrow J(\mathbf{x}), \nabla J(\mathbf{x}) \end{aligned} ,$$

with the initial point $\mathbf{x}_{ini} = \mathbf{x}_b$,
is the *preconditioned* problem:

$$\begin{aligned} \chi_a &= \text{Arg min } \hat{J} \\ \hat{J}:\chi &\Rightarrow \hat{J}(\chi) = J(L_\chi), \nabla \hat{J}(\chi) = L^T \nabla J(L_\chi) \end{aligned} ,$$

with the initial point $\chi_{ini} = L^{-1}\mathbf{x}_b$.
The solution is given by $\mathbf{x}_a = L\chi_a$.

The proof is left as an exercise. In 3D-Var, a simple and efficient preconditioner is the symmetric left-hand square root of $\mathbf{B}$, i.e. a matrix[31] L such that $\mathbf{B} = LL^T$. In this case one can show that

$$\hat{J}(\chi) = \chi^T\chi + J_o(L\chi)$$

i.e. the J_b term is now the canonical inner product. An ideal preconditioner would of course be provided by the symmetric square root of the Hessian matrix. Some sophisticated minimizer packages allow the user to provide his own preconditioner to the code, which can take the form of a clever specification of the inner product.

Ref: Gilbert and Lemaréchal 1989.

12.3 The incremental method

The incremental method is a relatively empirical technique designed to reduce the cost of solving a predefined variational problem, e.g. by reducing the resolution of the increments.

In the introduction it was explained how the control variable could be made smaller than the model state by requiring that the increments can only be non-zero in a subspace of the model. In this case there is no guarantee that the analysis verifies any optimality condition in the full model space. For instance, OI solves the problem separately in a set of subspaces (defined by the observation selection), but the result is not as optimal as a global least-squares analysis. With 3D- or 4D-Var is it usually not affordable to solve the variational problem at the full model resolution. However, it is expected that most of the complexity of the analysis is in the synoptic scales, because this is where most background errors are expected to be. If the increments are right at the synoptic scales, then one can expect the smaller scales to be more or less forced to be realistic features by the model dynamics. It is undesirable, though, to completely neglect the small scales in the analysis procedure because they are important in the comparison of the observations with the background state. In other words, one is looking for a low-resolution correction

31. The symmetric square root is not unique, it is defined modulo an orthogonal matrix.

to a high-resolution background. The incremental method described below has been designed for this particular problem. Mathematically, it can be thought the approximation of a large problem by a sequence of smaller problems. However, there is no proof of the convergence of the general procedure[32].

In the incremental method some high-resolution versions of the cost function, the observation operator and the model state are considered, denoted respectively $(J_\mathrm{h}, H_\mathrm{h}, \mathbf{x}_\mathrm{h})$. We are trying to minimize $J_\mathrm{h}(\mathbf{x}_\mathrm{h})$. One or several successive approximations to this problem are solved successfully. Each one is an *inner loop* that tries to *update* a high-resolution state $\mathbf{x}_{\mathrm{h},i}$ into another one $\mathbf{x}_{\mathrm{h},i+1}$ that is more optimal (in the first update, $\mathbf{x}_{\mathrm{h},i} = \mathbf{x}_\mathrm{b}$). The input information to the inner loop is given by the high-resolution departures:

$$d_{\mathrm{h},i} = \mathbf{y} - H_\mathrm{h}(\mathbf{x}_{\mathrm{h},i})$$

and by a low-resolution version $\mathbf{x}_i$ of $\mathbf{x}_{\mathrm{h},i}$ defined by a conversion operator $S_{\mathrm{h}\to l}$:

$$\mathbf{x}_i = S_{\mathrm{h}\to l}(\mathbf{x}_{\mathrm{h},i})$$

It is natural to linearize the low-resolution observation operator H in the vicinity of $\mathbf{x}_i$ which is the best currently available estimate of the analysis[33], which yields a linearized observation operator $\mathbf{H}_i$ that depends on the update index i, and defined as the differential of H in the vicinity of $\mathbf{x}_i$:

$$H(\mathbf{x}) \approx \mathbf{H}_i(\mathbf{x}-\mathbf{x}_i) + H(\mathbf{x}_i)$$

However, for consistency with the high-resolution problem, one also requires that the low-resolution is kept consistent with the high-resolution one for $\mathbf{x} = \mathbf{x}_i$, so that the linearized departures used at low resolution will be calculated as

$$\mathbf{y} - H(\mathbf{x}) \approx \mathbf{y} - [\mathbf{H}_i(\mathbf{x}-\mathbf{x}_i) + H_\mathrm{h}(x_{\mathrm{h},i})] = d_{\mathrm{h},i} - \mathbf{H}_i(\mathbf{x}-\mathbf{x}_i)$$

so that the low-resolution cost-function to minimize in the inner loop is

$$J_i(x) = (\mathbf{x}-\mathbf{x}_\mathrm{b})^\mathrm{T}\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}_\mathrm{b}) + [d_{\mathrm{h},i} - \mathbf{H}_i(\mathbf{x}-\mathbf{x}_i)]^\mathrm{T}\mathbf{R}^{-1}[d_{\mathrm{h},i} - \mathbf{H}_i(\mathbf{x}-\mathbf{x}_i)]$$

which is exactly quadratic. Its minimum is $\mathbf{x}_{i+1}$ which can in turn be used to update the high-resolution state using a (possibly nonlinear) ad hoc conversion operator $S_{l\to\mathrm{h}}$:

$$\mathbf{x}_{\mathrm{h},i+1} = \mathbf{x}_{\mathrm{h},i} + S_{l\to\mathrm{h}}(\mathbf{x}_{i+1}) - S_{l\to\mathrm{h}}(\mathbf{x}_i)$$

which ensures that the high-resolution state is not modified if the inner loop minimization does not change the state. From $\mathbf{x}_{\mathrm{h},i+1}$ the new high-resolution departures $d_{\mathrm{h},i+1}$ can be calculated and used to define the next low-resolution problem. If $d_{\mathrm{h},i+1} = d_{\mathrm{h},i}$ then the high- and low-resolution problems are fully consistent with each other and the whole algorithm has converged. However, it not guaranteed that there is a convergence at all. This is why one must be careful about the physical implications of changing the resolution. The intuitively important assumption for convergence (this can be proven in simplified systems) is that

32. It is possible to guarantee convergence for some special forms of the incremental algorithm.

33. One would rather like to use a low-resolution version of the linearized high-resolution H_h in the vicinity of $\mathbf{x}_{h,i}$ but it would be more expensive than the technique described here.

$$H_h \circ S_{l \to \mathrm{h}}(\mathbf{x}_{i+1}) - H_h \circ S_{l \to \mathrm{h}}(\mathbf{x}_i) \approx \mathbf{H}_i(\mathbf{x}_{i+1} - \mathbf{x}_i)$$

i.e. the changes in the model equivalents of the observations should be similar at high and low resolutions. If for instance they are of opposite signs, one can expect the model state at high resolution to go *away* from the observations during the procedure until it is stopped by the J_b term—a not very desirable behaviour. Whether this is a genuine problem is still an area of research. History has shown so far that 3D-Var with a simple incremental formulation and a rather low resolution of the inner loops can be much better than an OI algorithm at full resolution, for a similar numerical cost.

Ref: Courtier et al. 1994.

12.4 The adjoint technique

As shown in the explanation of the 4D-Var method, some computational savings can be achieved by a suitable ordering of the algebraic operations, in order to reduce the size and number of the matrix multiplications involved. For minimization problems in particular, when the derivative of a scalar function with respect to a large vector needs to be evaluated (e.g. J_o), it is advantageous to use the chain rule *backwards*, i.e. from the scalar function to the input vector. Algebraically this means replacing a set of matrices by their transposes, hence the name of *adjoint* technique. The definition of the adjoint depends on the scalar products[34] used:

Adjoint operator. By definition, given a linear operator A going from a space E to a space F, and scalar products $< . , . >_E$, and $< . , . >_F$ in these respective spaces, the *adjoint* of A is the linear operator A^* such that for any vectors (x, y) in the suitable spaces,

$$< Ax, y >_F \; = \; < x, A^* y >_E$$

Important remarks on the adjoints

- **Riesz theorem:** The adjoint always exists and it is unique, assuming spaces of finite dimension[35]. Hence, coding the adjoint does not raise questions about its existence, only questions of technical implementation.
- In the meteorological literature, the term *adjoint* is often improperly used to denote the adjoint of the tangent linear of a non-linear operator. One must be aware that discussions about the "existence of the adjoint" usually address the existence of the tangent linear operator (or the acceptability of using the adjoint of an improper tangent-linear in order to minimize a 4D-Var cost-function). As explained above, the adjoint itself always exist.
- In general, the adjoint depends on the definition of spaces E and F. For instance, a canonical injection (i.e. $Ax = x$ with E being a subspace of F) is not necessarily self-adjoint although A does not involve any arithmetic operation.
- In general, the adjoint depends on the choice of scalar products, even if $E = F$. For instance, a symmetric matrix may not be self-adjoint if the scalar product is not the canonical product (see below).

34. or: *inner products*

35. It is actually true for all continuous operators in Hilbert spaces, but this is outside the scope of this paper.

Theorem: adjoint and scalar product change. The operator $A: E \rightarrow F$ being identified with its matrix, and the scalar products $< . , . >_E$ and $< . , . >_F$ being identified with their symmetric positive definite matrices E and F such that e.g. $< x, x >_E = x^{\mathrm{T}} E x$, the matrix of the adjoint of A is

$$A^* = E^{-1} A^{\mathrm{T}} F$$

The proof is obvious from the definition of A^*:

$$< Ax, y >_F = x^{\mathrm{T}} A^{\mathrm{T}} F y = < x, A^* y >_E = x^{\mathrm{T}} E A^* y$$

and noting that *E* is invertible. In most practical cases (such as in the rest of this paper) the implicit scalar product used is the *canonical inner product*[36], so that the transpose is the adjoint: $A^T = A^*$. However, one must take care whenever another scalar product is used, because it has implications on the coding of the adjoint: the scalar product coefficients or their inverses must be used according to the above equation.

Adjoint of a sequence of operators. Like the transpose, the adjoint of a product of operators is the product of the adjoints in the reverse order. The scalar product matrices cancel out each other, so that if $A = A_1 A_2 \ldots A_n$ is a sequence of operators, its adjoint is

$$(A_1 A_2 \ldots A_n)^* = E^{-1} A_n^{\mathrm{T}} \ldots A_2^{\mathrm{T}} A_1^{\mathrm{T}} F$$

which shows that, even if the scalar products are not the canonical inner product, in most of the adjoint coding it can be considered that **the adjoint is the transpose.** The guidelines for practical adjoint coding are detailed in an appendix.

Ref: Errico and Vukicevic 1992.

13. DUAL FORMULATION OF 3D/4D-VAR (PSAS)

The 3D-Var formulation (A5) can be rewritten into a form called PSAS (Physical Space Assimilation System[37]) which is equivalent in the linear case only. The idea is to notice that the expression

$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{B}\mathbf{H}^{\mathrm{T}} (\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}_b)$$

can be split as the following two equalities

36. or: *inner dot product*, or: *Euclidean product.*

37. The misleading name PSAS was introduced for historical reasons and is widely used, probably because it sounds like the US slang word *pizzazz*.

$$\mathbf{w}_a = (\mathbf{HBH}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{Hx}_b)$$
$$\mathbf{x}_a - \mathbf{x}_b = \mathbf{BH}^T\mathbf{w}_a$$

where $\mathbf{w}_a$ has the same dimension as $\mathbf{y}$ and can be regarded as a kind of "increment" in observation space[38], whereas $\mathbf{BH}^T$ is a smoothing term that maps the increment from observation to model space. The aim is to solve the analysis problem in terms of $\mathbf{w}$ rather than in model space. One way is to solve for $\mathbf{w}$ the linear system

$$(\mathbf{HBH}^T + \mathbf{R})\mathbf{w} = \mathbf{y} - \mathbf{Hx}_b$$

which can be regarded as the dual of the OI algorithm. Another way is to find a cost function that $\mathbf{w}_a$ minimizes, for instance

$$F(\mathbf{w}) = \mathbf{w}^T(\mathbf{HBH}^T + \mathbf{R})\mathbf{w} - 2\mathbf{w}^T(\mathbf{y} - \mathbf{Hx}_b)$$

which is a quadratic cost function. The practical PSAS analysis algorithm is as follows:

1) Calculate the background departures $\mathbf{y} - \mathbf{Hx}_b$

2) Minimize $F(\mathbf{w})$. Some possible preconditionings are given by the symmetric square root of $\mathbf{R}$ or $\mathbf{HBH}^T$.

3) Multiply the minimum $\mathbf{w}_a$ by $\mathbf{BH}^T$ to obtain analysis increments.

4) Add the increments to the background $\mathbf{x}_b$.

A 4-D generalization of PSAS is obtained by a suitable redefinition of the space $\mathbf{w}$ to be a concatenation $(\mathbf{w}_1\ldots, \ldots, \mathbf{w}_i, \ldots)$ of all the values $\mathbf{w}_i$ at all observation time steps i. Then $\mathbf{H}$ must be replaced by an operator that uses the tangent linear model $\mathbf{M}_i$ to map the initial model state to the observation space at each time step i, i.e. $(\mathbf{H}_1\mathbf{M}_1, \ldots, \mathbf{H}_i\mathbf{M}_i, \ldots)$. The factorization of the cost function evaluation using the adjoint method is applied to the computation of the term $\mathbf{HBH}^T\mathbf{w}$, so that the evaluation of the 4D-PSAS cost function $F(\mathbf{w})$ is as follows:

1) Calculate the departures $\mathbf{y}_i - \mathbf{H}_i\mathbf{M}_i(\mathbf{x}_b)$ for each time step, (this needs only be done once)

2) Integrate the adjoint model from final to initial time, starting with a null model state, adding the forcing $\mathbf{H}_i^T\mathbf{w}_i$ at each observation timestep,

3) Multiply the resulting adjoint variable at initial time by $\mathbf{B}$, which yields $\mathbf{BHw}$,

4) Integrate the tangent-linear model, starting with $\mathbf{BHw}$ as model state, storing the state times $\mathbf{H}_i$ at each observation time step. The collection of the stored values is $\mathbf{HBH}^T\mathbf{w}$.

5) Add $\mathbf{w}^T\mathbf{Rw}$ and $\mathbf{w}^T(\mathbf{y} - \mathbf{Hx}_b)$ (both obtained by sums of already computed quantities) to obtain $F(\mathbf{w})$.

More comments on the 4D-PSAS algorithm are provided in Courtier (1997). The PSAS algorithm is equivalent to the *representer* method (Bennett and Thornburn 1992).

As of today it is still unclear whether PSAS is superior or not to the conventional variational formulations, 3D and 4D-Var. Here are some pros and cons:

- PSAS is only equivalent to 3D/4D-Var if $\mathbf{H}$ is linear, which means that it cannot be extended to weakly non-linear observation operators.

38. Note, though, that it does not have the right physical dimensions. The actual increment in observation space is $\mathbf{HBH}^T\mathbf{w}_a$, and a precise physical interpretation of $\mathbf{w}_a$ is difficult.

- However, most implementations of 3D/4D-Var are incremental, which means that they do rely on a linearization of $\mathbf{H}$ anyway: they include non-linearity through incremental updates, which can be used identically in an incremental version of PSAS.
- It is awkward to include a J_c term in PSAS for constraints expressed in model space.
- Background error models can be implemented directly in PSAS as the $\mathbf{B}$ operator. In 3D/4D-Var they need to be inverted (unless they are factorized and used as preconditioner).
- The size of the PSAS cost function is determined by the number of observations p instead of the dimension of the model space n. If $p \ll n$ then the PSAS minimization is done in a smaller space than 3D/4D-Var. In a 4D-Var context, p increases with the length of the minimization period whereas n is fixed, so that this apparent advantage of PSAS may disappear.
- The conditioning of a PSAS cost function preconditioned by the square root of $\mathbf{R}$ is identical to that of 3D/4D-Var preconditioned by the square root of $\mathbf{B}$. However the comparison may be altered if more sophisticated preconditionings are used, or if one square root or the other is easier to specify.
- Both 3D/4D-Var and PSAS can be generalized to include model errors. In 3D/4D-Var this means increasing the size of the control variable, which is not the case in PSAS, although the final cost of both algorithms looks the same.

Ref: Bennett and Thornburn 1992, Courtier 1997.

14. THE EXTENDED KALMAN FILTER (EKF)

The Kalman Filter and its extended version (EKF) are developments of the least-squares analysis method in the framework of a sequential data assimilation, in which each background is provided by a forecast that starts from the previous analysis. It is adapted to the real-time assimilation[39] of observations distributed in time into a forecast model M.

The analysis equations of the linear Kalman Filter are exactly the ones already described in the least-squares analysis theorem. The notation is the same, except that the background (i.e. forecast) and analysis error covariance matrices are now respectively denoted $\mathbf{P}_f$ and $\mathbf{P}_a$. The background state $\mathbf{x}_b$ is a forecast denoted $\mathbf{x}_f$.

14.1 Notation and hypotheses

They are the same as in the least-squares analysis theorem, except that:

- the background and analysis error covariance matrices $\mathbf{B}$ and $\mathbf{A}$ are respectively replaced by $\mathbf{P}_f$ and $\mathbf{P}_a$ to denote the fact that the background is now a forecast.
- The time index i of each quantity is denoted by the suffix i. The model forecast operator from dates i to $i+1$ is denoted by $M_{i \to i+1}$
- **forecast errors:** the deviation of the forecast prediction from the true evolution, $M_{i \to i+1}[\mathbf{x}_t(i)] - \mathbf{x}_t(i+1)$, is called the *model error*[40] and we assume that it is **not biased**[41] and that the **model error covariance matrix** $\mathbf{Q}(i)$ is known.

39. The word *filter* characterizes an assimilation techniques that uses only observation from the past to perform each analysis. An algorithm that uses observations from both past and future is called a *smoother*. 4D-Var can be regarded as a smoother. Observation smoothing can be useful for non-real time data assimilation, e.g. reanalysis, although the idea has not been used much yet. The Kalman filter has a smoother version called *Kalman smoother*.

40. Or *modelling error*.

41. This is equivalent to assuming that the background errors are unbiased, so it is not really a new hypothesis.

- **uncorrelated analysis and model errors:** the analysis errors $\mathbf{x}_a(i) - \mathbf{x}_t(i)$ and model errors of the subsequent forecast $M_{i \to i+1}[\mathbf{x}_t(i) - \mathbf{x}_t(i+1)]$ are assumed to be mutually uncorrelated.
- **linearized forecast operator:** the variations of the model prediction in the vicinity of the forecast state are assumed to be a linear function of the initial state: for any $\mathbf{x}(i)$ close enough to $\mathbf{x}_a(i)$, $M_{i \to i+1}[\mathbf{x}(i)] - M_{i \to i+1}[\mathbf{x}_a(i)] = \mathbf{M}_{i \to i+1}[\mathbf{x}(i) - \mathbf{x}_a(i+1)]$, where $\mathbf{M}$ is a linear operator.

14.2 Theorem: the KF algorithm

Under the specified hypotheses the optimal way (in the least squares sense) to assimilate sequentially the observations is given by the *Kalman filter algorithm* defined below by recurrence over the observation times i:

State forecast $$\mathbf{x}_f(i+1) = M_{i \to i+1}\mathbf{x}_a(i) \quad \text{(KF1)}$$

Error covariance forecast $$\mathbf{P}_f(i+1) = \mathbf{M}_{i \to i+1}\mathbf{P}_a\mathbf{M}^T_{i \to i+1} + \mathbf{Q}(i) \quad \text{(KF2)}$$

Kalman gain computation $$\mathbf{K}(i) = \mathbf{P}_f(i)\mathbf{H}^T(i)[\mathbf{H}(i)\mathbf{P}_f(i)\mathbf{H}^T(i) + R(i)]^{-1} \quad \text{(KF3)}$$

State analysis $$\mathbf{x}_a(i) = \mathbf{x}_f(i) + \mathbf{K}(i)[\mathbf{y}(i) - H(i)\mathbf{x}_f(i)] \quad \text{(KF4)}$$

Error covariance of analysis $$\mathbf{P}_a(i) = [\mathbf{I} - \mathbf{K}(i)\mathbf{H}(i)]\mathbf{P}(i) \quad \text{(KF5)}$$

and the analyses are the sequences of $\mathbf{x}_a(i)$.

Proof:

The forecast equation (KF1) just translates the fact that we use the model M to evolve the model state, starting from the previous optimal analysis $\mathbf{x}_a(i)$. The equation (KF2) is obtained by first subtracting $\mathbf{x}_t(i+1)$ from (KF1) and using the linearity of the forecast operator:

$$\mathbf{x}_f(i+1) - \mathbf{x}_t(i+1) = \mathbf{M}_{i \to i+1}[\mathbf{x}_a(i) - \mathbf{x}_t(i)] + [M_{i \to i+1}\mathbf{x}_t(i) - \mathbf{x}_t(i+1)]$$

Multiplying it on the right by its transpose and taking the expectation of the result yields, by definition, $\mathbf{P}_f(i+1)$ on the left-hand side, and on the right-hand side four terms. Two of these are $\mathbf{M}_{i \to i+1}\mathbf{P}_a(i)\mathbf{M}^T_{i \to i+1}$ and $\mathbf{Q}(i)$ by definition. The remaining two terms are cross-correlations between the analysis error $\mathbf{x}_a(i) - \mathbf{x}_t(i)$ and the model error for $M_{i \to i+1}$, which are assumed to be zero. This means that $\mathbf{P}_f(i+1)$ provided by (KF2) is the background error covariance matrix for the analysis at time $i+1$.

The equations (KF3), (KF4) and (KF5) are simply the least-square analysis equations (A2), (A1) and (A4) that were proven above, using $\mathbf{P}_f(i)$ as background errors, and assuming that $\mathbf{K}$ is computed optimally.

14.3 Theorem: KF/4D-Var equivalence

Over the same time interval $[0, N]$ assuming that $\mathbf{Q} = 0$ (i.e. the model is perfect), and that both algorithms use the same data (notably, $\mathbf{P}_f(0)$ is the initial background error covariance matrix), then there is equality between

1) the final analysis $\mathbf{x}_a(N)$ produced by the above Kalman filter algorithm, and

2) the final value of the optimal trajectory estimated by 4D-Var, i.e. $M_{0 \to N}\mathbf{x}_a(0)$.

This theorem means that the KF verifies the four-dimensional least-squares optimality theory expressed by the 4D-Var cost function, although it is defined by a sequence of 3-D analyses, whereas 4D-Var solves the 4-D problem globally.

14.4 The Extended Kalman Filter (EKF)

The Kalman filter algorithm can be generalized to non-linear H and M operators, although it means that neither the optimality of the analysis nor the equivalence with 4D-Var hold in that case. If H is non-linear, $\mathbf{H}$ can be defined as its tangent linear in the vicinity of $\mathbf{x}_b$, as discussed in a previous section. Similarly, if M is non-linear, which is the case of most meteorological and oceanographical models, $\mathbf{M}$ can be defined as the *tangent linear forecast model* in the vicinity of $\mathbf{x}_a$, i.e. we assume that for any likely initial state $\mathrm{x}(i)$ (notably $\mathbf{x}_t(i)$),

$$M_{i \to i+1}[\mathbf{x}(i)] - M_{i \to i+1}[\mathbf{x}_a(i)] \approx \mathbf{M}_{i \to i+1}[\mathbf{x}(i) - \mathbf{x}_a(i)]$$

and the realism of this hypothesis must be appreciated using physical arguments, as already discussed about the observation operator and 4D-Var. If H and/or M are non-linear, the algorithm written above is called the *Extended Kalman Filter*. Note that the linearization of M interacts with the model errors in a possibly complicated way, as can be seen from the proof of Eq. (KF2) above. If non-linearities are important, it may be necessary to include empirical correction terms in the equation, or to use a more general stochastic prediction method such as an *ensemble prediction* (or *Monte Carlo*) method, which yields an algorithm known as the *Ensemble Kalman Filter.*

14.5 Comments on the KF algorithm

The input to the algorithms is: the definition of the model and the observation operator, the initial condition for $(\mathbf{x}, \mathbf{P})$ when the recurrence of the filter is started[42], the sequence of observations $\mathbf{y}$, and the sequence of model and observation error covariance matrices $(\mathbf{Q}, \mathbf{R})$. The output is the sequence of estimates $(\mathbf{x}_a, \mathbf{P}_a)$ of the model state and its error covariance matrix. The organization of the KF assimilation looks like a coupled stream of estimations of model states and error covariances (Fig. 16).

42. Note that it is not well known whether, after a long time, the analysis ceases or not to depend significantly on the way the KF is initialized.

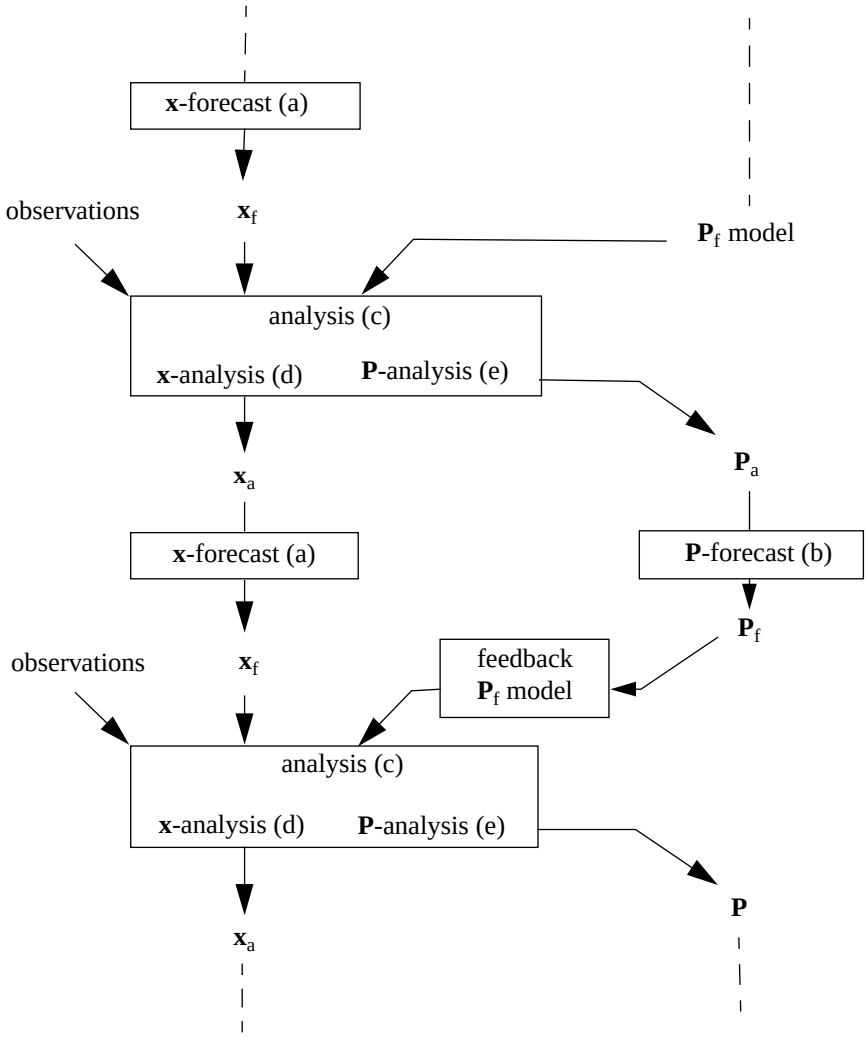

Figure 16. The organization of computations in a KF or EKF assimilation.

The variational form of the least-squares analysis can be used in the analysis step of the Kalman filter, instead of the explicit equations written above.

The numerical cost of the KF or EKF is that of the analysis itself, plus the estimation of the analysis error covariances, discussed in a specific section, plus the (KF2) covariance forecast equation which requires n forecasts of the tangent linear model (n being the dimension of the model state) to build the operator $\mathbf{M}$. The storage cost itself is significant, since each $\mathbf{P}$ matrix is $n \times n$ (only a half can be stored since they are symmetric) and in (KF5) the $\mathbf{KH}$ matrix must be evaluated and stored too (unless the variational form is used, in which case n evaluations of the gradient of the cost function must be performed to build the Hessian which must then be inverted). It means that the cost of the KF is much larger than 4D-Var, even with small models. The algorithm should rather be regarded as a reference in the design of more approximate assimilation algorithms which are being developed nowadays. It is still not clear what is the best way to approximate the KF, and the answer will probably be application-dependent.

There are many similarities between 4D-VAR and the EKF and it is important to understand the fundamental differences between them:

- 4D-VAR can be run for assimilation in a realistic NWP framework because it is computationally much cheaper than the KF or EKF.
- 4D-VAR is more optimal than the (linear or extended) KF inside the time interval for optimization because it uses all the observations at once, i.e. it is not sequential, it is a smoother.
- unlike the EKF, 4D-VAR relies on the hypothesis that the model is perfect (i.e. $\mathbf{Q} = 0$).
- 4D-VAR can only be run for a finite time interval, especially if the dynamical model is non-linear, whereas the EKF can in principle be run forever.
- 4D-VAR itself does not provide an estimate of $\mathbf{P}_f$, a specific procedure to estimate the quality of the analysis must be applied, which costs as much as running the equivalent EKF.

Ref: Ghil 1989, Lacarra and Talagrand 1988, Errico *et al.* 1993.

15. CONCLUSION

This presentation of analysis algorithms has been centred on the algebra of the least-squares analysis method. However one shall not forget the importance of other issues like observation screening and physical consistency of the assimilation, including bias correction, which can be of great importance for the quality of the assimilation system taken as a whole.

The recent trend in data assimilation is to combine the advantages of 4D-Var and the Kalman filter techniques. In a real-time assimilation system, 4D-Var over a short time interval is a very efficient analysis method. A Hessian estimation method can provide a good estimate of the analysis error covariance matrix. A simplified version of the extended Kalman filter forecast step is then used (SKF) to estimate the forecast error covariances at the time of the next analysis, which must then be combined with an empirical, more static model of the background error covariances. It is hoped that a good compromise between these algorithms can be achieved. There can be some constructive interactions with the problems of ensemble prediction, and specific studies of analysis quality like sensitivity studies and observation targeting. These new methods provide many by-products which still remain to be used as diagnostic tools for improving the assimilation and forecast system.

APPENDIX A A PRIMER ON LINEAR MATRIX ALGEBRA

Note: this is a simplified presentation for finite-dimensional real vector spaces. For more general results and rigorous mathematical definitions, refer to mathematical textbooks.

Matrix. A matrix $\mathbf{A}$ of dimension $n \times p$ is a two-dimensional array of real coefficients $(a_{ij})_{i = 1 \ldots n,\, j = 1 \ldots p}$ where i is the line index, j is the column index. A matrix is usually represented as a table:

$$\mathbf{A} = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1p} \\ a_{21} & a_{22} & & a_{2p} \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ a_{n1} & a_{n2} & \ldots & a_{np} \end{bmatrix}$$

A matrix for which $n = p$ is called a square matrix.

Diagonal. The diagonal of a square $n \times n$ matrix $\mathbf{A}$ is the set of n coefficients $(a_{ii})_{i = 1 \ldots n}$. A matrix is called

diagonal if all its non-diagonal coefficients are zero.

Transpose. The transpose of a $n \times p$ matrix $\mathbf{A}$ is a $p \times n$ matrix denoted $\mathbf{A}^{\mathrm{T}}$ with the coefficients defined by $(a_{ij})^{\mathrm{T}} = (a_{ji})$ i.e. the coefficients a_{ij} and a_{ji} are swapped, which looks like a symmetry with respect to the diagonal:

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & & a_{n2} \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ a_{1p} & a_{2p} & \cdots & a_{np} \end{bmatrix}$$

Symmetry. A square matrix is symmetric if it is equal to its transpose, i.e. $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$. This is equivalent to having $a_{ij} = a_{ji}$ for any i and j. A property of diagonal matrices is that they are symmetric.

Scalar multiplication. A $n \times p$ matrix $\mathbf{A}$ times a real scalar λ is defined as the $n \times p$ matrix $\lambda\mathbf{A}$ with coefficients (λa_{ij}).

Matrix sum. The sum of two $n \times p$ matrices $\mathbf{A}$ and $\mathbf{B}$ is defined as the $n \times p$ matrix $\mathbf{A} + \mathbf{B}$ with coefficients $(a_{ij} + b_{ij})$. It is easy to see that the sum and scalar multiplication define a vector space structure on the set of $n \times p$ matrices (the sum is associative and its neutral element is the zero matrix, with all coefficients set to zero).

Matrix product. The product between an $n \times p$ matrix $\mathbf{A}$ and a $p \times q$ matrix $\mathbf{B}$ is defined as the $n \times q$ matrix $\mathbf{C} = \mathbf{AB}$ with coefficients (c_{ij}) given by

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b$$

The product is not defined if the number of columns in $\mathbf{A}$ is not the same as the number of lines in $\mathbf{B}$. The product is not commutative in general. The neutral element of the product is the identity matrix $\mathbf{I}$ defined as the diagonal matrix with values 1 on the diagonal, and the suitable dimension. If $q = 1$ the product can be generalized to matrix times vector $\mathbf{x}$ by identifying the right-hand term of the product with the column (x_i) of vector coordinates in a suitable basis; then the multiplication (on the left) of a vector $\mathbf{x}$ by a matrix $\mathbf{A}$ can be identified to a linear application from $\mathbf{x}$ to $\mathbf{Ax}$. Likewise, 1×1 matrices can be identified with scalars.

Matrix inverse. A square $n \times n$ matrix $\mathbf{A}$ is called *invertible* if there exist an $n \times n$ matrix denoted $\mathbf{A}^{-1}$ and called inverse of $\mathbf{A}$, such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$

Trace. The trace of a square $n \times n$ matrix $\mathbf{A}$ is defined as the scalar $\mathrm{Tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$ which is the sum of the diagonal coefficients.

Useful properties.

(**A**, **B**, **C** are assumed to be such that the operations below have a meaning)

The transposition is linear: $(\mathbf{A} + \lambda\mathbf{B})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} + \lambda\mathbf{B}^{\mathrm{T}}$

Transpose of a product: $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$

Inverse of a product: $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

Inverse of a transpose: $(\mathbf{A}^{\mathrm{T}})^{-1} = (\mathbf{A}^{-1})^{\mathrm{T}}$

Associativity of the product: $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

Diagonal matrices: their products and inverses are diagonal, with coefficients given respectively by the products and inverses of the diagonals of the operands.

Symmetric matrices: the symmetry is conserved by scalar multiplication, sum and inversion, but not by the product (in general).

The trace is linear: $\mathrm{Tr}(\mathbf{A} + \lambda\mathbf{B}) = \mathrm{Tr}(\mathbf{A}) + \lambda\mathrm{Tr}(\mathbf{B})$

Trace of a transpose: $\mathrm{Tr}(\mathbf{A}^{\mathrm{T}}) = \mathrm{Tr}(\mathbf{A})$

Trace of a product: $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA}) = \sum_{i,j} a_{ij} b_{ij}$

Trace and basis change: $\mathrm{Tr}(\mathbf{B}^{-1}\mathbf{AB}) = \mathrm{Tr}(\mathbf{A})$, i.e. the trace is an intrinsic property of the linear application represented by $\mathbf{A}$.

Positive definite matrices. A symmetric matrix $\mathbf{A}$ is defined to be positive definite if, for any vector $\mathbf{x}$, the scalar $\mathbf{x}^{\mathrm{T}}\mathbf{Ax} > 0$ unless $\mathbf{x} = 0$. Positive definite matrices have real positive eigenvalues, and their positive definiteness is conserved through inversion.

APPENDIX B PRACTICAL ADJOINT CODING

As explained previously, coding the adjoint is mostly a problem of coding a transpose. Assuming a linear operator is available as a piece of code, called *direct code*, there are two approaches to implement the code for the adjoint operator. One is to take the operator as a whole, store its matrix (e.g. by taking the image of each canonical basis vector; the matrix of a tangent linear operator is called the *Jacobian* matrix and its coefficients are partial derivatives of the output with respect to the input) and code the multiplication by its transpose, which is only feasible if the matrix can be evaluated and stored at a reasonable price.

The other, more common approach, is to use the rule above for taking the adjoint of a sequence of operators, and to apply it to each elementary step of the direct code, called "model" here to fix ideas. Most of the time there is a piece of adjoint to code for each (or almost) active instruction of the direct code, considered as elementary linear operators, each in its little subspace. The concept of 'subspace' of a piece of code is justified by the fact that most components of the state are not modified by it, so that the corresponding operator is a block-diagonal matrix with just a little block spanning the variables that are actually used on input and modified on output:

$$\mathbf{A}_i = \begin{pmatrix} 1 & & & & \\ & 1 & & 0 & \\ & & \ddots & & \\ & & & \ddots & \\ & 0 & & \mathbf{A}_i & \\ & & & & \ddots \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ & 1 & & 0 & \\ & & \ddots & & \\ & & & \ddots & \\ & 0 & & \mathbf{A}_i^{\mathrm{T}} & \\ & & & & \ddots \end{pmatrix}$$

From a coding point of view, it is only necessary to code the action performed by $\mathbf{A}_i^{\mathrm{T}}$, the other variables are kept unchanged anyway. This allows one to work locally, by following a few simple rules:

- the adjoint of a sequence of operations is the reverse sequence of the transposes of each operation.

- the scalar products need to be considered only at the beginning and at the end of the code that is being adjointed (unless one wants to use some special properties of pieces of code with respect to particular products, like the unitary character of Fourier transforms with respect to the $\mathcal{L}^2$ norm).
- the input to a piece of code (e.g. a subroutine) becomes the output of the corresponding adjoint code, and vice versa. Care must be taken when specifying the interfaces between subroutines, so what is input and what is output at each stage must be clear. It means that the adjoint coding is much easier if good programming principles have been respected in the direct code to start with, such as modularity, consistent variable naming and interface control.
- it is recommended to use the same variable names for matching direct (i.e. tangent linear) and adjoint model states, in order to be able to reuse the direct code for array dimensioning and self-adjoint operations.
- the actual coding of the adjoint is performed at the smallest possible level of active subsets of code (one active instruction, or a small number of instructions that clearly depict an explicit linear operator) that must each be a linear operator with known coefficients. Its adjoint is the transpose operator, taken in the relevant space, which implies the following items.
- Each modified variable is a part of the input space unless this subset of code is *the first time it is used* in the whole direct code, i.e. it is being "defined" at this stage.
- Each input variable is a part of the output space unless this subset of code is *the last time it is used* in the whole direct code, i.e. it is being "undefined" at this stage.
- The adjoint of a variable "undefinition", i.e. the end of its scope, is its setting to zero.
- For code robustness, it is advised to consider that no variable is being undefined anywhere except at the end of code units like subroutines where they must all be pre-initialized to zero, so that each adjoint operation will be written as the addition of something to a variable.

The last items deserve some illustration. When a new variable starts to be used at some point in the code, (e.g. an array is allocated, or a variable is initialized for the first time) we go from a space e.g. (a) to a bigger space, e.g. (a, b). Hence in the adjoint we go from (a, b) to (a), which is a *projection* operator, and b is "undefined" in the adjoint code, although no matching instruction exists in a language like Fortran, so that no specific statement is needed in the adjoint. The undefinition is usually performed when returning from an adjoint subroutine. If b is used later in the adjoint code, it must have been re-initialized.

When a new variable stops being used, we go from space (a, b) to (a), and this is usually implicit in the direct code after the last instruction that uses b. One can consider that the definition of a local variable is lost when returning from a subroutine. This inconspicuous operation in the direct code is mathematically known as a *canonical injection.* Its matrix is obtained from the direct code matrix, which is a *projection*:

$$(a) = (1 \;\; 0)\begin{pmatrix} a \\ b \end{pmatrix}$$

(So that the transpose operator[43] reads, using the same variable letters (although they do not necessarily have the same values as in the direct operation):

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}(a)$$

43. Sometimes called the *adjoint* of identity.

or, in Fortran:

```
b=0.
```

If this instruction is forgotten it will result in a badly initialized b variable, with possibly erroneous results if the same variable name is used for other computations before.

Hence the adjoint of even a simple assignment a=b depends on the scope of the variables. If the input space is (b) and the output space is (a), the algebraic direct operation is

$$(a) = (1)(b)$$

so that the adjoint is trivially

$$(b) = (1)(a)$$

and the adjoint code is b=a. If however b may be used later in the direct code, it is not being undefined, the output space is (a, b) and the algebraic direct operation a=b is now

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} (b)$$

The adjoint is

$$(b) = (11) \begin{pmatrix} a \\ b \end{pmatrix}$$

(and the adjoint code is b=b+a, which is quite different, because b is now both in the output and in the output of the direct code[44]. If b is used later in the direct code, it will already contain something which will be used when doing b=b+a in the adjoint code. Physically speaking, it means that the sensitivity of the output to b (which is what the adjoint variable b contains) is the sum of the sensitivities to b in all operations that read the value of b in the direct code.

If one codes b=b+a although b is not used later in the code, b is still correctly initialized in the adjoint because the adjoint of its eventual undefinition is b=0 which will be placed before. It can be difficult to remember in a large code where each variable is used for the last time. Variable undefinition is usually easy to spot because it is always at the end of program sections (subroutines) or at variable de-allocation. If the interface between program sections is clearly documented, this makes it easy to pre-initialize the adjoint variables to zero at the right place. Hence the best adjoint programming rule is to **always** assume that a variable is being used later, and to set all adjoint code variables to zero when they are defined.

For instance, the adjoint of a typical line of code like a=s*b+t*c (the * is the multiplication, s and t are constants) is

```
a=0      ! when a is first defined in the adjoint code
b=0    ! when b is first defined in the adjoint code
c=0    ! when c is first defined in the adjoint code
```

44. Whether a is part of the input space in the direct code is not important, because it is being overwritten. In the adjoint, putting explicitly a in both input and output spaces would simply result in the additional useless line of adjoint code: a=a. One should worry more about the scope of input variables than about output variables when examining the direct code.

```
. . . . . .
b=b+s*a
c=c+t*a
```

If any of the `a,b,c` variables are defined as input arguments to a subroutine in the adjoint, then of course their initial value is defined outside and they should retain their input value.

However, there is no problem of undefinition of `a` in a statement like `a=s*a+t*c` which has the adjoint

```
c=0    ! when c is first defined in the adjoint code
.....
a=s*a       ! not a=a+s*a !
c=c+t*a
```

because `a` is both in the input and output spaces. Note that a conditional like this in the direct code:

```
if (a>0) then
  a=2*a
endif
```

defines a non-linear function of `a` which is not licit. The problem is in the linearization, not in taking the adjoint.

Most problems with writing adjoint codes are with in the handling of the trajectory (the linearization coefficients that appear when taking the differential of a non-linear operator), because the adjoint requires these values in an order that is the reverse of their computation order. They need to be stored, or recomputed on the fly, which is usually a matter of compromising between storage space (or disk I/O) and CPU time, to assess on a case-to-case basis.

APPENDIX C EXERCISES

The number of stars indicate roughly the degree of difficulty.

(i) Prove equation (A4) giving **A** if **K** is optimal.

(ii) Prove directly the equations given in the section on the scalar case.

(iii) Prove the theorem on preconditioning, including the case where the square root of **B** is used. Does the condition number depend on the choice of square root matrix?

(iv) Compare the BLUE equations with the linear regression equations between the model and observation values.

(v) Write and comment on the BLUE analysis in a one-dimensional model, with one and then with 2 observations.

(vi) rewrite the KF equations in the scalar case and examine its convergence in time if the model is the identity and if **R** and **Q** are constant.

(vii) Calculate the product of a vector with the Hessian using the simulator operator only.

(viii) The SWM (Sherley–Woodbury–Morrisson) approximation of a positive definite matrix is $I + \sum_i (\lambda_i - 1) v_i v_i^{\mathrm{T}}$ (λ_i are positive scalars, v_i are vectors). Prove that it is positive definite, and derive its inverse and a symmetric square root.

(ix) * Calculate the normalization factors to define properly the Gaussian pdfs for the background and the analysis states.

(x) Write the algorithm to implement a Cressman analysis. What happens if the observing network is very dense?

(xi) a primitive analysis technique is to fit a set of polynomials to the observations. Derive the algorithm in a one-dimensional framework.

(xii) * Generalize the polynomial fit technique to give different weights to different observations.

(xiii) Prove that $|\,\mathrm{cov}(i,j)\,| \le \sqrt{\mathrm{var}(i)\mathrm{var}(j)}$. Is it a sufficient condition for the covariance matrix to be positive definite?

(xiv) Prove that a covariance matrix can be factorized in the form $\mathbf{B} = L^{\mathrm{T}}L$ and describe some numerical methods to do it.

(xv) * Give examples in which the adjoint is not the inverse, and examples in which it is.

(xvi) * Derive in the scalar case what is the analysis error if the weight is calculated using an assumed σ_b that is not the genuine background standard error.

(xvii) * Prove that the background error covariance matrix can be factorized as $\mathbf{B} = SCS$ where S is a diagonal matrix and C is the correlation matrix. What is the physical meaning of S?

(xviii) * Rewrite the 4D-Var algorithm using the inverse of the model (assuming it exists), putting the analysis time at the end of the time interval.

(xix) * (physics regularization) In the scalar case, considering the observation operator $H(\mathbf{x}) = \max(0, \mathbf{x})$, design a continuously differentiable observation operator $\hat{H}$ with a tunable "regularization" parameter so that $|H - \hat{H}|$ can be as small as required and $H = \hat{H}$ outside a small interval around zero.

(xx) ** Design a scalar example using the previous observation operator, in which the cost-function has one or two minima, depending on the value of the regularization parameter.

(xxi) * Prove that the scalar KF, with the model equal to the identity and constant error statistics, is equivalent to a running average that is defined, in the limit of a continuous time variable, by an exponential weighting function. How does the e-folding time depend on the error statistics?

(xxii) * (adaptive filter) Rewrite the KF equation as an adaptive statistical adaptation scheme: $y = a + bx$, where the model state is the two scalars (a, b) and y is the scalar observation, x is an externally defined function of time. The forecast model is assumed to be the identity.

(xxiii) ** Generalize the Cressman algorithm in order to retain some background information at the analysis points, as in the least-squares analysis.

(xxiv) ** (retrieval and super-obing) Modify the BLUE equation for when the observations are replaced by a linear combination of them through a retrieval algorithm N, i.e. $\hat{y} = Nx$.

(xxv) ** Precondition the PSAS cost function with the symmetric square root of $\mathbf{R}$ and prove that the condition number is then the same as 3D-Var preconditioned by the symmetric square root of $\mathbf{B}^{-1}$.

(xxvi) *** (control variable remapping) In a continuous one-dimensional model, derive the adjoint of the "remapping" operator $[x = f(i)] \Rightarrow [x = f(g(i))]$ where i is the space coordinate and g is an invertible, continuously differentiable function. Does this make sense in a discrete model?

(xxvii) *** Derive the 4D-Var equations by expressing the minimization problem constrained by the model equations with its Lagrangian, and comment on the physical meaning of the Lagrange multiplier at the analysis point.

*(xxviii)*** (flow-dependency in 4D-Var) Derive the Hessian of a 4D-Var in which there is one single observation at the end of the analysis interval. How does the analysis increment compare with the singular vectors of the model? (see the training course on predictability)

(xxix) *** The NMC method assumes that the covariances of forecast error differences (differences between two forecasts starting from 2 consecutive analyses and valid at the same time) are similar to the forecast error covariances. Formulate this using the KF notation and discuss the validity of the assumption.

(xxx) *** (lagged innovation covariances) Assuming that the observing network is always the same in the KF, prove that if the analysis weight is optimal, then the innovation departures are not correlated in time.

(xxxi) *** (fixed-lag Kalman smoother) Derive the equations for the 1-lag Kalman smoother, i.e. a generalization of the KF equations in which the observations at both times of the current analysis

and of the next one are used at each analysis step. Tip: extend the KF control variable to include the model state at both analysis times.

APPENDIX D MAIN SYMBOLS

$\mathbf{x}$ model state vector
$\mathbf{x}_t$ true value of the model state i.e. perfect analysis
$\mathbf{x}_b$ background model state
$\mathbf{x}_a$ analysed model state
$\mathbf{y}$ observation vector
H observation operator (maps $\mathbf{x}$ into the $\mathbf{y}$ space by providing model equivalents of the observed values)
$\mathbf{H}$ linearized observation operator (in the vicinity of a predefined model state)
$\mathbf{B}$ background error covariances (estimation error covariance matrix of $\mathbf{x}_b$)
$\mathbf{A}$ analysis error covariances (estimation error covariance matrix of $\mathbf{x}_a$)
$\mathbf{R}$ observation error covariances (error covariance matrix of $\mathbf{y} - H(\mathbf{x}_t)$)
$\mathbf{K}$ analysis gain matrix
$\mathbf{I}$ identity matrix
J cost function of the variational analysis
J_b background term of the cost function
J_o observation term of the cost function
J_c penalization term of the cost function

REFERENCES

The references below have been chosen because of their educational value. It does not necessarily mean that they are the original historical references. These can usually be found in the reference lists of the paper below.

Bennett, A. and M. Thornburn, 1992: The generalized inverse of a non-linear quasi-geostrophic ocean circulation model. *J. Phys. Oceanogr.*, **3,** 213-230.

Derber, J. and F. Bouttier, 1999: A reformulation of the background error covariance in the ECMWF global data assimilation system. *Accepted for publication in Tellus.*

Courtier, P., J.-N. Thépaut and A. Hollingsworth, 1994: A strategy for operational implementation of 4D-VAR, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, **120,** 1367-1387.

Courtier, P.,1997: Dual formulation of four-dimensional variational assimilation. *Quart. J. Roy. Meteor. Soc.*, **123,** 2449-2461.

Courtier, P., E. Andersson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, F. Rabier and M. Fisher, 1998: The ECMWF implementation of three-dimensional variational assimilation (3D-Var). Part 1: formulation. *Quart. J. Roy. Meteor. Soc.*, **124,** 1783-1807.

Daley, R., 1991: Atmospheric Data Analysis. *Cambridge Atmospheric and Space Science Series,* Cambridge University Press. ISBN 0-521-38215-7, 457 pages.

Errico, R. and T. Vukicevic, 1992: Sensitivity Analysis using an Adjoint of the PSU-NCAR Mesoscale Model. *Mon. Wea. Rev.*, **120,** 1644-1660.

Errico, R., T. Vukicevic and K. Raeder, 1993: Examination of the accuracy of a tangent linear model. *Tellus*, **45A**, 462-477.

Eyre, J., 1987: Inversion of cloudy satellite sounding radiances by nonlinear optimal estimation: theory and simulation for TOVS. *Quart. J. Roy. Meteor. Soc*., **113.**

Ghil, M., 1989: Meteorological Data Assimilation for Oceanographers. Part I: Description and Theoretical Framework. *Dyn. of Atmos. Oceans*, **13**, 171-218.

Gilbert, J.-C. and C. Lemaréchal, 1989: Some numerical experiments with variable storage quasi-Newton algorithms. *Mathematical Programming.*

Hollingsworth, A., and P. Lonnberg, 1986: The statistical structure of short-range forecast errors as determined from radiosonde data. Part I: The wind field. *Tellus*, **38A**, 111-136.

Hollingsworth, A., D. Shaw, P. Lonnberg, L. Illari, K. Arpe and A. Simmons, 1986: Monitoring of observation and analysis quality by a data-assimilation system. *Mon. Wea. Rev.*, **114,** 1225-1242.

Lacarra, J.-F., and O. Talagrand, 1988: Short-Range Evolution of Small Perturbations in a Barotropic Model. *Tellus*, **40A**, 81-95.

Lorenc, A., 1986: Analysis methods for numerical weather prediction. *Quart. J. Roy. Meteor. Soc*., **112**, 1177-1194.

Lorenc, A., 1981: A Global Three-Dimensional Multivariate Statistical Interpolation Scheme. *Mon. Wea. Rev.*,

Parrish, D. and J. Derber, 1992: The National Meteorological Center's spectral statistical-interpolation analysis system. *Mon. Wea. Rev.*, **120,** 1747-1763.

Rabier, F., and P. Courtier, 1992: Four-dimensional assimilation in the presence of baroclinic instability. *Quart. J. Roy.* Meteor. Soc., 118, 649-672.

Talagrand, O. and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quart. J. Roy. Meteor. Soc*., **113,** 1311-1328.

Thepaut, J.-N. and P. Courtier, 1991: Four-dimensional data assimilation using the adjoint of a multi-level primitive-equation model. *Quart. J. Roy. Meteor. Soc.*, **117,** 1225-1254.

Vukicevic, T., 1991: Nonlinear and Linear Evolution of Initial Forecast Errors. *Mon. Wea. Rev.*, **119**, 1602-1611