Introduction to Data Assimilation

Eric Blayo[1,2], Emmanuel Cosme[1,3], Maëlle Nodet[1,2], Arthur Vidard[2]
(1 : Université de Grenoble – 2 : INRIA/LJK – 3 : LEGI)

Last revision: April 9, 2011

Contents

1 Introduction

Section contents

1.1 What is data assimilation?

The basic purpose of data assimilation is to combine different sources of information to estimate at best the state of a system. These sources generally are observations and a numerical model. Why not simply use observations? First, because observations are sparse or partial in geophysics. Some information is necessary to interpolate the information from observations to unobserved regions or quantities. A numerical model naturally does that. Second, because observations can be noised. Combining several noised data is an efficient way to filter out noise and provide a more accurate estimate.

The data assimilation problem may be tackled with different mathematical approaches: signal processing, control theory, estimation theory for example. Stochastic methods, such as the well known Kalman filter, are based on estimation theory. On the other hand, variational methods (3D-Var, 4D-Var...) come from control theory.

The historical development of data assimilation for geophysical fluids can hardly be disconnected from meteorology. It is indeed a necessary step to provide a good initialization for a prediction, and until the 90's data assimilation has been developed and used in that only purpose. Today, its application generalizes to many other fields (atmospheric chemistry, oceanic biochemistry, glaciology, seismology, oil industry, nuclear fusion, medicine, agronomy, etc.) and applications, for example:

- the estimation of the trajectory of a system to study its variability (reanalyses);
- the identification of systematic errors in numerical models;
- the optimization of observation network;
- the estimation of unobserved variables;
- the estimation of parameters.

1.2 A very simple scalar example

Assume we have two different available measurements for a same quantity, $y_1 = 1$ and $y_2 = 2$ of some unknown quantity x. Which estimation can we get for the true value ?

1.2.1 First method

We look for x minimizer of $(x-1)^2 + (x-2)^2$, and we find the estimator $\hat{x} = 3/2$ (least square solution). This solution has the following problems:

- Sensitivity to any change of unit: if $y_1 = 1$ is a measure of x, and $y_2 = 4$ is a measure of $2x$, then minimizing $(x-1)^2 + (2x-4)^2$, leads to $\hat{x} = 9/5$.
- No sensitivity to the accuracy of the measurement: we get the same estimate even if y_1 is more accurate than y_2.

1.2.2 Reformulation in a statistical framework

We define : $Y_i = x + e_i$, where the observation errors e_i satisfy the following hypotheses:

- $\mathrm{E}(e_i) = 0, (i = 1, 2)$ unbiased measurements
- $\mathrm{Var}(e_i) = \sigma_i^2, (i = 1, 2)$ the accuracy is known
- $\mathrm{Cov}(e_1, e_2) = 0$, i.e. $E(e_1 e_2) = 0$ errors are independent

We seek out an estimator (i.e. a random variable) $\hat{X}$ which is

- linear: $\hat{X} = \alpha_1 Y_1 + \alpha_2 Y_2$ (to be simple)
- unbiased: $\mathrm{E}(\hat{X}) = x$ (natural)
- of minimal variance: $\mathrm{Var}(\hat{X})$ minimal (optimal accuracy)

This estimator is called the *BLUE*: Best Linear Unbiased Estimator.
To compute α_i we use the unbiased hypothesis:

$$\mathrm{E}(\hat{x}) = (\alpha_1 + \alpha_2)x + \alpha_1 \mathrm{E}(e_1) + \alpha_2 \mathrm{E}(e_2) = (\alpha_1 + \alpha_2)x$$

So that $\alpha_1 + \alpha_2 = 1$, or $\alpha_2 = 1 - \alpha_1$.
Then we compute the variance of $\hat{x}$:

$$\begin{aligned} \mathrm{Var}(\hat{x}) &= \mathrm{E}((\hat{x} - x)^2) = \mathrm{E}((\alpha_1 e_1 + \alpha_2 e_2)^2) \\ &= \alpha_1^2 \mathrm{E}(e_1^2) + 2\alpha_1\alpha_2 \mathrm{E}(e_1 e_2) + \alpha_2^2 \mathrm{E}(e_2^2) \\ &= \alpha_1^2 \sigma_1^2 + \alpha_2^2 \sigma_2^2 \\ &= \alpha_1^2 \sigma_1^2 + (1 - \alpha_1)^2 \sigma_2^2 \end{aligned}$$

Estimator $\hat{x}$ has to be a minimizer of this variance, which is a function of α_1, minimum where its derivative with respect to α_1 is zero, so that:

$$\alpha_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Then:

$$\hat{x} = \frac{\frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2 y_1 + \sigma_1^2 y_2}{\sigma_1^2 + \sigma_2^2}$$

We get the same result if we minimize the following function:

$$J(x) = \frac{1}{2}\left(\frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right)$$

Remarks:

- This statistical approach allows to rationalize the choice of the norm in the least-square functional J.
- It solves both problems of sensitivity to the units and observation accuracies.
- The accuracy of the estimator is given by the second derivative of J:

$$J''(x) = \frac{1}{\mathrm{Var}(\hat{x})} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

  so that "accuracies are added".

- If we consider that $y_1 = x^b$ is a first guss of x and $y_2 = y$ an additional observation, then

$$\hat{x} = x^b + \frac{{\sigma_b}^2}{\sigma^2 + {\sigma_b}^2}(y - x^b)$$

  The value $y - x^b$ is called "innovation", it contains the additional information given by y with respect to x^b.

1.2.3 Data assimilation methods

Threr are two classes of methods :

- statistical methods: direct computation of the BLUE thanks to algebraic computations ;
- variational methods : minimization of J.

Common characteristics of these methods:

- they give the same result (in the linear case) ;
- they are optimal only in the linear case ;
- common difficulties:
  - accounting for non-linearities
  - accounting for large dimensions
  - error statistics are required but poorly known

1.3 Notations and vocabulary

Internation convention in data assimilation are as follows:

- $\mathbf{x}$ state vector
- $\mathbf{x}^t$ true state
- $\mathbf{x}^b$ background state
- $\mathbf{x}^a$ analyzed state

Superscripts denote vector types, whereas subscripts denote time or space.

1.3.1 Discretization and *true* state

Most of the time, we aim to estimate with the best possible accuracy a geophysical field that vary continuously in space and time. This real, continuous (and possibly multivariate) field is noted $\mathbf{x}$.

Numerical models are often used for that purpose. But numerical models handle only discrete representations of the physical field. Proceeding this way, one implicitly drops the idea of estimating the real state $\mathbf{x}$, and tries to estimate a projection of $\mathbf{x}$ in a discrete space. Let Π be this projector, and $\mathbf{x}^t$ the projection of $\mathbf{x}$:

$$\mathbf{x}^t = \Pi(\mathbf{x}) \tag{1}$$

$\mathbf{x}^t$ is called the *true* state, and is the state to estimate.

Here are considered the dynamical models, i.e. the models that compute the time evolution of the simulated state. Let x_k and x_{k+1} be the real (continuous) state at two consecutive observation times, k being a time index. They are related by a causality link:

$$x_{k+1} = g(x_k). \tag{2}$$

For the reasons already detailed previously, we quickly turn to express of the true state:

$$\mathbf{x}^t_{k+1} = \Pi\left(g(x_k)\right). \tag{3}$$

The model g is not known strictly, although we know (hopefully...) most of the physics involved in it. This physics is represented by our numerical model $\mathcal{M}$, that works with discrete state like $\mathbf{x}^t$. Let us introduce this piece of knowledge into equation 3:

$$\mathbf{x}^t_{k+1} = \mathcal{M}_{k,k+1}(\mathbf{x}^t_k) + \eta_{k,k+1} \tag{4}$$

where

$$\eta_{k,k+1} = \Pi\left(g(x_k)\right) - \mathcal{M}_{k,k+1}(\mathbf{x}^t_k). \tag{5}$$

The *model error* $\eta_{k,k+1}$ accounts for the errors in the numerical model (e.g., misrepresentation of physical processes) and for the errors due to the discretization. The covariance matrix $Q_{k,k+1}$ of the model error is given by

$$Q_{k,k+1} = \mathrm{Cov}(\eta_{k,k+1}) = \mathrm{E}\left((\eta_{k,k+1} - \overline{\eta_{k,k+1}})(\eta_{k,k+1} - \overline{\eta_{k,k+1}})^T\right)$$

1.3.2 Observations

The real field $\mathbf{x}$ results in a real signal $\mathbf{y}$ in the observation space. The causality relation involves a function h:

$$\mathbf{y} = h(\mathbf{x}) \tag{6}$$

Equation 6 is extremely simple but unusable in this form. First, $\mathbf{y}$ is not accessible. The actual, available observation $\mathbf{y}^o$ is spoilt by instrumental errors. Let ϵ^μ denote this measurement error, then

$$\mathbf{y}^o = h(\mathbf{x}) + \epsilon^\mu \tag{7}$$

As seen earlier, the real state $\mathbf{x}$ is also not accessible and only $\mathbf{x}^t$ is searched for. Also, h represents the physics of the measure. Although the physical processes can be known, we cannot known and handle h numerically. In practice, the physics is represented by a numerical model $\mathcal{H}$, which applied to the discrete state $\mathbf{x}^t$, and is called the *observation operator*. Involving $\mathcal{H}$ and Π, equation 7 can be rewritten:

$$\mathbf{y}^o = \mathcal{H}(\mathbf{x}^t) + \underbrace{h(x) - \mathcal{H}(\Pi(\mathbf{x}))}_{\epsilon^r} + \epsilon^\mu \tag{8}$$

ϵ^r is the *representativity error*. It includes the errors related to the representation of the physics in $\mathcal{H}$, and those due to the projection Π of the real state $\mathbf{x}$ on the discrete state space. The sum of the measurement error and the representativity error,

$$\epsilon^o = \epsilon^r + \epsilon^\mu, \tag{9}$$

is the *observation error*, and the final equation that links the true state to the observation is the *observation equation*:

$$\mathbf{y}^o = \mathcal{H}(\mathbf{x}^t) + \epsilon^o \tag{10}$$

The covariance matrix R of the observation error is given by

$$R = \mathrm{Cov}(\epsilon^o) = \mathrm{E}\left((\epsilon^o - \overline{\epsilon^o})(\epsilon^o - \overline{\epsilon^o})^T\right)$$

1.3.3 *A priori* (background) information

Often, particularly in geophysics, we have an *A priori* knowledge of the state $\mathbf{x}^t$, under the form of a vector $\mathbf{x}^b$ of the same dimension as $\mathbf{x}^t$. This is the *background* state. The background error is then defined as:

$$\epsilon^b = \mathbf{x}^b - \mathbf{x}^t \tag{11}$$

Later in the text, we will see that the background state often comes from a model simulation. In this case the background is a *forecast* and is noted $\mathbf{x}^f$ instead. The forecast error is noted ϵ^f. The covariance matrix B of the background error is given by

$$R = \mathrm{Cov}(\epsilon^b) = \mathrm{E}\left((\epsilon^b - \overline{\epsilon^b})(\epsilon^b - \overline{\epsilon^b})^T\right)$$

1.3.4 Analysis

The result of the assimilation process is called the analysis, denoted $\mathbf{x}^a$. The analysis error ϵ^a is given by

$$\epsilon^a = \mathbf{x}^a - \mathbf{x}^t \tag{12}$$

while the covariance matrix A of analysis error is:

$$A = \mathrm{Cov}(\epsilon^a) = \mathrm{E}\left((\epsilon^a - \overline{\epsilon^a})(\epsilon^a - \overline{\epsilon^a})^T\right)$$

Important remark: The choice of the covariance matrices and the background entirely characterize the problem, and its solution. Therefore, all the physics should be done while defining them. Finding the analysis is then only technical work.

1.4 A short history of data assimilation in meteorology

1.4.1 Subjective analysis (19th century)

Subjective analysis consists in extrapolating "by hand" a set of local observations (of pressure, historically) to provide a pressure map. Though subjective, this is a kind of data assimilation, where local observations are combined with the "good sense" and the experience of the meteorologist to provide a map.

1.4.2 Richardson's numerical weather prediction (1922)

Lewis Fry Richardson was the first scientist to try a numerical weather prediction. He made it by hand in 1917, while he was serving in a military unit in the north of France. Unfortunately, his attempt dramatically failed, due to a 145 mbar rise in pressure over 6 hours. The cause has been identified later: the prediction was initialized with in situ pressure observations, an unbalanced input for his numerical model. Lynch (1993) showed that with an appropriate smoothing of the initial condition, Richardson's prediction would have turned fairly accurate. His failure can thus be viewed as due to a deficiency with data assimilation.

1.4.3 Cressman's objective analysis (1950's)

The most relevant idea of Cressman was to acknowledge the poorness of the observation network, and to introduce a background: an *a priori* knowledge of the atmospheric state, to be modified when observations are available. At the grid point j, the correction writes:

$$\mathbf{x}_j^a = \mathbf{x}_j^b + \frac{\sum_{i=1}^s w(i,j)(\mathbf{y}_i - \mathbf{x}_i^b)}{\sum_{i=1}^s w(i,j)} \tag{13}$$

where $\mathbf{y}_i$ is the observation at the grid point i and $w(i,j)$ is the weight of $\mathbf{y}_i$ at the point j. To prescribe the weights, Cressman proposes:

$$\begin{cases} w(i,j) = \frac{R^2 - r(i,j)^2}{R^2 + r(i,j)^2} & \text{if} \quad r(i,j) \leq R \\ w(i,j) = 0 & \text{if} \quad r(i,j) > R \end{cases}$$

$r(i,j)$ is the distance between the points i and j. R is an influence radius to be prescribed. The main difficulty of this method is to determine objectively the weights. The method has other serious drawbacks: all the observations are processed identically, whatever their quality is; the physical balance is not controlled.

1.4.4 Nudging (1970's)

The idea is to force the numerical model toward the observations with an extra term for elastic relaxation. If the model writes:

$$\frac{d\mathbf{x}}{dt} = \mathcal{M}(\mathbf{x}) \tag{14}$$

then the nudging equation is:

$$\frac{d\mathbf{x}}{dt} = \mathcal{M}(\mathbf{x}) + \alpha(\mathbf{y} - \mathbf{x}) \tag{15}$$

where $\mathbf{y}$ is a direct observation of $\mathbf{x}$. This method also displays several drawbacks: the relaxation coefficient α must be determined. The method is no more applicable with undirect observations. Nudging, sometimes referred to as "the poor man's data assimation method", is also very simple to implement. For that reason it is still used for specific applications, mainly when the observations are not real observations but grided data from a reanalysis for example.

1.4.5 Recent methods

Recent methods include 3Dvar and Optimal Interpolation (1980's), as well as 4Dvar and the Kalman filter (1990's). They will be described in the sequel.

Part I
Stochastic Data Assimilation

Part contents

2 Stochastic Estimation

Section contents

2.1 Basic elements in probability and statistics

2.1.1 Probability

Random experiment A random experiment is mathematically described by:

- the set Ω of all possible outcomes of an experiment, the result of which cannot be perfectly anticipated;
- the subsets of Ω, called events;
- a probability function, P: a numerical expression of a state of knowledge. P is such as, for any disjoint events A and B:

$$0 \leq P(A) \leq 1,$$
$$P(\Omega) = 1,$$
$$P(A \cup B) = P(A) + P(B)$$

Conditional probability When the two events A and B are not independent, knowing that B has occurred changes our state of knowledge on A. This reads:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

2.1.2 Real random variables

The outcome of a random experiment is called a random variable. A random variable can be either an integer number (e.g., a die cast) or a real number (e.g., the lifetime of a electric light bulb).

Probability density function For real random variables, equality to a given number is not an event. Only the inclusion into an interval is an event. This defines the **probability density function**, commonly referred to as pdf:

$$P(a < X \leq b) = \int_a^b p(x)dx.$$

Joint and conditional pdf If x and y are two real random variables, $p(x,y)$ is the joint pdf of x and y. Conditioning applies as with the discrete random variables, so that

$$p(x|y) = \frac{p(x,y)}{p(y)}.$$

Expectation and variance A pdf is rarely known completely. Generally, only some properties are determined and handled. The two main properties are the expectation and the variance. The expectation of a random variable x is

$$\mathcal{E}(x) =< x >= \int_{-\infty}^{+\infty} xp(x)dx.$$

The variance is

$$\mathrm{Var}(x) = \mathcal{E}\left([x - \mathcal{E}(x)]^2\right) = \int_{-\infty}^{+\infty} x^2 p(x)dx.$$

The standard deviation is the square root of the variance.

The Gaussian distribution The random variable x has a Gaussian (or normal) distribution with parameters μ and σ^2, which is noted $x \sim \mathcal{N}(\mu, \sigma^2)$, when

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right].$$

The Gaussian distribution possesses some very nice properties, in particular:

- It is a natural distribution for signal noises (a consequence of the central limit theorem);
- the parameters μ and σ^2 of the distribution are the expectation and the variance, respectively;
- If $x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ are two independent variables, then $x_1 + x_2$ is also Gaussian and $x_1 + x_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$;
- If a is a real number and $x \sim \mathcal{N}(\mu, \sigma^2)$, then $ax \sim \mathcal{N}(a\mu, a^2\sigma^2)$.

2.1.3 Real random vectors

Real random vectors are vectors which components are real random variables. The pdf of a vector is the joint pdf of its real components.

Expectation and variance The expectation vector is the vector formed with the expected values of the real components. The second moment of the distribution is the covariance matrix. If $\mathbf{x}$ denotes the random vector, the covariance matrix is defined by

$$\mathcal{E}\left[(\mathbf{x} - \mathcal{E}(\mathbf{x}))\,(\mathbf{x} - \mathcal{E}(\mathbf{x}))^T\right].$$

A covariance matrix is symmetric, positive. The terms on the diagonal are the variances of the vector components. The non diagonal terms are covariances. If x_i and x_j denotes two different components of $\mathbf{x}$, their covariance is

$$\mathrm{Cov}(x_i, x_j) = \mathcal{E}\left[(x_i - \mathcal{E}(x_i))\,(x_j - \mathcal{E}(x_j))^T\right]$$

and their correlation is

$$\rho(x_i, x_j) = \frac{\mathrm{Cov}(x_i, x_j)}{\sqrt{\mathrm{Var}(x_i)\mathrm{Var}(x_j)}}.$$

The multivariate Gaussian distribution The random vector $\mathbf{x}$ of size n has a Gaussian (or normal) distribution with parameters μ and $\mathbf{P}$, which is noted $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{P})$, when

$$p(x) = \frac{1}{(2\pi)^{n/2}|\mathbf{P}|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T \mathbf{P}^{-1}(x-\mu)\right].$$

μ and $\mathbf{P}$ are the expectation and the covariance matrix of $\mathbf{x}$, respectively. $|\mathbf{P}|$ denotes the determinant of $\mathbf{P}$. The component of $\mathbf{x}$ are said to be jointly Gaussian.

2.2 The two pillars of estimation theory

If one has to remember only two formulas from section 2.1, here they are:

Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \tag{16}$$

Marginalization rule:

$$p(\mathbf{y}) = \int p(\mathbf{x},\mathbf{y})d\mathbf{x} = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{17}$$

where:

- $p(\mathbf{y}|\mathbf{x})$ is the measurement model (or likelihood);
- $p(\mathbf{x})$ is the prior distribution;
- $p(\mathbf{y})$ is the marginal distribution (or evidence).

2.3 Optimal estimates

The optimal estimate of the random variable $\mathbf{x}$ given the observation $\mathbf{y}$ is the value that best reflects what a realization of $\mathbf{x}$ can really be in regard to $\mathbf{y}$. This definition is subjective, so that several criteria can be proposed to define optimality. For illustration three optimal estimators are presented below, although in the rest of this course only the minimum variance estimator will be considered.

2.3.1 Minimum variance estimation

The estimate is defined such as the spread around it is minimal. The measure of the spread is the variance. If $p(\mathbf{x}|\mathbf{y})$ is the pdf of $\mathbf{x}$, the minimum variance estimate $\hat{\mathbf{x}}_{MV}$ is the solution to:

$$\frac{\partial \mathcal{J}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = 0 \tag{18}$$

where

$$\mathcal{J}(\hat{\mathbf{x}}) = \int (\mathbf{x}-\hat{\mathbf{x}})^T(\mathbf{x}-\hat{\mathbf{x}})p(\mathbf{x}|\mathbf{y})d\mathbf{x}.$$

It is easy to show that the solution is the expectation of the pdf, $\hat{\mathbf{x}}_{MV} = \mathcal{E}(\mathbf{x}|\mathbf{y})$.

2.3.2 Maximum *A Posteriori* estimation

The estimate is defined as the most probable value of $\mathbf{x}$ given $\mathbf{y}$, i.e., the value that maximizes the conditional pdf $p(\mathbf{x}|\mathbf{y})$. $\hat{\mathbf{x}}_{MAP}$ is such that

$$\frac{\partial p(\mathbf{x}|\mathbf{y})}{\partial \mathbf{x}} = 0 \tag{19}$$

With a Gaussian pdf, the minimum variance and the Maximum A Posteriori estimators are identical.

2.3.3 Maximum Likelihood estimation

The estimate is defined as the most probable value of $\mathbf{y}$ given $\mathbf{x}$, i.e., the value that maximizes the conditional pdf $p(\mathbf{y}|\mathbf{x})$. $\hat{\mathbf{x}}_{ML}$ is such that

$$\frac{\partial p(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x}} = 0 \tag{20}$$

The ML estimator can be seen as the MAP estimator without any prior information $p(\mathbf{x})$.

2.4 The best linear unbiased estimate (BLUE)

Here we aim at estimating the true state $\mathbf{x}^t$ of a system, assuming a background estimate $\mathbf{x}^b$ and a partial observation $\mathbf{y}^o$ are given. These data are assumed unbiased and their uncertainties are also given, in the form of the covariance matrices $\mathbf{P}^b$ and $\mathbf{R}$, respectively. The observation operator is assumed linear. To summarize, we have the following pieces of information:

$$\mathbf{H} \quad , \quad \text{with} \qquad \mathbf{y}^o = \mathbf{H}\mathbf{x}^t + \epsilon^o \tag{21}$$
$$\mathbf{x}^b = < \mathbf{x}^t > \tag{22}$$
$$\mathbf{P}^b = < \epsilon^b \epsilon^{bT} > \tag{23}$$
$$< \epsilon^o > = = 0 \tag{24}$$
$$\mathbf{R} = < \epsilon^o \epsilon^{oT} > \tag{25}$$
$$\tag{26}$$

The best estimate is searched for as a linear combination of the background estimate and the observation:

$$\mathbf{x}^a = \mathbf{A}\mathbf{x}^b + \mathbf{K}\mathbf{y}^o \tag{27}$$

where $\mathbf{A}$ and $\mathbf{K}$ are to be determined to make the estimation optimal. In that goal, some criteria must be formulated to define optimality. Given the information provided here, a wise choice is to search for an unbiased estimate, with minimum variance. Thus we try to find $\mathbf{A}$ and $\mathbf{K}$ that makes:

$$< \epsilon^a >= 0 \tag{28}$$
$$\mathrm{tr}(\mathbf{P}^a) \text{ minimum} \tag{29}$$

These requirements are reached when

$$\mathbf{A} = -\mathbf{K} \tag{30}$$
$$\mathbf{K} = \mathbf{P}^b\mathbf{H}^T(\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R})^{-1} \tag{31}$$

and $\mathbf{K}$ is called the *Kalman gain*. The *a posteriori* covariance matrix can also be computed. The final form of the update equations is

$$\mathbf{K} = \mathbf{P}^b\mathbf{H}^T(\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R})^{-1} \tag{32}$$
$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^b) \tag{33}$$
$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^b \tag{34}$$

and this constitutes the Best Linear Unbiased Estimate (BLUE) equations, under the constraint of the minimum variance.

2.5 The gaussian case

If we know that the a priori and the observation pieces of information are both gaussian, Bayes'rule may be applied to compute the a posteriori pdf. With:

$$\mathbf{x}^t \sim \mathcal{N}(\mathbf{x}^b, \mathbf{P}^b), \quad p(\mathbf{x}^t) = \frac{1}{(2\pi)^{n/2}|\mathbf{P}^b|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}^t - \mathbf{x}^b)^T \mathbf{P}^{b\,-1}(\mathbf{x}^t - \mathbf{x}^b)\right],$$

$$\mathbf{y}^o \sim \mathcal{N}(\mathbf{H}\mathbf{x}^t, \mathbf{R}), \quad p(\mathbf{y}^o|\mathbf{x}^t) = \frac{1}{(2\pi)^{n/2}|\mathbf{R}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)^T \mathbf{R}^{-1}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)\right].$$

Then Bayes'rule provides the a posteriori pdf:

$$p(\mathbf{x}^t|\mathbf{y}^o) \propto \exp(-J) \tag{35}$$

with

$$J(\mathbf{x}^t) = \frac{1}{2}\left[(\mathbf{x}^t - \mathbf{x}^b)^T \mathbf{P}^{b\,-1}(\mathbf{x}^t - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)^T \mathbf{R}^{-1}(\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)\right]. \tag{36}$$

It can be easily shown that equation 36 leads to

$$J(\mathbf{x}^t) = \frac{1}{2}\left[(\mathbf{x}^t - \mathbf{x}^a)^T \mathbf{P}^{a\,-1}(\mathbf{x}^t - \mathbf{x}^a)\right] + \beta, \tag{37}$$

with

$$\mathbf{P}^a = \left[\mathbf{P}^{b\,-1} + \mathbf{H}^T \mathbf{R}^{-1}\mathbf{H}\right]^{-1}, \tag{38a}$$

$$\mathbf{x}^a = \mathbf{P}^a\left[\mathbf{P}^{b\,-1}\mathbf{x}^b + \mathbf{H}^T \mathbf{R}^{-1}\mathbf{y}^o\right]. \tag{38b}$$

and β is independent of $\mathbf{x}^t$. With the help of the Sherman-Morrison-Woodbury (SMW) formula:

$$[\mathbf{A} + \mathbf{UDV}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}[\mathbf{D}^{-1} + \mathbf{VA}^{-1}\mathbf{U}]^{-1}\mathbf{VA}^{-1}, \tag{39}$$

it can be shown that these are the BLUE equations. The *a posteriori* pdf 35 is thus gaussian, and its parameters are given by the BLUE equations. Hence with gaussian pdfs and linear observation operator, there is no need to use Bayes'rule: the BLUE equations can be used instead to compute the parameters of the resulting pdf. Since the BLUE provides the same result as Bayes'rule, it is the best estimator of all.

In passing, one can recognize the 3D-Var cost function. By minimizing this cost function, 3D-Var finds the MAP estimate of the gaussian pdf, what is equivalent to the MV estimate found by the BLUE.

3 The Kalman filter

Section contents

3.1 Introduction

The system is now dynamical. Instead of a unique state, we aim at estimating a series of states $\mathbf{x}_k^t$, where the subscript k is a time index pointing observation dates. We assume to have the following *a priori* pieces of knowledge:

- the initial state $\mathbf{x}_0^t$ is gaussian-distributed with mean $\mathbf{x}_0^b$ and covariance $\mathbf{P}_0^b$;
- a linear dynamical model $\mathbf{M}_k$ that describes the state evolution;
- the model errors η_k are gaussian-distributed with mean 0 (unbiased error) and covariance $\mathbf{Q}_k$;
- the model errors are white in time: $< \eta_k \eta_j^T >= 0$ if $k \neq j$;
- linear observation operators that link the states to the observations;
- the observation errors ϵ_k^o are gaussian-distributed with mean 0 (unbiased errors) and covariance $\mathbf{R}_k$;
- the observation errors are white in time: $< \epsilon_k^o \epsilon_j^{oT} >= 0$ if $k \neq j$;
- Errors of different types are independent: $< \eta_k \epsilon_j^{oT} >= 0$, $< \eta_k {\epsilon_0^b}^T >= 0$, $< \epsilon_k^o {\epsilon_0^b}^T >= 0$.

Under these hypotheses, the Kalman filter provides the estimate of the states $\mathbf{x}_k^t$, conditioned to the past and present observations $\mathbf{y}_0, \ldots, \mathbf{y}_k$; in terms of pdf, it is $p(\mathbf{x}_k | \mathbf{y}_{0:k})$ where $\mathbf{y}_{0:k} = \{\mathbf{y}_0, \ldots, \mathbf{y}_k\}$. The Kalman filter algorithm is sequential and decomposed in 2 steps: *analysis* (or observational update) and *forecast*.

3.2 Kalman filter algorithm

3.2.1 Analysis step

At time t_k, $p(\mathbf{x}_k | \mathbf{y}_{0:k-1})$ is known through the mean $\mathbf{x}_k^f$, the covariance matrix $\mathbf{P}_k^f$, and the assumption of a gaussian distribution. The use of the notation f in superscript will be made clear in the next section. If $k = 0$, f must be replaced by b.

The analysis step consists in updating this pdf using the observation available at time t_k, and find $p(\mathbf{x}_k|\mathbf{y}_{0:k})$. This is straightforward: As exhibited in section 2.1.3, the latter pdf is gaussian and the parameters are computed with the BLUE equations. We note $\mathbf{x}_k^a$ and $\mathbf{P}_k^a$ the mean and covariance matrix of $p(\mathbf{x}_k|\mathbf{y}_{0:k})$. The superscript '$a$' stands for *analysis*.

3.2.2 Forecast step

After the analysis step, the gaussian pdf $p(\mathbf{x}_k|\mathbf{y}_{0:k})$ is known through the mean $\mathbf{x}_k^a$ and the covariance matrix $\mathbf{P}_k^a$. To find an estimate of $\mathbf{x}_{k+1}^t$ with the same conditioning, the dynamical model must be used. This provides the *forecast*, denoted with a superscript f. A straightforward computation leads to

$$\mathbf{x}_{k+1}^f = \mathbf{M}_{k,k+1}\mathbf{x}_k^a \tag{40}$$

and

$$\mathbf{P}_{k+1}^f = \mathbf{M}_{k,k+1}\mathbf{P}_k^a\mathbf{M}_{k,k+1}^T + \mathbf{Q}_k. \tag{41}$$

It must be noticed here that $\mathbf{x}_{k+1}^t$ is a linear transformation of a gaussian variable ($\mathbf{x}_k^t$) to which a gaussian noise (η_k) adds. It is then gaussian, so that the mean and covariance matrix suffice to describe the full pdf. Optimality will be guaranteed at the next analysis step.

3.2.3 Synthesis

The Kalman filter is initialized with an forecast state vector $\mathbf{x}_0^f$ and the associated error covariance matrix $\mathbf{P}_0^f$. The assimilation sequence is performed according to the Kalman filter equations:

Initialization: $\mathbf{x}_0^f$ and $\mathbf{P}_0^f$

Analysis step:

$$\begin{aligned}
\mathbf{K}_k &= (\mathbf{H}_k\mathbf{P}_k^f)^T[\mathbf{H}_k(\mathbf{H}_k\mathbf{P}_k^f)^T + \mathbf{R}_k]^{-1}, && (42a)\\
\mathbf{x}_k^a &= \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k^o - \mathbf{H}_k\mathbf{x}_k^f), && (42b)\\
\mathbf{P}_k^a &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f. && (42c)
\end{aligned}$$

Forecast step:

$$\begin{aligned}
\mathbf{x}_{k+1}^f &= \mathbf{M}_{k,k+1}\mathbf{x}_k^a, && (43a)\\
\mathbf{P}_{k+1}^f &= \mathbf{M}_{k,k+1}\mathbf{P}_k^a\mathbf{M}_{k,k+1}^T + \mathbf{Q}_k. && (43b)
\end{aligned}$$

Another formulation for the analysis starts with the computation of the inverse of the covariance matrix (sometimes called the *information* matrix):

$$\begin{aligned}
\mathbf{P}_k^{a\,-1} &= \mathbf{P}_k^{f\,-1} + \mathbf{H}_k{\mathbf{R}_k}^{-1}\mathbf{H}_k, && (44a)\\
\mathbf{K}_k &= \mathbf{P}_k^a{\mathbf{H}_k}^T{\mathbf{R}_k}^{-1}, && (44b)\\
\mathbf{x}_k^a &= \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k^o - \mathbf{H}_k\mathbf{x}_k^f), && (44c)
\end{aligned}$$

but this is not usually used in geophysics.

3.3 Implementation issues

3.3.1 Definition of covariance matrices, filter divergence

If the input statistical information is mis-specified, the filtering system may come to underestimate the state error variances. Too much confidence is then given to the state estimation and the effects of the analyses are then minimized. In the extreme case, observations are simply rejected. This is a filter divergence.
Very often filter divergence is quite easy to diagnose: state error variances are small and the time sequence of innovations is biased. But it is not always simple to correct. The main rule to follow is not to underestimate model errors. If possible, it is better to use an adaptive scheme to tune them online.

3.3.2 Problem dimensions

The first limitation of the straightforward implementation of the Kalman filter is the problem dimension. In oceanography or meteorology, models generally involve several millions (very often tens of millions, even hundreds of millions sometimes) of variables. Let us call n the number of variables. A state covariance matrix is then $n \times n$. With the dimensions considered, the storage of such matrix is simply impossible. The standard solution is *rank reduction*. The theoretical description holds in two steps:

Square-root decomposition of the covariance matrix: A covariance matrix is symmetric, positive definite. It can be square-root reduced as:

$$\mathbf{P}^f = \hat{\mathrm{S}}^f \hat{\mathrm{S}}^{f\,T} \tag{45}$$

where $\hat{\mathrm{S}}^f$ is a $n \times n$ matrix. It is not unique: a Cholesky decomposition provides a lower triangular matrix. A singular value decomposition provides a unitary matrix multiplied by a diagonal matrix (holding the square roots of the eigenvalues of $\mathbf{P}^f$. But anyway these methods can rarely be applied, simply because $\mathbf{P}^f$ cannot be explicited and stored.

Rank reduction: This consists in reducing by several orders of magnitude the number of columns of the square root matrix. A number m of typically a hundred of columns are considered and form a $n \times m$ matrix that we call $\mathbf{S}$ (f or a).

3.3.3 Evolution of the state error covariance matrix

The matrix propagation equation 43b theoretically provides a symmetric matrix. But its implementation can be not. An example is: $\mathbf{W} = \mathbf{M}_{k,k+1}\mathbf{P}^a_k$, then $\mathbf{P}^f_{k+1} = \mathbf{M}_{k,k+1}\mathbf{W}^T + \mathbf{Q}_k$. In certain circumstances numerical truncation errors may lead to an asymmetric covariance matrix and to the collapse of the filter. A simple recipe is to add an extra step that forces symmetry, for instance: $\mathbf{P}^f_{k+1} = (\mathbf{P}^f_{k+1} + \mathbf{P}^{f\,T}_{k+1})/2$. Another way is to use the square root formulation of the covariance matrix, and compute

$$\mathbf{P}^f_{k+1} = (\mathbf{M}_{k,k+1}\mathbf{S}^a_k)(\mathbf{M}_{k,k+1}\mathbf{S}^a_k)^T + \mathbf{Q}_k. \tag{46}$$

3.3.4 Nonlinear dynamics

Nonlinear dynamics poses two problems to the Kalman filter. First, the transposed model is not defined. Then, nonlinearity spoils gaussianity of statistics. The way to proceed with nonlinearity

is given by the Extended Kalman Filter (EKF), presented next. One must be aware that the EKF is no more optimal, even with initially gaussian statistics, and is valid only for weakly nonlinear dynamics.

3.4 The Extended Kalman Filter (EKF)

When the dynamical model $\mathcal{M}$ and the observation operator $\mathcal{H}$ are (weakly) nonlinear, the Kalman is said to be extended (to nonlinear models). $\mathbf{M}$ and $\mathbf{H}$ denote the tangent linear models of $\mathcal{M}$ and $\mathcal{H}$ respectively.

Initialization: $\mathbf{x}_0^f$ and $\mathbf{P}_0^f$
Analysis step:

$$\mathbf{K}_k = (\mathbf{H}_k\mathbf{P}_k^f)^T[\mathbf{H}_k(\mathbf{H}_k\mathbf{P}_k^f)^T + \mathbf{R}_k]^{-1}, \tag{47a}$$

$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{y}_k^o - \mathcal{H}_k(\mathbf{x}_k^f)), \tag{47b}$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f. \tag{47c}$$

Forecast step:

$$\mathbf{x}_{k+1}^f = \mathcal{M}_{k,k+1}(\mathbf{x}_k^a), \tag{48a}$$

$$\mathbf{P}_{k+1}^f = \mathbf{M}_{k,k+1}\mathbf{P}_k^a\mathbf{M}_{k,k+1}^T + \mathbf{Q}_k. \tag{48b}$$

Part II
Variational Data Assimilation

Part contents

4 Adjoint Method

Section contents

4.1 First example

We consider the following ordinary differential equation:

$$\left\{ \begin{array}{l} -bu''(x) + cu'(x) = f(x), \quad x \in]0,1[\\ u(0) = 0, \quad u(1) = 0 \end{array} \right.$$

where f is given in $L^2(]0,1[)$, b and c are unknown parameters, which we would like to estimate thanks to an observation of $u(x)$ on $]0,1[$. The cost function associated to this problem is:

$$J(b,c) = \int_0^1 \left(u(x) - u^{obs}(x) \right)^2 dx$$

We compute its gradient:

$$\begin{array}{rcl} J(b+\alpha\delta b, c+\alpha\delta c) - J(b,c) & = & \int_0^1 \left(u_{b+\alpha\delta b, c+\alpha\delta c} - u^{obs} \right)^2 - \left(u_{b,c} - u^{obs} \right)^2 \\ & = & \int_0^1 \left(u_{b+\alpha\delta b, c+\alpha\delta c} + u_{b,c} - 2u^{obs} \right) \left(u_{b+\alpha\delta b, c+\alpha\delta c} - u_{b,c} \right) \end{array}$$

If we denote $\widetilde{u} = u_{b+\alpha\delta b, c+\alpha\delta c}$, $u = u_{b,c}$, we have:

$$J(b+\alpha\delta b, c+\alpha\delta c) - J(b,c) = \int_0^1 \left(\widetilde{u} + u - 2u^{obs} \right) \left(\widetilde{u} - u \right)$$

Dividing by α and with $\alpha \to 0$, we get:

$$\hat{J}[b,c](\delta b, \delta c) = 2 \int_0^1 \left(u - u^{obs} \right) \hat{u}, \qquad \text{with } \hat{u} = \lim_{\alpha \to 0} \frac{\widetilde{u} - u}{\alpha}$$

Let us now look for the equation satisfied by $\hat{u}$. We have

$$\left\{ \begin{array}{l} -(b+\alpha\delta b)\widetilde{u}'' + (c+\alpha\delta c)\widetilde{u}' = f \\ \widetilde{u}(0) = 0, \quad \widetilde{u}(1) = 0 \end{array} \right.$$

$$\left\{ \begin{array}{l} -bu'' + cu' = f \\ u(0) = 0, \quad u(1) = 0 \end{array} \right.$$

Therefore

$$\begin{cases} -b\hat{u}'' - \delta b u'' + c\hat{u}' + \delta c u' = 0 \\ \hat{u}(0) = 0, \quad \hat{u}(1) = 0 \end{cases}$$

And we get the so-called *tangent linear model*:

$$\begin{cases} -b\hat{u}'' + c\hat{u}' = \delta b u'' - \delta c u' \\ \hat{u}(0) = 0, \quad \hat{u}(1) = 0 \end{cases}$$

We want to reformulate $\int_0^1 (u - u^{obs})\hat{u}$, so we multiply the linar tangent model by a variable p and we integrate:

$$-b\int_0^1 \hat{u}''p + c\int_0^1 \hat{u}'p = \int_0^1 (\delta b\, u'' - \delta c\, u')p$$

We compute separately:

$$\begin{aligned} \int_0^1 \hat{u}''p &= [\hat{u}'p]_0^1 - \int_0^1 \hat{u}'p' \\ &= [\hat{u}'p - \hat{u}p']_0^1 + \int_0^1 \hat{u}p'' \\ &= \hat{u}'(1)p(1) - \hat{u}'(0)p(0) + \int_0^1 \hat{u}p'' \\ \int_0^1 \hat{u}'p &= [\hat{u}p]_0^1 - \int_0^1 \hat{u}p' \\ &= -\int_0^1 \hat{u}p' \end{aligned}$$

Then:

$$\begin{aligned} & -b\left(\hat{u}'(1)p(1) - \hat{u}'(0)p(0) + \int_0^1 \hat{u}p''\right) + c\left(-\int_0^1 \hat{u}p'\right) = \int_0^1 (\delta b\, u'' - \delta c\, u')p \\ \Leftrightarrow \quad & \int_0^1 (-bp'' - cp')\hat{u} = b\hat{u}'(1)p(1) - b\hat{u}'(0)p(0) + \int_0^1 (\delta b\, u'' - \delta c\, u')p \end{aligned}$$

Let us now write

$$\begin{cases} -bp'' - cp' = 2(u - u^{obs}) \\ p(0) = 0, \quad p(1) = 0 \end{cases}$$

(these equations are called the *adjoint model*)
We then have

$$2\int_0^1 (u - u^{obs})\hat{u} = \int_0^1 (-bp'' - cp')\hat{u} = \delta b\left(\int_0^1 pu''\right) + \delta c\left(-\int_0^1 pu'\right)$$

Therefore

$$\nabla J(b, c) = \left(\int_0^1 pu'', -\int_0^1 pu'\right)$$

We just computed the gradient thanks to the adjoint model.

4.2 General adjoint method for initial state estimation

We consider the following model:

$$\begin{cases} \dfrac{dX}{dt} = M(X), \quad \text{in } \Omega \times [0, T] \\ X(t = 0) = U \end{cases}$$

with the cost function

$$J(U) = \frac{1}{2}\int_0^T \|HX - Y^o\|^2$$

4.2.1 Directional derivative of J

We perturb U along u. We denote by $\widetilde{X}$ the associated solution:

$$\left\{ \begin{array}{l} \dfrac{d\widetilde{X}}{dt} = M(\widetilde{X}) \\ \widetilde{X}(t=0) = U + \alpha u \end{array} \right.$$

We then have

$$\begin{array}{rcl} J(U+\alpha u) - J(u) & = & \dfrac{1}{2}\displaystyle\int_0^T \|H\widetilde{X} - Y\|^2 - \|HX - Y\|^2 \\ & = & \dfrac{1}{2}\displaystyle\int_0^T (H\widetilde{X} - Y, H\widetilde{X} - HX + HX - Y) - (HX - Y, HX - Y) \\ & = & \dfrac{1}{2}\displaystyle\int_0^T (H\widetilde{X} - Y, H(\widetilde{X} - X)) + (H\widetilde{X} - Y - (HX - Y), HX - Y) \\ & = & \dfrac{1}{2}\displaystyle\int_0^T (H\widetilde{X} - Y, H(\widetilde{X} - X)) + (H(\widetilde{X} - X), HX - Y) \end{array}$$

Let us write

$$\hat{X} = \lim_{\alpha \to 0} \frac{\widetilde{X} - X}{\alpha}$$

and we compute the gradient of J:

$$\begin{array}{rcl} \hat{J}[U](u) & = & \displaystyle\lim_{\alpha \to 0} \frac{J(U+\alpha u) - J(u)}{\alpha} \\ & = & \dfrac{1}{2}\displaystyle\int_0^T (HX - Y, H\hat{X}) + (H\hat{X}, HX - Y) \\ & = & \displaystyle\int_0^T (H\hat{X}, HX - Y) \\ & = & \displaystyle\int_0^T (\hat{X}, H^T(HX - Y)) \end{array}$$

4.2.2 Tangent linear model

We substract equations of $\widetilde{X}$ and X and we get:

$$\left\{ \begin{array}{l} \dfrac{d(\widetilde{X} - X)}{dt} = M(\widetilde{X}) - MX = \left[\dfrac{\partial M}{\partial X}\right](\widetilde{X} - X) + \dfrac{1}{2}(\widetilde{X} - X)^T \left[\dfrac{\partial^2 M}{\partial X^2}\right](\widetilde{X} - X) + \ldots \\ (\widetilde{X} - X)(t=0) = \alpha u \end{array} \right.$$

Dividing by α and passing to the limit $\alpha \to 0$, we get:

$$\left\{ \begin{array}{l} \dfrac{d\hat{X}}{dt} = \left[\dfrac{\partial M}{\partial X}\right]\hat{X} \\ \hat{X}(t=0) = u \end{array} \right.$$

These equations are called the *tangent linear model*.

4.2.3 Adjoint model

As before, we multiply the tangent linear equation by P and we integrate by parts over $[0,T]$:

$$\begin{aligned}
\int_0^T \left(\frac{d\hat{X}}{dt}, P\right) &= -\int_0^T \left(\hat{X}, \frac{dP}{dt}\right) + \Big[(\hat{X}, P)\Big]_0^T \\
&= -\int_0^T \left(\hat{X}, \frac{dP}{dt}\right) + (\hat{X}(T), P(T)) - (\hat{X}(0), P(0)) \\
&= -\int_0^T \left(\hat{X}, \frac{dP}{dt}\right) + (\hat{X}(T), P(T)) - (u, P(0)) \\
\int_0^T \left(\left[\frac{\partial M}{\partial X}\right]\hat{X}, P\right) &= \int_0^T \left(\hat{X}, \left[\frac{\partial M}{\partial X}\right]^T P\right)
\end{aligned}$$

So that we get

$$\int_0^T \left(\frac{d\hat{X}}{dt} - \left[\frac{\partial M}{\partial X}\right]\hat{X}, P\right) = 0 = \int_0^T \left(\hat{X}, -\frac{dP}{dt} - \left[\frac{\partial M}{\partial X}\right]^T P\right) + (\hat{X}(T), P(T)) - (u, P(0))$$

If we identify with

$$\hat{J}[U](u) = \int_0^T (\hat{X}, H^T(HX - Y))$$

We get the *adjoint model* equations:

$$\begin{cases} \dfrac{dP}{dt} + \left[\dfrac{\partial M}{\partial X}\right]^T P = H^T(HX - Y) \\ P(t = T) = 0 \end{cases}$$

We can note that the adjoint model is backward in time: the equation goes from $t = T$ to $t = 0$.

4.2.4 Gradient computation

The adjoint model allows to write the gradient in a simpler form:

$$\begin{aligned}
\hat{J}[U](u) &= \int_0^T (\hat{X}, H^T(HX - Y)) \\
&= \int_0^T (\hat{X}, \frac{dP}{dt} + \left[\frac{\partial M}{\partial X}\right]^T P) \\
&= -(u, P(0))
\end{aligned}$$

As we have

$$\hat{J}[U](u) = (\nabla J_U, u)$$

We therefore get

$$\nabla J_U = -P(0)$$

Remark: The gradient is thus computable thanks to one backward integration of the adjoint model. This should be compared to the huge number of direct model integration required using rate of change method.

4.3 Optimization algorithms: descent methods

4.3.1 General principle

We consider the following problem:

Problème 1 *Find the minimizer* $\hat{x}$ *:*

$$J(\hat{x}) = \min_{x \in \mathbb{R}^n} J(x)$$

We call *descent method* algorithm of the type

$$x_{k+1} = x_k + \alpha_k d_k, \text{ tel que } J(x_{k+1}) < J(x_k)$$

where

- $d_k \in \mathbb{R}^n$ is the descent direction at iteration k,
- $\alpha_k \in \mathbb{R}$ is the descent step at iteration k.

Descent methods differ by the choices of α_k and d_k.

4.3.2 Optimal and fixed step methods

Let us assume here that d_k has been chosen. *Optimal step methods* compute α_k so that

$$J(x_k + \alpha_k d_k) = \min_{\alpha \in \mathbb{R}} J(x_k + \alpha d_k)$$

In other words, we minimize J in the direction d_k (can be costful).
Fixed step methods just set:

$$\alpha_k = \alpha, \ \forall k$$

4.3.3 Relaxation method

The idea is to choose for d_k the basis vectors:

$$x_{k+1} = x_k + \alpha_k e_k$$

In other words, we update x_i coefficient by coefficient. This method is simple, but can be very slow to converge if the dimension of the problem is large.

4.3.4 Gradient descent method

The gradient is defined by the formula:

$$J(x_k + h) = J(x_k) + (\nabla J(x_k), h) + o(h)$$

So that, if $\nabla J(x_k) \neq 0$, the change of J is maximized if we set $h = -\alpha \nabla J(x_k)$, in other words:

$$d_k = -\nabla J(x_k)$$

In the particular case where $J(x) = \frac{1}{2}(Ax, x) - (b, x)$, with A square, symmetric and positive definite, we can use the *conjugate gradient algorithm*:

$$\left\{ \begin{array}{rcll} d_k & = & \nabla J(x_k) + d_{k-1} \dfrac{\|\nabla J(x_k)\|^2}{\|\nabla J(x_{k-1})\|^2} & \\ \alpha_k & = & -\dfrac{(\nabla J(x_k), d_k)}{(Ad_k, d_k)} & (\text{et } \nabla J(x_k) = Ax_k - b) \end{array} \right.$$

Remarks:

- Optimal step method ;
- Converges at least in n iterations ;
- It costs $O(n^3)$: not interesting for a full matrix, because looking for the minimum is in this case equivalent to solving $Ax = b$ and Choleski method is better. However, if A is sparse, this algorithm does not require the storage of A, only the matrix-vector products Ad_k and Ax_k.

4.3.5 Newton methods

In 1-D: we want to solve $f(x) = 0$. We assume that x_k is known, and we define x_{k+1} as the intersection point of the tangent line to the graph of f in x_k, with the x-axis:

$$\frac{f(x_k) - 0}{x_k - x_{k+1}} = f'(x_k) \quad \Rightarrow \quad x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

In n-D:

$$\left\{ \begin{array}{rcll} f_1(x_1, x_2, \ldots, x_n) & = & 0 & \\ \ldots & = & 0 & (\Leftrightarrow f(x) = 0) \\ f_n(x_1, x_2, \ldots, x_n) & = & 0 & \end{array} \right.$$

The Newton's method is then

$$x^{(k+1)} = x^{(k)} - \left[f'(x^{(k)}) \right]^{-1} f(x^{(k)})$$

where $f'(x^{(k)})$ is the Jacobian matrix $\left[\partial_j f_i(x^{(k)}) \right]$. At every iteration we have to compute this matrix and solve the linear system $\left[f'(x^{(k)}) \right] \delta x^{(k)} = -f(x^{(k)})$.

Application to optimization Newton's method applied to the Euler equation $\nabla J(x) = 0$ gives:

$$x_{k+1} = x_k - \left[\nabla^2 J(x_k) \right]^{-1} \nabla J(x_k)$$

where $\nabla^2 J(x_k)$ is the Hessian matrix of J.
The main difference with gradient descent methods is that the descent direction is not $\nabla J(x_k)$ anymore, but $\left[\nabla^2 J(x_k) \right]^{-1} \nabla J(x_k)$.

Remarks: For large scale systems, computing $\nabla^2 J(x_k)$ is out of range. Algorithms such as *Quasi-Newton* do not require Hessian computations, but provide approximations of the Hessian which improve with iterations, at a reasonable cost (e.g. M1QN3 algorithm).

5 Variational data assimilation algorithms

Section contents

5.1 Introduction

Definitions and notations:

- *Assimilation window*: time window over which the data will be considered all at once
- *First guess* or *background*: prior estimation of the control vector ($\mathbf{x}^b$)
- *Analysis*: estimation of the control vector after data assimilation ($\mathbf{x}^a$)
- *Increment*: correction to the control vector ($\mathbf{x}^a - \mathbf{x}^b$)
- *Innovation vector*: misfit to the observation ($\mathbf{d} = \mathbf{y}^o - H(\mathbf{x})$)

In this chapter we consider a time evolving system described by a set of non linear PDE (aka the model M)

$$\left\{ \begin{array}{ll} \mathbf{x}_i = M_i(\mathbf{x}_{i-1}), & i = 1, N \\ \mathbf{x}_0 = \quad \mathbf{x}^b & \end{array} \right. \tag{49}$$

If we note the model state trajectory:

$$\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_{N-1}, \mathbf{x}_N)^T$$

N being the number of time steps per assimilation window, the full variational data assimilation scheme can be defined by the minimization of a cost function of the form:

$$J(\mathbf{x}) = \overbrace{(\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b)}^{J^b} + \overbrace{(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x}))}^{J^o} \tag{50}$$

Where $\mathbf{B}$ and $\mathbf{R}$ are the background and observation error correlation matrices respectively, $\mathbf{y}$ is the observation vector, and H the observation operator.

The meaning of this cost function is that we seek for a state trajectory $\mathbf{x}$ that is satisfying the background error statistics (J^b), that is not far from the observation (J^o).

In practice this algorithm is not doable for large time dependent problems. The size of $\mathbf{x}$ (the size of the state vector times the number of time step of the assimilation window) becomes huge and the definition of $\mathbf{B}$ is problematic (it should include statistics of the model error).

Therefore, in practice, additional hypothesis or approximations have to be made in order to implement variational data assimilation.

5.2 3D-Var

5.2.1 Cost function and algorithm

For time depending problems, the 3D-Var algorithm is a simplification of the *full* variational data assimilation scheme, making the assumption $M_i = I$ over the assimilation window. Then $\mathbf{x} = \mathbf{x}_0$ and the cost function of 3D-Var becomes:

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}^b) + (\mathbf{y} - H(\mathbf{x}_0))^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x}_0))$$

with usual notations. The gradient of J is given by

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

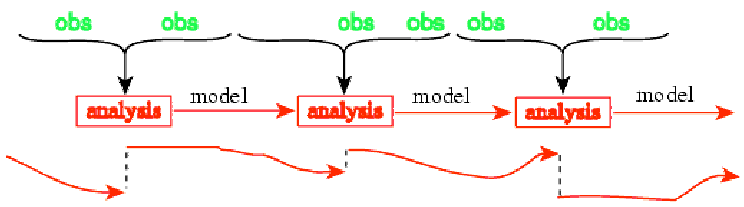
The iterative algorithm use the fact that ∇J is small enough as stopping criterion, in general a maximal number of iteration is also given:

Algorithme 2 (3D-Var)

- *Initialisation :* $\mathbf{x}_0 = \mathbf{x}^b$, $n = 0$
- *While* $\|\nabla J\| > \varepsilon$ *or* $n \leq n_{max}$, *do :*
  1. *Compute* J
  2. *Compute* ∇J
  3. *Descente and update of* $\mathbf{x}_0$
  4. $n = n + 1$

Regarding the B matrix: as for the BLUE, the dimension of this matrix makes its explicit storage impossible in general, therefore it is necessary to model this matrix. Luckily, only matrix-vector product using B^{-1} are required, this allows complex modelling as operator (*i.e.* one defines a function with ϕ as input and $B^{-1}\phi$ as output).

Regarding the observations: This algorithm is more dedicated to stationary problems, however it has been used for long (and still is) for non stationary problems but with large dimension . In that case, the observations y^o although depending on time, are considered as observation of the initial time. This highly simplify the computation of the gradient because the 3D-Var algorithm does not require the model adjoint M^T integration (nor the direct model M integration).



Additional remarks:

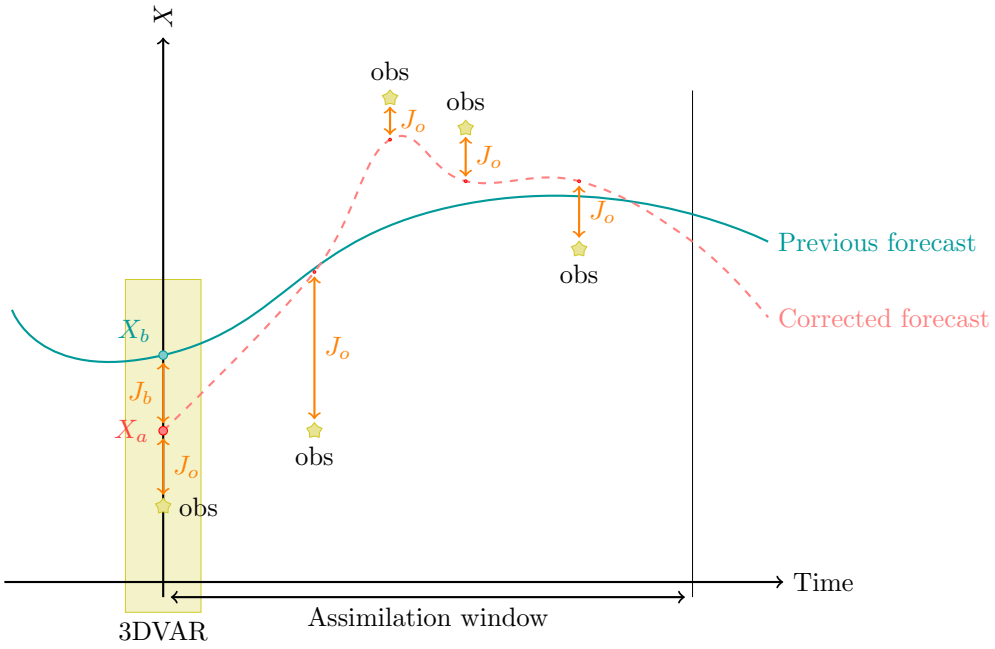
- The 3D-Var is a sequential assimilation algorithm like the Optimal Interpolation and the Kalman Filters, meaning it alternates prediction and correction steps.
- If H is linear, the 3D-Var is strictly equivalent to the BLUE

5.3 4D-Var

5.3.1 Cost function and gradient

In the so-called 4D-Var algorithm, the assumption is now that the model is perfect, meaning $\mathbf{x}$ is fully determined by the initial condition $\mathbf{x}_0$. It is also called strong constraint 4D-Var, meaning that the model is a strong constrain of the minimization process. This algorithm has been made feasible in practice through the introduction of the adjoint methods for data assimilation (Le Dimet, 1982).

The cost function is still relative to the initial condition $\mathbf{x}_0$, but now include the model M since the observation $\mathbf{y}_i^o$ at time i is compared to $H_i(\mathbf{x}_i)$,where $\mathbf{x}_i$ is the state vector at time i.



The cost function is then

$$J(\mathbf{x}_0) = J^b(\mathbf{x}_0) + J^o(\mathbf{x}_0) \tag{51}$$

where the background term J^b is the same as before:

$$J^b(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b)$$

The background $\mathbf{x}_0^b$, as $\mathbf{x}_0$, is one possible state vector at initial time $i = 0$.
The observation term J^o is a bit more complex:

$$J^o(\mathbf{x}_0) = \sum_{i=0}^{n} (\mathbf{y}_i^o - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(\mathbf{x}_i))$$

with

$$\begin{aligned} \mathbf{x}_i &= M_{0\to i}(\mathbf{x}) \\ &= M_{i-1,i}(M_{i-2,i-1}(\dots M_{1,2}(M_{0,1}(\mathbf{x})))) \\ &= M_i(M_{i-1}(\dots(M_2(M_1(\mathbf{x}))))) \end{aligned}$$

The observation term of the 4D-Var cost function can be rewritten relative to $\mathbf{x_0}$:

$$J^o(\mathbf{x}_0) = \sum_{i=0}^{n} \left[\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots M_1(\mathbf{x}_0))))\right]^T \mathbf{R}_i^{-1} \left[\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots M_1(\mathbf{x}_0))))\right]$$

Finally, the gradient of J is given by

$$\nabla J(\mathbf{x}_0) = 2\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - 2\sum_{i=0}^{n} \mathbf{M_1}^T \dots \mathbf{M_{i-1}}^T \mathbf{M_i}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \left[y_i^o - H_i(M_i(M_{i-1}(\dots M_1(\mathbf{x}))))\right]$$

If we define the innovation vector $\mathbf{d}_i$:

$$\mathbf{d}_i = y_i^o - H_i(M_i(M_{i-1}(\dots M_1(\mathbf{x}))))$$

We than get

$$\begin{aligned} -\tfrac{1}{2}\nabla J^o(\mathbf{x}) &= \sum_{i=0}^{n} \mathbf{M}_1^T \dots \mathbf{M}_{i-1}^T \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{d}_i \\ &= \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{d}_0 + \mathbf{M}_1^T \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{d}_1 + \mathbf{M}_1^T \mathbf{M}_2^T \mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{d}_2 + \dots + \\ &\quad \mathbf{M}_1^T \dots \mathbf{M}_{n-1}^T \mathbf{M}_n^T \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{d}_n \\ &= \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{d}_0 + \mathbf{M}_1^T \left[\mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{d}_1 + \mathbf{M}_2^T \left[\mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{d}_2 + \dots + \mathbf{M}_n^T \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{d}_n\right]\right] \end{aligned}$$

Such factorization allows to compute J^o and then ∇J^o thanks to one direct model integration and one adjoint model.

In other words, ∇J^o is the result of:

$$\left\{ \begin{array}{ll} \mathbf{x}_i^* = & \mathbf{M}_i^T \mathbf{x}_{i+1}^* + \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{d}_i, \quad i = n, 1 \\ \mathbf{x}_n^* = & \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{d}_n \end{array} \right. \tag{52}$$

The direct model 49, the adjoint model 52 along with the cost function 51 form the *optimality system.*

5.3.2 Algorithm and remarks

Algorithme 3 (4D-Var)

- *Initialization :* $\mathbf{x} = \mathbf{x}^0$, $n = 0$
- *While* $\|\nabla J\| > \varepsilon$ *or* $n \leq n_{max}$, *do :*
  1. *Compute* J *thanks to the direct model* M *and the observation operator* H
  2. *Compute* ∇J *thanks to the backward integration of the adjoint model* $\mathbf{M}^T$ *and the adjoint of the observation operator* $\mathbf{H}^T$.
  3. *Descent and update of* $\mathbf{x}$
  4. $n = n + 1$

Parameter estimation. If we wish to get an estimate of a parameter set

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_p)$$

We simply add these parameters to the control variables and a corresponding term to the cost function:

$$J(\mathbf{x}, \alpha) = J_1^b(\mathbf{x}) + J_2^b(\alpha) + J^o(\mathbf{x}, \alpha)$$

The observation term shows then an α-dependence and it is frequently necessary to introduce a regularization term for α, like, for instance:

$$J_2^b(\alpha) = \|\alpha - \alpha^b\|^2, \text{ ou } = (\alpha - \alpha^b)^T B_\alpha^{-1} (\alpha - \alpha^b), \text{ ou } = \|\nabla\alpha - \beta\|^2 \dots$$

Why using the adjoint to compute the gradient? Another way would be to go back to the definition of the gradient:

$$\nabla J = (\nabla J_1, \nabla J_2, \ldots, \nabla J_m)^T$$

with

$$\nabla J_i = \lim_{h \to 0} \frac{J(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_j + h, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_n) - J(\mathbf{x}_1, \ldots, \mathbf{x}_n)}{h}$$

But then two problems arise:

1. The result is not exact since we can only get an approximation of the limit.
2. The computation for all $j \in \{1, \ldots, n\}$ is needed.

The adjoint method gives an exact result.

Remarks: Non linearities If M and H are non linear, the computation of the gradient by adjoint methods remains exact, but J becomes non convex.

5.4 Incremental 4D-Var

In the case of M and/or H non linear, in order to avoid the minimization of a non convex function, the incremental 4D-Var algorithm has been implemented. This is an approximation of the 4D-Var, making the so-called *Tangent Linear Hypothesis* (TLH):

$$M_{0\to i}(\mathbf{x}_0) - M_{0\to i}(\mathbf{x}_0^b) \simeq \mathbf{M}_{0\to i}(\mathbf{x}_0 - \mathbf{x}_0^b)$$

and

$$H_i(\mathbf{x}_i) - H_i(\mathbf{x}_i^b) \simeq \mathbf{H}_i(\mathbf{x}_i - \mathbf{x}^b)$$

Defining the *increment* $\delta\mathbf{x}_0$ as

$$\delta\mathbf{x}_0 = \mathbf{x}_0 - \mathbf{x}_0^b$$

we can rewrite the cost function of the 4D-Var as an equivalent function of $\delta\mathbf{x}$:

$$J(\delta\mathbf{x}_0) = \delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \sum_{i=0}^{n}\left[\mathbf{y}_i^o - H_i(M_{0\to i}(\mathbf{x}_0^b + \mathbf{x}_0)))\right]^T \mathbf{R}_i^{-1}\left[\mathbf{y}_i^o - H_i(M_{0\to i}(\mathbf{x}_0^b + \mathbf{x}_0)))\right]$$

If we define the *innovation vector* $\mathbf{d}_i$ as

$$\mathbf{d}_i = \mathbf{y}_i^o - H_i(M_{0\to i}(\mathbf{x}^b))$$

and using the TLH, we can approximate J by :

$$\tilde{J}(\delta\mathbf{x}) = \delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \sum_{i=1}^{n}(\mathbf{d}_i - \mathbf{H}_i\mathbf{M}_i \ldots \mathbf{M}_1\delta\mathbf{x})^T\mathbf{R}_i^{-1}(\mathbf{d}_i - \mathbf{H}_i\mathbf{M}_i \ldots \mathbf{M}_1\delta\mathbf{x})$$

$\tilde{J}(\delta\mathbf{x})$ is now a quadratic function and therefore has a unique minimum.
During the minimization, $\delta\mathbf{x}$ will grow and bcome too large, then the TLH will be invalidated. In order to sort this out, one stops the minimization, updates the linearized operators $\mathbf{M}_i$ and $\mathbf{H}_i$ and the innovation vector $\mathbf{d}$ by recomputing the non linear trajectory starting from the initial condition $\mathbf{x}_0 = \mathbf{x}_0^b + \delta\mathbf{x}_0$.

Algorithme 4 (incremental 4D-Var) – *Initialization :* $\mathbf{x}_0^r = \mathbf{x}_0^b$
($\mathbf{x}^r$ is called reference state ; $\mathbf{x}_0^b$ is the first guess).

Start the outer loop

- *Non linear model integration:* $\mathbf{x}_i^r = M_{0\to i}[\mathbf{x}^r]$
- *Innovation vector computation thanks to the non linear observation operator* $d_i = \mathbf{y}_i^o - H_i(\mathbf{x}_i^r)$

  Start the inner loop

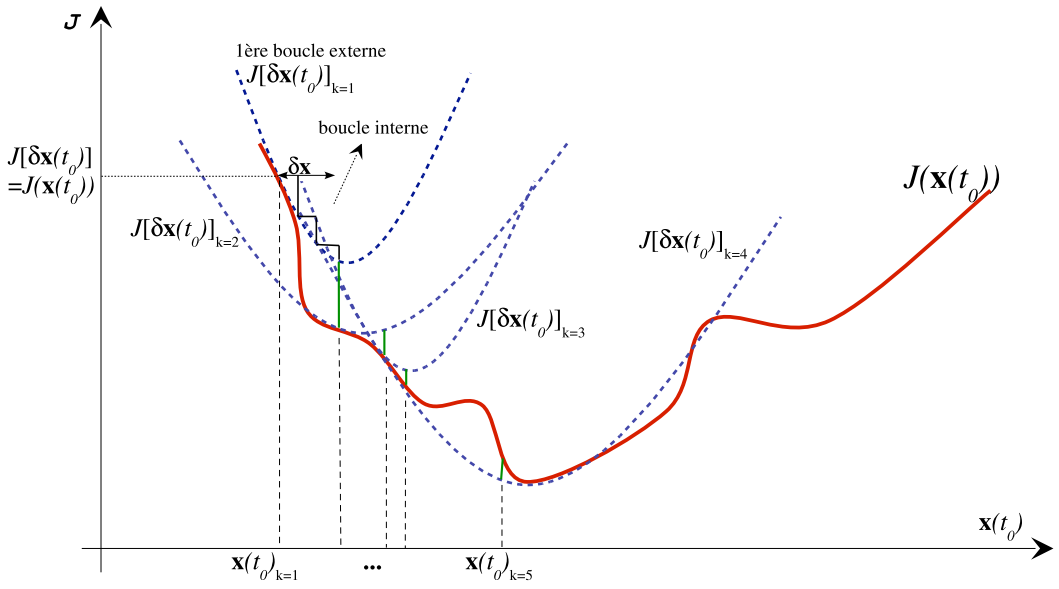
  – *Computation of the incremental cost function* $\tilde{J}^o(\delta\mathbf{x}_0)$ *using* $\mathbf{M}$ *and* $\mathbf{H}$ *the linearized operators around* $\mathbf{x}^r$
  – *Computation of the gradient* $\nabla\tilde{J}^o(\delta\mathbf{x}_0)$ *thanks to the adjoint operators* $\mathbf{M}^T$ *et* $\mathbf{H}^T$
  – *Minimization via a descent method*

  End of the inner loop

- *Update the analysis increment* $\delta\mathbf{x}_0^a = \delta\mathbf{x}_0$
- *Update the reference state* $\mathbf{x}_0^r = \mathbf{x}_0^r + \delta\mathbf{x}_0^a$

End of the outer loop

– *Compute the final analysis:* $\mathbf{x}_0^a = \mathbf{x}_0^r$, $\mathbf{x}_i^a = \mathbf{M}_{0,i}[\mathbf{x}_0^a]$.



(from YAO user's guide)

5.5 3D-FGAT

The 3D-FGAT (First Guess at Appropriate Time) is a further approximation of the incremental 4D-Var algorithm where the evolution of the increment during the assimilation window is assume to be stationnary, *i.e.*:

$$M_{0\to i}(\mathbf{x}_0^b + \delta\mathbf{x}_0) - M_{0\to i}(\mathbf{x}_0^b) \simeq \delta\mathbf{x}_0$$

In other words it assumes that $\mathbf{M} = \mathbf{I}$ and $\mathbf{M}^T = \mathbf{I}$ for the length of the assimilation window.

Algorithme 5 (3D-FGAT) – *Initialization :* $\mathbf{x}_0^r = \mathbf{x}_0^b$
($\mathbf{x}^r$ *is called reference state ;* $\mathbf{x}_0^b$ *is the first guess).*

START THE OUTER LOOP

- *Non linear model integration:* $\mathbf{x}_i^r = M_{0\to i}[\mathbf{x}^r]$ *;*
  - *compute the* $\mathbf{d}_i = \mathbf{y}_i^o - H_i(\mathbf{x}_i^r)$
  - *store the non linear trajectory* $\mathbf{x}_i^r$ *for the tangent and adjoint Observation operator (*if required)

  START THE INNER LOOP

  - *Linear model integration:*$\delta\mathbf{x}_i = \mathbf{M}_{0\to i}\delta\mathbf{x}$
    - *compute* $\mathbf{d}_i^o - \mathbf{H}_i\delta\mathbf{x}_0$
  - $\nabla J = -\sum_i \mathbf{H}_i{}^T[\mathbf{d}_i^o - \mathbf{H}_i\delta\mathbf{x}_0]$
  - *update* $\delta\mathbf{x}_0$ *thanks to the descent algorithm*

    END OF THE INNER LOOP

  - *Update the reference state* $\mathbf{x}_0^r = \mathbf{x}_0^r + \delta\mathbf{x}_0^a$

  END OF THE OUTER LOOP

  – *Compute the final analysis:* $\mathbf{x}_0^a = \mathbf{x}_0^r$, $\mathbf{x}_i^a = \mathbf{M}_{0,i}[\mathbf{x}_0^a]$.

Remarks: As 3D-Var, this algorithm does not require the adjoint model M^T. However, it is more satisfactory than 3D-Var, for the following reasons:

1. innovation vectors are exact, the observation misfit is computed at the appropriate time (therefore the name FGAT: First Guess at Appropriate Time) ;
2. the algorithm structure is the same as 4D-Var's, so that 3D-FGAT can be used in development phase, to do first experiments while waiting for adjoint implementation.

5.6 Practical adjoint implementation

Adjoint code construction can be a tedious and difficult task, which we will not cover here. For details, please contact us or refer to:

- Giering and Kaminski, Recipes for adjoint code construction, ACM Trans. On Math. Software, 1998, Volume 24, Issue 4, 437–474.
- Automatic Differentiation software TAPENADE, http://www-sop.inria.fr/tropics.

References

General DA

[1] A.F. Bennett. *Inverse modeling of the Ocean an Atmosphere*. Cambridge University Press, Cambridge, 2002.

[2] F. Bouttier and P. Courtier. Data assimilation concepts and methods. *Meteorological Training Course Lecture Series*, 1999.

[3] M. Ghil and P. Manalotte-Rizzoli. Data assimilation in meteorology and oceanography. *Adv. Geophys.*, 23:141–265, 1991.

[4] K. Ide, P. Courtier, M. Ghil, and Lorenc A. C. Unified notation for data assimilation : operational, sequential and variational. *Journal of Meteorological Society of Japan*, 75:181–189, 1997.

[5] A. Weaver and P. Courtier. Correlation modelling on the sphere using a generalized diffusion equation. *Quarterly Journal of the Royal Meteorological Society*, 127:1815–1846, July 2001.

Variational DA

[1] P. Courtier, J. N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120:1367–1387, July 1994.

[2] R. Giering and T. Kaminski. Recipes for Adjoint Code Construction. *ACM Trans. On Math. Software*, 24(4):437–474, 1998.

[3] Jean Charles Gilbert and Claude Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Programming*, 45(3, (Ser. B)):407–435, 1989.

[4] Laurent Hascoët. Tapenade: a tool for automatic differentiation of programs. In *Proceedings of 4th European Congress on Computational Methods, ECCOMAS'2004, Jyvaskyla, Finland*, 2004.

[5] F.-X. Le Dimet, I. Navon, and D. Daescu. Second order information in data assimilation. *Mont. Wea. Rev.*, 130(629–648), 2002.

[6] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus Series A*, 38:97–+, 1986.

Stochastic DA

[1] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[2] A. H. Jazwinski. *Stochastic processes and filtering theory*, volume 64 of *Applied Mathematical Sciences*. Academic Press, 1970.

[3] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Physical Oceanography*, 23:2541–2566, 1960.

[4] D. T. Pham, J. Verron, and M. C. Roubaud. A Singular Evolutive Extended Kalman filter for data assimilation in oceanography. *Inverse Problems*, 14:979–997, 1998.

[5] Peter Jan Van Leeuwen. Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12), 2009.

Simple (nudging-based) methods

[1] D. Auroux and J. Blum. Back and forth nudging algorithm for data assimilation problems. *Comptes Rendus de l'Académie des Sciences*, 340(12):873–878, 2005.

[2] J. Verron. Nudging Satellite Altimeter Data Into Quasi-Geostrophic Ocean Models. *J. Geophys. Res.*, 97:7479–7491, May 1992.

Contact

Eric.Blayo@imag.fr
http://ljk.imag.fr/membres/Eric.Blayo

Emmanuel.Cosme@hmg.inpg.fr
http://www-meom.hmg.inpg.fr/Web/pages-perso/Cosme

Maelle.Nodet@inria.fr
http://ljk.imag.fr/membres/Maelle.Nodet

Arthur.Vidard@imag.fr
http://ljk.imag.fr/membres/Arthur.Vidard