Data Assimilation

Mark Asch

LAMFA, UMR CNRS 7352, Université de Picardie Jules Verne, France.

Environmental Hydroacoustics Lab, Université libre de Bruxelles, Belgium.

CIMPA-UNESCO School

Caracas, Venezuela

16-27 April 2012

Contents

CHAPTER 1

Introduction to Data Assimilation

Data assimilation is the "art" of compromise... and has been defined in different ways by different researchers : "interpolating fields for subsequent use as initial data in a model integration" (Bennett, 2002); "statistical combination of observations and short-range forecasts" (Kalnay, 2003); "using all the available information, to define as accurate as possible the state" (Talagrand, 1997). From a statistical perspective DA can be see as fusing data (observations) with prior knowledge (e.g., physical laws; model output) to get an estimate of the (distribution of) the true state of the physical system. This requires a statistical model for the observations (data model; direct or indirect) and a (prior) statistical model for the system (process model) which can be either a deterministic model (with additive errors) or a long term history of observations (e.g., climatology).

1.1. Acknowledgements

These course notes have used, quite liberally, the excellent notes of my good colleagues:

- Eric Blayo, Emmanuel Cosme and Maëlle Nodet (U. of Grenoble, France).
- Alexandre Fournier (U. Paris VI, France).
- Marc Bocquet (ENPC, France).

However, I take full responsibility for the actual contents.

1.2. History

Data assimilation was introduced by the meteorologist Lewis Fry Richardson in 1922. He proposed to use a big hall full of humans working as computers, each one performing elmentary computations... Starting in the 1950's, numerical forecasts with computers became feasible. The first weather forecasts derived this way used barotropic (that means, single-vertical-level) models, and could successfully predict the large-scale movement of midlatitude Rossby waves, that is, the pattern of atmospheric lows and highs. In the 1960's, the chaotic nature of the atmosphere was first observed and mathematically described by Edward Lorenz, founding the field of chaos theory. These advances have led to the current use of ensemble forecasting in most major forecasting centers, to take into account uncertainty arising from the chaotic nature of the atmosphere. The variational approaches were introduced in the 1980's and the 4D-Var method, that is still used today, was introduced in the early 2000's.

Climate models have been developed that feature a resolution comparable to older weather prediction models. These climate models are used to investigate long-term climate shifts, such as what effects might be caused by human emission of greenhouse gases.

1.3. Standard notation

A standard set of notation has been proposed by Ide et al. in [9]. A discrete model for the evolution of a physical (atmospheric, oceanic, etc.) system from time t_k to time t_{k+1} is described by an equation

$$\mathbf{x}^f(t_{k+1}) = M\left[\mathbf{x}^f(t_k),\right] \tag{1.3.1}$$

where $\mathbf{x}$ is the model's state vector if dimension n, and M is the corresponding dynamics operator, usually obtained by finite difference or finite element discretization. The error covariance matrix associated with $\mathbf{x}$ is given by $\mathbf{P}$ since the true state will differ from the simulated state (1.3.1) by random or systematic errors.

Observations, or measurements, at time t_k are defined by

$$\mathbf{y}_k^o = H_k\left[\mathbf{x}^t(t_k)\right] + \varepsilon_k,$$

where H is an observation operator and ε is a noise process. The observation vector $\mathbf{y}_k^o = \mathbf{y}^o(t_k)$ has dimension p_k. A major problem of data assimilation is that usually $p_k \ll n$. The noise process ε is assumed to have zero mean and covarinace matrix $\mathbf{R}$. It is made up of instrument errors and representation errors (due to the discretization).

Subscripts are used to denote the discrete time index, the corresponding spatial indices or the vector with respect to which an error covarinace matrix is defined. Superscripts refer to the nature of the vectors/matrices in the data assimilation process : "a" for analysis, "b" for background (or 'initial guess'), "f" for forecast, "o" for observation and "t" for the (unknown) true state.

1.4. Data assimilation methods

There are two major classes of methods:

(1) Statistical methods where we compute the best linear unbiased estimate (BLUE) by algebraic computations using the Kalman filter.
(2) Variational methods where we explicitly minimize a cost function using optimization methods.

They provide the same result in the linear case, which is the only context where their optimality can be rigourously proved. They both have difficulties in dealing with non-linearities and large problems. In addition, the error statistics that are required by both, are in general poorly known.

CHAPTER 2

Statistical estimation and Sequential DA

2.1. Statistical estimation theory

Practical inverse problems and data assimilation problems involve measured data. These data are inexact and are mixed with random noise. Statistical models can provide rigorous, effective means for dealing with this measurement error.

2.1.1. Preliminary definitions and notation. In statistical modeling, the concepts of sample space, probability and random variable play key roles. A sample space $\mathcal{S}$ is the set of all possible outcomes of a random, unpredictable experiment. Probability provides a means for quantifying how likely it is for an outcome to take place. Random variables assign numerical values to outcomes in the sample space. Once this has been done, we can systematically work with notions such as average value, or mean, or variability.

It is customary in mathematical statistics to use capital letters to denote random variables (r.v.) and correpsponding lowercase letters to denote values taken by the r.v. in its range. If $X : \mathcal{S} \to \mathbb{R}$ is a r.v., then for any $x \in \mathbb{R}$, by $\{X \leq x\}$ we mean $\{s \in \mathcal{S} \mid X(s) \leq x\}$.

Definition 1. A *probability space* $(\mathcal{S}, \mathcal{B}, \mathcal{P})$ consists of a set $\mathcal{S}$ called the *sample space*, a collection $\mathcal{B}$ of (Borel) subsets of $\mathcal{S}$ and a *probability function* $\mathcal{P} : \mathcal{B} \to \mathbb{R}_+$ for which

- $\mathcal{P}(\emptyset) = 0$,
- $\mathcal{P}(\mathcal{S}) = 1$ and
- $\mathcal{P}\left(\bigcup_i S_i\right) = \sum_i \mathcal{P}(S_i)$ for any disjoint, countable collection of sets $S_i \in \mathcal{B}$.

A *random variable* X is a measurable function $X : \mathcal{S} \to \mathbb{R}$. Associated with the r.v. X is its *distribution function*,

$$F_X(x) = \mathcal{P}\{X \leq x\}, \quad x \in \mathbb{R}.$$

The distribution function is non-decreasing, right continuous and satisfies

$$\lim_{x \to -\infty} F_X(x) = 0, \qquad \lim_{x \to +\infty} F_X(x) = 1.$$

Definition 2. A random variable X is called *discrete* if there exist countable sets $\{x_i\} \subset \mathbb{R}$ and $\{p_i\} \subset \mathbb{R}_+$ for which

$$p_i = \mathcal{P}\{X = x_i\} > 0$$

for each i, and

$$\sum_i p_i = 1.$$

In this case, the *probability density function* for X is the real-valued function with discrete support

$$p_X(x) = \begin{cases} p_i & \text{if } x = x_i, \quad i = 1, 2, \dots \\ 0 & \text{otherwise.} \end{cases}$$

The x_i's are the points of discontinuity of the distribution function,

$$F_X(x) = \sum_{\{i | x_i \leq x\}} p_X(x_i).$$

DEFINITION 3. A random variable X is called *continuous* if its distribution function F_X is absolutely continuous. In this case,

$$F_X(x) = \int_{-\infty}^{x} p_X(u)\, du$$

and the derivative,

$$p_X(x) = \frac{dF_X}{dx}$$

is called the *probability density function* for X.

DEFINITION 4. The *mean*, or *expected value*, of a r.v. X is given by the Riemann-Stieltjes integral

$$E(X) = \int_{-\infty}^{\infty} x\, dF_X(x).$$

If X is a continuous random variable, then

$$dF_X(x) = p_X(x)\, dx$$

and in the discrete case,

$$dF_X(x) = p_X(x_i)\delta(x - x_i).$$

In the latter case,

$$E(X) = \sum_i x_i p_X(x_i).$$

The *expectation operator*, E, is a linear operator.

DEFINITION 5. The *variance* of a r.v. X is given by

$$\sigma^2 = E\left[(X - \mu)^2\right] = E(X^2) - (E(X))^2,$$

where

$$\mu = E(X).$$

DEFINITION 6. Two random variables, X and Y, are *jointly distributed* if they are both defined on the same probability space $(\mathcal{S}, \mathcal{B}, \mathcal{P})$.

DEFINITION 7. A *random vector*, $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, is a mapping from $\mathcal{S}$ into $\mathbb{R}^n$ for which all the components X_i are jointly distributed. The *joint distribution function* of $\mathbf{X}$ is given by

$$F_{\mathbf{X}}(\mathbf{x}) = \mathcal{P}\{X_1 \leq x_1, \ldots, X_n \leq x_n\}, \quad \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n.$$

The components X_i are *independent* if the joint distribution function is the product of the distribution functions of the components,

$$F_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} F_{X_i}(x_i).$$

DEFINITION 8. A random vector $\mathbf{X}$ is *continuous* with *joint probability density function* $p_{\mathbf{X}}$ if

$$F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_1} p_{\mathbf{X}}(\mathbf{u})\, du_1 \ldots du_n.$$

DEFINITION 9. The mean, or *expected value*, of a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, is the n-vector $E(\mathbf{X})$ with components

$$[E(\mathbf{X})]_i = E(X_i), \quad i = 1, \ldots, n.$$

The *covariance* of $\mathbf{X}$ is the $n \times n$ matrix $\mathrm{cov}(\mathbf{X})$ with components

$$[\mathrm{cov}(\mathbf{X})]_{ij} = E\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \sigma_{ij}, \quad 1 \leq i, j \leq n,$$

where

$$\mu_i = E(X_i).$$

2.1.2. Gaussian distributions. A continuous random vector, $\mathbf{X}$ has a Gaussian distribution if its joint probability density function has the form

$$p_{\mathbf{X}}(\mathbf{x}; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right),$$

where $\mathbf{x}, \mu \in \mathbb{R}^n$, Σ is an $n \times n$ symmetric positive definite matrix. The mean is given by

$$E(\mathbf{X}) = \mu$$

and the covariance matrix

$$\mathrm{cov}(\mathbf{X}) = \Sigma.$$

These two parameters completely characterize the distribution and we indicate this situation by

$$\mathbf{X} \sim \mathcal{N}(\mu, \Sigma).$$

Note, that in the scalar case, we have the familiar "bell curve"

$$p(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}.$$

2.1.3. Maximum likelihood estimation (MLE). Suppose a random vector $\mathbf{X}$ has a joint probability density function $p_{\mathbf{X}}(\mathbf{x};\theta)$, where θ is an unknown parameter vector that we would like to estimate. Suppose also, that we have a data vector $\mathbf{d} = (d_1, \ldots, d_n)$, a given realization of $\mathbf{X}$ (an outcome of a random experiment).

DEFINITION 10. A maximum likelihood estimator for θ given $\mathbf{d}$, is a parameter vector $\hat{\theta}$ that maximizes the likelihood function,

$$L(\theta) = p_{\mathbf{X}}(\mathbf{x};\theta),$$

which is the joint p.d.f. considered as a function of θ. The MLE is also a maximizer of the log-likelihood function,

$$l(\theta) = \log p_{\mathbf{X}}(\mathbf{x};\theta).$$

2.1.4. Bayesian estimation. We first discuss conditional probability and conditional expectation. Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$ be jointly distributed discrete random vectors. Then $(\mathbf{X}, \mathbf{Y})$ is also a discrete random vector.

DEFINITION 11. The joint probability density function for $(\mathbf{X}, \mathbf{Y})$ is given by

$$p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y}) = \mathcal{P}\{\mathbf{X} = \mathbf{x}, \quad \mathbf{Y} = \mathbf{y}\}, \quad (\mathbf{x},\mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n.$$

The *marginal probability density function* of $\mathbf{X}$ is then defined as

$$p_{\mathbf{X}}(\mathbf{x}) = \sum_{\mathcal{P}\{\mathbf{Y}=\mathbf{y}\}>0} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y}), \quad \mathbf{x} \in \mathbb{R}^n. \tag{2.1.1}$$

The *conditional probability density function* for $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ is then defined as

$$p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x}) = \frac{p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y})}{p_{\mathbf{X}}(\mathbf{x})} \tag{2.1.2}$$

where the denominator is nonzero.

REMARK 12. If $\mathbf{X}$ and $\mathbf{Y}$ are independent random vectors, then the conditional density function of $\mathbf{Y}$ given $\mathbf{X} = \mathbf{x}$ does not depend on $\mathbf{x}$ and it satisfies

$$p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x}) = p_{\mathbf{Y}}(\mathbf{y}). \tag{2.1.3}$$

DEFINITION 13. Let $\phi : \mathbb{R}^n \to \mathbb{R}^k$ be a measurable mapping. The conditional expectation of $\phi(\mathbf{Y})$ given $\mathbf{X} = \mathbf{x}$ is

$$E(\phi(\mathbf{Y}) \mid \mathbf{X} = \mathbf{x}) = \sum_{\mathcal{P}\{\mathbf{Y}=\mathbf{y}\}>0} \phi(\mathbf{y}) p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n. \tag{2.1.4}$$

REMARK 14. For continuous random vectors $\mathbf{X}$ and $\mathbf{Y}$, we can define the analogous concepts by replacing the summations in (2.1.1)-(2.1.4) with appropriate integrals.

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y})\, dF_{\mathbf{Y}}(\mathbf{y}),$$

$$E\left(\phi(\mathbf{Y}) \mid \mathbf{X} = \mathbf{x}\right) = \int_{-\infty}^{\infty} \phi(\mathbf{y}) p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x})\, dF_{\mathbf{Y}}(\mathbf{y}).$$

We are now ready to state Bayes' Law that relates the conditional random vector $\mathbf{X}|_{\mathbf{Y}=\mathbf{y}}$ to the inverse conditional random vector, $\mathbf{Y}|_{\mathbf{X}=\mathbf{x}}$.

THEOREM 15. *Let $\mathbf{X}$ and $\mathbf{Y}$ be jointly distributed random vectors. Then*

$$p_{(\mathbf{X}|\mathbf{Y})}(\mathbf{x} \mid \mathbf{y}) = \frac{p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x}) p_{\mathbf{X}}(\mathbf{x})}{p_{\mathbf{Y}}(\mathbf{y})}. \tag{2.1.5}$$

DEFINITION 16. In the context of Bayes' Law (2.1.5), suppose that $\mathbf{X}$ represents the variable of interest and that $\mathbf{Y}$ represents an observable (measured) quantity that depends on $\mathbf{X}$. Then:

- $p_{\mathbf{X}}(\mathbf{x})$ is called the *a priori* probability density function, or the *prior*,
- $p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y} \mid \mathbf{x})$ is called the *a posteriori* probability density function,
- the denominator, $p_{\mathbf{Y}}(\mathbf{y})$, can be considered as a normalization factor.

2.1.5. Linear least squares estimation (LLSE): Best linear unbiased estimation (BLUE), Minimum variance linear estimation (MVLE). Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_n)$ be jointly distributed, real-valued random vectors with finite expected squared components,

$$E(X_i^2) < \infty, \quad i = 1, \ldots, n, \qquad E(Z_j^2) < \infty, \quad j = 1, \ldots, m.$$

DEFINITION 17. The *cross-correlation* matrix for $\mathbf{X}$ and $\mathbf{Z}$ is the $n \times m$ matrix $\Gamma_{\mathbf{XZ}} = E(\mathbf{X}\mathbf{Z}^T)$ with entries

$$[\Gamma_{\mathbf{XZ}}]_{ij} = E(X_i Z_j), \quad i = 1, \ldots, n,\ j = 1, \ldots, m.$$

The *autocorrelation* matrix for $\mathbf{X}$ is $\Gamma_{\mathbf{XX}} = E(\mathbf{X}\mathbf{X}^T)$ with entries

$$[\Gamma_{\mathbf{XX}}]_{ij} = E(X_i X_j), \quad 1 \le i, j \le n.$$

REMARK 18. Note that $\Gamma_{\mathbf{ZX}} = \Gamma_{\mathbf{XZ}}^T$ and that $\Gamma_{\mathbf{XX}}$ is symmetric and positive semi-definite. Also, if $E(\mathbf{X}) = 0$, then the auto-correlation reduces to the covariance, $\Gamma_{\mathbf{XX}} = \mathrm{cov}(\mathbf{X})$.

We can relate the trace of the autocorrelation matrix to the second moment of the random vector $\mathbf{X}$.

PROPOSITION 19. *If a random vector $\mathbf{X}$ has finite expected squared components, then*

$$E\left(\|\mathbf{X}\|^2\right) = \mathrm{trace}(\Gamma_{\mathbf{XX}}).$$

For the BLUE, we consider a linear model,

$$\mathbf{Z} = K\mathbf{x} + \mathbf{N},$$

where K is an $m \times n$ matrix, $\mathbf{x} \in \mathbb{R}^n$ is deterministic and $\mathbf{N}$ is a random (noise) n-vector with

$$E(\mathbf{N}) = 0, \quad C_{\mathbf{N}} = \mathrm{cov}(\mathbf{N})$$

and $C_{\mathbf{N}}$is a known, non-singular, $n \times n$ covariance matrix.

DEFINITION 20. The *best linear unbiased estimator* for $\mathbf{x}$ from $\mathbf{Z}$ is the vector $\hat{\mathbf{X}}_{\mathrm{BLUE}}$ that minimizes the quadratic cost function

$$J(\hat{\mathbf{X}}) = E\left(\left\|\hat{\mathbf{X}} - \mathbf{x}\right\|^2\right)$$

subject to the constraints of linearity

$$\hat{\mathbf{X}} = B\mathbf{Z}, \quad B \in \mathbb{R}^{n\times m},$$

and *unbiased*ness

$$E(\hat{\mathbf{X}}) = \mathbf{x}.$$

In the case of a full rank matrix K, the *Gauss-Markov Theorem* gives us an explicit form for the BLUE.

THEOREM 21. *If* K *has full rank, then the best linear unbiased estimator is given by*

$$\hat{\mathbf{X}}_{\mathrm{BLUE}} = \hat{B}\mathbf{Z},$$

where

$$\hat{B} = \left(K^T C_{\mathbf{N}}^{-1} K\right)^{-1} K^T C_{\mathbf{N}}^{-1}.$$

REMARK 22. If the noise covariance matrix $C_{\mathbf{N}} = \sigma^2 I$ (white, uncorrelated noise) and K has full rank, then

$$\hat{\mathbf{X}}_{\mathrm{BLUE}} = \left(K^T K\right)^{-1} K^T \mathbf{Z} = K^\dagger \mathbf{Z}.$$

This corresponds, in the deterministic case, to the *least squares problem*

$$\min_{\mathbf{x}} \|K\mathbf{x} - \mathbf{z}\|.$$

Due to the dependence of the BLUE on the inverse of the noise covariance matrix, it is unsuitable for the solution of noisy, ill-conditioned linear systems. To remedy this situation, we assume that $\mathbf{x}$ is a realization of a random vector $\mathbf{X}$ and we formulate a linear least-squares analogue of Bayesian estimation.

DEFINITION 23. Suppose that $\mathbf{X}$ and $\mathbf{Z}$ are jointly distributed, random vectors with finite expected squares. The *minimum variance linear estimator* of $\mathbf{X}$ from $\mathbf{Z}$ is given by

$$\hat{\mathbf{X}}_{\mathrm{MVLE}} = \hat{B}\mathbf{Z},$$

where

$$\hat{B} = \arg\min_{B\in\mathbb{R}^{n\times m}} E\left(\|B\mathbf{Z} - \mathbf{X}\|^2\right).$$

PROPOSITION 24. *If $\Gamma_{\mathbf{ZZ}}$ is non-singular, then the MVLE of $\mathbf{X}$ from $\mathbf{Z}$ is given by*

$$\hat{\mathbf{X}}_{\mathrm{MVLE}} = \left(\Gamma_{\mathbf{XZ}}\Gamma_{\mathbf{ZZ}}^{-1}\right)\mathbf{Z}.$$

2.1.6. Summary. The two important formulas from this section, that should be particularly recalled, are the following.

(1) Bayes' theorem:

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}. \tag{2.1.6}$$

(2) The marginalization rule:

$$p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y})\, d\mathbf{x} = \int p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x})\, d\mathbf{x}, \tag{2.1.7}$$

where

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

2.2. Sequential Data Assimilation and Kalman filters

Within the significant toolbox of mathematical tools that can be used for statistical estimation from noisy sensor measurements, one of the most well-known and often-used tools is the Kalman filter. The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem (Kalman 1960). A very "friendly" introduction to the general idea of the Kalman filter is offered in Chapter 1 of (Maybeck 1979)—which we have included in this course pack.

We now consider a dynamical system that evolves in time and we seek to estimate a series of true states, $\mathbf{x}_k^t$ (a sequence of random vectors) where discrete time is indexed by the letter k. These times are those when the observations or measurements are taken see Figure 2.2.1. The assimilation starts with an unconstrained model trajectory from $t_0, t_1, \ldots, t_{k-1}, t_k, \ldots, t_n$ and aims to provide an optimal fit to the available observations/measurements given their uncertainties (error bars). In current weather forecasts, $t_k - t_{k-1} = 6$ hours.

2.2.1. Bayesian modeling. Let us recall the principles of Bayesian modeling from the previous section on statistical estimation and write them in the terminology of the assimilation problem. We have a vector $\mathbf{X}$ of (unknown) unobserved quantities of interest (temperature, presssure, wind, etc.) and a vector $\mathbf{Y}$ of (known) observed data (at various locations, and at various times). The full, joint probability model can always be factored into components:

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) &= p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x}) \\ &= p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y}) \end{aligned}$$

and thus

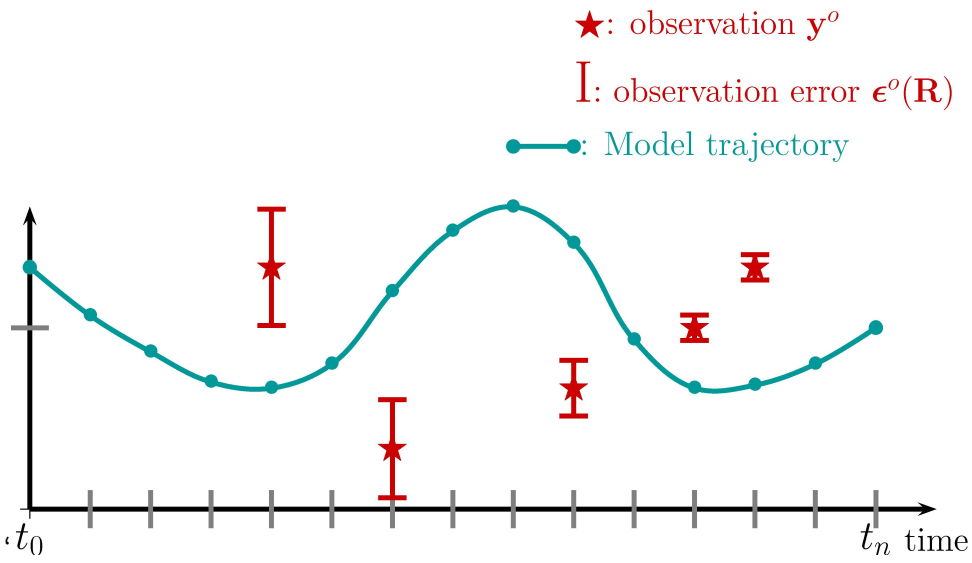

FIGURE 2.2.1. Sequential assimilation.(Credit: A. Fournier)

$$p(\mathbf{x} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}$$

provided that $p(\mathbf{y}) \neq 0$.

The Kalman filter can be rigourously derived from this Bayesian perspective.

2.2.2. Stochastic model of the system. We seek to estimate the state $\mathbf{x} \in \mathbb{R}^n$ of a discrete-time dynamic process that is governed by the linear stochastic difference equation

$$\mathbf{x}_{k+1} = M_{k+1}[\mathbf{x}_k] + \mathbf{w}_k$$

with a measurement/observation $\mathbf{y} \in \mathbb{R}^m$

$$\mathbf{y}_k = H_k[\mathbf{x}_k] + \mathbf{v}_k.$$

The random vectors, $\mathbf{w}_k$ and $\mathbf{v}_k$, represent the process/modeling and measurement/observation errors respectively. They are assumed to be independent, white and with Gaussian/normal probability distributions

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, Q) \\ \mathbf{v}_k &\sim \mathcal{N}(0, R), \end{aligned}$$

where Q and R are the covariance matrices (supposed known).

2.2.3. Sequential assimilation scheme. The typical assimilation scheme is made up of two major steps: a prediction/forecast step and a correction/analysis step. At time t_k we have the result of a previous forecast, $\mathbf{x}_k^f$, (the analogue of the baclground state $\mathbf{x}_k^b$) and the result of an ensemble of observations in $\mathbf{y}_k$. Based on these two vectors, we perform an analysis that produces $\mathbf{x}_k^a$. We then use the evolution model to obtain a prediction of the state at time t_{k+1}. The result of the

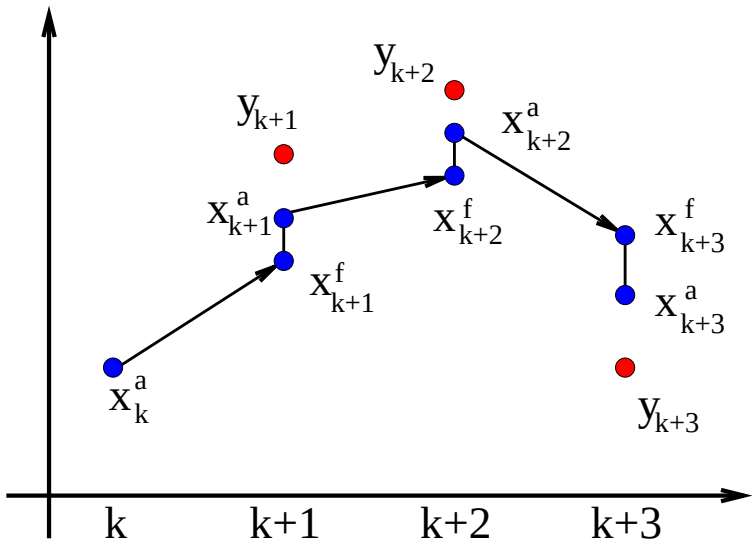

FIGURE 2.2.2. Sequential assimilation scheme for the Kalman filter. (Credit: M. Bocquet)

forecast is denoted $\mathbf{x}^f_{k+1}$, and becomes the bakground (or initial guess) for the next time-step. This process is summarized in Figure 2.2.2.

We can now define forecast (*a priori*) and analysis (*a posteriori*) estimate errors as

$$\begin{aligned} \mathbf{e}^f_k &= \mathbf{x}^f_k - \mathbf{x}^t_k \\ \mathbf{e}^a_k &= \mathbf{x}^a_k - \mathbf{x}^t_k \end{aligned}$$

with their respective error covariance matrices

$$\begin{aligned} P^f_k &= \mathrm{cov}(\mathbf{e}^f_k) = E\left[\mathbf{e}^f_k (\mathbf{e}^f_k)^T\right] \\ P^a_k &= \mathrm{cov}(\mathbf{e}^a_k) = E\left[\mathbf{e}^a_k (\mathbf{e}^a_k)^T\right]. \end{aligned} \tag{2.2.1}$$

The goal of the Kalman filter is to compute an optimal *a posteriori* estimate $\mathbf{x}^a_k$ that is a linear combination of an *a priori* estimate $\mathbf{x}^f_k$ and a weighted difference between the actual measurement $\mathbf{y}_k$ and the measurement prediction $H_k\left[\mathbf{x}^f_k\right]$. This is none other than the BLUE that we have seen above. The filter is thus of the form

$$\mathbf{x}^a_k = \mathbf{x}^f_k + K_k\left(\mathbf{y}_k - H_k \mathbf{x}^f_k\right). \tag{2.2.2}$$

The difference $\left(\mathbf{y}_k - H_k \mathbf{x}^f_k\right)$ is called the *innovation* and reflects the discrepancy between the the actual and the predicted measurements at time t_k. Note that for generality, the matrices are shown with a time-dependence. Often this is not the case, and the subscripts k can then be dropped. The *Kalman gain* matrix, K, is chosen to minimize the *a posteriori* error covariance equation (2.2.1). This is

straightforward to compute: substitute (2.2.2) into the definition of $\mathbf{e}_k^a$, then substitute in the error covariance equation (2.2.1) performing the indicated expectations, take the derivative of the trace of the result with respect to K, set the result equal to zero, and finally solve for the optimal gain K. The resulting K that minimizes equation (2.2.1) is given by (there are various possible forms...)

$$K_k = P_k^f H^T \left(HP_k^f H^T + R\right)^{-1}.$$

Looking at this equation, we see that as the measurement error covariaace R approaches zero, the gain K weights the innovation more heavily, since

$$\lim_{R \to 0} K_k = H^{-1}.$$

On the other hand, as the *a priori* error estimate covariance P_k^f approaches zero, the gain K weights the innovation less heavily, and

$$\lim_{P_k^f \to 0} K_k = 0.$$

Another way of thinking about the weighting of K is that as the measurement error covariance R approaches zero, the actual measurement $\mathbf{y}_k$ is "trusted" more and more, while the predicted measurement $H_k \mathbf{x}_k^f$ is trusted less and less. On the other hand, as the *a priori* error estimate covariance P_k^f approaches zero, the actual mesurement $\mathbf{y}_k$ is trusted less and less, while the predicted measurement $H_k \mathbf{x}_k^f$ is trusted more and more - see the computational example below.

2.2.3.1. *Predictor/forecast step.* We start from a previous analyzed state, $\mathbf{x}_k^a$, or from the initial state if $k = 0$, characterized by the Gaussian p.d.f. $p(\mathbf{x}_k^a \mid \mathbf{y}_{1:k}^o)$ of mean $\mathbf{x}_k^a$ and covariance matrix P_k^a. An estimate of $\mathbf{x}_{k+1}^t$ is given by the dynamical model which defines the forecast:

$$\mathbf{x}_{k+1}^f = M_{k+1}\left[\mathbf{x}_k^a\right] \tag{2.2.3}$$

$$P_{k+1}^f = M_{k+1} P_k^a M_{k+1}^T + Q_{k+1}. \tag{2.2.4}$$

2.2.3.2. *Corrector/analysis step.* At time t_{k+1},the pdf $p(\mathbf{x}_{k+1}^f \mid \mathbf{y}_{1:k}^o)$ is known thanks to the mean $\mathbf{x}_{k+1}^f$ and covariance matrix P_{k+1}^f as well as the assumption of a Gaussian distribution. The analysis step then consistes of correcting this pdf using the observation available at time t_{k+1} in order to compute $p(\mathbf{x}_{k+1}^a \mid \mathbf{y}_{1:k+1}^o)$.This comes from the BLUE in the dyanmical context:

$$K_{k+1} = P_{k+1}^f H^T \left(HP_{k+1}^f H^T + R_{k+1}\right)^{-1}, \tag{2.2.5}$$

$$\mathbf{x}_{k+1}^a = \mathbf{x}_{k+1}^f + K_{k+1}\left(\mathbf{y}_{k+1} - H\mathbf{x}_{k+1}^f\right), \tag{2.2.6}$$

$$P_{k+1}^a = \left(I - K_{k+1} H\right) P_{k+1}^f. \tag{2.2.7}$$

The predictor-corrector loop is illustrated in Figure 2.2.3.

2.2.4. Note on relation between Bayes and BLUE. If we know that the *a priori* and the observation data are both Gaussian, Bayes' rule can be applied to compute the *a posteriori* pdf. The *a posteriori* pdf is then Gaussian, and its parameters are given by the BLUE equations. Hence with Gaussian pdf's and a 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.

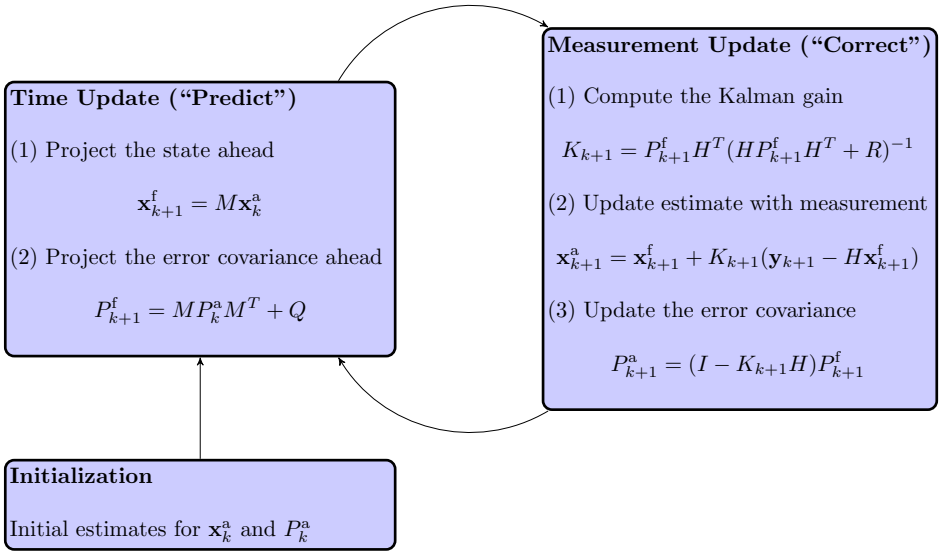

FIGURE 2.2.3. Kalman filter loop.

Since the BLUE provides the same result as Bayes' rule, it is the best estimator of all.

In addition (see the next Chapter) one can recognize the 3D-Var cost function. By minimizing this cost function, 3D-Var finds the MAP estimate of the Gaussian pdf, which is equivalent to the MV estimate found by the BLUE.

2.2.5. Implementation of the Kalman filter. We describe three important implementation issues and discuss ways to overcome the difficulties that they give rise to.

(1) **Definition of covariance matrices and filter divergence.** If the *a priori* statistical information is not well specified, the filter might underestimate the variances of the state errors, $\mathbf{e}^{\mathrm{a}}_{\mathrm{k}}$. Too much confidence is put on the state estimation and too little confidence is put on the information contained in the observations. The effect of the analysis is minimized, and the gain becomes too small. In the most extreme case, observations are simply rejected. This is a *filter divergence*. Very often filter divergence is easy to diagnose: state error variances are small, and the time sequence of innovations is biased. The main rule to follow, to avoid this, is not to underestimate model errors. If possible, it is better to use an adaptive scheme to tune them on-the-fly.

(2) **Size and optimal interpolation.** The straightforward application of the Kalman filter implies the "propagation" of an $\mathrm{n} \times \mathrm{n}$ sized covariance matrix at each time step. This can result in a very large size problem (computations and storage). If the computational cost of propagating

P^a_{k+1} is an issue, we can use a *frozen covariance matrix*,

$$P^a_k = P^b, \quad k = 1, \ldots, n.$$

This defines a class of methods known as optimal interpolation (OI). Under this simplifying hypothesis, the two-step assimilation cycle defined above becomes,

(a) Forecast:

$$\begin{aligned} \mathbf{x}^f_{k+1} &= M_{k+1}\left[\mathbf{x}^a_k\right] \\ P^f_{k+1} &= P^b. \end{aligned}$$

(b) Analysis:

$$\begin{aligned} K_{k+1} &= P^b H^T \left(H P^b H^T + R_{k+1}\right)^{-1}, \\ \mathbf{x}^a_{k+1} &= \mathbf{x}^f_{k+1} + K_{k+1}\left(\mathbf{y}_{k+1} - H\mathbf{x}^f_{k+1}\right), \\ P^a_{k+1} &= P^b. \end{aligned}$$

There are at least two ways to compute the static covariance matrix P^b. The first is an analytical formulation,

$$P^b = D^{1/2} C D^{1/2},$$

where Dis a diagonal matrix of variances and C is a correlation matrix that can be defined, for example, as

$$C_{ij} = \left(1 + \alpha h + \frac{1}{3}\alpha^2 h^2\right) e^{-\alpha h},$$

where α is a tunable parameter and h is the grid size. The second approach uses an ensemble of N_e snapshots of the state vector taken from a model free run, out of which we compute the first and second statistical moments as follows

$$\mathbf{x}^b = \frac{1}{N_e}\sum_{l=1}^{N_e} \mathbf{x}_l,$$

$$P^b = \frac{1}{N_e - 1}\sum_{l=1}^{N_e}\left(\mathbf{x}_l - \mathbf{x}^b\right)\left(\mathbf{x}_l - \mathbf{x}^b\right)^T.$$

The static approach is more suited to successive assimilation cycles that are separated by a long enough time delay so that the corresponding dynamical states are sufficiently decorrelated.

(3) **Evolution of the state error covariance matrix.** In principle, equation (2.9) generates a symmetric matrix. In practice, this may not be the case and numerical truncation errors may lead to an asymmetric covariance matrix and a subsequent collapse of the filter. A remedy is to add an extra step to enforce symmetry, such as

$$P^f_{k+1} = \frac{1}{2}\left(P^f_{k+1} + (P^f_{k+1})^T\right),$$

or a square root decomposition.

2.2.6. Nonlinearities and extensions of the Kalman filter.

- extended Kalman filter
- ensemble Kalman filter
- others: unscented Kalman filter, particle methods

In real life problems, we are most often confronted with either a nonlinear process and/or a nonlinear measurement operator. To deal with these nonlinearities, one approach is to linearize about the current mean and covariance, which is called the *extended Kalman filter* (EKF).

As previously mentioned, the Kalman filter is only optimal in the case of Gaussian statistics and linear operators, in which case the first two moments (the mean and the covariances) suffice to describe the pdf entering the estimation problem. Practitioners report that the linearized extension to nonlinear problems, the EKF, only works for moderate deviations from linearity and Gaussianity. The *ensemble Kalman filter* (Evensen, 1994, 2009) is a method which has been designed to deal with strong nonlinearities and non-Gaussian statistics, whereby the pdf is described by an ensemble of N_e time-dependent states, $\mathbf{x}_{k,e}$.

What happens if the models are nonlinear and the pdf's are non Gaussian? The Kalman filter and its extensions are no longer optimal and, more importantly, can easily fail the estimation process. Another approach must be used. A promising candidate is the *particle filter*. The particle filter works sequentially in the spirit of the Kalman filter, but unlike the latter, it handles an ensemble of states (the particles) whose distribution approximates the pdf of the true state. Bayes' rule (2.1.6) and the marginalization formula (2.1.7) are explicitly used in the estimation process. The linear and Gaussian hypotheses can then be ruled out, in theory. In practice though, the particle filter cannot yet be applied to very high dimensional systems (this is often referred to as "the curse of dimensionality").

For further deatils of these extensions, the reader should consult the references.

2.2.7. Example: estimation of a random constant. In this simple example let us attempt to estimate a scalar random constant, a voltage for example. Let's assume that we have the ability to take measurements of the constant, but that the measurements are corrupted by a 0.1 volt RMS white measurement noise (e.g. our analog to digital converter is not very accurate). In this example, our process is governed by the state equation,

$$x_k = Mx_{k-1} + w_k = x_{k-1} + w_k$$

and the measurement equation,

$$y_k = Hx_k + v_k = x_k + v_k.$$

The state does not change from step to step so$M = I$. Our noisy measurement is of the state directly so $H = 1$.

The time-update (forecast) equations are,

$$\begin{aligned} x^f_{k+1} &= x^a_k\,, \\ P^f_{k+1} &= P^a_k + Q \end{aligned}$$

and the measurement update (analysis) equations are

$$\begin{aligned} K_{k+1} &= P^f_{k+1}(P^f_{k+1} + R)^{-1}, \\ x^a_{k+1} &= x^f_{k+1} + K_{k+1}(y_{k+1} - x^f_{k+1}), \\ P^a_{k+1} &= (1 - K_{k+1})P^f_{k+1}. \end{aligned}$$

2.2.7.1. *Initialization.* Presuming a very small process variance, we let $Q = 1.e-5$. (We could certainly let $Q = 0$ but assuming a small but non-zero value gives us more flexibility in "tuning" the filter as we will demonstrate below.) Let's assume that from experience we know that the true value of the random constant has a standard Gaussian probability distribution, so we will "seed" our filter with the guess that the constant is 0. In other words, before starting we let $x_0 = 0$. Similarly we need to choose an initial value for P^a_k, call it P_0. If we were absolutely certain that our initial state estimate was correct, we would let $P_0 = 0$. However given the uncertainty in our initial estimate x_0, choosing $P_0 = 0$ would cause the filter to initially and always believe that $x^a_k = 0$. As it turns out, the alternative choice is not critical. We could choose almost any $P_0 \neq 0$ and the filter would eventually converge. We'll start our filter with $P_0 = 1$.

2.2.7.2. *Simulations.* To begin with, we randomly chose a scalar constant $y = -0.37727$. We then simulated 100 distinct measurements that had error normally distributed around zero with a standard deviation of 0.1 (remember we presumed that the measurements are corrupted by a 0.1 volt RMS white measurement noise).

In the first simulation we fixed the measurement variance at $R = (0.1)^2 = 0.01$. Because this is the "true" measurement error variance, we would expect the "best" performance in terms of balancing responsiveness and estimate variance. This will become more evident in the second and third simulations. Figure 2.2.4 depicts the results of this first simulation. The true value of the random constant $x = -0.37727$ is given by the solid green line, the noisy measurements by the red dots and the filter estimate by the remaining blue curve.

In Figure 2.2.5 and Figure 2.2.6 below we can see what happens when R is increased or decreased by a factor of 100 respectively. In Figure 2.2.5 the filter was told that the measurement variance was 100 times greater (i.e. $R = 1$) so it was "slower" to believe the measurements. In Figure 2.2.6 the filter was told that the measurement variance was 100 times smaller (i.e. $R = 0.0001$) so it was very "quick" to believe the noisy measurements. While the estimation of a constant is relatively straight-forward, it clearly demonstrates the workings of the Kalman filter. In Figure 2.2.5 in particular the Kalman "filtering" is evident as the estimate appears considerably smoother than the noisy measurements.

Here is the MALAB code used to perform the simulations.

```
% SCALAR EXAMPLE (estimate a constant):
%
% Define the system as a constant of  -0.37727 volts:
clear s
s.x =  -0.37727;
s.A = 1;
```

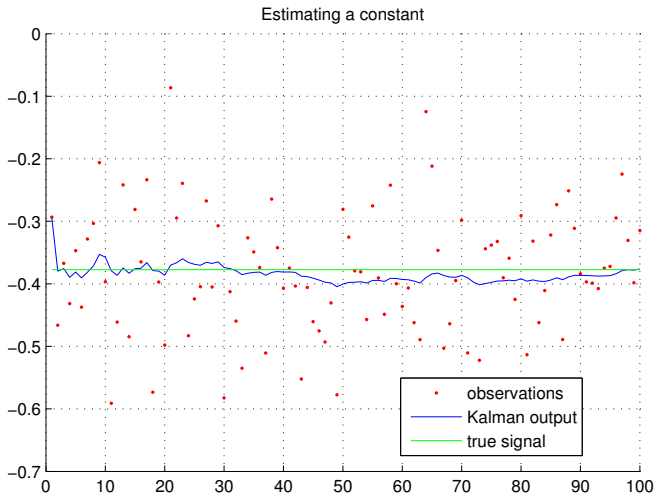

FIGURE 2.2.4. Estimating a constant - simulation with $R = 0.01$.

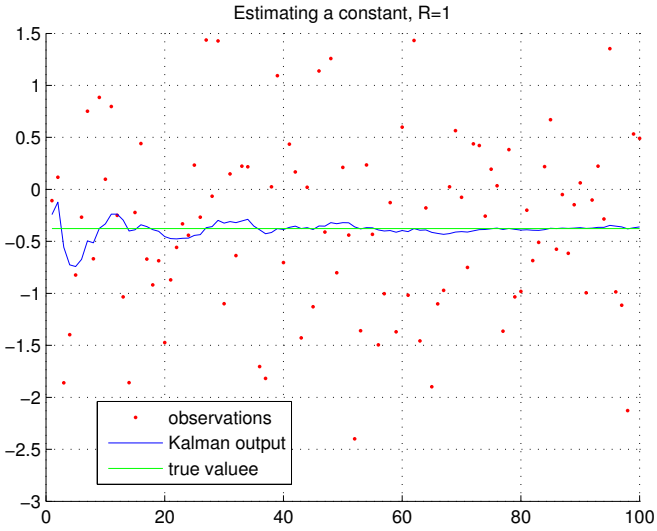

FIGURE 2.2.5. Estimating a constant - simulation with $R = 1$.

```
% Define a process noise (stdev):
s.Q = 0.00001; % variance, hence stdev^2
% Define the voltmeter to measure the voltage itself:
s.H = 1;
% Define a measurement error (stdev):
s.R = 0.1^2; % variance, hence stdev^2
```

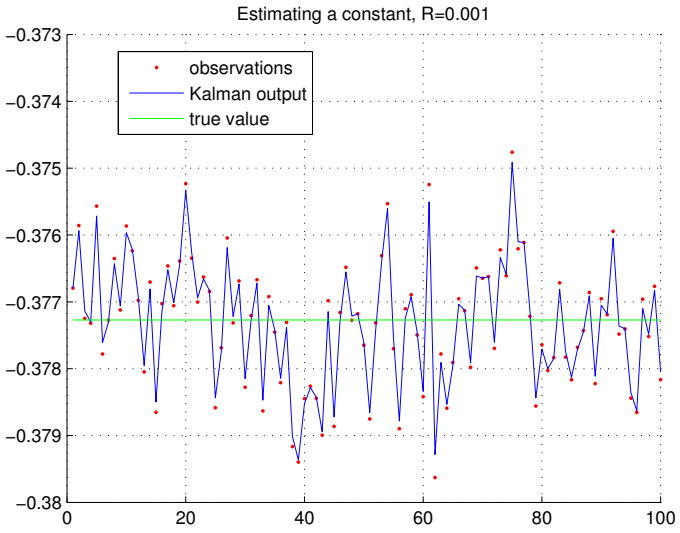

FIGURE 2.2.6. Estimating a constant - simulation with R = 0.0001.

```
RR  = 0.01;
% Do not define any system input (control) functions:
s.B = 0; s.u = 0;
% Specify an initial state:
s.x =  -0.37727;
s.P = 1;
% Generate random voltages and watch the filter operate.
tru=[]; % true voltage
for t=1:100
   tru(end+1) =  -0.37727;
   s(end).z = tru(end) + RR*randn; % create a measurement
   s(end+1)=kalmanf(s(end)); % perform a Kalman filter iteration
end
figure, hold on, grid on
% plot measurement data:
hz=plot([s(1:end-1).z],'r.');
% plot a-posteriori state estimates:
hk=plot([s(2:end).x],'b-'); ht=plot(tru,'g-');
legend([hz hk ht],'observations','Kalman output','true voltage',0)
title('Estimating a constant')
hold off
```

CHAPTER 3

Optimal control and variational data assimilation

3.1. Introduction

Unlike sequential assimilation (which emanates from estimation theory), variational assimilation is based on optimal control theory. The analyzed state is not defined as the one that maximizes a certain pdf, but as the one that *minimizes a cost functional.* The minimization requires numerical optimization techniques. These techniques all rely on the *gradient* of the cost function and this gradient will be obtained here with the aid of *adjoint methods.*

3.2. Optimization

3.2.1. Definitions and notation. Let V be a vector space and U a non-empty subset of V. Let $J : V \to \mathbb{R}$. We want to solve the following problem.

PROBLEM 25. Find $u \in U$ such that

$$J(u) = \min_{v \in U} J(v).$$

Notation:

- J is called the "criterion", "functional", "cost function";
- U is the space of admissible solutions, or the "control space";
- if $U = V$ the problem is unconstrained, if $U \neq V$ then the problem is constrained.

3.2.1.1. *Basic definitions.* We will use Hilbert spaces throughout, with the standard L^2 scalar product and norm. Now let $f : E \to \mathbb{R}$, where E is not necessarily of finnite dimension. We then have the following defintions of extremal points of f.

DEFINITION 26. We say that x^* is a *global minimum* of f if

$$f(x^*) \leq f(x), \quad \forall x \in E.$$

DEFINITION 27. We say that x^* is a *local minimum* of f if there exists a neighbourhood V of x^* such that

$$f(x^*) \leq f(x), \quad \forall x \in V.$$

DEFINITION 28. The set F is *convex* if

$$\forall x_1, x_2 \in F, \ \forall \alpha \in [0,1] \ : \quad \alpha x_1 + (1-\alpha) x_2 \in F.$$

DEFINITION 29. The function f is *convex* if E is convex and if

$$\forall x_1, x_2 \in F,\ \forall \alpha \in [0,1] \ : \quad f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2).$$

We say that f is *strictly convex* if

$$\forall x_1, x_2 \in F,\ \forall \alpha \in [0,1] \ : \quad f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha) f(x_2).$$

3.2.1.2. *Directional and Gâteaux derivatives.* Directional derivatives are generalizations of partial derivatives and measure the rate of change of a function in an arbitrary direction (the partial derivative is always in one of the coordinate directions). The Gâteaux derivative generalizes the directional derivative to infinite dimensional spaces. These generalized derivatives are very important for optimization and data assimilation problems.

DEFINITION 30. Let $f : E \to \mathbb{R}$. Then the *Gâteaux derivative* or *directional derivative* of f at the point x in the direction $d \in E$ is the limit

$$\lim_{\alpha \to 0} \frac{f(x+\alpha d) - f(x)}{\alpha} = \frac{d}{d\alpha} f(x+\alpha d)|_{\alpha=0}$$

if it exists. This directional derivative will denoted as $\frac{\partial f}{\partial d}(x)$ or $f'_d(x)$ or $\nabla f \cdot d$ or $\widehat{f}(x)[d]$, or $df(x,d)$.

EXAMPLE 31. An important example concerns the Gâteaux derivative of a *cost function*. Suppose that $X \in L^2(\Omega)$ where $\Omega \subset \mathbb{R}^n$ is a bounded, measurable set. The functional $J : X \to \mathbb{R}$ given by

$$J(u) = \int_\Omega F(u(x))\, dx$$

where F is a real-valued function with $F' = f$ and u is defined on Ω, has the Gâteaux derivative

$$\widehat{J}(u)[\psi] = (f(u), \psi),$$

where $(\cdot,\cdot)$ is the L^2 inner product, since

$$\begin{aligned} \frac{J(u+\alpha\psi) - J(u)}{\alpha} &= \frac{1}{\alpha}\left(\int_\Omega F(u+\alpha\psi)dx - \int_\Omega F(u)dx\right) \\ &= \frac{1}{\alpha}\left(\int_\Omega \int_0^1 \frac{d}{ds} F(u+s\alpha\psi)ds\, dx\right) \\ &= \int_\Omega \int_0^1 f(u+s\alpha\psi)ds\, dx \end{aligned}$$

and letting $\alpha \to 0$, gives

$$\widehat{J}(u)[\psi] = \int_\Omega f(u(x))\psi(x)\, dx.$$

As an application of this result, consider the *observation functional*

$$J^{obs}(K) = \frac{1}{2}\int_\Omega \left(u - u^{obs}\right)^2$$

where u depends indirectly on the parameters K. We compute the Gâteaux derivative

$$\begin{aligned}\hat{J}(K)[k] &= \lim_{\alpha\to 0}\frac{J(K+\alpha k)-J(K)}{\alpha}\\ &= \int_\Omega \left(u-u^{obs}\right)\hat{u},\end{aligned}$$

where

$$\hat{u}=\lim_{\alpha\to 0}\frac{u_{K+\alpha k}-u_K}{\alpha}.$$

Note that in practice, the computation of the directional derivative can be done by finite increments. We thus calculate

$$T(\alpha)=\frac{f(x+\alpha d)-f(x)}{\alpha}$$

for smaller and smaller values of α until we obtain numerical convergence.

In mathematics, the *Fréchet derivative* is a derivative defined on Banach spaces. It is commonly used to formalize the concept of the functional derivative used widely in the calculus of variations. Intuitively, it generalizes the idea of linear approximation from functions of one variable to functions on Banach spaces. The Fréchet derivative should be contrasted to the more general Gâteaux derivative which is a generalization of the classical directional derivative, as we saw above. The Fréchet derivative has applications throughout mathematical analysis, and in particular to the calculus of variations and much of nonlinear analysis and nonlinear functional analysis. It has applications to nonlinear problems throughout the sciences.

DEFINITION 32. Let E be a Hilbert space. We say that f is Fréchet differentiable at x if there exists a $p\in E$ such that

$$f(x+h)=f(x)+(p,h)+o(h),\quad \forall h\in E.$$

We then say that p is the derivative, the differential, or the *gradient* of f at x and we denote it as $f'(x)$ or $\nabla f(x)$. Note that in higher dimensions, the gradient becomes the *Jacobian matrix*.

3.2.2. Unconstrained optimization.

3.2.2.1. *Existence in finite dimension.* Let $f : E\to\mathbb{R}$ with $E\subset\mathbb{R}^n$ a closed, non-empty set and suppose that f is continuous.

THEOREM 33. *Suppose that* $\lim_{\|x\|\to\infty} f(x)=+\infty$, *then there exists at least one minimum of* f *in* E. *If* E *is a convex set of* $\mathbb{R}^n$ *and if* f *is strictly convex on* E *then there exists at most one minimum of* f *in* E. *If in addition,* $\lim_{\|x\|\to\infty} f(x)=+\infty$, *then ere exists a unique minimum of* f *in* E.

3.2.2.2. *Existence in infinite dimension.*

THEOREM 34. *Let* $f : E\to\mathbb{R}$ *with* E *a closed, non-empty, convex part of a Hilbert space. Suppose that* f *is convex, continuous and that* $\lim_{\|x\|\to\infty} f(x)=+\infty$. *Then here exists at least one minimum of* f *in* E.

3.2.2.3. *Optimality conditions.* The first condition is a *necessary condition for optimality*, also known as the Euler equation.

THEOREM 35. *Let* $f : E \to \mathbb{R}$ *with* E *a convex part of a Hilbert space. Suppose that* x^* *is an interior point of* E *and that there exists a bounded subset* Ω *such that* $x^* \in \Omega \subset E$. *If* f *is differentiable at* x^*, *then the fact that* x^* *is a local minimum of* f *implies that* $\nabla f(x^*) = 0$.

If we add the convexity condition, we obtain a necesssary and sufficient condition.

THEOREM 36. *Let* $f : E \to \mathbb{R}$ *with* E *a convex part of a Hilbert space. Suppose that* x^* *is an interior point of* E *and that there exists a bounded subset* Ω *such that* $x^* \in \Omega \subset E$. *If* f *is convex on* E *and differentiable at* x^*, *then* x^* *is a local minimum of* f *if and only if* $\nabla f(x^*) = 0$.

3.2.3. Minimization of a quadratic functional in finite dimension. We begin by defining a generalization of the inverse for rectanglar matrices, that for invertible matrices reduces to the standard inverse.

DEFINITION 37. Let M be an $m \times n$ matrix. The pseudo-inverse (Moore-Penrose inverse) of M is a matrix denoted $M^\dagger$ that satisfies
(3.2.1)
$$MM^\dagger M = M, \quad M^\dagger MM^\dagger = M^\dagger, \quad \left(MM^\dagger\right)^T = MM^\dagger, \quad \left(M^\dagger M\right)^T = M^\dagger M.$$

THEOREM 38. *Let* M *be an* $m \times n$ *matrix. If* M *is of rank* n, *then*

$$M^\dagger = \left(M^T M\right)^{-1} M^T.$$

PROOF. It suffices to verify the relations (3.2.1). □

Now let us consider the minimization of a quadratic functional.

THEOREM 39. *Consider the following quadratic functional*

$$J(x) = \frac{1}{2} \|Mx - b\|^2$$

for $x \in \mathbb{R}^n$ *where* M *is an* $m \times n$ *matrix of rank* n *and* $b \in \mathbb{R}^m$. *The solution of the minimization problem 25 is given by*

$$x^* = M^\dagger b.$$

PROOF. We expand the norm:

$$\begin{aligned} \|Mx - b\|^2 &= (Mx - b)^T (Mx - b) \\ &= x^T M^T Mx - b^T Mx - x^T M^T b + b^T b \\ &= x^T M^T Mx - 2b^T Mx + b^T b. \end{aligned}$$

Now compute the directional derivative by a finite increment,

$$\begin{aligned} J(x+\alpha\delta x)-J(x) &= \frac{1}{2}\left[(x+\alpha\delta x)^T M^T M(x+\alpha\delta x)-2b^T M(x+\alpha\delta x)+b^T b\right. \\ &\quad \left.-x^T M^T Mx+2b^T Mx-b^T b\right] \\ &= \frac{\alpha}{2}\left(\delta x^T M^T Mx+x^T M^T M\delta x-2b^T M\delta x\right)+\mathcal{O}(\alpha^2) \\ &= \alpha\left(x^T M^T M\delta x-b^T M\delta x\right)+\mathcal{O}(\alpha^2). \end{aligned}$$

Thus,

$$\lim_{\alpha\to 0}\frac{J(x+\alpha\delta x)-J(x)}{\alpha}=\left(x^T M^T M-b^T M\right)\delta x=\left(M^T Mx-M^T b,\delta x\right)$$

and the gradient is, by definition,

$$\nabla J(x)=M^T Mx-M^T b.$$

Finally, setting the gradient equal to zero, we obtain

$$\nabla J(x)=0 \quad\Longleftrightarrow\quad x=\left(M^T M\right)^{-1}M^T b=M^\dagger b.$$

□

3.2.3.1. *Relation with variational data assimilation.* We apply the above results to the minimization of the weighted functional

$$J_1(x)=\frac{1}{2}\|Mx-b\|_N^2=\frac{1}{2}\left(Mx-b\right)^T N\left(Mx-b\right),$$

for $x\in\mathbb{R}^n$, where M is an $m\times n$ matrix of rank n, $b\in\mathbb{R}^m$ and N is a symmetric positive-definite matrix. Then it is quite easy to see that the minimum of J_1 is attained for

$$x^*=\left(M^T NM\right)^{-1}M^T Nb.$$

We have just solved the following *data assimilation problem.*

PROBLEM 40. Let H be a linear observation operator, of rank n. We want to find x^* such that

$$J^o(x^*)=\min_x J^o(x)=\min_x\|Hx-y\|^2.$$

Note that the rank hypothesis on H implies that $n\leq m$ and we have more observations (y) than unknowns to estimate (x). Now, with the previous notation,

$$J^o(x)=\frac{1}{2}\left(Hx-y\right)^T R^{-1}\left(Hx-y\right)$$

and

$$x^*=\left(H^T R^{-1}H\right)^{-1}H^T R^{-1}y,$$

where R is the observation error covariance matrix.

In the same way, we can consider a cost function with a background term (see below),

$$\begin{aligned} J(x) &= J^b(x)+J^o(x) \\ &= \frac{1}{2}\left(x-x^b\right)^T B^{-1}\left(x-x^b\right)+\frac{1}{2}\left(Hx-y\right)^T R^{-1}\left(Hx-y\right)., \end{aligned}$$

where B is the background error covariance matrix.

For this case we just have to set

$$M = \begin{pmatrix} I \\ H \end{pmatrix}, \quad Y = \begin{pmatrix} x^b \\ y \end{pmatrix}, \quad N = \begin{pmatrix} B^{-1} & 0 \\ 0 & R^{-1} \end{pmatrix}$$

and we calculate

$$M^T N M = B^{-1} + H^T R^{-1} H,$$

$$M^T N Y = B^{-1} x^b + H^T R^{-1} Y.$$

It is then easy to see that the optimum is attained for

$$x^* = x^b + \left(B^{-1} + H^T R^{-1} H\right)^{-1} H^T R^{-1} (y - H x^b).$$

The matrix $\left(B^{-1} + H^T R^{-1} H\right)^{-1} H^T R^{-1}$ is called the *gain matrix* and the vector $(y - Hx^b)$ the *innovation* (we recognize the form of an optimal linear filter...). The gain matrix is often rewritten in the form

$$BH^T \left(R + HBH^T\right)^{-1}$$

according to the Sherman-Morrison-Woodbury formula.

In practice, due to the very large dimensions of the problem, m and n, it is often impossible to explicitly obtain the gain matrix. The computation of x^* is then performed by optimization algorithms (or by reduced-order methods).

3.2.4. Constrained optimization.

3.2.4.1. *Minimization with equality constraints.* Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and let K, the set of constraints, be

$$K = \{x \in \mathbb{R}^n \mid h_1(x) = 0, \ldots, h_p(x) = 0\},$$

where the functions $h_i : \mathbb{R}^n \to \mathbb{R}$ are at least of class C^1. We seek $x^* \in K$ such that

$$f(x^*) = \min_{x \in K} f(x).$$

THEOREM 41. *If $x^* \in K$ is a local minimum of f in K, and if the vectors $\nabla h_i(x^*)$, $i = 1, \ldots, p$ are linearly independent, then there exists $\lambda^* = (\lambda_1^*, \ldots, \lambda_p^*) \in \mathbb{R}^p$ such that*

$$\nabla f(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla h_i(x^*) = 0.$$

We introduce the *Lagrangian* $\mathcal{L}$ defined by

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i=1}^{p} \lambda_i h_i(x).$$

Then solving $\nabla f(x) = 0$ in K is exactly equivalent to solving $\nabla \mathcal{L} = 0$ (finding a saddle point). This means that we solve the two following equations:

$$\begin{aligned} \nabla_x \mathcal{L} &= \nabla f + \sum_{i=1}^{p} \lambda_i \nabla h_i = 0, \\ \nabla_\lambda \mathcal{L} &= (h_1, h_2, \ldots, h_p)^T = 0. \end{aligned}$$

The second line is the constraint equations, and hence $x \in K$, and the first line then gives the necessary condition of the theorem.

3.2.5. Optimization algorithms: gradient methods. We consider the following optimization problem.

PROBLEM 42. Find the minimum x^* of a cost function J, such that

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

We will call *descent method,* any algorithm of the type

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

where $d_k \in \mathbb{R}^n$ is the descent direction at iteration k and $\alpha_k \in \mathbb{R}$ is the step-size at iteration k. The descent methods differ in their choices of α_k et d_k.

By defintion of the gradient we can expand the objective function,

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

Since J is positive, the principal decrease will be reached when $h = -\alpha \nabla J(x_k)$, which is the same as setting

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

This is the direction of *steepest descent.*

3.2.5.1. *Conjiugate gradient method.* For the special case where

$$J(x) = \frac{1}{2}(Ax, x) - (b, x) = \frac{1}{2}x^T Ax - b^T x$$

with A symmetric and positive definite, we can define the *conjugate gradient* algorithm as follows:

The algorithm can be summarized as follows:

$$\begin{aligned} d_k &= \nabla J(x_k) + d_{k-1} \frac{\|\nabla J(x_k)\|^2}{\|\nabla J(x_{k-1})\|^2} \\ \alpha_k &= -\frac{(\nabla J(x_k), d_k)}{(Ad_k, d_k)}, \end{aligned}$$

where $\nabla J(x_k) = Ax_k - b$ is the "residual". This algorithm converges in at most n iterations, but it's cost is $\mathcal{O}(n^2)$ which can be prohibitive for large, full matrices. However, for sparse matrices it is often used since only the matix-vector products Ad_k and Ax_k need to be stored and the cost reduces to $\mathcal{O}(m)$ only, where m is the number of non-zero entries in A.

Here is a pseudo-code implementaion of the conjugate gradient method.

$k = 0,\ r_0 = Ax_0 - b,\ d_0 = -r_0$
`while` $r_k \neq 0$
$\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$
$x_{k+1} = x_k + \alpha_k d_k$
$r_{k+1} = r_k + \alpha_k A d_k \quad (r_{k+1} = Ax_{k+1} - b\ )$
$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
$d_{k+1} = -r_{k+1} + \beta_{k+1} d_k$
$k = k + 1$
`end`

3.2.5.2. *Newton methods.* The basic idea is to use the aproximation of the derivative by a tangent. If we want to solve $f(x) = 0$, we can suppose that x_k is known and we define the next point, x_{k+1}, as the intersection of the tangent to the curve f at x_k with the x-axis,

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

Generalizing to dimension n, we obtain

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

where $f'(x^{(k)})$ is the Jacobian matrix

$$\left[\frac{\partial f_i}{\partial x_j}(x^{(k)})\right].$$

At each iteration we thus have to compute the Jacobian and to solve a linear system

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

Let us now apply Newton's method to the Euler equation $\nabla J(x) = 0$. We obtain

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

where $\nabla^2 J(x_k)$ is the Hessian matrix of J. The advantage of this method is that the convergence is now quadratic, but the major disadvantage is the expense of computing the Hessian at each iteration and solving the resulting linear system. To remedy this, *quasi-Newton methods* are systematically used. These methods are based on suitable approximations of the inverse of the Hessian.

EXAMPLE 43. Use Newton's method to minimize the function

$$f(\mathbf{x}) = 0.5x_1^2 + 2.5x_2^2.$$

We easily compute the gradient and the Hessian of $f(x)$:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} x_1 \\ 5x_2 \end{bmatrix}, \quad \nabla^2 f(\mathbf{x}) = H_f(\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}.$$

Taking an initial vector, $x_0 = [5\ 1]^T$, we find $\nabla f(x_0) = [5\ 5]^T$, and the linear system for the Newton iteration is

$$\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix} d_0 = \begin{bmatrix} -5 \\ -5 \end{bmatrix}$$

and thus

$$x_1 = x_0 + d_0 = \begin{bmatrix} 5 \\ 1 \end{bmatrix} + \begin{bmatrix} -5 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

As expected, we have found the exact solution to this quadratic problem.

3.2.5.3. *Quasi-Newton methods.* The two most widely used methods are the *Davidon-Fletcher-Powell* (DFP), or variable metric method, and the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method. [**10**].

3.3. Adjoint methods

3.3.1. Introduction. All the optimization methods seen above require the computation of the gradient ∇J. If the dependence of J on the control variables is complex or indirect, this computation can be very difficult. We saw above that the minimization of

$$J^o(x) = \int_0^T \|H(x(t)) - y^o(t)\|^2 \, dt$$

involves the inversion of a very large linear system. Numerically, we can always manage by computing finite increments, but this would have to be done in all the possible perturbation directions. We thus need to find a less expensive way to compute the gradient.

3.3.2. A simple ODE example. Let us consider an inverse problem based on the ordinary differential equation (of convection-diffusion type)

$$\begin{cases} -bu''(x) + cu'(x) = f(x) \quad 0 < x < 1 \\ u(0) = 0,\ u(1) = 0, \end{cases} \tag{3.3.1}$$

where f is a given function in $L^2(0,1)$, and b and c are unknwon parameters that we seek to identify using observations of $u(x)$ on $[0,1]$. The cost function is then

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

Let us calculate its gradient. This can be done in two ways: using the *Lagrangian* or using the *tangent linear model*. We will start with the second approach. Perturbing the cost function gives

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

where $\tilde{u} = u_{b+\alpha\delta b, c+\alpha\delta c}$ and $u = u_{b,c}$. Expanding and rearranging, we obtain

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

Now we divide by α and pass to the limit $\alpha \to 0$, to obtain the directional derivative

$$\hat{J}[b,c](\delta b, \delta c) = 2\int_0^1 \left(u - u^{obs}\right)\hat{u}\,dx, \tag{3.3.2}$$

where

$$\hat{u} = \lim_{\alpha\to 0} \frac{\tilde{u}-u}{\alpha}.$$

Let us now use this definition to find the equation satisfied by $\hat{u}$. We have

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

and

$$\begin{cases} -bu'' + cu' = f, \\ u(0) = 0,\ u(1) = 0. \end{cases}$$

Then, subtracting the two equations and passing to the limit, we obtain,

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

Finally, we can define the *tangent linear model*

$$\begin{cases} -b\hat{u}'' + c\hat{u}' = (\delta b)u'' - (\delta c)u' \\ \hat{u}(0) = 0,\ \hat{u}(1) = 0. \end{cases} \tag{3.3.3}$$

We want to be able to reformulate the directional derivative (3.3.2), in order to obtain an expression for the gradient. So we multiply the tangent linear model (3.3.3) by a variable p and we integrate twice by parts:

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

which gives

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

and

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

Putting these results together, we get

$$-b\left(\hat{u}'(1)p(1) - \hat{u}'(0)p(0) + \int_0^1 \hat{u}p''\right) + c\left(-\int_0^1 \hat{u}p'\right) = \int_0^1 ((\delta b)u'' - (\delta c)u')\,p$$

or

$$\int_0^1 (-bp'' - cp')\,\hat{u} = b\hat{u}'(1)p(1) - b\hat{u}'(0)p(0) + \int_0^1 ((\delta b)u'' - (\delta c)u')\,p. \tag{3.3.4}$$

Now we impose that p must satisfy *the adjoint model*

$$\begin{cases} -bp'' - cp' = 2(u - u^{obs}), \\ p(0) = 0,\ p(1) = 0. \end{cases} \tag{3.3.5}$$

Integrating (3.3.5) and using the expression (3.3.4), we obtain

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

We recognize the L^2 inner product, which enables us to finally write an explicit expression for the gradient based on (3.3.2),

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

or

$$\begin{aligned} \nabla_b J(b,c) &= \int_0^1 pu'' \\ \nabla_c J(b,c) &= -\int_0^1 pu'. \end{aligned}$$

Thus, for the additional cost of solving the adjoint model (3.3.5), we can compute the gradient of the cost function with respect to either one, or both, of the unknown parameters.

3.3.3. Initial condition control. For data assimilation problems in meteorology and oceanography, the objective is to reconstruct the initial conditions of the model. We redo the above gradient calculations in this context. Let us consider the following system of (possibly nonlinear) ordinary differential equations,

$$\begin{cases} \dfrac{dX}{dt} = M(X) \quad \text{in } \Omega \times [0,T], \\ X(t=0) = U, \end{cases} \tag{3.3.6}$$

with the cost function

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

To compute the directional derivative, we perturb U in the direction u and denote by $\tilde{X}$ the correponding trajectory, satisfying

$$\begin{cases} \dfrac{d\tilde{X}}{dt} = M(\tilde{X}) \quad \text{in } \Omega \times [0,T], \\ \tilde{X}(t=0) = U + \alpha u. \end{cases} \tag{3.3.7}$$

We then have

$$\begin{aligned} J(U+\alpha u)-J(U) &= \frac{1}{2}\int_0^T \left\|H\tilde{X}-Y^o\right\|^2 - \|HX-Y^o\|^2\,dt \\ &= \frac{1}{2}\int_0^T \left(H\tilde{X}-Y, H\tilde{X}-HX+HX-Y\right) - (HX-Y, HX-Y) \\ &= \frac{1}{2}\int_0^T \left(H\tilde{X}-Y, H(\tilde{X}-X)\right) - (H\tilde{X}-Y-(HX-Y), HX-Y) \\ &= \frac{1}{2}\int_0^T \left(H\tilde{X}-Y, H(\tilde{X}-X)\right) + (H(\tilde{X}-X), HX-Y). \end{aligned}$$

We set

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

and we compute

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

By subtracting the equations (3.3.7) and (3.3.6) satisfied by $\tilde{X}$ and X we obtain,

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

Now we divide by α and pass to the limit $\alpha\to 0$, to obtain

$$\begin{cases} \dfrac{d\hat{X}}{dt} = \left[\dfrac{\partial M}{\partial X}\right]\hat{X}, \\ \hat{X}(t=0) = u. \end{cases} \tag{3.3.8}$$

These equations are the *tangent linear model* (TLM).

We will now proceed to compute the adjoint model. As in the ODE example, we multiply the TLM (3.3.8) by P and integrate by parts on $[0,T]$. We find,

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

and

$$\int_0^T \left(\left[\frac{\partial M}{\partial X} \right] \hat{X}, P \right) = \int_0^T \left(\hat{X}, \left[\frac{\partial M}{\partial X} \right]^T P \right).$$

Thus, substituting in equation (3.3.8), we get

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

Identifying with the directional derivative

$$\hat{J}[U](u) = \int_0^T (\hat{X}, H^T(HX - Y)), \tag{3.3.9}$$

we obtain the equations of the *adjoint model*

$$\begin{cases} \dfrac{dP}{dt} + \left[\dfrac{\partial M}{\partial X} \right]^T P = H^T(HX - Y), \\ P(t = T) = 0, \end{cases} \tag{3.3.10}$$

which is a backward model, integrated from $t = T$ to $t = 0$.

We can now find the expression for the gradient. Using the adjoint model (3.3.10) in (3.3.9), we find

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

But

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

and thus

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

Once again, with a single integration of the adjoint model, we obtain a particularly simple expression for the gradient of the cost function with respect to the control parameter.

3.3.4. Application: Burgers' equation. We consider a realistic application based on Burgers' equation with control of the initial condition and the boundary conditions. The viscous Burgers' equation in the interval $x \in [0, L]$ is defined as

$$\begin{gathered} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f, \\ u(0,t) = \psi_1(t), \quad u(L,t) = \psi_2(t), \\ u(x,0) = u_0(x). \end{gathered}$$

The control vector is

$$(u_0, \psi_1, \psi_2)$$

and the cost function is given by

$$J(u_0, \psi_1, \psi_2) = \frac{1}{2} \int_0^T \int_0^L \left(u - u^{obs}\right)^2 .$$

We know that the derivative of J in the direction (h_u, h_1, h_2) is given (as above) by

$$\hat{J}\left[u_0, \psi_1, \psi_2\right](h_u, h_1, h_2) = \int_0^T \int_0^L \left(u - u^{obs}\right) \hat{u},$$

where $\hat{u}$ is defined as

$$\begin{aligned} \hat{u} &= \lim_{\alpha \to 0} \frac{\tilde{u} - u}{\alpha} \\ &= \lim_{\alpha \to 0} \frac{u(u_0 + \alpha h_u, \psi_1 + \alpha h_1, \psi_2 + \alpha h_2) - u(u_0, \psi_1, \psi_2)}{\alpha} \end{aligned}$$

and it is the solution of the tangent linear model,

$$\begin{gathered} \frac{\partial \hat{u}}{\partial t} + \frac{\partial (u\hat{u})}{\partial x} - \nu \frac{\partial^2 \hat{u}}{\partial x^2} = 0, \\ \hat{u}(0,t) = h_1(t), \quad \hat{u}(L,t) = h_2(t), \\ \hat{u}(x,0) = h_u(x). \end{gathered}$$

We can now compute the equation of the adjoint model. As usual, we multiply the TLM by p and integrate by parts on $[0, T]$.

$$\begin{aligned} \int_0^T \left(\frac{\partial \hat{u}}{\partial t}, p\right) &= \int_0^T \int_0^L \frac{\partial \hat{u}}{\partial t} p \\ &= \int_0^L \left[\hat{u}p\right]_0^T - \int_0^L \int_0^T \frac{\partial p}{\partial t} \hat{u} \\ &= \int_0^L \left(\hat{u}(T)p(x,T) - h_u p(x,0)\right) - \int_0^L \int_0^T \frac{\partial p}{\partial t} \hat{u} \end{aligned}$$

$$\begin{aligned} \int_0^T \left(\frac{\partial (u\hat{u})}{\partial x}, p\right) &= \int_0^T \int_0^L \frac{\partial (u\hat{u})}{\partial x} p \\ &= \int_0^T \left[u\hat{u}p\right]_0^L - \int_0^T \int_0^L u\hat{u} \frac{\partial p}{\partial x} \\ &= \int_0^T \left(\psi_2 h_2 p(L,t) - \psi_1 h_1 p(0,t)\right) - \int_0^T \int_0^L u\hat{u} \frac{\partial p}{\partial x} \end{aligned}$$

$$\begin{aligned}
\int_0^T \left(\frac{\partial^2 \hat{u}}{\partial x^2}, p\right) &= \int_0^T \int_0^L \frac{\partial^2 \hat{u}}{\partial x^2} p \\
&= \int_0^T \left[p \frac{\partial \hat{u}}{\partial x}\right]_0^L - \int_0^T \int_0^L \frac{\partial \hat{u}}{\partial x} \frac{\partial p}{\partial x} \\
&= \int_0^T \left[p \frac{\partial \hat{u}}{\partial x} - \hat{u} \frac{\partial p}{\partial x}\right]_0^L + \int_0^T \int_0^L \hat{u} \frac{\partial^2 p}{\partial x^2} \\
&= \int_0^T \left(p(L,t) \frac{\partial \hat{u}}{\partial x}(L,t) - h_2 \frac{\partial p}{\partial x}(L,t) - p(0,t) \frac{\partial \hat{u}}{\partial x}(0,t) + h_1 \frac{\partial p}{\partial x}(0,t)\right) \\
&\quad + \int_0^T \int_0^L \hat{u} \frac{\partial^2 p}{\partial x^2}.
\end{aligned}$$

The natural boundary conditions for p are thus

$$p(x,T) = 0, \quad p(0,t) = p(L,t) = 0$$

and this gives

$$\begin{aligned}
0 &= \int_0^T \int_0^L \left(\frac{\partial \hat{u}}{\partial t} + \frac{\partial (u\hat{u})}{\partial x} - \nu \frac{\partial^2 \hat{u}}{\partial x^2}\right) p \\
&= \int_0^T \int_0^L \hat{u} \left(-\frac{\partial p}{\partial t} - u \frac{\partial p}{\partial x} - \nu \frac{\partial^2 p}{\partial x^2}\right) \\
&\quad + \int_0^L -h_u p(x,0) + \int_0^T \nu h_2 \frac{\partial p}{\partial x}(L,t) - \nu h_1 \frac{\partial p}{\partial x}(0,t).
\end{aligned}$$

In other words,

$$\int_0^T \int_0^L \hat{u} \left(-\frac{\partial p}{\partial t} - u \frac{\partial p}{\partial x} - \nu \frac{\partial^2 p}{\partial x^2}\right) = -\int_0^L h_u p(x,0) + \int_0^T \nu h_2 \frac{\partial p}{\partial x}(L,t) - \nu h_1 \frac{\partial p}{\partial x}(0,t).$$

We thus set the adjoint model as

$$\begin{gathered}
\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} - \nu \frac{\partial^2 p}{\partial x^2} = u - u^{obs}, \\
p(0,t) = 0, \quad p(L,t) = 0, \\
p(x,T) = 0.
\end{gathered}$$

Now we can rewrite the gradient of J in the form,

$$\hat{J}\left[u_0, \psi_1, \psi_2\right](h_u, h_1, h_2) = -\int_0^L h_u p(x, t=0) + \int_0^T \nu h_2 \frac{\partial p}{\partial x}(x = L, t) - \nu h_1 \frac{\partial p}{\partial x}(x = 0, t)$$

which yields,

$$\begin{aligned}
\nabla_{u_0} J &= -p(x, t=0) \\
\nabla_{\psi_1} J &= -\nu \frac{\partial p}{\partial x}(x = 0, t) \\
\nabla_{\psi_2} J &= \nu \frac{\partial p}{\partial x}(x = L, t).
\end{aligned}$$

3.3.5. Practical implementation of the adjoint method. See the enclosed course notes of A. Vidard, where the following topics are treated.

3.3.5.1. *Continuous and discrete adjoints.*

3.3.5.2. *Adjoint for data assimilation.*
3.3.5.3. *Validation of an adjoint code.*

3.4. Variational DA

3.4.1. Introduction. We have seen above that he BLUE requires the computation of the gain matrix,

$$K = BH^T(HBH^T + R)^{-1}.$$

in order to obtain an analyzed state

$$x^a = x^b + K(y - H(x^b))$$

that minimizes the cost function,

$$J(x) = \frac{1}{2}\left(x - x^b\right)^T B^{-1}\left(x - x^b\right) + \frac{1}{2}\left(Hx - y\right)^T R^{-1}\left(Hx - y\right).$$

But the matrices involved in this calculation are often neither storable in memory, nor manipulable because of their very large dimensions. The basic idea of variational methods is to overcome these difficulties by attempting to directly minimize the cost function J. This is achieved by a combination of a descent algorithm in the direction of the gradient, with an adjoint method for the computation of the gradient.

In the case where there is no time dependence, the approach is named 3D-Var, whereas for time-dependent problems we use the 4D-Var approach. Since we have already prepared most of the theory in the above chapters, the presentation here will be quite brief.

3.4.2. 3D-Var. The cost function for 3D-Var is

$$J(x) = \frac{1}{2}\left(x - x^b\right)^T B^{-1}\left(x - x^b\right) + \frac{1}{2}\left(y - H(x)\right)^T R^{-1}\left(y - H(x)\right).$$

We recall that R and B are the observation and background error covariance matrices respectively. When the observation operator H is linear, the gradient of J is given by

$$\nabla J = B^{-1}\left(x - x^b\right) - H^T R^{-1}\left(y - Hx\right).$$

The iterative 3D-Var algorithm uses as stopping criteria the fact that ∇J is small or that the maximum number of iterations is reached.

```
k = 0,   x = x_0
while ||∇J|| > ε or k ≤ k_max
compute J
compute ∇J
gradient descent and update of x_{k+1}
k = k + 1
end
```

A few remarks are necessary. Firstly, the matrix B cannot, in general, be explicitly stored and it must be modeled somehow. However, only matrix-vector products involving B^{-1} are encountered. This permits complex modeling using an operator form. To achieve this, we define a complex function that receives ϕ as input and returns $B^{-1}\phi$ as output. A second remark concerns the possibility of using this algorithm for time dependent problems that are of very large dimension. In this case, x represents the initial state and the observations y^o are reported to the initial instant. This greatly simplifies the gradient computation since there is no integration of the adjoint, nor integration of the model M.

EXAMPLE 44. We seek two temperatures, x_1 and x_2, in Caracas and Barquisimeto. The climatologist gives us an initial guess (based on climate records) $x^b = (10\ 5)^T$ with background error covariance matrix

$$B = \begin{pmatrix} 1 & 0.25 \\ 0.25 & 1 \end{pmatrix}.$$

We observe $y^o = 4$ in Barquisimeto, which implies that $H = (0\ 1)$, with an observation error variance $R = (0.25)$. We can now write the cost function

$$\begin{aligned} J(x) &= \begin{pmatrix} x_1 - 10 & x_2 - 5 \end{pmatrix} \begin{pmatrix} 1 & 0.25 \\ 0.25 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_1 - 10 \\ x_2 - 5 \end{pmatrix} + R^{-1}(x_2 - 4)^2 \\ &= \begin{pmatrix} x_1 - 10 & x_2 - 5 \end{pmatrix} \frac{16}{15} \begin{pmatrix} 1 & -0.25 \\ -0.25 & 1 \end{pmatrix} \begin{pmatrix} x_1 - 10 \\ x_2 - 5 \end{pmatrix} + 4(x_2 - 4)^2 \\ &= \frac{16}{15}\left((x_1 - 10)^2 + (x_2 - 5)^2 - 0.5(x_1 - 10)(x_2 - 5)\right) + 4(x_2 - 4)^2 \\ &= \frac{16}{15}\left(x_1^2 - 17.5x_1 + 100 + x_2^2 - 5x_2 - 0.5x_1x_2\right) + 4(x_2^2 - 8x + 16) \end{aligned}$$

and its gradient

$$\nabla J(x) = \frac{16}{15}\begin{pmatrix} 2x_1 - 0.5x_2 - 17.5 \\ 2x_2 - 5 - 0.5x_1 + \frac{15}{4}(2x_2 - 8) \end{pmatrix} = \frac{1}{15}\begin{pmatrix} 32x_1 - 8x_2 - 280 \\ -8x_1 + 152x_2 - 560 \end{pmatrix}.$$

The minimum is obtained for $\nabla J(x) = 0$, which gives

$$x_1 = 9.8, \quad x_2 = 4.2.$$

3.4.3. 4D-Var. The 4D-Var generalizes the 3D-Var to the case where the observations are obtained at different times - see Figure 3.4.1.

3.4.3.1. *Cost function and gradient.* The cost function is still expressed in terms of the initial state x, but it now includes the model because the observation y_i^o at time i is compared to $H_i(x_i)$, where x_i is the state at time i initialized by x. The cost function is once again the sum of the baclground and the observation errors,

$$J(x) = J^b(x) + J^o(x),$$

where the background term is the same as above,

$$J^b(x) = \frac{1}{2}\left(x - x^b\right)^T B^{-1}\left(x - x^b\right).$$

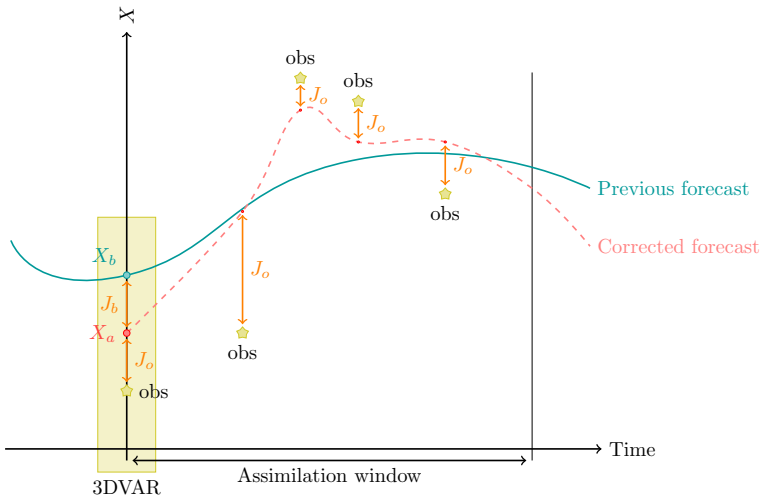

FIGURE 3.4.1. 3D- and 4D-Var (Credit: A. Vidard).

The background x^b, as with x, is taken as a vector at the initial time, $i = 0$. The observation term is more complicated. We define

$$J^o(x) = \frac{1}{2}\sum_{i=0}^{n} \left(y_i^o - H_i(x_i)\right)^T R_i^{-1} \left(y_i^o - H_i(x_i)\right),$$

where the state at time i is obtained by an iterated composition of the model matrix,

$$\begin{aligned} x_i &= M_{0\to i}(x) \\ &= M_{i-1,i}M_{i-2,i-1}\dots M_{1,2}M_{0,1}x \\ &= M_iM_{i-1}\dots M_2M_1x. \end{aligned}$$

This gives the final form of the observation term,

$$J^o(x) = \frac{1}{2}\sum_{i=0}^{n} \left(y_i^o - H_iM_iM_{i-1}\dots M_2M_1x\right)^T R_i^{-1} \left(y_i^o - M_iM_{i-1}\dots M_2M_1x\right).$$

Now we can compute the gradient,

$$\nabla J(x) = B^{-1}\left(x - x^b\right) - \sum_{i=0}^{n} M_1^TM_2^T\dots M_{i-1}^TM_i^TH_i^TR_i^{-1}\left(y_i^o - M_iM_{i-1}\dots M_2M_1x\right).$$

If we denote the innovation vector

$$d_i = y_i^o - H_iM_iM_{i-1}\dots M_2M_1x,$$

then we have

$$\begin{aligned}
-\nabla J^o(x) &= \sum_{i=0}^{n} M_1^T M_2^T \dots M_{i-1}^T M_i^T H_i^T R_i^{-1} d_i \\
&= H_0^T R_0^{-1} d_0 + M_1^T H_1^T R_1^{-1} d_1 + M_1^T M_2^T H_2^T R_2^{-1} d_2 + \dots + \\
&\quad M_1^T \dots M_{n-1}^T M_n^T H_n^T R_n^{-1} d_n \\
&= H_0^T R_0^{-1} d_0 + M_1^T \left[H_1^T R_1^{-1} d_1 + M_2^T \left[H_2^T R_2^{-1} d_2 + \dots + M_n^T H_n^T R_n^{-1} d_n \right] \right].
\end{aligned}$$

This factorization enables us to compute J^o followed by ∇J^o with one integration of the direct model followed by one integration of the adjoint model.

3.4.3.2. *Algorithm.*

```
n = 0,   x = x_0
while ||∇J|| > ε or n ≤ n_max
(1) compute J with the direct model M and H
(2) compute ∇J with the adjoint model M^T and H^T in reverse mode
gradient descent and update of x_{n+1}
n = n + 1
end
```

In step (1), we use the equations

$$d_i = y_i^o - H_i M_i M_{i-1} \dots M_2 M_1 x$$

and

$$J(x) = \frac{1}{2} \left(x - x^b \right)^T B^{-1} \left(x - x^b \right) + \sum_{i=0}^{n} d_i^T R_i^{-1} d_i.$$

In step (2), we use

$$\nabla J(x) = B^{-1} \left(x - x^b \right) - \left[H_0^T R_0^{-1} d_0 + M_1^T \left[H_1^T R_1^{-1} d_1 + M_2^T \left[H_2^T R_2^{-1} d_2 + \dots + M_n^T H_n^T R_n^{-1} d_n \right] \right] \right].$$

3.4.4. Extensions and complements.

3.4.4.1. *Parameter estimation.* If we want to optimise a set of parmeters,

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

we only have to include the control variables as terms in the cost function,

$$J(x, \alpha) = J_1^b(x) + J_2^b(\alpha) + J^o(x, \alpha).$$

The observation term includes a dependence on α and it is often necessary to add a regularization term for α, such as

$$J_2^b(\alpha) = \left\| \alpha - \alpha^b \right\|^2, \text{ or } \quad J_2^b(\alpha) = \left(\alpha - \alpha^b \right) B_\alpha^{-1} \left(\alpha - \alpha^b \right),$$

or

$$J_2^b(\alpha) = \left\| \nabla \alpha - \beta \right\|^2.$$

3.4.4.2. *Nonlinearities.* When the nonlinearities in the model and/or the observation operator are weak, we can extend the 3D- and 4D-Var algorithms to take their effects into account. One can define then the *incremental 4D-Var algorithm.* [**9**].

3.4.4.3. *Effect of a single observation.* Suppose that we have a single onservation, at one point corresponding to the k-th element of the state vector. The observation operator is then

$$H = (\ 0 \quad \dots \quad 0 \quad 1 \quad 0 \quad \dots \quad 0\ ).$$

The gradient of J is

$$\nabla J = B^{-1}\left(x - x^b\right) + H^T R^{-1}\left(Hx - y^o\right).$$

It is equal to zero at the minimum x^a, so that

$$\left(x^a - x^b\right) = BH^T R^{-1}\left(y^o - Hx^a\right).$$

But $R = \sigma^2$, $Hx^a = x^a_k$ and BH^T is the k-th column of B. So we see that

$$x^a - x^b = \frac{y^o - x^a_k}{\sigma^2}\begin{pmatrix} B_{1,k} \\ B_{2,k} \\ \vdots \\ B_{n,k} \end{pmatrix}.$$

The increment is proportional to a column of B. The choice of B is thus crucial and will determine how this observation provides information about what happens around the k-th variable.

In the 4D-Var case, the increment at time i will be proportional to a single column of MBM^T which describes the error covariances of the background at the time of the observation i.

3.4.4.4. *Preconditioning.* We recall that the condition number of a matrix A is the product $\|A\|\left\|A^{-1}\right\|$. In general, variational data assimilation problems are badly conditioned. The rate of convergence of the minimization algorothms depends on the conditioning of the Hessian of the cost function: the closer it is to 1, the better the convergence is. For 4D-Var the Hessian is equal to $B^{-1} + H^T R^{-1} H$ and its condition number is usually very high.

Preconditioning is a technique for improving the condition number and thus acelerating the convergence of the optimization. We make a change of variable

$$\delta x = x - x^b$$

such that

$$w = L^{-1}\delta x, \quad B^{-1} = LL^T$$

where L is a given, simple matrix. This is commonly used in meteorology and oceanography. The modified cost function is

$$\tilde{J}(w) = \frac{1}{2}w^T w + \frac{1}{2}(HLw - d)^T R^{-1}(HLw - d)$$

and its Hessian is equal to

$$\tilde{J}'' = I + L^T H^T R^{-1} HL.$$

It is in general much better conditioned ans the resulting improvement in convergence can be spectacular.

3.5. Advanced techniques

These will be the object of an advanced course...

3.5.1. Sensitivity analysis.

3.5.2. Reduced-order methods.

3.5.3. Error covariance modeling.

Bibliography

[1] Ide, Ghil. Unified Notation for Data Assimilation Operational Sequential and Variational. *J Met Soc Japan* (1997).

[2] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME—Journal of Basic Engineering*, 82(Series D), 35-45.

[3] Maybeck, P. S. (1979). *Stochastic models, estimation, and control* (Vol. 141). Academic Press.

[4] G. Evensen. *Data assimilation, The Ensemble Kalman Filter*, 2nd ed., Springer, 2009.

[5] A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* SIAM. 2005.

[6] O. Talagrand. Assimilation of observations, an introduction. *J. Meteorological Soc. Japan*, **75**, 191–209, 1997.

[7] F.X. Le Dimet, O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus,* **38**(2), 97–110, 1986.

[8] J.-L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Rev.*, **30**(1):1–68, 1988.

[9] P. Courtier, J.-N. Thépaut, A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Q. J. R. Meteorol. Soc.* (1994), 120, pp. 1367-1387.

[10] Press, Teukolsky, *et al. Numerical Recipes: The Art of Scientific Computing*, Third Edition (2007), 1256 pp. Cambridge University Press.