Chapter 4

Variational Data Assimilation for Unsteady PDE Systems

This chapter aims at extending the method of the previous chapter (linear tangent model, adjoint model, local sensitivities, optimal set of parameters) to unsteady PDEs systems, parabolic or hyperbolic, non-linear or not.
Calculations derived in the present section are formal. The final computational method obtained is the so-called 4D-var algorithms. Few variants of the so-called "4D-var" algorithms are presented (3D-var, incremental method). Few examples from recent research studies in fluid mechanics (shallow-water model) and in geophysics are presented. Finally, in the linear case (the forward model is linear), the equivalency between the Best Linear Unbiassed Estimate (BLUE) and the variational optimal solution is shown.

4.1 Introduction

Data assimilation is the science combining "at best" the three knowledges of a system: the model (mathematical equations), the reality (observations - measurements) and some statistical errors (errors of measurements, model errors).

Data assimilation refers to two different (and complimentary) approaches:

1) *filtering* such as the Kalman's filter (roughly, we calculate the Best Linear Unbiased Estimate -BLUE- using algebric calculations) or the Ensemble Kalman Filter (EnKF). It is stochastic methods.

2) *variational data assimilation* based on the present control theory. It is more deterministic methods, even if one can introduce statistical errors too. The basic principle is to minimize a cost function which measures the misfit between the output of the model and the measurements.

Both approaches lead to the same estimate in the linear case. At the end of the chapter, we show the equivalency between the (basic) Best Linear Unbiased Estimate (BLUE) and the

variational approach in the linear-quadratic pde case. This equivalency show us how one may introduce the statistics of errors in the variational data assimilation processes.

Both approaches present similar difficulties if the direct dynamical model is non-linear and with large dimensions. In other respect, one never really known the a-priori statistics error... Let us recall that the present course addresses only the variational data assimilation approach. One can cite a couple of historical research articles, [17, 10].

4.2 Forward model

4.2.1 Linear parabolic example: diffusion

The typical simplest (linear) parabolic model is the 'heat equation'. This equation models any diffusion phenomena in a homogeneous media (diffusion phenomena occurs in all applications -e.g. fluids, structures, chemical, images, finance etc-). See the diffusion model of the previous chapter but time-dependent.

4.2.2 Non-linear hyperbolic system: Saint-Venant equations (shallow-water models)

A famous model widely used for geophysical shallow flows, or industrial film flows, is the shallow-water model (or Saint-Venant equations).It offers a very interesting "reduced dimension" approach for fluid flows occurring in thin geometries i.e. geometries with a ratio between the characteristics depth and the characteristic length very small: $\varepsilon = H/L << .$ ε lower than 10^{-2} in practice.

Such geometries are common both in industrial flows (eg coating, thin films on a structure) and in geophysical flows (eg oceans, rivers, glaciers, lavas and historically for atmosphere). For environmental flow modeling, we refer the reader to [24].

Shallow-water models are non-linear hyperbolic systems, almost identical to the Euler's equations in compressible fluid mechanics.
A first derivation of the shallow equations has been done by Barre de Saint-Venant (1871) for river hydraulics. Researches are still active in the topic (in order to include non-linear turbulence terms etc). Roughly, these equations are obtained as follows. The Navier-Stokes equations free-surface, incompressible, are depth-integrated; next, one develop asymptotic expansions with respect to the small (geometric) parameter ε. At the end, instead of a 3d Navier-Stokes free-surface model (mobile computational domain), we obtain a 2d model posed in a fixed domain; thus the reduction of dimension and the computational time saving.

The variable are classically (h, q) the depth and the discharge of the flow. With such an assumption the pressure is not a direct variable anymore (h plays somehow the role of p).The

pressure value is nearly the hydrostatic value $\rho g h$.

Let us present the 2d shallow water equations in their conservative formulation. The state variables are the water depth h and the local discharge $\mathbf{q} = h\mathbf{u}$, where $\mathbf{u} = (u, v)^T$ is the depth-averaged velocity vector. On a 2d domain Ω and for a computational time interval $[0, T]$, the equations are:

$$\left\{\begin{array}{l} \partial_t h \ + \ \mathrm{div}(\,\mathbf{q}\,) \ = \ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{in } \Omega\times]\,0,T] \\ \partial_t \mathbf{q} \ + \ \mathrm{div}\left(\frac{1}{h}\,\mathbf{q}\otimes\mathbf{q}\right) \ + \ \frac{1}{2}g\nabla h^2 + \ gh\nabla z_b \ = \ S_f \quad \text{in } \Omega\times]\,0,T] \\ h(0) \ = \ h_0, \qquad \mathbf{q}(0) \ = \ \mathbf{q}_0 \\ + \text{ boundary conditions} \end{array}\right. \tag{4.1}$$

where g is the magnitude of the gravity, z_b the bed elevation, h_0 and q_0 the initial conditions for the state variables. The quantity $c = \sqrt{gh}$ denotes the local wave celerity.

In river hydraulics, we introduce the *empirical* source term S_f modeling the *friction* at bottom. It writes:

$$S_f = g\frac{n^2\|\mathbf{q}\|_2}{h^{7/3}}$$

It is the Manning-Strickler law, with the scalar parameter n to be tuned... Classically it is tuned by hand but potentially using the present optimal control process ! (see numerical examples in next section).

Without the friction term S_f, these equations constitute a non-linear hyperbolic system (see your course on the subject). They are very similar to the Euler equation (compressible flow, unviscid, in a fixed domain).

This system must be closed with boundary conditions. Typical B.C. are as follows:
- at inflow, inflow discharge is imposed.
- at outflow, incoming characteristic are imposed (or rating curves, or at worth Neumann conditions),
- at obstacle ('lateral boundaries'), wall conditions are imposed: $u.n = 0$.

Let us point out that the solution $(h, \mathbf{q})$ of the shallow-water system can develop discontinuities and shocks in finite time and even with a regular initial condition. Hence, the solution can locally *be not* differentiable with respect to parameters (e.g. the initial condition, n) ... Keeping in mind such a warning, we will however develop later an adjoint approach on this system.

4.2.3 Formal direct model

Let V be a Hilbert space (e.g. the Sobolev space $H_0^1(\Omega)$). For a first order PDEs system in time (eg. the heat equation, the Navier-Stokes equations or the Saint-Venant model), we set:

$$W(0,T) \ = \ \left\{ f \middle| f \in L^2(0,T;V),\ \partial_t f \in L^2(0,T;V') \right\} .$$

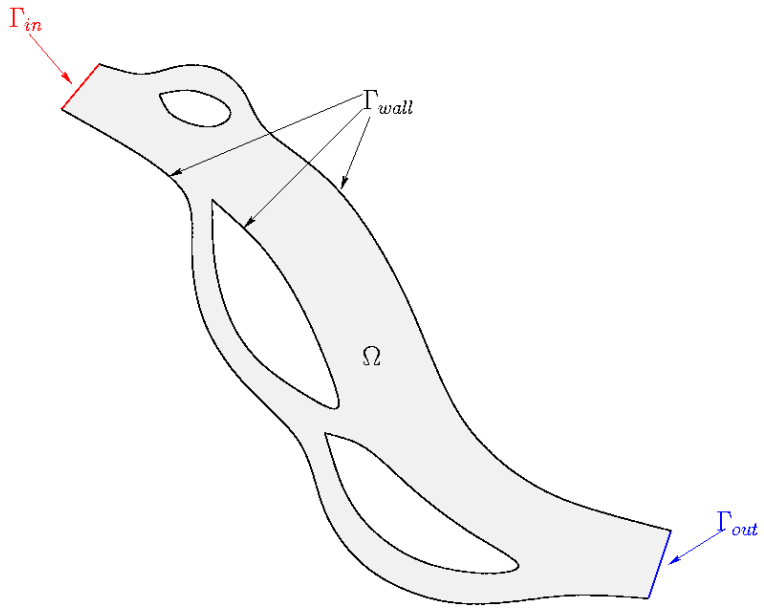

Figure 4.1: Example of application of the 2d shallow water model to river hydraulics (3d free-surface flow computed in a 2d fixed domain). The computational domain and its boundaries.

We denote the state of the system by y. We have: $y \in W(0,T)$.

Let $\mathcal{U}$ the space of parameters. it is supposed to be a Hilbert space. Since parameters can depend on time, we introduce the space $\mathcal{U}_T = L^2(0,T;\mathcal{U})$ function time dependent with value in $\mathcal{U}$, and with its power 2 integrable. It is a Hilbert space, with its norm defined by:

$$\langle u_1, u_2 \rangle_{\mathcal{U}_T} \;=\; \int_0^T \langle u_1(t), u_2(t) \rangle_{\mathcal{U}} dt$$

The state of the system is solution of the direct model:

$$\boxed{(D) \left\{ \begin{array}{l} \text{Given } (y_0, v) \in H \times \mathcal{U}_T, \text{ find } y \in W(0,T) \text{ such that :} \\ \partial_t y(t) \;+\; A\big(v(t)\,;y(t)\big) \;=\; 0 \qquad \forall\, t \in \,]0,T[\\ y(0) \;=\; y_0 \end{array} \right.} \tag{4.2}$$

where $A : \mathcal{U} \times V \longrightarrow V'$ is partial derivatives operator, $y_0 \in H$ is the initial condition, and $v(t) \in \mathcal{U}$ is the parameter at time t.

The solution y depend on y_0 and v. We set the control variable:

$$\mathbf{k} = (y_0, v)^T$$

$\mathbf{k}$ belongs to the space: $\mathcal{K} = H \times \mathcal{U}_T$. In the following, we will denote the state as follows: $y(\mathbf{k};t)$, $y(t)$, $y(\mathbf{k})$ or even y, depending on the context.

We assume that the direct model is well posed in the following sense:

Assumption 4.2.1. *Given* $\mathbf{k} \in \mathcal{K}$ *and* $T > 0$*, it exists an unique function* $y \in W(0,T)$ *solution of the problem* (D)*. Furthermore, this unique solution depends continuously of* $\mathbf{k}$*. In other words, the mapping* $\mathcal{K} \to V \ : k = (y_0, v) \mapsto y(\mathbf{k}; t)$ *is continuous for all* $t \in]0,T[$*.*

Next, we assume the *differentiability* of the solution with respect to parameters.

Assumption 4.2.2. *The mapping* $\mathcal{K} \to V \ : \ \mathbf{k} = (y_0, v) \mapsto y(\mathbf{k}; t)$ *is continuously differentiable for all* $t \in]0,T[$*.*

Under assumption above, one can write:

$$y(\mathbf{k} + \delta\mathbf{k}; t) \ = \ y(\mathbf{k}; t) + \frac{dy}{dk}(\mathbf{k}; t) \cdot \delta\mathbf{k} + o\big(\|\delta\mathbf{k}\|_{\mathcal{K}}\big)$$

This assumption of differentiability is necessary in next section to obtain a differentiable cost function and calculate its gradient. This differentiability is not always satisfied (at least globally), for example for non-linear hyperbolic systems (e.g. the shallow-water system)... To prove such a result of differentiability a useful mathematical result is the implicit function theorem (roughly, the operator and the linearized model must be regular in the sense C^1).

4.2.4 Observations and cost function

Variational data assimilation aim to identify variables (control variables) which minimize the discrepancy between some outputs of the model and observations denoted by y^{obs}. The observations are not necessarily the state of the system. Thus we introduce the *observation operator* C. C maps the state of the system y onto the observation space $\mathcal{O}$ (it is supposed to be a Hilbert space).

Linear observation operator

If C is linear, we have: $C \in \mathcal{L}(V, \mathcal{O})$.

Then, a typical observation function J is as follows:

$$J\big(\mathbf{k}\,; y(\mathbf{k})\big) \ = \ \frac{1}{2}\int_0^T \Big\| \ C\,y(\mathbf{k}\,; t) - y^{obs}(t)\Big\|_{\mathcal{O}}^2 \, dt \ + \ \frac{1}{2}\big\langle N(y_0 - y_b), y_0 - y_b\big\rangle_H \tag{4.3}$$

Next, the cost function to be minimized is defined from the observation function J as follows:

$$j(\mathbf{k}) \ = \ J\big(\mathbf{k}\,; y(\mathbf{k})\big) \tag{4.4}$$

The cost function is defined from $\mathcal{K}$ into $\mathbb{R}$.

The first term measures the discrepancy between the output of the model and the observations, *in the observation space*. The norm $\|.\|_{\mathcal{O}}$ is defined from the inverse covariance error matrix, see section 'statistical interpolation'; it models some a-priori knowledge we may have on the system. In other words, the first term measures the deviations of the 'analysis value' from the observations, weighted by the a-priori confidence we have on the observations.

The second term measures the discrepancy between the initial state (which need to be identified) and a *first guess* given by the modeler.
The *first guess* is a first estimate of the initial condition we think a-priori not too bad (an almost expected initial state). Its choice is crucial to 'select' the local minimum the algorithm will potentially converge to.

N is an endomorphism symmetrical positive define. It is classically builded from the inverse covariance error matrix, see section 'statistical interpolation'.
As previously, it models some a-priori knowledge we may have on the initial state.

Tykhonov's regularization. The second term of the observation function J, see (4.3), can be viewed as a *regularization term* of the inverse problem; it is classically called the Tykhonov's regularization. As a matter it is a *quadratic term* with respect to k. It improves the conditioning of the inverse problem, and somehow it determines the local minimum a descent algorithm will converge to. Hence its choice is essential.

In a practical point of view, we have two classical (and empirical) ways to define it.

. The term measures the discrepancy between the current control and a-priori good first guess. The latter is based on a good understanding of the physical system.

. The term measures the "irregularity" of the control variable. Typically, one can be interested into a control with a minimal 1st or 2nd order variation.

Let us recall that if the direct model is linear (i.e. the mapping $k \mapsto y(k)$ is linear), and if the observation operator is linear, then the cost function j is convex and even quadratic in presence of the Tykhonov regularization term. Thus it admits an unique minimum in a convex set.
If the model is non-linear, a-priori, this result does not hold anymore.

Principle of variational data assimilation The optimal control writes as in the steady-state case:

$$\left\{ \begin{array}{l} \text{Find } \mathbf{k}^* \in H \times \mathcal{U}_T \text{ such that:} \\ j(\mathbf{k}^*) = \min\limits_{H \times \mathcal{U}_T} j(\mathbf{k}) \end{array} \right. \tag{4.5}$$

The minimization of this cost function has the effect that the "analysis" does not drift too far away from observations. Then, the model should be good (at least better...) for forecast.

Observation operator non-linear and time dependent

The most general case is to have a non-linear observation operator C, and time-dependent. Hence, $C\colon]0,T[\times V \to \mathcal{O}$. In such a case, the cost function writes:

$$J\big(\mathbf{k}\,;y(\mathbf{k})\big) \;=\; \frac{1}{2}\int_0^T \Big\| C\big(y(\mathbf{k}\,;t)\big) - y^{obs}(t)\Big\|_{\mathcal{O}}^2 \, dt \;+\; \frac{1}{2}\big\langle N(y_0 - y_b), y_0 - y_b\big\rangle_H \;.$$

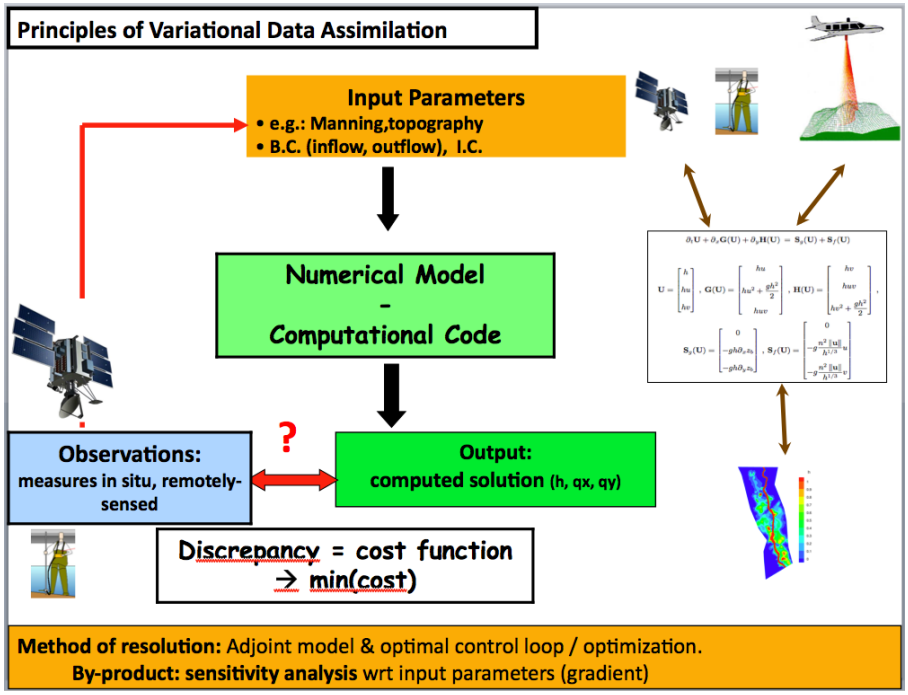

Figure 4.2: Goal of variational data assimilation: make fit observations with model outputs as best as possible.

Isolated observations

Often, one observes the physical system with a finite number sensors at few locations only (ie isolated).
If these observations are continuous in time, the cost function writes:

$$J\big(\mathbf{k}\,;y(\mathbf{k})\big) \;=\; \frac{1}{2}\int_0^T \Big\| Cy(\mathbf{k};t) - y^{obs}(t)\Big\|^2_{\mathbb{R}^M} dt\ .$$

At contrary, if these observations are punctual, the cost function writes:

$$J\big(\mathbf{k}\,;y(\mathbf{k})\big) \;=\; \frac{1}{2}\sum_{i=1}^{N} \Big\| C_i\, y(\mathbf{k};t_i^{obs}) - y_i^{obs}\Big\|^2_{\mathcal{O}}$$

where $C_i \in \mathcal{L}(V,\mathcal{O})$ is the observation operator, y_i^{obs} is the measurement at time t_i^{obs}, $i = 1,\dots,N$. Thus, we have: $y^{obs} = \big\{y_i^{obs}\big\}_{i=1,\dots,N} \in \mathcal{O}^N$.

4.3 Linearized model

4.3.1 The linear tangent model

The linearized model describes the dynamic of the derivative of the state (solution of (D)) for a perturbation δk given. We assumed that the (unique) state of the system $y(k)$ is differentiable with respect to k.

Given a perturbation $\delta k \in \mathcal{K}$, the derivative of $y(k)$ in the direction δk (Gâteaux's derivative) satisfies:

$$\widehat{dy}(\mathbf{k}, \delta\mathbf{k}) \equiv \tfrac{dy}{dk}(\mathbf{k}) \cdot \delta\mathbf{k} \;=\; \tfrac{\partial y}{\partial y_0}(\mathbf{k}) \cdot \delta y_0 \;+\; \tfrac{\partial y}{\partial v}(\mathbf{k}) \cdot \delta v$$

For a sake of simplicity, we will denote: $dy \equiv \widehat{dy}(\mathbf{k}, \delta\mathbf{k})$.

Let us derive the direct model (D) with respect to k in a direction δk (and at the 'point' $y(k)$). We obtain the so-called tangent linear model which writes:

$$(LT) \left\{ \begin{array}{l} \text{Given } k = (y_0, v) \in H \times \mathcal{U}_T \text{ and } y(k) \in W(0,T) \text{ solution of the direct problem } (D), \\ \text{given } \delta k = (\delta y_0, \delta v) \in H \times \mathcal{U}_T, \text{ find } dy \in W(0,T) \text{ such that:} \\ \partial_t\, dy(t) \;+\; \frac{\partial A}{\partial y}\big(v(t), y(t)\big) \cdot dy(t) \;=\; -\,\frac{\partial A}{\partial v}\big(v(t), y(t)\big) \cdot \delta v(t) \qquad \forall t \in \,]0,T[\\ dy(0) \;=\; \delta y_0 \end{array} \right. \tag{4.6}$$

Remark 4.3.1. *If the operator A is linear with respect to the state variable y (i.e. a linear model), then we have: $\frac{\partial A}{\partial y}\big(v(t), y(t)\big) \cdot dy(t) = A(v(t), dy(t))$ and obviously the linearized tangent model is the same as the direct model (modulo the initial condition and the right hand side).*

Remark 4.3.2. *In order to compute $dy = \widehat{dy}(\mathbf{k}, \delta\mathbf{k})$, we can solve the linearized problem (LT). But for a different perturbation δk^*, one must solve again the linearized problem (LT) in order to obtain $\widehat{dy}(\mathbf{k}, \delta\mathbf{k}^*)$!*
That is why we introduce below an auxiliary problem (the adjoint model) which circumvents this difficulty.

4.3.2 A first expression of dj

In order to solve numerically the minimization problem, we will use a descent algorithm. Hence, one need to evaluate the differentiable of j at any point k and in any direction δk i.e. $\widehat{dj}(\mathbf{k}, \delta\mathbf{k})$. It is what we call classically but wrongly 'the gradient'. (As a matter of fact, it is in fact the differentiable; the term gradient refers usually to a vector, i.e. in a finite dimension space).

From (4.4) , we obtain :

$$\widehat{dj}(\mathbf{k}, \delta\mathbf{k}) \;=\; \partial_k J\big(\mathbf{k}\,; y(\mathbf{k})\big).\delta k + \partial_y J\big(\mathbf{k}\,; y(\mathbf{k})\big).\widehat{dy}$$

Hence

$$\widehat{dj}(\mathbf{k}, \delta\mathbf{k}) \;=\; \int_0^T \Big\langle\, C^*\Lambda_{\mathcal{O}}\big(C\,y(t) - y^{obs}(t)\big),\, dy(t) \,\Big\rangle_{V'\times V} dt \;+\; \big\langle N(y_0 - y_b), \delta y_0 \big\rangle_H \tag{4.7}$$

where $\langle \cdot, \cdot \rangle_{V'\times V}$ denotes the duality product, $\Lambda_{\mathcal{O}}$ is the canonical isomorphism from $\mathcal{O}$ into $\mathcal{O}'$, $C^* \in \mathcal{L}(\mathcal{O}', V')$ is the adjoint operator of C. The latter is defined by:

$$\forall\, \eta \in \mathcal{O}', \quad \forall\, \xi \in V \qquad \big\langle\, C^*\eta,\, \xi \,\big\rangle_{V'\times V} \;=\; \big\langle\, \eta,\, C\xi \,\big\rangle_{\mathcal{O}'\times\mathcal{O}}\,.$$

Let us point out that the present expression of $\widehat{dj}(\mathbf{k}, \delta\mathbf{k})$ depends explicitly on $\widehat{dy}(\mathbf{k}, \delta\mathbf{k})$. Thus, in a descent algorithm since we need to evaluate $\widehat{dj}(\mathbf{k}, \delta\mathbf{k})$ for large number of direction δk, we

must a-priori solve as much linear tangent model (LT) as the dimension of δk (discrete version). Such an approach would be unaffordable in a time computation point of view. To avoid this problem, we introduce below the adjoint equations (like in the previous chapter).

As previously, for a sake of simplicity, we will denote: $dj = \widehat{dj}(\mathbf{k}, \delta\mathbf{k})$.

4.4 The adjoint model, gradient and optimality system

We write below the so-called adjoint equations (or adjoint model).The adjoint equations allow to obtain an expression of $\widehat{dj}(\mathbf{k}, \delta\mathbf{k})$ linear dependent with respect to δk, thus evaluable *for any* direction δk.

4.4.1 The adjoint equations

The starting point to derive the adjoint equations is to write:

$$\int_0^T < (LT), p >_{V'\times V} dt \text{ then an integration by part in time etc}$$

It gives:

$$\int_0^T \big\langle \partial_t\, dy(t), p(t)\big\rangle_{V'\times V}\, dt \;+\; \int_0^T \big\langle \tfrac{\partial A}{\partial y}\cdot dy(t), p(t)\big\rangle_{V'\times V}\, dt + \int_0^T \big\langle \tfrac{\partial A}{\partial v}\cdot \delta v, p(t)\big\rangle_{V'\times V}\, dt \;=\; 0$$

After an integration by part, and by making appear the adjoint operators, we obtain:

$$\int_0^T \big\langle \partial_t\, p(t) - \big[\tfrac{\partial A}{\partial y}\big]^* p(t), dy(t)\big\rangle_{V'\times V} dt \;= \big\langle p(T), dy(T)\big\rangle_V \;-\; \big\langle p(0), \delta y_0\big\rangle_V \;+\; \int_0^T \big\langle \big[\tfrac{\partial A}{\partial v}\big]^* p(t), \delta v(t)\big\rangle_{\mathcal{U}'\times\mathcal{U}} dt \tag{4.8}$$

Let us now introduce the "right equations" to make disappear all terms in $dy(t)$.

$$(A)\begin{cases} \text{Given } k=(y_0,v)\in\mathcal{K} \text{ and } y(k)\in W(0,T) \text{ solution of the direct problem } (D),\\ \text{find } p\in W(0,T) \text{ such that:}\\ \partial_t\, p(t) - \Big[\tfrac{\partial A}{\partial y}\big(y(t),v(t)\big)\Big]^* p(t) = \partial_y J\big(\mathbf{k}\,; y(t)\big) \quad \forall t\in\,]0,T[\\ p(T) \;=\; 0 \end{cases} \tag{4.9}$$

with $\partial_y J\big(\mathbf{k}\,; y(t)\big) = C^*\Lambda_{\mathcal{O}}\big(C\,y(t) - y^{obs}(t)\big)$.
It is the adjoint equations (adjoint model).

4.4.2 The gradient expression

By combining equations (4.7), (4.8) (4.9), we obtain:

$$\widehat{dj}(\mathbf{k},\delta\mathbf{k}) \;=\; \big\langle N(y_0-y_b)-p(0),\delta y_0\big\rangle_H + \int_0^T \big\langle \big[\tfrac{\partial A}{\partial v}\big]^* p(t),\delta v(t)\big\rangle_{\mathcal{U}'\times\mathcal{U}}\,dt \tag{4.10}$$

The expression (4.10) of dj does not depend anymore on dy. It depends linearly on δk. Thus, we obtain explicit expressions of the "gradient" i.e. the two components of dj with respect to y_0 and v:

$$\frac{\partial j}{\partial y_0}(\mathbf{k}) \;=\; -\,p(0) \;+\; N(y_0-y_b) \tag{4.11}$$

$$\frac{\partial j}{\partial v}(\mathbf{k}) \;=\; \left[\frac{\partial A}{\partial v}\Big(y(\mathbf{k}),v\Big)\right]^* p(\mathbf{k}) \tag{4.12}$$

Some important remarks. Let us point out that:

- the adjoint model is linear (whatever if the direct model is linear or not).

- the adjoint model is retroacting in time; its initial condition must be given at final time T.

- the sensitivity with respect to the initial condition equals the adjoint state at initial time (modulo the background term).

- the adjoint model contains a sort of *nudging* term. As a matter of fact, its right hand side writes:

$$C^*\Lambda_{\mathcal{O}}(Cy(t)-y^{obs}(t))$$

In other words, the source term of the adjoint model is a measurement of the discrepancy between the computed state and the observations.
NB. The terminology 'nudging' denotes a historical and basic data assimilation method for a dynamical system (historically applied to meteorology). The idea is to add in the right hand side of the dynamical equations a source term measuring the misfit of the model trajectory.
In the historical method, one introduced this extra term with a negative sign in order to "keep" the calculated state vector closer to the observations.

Exercice 4.4.1. *Write the adjoint equations of your practical work , in the unsteady case.*

4.4.3 The local sensitivity analysis: a precious information

As mentioned above, the adjoint method allows to evaluate dj independently of the linearized state dy, and to obtain explicit expressions of all "gradient" components of dj. This property is essential in order to implement a descent algorithm.
In other respect, computing a gradient even without performing a descent algorithm, presents a

huge interest in a modeling point of view. As a matter of fact, the gradient $\widehat{dj}(\mathbf{k}, \delta\mathbf{k})$ represents the *local sensitivity* of the output of the model j with respect the parameters k.
Such an information is precious for hand-users which seek to model and simulate a complex system (whatever if physical, industrial or geophysical).
The gradient helps both to understand the important parameters of the model and to improve the understanding of the system.

Nevertheless this gradient - sensitivity is local in the sense that it is relative to the given point k_0 only, and related to the observations used only ....

Let us remark that this idea of computing sensitivities in order to better understand both the model and the physics, can be applied without observations but on cost functions depending on the state of the system only (e.g. in view to study system stability for example).
Typically, one can be interested to quantify the sensitivity of the following response:

$$j(k) = \frac{1}{2}\int_{\omega} \|\nabla y\|^2_{\mathbb{R}^N} dx$$

with ω a subset of Ω.

4.4.4 Optimality system

The 1st order optimality system As previously (steady-state system in the previous chapter), we can define the optimality system. It is the set of equations characterizing the optimal solution: the state equation, the adjoint state equation and the first order necessary optimality condition (the gradient must vanishes).

Towards 2nd order sensitivities No time enough to develop.

4.4.5 Extension: imperfect model and extra blind control variable

When the flow modeled by the equations is highly non-linear, multi-scale (either in space or/and in time) the models we elaborate are incomplete, imperfect. For complex flows, models do not take into account all phenomena, all scales. Thus when performing a data assimilation process, which consist to make fit as best as possible the imperfect model with real data, we can obtain undesirable (or unrealistic) control variable values...
To prevent this problem, a potential very basic approach consists to add an extra "blind" source term in the model, which will be controlled simultaneously with the other control variables.
The modified direct model writes:

$$(D_{imp})) \left\{ \begin{array}{l} \text{Given } \mathbf{k} = (y_0, v) \in H \times \mathcal{U} \text{ find } y \in W(0,T) \text{ such that:} \\ \partial_t\, y(t) \;+\; A\big(y(t)\big) \;=\; B\, v \qquad \forall\, t \in\,]0,T[\\ y(0) \;=\; y_0 \end{array} \right. \tag{4.13}$$

Where v is part of the control variable. This extra source term v would model blindly some uncertainties of the model. Next, one can easily write as previously the corresponding adjoint equations and the gradient expression.

4.5 The computational algorithms

4.5.1 The 4D-var algorithm

The so-called 4D-var algorithm is the time-dependent extension of the optimal control algorithm presented in the previous chapter. For unsteady PDEs systems (a-priori in 3 space dimension plus time thus 4D variable), we showed previously that the adjoint model is time dependent and reverse in time.
In the case, we solve the optimal control problem using a local descent algorithm (BFGS typically), the algorithm would write as follows, see Fig. (4.3).
Given a *first guess* k_0, we seek iterates $(k^m)_m$ which make decrease the cost function using a first order descent algorithm (here the Quasi-Newton method BFGS). To do so, at each iterate:

`1)` compute the cost function by solving the direct model from 0 to T,

`2)` compute the gradient by solving the adjoint model from T to 0,

`3)` given the current iterate k^n, the current cost function value $j(k^n)$ and the current gradient value $\nabla j(k^n)$, compute a new iterate k^{n+1} such that:

$$j(k^{n+1}) < j(k^n)$$

`*)` Iterate till convergence

Let us point out that the 4D-var method can be used with two different goals:

1) Estimate the value of input parameters (time dependent or not), it is a *identification problem*.
In such a case, immediate prediction is not necessarily the goal (see the applications in river hydraulics and glaciology presented in next chapter for example).

2) Calibrate the model at best in order to make a better prediction (a 'real-time' good forecast is the goal). In this case, the data assimilation proceeds by *analysis cycles*. One cycle would be the assimilation process represented in Fig. 4.4. In each analysis cycle, observations (of present and past time) are assimilated to obtain the so-called analysis (i.e. the optimal estimate of the state). This stage is called the *analysis step*.
Next, the model is advanced in time (*forecasting step*). Next, the forecast is used in the next analysis cycle etc

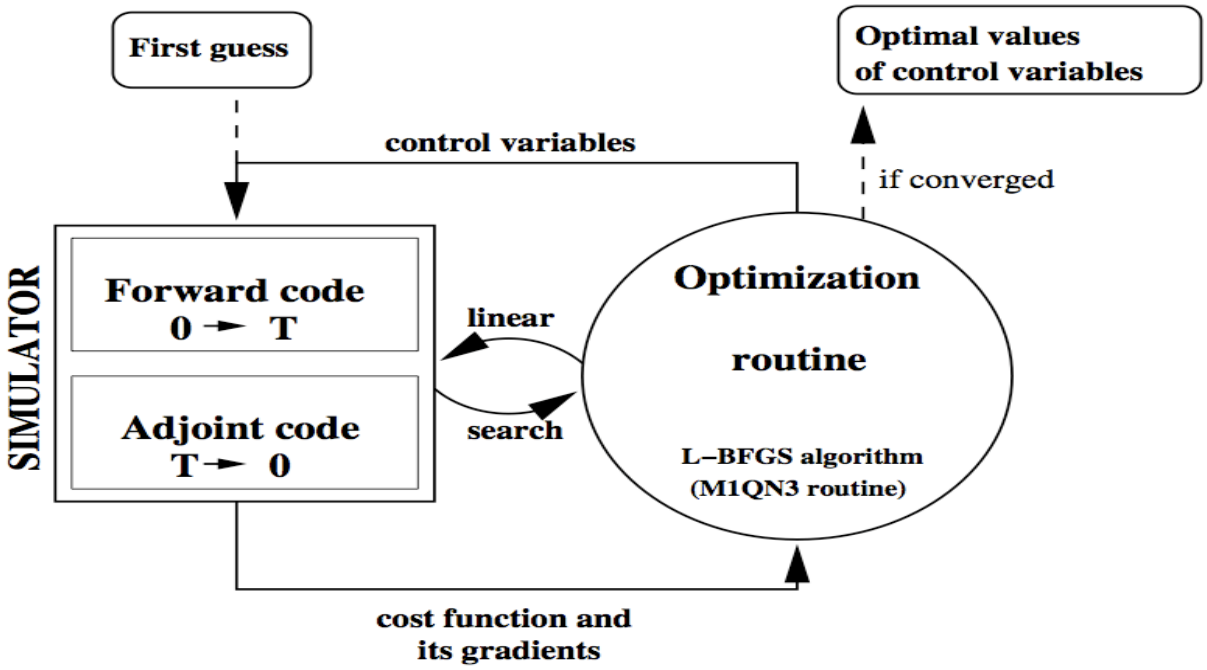

Figure 4.3: Algorithm of optimal control of the PDE system (identification process). 4d-var algorithm.

4.5.2 A discrete presentation of the optimality system

Let us rewrite the variational data assimilation problem and the resulting optimality system in its discrete form (for such a presentation, the reader can refer for example to [27] or the ECMWF data assimilation course available on-line (www.ecmwf.int)).

The direct model writes:

$$(D_d) \begin{cases} \text{Given the initial condition } y_0 \in \mathbb{R}^n, \text{ find } \{y_k\}_k \in \mathbb{R}^n \text{ such that :} \\ y_{n+1} = \mathcal{M}_n(y_n) \quad n = 0, 1, ..., N \text{ (time steps)} \\ y_0 \text{ given} \end{cases} \tag{4.14}$$

where:
- $\mathcal{M}(y_n)$ denotes the (non-linear) model discretized in space and time; and $M_n = \partial_x \mathcal{M}(y_n)$ is the linear tangent model at y_n (hence M^T the adjoint operator).
- $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}^m$ is the (non-linear) observation operator, and $H_n = \partial_x \mathcal{H}(y_n)$ is the linear tangent observation operator at y_n.

For a sake of simplicity, here we seek to identify the initial condition y_0 only. The cost function writes:

$$j(y_0) = \frac{1}{2} \sum_{n=0}^{N} (\mathcal{H}(y_n) - z_n)^T R^{-1} (\mathcal{H}(y_n) - z_n) + \frac{1}{2} (y_0 - y_b)^T B^{-1} (y_0 - y_b)$$

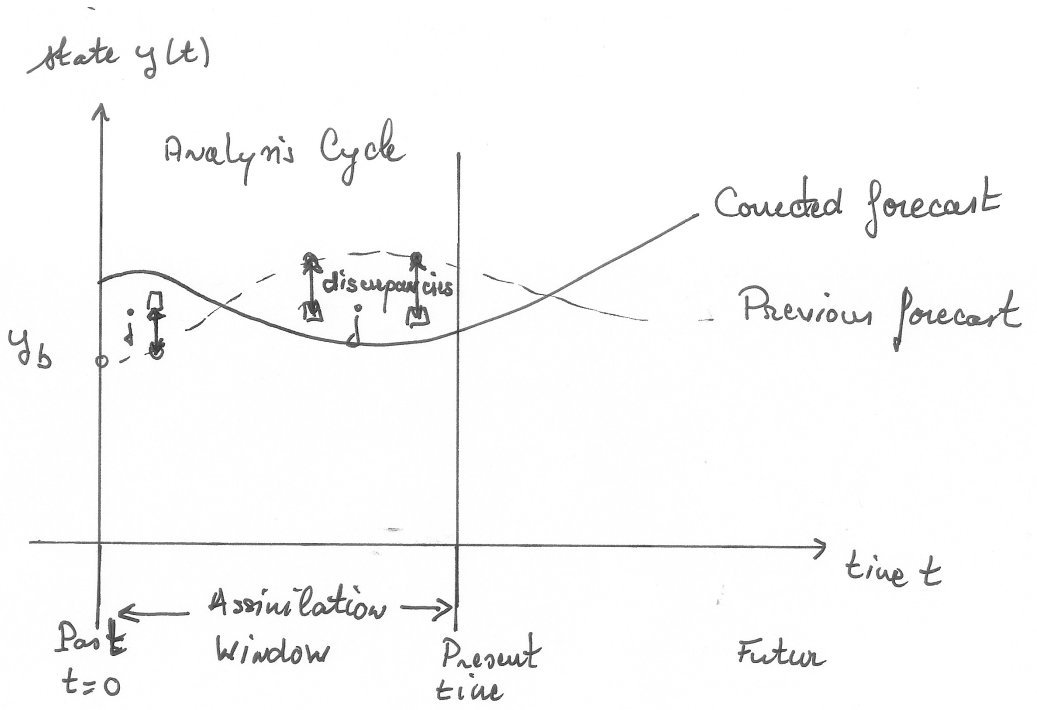

Figure 4.4: Algorithm of 4d-var (here, identification of the I.C. y_0). All observation within the time interval assimilation are assimilated in the global least-square formulation. Next, the resulting calibrated model is runned for prediction (forecast).

with, R the matrix of the covariance observation errors, B the matrix of the covariance background errors, z_n the observation vector at time t_n, y_b the background (first guess).

The corresponding adjoint system writes:

$$(A_d) \begin{cases} \text{Given the states } \{y_k\}_k \in \mathbb{R}^n, \text{ find } \{p_k\}_k \in \mathbb{R}^n \text{ such that :} \\ p_{n-1} = M_n^T(p_n) + H_n^T R^{-1}(\mathcal{H}(y_n) - z_n) \quad n = N, ..., 1 \text{ (time steps)} \\ p_N = H_N^T R^{-1}(\mathcal{H}(y_N) - z_N) \end{cases} \tag{4.15}$$

And the gradient (with respect to the initial condition) writes:

$$\nabla j(y_0) = -p_0$$

Let us remark that the gradient with respect to the initial condition equals the final state (at $t = 0$) of the adjoint.

4.5.3 Variant: incremental 4D-var

For very large scale problems (e.g. oceans and atmosphere in geophysics), it can be unaffordable to perform the 4D-var process for the full model resolution (i.e. the direct model including the fine physics, and solved on the fine grids). Furthermore, if the assimilation is required for

prediction (e.g. weather forecast), the forecast needs to be ready in time !...
For smaller scale problems (e.g. based on already reduced models such as 2D shallow-water equations), even if full 4d-var is possible, it remains CPU time consuming (see examples in river hydraulics next section).

An option to reduce the cost computation is to implement the so-called incremental 4D-var method. The latter is a mix of empirical techniques and linearization.

Roughly, one make the corrections (i.e. the analysis cycle) on a lower resolution model (both in physics and grids), in view to a high resolution background update (i.e. we keep the full fine forward model for prediction).
Let us consider the discrete notation of the previous paragraph.
We denote the *innovation vectors* $\{d_n\}_n$:

$$d_n = (z_n - \mathcal{H}(y_n))$$

The innovation vector d_n measures the discrepancy between the model output and the observed quantity at time t_n.

Let us set: $y_b = y_0 + \delta y_0$, then we have:

$$j(y_0) = \frac{1}{2}\sum_{n=0}^{N}(\mathcal{H}(y_n) - z_n)^T R^{-1}(\mathcal{H}(y_n) - z_n) + \frac{1}{2}\delta y_0^T B^{-1}\delta y_0$$

Let δy_n be a perturbation of the state at time t_n. By linearization (Taylor's expansion order 1), we have:

$$\mathcal{M}_n(y_n + \delta y_n) \approx \mathcal{M}_n(y_n) + M_n \cdot \delta y_n \text{ and } \mathcal{H}(y_n + \delta y_n) \approx \mathcal{H}(y_n) + H_n \cdot \delta y_n$$

where M_n is the linear tangent model at time step t_n and H_n the linearized observation operator.

Recall that a perturbation δy_0 to the initial condition implies a "perturbation" δy_n to the state values y_n, $n = 0, 1, .., N$.
The basic idea of the incremental 4D-var method is to minimize the cost function with respect to increments δy_0. To do so, we replace in the cost function expression, the model and observation operators (a-priori non-linear) by their linearized counterpart:

$$j(y_0) \approx \frac{1}{2}\sum_{n=0}^{N}(H_n \cdot \delta y_n - d_n)^T R^{-1}(H \cdot \delta y_n - d_n) + \frac{1}{2}\delta y_0^T B^{-1}\delta y_0 = g(\delta y_0)$$

Now, the cost function $g(\delta y_0)$ to be minimized is *quadratic*. It is a linear-quadratic optimal control problem, hence much faster to solve. The minimization can be done using the Conjugate Gradient with preconditioning.

Hence, the 4D-var incremental algorithm may write as follows (see Figure 4.5).
*) Inner loop.

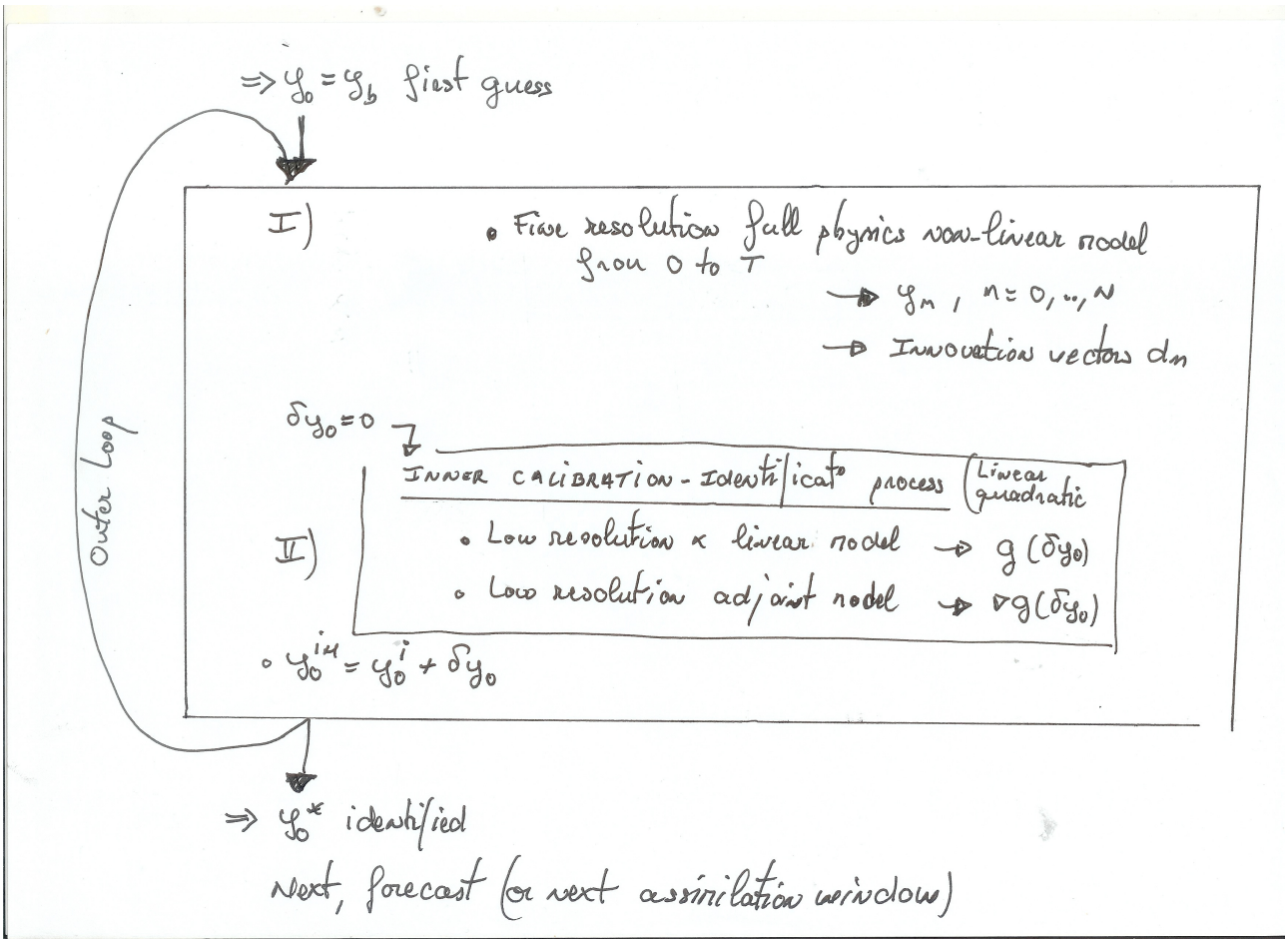

Figure 4.5: Incremental 4d-var algorithm

The minimization process is performed with: coarser grids, linearized operators (M and H) and potentially simplified physics.
Furthermore, since the cost function $g(\delta y_0)$ is quadratic, one can use the efficient and simple Conjugate Gradient method with preconditioning to minimize it.

*) Outer loop.
Once the lower resolution minimization process converged (the analysis step is done), the high resolution model is performed (prediction stage). The prediction stage gives new innovation vectors (the discrepancies between the model and the observations).

Let us point out that the innovation vectors d_n are the crucial input of the assimilation method, thus they are computed using the high resolution model.
(Recall that the high resolution model is the original one with full physics and solve on the finest grids).

*) Updating of the increment and the reference state.

Thus, the incremental 4D-var method is a mix of calibration on linearized operators, full physic predictions and discrepancy measurements (through the innovation vectors).

For more detail and in a weather forecast context, the reader can refer for example to [6, 25].

4.5.4 The simple sequential 3D-var

The 3D-var algorithm is a time-independent version of the 4D-var. Historically, it has been used for time-dependent problems because the 4D-var was too much CPU-time consuming. The 3D-var algorithm can be viewed as follows. We regroup all observations available during a time-interval at the final time, then we compute the discrepancies between the model output and the observations at a single time. Next, the minimization process is performed.
In other words, all observations are grouped together (averaged) at the assimilation time.
Thus, no time integration of the direct model nor of the adjoint model have to be performed; only steady-state assimilations are performed at given time.
Next the unsteady model runs until the next assimilation time.
The drawback is obvious: we compare the model outputs and the observations not at the right times.

Variant: 3D-FGAT. The 3D-FGAT (First Guess at Appropriate Time) is an extra approximation of the 4D-Var incremental. In short, the linear tangent model and adjoint model are replaced by the identity operator. Thus, it is an empirical method between the 3D-var and the 4D-var. The only advantage compared to the 3D-var is the innovation vector is evaluated at the right observation times.

4.5.5 Preconditioning

Let us go back to the previous 4D-var formulation (subsection 4.5.2) but with a linear model $\mathcal{M}$ and a linear observation operator $\mathcal{H}$.
In such a case, the cost function is quadratic and its "convexity is measured" by its Hessian. We have: $D^2j(k) = H^T R^{-1} H + B^{-1}$.

Since j is quadratic, the right method to minimize it is the Conjugate Gradient method. Then, one can prove that the convergence algorithm is determined by the condition number of the Hessian matrix (see your previous courses treating of computational methods for linear systems). Thus, classically we make a preconditioning of the system: we solve an equivalent system but (we hope) with a lower condition number.
In short, in the *incomplete Cholesky preconditioning* case we make the following change of variable:

$$v = L^{-1}k$$

with $B^{-1} = LL^T$ (Cholesky factorization), B^{-1} symmetric positive define. In practice, L is defined with few lower-diagonals only (incomplete Cholesky). Then, the cost function to be minimized writes:

$$G(v) = \frac{1}{2}(HLv - d)^T R^{-1}(HLv - d) + \frac{1}{2}v^T v$$

Its Hessian is: $D^2G = L^T H^T R^{-1} HL + I$. The latter should have a better conditioning number than the original Hessian hence the Conjugate Gradient algorithm converges faster.
In non-linear cases, at best, one knowns to define an empirical change of variables which may improve the convergence speed of the minimization...