Chapter 2

Optimal control and variational data assimilation

2.1 ▪ Introduction

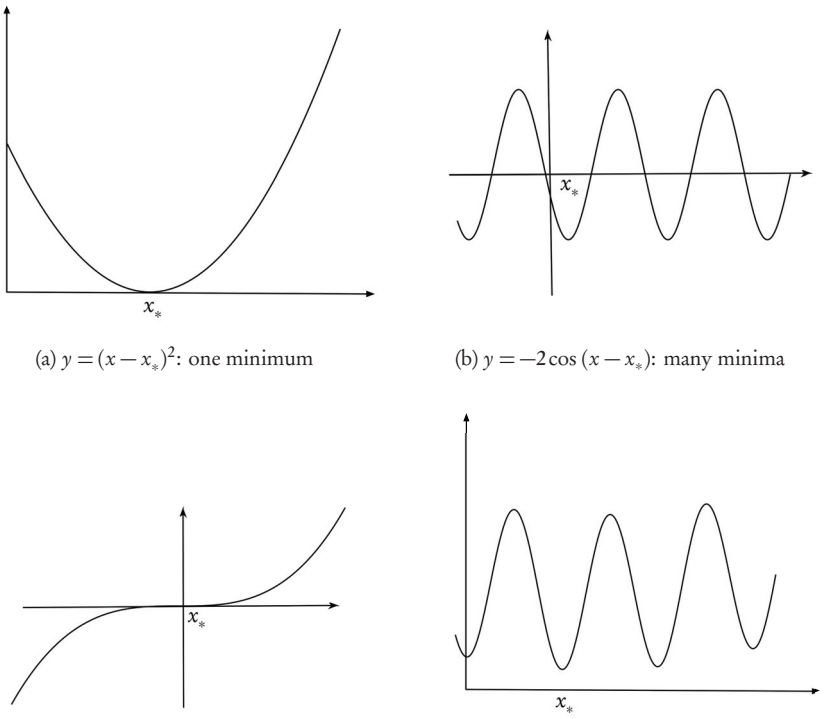
Unlike sequential assimilation (which emanates from statistical estimation theory and will be the subject of the next chapter), variational assimilation is based on optimal control theory.[13] The analyzed state is not defined as the one that maximizes a certain probability density function (PDF), but as the one that *minimizes a cost function.* The minimization requires numerical optimization techniques. These techniques can rely on the *gradient* of the cost function, and this gradient will be obtained here with the aid of *adjoint methods*.

The theory of adjoint operators, coming out of functional analysis, is presented in Kreyszig [1978] and Reed and Simon [1980]. A special case is that of matrix systems, which are simply the finite dimensional operator case. The necessary ingredients of optimization theory are described in Nocedal and Wright [2006].

In this chapter, we will show that the adjoint approach is an extremely versatile tool for solving a very wide range of inverse problems—DA problems included. This will be illustrated via a sequence of explicitly derived examples, from simple cases to quite complex nonlinear cases. We will show that once the theoretical adjoint technique is understood and mastered, almost any model equation can be treated and almost any inverse problem can be solved (at least theoretically). We will not neglect the practical implementation aspects that are vital for any real-world, concrete application. These will be treated in quite some detail since they are often the crux of the matter—that is, the crucial steps for succeeding in solving DA and inverse problems.

The chapter begins with a presentation of the *calculus of variations*. This theory, together with the concept of *ill-posedness*, is the veritable basis of inverse problems and DA, and its mastery is vital for formulating, understanding, and solving real-world problems. We then consider adjoint methods, starting from a general setting and moving on through a series of parameter identification problems—all of these in a differential equation (infinite-dimensional) setting. Thereafter, we study finite-dimensional cases, which lead naturally to the comparison of continuous and discrete adjoints. It is here that we will introduce *automatic differentiation*, which generalizes the calculation of the adjoint to intractable, complex cases. After all this preparation, we will be ready to study the two major variational DA approaches: 3D-Var and 4D-Var. Once

[13] This is basically true; however, 4D-Var applied to a chaotic model is in fact a sequential algorithm.



(a) $y = (x - x_*)^2$: one minimum

(b) $y = -2\cos(x - x_*)$: many minima

(c) $y = x^3$: saddle point

(d) $y = 0.015(x - x_*)^2 - 2\cos(x - x_*)$: one global minimum, many local minima

Figure 2.1. *A variety of local extrema, denoted by* x_*.

completed, we present a few numerical examples. We end the chapter with a brief description of a selection of advanced topics: preconditioning, reduced-order methods, and error covariance modeling. These will be expanded upon in Chapter 5.

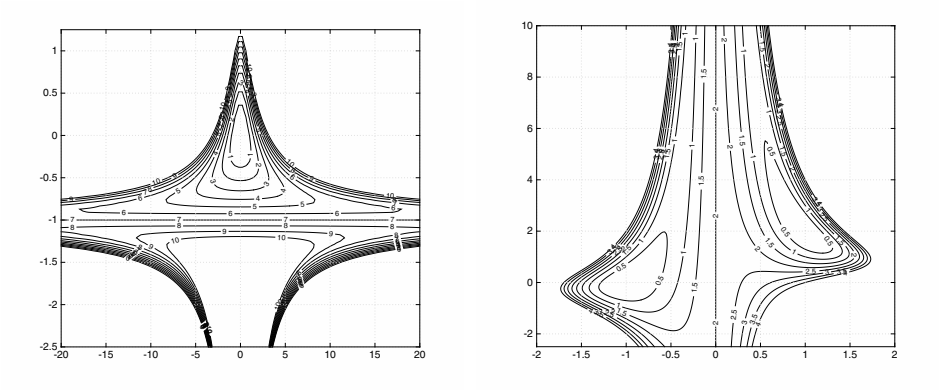
2.2 ▪ The calculus of variations

The calculus of variations is, to quote Courant and Hilbert [1989a], one of the "very central fields of analysis." It is also the central tool of variational optimization and DA, since it generalizes the theory of maximization and minimization. If we understand the workings of this theory well, we will be able to understand variational DA and inverse problems in a much deeper way and thus avoid falling into phenomenological traps. By this we mean that when, for a particular problem, things go wrong, we will have the theoretical and methodological distance/knowledge that is vital for finding a way around, over, or through the barrier (be it in the formulation or in the solution of the problem). Dear reader, please bear with us for a while, as we review together this very important theoretical tool.

To start out from a solid and well-understood setting, let us consider the basic theory of optimization[14] (maximization or minimization) of a continuous function

$$f(x, y, \ldots): \mathbb{R}^d \to \mathbb{R}$$

[14] An excellent reference for a more complete treatment is the book of Nocedal and Wright [2006].



(a) $z = F_1(x,y)$. (b) $z = F_2(x,y)$.

Figure 2.2. *Counterexamples for local extrema in* $\mathbb{R}^2$.

in a closed region Ω. We seek a point $\boldsymbol{x}_* = (x_*, y_*, \ldots) \in \mathbb{R}^d$ in Ω for which f has an extremum (maximum or minimum) in the vicinity of $\boldsymbol{x}_*$ (what is known as a *local* extremum—see Figure 2.1). A classical theorem of Weierstrass guarantees the existence of such an object.

Theorem 2.1. *Every continuous function in a bounded domain attains a maximal and a minimal value inside the domain or on its boundary.*

If f is differentiable in Ω and if $\boldsymbol{x}_*$ is an interior point, then the first derivatives of f, with respect to each of its variables, vanish at $\boldsymbol{x}_*$—we say that the gradient of f is equal to zero. However, this *necessary condition* is by no means sufficient because of the possible existence of saddle points. For example, see Figure 2.1c, where $f(x) = x^3$ at $\boldsymbol{x}_* = 0$. Moreover, as soon as we pass from $\mathbb{R}$ to even $\mathbb{R}^2$, we lose the simple Rolle's and intermediate-value theorem [Wikipedia, 2015c] results, and very counterintuitive things can happen. This is exhibited by the following two examples (see Figure 2.2):

- $F_1(x,y) = x^2(1+y)^3 + 7y^2$ has a single critical point—the gradient of F_1 vanishes at $(0,0)$—which is a local minimum, but not a global one. This cannot happen in one dimension because of Rolle's theorem. Note also that $(0,0)$ is *not* a saddle point.
- $F_2(x,y) = (x^2y - x - 1)^2 + (x^2 - 1)^2$ has exactly two critical points, at $(1,2)$ and $(-1,0)$, both of which are local minima—again, an impossibility in one dimension where we would have at least one additional critical point between the two.

A *sufficient condition* requires that the second derivative of the function exist and be positive. In this case the point $\boldsymbol{x}_*$ is indeed a local minimizer. The only case where things are simple is when we have smooth, convex functions [Wikipedia, 2015d]—in this case any local minimizer is a global minimizer, and, in fact, any stationary point is a global minimizer. However, in real problems, these conditions are (almost) never satisfied even though we will in some sense "convexify" our assimilation and inverse problems—see below.

Now, if the variables are subject to n constraints of the form $g_j(x,y,\ldots) = 0$ for $j = 1,\ldots,n$, then by introducing *Lagrange multipliers* we obtain the necessary conditions

for an extremum. For this, we define an augmented function,

$$F = f + \sum_{j=1}^{n} \lambda_j g_j,$$

and write down the necessary conditions

$$\frac{\partial F}{\partial x} = 0, \quad \frac{\partial F}{\partial y} = 0, \quad \ldots \quad (d \text{ equations}),$$

$$\frac{\partial F}{\partial \lambda_1} = g_1 = 0, \quad \ldots, \quad \frac{\partial F}{\partial \lambda_n} = g_n = 0 \quad (n \text{ equations}),$$

which gives a system of equations (m equations in m unknowns, where $m = d + n$) that are then solved for $\boldsymbol{x}_* \in \mathbb{R}^d$ and λ_j, $j = 1, \ldots, n$.

We will now generalize the above to finding the extrema of functionals.[15] The domain of definition becomes a *space of admissible functions*[16] in which we will seek the extremal member.

> The calculus of variations deals with the following problem: find the maximum or minimum of a functional, over the given domain of admissible functions, for which the functional attains the extremum with respect to all argument functions in a small neighborhood of the extremal argument function.

But we now need to generalize our definition of the vicinity (or neighborhood) of a function. Moreover, the problem may not have a solution because of the difficulty of choosing the set of admissible functions to be compact, which means that any infinite sequence of functions must get arbitrarily close to some function of the space—an *accumulation point*. However, if we restrict ourselves to necessary conditions (the vanishing of the first "derivative"), then the existence issue can be left open.

2.2.1 ▪ Necessary conditions for functional minimization

What follows dates from Euler and Lagrange in the 18th century. Their idea was to solve minimization problems by means of a general variational approach that reduces them to the solution of differential equations. Their approach is still completely valid today and provides us with a solid theoretical basis for the variational solution of inverse and DA problems.

We begin, as in Courant and Hilbert [1989a], with the "simplest problem" of the calculus of variations, where we seek a real-valued function $y(x)$, defined on the interval $[a, b]$, that minimizes the integral cost function

$$J[y] = \int_a^b F(x, y, y')\,\mathrm{d}x. \tag{2.1}$$

A classical example is to find the curve, $y(x)$, known as the *geodesic*, that minimizes the length between $x = a$ and $x = b$, in which case $F = \sqrt{1 + (y')^2}$. We will assume here

[15] These are functions of functions, rather than of scalar variables.

[16] An example of such a space is the space of continuous functions with continuous first derivatives, usually denoted $C^1(\Omega)$, where Ω is the domain of definition of each member function.

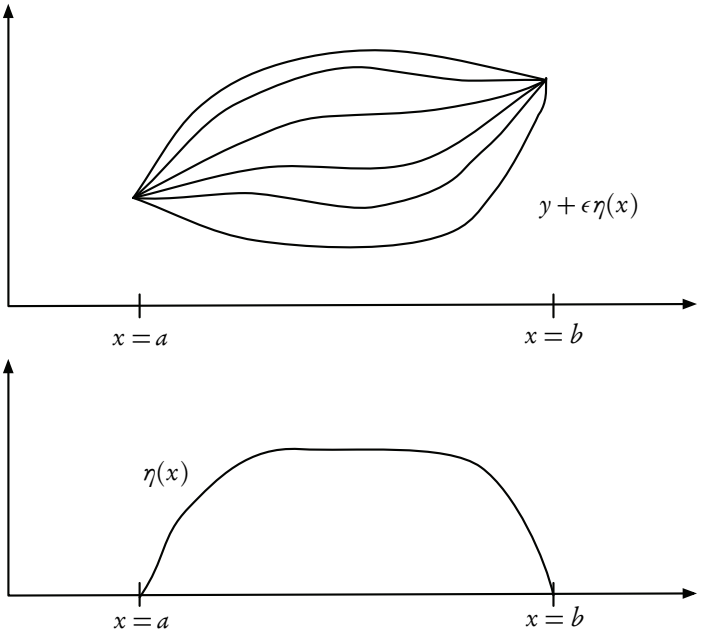


Figure 2.3. *Curve $\eta(x)$ and admissible functions $y + \epsilon\eta(x)$.*

that the functions F and y possess all the necessary smoothness (i.e., they are differentiable up to any order required). Suppose that y_* is the extremal function that gives the minimum value to J. This means that in a sufficiently small neighborhood (recall our difficulties with multiple extremal points—the same thing occurs for functions) of the function $y_*(x)$ the integral (2.1) is smallest when $y = y_*(x)$. To quantify this, we will define the *variation*, δy, of the function y. Let $\eta(x)$ be an arbitrary, smooth function defined on $[a, b]$ and vanishing at the endpoints, i.e., $\eta(a) = \eta(b) = 0$. We construct the new function $\tilde{y} = y + \delta y$, with $\delta y = \epsilon\eta(x)$, where ϵ is an arbitrary (small) parameter—see Figure 2.3. This implies that all the functions $\tilde{y}$ will lie in an arbitrarily small neighborhood of y_*, and thus the cost function $J[\tilde{y}]$, taken as a function of ϵ, must have its minimum at $\epsilon = 0$, and its "derivative" with respect to ϵ must vanish there. If we now take the integral

$$J(\epsilon) = \int_a^b F(x, y + \epsilon\eta, y' + \epsilon\eta')\,\mathrm{d}x$$

and differentiate it with respect to ϵ, we obtain at $\epsilon = 0$

$$J'(0) = \int_a^b \left(F_y\eta + F_{y'}\eta'\right)\mathrm{d}x = 0,$$

where the subscripts denote partial differentiation. Integrating the second term by parts and using the boundary values of η, we get

$$\int_a^b \eta\left(F_y - \frac{\mathrm{d}}{\mathrm{d}x}F_{y'}\right)\mathrm{d}x = 0,$$

and since this must hold for all functions η, we can invoke the following fundamental lemma.

Lemma 2.2. (Fundamental lemma of the calculus of variations). *If* $\int_{x_0}^{x_1} \eta(x)\phi(x)dx = 0$, *with* $\phi(x)$ *continuous, holds for all functions* $\eta(x)$ *vanishing on the boundary and continuous with two continuous derivatives, then* $\phi(x) = 0$ *identically.*

Using this lemma, we conclude that the necessary condition, known as the *Euler–Lagrange equation,* is

$$F_y - \frac{\mathrm{d}}{\mathrm{d}x}F_{y'} = 0. \tag{2.2}$$

We can expand the second term of this expression, obtaining (we have flipped the signs of the terms)

$$y''F_{y'y'} + y'F_{y'y} + F_{y'x} - F_y = 0. \tag{2.3}$$

If we want to solve this equation for the highest-order derivative, we must require that

$$F_{y'y'} \neq 0,$$

which is known as the *Legendre condition* and plays the role of the second derivative in the optimization of scalar functions, by providing a *sufficient* condition for the existence of a maximum or minimum. We will invoke this later when we discuss the study of sensitivities with respect to individual parameters in multiparameter identification problems.

Example 2.3. Let us apply the Euler–Lagrange necessary condition to the geodesic problem with

$$J[y] = \int_a^b F(x, y, y')\,\mathrm{d}x = \int_a^b \sqrt{1 + (y')^2}\,\mathrm{d}x.$$

Note that F has no explicit dependence on the variables y and x. The partial derivatives of F are

$$F_y = 0, \quad F_{y'} = \frac{y'}{\sqrt{1 + (y')^2}}, \quad F_{y'y} = 0, \quad F_{y'y'} = \left(1 + \left(y'\right)^2\right)^{-3/2}, \; F_{y'x} = 0.$$

Substituting in (2.3), we get $y'' = 0$, which implies that

$$y = cx + d,$$

and, unsurprisingly, we indeed find that a straight line is the shortest distance between two points in the Cartesian x-y plane. This result can be extended to finding the geodesic on a given surface by simply substituting parametric equations for x and y. ■

2.2.2 ▪ Generalizations

The Euler–Lagrange equation (2.2) can be readily extended to functions of several variables and to higher derivatives. This will then lead to the generalization of the Euler–Lagrange equations that we will need in what follows.

In fact, we can consider a more general family of admissible functions, $y(x,\epsilon)$, with

$$\eta(x) = \frac{\partial}{\partial \epsilon} y(x,\epsilon)\Big|_{\epsilon=0}.$$

Recall that we defined the variation of y ($\tilde{y} = y + \delta y$) as $\delta y = \epsilon\eta$. This leads to an analogous definition of the (first) *variation* of J,

$$\begin{aligned}\delta J = \epsilon J'(0) &= \epsilon \int_{x_0}^{x_1} \left(F_y \eta + F_{y'}\eta'\right) \mathrm{d}x \\ &= \epsilon \int_{x_0}^{x_1} \left(F_y - \frac{\mathrm{d}}{\mathrm{d}x} F_{y'}\right)\eta \,\mathrm{d}x + \left[\epsilon F_{y'}\eta\right]_{x=x_0}^{x=x_1} \\ &= \int_{x_0}^{x_1} [F]_y \,\delta y \,\mathrm{d}x + \left[F_{y'}\delta y\right]_{x=x_0}^{x=x_1},\end{aligned}$$

where

$$[F]_y \doteq \left(F_y - \frac{\mathrm{d}}{\mathrm{d}x} F_{y'}\right)$$

is the *variational derivative* of F with respect to y. We conclude that the *necessary condition for an extremum* is that the first variation of J be equal to zero for all admissible $y + \delta y$. The curves for which δJ vanishes are called *stationary functions*. Numerous examples can be found in Courant and Hilbert [1989a].

Let us now see how this fundamental result generalizes to other cases. If F depends on higher derivatives of y (say, up to order n), then

$$J[y] = \int_{x_0}^{x_1} F(x, y, y', \ldots, y^{(n)})\,\mathrm{d}x,$$

and the Euler–Lagrange equation becomes

$$F_y - \frac{d}{dx}F_{y'} + \frac{d^2}{dx^2}F_{y''} - \cdots + (-1)^n \frac{d^n}{dx^n}F_{y^{(n)}} = 0.$$

If F consists of several scalar functions $(y_1, y_2, \ldots, y_n)$, then

$$J[y_1, y_2, \ldots, y_n] = \int_{x_0}^{x_1} F(x, y_1, \ldots, y_n, y_1', \ldots, y_n')\,\mathrm{d}x,$$

and the Euler–Lagrange equations are

$$F_{y_i} - \frac{d}{dx}F_{y_i'} = 0, \quad i = 1, \ldots, n.$$

If F depends on a single function of n variables and if Ω is a surface, then

$$J[y] = \int_{\Omega} F(x_1, \ldots, x_n, y, y_{x_1}, \ldots, y_{x_n})\,\mathrm{d}x_1 \cdots \mathrm{d}x_n,$$

and the Euler–Lagrange equations are now PDEs:

$$F_y - \sum_{i=1}^{n} \frac{\partial}{\partial x_i} F_{y_{x_i}'} = 0, \quad i = 1, \ldots, n.$$

Finally, there is the case of several functions of several variables, which is just a combination of the above.

Example 2.4. We consider the case of finding an extremal function, u, of two variables, x and y, from the cost function

$$J[u] = \iint_{\Omega} F(x, y, u, u_x, u_y)\,\mathrm{d}x\,\mathrm{d}y \tag{2.4}$$

over the domain Ω. The necessary condition is

$$\delta J = \epsilon \left(\frac{d}{d\epsilon} J[u + \epsilon\eta] \right)_{\epsilon=0} = 0, \tag{2.5}$$

where $\eta(x,y)$ is a "nice" function satisfying zero boundary conditions on $\partial\Omega$, the boundary of Ω. Substituting (2.4) in (2.5), we obtain

$$\delta J = \epsilon \iint_{\Omega} \left(F_u \eta + F_{u_x} \eta_x + F_{u_y} \eta_y \right) \mathrm{d}x\,\mathrm{d}y = 0,$$

which we integrate by parts (by applying the Gauss divergence theorem), getting

$$\delta J = \epsilon \iint_{\Omega} \eta \left(F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} \right) \mathrm{d}x\,\mathrm{d}y = 0$$

(we have used the vanishing of η on the boundary), which yields the Euler–Lagrange equation

$$[F]_u = F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0.$$

This can be expanded to

$$F_{u_x u_x} u_{xx} + 2F_{u_x u_y} u_{xy} + F_{u_y u_y} u_{yy} + F_{u_x u} u_x + F_{u_y u} u_y + F_{u_x x} + F_{u_y y} - F_u = 0.$$

We can now apply this result to the case where

$$F = \frac{1}{2}\left(u_x^2 + u_y^2 \right).$$

Clearly,

$$F_{u_x u_x} = F_{u_y u_y} = 1,$$

with all other terms equal to zero, and the Euler–Lagrange equation is precisely Laplace's equation,

$$\Delta u = u_{xx} + u_{yy} = 0,$$

which can be solved subject to the boundary conditions that must be imposed on u. ■

2.2.3 ▪ Concluding remarks

The calculus of variations, via the Euler–Lagrange (partial) differential equations, provides a very general framework for minimizing functionals, taking into account both the functional to be minimized and the (partial) differential equations that describe the underlying physical problem. The calculus of variations also covers the minimization of more general integral equations, often encountered in imaging problems, but

these will not be dealt with here. Entire books are dedicated to this subject—see, for example, Aster et al. [2012] and Colton and Kress [1998].

In what follows, for DA problems we will study a special case of calculus of variations and generalize it. Let us explain: the special case lies in the fact that our cost function will be a "mismatch" function, expressing the squared difference between measured values and simulated (predicted) values, integrated over a spatial (or space-time) domain. To this we will sometimes add "regularization" terms to ensure well-posedness. The generalization takes the form of the constraints that we add to the optimization problem: in the case of DA these constraints are (partial) differential equations that must be satisfied by the extremal function that we seek to compute.

2.3 ▪ Adjoint methods

Having, we hope by now, acquired an understanding of the calculus of variations, we will proceed to study the *adjoint approach* for solving (functional) optimization problems. We will emphasize the generality and the inherent power of this approach. Note that this approach is also used frequently in optimal control and optimal design problems—these are just special cases of what we will study here. We note that the Euler–Lagrange system of equations, amply seen in the previous section, will here be composed of the direct and adjoint equations for the system under consideration. A very instructive toy example of an ocean circulation problem can be found in Bennett [2004], where the Euler–Lagrange equations are carefully derived and their solution proposed using a special decomposition based on "representer functions."

In this section, we will start from a general setting for the adjoint method, and then we will back up and proceed progressively through a string of special cases, from a "simple" ODE-based inverse problem of parameter identification to a full-blown, nonlinear PDE-based problem. Even the impatient reader, who may be tempted to skip the general setting and go directly to the special case examples, is encouraged to study the general presentation, because a number of fundamental and very important points are dealt with here. After presenting the continuous case, the discrete (finite-dimensional) setting will be explained. This leads naturally to the important subject of *automatic differentiation*, which is often used today for automatically generating the adjoints of large production codes, though it can be very efficient for smaller codes too.

2.3.1 ▪ A general setting

We will now apply the calculus of variations to the solution of variational inverse problems. Let $\mathbf{u}$ be the state of a *dynamical system* whose behavior depends on model parameters $\mathbf{m}$ and is described by a differential operator equation

$$\mathbf{L}(\mathbf{u}, \mathbf{m}) = \mathbf{f},$$

where $\mathbf{f}$ represents external forces. Define a *cost function*, $J(\mathbf{m})$, as an energy functional or, more commonly, as a misfit functional that quantifies the L^2-distance[17] between the observation and the model prediction $\mathbf{u}(\mathbf{x}, t; \mathbf{m})$. For example,

$$J(\mathbf{m}) = \int_0^T \int_\Omega \left(\mathbf{u}(\mathbf{x}, t; \mathbf{m}) - \mathbf{u}^{\text{obs}}(\mathbf{x}, t)\right)^2 \delta(\mathbf{x} - \mathbf{x}_r)\, d\mathbf{x}\, dt,$$

[17] The space L^2 is a Hilbert space of (measurable) functions that are square-integrable (in the Lebesgue sense). Readers unfamiliar with this should consult a text such as Kreyszig [1978] for this definition as well as all other (functional) analysis terms used in what follows.

where $x \in \Omega \subset \mathbb{R}^n$; $n = 2,3$; $0 \le t \le T$; δ is the Dirac delta function; and $\mathbf{x}_r$ are the observer positions. Our objective is to choose the model parameters, $\mathbf{m}$, as a function of the observed output, $\mathbf{u}^{\text{obs}}$, such that the cost function, $J(\mathbf{m})$, is minimized.

We define the variation of $\mathbf{u}$ with respect to $\mathbf{m}$ in the direction $\delta\mathbf{m}$ (known as the Gâteaux differential, which is the directional derivative, but defined on more general spaces of functions) as

$$\delta\mathbf{u} \doteq \nabla_m \mathbf{u}\, \delta\mathbf{m},$$

where $\nabla_m(\cdot)$ is the gradient operator with respect to the model parameters (known, in the general case, as the Fréchet derivative). Then the corresponding directional derivative of J can be written as

$$\begin{aligned} \delta J &= \nabla_m J\, \delta\mathbf{m} \\ &= \nabla_u J\, \delta\mathbf{u} \\ &= \langle \nabla_u J_1\, \delta\mathbf{u} \rangle, \end{aligned} \tag{2.6}$$

where in the second line we have used the chain rule together with the definition of $\delta\mathbf{u}$, and in the third line $\langle\cdot\rangle$ denotes the space-time integral. Here we have passed the "derivative" under the integral sign, and J_1 is the integrand. There remains a major difficulty: the variation $\delta\mathbf{u}$ is impossible or unfeasible to compute numerically (for all directions $\delta\mathbf{m}$). To overcome this, we would like to eliminate $\delta\mathbf{u}$ from (2.6) by introducing an adjoint state (which can also be seen as a Lagrange multiplier).

To achieve this, we differentiate the state equation with respect to the model $\mathbf{m}$ and apply the necessary condition for optimality (disappearance of the variation) to obtain

$$\delta\mathbf{L} = \nabla_m \mathbf{L}\, \delta\mathbf{m} + \nabla_u \mathbf{L}\, \delta\mathbf{u} = 0.$$

Now we multiply this equation by an arbitrary test function $\mathbf{u}^\dagger$ (Lagrange multiplier) and integrate over space-time to obtain

$$\left\langle \mathbf{u}^\dagger \cdot \nabla_m \mathbf{L}\, \delta\mathbf{m} \right\rangle + \left\langle \mathbf{u}^\dagger \cdot \nabla_u \mathbf{L}\, \delta\mathbf{u} \right\rangle = 0.$$

Add this null expression to (2.6) and integrate by parts, regrouping terms in $\delta\mathbf{u}$:

$$\begin{aligned} \nabla_m J\, \delta\mathbf{m} &= \langle \nabla_u J_1\, \delta\mathbf{u} \rangle + \left\langle \mathbf{u}^\dagger \cdot \nabla_m \mathbf{L}\, \delta\mathbf{m} \right\rangle + \left\langle \mathbf{u}^\dagger \cdot \nabla_u \mathbf{L}\, \delta\mathbf{u} \right\rangle \\ &= \left\langle \delta\mathbf{u} \cdot \left(\nabla_u J_1^\dagger + \nabla_u \mathbf{L}^\dagger \mathbf{u}^\dagger \right) \right\rangle + \left\langle \mathbf{u}^\dagger \cdot \nabla_m \mathbf{L}\, \delta\mathbf{m} \right\rangle, \end{aligned}$$

where we have defined the adjoint operators $\nabla_u J_1^\dagger$ and $\nabla_u \mathbf{L}^\dagger$ via the appropriate inner products as

$$\langle \nabla_u J_1\, \delta\mathbf{u} \rangle = \left\langle \delta\mathbf{u} \cdot \nabla_u J_1^\dagger \right\rangle$$

and

$$\left\langle \mathbf{u}^\dagger \cdot \nabla_u \mathbf{L}\, \delta\mathbf{u} \right\rangle = \left\langle \delta\mathbf{u} \cdot \nabla_u \mathbf{L}^\dagger \mathbf{u}^\dagger \right\rangle.$$

Finally, to eliminate $\delta\mathbf{u}$, the adjoint state, $\mathbf{u}^\dagger$, should satisfy

$$\nabla_u \mathbf{L}^\dagger \mathbf{u}^\dagger = -\nabla_u J_1^\dagger,$$

which is known as the adjoint equation.

Once the adjoint solution, $\mathbf{u}^\dagger$, is found, the derivative/variation of the objective functional becomes

$$\nabla_m J\, \delta\mathbf{m} = \left\langle \mathbf{u}^\dagger \cdot \nabla_m \mathbf{L}\, \delta\mathbf{m} \right\rangle. \tag{2.7}$$

This key result enables us to compute the desired gradient, $\nabla_m J$, without the explicit knowledge of $\delta\mathbf{u}$. A number of *important remarks* are necessary here:

1. We obtain *explicit* formulas for the gradient with respect to each/any model parameter. Note that this has been done in a completely general setting, without any restrictions on the operator, **L**, or on the model parameters, **m**.

2. The *computational cost* is one solution of the adjoint equation, which is usually of the same order as (if not identical to) the direct equation,[18] but with a reversal of time.

3. The *variation* (Gâteaux derivative) of **L** with respect to the model parameters, **m**, is, in general, straightforward to compute.

4. We have not considered boundary (or initial) conditions in the above general approach. In real cases, these are potential sources of difficulties for the use of the adjoint approach—see Section 2.3.9, where the *discrete adjoint* can provide a way to overcome this hurdle.

5. For complete mathematical rigor, the above development should be performed in an appropriate *Hilbert space* setting that guarantees the existence of all the inner products and adjoint operators—the interested reader could consult the excellent short course notes of Estep [2004] and references therein, or the monograph of Tröltzsch [2010].

6. In many real problems, the optimization of the misfit functional leads to *multiple local minima* and often to very "flat" cost functions—these are hard problems for gradient-based optimization methods. These difficulties can be (partially) overcome by a panoply of tools:

   (a) *Regularization* terms can alleviate the nonuniqueness problem—see Engl et al. [1996] and Vogel [2002].

   (b) *Rescaling* the parameters and/or variables in the equations can help with the "flatness"—this technique is often employed in numerical optimization—see Nocedal and Wright [2006].

   (c) *Hybrid* algorithms, which combine stochastic and deterministic optimization (e.g., simulated annealing), can be used to avoid local minima—see Press et al. [2007].

7. When measurement and modeling errors can be modeled by Gaussian distributions and a background (prior) solution exists, the objective function may be generalized by including suitable *covariance matrices*. This is the approach employed systematically in DA—see below for full details.

We will now present a series of examples where we apply the adjoint approach to increasingly complex cases. We will use two alternative methods for the derivation of the adjoint equation: a Lagrange multiplier approach and the *tangent linear model* (TLM) approach. After seeing the two in action, the reader can adopt the one that suits her/him best. Note that the Lagrangian approach supposes that we perturb the sought-for parameters (as seen above in Section 2.2) and is thus not applicable to inverting for constant-valued parameters, in which case we must resort to the TLM approach.

[18] Note that for nonlinear equations this may not be the case, and one may require four or five times the computational effort.

2.3.2 ▪ Parameter identification example

A basic example: Let us consider in more detail the parameter identification problem (already encountered in Chapter 1) based on the convection-diffusion equation (1.6),

$$\begin{cases} -bu''(x)+cu'(x)=f(x), \quad 0<x<1, \\ u(0)=0,\ u(1)=0, \end{cases} \tag{2.8}$$

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

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

Let us, once again, calculate its gradient by introducing the TLM. Perturbing the cost function by a small perturbation in the direction α with respect to the two parameters gives

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

where $\tilde{u}=u_{b+\alpha\delta b,c+\alpha\delta c}$ is the perturbed solution and $u=u_{b,c}$ is the unperturbed one. Expanding and rearranging, we obtain

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

Now we divide by α on both sides of the equation and pass to the limit $\alpha\to 0$ to obtain the directional derivative (which is the derivative with respect to the parameters, in the direction of the perturbations),

$$\hat{J}[b,c](\delta b,\delta c)=\int_0^1 \left(u-u^{\text{obs}}\right)\hat{u}\,\mathrm{d}x, \tag{2.9}$$

where we have defined

$$\hat{u}=\lim_{\alpha\to 0}\frac{\tilde{u}-u}{\alpha}, \quad \hat{J}[b,c](\delta b,\delta c)=\lim_{\alpha\to 0}\frac{J(b+\alpha\delta b,c+\alpha\delta c)-J(b,c)}{\alpha},$$

and we have moved the limit under the integral sign. 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 the given model (2.8),

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

Then, subtracting these two equations and passing to the limit (using the definition of $\hat{u}$), we obtain

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

We can now define the TLM

$$\begin{cases} -b\hat{u}'' + c\hat{u}' = (\delta b)u'' - (\delta c)u', \\ \hat{u}(0) = 0,\ \hat{u}(1) = 0. \end{cases} \tag{2.10}$$

We want to be able to reformulate the directional derivative (2.9) to obtain a calculable expression for the gradient. So we multiply the TLM (2.10) by a variable p and integrate twice by parts, transferring derivatives from $\hat{u}$ onto p:

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

which gives (term by term)

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

and

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

Putting these results together, we have

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

or, grouping terms,

$$\int_0^1 \big(-b\,p'' - c\,p'\big)\,\hat{u} = b\,\hat{u}'(1)p(1) - b\,\hat{u}'(0)p(0) + \int_0^1 \big((\delta b)u'' - (\delta c)u'\big)\,p. \tag{2.11}$$

Now, to get rid of *all* the terms in $\hat{u}$ in this expression, we impose that p must satisfy *the adjoint model*

$$\begin{cases} -b\,p'' - c\,p' = (u - u^{\mathrm{obs}}), \\ p(0) = 0,\ p(1) = 0. \end{cases} \tag{2.12}$$

Integrating (2.12) and using the expression (2.11), we obtain

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

We recognize, in the last two terms, the L^2 inner product, which enables us, based on the key result (2.7), to finally write an explicit expression for the gradient, based on (2.9),

$$\nabla J(b,c) = \left(\int_0^1 p\,u''\mathrm{d}x, -\int_0^1 p\,u'\mathrm{d}x\right)^{\mathrm{T}}$$

or, separating the two components,

$$\nabla_b J(b,c) = \int_0^1 p u'' \mathrm{d}x, \tag{2.13}$$

$$\nabla_c J(b,c) = -\int_0^1 p u' \mathrm{d}x. \tag{2.14}$$

Thus, in this example, to compute the gradient of the least-squares error cost function, we must

- solve the direct equation (2.8) for u and derive u' and u'' from the solution, using some form of numerical differentiation (if we solved with finite differences), or differentiating the shape functions (if we solved with finite elements);
- solve the adjoint equation (2.12) for p (using the same solver[19] that we used for u);
- compute the two terms of the gradient, (2.13) and (2.14), using a suitable numerical integration scheme [Quarteroni et al., 2007].

Thus, for the additional cost of one solution of the adjoint model (2.12) plus a numerical integration, we can compute the gradient of the cost function with respect to either one, or both, of the unknown parameters. It is now a relatively easy task to find (numerically) the optimal values of b and c that minimize J by a suitable descent algorithm, for example, a quasi-Newton method [Nocedal and Wright, 2006; Quarteroni et al., 2007].

2.3.3 ▪ A simple ODE example: Lagrangian method

We now consider a variant of the convection-diffusion example, where the diffusion coefficient is spatially varying. This model is closer to many physical situations, where the medium is not homogeneous and we have zones with differing diffusive properties. The system is

$$\begin{cases} -\big(a(x)u'(x)\big)' - u'(x) = q(x), \quad 0 < x < 1, \\ u(0) = 0,\ u(1) = 0, \end{cases} \tag{2.15}$$

with the cost function

$$J[a] = \frac{1}{2}\int_0^1 \Big(u(x) - u^{\mathrm{obs}}(x)\Big)^2 \, dx,$$

where $u^{\mathrm{obs}}(x)$ denotes the observations on $[0,1]$. We now introduce an alternative approach for deriving the gradient, based on the Lagrangian (or variational formulation). Let the cost function be

$$J^*[a,p] = \frac{1}{2}\int_0^1 \Big(u(x) - u^{\mathrm{obs}}(x)\Big)^2 \, dx + \int_0^1 p\Big(-\big(au'\big)' - u' - q\Big)\, dx,$$

[19] This is not true when we use a *discrete* adjoint approach—see Section 2.3.9.

noting that the second integral is zero when u is a solution of (2.15) and that the adjoint variable, p, can be considered here to be a Lagrange multiplier function. We begin by taking the variation of J^* with respect to its variables, a and p:

$$\delta J^* = \int_0^1 \left(u - u^{\text{obs}}\right)\delta u\, dx + \int_0^1 \delta p \overbrace{\left(-\left(a u'\right)' - u' - q\right)}^{=0} dx + \int_0^1 p\left[\left(-\delta a\, u' - a\,\delta u'\right)'\right].$$

Now the strategy is to "kill terms" by imposing suitable, well-chosen conditions on p. This is achieved by integrating by parts and then defining the adjoint equation and boundary conditions on p as follows:

$$\begin{aligned}\delta J^* &= \int_0^1 \left[(u - u^{\text{obs}}) + p' - (a p')'\right]\delta u\, dx + \int_0^1 \delta a\, u' p' dx \\ &\quad \left[-p(\delta u + u'\delta a + a\delta u') + p' a \delta u\right]_0^1 \\ &= \int_0^1 \delta a\, u' p' dx,\end{aligned}$$

where we have used the zero boundary conditions on δu and assumed that the following adjoint system must be satisfied by p:

$$\begin{cases} -\left(a p'\right)' + p' = -(u - u^{\text{obs}}), \quad 0 < x < 1, \\ p(0) = 0,\ p(1) = 0. \end{cases} \tag{2.16}$$

And, as before, based on the key result (2.7), we are left with an explicit expression for the gradient,

$$\nabla_{a(x)} J^* = u' p'.$$

Thus, with one solution of the direct system (2.15) plus one solution of the adjoint system (2.16), we recover the gradient of the cost function with respect to the sought-for diffusion coefficient, $a(x)$.

2.3.4 ▪ Initial condition control

For DA problems in meteorology and oceanography, the objective is to reconstruct the *initial conditions* of the model. This is also the case in certain source identification problems for environmental pollution. We redo the above gradient calculations in this context. Let us consider the following system of (possibly nonlinear) ODEs:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{M}(\mathbf{X}) \quad \text{in } \Omega \times [0, T], \\ \mathbf{X}(t = 0) = \mathbf{U}, \end{cases} \tag{2.17}$$

with the cost function

$$J(\mathbf{U}) = \frac{1}{2}\int_0^T \|\mathbf{H}\mathbf{X} - \mathbf{Y}^{\text{o}}\|^2\, \mathrm{d}t,$$

where we have used the classical vector-matrix notation for systems of ODEs and $\|\cdot\|$ denotes the L^2-norm over the space variable. To compute the directional derivative,

we perturb the initial condition $\mathbf{U}$ by a quantity α in the direction $\mathbf{u}$ and denote by $\tilde{\mathbf{X}}$ the corresponding trajectory, satisfying

$$\begin{cases} \dfrac{\mathrm{d}\tilde{\mathbf{X}}}{\mathrm{d}t} = \mathbf{M}(\tilde{\mathbf{X}}) \quad \text{in } \Omega \times [0,T], \\ \tilde{\mathbf{X}}(t=0) = \mathbf{U} + \alpha\mathbf{u}. \end{cases} \tag{2.18}$$

We then have

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

Now, we set

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

and we compute the directional derivative,

$$\begin{aligned} \hat{J}[\mathbf{U}](u) &= \lim_{\alpha \to 0} \frac{J(\mathbf{U}+\alpha\mathbf{u}) - J(\mathbf{U})}{\alpha} \\ &= \frac{1}{2}\int_0^T \left(\mathbf{H}\mathbf{X} - \mathbf{Y}, \mathbf{H}\hat{\mathbf{X}}\right) + (\mathbf{H}\hat{\mathbf{X}}, \mathbf{H}\mathbf{X} - \mathbf{Y}) \\ &= \int_0^T (\mathbf{H}\hat{\mathbf{X}}, \mathbf{H}\mathbf{X} - \mathbf{Y}) \\ &= \int_0^T (\hat{\mathbf{X}}, \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{X} - \mathbf{Y})). \end{aligned} \tag{2.19}$$

By subtracting the equations (2.18) and (2.17) satisfied by $\tilde{\mathbf{X}}$ and $\mathbf{X}$, we obtain

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

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

$$\begin{cases} \dfrac{\mathrm{d}\hat{\mathbf{X}}}{\mathrm{d}t} = \left[\dfrac{\partial \mathbf{M}}{\partial \mathbf{X}}\right]\hat{\mathbf{X}}, \\ \hat{\mathbf{X}}(t=0) = \mathbf{u}. \end{cases} \tag{2.20}$$

These equations are the TLM.

We will now proceed to compute the adjoint model. As in the ODE example of Sections 1.5.3.1 and 2.3.2, we multiply the TLM (2.20) by $\mathbf{P}$ and integrate by parts on $[0, T]$. We find

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

and

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

Thus, substituting in equation (2.20), we get

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

Identifying with the directional derivative (2.19), we obtain the equations of the *adjoint model*

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} + \left[\dfrac{\partial \mathbf{M}}{\partial \mathbf{X}}\right]^{\mathrm{T}} \mathbf{P} = \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{X} - \mathbf{Y}), \\ \mathbf{P}(t = T) = 0, \end{cases} \tag{2.21}$$

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

We can now find the expression for the gradient. Using the adjoint model (2.21) in (2.19), we find

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

But, by definition,

$$\hat{J}[\mathbf{U}](\mathbf{u}) = (\nabla J_{\mathrm{U}}, \mathbf{u}),$$

and thus

$$\nabla J_{\mathrm{U}} = -\mathbf{P}(0).$$

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

2.3.5 ▪ Putting it all together: The case of a linear PDE

The natural extension of the ODEs seen above is the initial boundary value problem known as the diffusion equation:

$$\frac{\partial u}{\partial t} - \nabla \cdot (\nu \nabla u) = 0,\ x \in (0,L),\ t > 0,$$
$$u(x,0) = u_0(x),\ \ u(0,t) = 0,\ \ u(L,t) = \eta(t).$$

This equation has multiple origins emanating from different physical situations. The most common application is particle diffusion, where u is a concentration and ν is a diffusion coefficient. Then there is heat diffusion, for which u is temperature and ν is thermal conductivity. The equation is also found in finance, being closely related to the Black–Scholes model. Another important application is population dynamics. These diverse application fields, and hence the diffusion equation, give rise to a number of inverse and DA problems.

A variety of different controls can be applied to this system:

- *internal* control: $\nu(x)$—this is the parameter identification problem, also known as tomography;
- *initial* control: $\xi(x) = u_0(x)$—this is a source detection inverse or DA problem;
- *boundary* control: $\eta(t) = u(L,t)$—this is the "classical" boundary control problem, also a parameter identification inverse problem.

As above, we can define the cost function,

$$J[\nu,\xi,\eta] = \frac{1}{LT}\int_0^T \int_0^L (u - u^{\mathrm{o}})^2 \,\mathrm{d}x\,\mathrm{d}t,$$

which is now a space-time multiple integral, and its related Lagrangian,

$$J^* = \frac{1}{LT}\int_0^T \int_0^L (u - u^{\mathrm{o}})^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{1}{LT}\int_0^T \int_0^L p\,[u_t - (\nu u_x)_x]\,\mathrm{d}x\,\mathrm{d}t.$$

Now take the variation of J^*,

$$\delta J^* = \frac{1}{LT}\int_0^T \int_0^L 2(u - u^{\mathrm{o}})\delta u \,\mathrm{d}x\,\mathrm{d}t + \frac{1}{LT}\int_0^T \int_0^L \delta p \overbrace{[u_t - (\nu u_x)_x]}^{=0}\,\mathrm{d}x\,\mathrm{d}t$$
$$+ \frac{1}{LT}\int_0^T \int_0^L p\,[\delta u_t - (\delta\nu\, u_x + \nu \delta u_x)_x]\,\mathrm{d}x\,\mathrm{d}t,$$

and perform integration by parts to obtain

$$\delta J^* = \frac{1}{LT}\int_0^T \int_0^L \delta\nu\, u_x p_x \,\mathrm{d}x\,\mathrm{d}t - \frac{1}{LT}\int_0^L p\ \delta u|_{t=0}\,\mathrm{d}x + \frac{1}{LT}\int_0^T p\ \delta\eta|_{x=L}\,\mathrm{d}t, \quad (2.22)$$

where we have defined the adjoint equation as

$$\frac{\partial p}{\partial t} + \nabla \cdot (\nu \nabla u) = 2(u - u^{\mathrm{o}}), \quad x \in (0,L), \quad t > 0,$$
$$p(0,t) = 0, \quad p(L,t) = 0,$$
$$p(x,T) = 0.$$

As before, this equation is of the same type as the original diffusion equation but must be solved backward in time. Finally, from (2.22), we can pick off each of the three desired terms of the gradient:

$$\nabla_{\nu(x)} J^* = \frac{1}{T}\int_0^T u_x p_x \, dt,$$
$$\nabla_{u|_{t=0}} J^* = -p|_{t=0},$$
$$\nabla_{\eta|_{x=L}} J^* = p|_{x=L}.$$

Once again, at the expense of a single (backward) solution of the adjoint equation, we obtain explicit expressions for the gradient of the cost function with respect to each of the three control variables. This is quite remarkable and completely avoids "brute force" or exhaustive minimization, though, as mentioned earlier, we only have the guarantee of finding a local minimum. However, if we have a good starting guess, which is usually obtained from historical or other "physical" knowledge of the system, we are sure to arrive at a good (or, at least, better) minimum.

2.3.6 ▪ An adjoint "zoo"

As we have seen above, every (partial) differential operator has its very own adjoint form. We can thus derive, and categorize, adjoint equations for a whole variety of partial differential operators. Some common examples can be found in Table 2.1.

Table 2.1. *Adjoint forms for some common ordinary and partial differential operators.*

Operator	Adjoint
$\frac{du}{dx} - \gamma\frac{d^2u}{dx^2}$	$-\frac{dp}{dx} - \gamma\frac{d^2p}{dx^2}$
$\nabla\cdot(k\nabla u)$	$\nabla\cdot(k\nabla p)$
$\frac{\partial u}{\partial t} - c\frac{\partial^2 u}{\partial x^2}$	$-\frac{\partial p}{\partial t} - c\frac{\partial^2 p}{\partial x^2}$
$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x}$	$-\frac{\partial p}{\partial t} - c\frac{\partial p}{\partial x}$

The principle is simple: all second-order (or even) derivatives remain unchanged, whereas all first-order (or uneven) derivatives undergo a change of sign.

2.3.7 ▪ Application: Burgers' equation (a nonlinear PDE)

We will now consider a more realistic application based on Burgers' equation [Lax, 1973] with control of the initial condition and the boundary conditions. Burgers' equation is a very good approximation to the Navier–Stokes equation in certain contexts where viscous effects dominate convective effects. The Navier–Stokes equation itself is the model equation used for all aerodynamic simulations and for many flow problems. In addition, it is the cornerstone of *numerical weather prediction* (NWP) codes.

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 will be taken as a combination of the initial state and the two boundary conditions,

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

and the cost function is given by the usual mismatch,

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

We know that the derivative of J in the direction[20] (h_u, h_1, h_2) is given (as above in (2.9)) by

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

where $\hat{u}$ is defined, as usual, by

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

which is the solution of the TLM

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

We can now compute the equation of the adjoint model. As before, we multiply the TLM by p and integrate by parts on $[0, T]$. For clarity, we do this term by term:

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

[20] Instead of the δ notation, we have used another common form—the letter h—to denote the perturbation direction.

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

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

The natural initial[21] and boundary conditions for p are thus

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

which give

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

In other words,

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

We thus define the adjoint model as

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

[21] This is in fact a terminal condition, as we have encountered above.

Now we can rewrite the gradient of J in the form

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

which immediately yields

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

These explicit gradients enable us to solve inverse problems for either (1) the initial condition, which is a data assimilation problem, or (2) the boundary conditions, which is an optimal boundary control problem, or (3) both. Another extension would be a parameter identification problem for ν. This would make an excellent project or advanced exercise.

2.3.8 ▪ Adjoint of finite-dimensional (matrix) operators

Suppose now that we have a solution vector, $\mathbf{x}$, of a *discretized* PDE, or of any other set of n equations. Assume that $\mathbf{x}$ depends as usual on a parameter vector, $\mathbf{m}$, made up of p components—these are sometimes called control variables, design parameters, or decision parameters. If we want to optimize these values for a given cost function, $J(\mathbf{x},\mathbf{m})$, we need to compute, as for the continuous case, the gradient, $\mathrm{d}J/\mathrm{d}\mathbf{m}$. As we have seen above, this should be possible with an adjoint method at a cost that is independent of p and comparable to the cost of a single solution for $\mathbf{x}$. In the finite-dimensional case, this implies the inversion of a linear system, usually $\mathcal{O}(n^3)$ operations. This efficiency, especially for large values of p, is what makes the solution of the inverse problem tractable—if it were not for this, many problems would be simply impossible to solve within reasonable resource limits.

We will first consider systems of linear algebraic equations, and then we can readily generalize to nonlinear systems of algebraic equations and to initial-value problems for linear systems of ODEs.

2.3.8.1 ▪ Linear systems

Let $\mathbf{x}$ be the solution of the $(n \times n)$ linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.23}$$

and suppose that $\mathbf{x}$ depends on the parameters $\mathbf{m}$ through $\mathbf{A}(\mathbf{m})$ and $\mathbf{b}(\mathbf{m})$. Define a cost function, $J = J(\mathbf{x},\mathbf{m})$, that depends on $\mathbf{m}$ through $\mathbf{x}$. To evaluate the gradient of J with respect to $\mathbf{m}$ directly, we need to compute by the chain rule

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{m}} = \frac{\partial J}{\partial \mathbf{m}} + \frac{\partial J}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{m}} = J_{\mathbf{m}} + J_{\mathbf{x}}\mathbf{x}_{\mathbf{m}}, \tag{2.24}$$

where $J_{\mathbf{m}}$ is a $(p \times 1)$ column vector, $J_{\mathbf{x}}$ is a $(1 \times n)$ row vector, and $\mathbf{x}_{\mathbf{m}}$ is an $(n \times p)$ matrix. For a given function J the derivatives with respect to $\mathbf{x}$ and $\mathbf{m}$ are assumed to be easily computable. However, it is clearly much more difficult to differentiate $\mathbf{x}$ with respect to $\mathbf{m}$. Let us try and do this directly. We can differentiate, term by term, equation (2.23) with respect to the parameter m_i and solve for $\mathbf{x}_{m_i}$ from (applying the chain rule)

$$\mathbf{x}_{m_i} = \mathbf{A}^{-1}(\mathbf{b}_{m_i} - \mathbf{A}_{m_i}\mathbf{x}).$$

This must be done p times, rapidly becoming unfeasible for large n and p. Recall that p can be of the order of 10^6 in practical DA problems.

The adjoint method, which reduces this to a *single* solve, relies on the trick of adding zero in an astute way. We can do this, as was done above in the continuous case, by introducing a Lagrange multiplier. Since the residual vector $\mathbf{r}(\mathbf{x},\mathbf{m}) = \mathbf{A}\mathbf{x} - \mathbf{b}$ vanishes for the true solution $\mathbf{x}$, we can replace the function J by the augmented function

$$\hat{J} = J - \lambda^{\mathrm{T}}\mathbf{r}, \tag{2.25}$$

where we are free to choose λ at our convenience and we will use this liberty to make the difficult-to-compute term in (2.24), $\mathbf{x}_{\mathbf{m}}$, disappear. So let us take the expression for the gradient (2.24) and evaluate it at $\mathbf{r} = 0$,

$$\begin{aligned}\left.\frac{\mathrm{d}J}{\mathrm{d}\mathbf{m}}\right|_{\mathbf{r}=0} &= \left.\frac{\mathrm{d}\hat{J}}{\mathrm{d}\mathbf{m}}\right|_{\mathbf{r}=0} \\ &= J_{\mathbf{m}} - \lambda^{\mathrm{T}}\mathbf{r}_{\mathbf{m}} + \left(J_{\mathbf{x}} - \lambda^{\mathrm{T}}\mathbf{r}_{\mathbf{x}}\right)\mathbf{x}_{\mathbf{m}}.\end{aligned} \tag{2.26}$$

Then, to "kill" the troublesome $\mathbf{x}_{\mathbf{m}}$ term, we must require that $\left(J_{\mathbf{x}} - \lambda^{\mathrm{T}}\mathbf{r}_{\mathbf{x}}\right)$ vanish, which implies

$$\mathbf{r}_x^{\mathrm{T}}\lambda = J_{\mathbf{x}}^{\mathrm{T}}.$$

But $\mathbf{r}_{\mathbf{x}} = \mathbf{A}$, and hence λ must satisfy the *adjoint equation*

$$\mathbf{A}^{\mathrm{T}}\lambda = J_{\mathbf{x}}^{\mathrm{T}}, \tag{2.27}$$

which is a single $(n \times n)$ linear system. Equation (2.27) is of identical complexity as the original system (2.23), since the adjoint matrix $\mathbf{A}^{\mathrm{T}}$ has the same condition number, sparsity, and preconditioner as $\mathbf{A}$; i.e., if we have a numerical scheme (and hence a computer code) for solving the direct system, we will use precisely the same one for the adjoint.

With λ now known, we can compute the gradient of J from (2.26) as follows:

$$\begin{aligned}\left.\frac{\mathrm{d}J}{\mathrm{d}\mathbf{m}}\right|_{\mathbf{r}=0} &= J_{\mathbf{m}} - \lambda^{\mathrm{T}}\mathbf{r}_{\mathbf{m}} + 0 \\ &= J_{\mathbf{m}} - \lambda^{\mathrm{T}}(\mathbf{A}_{\mathbf{m}}\mathbf{x} - \mathbf{b}_{\mathbf{m}}).\end{aligned}$$

Once again, we assume that when $\mathbf{A}(\mathbf{m})$ and $\mathbf{b}(\mathbf{m})$ are explicitly known, this permits an easy calculation of the derivatives with respect to $\mathbf{m}$. If this is not the case, we must resort to automatic differentiation to compute these derivatives. The automatic differentiation approach will be presented below, after we have discussed nonlinear and initial-value problems.

2.3.8.2 ▪ Nonlinear systems

In general, the state vector $\mathbf{x}$ will satisfy a nonlinear functional equation of the general form

$$f(\mathbf{x},\mathbf{m})=0.$$

In this case the workflow is similar to the linear system. We start by solving for $\mathbf{x}$ with an iterative Newton-type algorithm, for example. Now define the augmented J as in (2.25), take the gradient as in (2.26), require that $\mathbf{r}_{\mathbf{x}}^{\mathrm{T}}\lambda = J_{\mathbf{x}}^{\mathrm{T}}$, and finally compute the gradient

$$\left.\frac{\mathrm{d}J}{\mathrm{d}\mathbf{m}}\right|_{\mathbf{r}=0} = J_{\mathbf{m}} - \lambda^{\mathrm{T}}\mathbf{r}_{\mathbf{m}}. \tag{2.28}$$

There is, of course, a slight modification needed: the adjoint equation is not simply the adjoint as in (2.27) but rather a tangent linear equation obtained by analytical (or automatic) differentiation of J with respect to $\mathbf{x}$.

2.3.8.3 ▪ Initial-value problems

We have, of course, seen this case in quite some detail above. Here we will reformulate it in matrix-vector form. We consider an initial-value problem for a linear, time-independent, homogeneous system of ODEs,

$$\dot{\boldsymbol{x}} = \mathbf{B}\boldsymbol{x},$$

with $\boldsymbol{x}(0)=\boldsymbol{b}$. We know that the solution is given by

$$\boldsymbol{x}(t)=\mathrm{e}^{\mathbf{B}t}\boldsymbol{b},$$

but this can be rewritten as a linear system, $\mathbf{A}\boldsymbol{x}=\boldsymbol{b}$, where $\mathbf{A}=\mathrm{e}^{-\mathbf{B}t}$. Now we can simply use our results from above. Suppose we want to minimize $J(\mathbf{x},\mathbf{m})$ based on the solution, $\boldsymbol{x}$, at time, t. As before, we can compute the adjoint vector, λ, using (2.27),

$$\mathrm{e}^{-\mathbf{B}^{\mathrm{T}}t}\lambda = J_{\mathbf{x}}^{\mathrm{T}},$$

but this is equivalent to the adjoint ODE,

$$\dot{\lambda} = \mathbf{B}^{\mathrm{T}}\lambda,$$

with $\lambda(0)=J_{\mathbf{x}}^{\mathrm{T}}$. This is exactly what we would expect: solving for the adjoint state vector, λ, is a problem of the same complexity and type as that of finding the state vector, $\boldsymbol{x}$. Clearly we are not obliged to use matrix exponentials for the solution, but we can choose among Runge–Kutta formulas, forward Euler, Crank–Nicolson, etc. [Quarteroni et al., 2007]. What about the important issue of stability? The eigenvalues of $\mathbf{B}$ and $\mathbf{B}^{\mathrm{T}}$ are complex conjugates and thus the stability of one (spectral radius less than one) implies the stability of the other. Finally, using (2.28), we obtain the gradient of the cost function in the time-dependent case,

$$\begin{aligned}\frac{\mathrm{d}J}{\mathrm{d}\mathbf{m}} &= J_{\mathbf{m}} - \lambda^{\mathrm{T}}(\mathbf{A}_{\mathbf{m}}\mathbf{x}-\mathbf{b}_{\mathbf{m}})\\ &= J_{\mathbf{m}} + \int_0^t \lambda^{\mathrm{T}}(t-t')\mathbf{B}_{\mathbf{m}}\mathbf{x}(t')\,\mathrm{d}t' + \lambda^{\mathrm{T}}\mathbf{b}_{\mathbf{m}},\end{aligned}$$

where we have differentiated the expression for $\mathbf{A}$. We observe that this computation of the gradient via the adjoint requires that we save in memory $\mathbf{x}(t')$ for all times $0 \leq t' \leq t$ to be able to compute the gradient. This is a well-known issue in adjoint approaches for time-dependent problems and can be dealt with in three ways (that are problem or, more precisely, dimension dependent):

1. Store everything in memory, if feasible.
2. If not, use some kind of checkpointing [Griewank and Walther, 2000], which means that we divide the time interval into a number of subintervals and store consecutively subinterval by subinterval.
3. Re-solve "simultaneously" forward and adjoint, and at the same time compute the integral; i.e., at each time step of the adjoint solution process, recompute the direct solution up to this time.

2.3.9 ▪ Continuous and discrete adjoints

In the previous section, we saw how to deal with finite-dimensional systems. This leads us naturally to the study of discrete adjoints, which can be computed by automatic differentiation, as opposed to analytical methods, where we took variations and used integration by parts. In the following discussion, the aim is not to show exactly how to write an adjoint code generator but to provide an understanding of the principles. Armed with this knowledge, the reader will be able to critically analyze (if needed) the eventual reasons for failure of the approach when applied to a real problem. An excellent reference is Hascoet [2012]—see also Griewank [2000].

To fix ideas, let us consider a second-order PDE of the general form (without loss of generality)

$$F(t, u, u_t, u_x, u_{xx}, \theta) = 0$$

and an objective function, $J(u, \theta)$, that depends on the unknown u and the parameters θ. As usual, we are interested in calculating the gradients of the cost function with respect to the parameters to find an optimal set of parameter values—usually one that attains the least-squares difference between simulated model predictions and real observations/measurements.

There are, in fact, two possible approaches for computing an adjoint state and the resulting gradients or sensitivities:

- discretization of the (analytical) adjoint, which we denote by **AtD** = Adjoint then Discretize (we have amply seen this above);
- adjoint of the discretization (the code), which we denote as **DtA** = Discretize then Adjoint.

The first is the *continuous* case, where we differentiate the PDE with respect to the parameters and then discretize the adjoint PDE to compute the approximate gradients. In the second, called the *discrete* approach, we first approximate the PDE by a discrete (linear or nonlinear) system and then differentiate the resulting discrete system with respect to the parameters. This is done by automatic differentiation of the code, which solves the PDE using tools such as TAPENADE, YAO, OpenAD, ADIFOR, ADMat, etc.—see `www.autodiff.org`. Note that numerical computation of gradients can be

achieved by two other means: divided/finite differences or symbolic differentiation.[22] The first is notoriously unstable, and the latter cannot deal with complex functionals. For these reasons, the adjoint method is largely preferable.

In "simpler" problems, AtD is preferable,[23] but this assumes that we are able to calculate analytically the adjoint equation by integration by parts and that we can find compatible boundary conditions for the adjoint variable—see, for example, Bocquet [2012a]. This was largely developed above. In more realistic, complex cases, we must often resort to DtA, but then we may be confronted with serious difficulties each time the code is modified, since this implies the need to regenerate the adjoint. DtA is, however, well-suited for a nonexpert who does not need to have a profound understanding of the simulation codes to compute gradients. The DtA approach works for any cost functional, and no explicit boundary conditions are needed. However, DtA may turn out to be inconsistent with the adjoint PDE if a nonlinear, high-resolution scheme (such as upwinding) is used—a comparison of the two approaches can be found in Li and Petzold [2004], where the important question of consistency is studied and a simple example of the 1D heat equation is also presented.

2.4 ▪ Variational DA

2.4.1 ▪ Introduction

2.4.1.1 ▪ History

Variational DA was formally introduced by the meteorological community for solving the problem of numerical weather prediction (NWP).

In 1922, Lewis Fry Richardson published the first attempt at forecasting the weather numerically. But large errors were observed that were caused by inaccuracies in the fields used as the initial conditions in his analysis [Lynch, 2008], thus indicating the need for a DA scheme.

Originally, subjective analysis was used to correct the simulation results. In this approach, NWP forecasts were adjusted manually by meteorologists using their operational expertise and experience. Then objective analysis (e.g., Cressman's successive correction algorithm), which fitted data to grids, was introduced for automated DA. These objective methods used simple interpolation approaches (e.g., a quadratic polynomial interpolation scheme based on least-squares regression) and thus were 3D DA methods.

Later, 4D DA methods, called nudging, were developed. These are based on the simple idea of Newtonian relaxation and introduce into the right-hand side of the model dynamical equations a term that is proportional to the difference of the calculated meteorological variable and the observed value. This term has a negative sign and thus keeps the calculated state vector closer to the observations. Nudging can be interpreted as a variant of the Kalman filter (KF) with the gain matrix prescribed, rather than obtained from covariances. Various nudging algorithms are described in Chapter 4.

A major development was achieved by L. Gandin [1963], who introduced the statistical interpolation (or optimal interpolation) method, which developed earlier ideas of Kolmogorov. This is a 3D DA method and is a type of regression analysis that utilizes information about the spatial distributions of covariance functions of the errors

[22] By packages such as Maple, Mathematica, SAGE, etc.

[23] Though there are differences of opinion among practitioners who prefer the discrete adjoint for these cases as well. Thus, the final choice depends on one's personal experience and competence.

of the first guess field (previous forecast) and true field. The optimal interpolation algorithm is the reduced version of the KF algorithm in which the covariance matrices are not calculated from the dynamical equations but are predetermined. This is treated in Chapter 3.

Attempts to introduce KF algorithms as a 4D DA tool for NWP models came later. However, this was (and remains) a difficult task due to the very high dimensions of the computational grid and the underlying matrices. To overcome this difficulty, approximate or suboptimal KFs were developed. These include the ensemble Kalman filter (EnKF) and the reduced-rank Kalman filters (such as RRSQRT)—see Chapters 3, 5, and 6.

Another significant advance in the development of the 4D DA methods was the use of optimal control theory, also known as the variational approach. In the seminal work of Le Dimet and Talagrand [1986] based on earlier work of G. Marchuk, they were the first to apply the theory of Lions [1988] (see also Tröltzsch [2010]) to environmental modeling. The significant advantage of the variational approach is that the meteorological fields satisfy the dynamical equations of the NWP model, and at the same time they minimize the functional characterizing the difference between simulations and observations. Thus, a problem of constrained minimization is solved, as has been amply shown above in this chapter.

As has been shown by Lorenc [2003], Talagrand [2012], and others, all the above-mentioned 4D DA methods are in some limit equivalent. Under certain assumptions they minimize the same cost function. However, in practical applications these assumptions are never fulfilled and the different methods perform differently. This raises the still disputed question: Which approach, Kalman filtering or variational assimilation, is better? Further fundamental questions arise in the application of advanced DA techniques. A major issue is that of the convergence of the numerical method to the global minimum of the functional to be minimized—please refer to the important discussions in the first two sections of this chapter.

The 4D DA method that is currently most successful is hybrid incremental 4D-Var (see below and Chapters 5 and 7), where an ensemble is used to augment the climatological background error covariances at the start of the DA time window, but the background error covariances are evolved during the time window by a simplified version of the NWP forecast model. This DA method is used operationally at major forecast centers, though there is currently a tendency to move toward the more efficient ensemble variational (EnVar) methods that will be described in Chapter 7.

2.4.1.2 ▪ Formulation

In variational DA we describe the state of the system by a state variable, $\mathbf{x}(t) \in \mathscr{X}$, a function of space and time that represents the physical variables of interest, such as current velocity (in oceanography), temperature, sea-surface height, salinity, biological species concentration, or chemical concentration. The evolution of the state is described by a system of (in general nonlinear) differential equations in a region Ω,

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}) \quad \text{in } \Omega \times [0, T], \\ \mathbf{x}(t=0) = \mathbf{x}_0, \end{cases} \tag{2.29}$$

where the initial condition is unknown (or inaccurately known). Suppose that we are in possession of observations $\mathbf{y}(t) \in \mathscr{O}$ and an observation operator $\mathscr{H}$ that describes

the available observations. Then, to characterize the difference between the observations and the state, we define the objective (or cost) function

$$J(\mathbf{x}_0) = \frac{1}{2}\int_0^T \|\mathbf{y}(t) - \mathscr{H}(\mathbf{x}(\mathbf{x}_0, t))\|^2_{\mathscr{O}}\, \mathrm{d}t + \frac{1}{2}\left\|\mathbf{x}_0 - \mathbf{x}^{\mathrm{b}}\right\|^2_{\mathscr{X}}, \tag{2.30}$$

where $\mathbf{x}^{\mathrm{b}}$ is the background (or first guess) and the second term plays the role of a regularization (in the sense of Tikhonov—see Vogel [2002] and Hansen [2010]). The two norms under the integral, in the finite-dimensional case, will be represented by the error covariance matrices $\mathbf{R}$ and $\mathbf{B}$, respectively—see Chapter 1 and Section 2.4.3 below. Note that, for mathematical rigor, we have indicated the relevant functional spaces on which the norms are defined.

In the continuous context, the DA problem is formulated as follows: find the analyzed state, $\mathbf{x}_0^{\mathrm{a}}$, that minimizes J and satisfies

$$\mathbf{x}_0^{\mathrm{a}} = \operatorname{argmin} J(\mathbf{x}_0).$$

As seen above, the necessary condition for the existence of a (local) minimum is

$$\nabla J(\mathbf{x}_0^{\mathrm{a}}) = 0.$$

2.4.2 ▪ Adjoint methods in DA

To solve the above minimization problem for variational DA, we will use the adjoint approach. In summary, the *adjoint method* for DA is an iterative scheme that involves searching for the minimum of a scalar cost function with respect to a multidimensional initial state. The search algorithm is called a descent method and requires the derivative of the cost function with respect to arbitrary perturbations of the initial state. This derivative, or gradient, is obtained by running an adjoint model backward in time. Once the derivative is obtained, a direction that leads to lower cost has been identified, but the step size has not. Therefore, further calculations are needed to determine how far along this direction one needs to go to find a lower cost. Once this initial state is found, the next iteration is started. The algorithm proceeds until the minimum of the cost function is found. It should be noted that the adjoint method is used in 4D-Var to find the initial conditions that minimize a cost function. However, one could equally well have chosen to find the boundary conditions, or model parameters, as was done in the numerous examples presented in Section 2.3.

We point out that a truly unified derivation of variational DA should start from a probabilistic/statistical model. Then, as was mentioned above (see Section 1.5), we can obtain the 3D-Var model as a special case. We will return to this in Chapter 3. Here, as opposed to the presentation in Chapter 1, we will give a unified treatment of 3D- and 4D-Var that leads naturally to variants of the approach.

2.4.3 ▪ 3D-Var and 4D-Var: A unified framework

The 3D-Var and 4D-Var approaches were introduced in Chapter 1. Here we will recall the essential points of the formulation, present them in a unified fashion (after Talagrand [2012]), and expand on some concrete aspects and variants.

Unlike sequential/statistical assimilation (which emanates from estimation theory), we saw that variational assimilation is based on optimal control theory, itself

derived from the *calculus of variations*. The analyzed state was defined as the one that *minimizes a cost function.* The minimization requires numerical optimization techniques. These techniques can rely on the *gradient* of the cost function, and this gradient will be obtained with the aid of *adjoint methods*, which we have amply discussed above. Note that variational DA is a particular usage of the adjoint approach.

Usually, 3D-Var and 4D-Var are introduced in a finite-dimensional or discrete context—this approach will be used in this section. For the infinite-dimensional or continuous case, we must use the calculus of variations and PDEs, as was done in the previous sections of this chapter.

We start out with the following cost function:

$$J(x) = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^{\mathrm{b}}\right)^{\mathrm{T}}\mathbf{B}^{-1}\left(\mathbf{x}-\mathbf{x}^{\mathrm{b}}\right)+\frac{1}{2}(\mathbf{H}\mathbf{x}-\mathbf{y})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{H}\mathbf{x}-\mathbf{y}), \tag{2.31}$$

where, as was defined in the notation of Section 1.5.1, $\mathbf{x}$, $\mathbf{x}^{\mathrm{b}}$, and $\mathbf{y}$ are the state, the background state, and the measured state, respectively; $\mathbf{H}$ is the observation matrix (a linearization of the observation operator $\mathscr{H}$); and $\mathbf{R}$ and $\mathbf{B}$ are the observation and background error covariance matrices, respectively. This quadratic function attempts to strike a balance between some a priori knowledge about a background (or historical) state and the actual measured, or observed, state. It also assumes that we know and can invert the matrices $\mathbf{R}$ and $\mathbf{B}$—this, as will be pointed out below, is far from obvious. Furthermore, it represents the sum of the (weighted) background deviations and the (weighted) observation deviations.

2.4.3.1 ▪ The stationary case: 3D-Var

We note that when the background, $\mathbf{x}^{\mathrm{b}} = \mathbf{x}^{\mathrm{b}} + \epsilon^{\mathrm{b}}$, is available at some time t_k, together with observations of the form $\mathbf{y} = \mathbf{H}\mathbf{x}^{\mathrm{t}} + \epsilon^{\mathrm{o}}$ that have been acquired at the same time (or over a short enough interval of time when the dynamics can be considered stationary), then the minimization of (2.31) will produce an estimate of the system state at time t_k. In this case, the analysis is called three-dimensional variational analysis and is abbreviated as *3D-Var*.

We have seen above, in Section 1.5.2, that the best linear unbiased estimator (BLUE) requires the computation of an optimal gain matrix. We will show (in Chapter 3) that the optimal gain takes the form

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

where $\mathbf{B}$ and $\mathbf{R}$ are the covariance matrices, to obtain an analyzed state,

$$\mathbf{x}^{\mathrm{a}} = \mathbf{x}^{\mathrm{b}} + \mathbf{K}(\mathbf{y}-\mathbf{H}(\mathbf{x}^{\mathrm{b}})).$$

But this is precisely the state that minimizes the 3D-Var cost function. This is quite easily verified by taking the gradient, term by term, of the cost function (2.31) and equating to zero,

$$\nabla J(\mathbf{x}^{\mathrm{a}}) = \mathbf{B}^{-1}\left(\mathbf{x}^{\mathrm{a}}-\mathbf{x}^{\mathrm{b}}\right)-\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathrm{a}}) = 0, \tag{2.32}$$

where

$$\mathbf{x}^{\mathrm{a}} = \operatorname{argmin} J(\mathbf{x}).$$

Solving the equation, we find

$$\begin{aligned}
\mathbf{B}^{-1}\left(\mathbf{x}^{\mathrm{a}}-\mathbf{x}^{\mathrm{b}}\right) &= \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathrm{a}}\right),\\
\left(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)\mathbf{x}^{\mathrm{a}} &= \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}+\mathbf{B}^{-1}\mathbf{x}^{\mathrm{b}},\\
\mathbf{x}^{\mathrm{a}} &= \left(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}+\mathbf{B}^{-1}\mathbf{x}^{\mathrm{b}}\right)\\
&= \left(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\left(\left(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)\mathbf{x}^{\mathrm{b}}\right.\\
&\quad \left.-\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{x}^{\mathrm{b}}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}\right)\\
&= \mathbf{x}^{\mathrm{b}}+\left(\mathbf{B}^{-1}+\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathrm{b}}\right)\\
&= \mathbf{x}^{\mathrm{b}}+\mathbf{K}\left(\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathrm{b}}\right),
\end{aligned} \tag{2.33}$$

where we have simply added and subtracted the term $\left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)\mathbf{x}^{\mathrm{b}}$ in the third-to-last line, and in the last line we have brought out what are known as the *innovation* term,

$$\mathbf{d}=\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathrm{b}},$$

and the *gain matrix*,

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

This matrix can be rewritten as

$$\mathbf{K}=\mathbf{B}\mathbf{H}^{\mathrm{T}}\left(\mathbf{R}+\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}\right)^{-1} \tag{2.34}$$

using the well-known Sherman–Morrison–Woodbury formula of linear algebra [Golub and van Loan, 2013], which completely avoids the direct computation of the inverse of the matrix $\mathbf{B}$. The linear combination in (2.33) of a background term plus a multiple of the innovation is a classical result of linear-quadratic control theory [Friedland, 1986; Gelb, 1974; Kwakernaak and Sivan, 1972] and shows how nicely DA fits in with and corresponds to (optimal) control theory. The form of the gain matrix (2.34) can be explained quite simply. The term $\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}$ is the background covariance transformed to the observation space. The denominator term, $\mathbf{R}+\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}$, expresses the sum of observation and background covariances. The numerator term, $\mathbf{B}\mathbf{H}^{\mathrm{T}}$, takes the ratio of $\mathbf{B}$ and $\mathbf{R}+\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}}$ back to the model space. This recalls (and is completely analogous to) the variance ratio,

$$\frac{\sigma_{\mathrm{b}}^2}{\sigma_b^2+\sigma_{\mathrm{o}}^2},$$

that appears in the optimal BLUE (see Chapter 1 and Chapter 3) solution. This is the case for a single observation, $\mathbf{y}$, of a quantity, $\mathbf{x}$,

$$\begin{aligned}
x^{\mathrm{a}} &= x^{\mathrm{b}}+\frac{\sigma_{\mathrm{b}}^2}{\sigma_{\mathrm{b}}^2+\sigma_{\mathrm{o}}^2}\left(x^{\mathrm{o}}-x^{\mathrm{b}}\right)\\
&= x^{\mathrm{b}}+\frac{1}{1+\alpha}\left(x^{\mathrm{o}}-x^{\mathrm{b}}\right),
\end{aligned}$$

where

$$\alpha=\frac{\sigma_{\mathrm{o}}^2}{\sigma_{\mathrm{b}}^2}.$$

In other words, the best way to estimate the state is to take a weighted average of the background (or prior) and the observations of the state. And the best weight is the ratio of the mean squared errors (variances). This statistical viewpoint is thus perfectly reproduced in the 3D-Var framework.

2.4.3.2 ▪ The nonstationary case: 4D-Var

A more realistic, but complicated, situation arises when one wants to assimilate observations that are acquired over a time interval during which the system dynamics (flow, for example) cannot be neglected. Suppose that the measurements are available at a succession of instants, t_k, $k = 0, 1, \ldots, K$, and are of the form

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\epsilon}_k^{\mathrm{o}}, \tag{2.35}$$

where $\mathbf{H}_k$ is a linear observation operator and $\boldsymbol{\epsilon}_k^{\mathrm{o}}$ is the observation error with covariance matrix $\mathbf{R}_k$, and suppose that these observation errors are uncorrelated in time. Now we add the dynamics described by the state equation,

$$\mathbf{x}_{k+1} = \mathbf{M}_{k+1} \mathbf{x}_k, \tag{2.36}$$

where we have neglected any model error.[24] We suppose also that at time index $k = 0$ we know the background state, $\mathbf{x}_0^{\mathrm{b}}$, and its error covariance matrix, $\mathbf{P}_0^{\mathrm{b}}$, and we suppose that the errors are uncorrelated with the observations in (2.35). Then a given initial condition, $\mathbf{x}_0$, defines a unique model solution, $\mathbf{x}_{k+1}$, according to (2.36). We can now generalize the objective function (2.31), which becomes

$$J(\mathbf{x}_0) = \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right)^T \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right) + \frac{1}{2}\sum_{k=0}^{K} (\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k)^T \mathbf{R}_k^{-1} (\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k). \tag{2.37}$$

The minimization of $J(\mathbf{x}_0)$ will provide the initial condition of the model that fits the data most closely. This analysis is called *strong constraint four-dimensional variational assimilation*, abbreviated as *strong constraint 4D-Var*. The term *strong constraint* implies that the model found by the state equation (2.36) must be exactly satisfied by the sequence of estimated state vectors.

In the presence of model uncertainty, the state equation becomes

$$\mathbf{x}_{k+1}^{\mathrm{t}} = \mathbf{M}_{k+1} \mathbf{x}_k^{\mathrm{t}} + \boldsymbol{\eta}_{k+1}, \tag{2.38}$$

where the model noise has covariance matrix $\mathbf{Q}_k$,which we suppose to be uncorrelated in time and uncorrelated with the background and observation errors. The objective function for the BLUE for the sequence of states

$$\{\mathbf{x}_k, k = 0, 1, \ldots, K\}$$

is of the form

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_K) = {} & \frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right)^{\mathrm{T}} \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right) \\ & + \frac{1}{2}\sum_{k=0}^{K} (\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k)^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k) \\ & + \frac{1}{2}\sum_{k=0}^{K-1} \left(\mathbf{x}_{k+1} - \mathbf{M}_{k+1}\mathbf{x}_k\right)^{\mathrm{T}} \mathbf{Q}_{k+1}^{-1} \left(\mathbf{x}_{k+1} - \mathbf{M}_{k+1}\mathbf{x}_k\right). \end{aligned} \tag{2.39}$$

[24] This will be taken into account in Section 2.4.7.5.

This objective function has become a function of the complete sequence of states

$$\{\mathbf{x}_k, k = 0, 1, \ldots, K\},$$

and its minimization is known as *weak constraint four-dimensional variational assimilation*, abbreviated as *weak constraint* 4*D-Var*. Equations (2.37) and (2.39), with an appropriate reformulation of the state and observation spaces, are special cases of the BLUE objective function—see Talagrand [2012].

All the above forms of variational assimilation, as defined by (2.31), (2.37), and (2.39), have been used for real-world DA, in particular in meteorology and oceanography. However, these methods are directly applicable to a vast array of other domains, among which we can cite geophysics and environmental sciences, seismology, atmospheric chemistry, and terrestrial magnetism. Examples of all these can be found in the applications chapters of Part III. We remark that in real-world practice, variational assimilation is performed on nonlinear models. If the extent of nonlinearity is sufficiently small (in some sense), then variational assimilation, even if it does not solve the correct estimation problem, will still produce useful results.

Some remarks concerning implementation: Now, our problem reduces to quantifying the covariance matrices and then, of course, computing the analyzed state. The quantification of the covariance matrices must result from extensive data studies (or the use of a KF approach—see Chapter 3). The computation of the analyzed state will be described in the next subsection—this will not be done directly, but rather by an adjoint approach for minimizing the cost functions. There is of course the inverse of $\mathbf{B}$ or $\mathbf{P}^{\mathrm{b}}$ to compute, but we remark that there appear only matrix-vector products of $\mathbf{B}^{-1}$ and $\left(\mathbf{P}^{\mathrm{b}}\right)^{-1}$, and we can thus define operators (or routines) that compute these efficiently without the need for large storage capacities.

2.4.3.3 ▪ The adjoint approach

We explain the adjoint approach in the case of strong constraint 4D-Var, taking into account a completely general nonlinear setting for the model and for the observation operators. Let $\mathbf{M}_k$ and $\mathbf{H}_k$ be the nonlinear model and observation operators, respectively. We reformulate (2.36) and (2.37) in terms of the nonlinear operators as

$$\begin{aligned} J(\mathbf{x}_0) = &\frac{1}{2}\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right)^{\mathrm{T}} \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right) \\ &+ \frac{1}{2}\sum_{k=0}^{K} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k)^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k), \end{aligned} \tag{2.40}$$

with the dynamics

$$\mathbf{x}_{k+1} = \mathbf{M}_{k+1}(\mathbf{x}_k), \quad k = 0, 1, \ldots, K-1. \tag{2.41}$$

The minimization problem requires that we now compute the gradient of J with respect to $\mathbf{x}_0$. The gradient is determined from the property that for a given perturbation $\delta\mathbf{x}_0$ of $\mathbf{x}_0$, the corresponding first-order variation of J is

$$\delta J = \left(\nabla_{\mathbf{x}_0} J\right)^{\mathrm{T}} \delta\mathbf{x}_0. \tag{2.42}$$

The perturbation is propagated by the tangent linear equation,

$$\delta\mathbf{x}_{k+1} = M_{k+1}\delta\mathbf{x}_k, \quad k = 0, 1, \ldots, K-1, \tag{2.43}$$

obtained by differentiation of the state equation (2.41), where M_{k+1} is the Jacobian matrix (of first-order partial derivatives) of $\mathbf{x}_{k+1}$ with respect to $\mathbf{x}_k$. The first-order variation of the cost function is obtained similarly by differentiation of (2.40),

$$\delta J = \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right)^{\mathrm{T}} \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \delta \mathbf{x}_0 + \sum_{k=0}^{K} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k)^{\mathrm{T}} \mathbf{R}_k^{-1} \mathrm{H}_k \delta \mathbf{x}_k, \tag{2.44}$$

where H_k is the Jacobian of $\mathbf{H}_k$ and $\delta \mathbf{x}_k$ is defined by (2.43). This variation is a compound function of $\delta \mathbf{x}_0$ that depends on all the $\delta \mathbf{x}_k$'s. But if we can obtain a direct dependence on $\delta \mathbf{x}_0$ in the form of (2.42), eliminating the explicit dependence on $\delta \mathbf{x}_k$, then we will (as in the previous sections of this chapter) arrive at an explicit expression for the gradient, $\nabla_{\mathbf{x}_0} J$, of our cost function, J. This will be done, as we have done before, by introducing an adjoint state and requiring that it satisfy certain conditions—namely, the adjoint equation. Let us now proceed with this program.

We begin by defining, for $k = 0, 1, \ldots, K$, the adjoint state vectors $\mathbf{p}_k$ that belong to the dual of the state space. Now we take the null products (according to the tangent state equation (2.43)),

$$\mathbf{p}_k^{\mathrm{T}} \left(\delta\ x_k - M_k \delta \mathbf{x}_{k-1}\right),$$

and subtract them from the right-hand side of the cost function variation (2.44),

$$\begin{aligned}\delta J = &\left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right)^{T} \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \delta \mathbf{x}_0 + \sum_{k=0}^{K} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k)^{T} \mathbf{R}_k^{-1} \mathrm{H}_k \delta \mathbf{x}_k \\ &- \sum_{k=0}^{K} \mathbf{p}_k^{T} \left(\delta \mathbf{x}_k - M_k \delta \mathbf{x}_{k-1}\right).\end{aligned}$$

Rearranging the matrix products, using the symmetry of $\mathbf{R}_k$, and regrouping terms in $\delta \mathbf{x}$, we obtain

$$\begin{aligned}\delta J = &\left[\left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right) + \mathrm{H}_0^{\mathrm{T}} \mathbf{R}_0^{-1} (\mathbf{H}_0(\mathbf{x}_0) - \mathbf{y}_0) + M_0^{\mathrm{T}} \mathbf{p}_1\right] \delta \mathbf{x}_0 \\ &+ \left[\sum_{k=1}^{K-1} \mathrm{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k) - \mathbf{p}_k + M_k^{\mathrm{T}} \mathbf{p}_{k+1}\right] \delta \mathbf{x}_k \\ &+ \left[\mathrm{H}_K^{\mathrm{T}} \mathbf{R}_K^{-1} (\mathbf{H}_K(\mathbf{x}_K) - \mathbf{y}_K) - \mathbf{p}_K\right] \delta \mathbf{x}_k.\end{aligned}$$

Notice that this expression is valid for any choice of the adjoint states, $\mathbf{p}_k$, and, in order to "kill" all $\delta \mathbf{x}_k$ terms, except $\delta \mathbf{x}_0$, we must simply impose that

$$\mathbf{p}_K = \mathrm{H}_K^{\mathrm{T}} \mathbf{R}_K^{-1} (\mathbf{H}_K(\mathbf{x}_K) - \mathbf{y}_K), \tag{2.45}$$

$$\mathbf{p}_k = \mathrm{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{H}_k(\mathbf{x}_k) - \mathbf{y}_k) + M_k^{\mathrm{T}} \mathbf{p}_{k+1}, \quad k = K-1, \ldots, 1, \tag{2.46}$$

$$\mathbf{p}_0 = \left(\mathbf{P}_0^{\mathrm{b}}\right)^{-1} \left(\mathbf{x}_0 - \mathbf{x}_0^{\mathrm{b}}\right) + \mathrm{H}_0^{\mathrm{T}} \mathbf{R}_0^{-1} (\mathbf{H}_0(\mathbf{x}_0) - \mathbf{y}_0) + M_0^{\mathrm{T}} \mathbf{p}_1. \tag{2.47}$$

We recognize the backward adjoint equation for $\mathbf{p}_k$, and the only term remaining in the variation of J is then

$$\delta J = \mathbf{p}_0^{T} \delta \mathbf{x}_0,$$

so that $\mathbf{p}_0$ is the sought-for gradient, $\nabla_{\mathbf{x}_0} J$, of the objective function with respect to the initial condition, $\mathbf{x}_0$, according to (2.42). The system of equations (2.45)–(2.47) is

Algorithm 2.1 Iterative 3D-Var algorithm.

$k = 0$, $x = x_0$
while $||\nabla J|| > \epsilon$ **or** $k \leq k_{\max}$
  compute J with (2.31)
  compute ∇J with (2.32)
  gradient descent and update of x_{k+1}
  $k = k + 1$
end

the adjoint of the tangent linear equation (2.43). The term *adjoint* here corresponds to the transposes of the matrices $\mathrm{H}_k^{\mathrm{T}}$ and M_k^{T} that, as we have seen before, are the finite-dimensional analogues of an adjoint operator. We can now propose the "usual" algorithm for solving the optimization problem by the adjoint approach:

1. For a given initial condition, $\mathbf{x}_0$, integrate forward the (nonlinear) state equation (2.41) and store the solutions, $\mathbf{x}_k$ (or use some sort of checkpointing).

2. From the final condition, (2.45), integrate backward in time the adjoint equations (2.46).

3. Compute directly the required gradient (2.47).

4. Use this gradient in an iterative optimization algorithm to find a (local) minimum.

The above description for the solution of the 4D-Var DA problem clearly covers the case of 3D-Var, where we seek to minimize (2.31). In this case, we need only the transpose Jacobian H^T of the observation operator.

2.4.4 ▪ The 3D-Var algorithm

The matrices involved in the calculation of equation (2.33) are often neither storable in memory nor manipulable because of their very large dimensions, which can be as much as 10^6 or more. Thus, the direct calculation of the gain matrix, $\mathbf{K}$, is unfeasible. The 3D-Var variational method overcomes these difficulties by attempting to iteratively minimize the cost function, J. This minimization can be achieved, for inverse problems in general, by a combination of an adjoint approach for the computation of the gradient with a descent algorithm in the direction of the gradient. For DA problems where there is no time dependence, the adjoint operation requires only a matrix adjoint (and not the solution of an adjoint equation[25]), and the approach is called 3D-Var, whereas for time-dependent problems we will use the 4D-Var approach, which is presented in the next subsection.

The iterative 3D-Var Algorithm 2.1 is a classical case of an optimization algorithm [Nocedal and Wright, 2006] that uses as a stopping criterion the fact that ∇J is small or that the maximum number of iterations, $k_{\max}$, is reached. For the gradient descent, there is a wide choice of algorithmic approaches, but quasi-Newton methods [Nocedal and Wright, 2006; Quarteroni et al., 2007] are generally used and recommended.

[25] This may not be valid for complicated observation operators.

2.4.4.1 ▪ On the roles of R and B

The relative magnitudes of the errors due to measurement and background provide us with important information as to how much "weight" to give to the different information sources when solving the assimilation problem. For example, if background errors are larger than observation errors, then the analyzed state solution to the DA problem should be closer to the observations than to the background and vice versa.

The background error covariance matrix, **B**, plays an important role in DA. This is illustrated by the following example.

Example 2.5. Effect of a single observation. Suppose that we have a single observation at a point corresponding to the jth element of the state vector. The observation operator is then

$$\mathbf{H} = (\ 0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0\ ).$$

The gradient of J is

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

Since it must be equal to zero at the minimum $\mathbf{x}^{\mathrm{a}}$,

$$\left(\mathbf{x}^{\mathrm{a}} - \mathbf{x}^{\mathrm{b}}\right) = \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y}^{\mathrm{o}} - \mathbf{H}\mathbf{x}^{\mathrm{a}}).$$

But $\mathbf{R} \doteq \sigma^2$; $\mathbf{H}\mathbf{x}^{\mathrm{a}} = x_j^{\mathrm{a}}$; and $\mathbf{B}\mathbf{H}^T$ is the jth column of **B**, whose elements are denoted by $B_{i,j}$ with $i = 1, \ldots, n$. So we see that

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

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 jth variable. ∎

In the 4D-Var case, the increment at time t will be proportional to a single column of $\mathbf{M}\mathbf{B}\mathbf{M}^T$, which describes the error covariances of the background at the time, t, of the observation.

2.4.5 ▪ The 4D-Var algorithm

In this section, we reformulate the 4D-Var approach in a form that is better adapted to algorithmic implementation. As we have just seen, the 4D-Var method generalizes 3D-Var to the case where the observations are obtained at different times—this is depicted in Figure 2.4. As was already stated in Chapter 1, the difference between three-dimensional (3D-Var) and four-dimensional (4D-Var) DA is the use of a numerical forecast model in the latter. In 4D-Var, the cost function is still expressed in terms of the initial state, $\mathbf{x}_0$, but it includes the model because the observation $\mathbf{y}_k^{\mathrm{o}}$ at time k is compared to $\mathbf{H}_k(\mathbf{x}_k)$, where $\mathbf{x}_k$ is the state at time k initialized by $\mathbf{x}_0$ and the adjoint is not simply the transpose of a matrix, but the "transpose" of the model/operator dynamics.

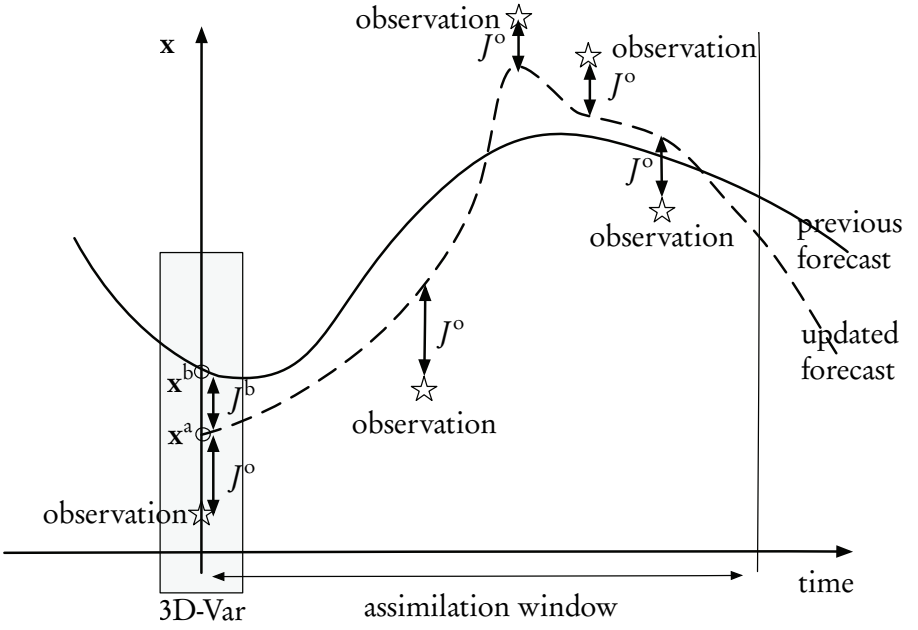


Figure 2.4. *3D- and 4D-Var.*

2.4.5.1 ▪ Cost function and gradient

The cost function (2.37) is still expressed in terms of the initial state, $\mathbf{x}$ (we have dropped the zero subscript, for simplicity), but it now includes the model because the observation $\mathbf{y}_k^o$ at time k is compared to $\mathbf{H}_k(\mathbf{x}_k)$, where $\mathbf{x}_k$ is the state at time k initialized by $\mathbf{x}$. The cost function is the sum of the background and the observation errors,

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

where the background term is the same as above:

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

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

$$J^o(\mathbf{x}) = \frac{1}{2}\sum_{k=0}^{K}\left(\mathbf{y}_k^o - \mathbf{H}_k(\mathbf{x}_k)\right)^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k^o - \mathbf{H}_k(\mathbf{x}_k)\right),$$

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

$$\begin{aligned}\mathbf{x}_k &= \mathbf{M}_{0\to k}(\mathbf{x})\\ &= \mathbf{M}_{k-1,k}\mathbf{M}_{k-2,k-1}\ldots\mathbf{M}_{1,2}\mathbf{M}_{0,1}\mathbf{x}\\ &= \mathbf{M}_k\mathbf{M}_{k-1}\ldots\mathbf{M}_2\mathbf{M}_1\mathbf{x}.\end{aligned}$$

This gives the final form of the observation term,

$$J^o(\mathbf{x}) = \frac{1}{2}\sum_{k=0}^{K}\left(\mathbf{y}_k^o - \mathbf{H}_k\mathbf{M}_k\mathbf{M}_{k-1}\ldots\mathbf{M}_2\mathbf{M}_1\mathbf{x}\right)^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k^o - \mathbf{M}_k\mathbf{M}_{k-1}\ldots\mathbf{M}_2\mathbf{M}_1\mathbf{x}\right).$$

Algorithm 2.2 4D-Var

$n = 0$, $\mathbf{x} = \mathbf{x}_0$
while $||\nabla J|| > \epsilon$ **or** $n \leq n_{\max}$
(1) compute J with the direct model $\mathbf{M}$ and $\mathbf{H}$
(2) compute ∇J with adjoint model $\mathbf{M}^{\mathrm{T}}$ and $\mathbf{H}^{\mathrm{T}}$ (reverse mode)
gradient descent and update of $\mathbf{x}_{n+1}$
$n = n + 1$
end

Now we can compute the gradient directly (whereas in the previous subsection we computed the variation, δJ):

$$\nabla J(\mathbf{x}) = \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right) - \sum_{k=0}^{K} \mathbf{M}_1^T \mathbf{M}_2^T \ldots \mathbf{M}_{k-1}^T \mathbf{M}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k^{\mathrm{o}} - \mathbf{M}_k \mathbf{M}_{k-1} \ldots \mathbf{M}_2 \mathbf{M}_1 \mathbf{x}\right).$$

If we denote the innovation vector as

$$\mathbf{d}_k = \mathbf{y}_k^{\mathrm{o}} - \mathbf{H}_k \mathbf{M}_k \mathbf{M}_{k-1} \ldots \mathbf{M}_2 \mathbf{M}_1 \mathbf{x},$$

then we have

$$\begin{aligned} -\nabla J^{\mathrm{o}}(\mathbf{x}) &= \sum_{k=0}^{K} \mathbf{M}_1^T \mathbf{M}_2^T \ldots \mathbf{M}_{k-1}^T \mathbf{M}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{d}_k \\ &= \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{d}_0 + \mathbf{M}_1^T \mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{d}_1 + \mathbf{M}_1^T \mathbf{M}_2^T \mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{d}_2 + \cdots \\ &\quad + \mathbf{M}_1^T \ldots \mathbf{M}_{K-1}^T \mathbf{M}_K^T \mathbf{H}_K^T \mathbf{R}_K^{-1} \mathbf{d}_K \\ &= \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{d}_0 + \mathbf{M}_1^T \left[\mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{d}_1 + \mathbf{M}_2^T \left[\mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{d}_2 + \cdots + \mathbf{M}_K^T \mathbf{H}_K^T \mathbf{R}_K^{-1} \mathbf{d}_K\right]\right]. \end{aligned}$$

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.

2.4.5.2 ▪ Algorithm

For Algorithm 2.2, in step (1) we use the equations

$$\mathbf{d}_k = \mathbf{y}_k^{\mathrm{o}} - \mathbf{H}_k \mathbf{M}_k \mathbf{M}_{k-1} \ldots \mathbf{M}_2 \mathbf{M}_1 \mathbf{x}$$

and

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right)^T \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right) + \sum_{k=0}^{K} \mathbf{d}_k^T \mathbf{R}_k^{-1} \mathbf{d}_k.$$

In step (2), we use

$$\begin{aligned} \nabla J(\mathbf{x}) = \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right) - \big[\mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{d}_0 + \mathbf{M}_1^T \big[\mathbf{H}_1^T \mathbf{R}_1^{-1} \mathbf{d}_1 + \mathbf{M}_2^T \big[\mathbf{H}_2^T \mathbf{R}_2^{-1} \mathbf{d}_2 + \cdots \\ + \mathbf{M}_K^T \mathbf{H}_K^T \mathbf{R}_K^{-1} \mathbf{d}_K\big]\big]\big]. \end{aligned}$$

2.4.5.3 ▪ A very simple scalar example

We consider an example with a single observation at time step 3 and a known background at time step 0. In this case, the 4D-Var cost function (2.37) for determining the

initial state becomes scalar,

$$J(x_0) = \frac{1}{2}\frac{\left(x_0 - x_0^{\mathrm{b}}\right)^2}{\sigma_B^2} + \frac{1}{2}\sum_{k=1}^{K}\frac{\left(x_k - x_k^{\mathrm{o}}\right)^2}{\sigma_R^2},$$

where σ_B^2 and σ_R^2 are the (known) background and observation error variances, respectively. With a single observation at time step 3, the cost function is

$$J(x_0) = \frac{1}{2}\frac{\left(x_0 - x_0^{\mathrm{b}}\right)^2}{\sigma_B^2} + \frac{1}{2}\frac{\left(x_3 - x_3^{\mathrm{o}}\right)^2}{\sigma_R^2}.$$

The minimum is reached when the gradient of J disappears,

$$J'(x_0) = 0,$$

which can be computed as

$$\frac{\left(x_0 - x_0^{\mathrm{b}}\right)}{\sigma_B^2} + \frac{\left(x_3 - x_3^{\mathrm{o}}\right)}{\sigma_R^2}\frac{dx_3}{dx_2}\frac{dx_2}{dx_1}\frac{dx_1}{dx_0} = 0. \tag{2.48}$$

We now require a dynamic relation between the x_k's to compute the derivatives. To this end, let us take the most simple linear forecast model,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\alpha x,$$

with α a known positive constant. This is a typical model for describing decay, for example, of a chemical compound whose behavior over time is then given by

$$x(t) = x(0)\mathrm{e}^{-\alpha t}.$$

To obtain a discrete representation of the dynamics, we can use an upstream finite difference scheme [Strikwerda, 2004],

$$x(t_{k+1}) - x(t_k) = (t_{k+1} - t_k)\left[-\alpha x(t_{k+1})\right], \tag{2.49}$$

which can be rewritten in the explicit form

$$x(t + \Delta t) = \left(\frac{1}{1 + \alpha\Delta t}\right)x(t),$$

where we have assumed a fixed time step, $\Delta t = t_{k+1} - t_k$, for all k. We thus have the scalar relation

$$x_{k+1} = M(x_k) = \gamma x_k, \tag{2.50}$$

where the constant is

$$\gamma = \frac{1}{1 + \alpha\Delta t}.$$

The necessary condition (2.48) then becomes

$$\frac{\left(x_0 - x_0^{\mathrm{b}}\right)}{\sigma_B^2} + \frac{\left(x_3 - x_3^{\mathrm{o}}\right)}{\sigma_R^2}\gamma^3 = 0.$$

This can be solved for x_0 and then for x_3 to obtain the analyzed state

$$\begin{aligned} x_0 &= x_0^{\mathrm{b}} + \frac{\gamma^3\sigma_B^2}{\sigma_R^2}(x_3^{\mathrm{o}} - x_3) \\ &= x_0^{\mathrm{b}} + \frac{\gamma^3\sigma_B^2}{\sigma_R^2}\left(x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}}\right) \\ &= \frac{\sigma_R^2}{\sigma_R^2 + \gamma^6\sigma_B^2} x_0^{\mathrm{b}} + \frac{\gamma^3\sigma_B^2}{\sigma_R^2 + \gamma^6\sigma_B^2}\left(x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}}\right) \\ &= x_0^{\mathrm{b}} + \frac{\gamma^3\sigma_B^2}{\sigma_R^2 + \gamma^6\sigma_B^2}\left[x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}}\right], \end{aligned}$$

where we have added and subtracted x_0^{b} to obtain the last line and used the system dynamics (2.50). Finally, by again using the dynamics, we find the 4D-Var solution

$$x_3 = \gamma^3 x_0^{\mathrm{b}} + \frac{\gamma^6\sigma_B^2}{\sigma_R^2 + \gamma^6\sigma_B^2}\left[x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}}\right]. \tag{2.51}$$

Let us examine some asymptotic cases. If the parameter α tends to zero, then the dynamic gain, γ, tends to one and the model becomes stationary, with

$$x_{k+1} = x_k.$$

The solution then tends to the 3D-Var case, with

$$x_3 = x_0 = x_0^{\mathrm{b}} + \frac{\sigma_B^2}{\sigma_R^2 + \sigma_B^2}\left[x_3^{\mathrm{o}} - x_0^{\mathrm{b}}\right]. \tag{2.52}$$

If the model is stationary, we can thus use all observations whenever they become available, exactly as in the 3D case.

The other asymptotic occurs when the step size tends to infinity and the dynamic gain goes to zero. The dynamic model becomes

$$x_{k+1} = 0,$$

with the initial condition $x_0 = x_0^{\mathrm{b}}$, and there is thus no connection between states at different time steps. Finally, if the observation is perfect, then $\sigma_R^2 = 0$ and

$$x_3 = x_3^{\mathrm{o}}.$$

But there is no link to x_0, and there is once again no dynamical connection between states at two different instants.

2.4.6 ▪ Practical variants of 3D-Var and 4D-Var

We have described above the simplest classical 3D-Var and 4D-Var algorithms. To overcome the numerous problems encountered in their implementation, there are several extensions and variants of these methods. We will describe two of the most important here. Further details can be found in Chapter 5.

2.4.6.1 ▪ Incremental 3D-Var and 4D-Var

We saw above that the adjoint of the complete model (2.40) is required for computing the gradient of the cost function. In NWP, the full nonlinear model is extremely complex [Kalnay, 2003]. To alleviate this, Courtier et al. [1994] proposed an incremental approach to variational assimilation, several variants of which now exist. Basically, the idea is to simplify the dynamical model (2.41) to obtain a formulation that is cheaper for the adjoint computation. To do this, we modify the tangent model (2.43), which becomes

$$\delta \mathbf{x}_{k+1} = \mathbf{L}_{k+1} \delta \mathbf{x}_k, \quad k = 0, 1, \ldots, K-1, \tag{2.53}$$

where $\mathbf{L}_k$ is an appropriately chosen simplified version of the Jacobian operator M_k. To preserve consistency, the basic model (2.41) must be appropriately modified so that the TLM corresponding to a known (e.g., from the background) reference solution, $\mathbf{x}_k^{(0)}$, is given by (2.53). This is easily done by letting the initial condition

$$\mathbf{x}_0 = \mathbf{x}_0^{(0)} + \delta \mathbf{x}_0$$

evolve according to (2.53) into

$$\mathbf{x}_k = \mathbf{x}_k^{(0)} + \delta \mathbf{x}_k.$$

The resulting dynamics are then linear.

Several possibilities exist for simplifying the objective function (2.40). One can linearize the observation operator H_k, as was done for the model M_k. We use the substitution

$$H_k(\mathbf{x}_k) \longmapsto H_k(\mathbf{x}_k^{(0)}) + \mathbf{N}_k \delta \mathbf{x}_k,$$

where $\mathbf{N}_k$ is some simplified linear approximation, which could be the Jacobian of H_k at $\mathbf{x}_k$. The objective function (2.40) then becomes

$$\begin{aligned} J_1(\delta \mathbf{x}_0) = {} & \frac{1}{2} \left(\delta \mathbf{x}_0 + \mathbf{x}_0^{(0)} - \mathbf{x}_0^{\mathrm{b}} \right)^T \left(\mathbf{P}_0^{\mathrm{b}} \right)^{-1} \left(\delta \mathbf{x}_0 + \mathbf{x}_0^{(0)} - \mathbf{x}_0^{\mathrm{b}} \right) \\ & + \frac{1}{2} \sum_{k=0}^{K} (\mathbf{N}_k \delta \mathbf{x}_k - \mathbf{d}_k)^T \mathbf{R}_k^{-1} (\mathbf{N}_k \delta \mathbf{x}_k - \mathbf{d}_k), \end{aligned} \tag{2.54}$$

where the $\delta \mathbf{x}_k$ satisfy (2.53) and the innovation at time k is $\mathbf{d}_k = \mathbf{y}_k - H_k(\mathbf{x}_k^{(0)})$. This objective function, J_1, is quadratic in the initial perturbation $\delta \mathbf{x}_0$, and the minimizer, $\delta \mathbf{x}_{0,m}$, defines an updated initial state

$$\mathbf{x}_0^{(1)} = \mathbf{x}_0^{(0)} + \delta \mathbf{x}_{0,m},$$

from which a new solution, $\mathbf{x}_k^{(1)}$, can be computed using the dynamics (2.41). Then we loop and repeat the whole process for $\mathbf{x}_k^{(1)}$. This defines a system of two-level nested loops (outer and inner) for minimizing the original cost function (2.40). The savings are thanks to the flexible choice that is possible for the simplified linearized operators $\mathbf{L}_k$ and $\mathbf{N}_k$. These can be chosen to ensure a reasonable trade-off between ease of implementation and physical fidelity. One can even modify the operator $\mathbf{L}_k$ in (2.53) during the minimization by gradually introducing more complex dynamics in the successive outer loops—this is the multi-incremental approach that is described in Section 5.4.1. Convergence issues are of course a major concern—see, for example, Tremolet [2007a].

These incremental methods together with the adjoint approach are what make variational assimilation computationally tractable. In fact, they have been used until now in most operational NWP systems that employ variational DA.

2.4.6.2 ▪ FGAT 3D-Var

This method, "first guess at appropriate time," (abbreviated FGAT) is best viewed as a special case of 4D-Var. It is in fact an extreme case of the incremental approach (2.53)–(2.54), in which the simplified linear operator $\mathbf{L}_k$ is set equal to the identity.

The process is 4D in the sense that the observations, distributed over the assimilation window, are compared with the computed values in the time integration of the assimilating model. But it is 3D because the minimization of the cost function (2.54) does not use the correct dynamics, i.e.,

$$\delta \mathbf{x}_{k+1} = \delta \mathbf{x}_k, \quad k = 0, 1, \ldots, K-1.$$

The FGAT 3D-Var approach, using a unique minimization loop (there is no nesting any more), has been shown to improve the accuracy of the assimilated variables. The reason for this is simple: FGAT uses a more precise innovation vector than standard 3D-Var, where all observations are compared with the same first-guess field.

2.4.7 ▪ Extensions and complements

2.4.7.1 ▪ Parameter estimation

If we want to optimize a set of parameters,

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

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

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

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

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

$$\text{or} \quad J_2^{\mathrm{b}}(\alpha) = \| \nabla \alpha - \beta \|^2.$$

2.4.7.2 ▪ 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 then define the *incremental 4D-Var algorithm*—see above.

2.4.7.3 ▪ Preconditioning

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

Preconditioning [Golub and van Loan, 2013] is a technique for improving the condition number and thus accelerating the convergence of the optimization. We make a change of variable

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

such that

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

where $\mathbf{L}$ is a given simple matrix. This is commonly used in meteorology and oceanography. The modified cost function is

$$\tilde{J}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{T}\mathbf{w} + \frac{1}{2}(\mathbf{H}\mathbf{L}\mathbf{w} - \mathbf{d})^{T}\mathbf{R}^{-1}(\mathbf{H}\mathbf{L}\mathbf{w} - \mathbf{d}),$$

and its Hessian is equal to

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

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

2.4.7.4 ▪ Covariance matrix modeling

The modeling of the covariance matrices of the background error $\mathbf{B}$ and the observation error $\mathbf{R}$ is an important operational research subject. Reduced-cost models are particularly needed when the matrices are of high dimensions—in weather forecasting or turbulent flow control problems, for example, this can run into tens of millions. One may also be interested in having better-quality approximations of these matrices.

In background error covariance modeling [Fisher, 2003], compromises have to be made to produce a computationally viable model. Since we do not have access to the true background state, we must either separate out the information about the statistics of background error from the innovation statistics or derive statistics for a surrogate quantity. Both approaches require assumptions to be made, for example about the statistical properties of the observation error. The "separation" approach can be addressed by running an ensemble of randomly perturbed predictions, drawn from relevant distributions. This method of generating surrogate fields of background error is strongly related to the EnKF, which is fully described in Chapter 6—see also Evensen [2009].

Other approaches for modeling the $\mathbf{B}$ matrix by reduced bases, factorization, and spectral methods are fully described in Chapter 5.

2.4.7.5 ▪ Model error

In standard variational assimilation, we invert for the initial condition only. The underlying hypothesis that the model is perfectly known is not a realistic one. In fact, to take into account eventual model error, we should add an appropriate error term to the state equation and insert a cost term into the objective function. We thus arrive at a *parameter identification* inverse problem, similar to those already studied above in Section 2.3.

In the presence of model uncertainty, the state equation and objective functions become (see also the above equations (2.38) and (2.39))

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathscr{M}(\mathbf{x}) + \eta(t) \quad \text{in } \Omega \times [0, T], \\ \mathbf{x}(t=0) = \mathbf{x}_0, \end{cases}$$

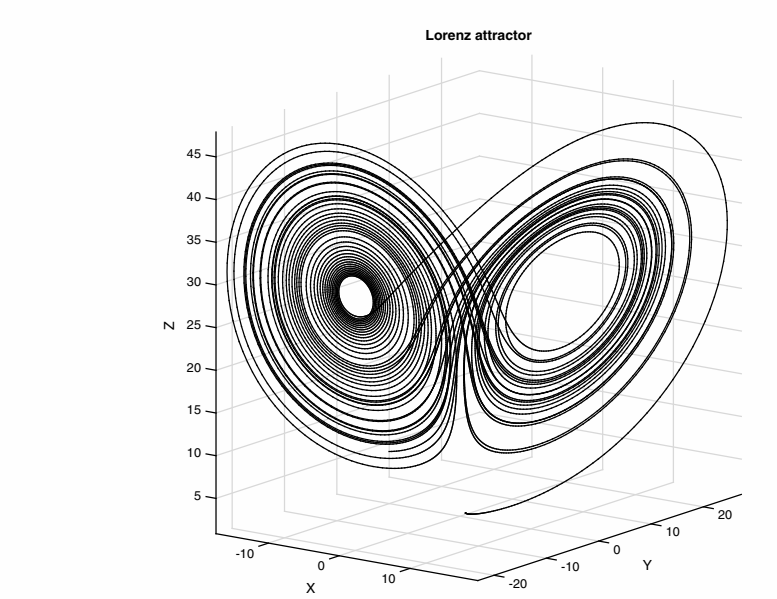


Figure 2.5. *Simulation of the chaotic Lorenz-63 system of three equations.*

where $\eta(t)$ is a suitably uncorrelated white noise. The new cost functional is

$$J(\mathbf{x}_0,\eta)=\frac{1}{2}\left\|\mathbf{x}_0-\mathbf{x}^{\mathrm{b}}\right\|^2_{\mathscr{X}}+\frac{1}{2}\int_0^T\|\mathbf{y}(t)-\mathscr{H}(\mathbf{x}(\mathbf{x}_0,t))\|^2_{\mathscr{O}}\,\mathrm{d}t+\frac{1}{2}\int_0^T\|\eta(t)\|^2_{\mathscr{E}}\,\mathrm{d}t,$$

where the model noise has covariance matrix $\mathbf{Q}$, which we suppose to be uncorrelated in time and uncorrelated with the background and observation errors. However, in cases with high dimensionality, this approach is not feasible, especially for practical problems. Numerous solutions have been proposed to overcome this problem—see Griffith and Nichols [2000], Tremolet [2007b], Vidard et al. [2004], and Tremolet [2007c].

2.5 ▪ Numerical examples

2.5.1 ▪ DA for the Lorenz equation

We study the nonlinear Lorenz system of equations [Lorenz, 1963],

$$\begin{aligned}\frac{dx}{dt}&=-\sigma(x-y),\\ \frac{dy}{dt}&=\rho x-y-xz,\\ \frac{dz}{dt}&=xy-\beta z,\end{aligned}$$

which exhibits chaotic behavior when we fix the parameter values $\sigma=10$, $\rho=28$, and $\beta=8/3$ (see Figure 2.5) This equation is a simplified model for atmospheric convection and is an excellent example of the lack of predictability. It is ill-posed in the

sense of Hadamard. In fact, the solution switches between two stable orbits around the points

$$\left(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1\right)$$

and

$$\left(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1\right).$$

We now perform 4D-Var DA on this equation with only the observation term,

$$J^{\mathrm{o}}(\mathbf{x}) = \frac{1}{2}\sum_{i=0}^{n}\left(\mathbf{y}_k^{\mathrm{o}} - \mathbf{H}_k(\mathbf{x}_k)\right)^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k^{\mathrm{o}} - \mathbf{H}_k(\mathbf{x}_k)\right).$$

This relatively simple model enables us to study a number of important effects and to answer the following practical questions:

- What is the influence of observation noise?
- What is the influence of the initial guess?
- What is the influence of the length of the assimilation window and the number of observations?

In addition, we can compare the performance of the standard 4D-Var with that of an incremental 4D-Var algorithm. All computations are based on the codes provided by A. Lawless of the DARC (Data Assimilation Research Centre) at Reading University [Lawless, 2002]. Readers are encouraged to obtain the software and experiment with it.

The assimilation results shown in Figures 2.6 and 2.7 were obtained from twin experiments with the following conditions:

- True initial condition is $(1.0, 1.0, 1.0)$.
- Initial guess is $(1.2, 1.2, 1.2)$.
- Time step is 0.05 seconds.
- Assimilation window is 2 seconds.
- Forecast window is 3 seconds.
- Observations are every 2 time steps.
- Number of outer loops (for incremental 4D-Var) is 4.

We remark that the incremental algorithm produces a more accurate forecast, over a longer period, in this case—see Figure 2.7.

2.5.2 ▪ Additional DA examples

Numerous examples of variational DA can be found in the advanced Chapters 4, 5, and 7, as well as in the applications sections—see Part III. Another rich source is the training material of the ECMWF [Bouttier and Courtier, 1997]—see `http://www.ecmwf.int/en/learning/education-material/lecture-notes`.

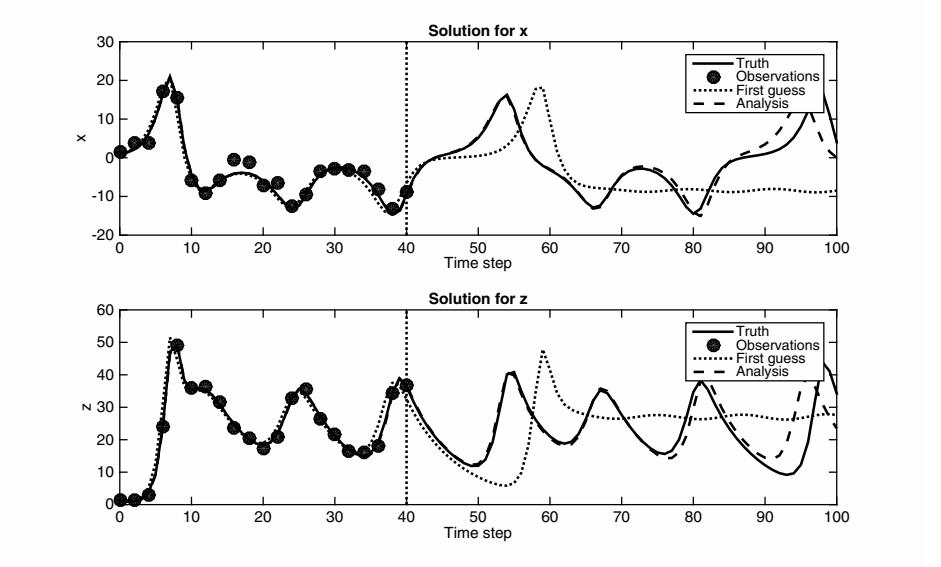


Figure 2.6. *Assimilation of the Lorenz-63 equations by standard 4D-Var, based on a twin experiment. The assimilation window is from step 0 to step 40 (2 seconds). The forecast window is from step 41 to 100 (3 seconds).*

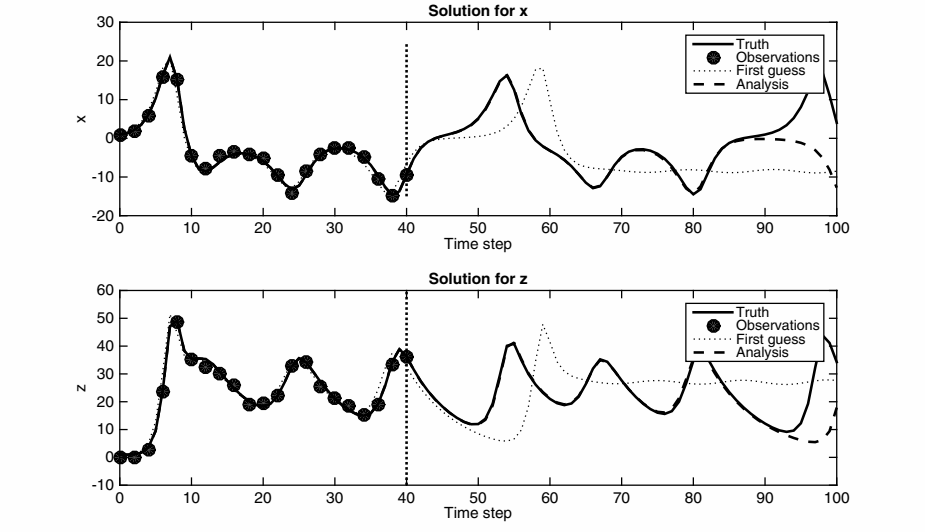


Figure 2.7. *Assimilation of the Lorenz-63 equations by incremental 4D-Var, based on a twin experiment. The assimilation window is from step 0 to step 40 (2 seconds). The forecast window is from step 41 to 100 (3 seconds).*