Part I

Basic methods and algorithms for data assimilation

After an introduction that sets up the general theoretical framework of the book and provides several simple (but important) examples, the two approaches for the solution of DA problems are presented: classical (variational) assimilation and statistical (sequential) assimilation. We begin with the variational approach. Here we take an *optimal control* viewpoint based on classical variational calculus and show its impressive power and generality. A sequence of carefully detailed inverse problems, ranging from an ODE-based to a nonlinear PDE-based case, are explained. This prepares the ground for the two major variational DA algorithms, 3D-Var and 4D-Var, that are currently used in most large-scale forecasting systems. For statistical DA, we employ a Bayesian approach, starting with optimal statistical estimation and showing how the standard KF is derived. This lays the foundation for various extensions, notably the ensemble Kalman filter (EnKF).

Chapter 1

Introduction to data assimilation and inverse problems

1.1 ▪ Introduction

What exactly is DA? The simplest view is that it is an approach/method for combining observations with model output with the objective of improving the latter. But do we really need DA? Why not just use the observations and average them or extrapolate them (as is done with regression techniques (McPherson, 2001), or just long-term averaging)? The answer is that we want to predict the state of a system, or its future, in the best possible way! For that we need to rely on models. But when models are not corrected periodically by reality, they can be of little value. Thus, we need to fit the model state in an optimal way to the observations, before an analysis or prediction is made. This fitting of a model to observations is a special case (but highly typical) of an *inverse problem*.

According to J. B. Keller [1966], two problems are inverse to each other if "the *formulation* of each involves all or part of the *solution* of the other." One of the two is named the direct problem, whereas the other is the inverse problem. The direct problem is usually the one that we can solve satisfactorily/easily. There is a back-and-forth transmission of information between the two. This is depicted in Figure 1.1, which represents a typical case: we replace the unknown (or partially known) medium by a model that depends on some unknown model parameters, $\boldsymbol{m}$. The inverse problem involves reversing the arrows—by comparing the simulations and the observations (at the array) to find the model parameters. In fact, the direct problem involves going from cause to effect, whereas the inverse problem attempts to go from the effects to the cause.

The comparison between model output and observations is performed by some form of optimization—recall that we seek an optimal match between simulations of the model and measurements taken of the system that we are trying to elucidate. This optimization takes two forms: classical and statistical. Let us explain. Classical optimization involves minimization of a positive, usually quadratic cost function that expresses the quantity that we seek to optimize. In most of the cases that we will deal with, this will be a function of the error between model and measurements—in this case we will speak of least-squares error minimization. The second form, statistical optimization, involves minimization of the variability or uncertainty of the model error and is based on statistical estimation theory.

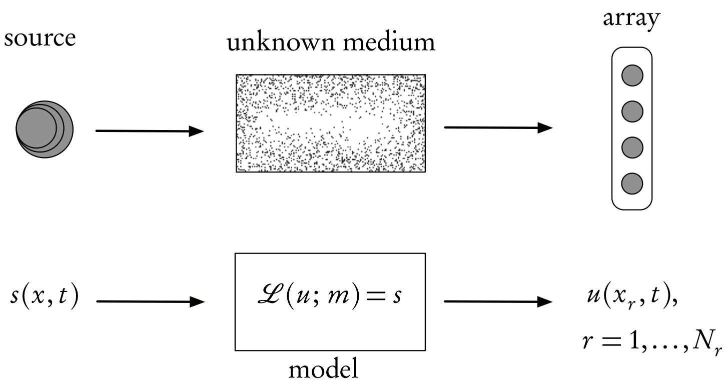


Figure 1.1. *Ingredients of an inverse problem: the physical reality (top) and the direct mathematical model (bottom). The inverse problem uses the difference between the model-predicted observations, u (calculated at the receiver array points, x_r), and the real observations measured on the array to find the unknown model parameters, m, or the source, s (or both).*

The main sources of inverse problems are science (social sciences included!) and engineering—in fact any process that we can model and measure satisfactorily. Often these problems concern the determination of the properties of some inaccessible region from observations on the boundary of the region, or at discrete instants over a given time interval. In other words, our information is *incomplete*. This incompleteness is the source of the major difficulties (and challenges...) that we will encounter in the solution of DA and inverse problems.

1.2 ▪ Uncertainty quantification and related concepts

Definition 1.1. *Uncertainty quantification (UQ) is the science of quantitative characterization and reduction of uncertainties in both computational and real-world applications. It tries to determine how likely certain outcomes are if some aspects of the system are not exactly known.*

The system-science paradigm, as expounded in Jordan [2015], exhibits the important place occupied by DA and inverse methods within the "deductive spiral"—see Figure 1.2. These methods furnish an essential link between the real world and the model of the system. They are intimately related to the concepts of *validation* and *verification*. Verification asks the question, "are we solving the equations correctly?"—this is an exercise in mathematics. Validation asks, "are we solving the correct equations?"—this is an exercise in physics. In geophysics, for example, the concept of validation is replaced by *evaluation*, since complete validation is not possible.

UQ is a basic component of model validation. In fact it is vital for characterizing our confidence in results coming out of modeling and simulation and provides a mathematically rigorous certification that is often needed in decision-making. In fact, it gives a precise notion of what constitutes a validated model by replacing the subjective concept of *confidence* by mathematically rigorous methods and measures.

There are two major categories of uncertainties: epistemic and aleatory. The first is considered to be reducible in that we can control it by improving our knowledge of

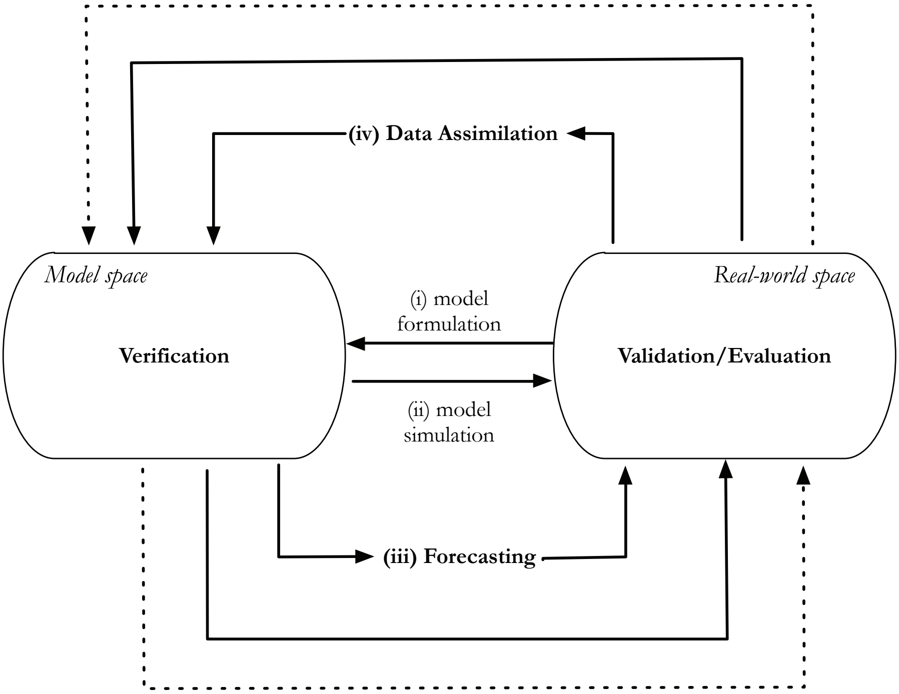


Figure 1.2. *The deductive spiral of system science (adapted from Jordan [2015]). The bottom half represents the direct problem (from model to reality); the top half represents the inverse problem (from reality to model). Starting from the center, with* (i) *model formulation and* (ii) *simulation, one works one's way around iteratively over* (iii) *forecasting and* (iv) *DA, while passing through the two phases of UQ (validation/evaluation and verification).*

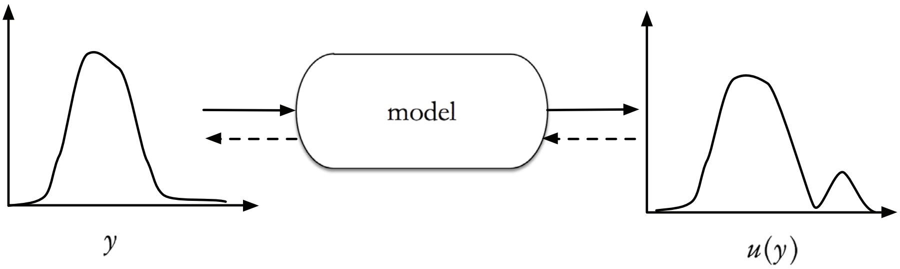


Figure 1.3. *UQ for a random quantity y: uncertainty propagation (left to right); uncertainty definition (right to left).*

the system. The second is assumed to be irreducible and has to do with the inherent noise in, or stochastic nature of, any natural system. Any computation performed under uncertainty will forcibly result in predictive simulations (see the introduction to Chapter 3 for more details on this point).

Uncertainty, in models of physical systems, is almost always represented as a probability density function (PDF) through samples, parameters, or kernels. The central objective of UQ is then to represent, propagate, and estimate this density—see Figure 1.3.

As a process, UQ can be decomposed into the following steps:

1. Define the system of interest, its response, and the desired performance measures.
2. Write a mathematical formulation of the system—governing equations, geometry, parameter values.
3. Formulate a discretized representation and the numerical methods and algorithms for its solution.
4. Perform the simulations and the analysis.
5. Loop back to step 1.

The numerical simulations themselves can be decomposed into three steps:

1. DA, whose objective is to compute the PDFs of the input quantities of interest. This is the major concern of this book and the methods described herein.
2. Uncertainty propagation, whose objective is to compute the PDFs of the output quantities of interest. This is usually the most complex and computationally intensive step and is generally based on Monte Carlo and stochastic Galerkin (finite element) methods—see Le Maitre and Knio [2010].
3. Certification, whose objective is to estimate the likelihood of specific outcomes and compare them with risk or operating margins.

For a complete, recent mathematical overview of UQ, the reader is referred to Owhadi et al. [2013]. There are a number of research groups dedicated to the subject—please consult the websites of UQ groups at Stanford University,[3] MIT,[4] and ETH Zurich[5] (for example).

1.3 ▪ Basic concepts for inverse problems: Well- and ill-posedness

There is a fundamental, mathematical distinction between the direct and the inverse problem: direct problems are (invariably) well-posed, whereas inverse problems are (notoriously) ill-posed. Hadamard [1923] defined the concept of a well-posed problem as opposed to an ill-posed one.

Definition 1.2. *A mathematical model for a physical problem is* well-posed *if it possesses the following three properties:*

WP1 *Existence of a solution.*

WP2 *Uniqueness of the solution.*

WP3 *Continuous dependence of the solution on the data.*

[3] `http://web.stanford.edu/group/uq/`.

[4] `http://uqgroup.mit.edu`.

[5] `http://www.sudret.ibk.ethz.ch`.

Note that existence and uniqueness together are also known as "identifiability," and the continuous dependence is related to the "stability" of the inverse problem. A more rigorous mathematical formulation is the following (see Kirsch [1996]).

Definition 1.3. *Let X and Y be two normed spaces, and let $K : X \to Y$ be a linear or nonlinear map between the two. The problem of finding x given y such that*

$$Kx = y$$

is well-posed if the following three properties hold:

WP1 *Existence—for every $y \in Y$ there is (at least) one solution $x \in X$ such that $Kx = y$.*

WP2 *Uniqueness—for every $y \in Y$ there is at most one $x \in X$ such that $Kx = y$.*

WP3 *Stability—the solution, x, depends continuously on the data, y, in that for every sequence $\{x_n\} \subset X$ with $Kx_n \to Kx$ as $n \to \infty$, we have that $x_n \to x$ as $n \to \infty$.*

This concept of ill-posedness will be the "red thread" running through the entire book. It will help us to understand and distinguish between direct and inverse models. It will provide us with basic comprehension of the methods and algorithms that will be used to solve inverse problems. Finally, it will assist us in the analysis of what went wrong in our attempt to solve the inverse problems.

1.4 ▪ Examples of direct and inverse problems

Take a parameter-dependent dynamical system,

$$\frac{dz}{dt} = g(t, z; \theta), \qquad z(t_0) = z_0,$$

with g known, z_0 an initial state, $\theta \in \Theta$ (a space, or set, of possible parameter values), and the state $z(t) \in \mathbb{R}^n$. We can now define the two classes of problems.

Direct Given parameters θ and initial state z_0, find $z(t)$ for $t \geq t_0$.

Inverse Given observations $z(t)$ for $t \geq t_0$, find $\theta \in \Theta$.

Since the observations are incompletely known (over space-time), they must be modeled by an observation equation,

$$f(t, \theta) = \mathcal{H} z(t, \theta),$$

where $\mathcal{H}$ is the observation operator (which could, in ideal situations, be the identity). Usually we have a finite number, p, of discrete (space-time) observations

$$\left\{\tilde{y}_j\right\}_{j=1}^{p},$$

where

$$\tilde{y}_j \approx f(t_j, \theta)$$

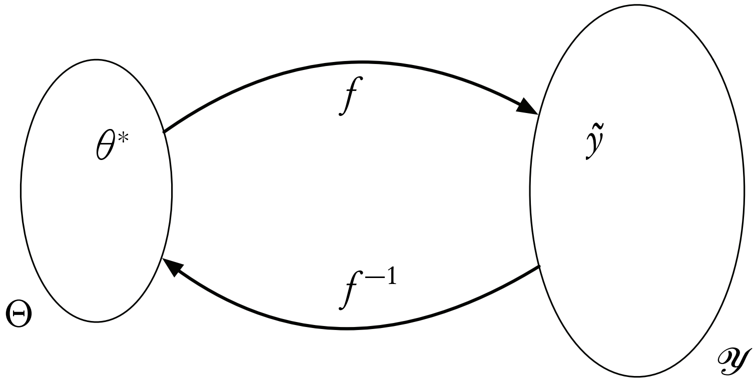
and the approximately equal sign denotes the possibility of measurement errors.

We now present a series of simple examples that clearly illustrate the three properties of well-posedness.

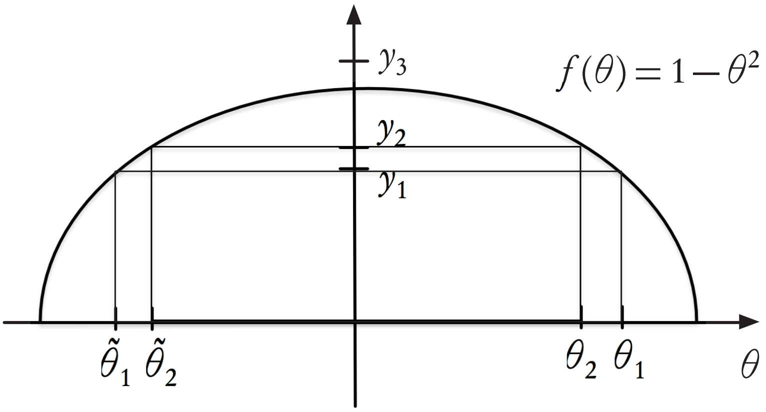
Example 1.4. (Simplest case—inspired by an oral presentation of H.T. Banks). Suppose that we have one observation, $\tilde{y}$, for $f(\theta)$ and we want to find the pre-image

$$\theta^* = f^{-1}(\tilde{y})$$

for a given $\tilde{y}$:



This problem can be severely ill-posed! Consider the following function:



Nonexistence There is no θ_3 such that $f(\theta_3) = y_3$.

Nonuniqueness $y_j = f(\theta_j) = f(\tilde{\theta}_j)$ for $j = 1, 2$.

Lack of continuity $|y_1 - y_2|$ small $\nRightarrow \left|f^{-1}(y_1) - f^{-1}(y_2)\right| = \left|\theta_1 - \tilde{\theta}_2\right|$ small.

Note that all three well-posedness properties, WP1, WP2, and WP3, are violated by this very basic case. Why is this so important? Couldn't we just apply a good least-squares algorithm (for example) to find the best possible solution? Let's try this. We define a mismatch-type cost function,

$$J(\theta) = |y_1 - f(\theta)|^2,$$

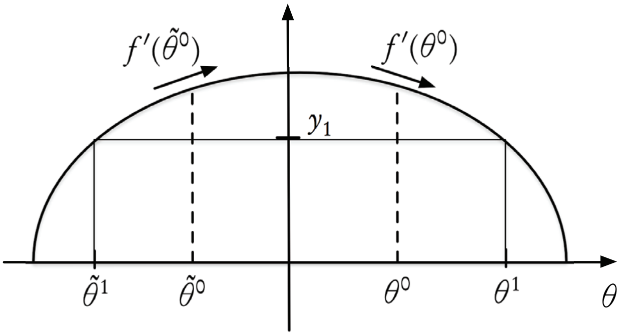
for a given y_1 and try to minimize this square error by applying a standard iterative scheme, such as direct search or gradient-based minimization [Quarteroni et al., 2007], to obtain a solution. For example, if we apply Newton's method, we obtain the following iteration:

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

where

$$J'(\theta)=2(y_1-f(\theta))\left(-f'(\theta)\right).$$

Let us graphically perform a few iterations on the above function:



- $J'(\theta^0)=2(\overbrace{y_1-f(\theta^0)}^{(-)})(\overbrace{-f'(\theta^0)}^{(+)})<0 \quad\Rightarrow\quad \theta^1>\theta^0$, etc.,
- $J'(\tilde{\theta}^0)=2(\overbrace{y_1-f(\tilde{\theta}^0)}^{(-)})(\overbrace{-f'(\tilde{\theta}^0)}^{(-)})>0 \quad\Rightarrow\quad \tilde{\theta}^1<\tilde{\theta}^0$, etc.,

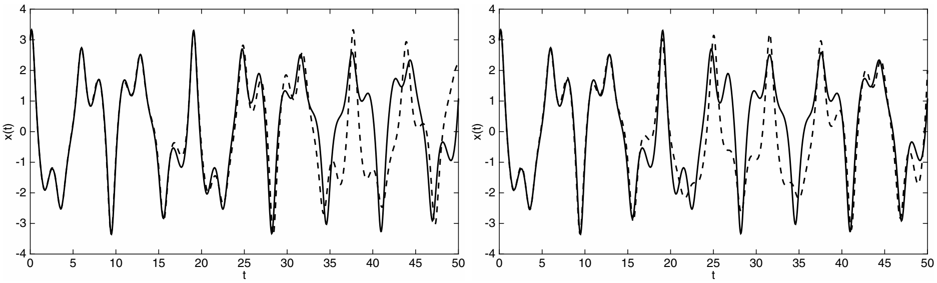
where in the last two formulas for J' we have indicated the sign (+/—) above each of the terms. We observe that in this simple case we have a highly unstable, oscillating behavior: at each step we move from positive to negative increments due to the changing sign of the gradient, and convergence is not possible. So what went wrong here? This behavior is not the fault of descent algorithms. It is a manifestation of the *inherent ill-posedness* of the problem. How to fix this problem has been the subject of much research over the past 50 years! Many remedies (fortunately) exist, such as explicit and implicit constrained optimizations, regularization, and penalization—these will be referred to, when necessary, in what follows. ■

To further appreciate the complexity, let us briefly consider one of the remedies: Tykhonov regularization (TR)—see, for example, Engl et al. [1996] and Vogel [2002]. The idea here is to replace the ill-posed problem for $J(\theta)=|y_1-f(\theta)|^2$ by a "nearby" problem for

$$J_\beta(\theta)=|y_1-f(\theta)|^2+\beta\,|\theta-\theta_0|^2,$$

where β is a suitably chosen regularization/penalization parameter. When it is done correctly, TR provides convexity and compactness,[6] thus ensuring the existence of a

[6] Convexity means that the function resembles a quadratic function, $f(x)=x^2$, with positive second derivative; compactness means that any infinite sequence of functions must get arbitrarily close to some function of the space.



(a) Initial error of 0.03%

(b) Initial error of 0.06%

Figure 1.4. *Duffing's equation with small initial perturbations. Unperturbed (solid line) and perturbed (dashed line) trajectories.*

unique solution. However, even when done correctly, it *modifies the problem*, and new solutions may be far from the original ones. In addition, it is not trivial to regularize correctly or even to know if we have succeeded in finding a solution.

Example 1.5. The highly nonlinear Duffing's equation [Guckenheimer and Holmes, 1983],

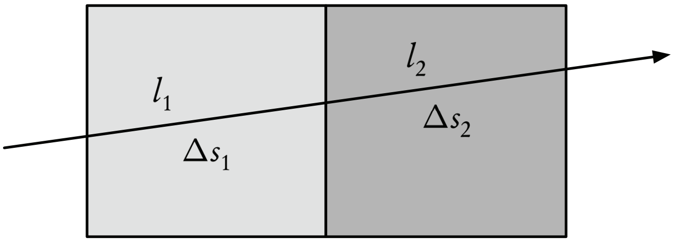
$$\ddot{x} + 0.05\dot{x} + x^3 = 7.5 \cos t,$$

exhibits great *sensitivity to the initial conditions* (WP3). We will observe that two very closely spaced initial states can lead to a large discrepancy in the trajectories.

- Let $x(0) = 3$ and $\dot{x}(0) = 4$ be the true initial state.
- Introduce an error of 0.03% in the initial state—here we have an accurate forecast until $t = 35$ (see Figure 1.4(a)).
- Introduce an error of 0.06% in the initial state—here we only have an accurate forecast until $t = 20$ (see Figure 1.4(b)).

The initial perturbations are scarcely visible (and could result from measurement error), but the terminal states can differ considerably. ∎

Example 1.6. Seismic travel-time tomography provides an excellent example of nonuniqueness (WP2):



A signal seismic ray (or any other ray used in medical or other imaging) passes through a two-parameter block model.

- The *unknowns* are the two block slownesses (inverse of seismic velocity), $(\Delta s_1, \Delta s_2)$.
- The *data* consist of the observed travel time of the ray, Δt_1.
- The *model* is the linearized travel time equation

  $$\Delta t_1 = l_1 \Delta s_1 + l_2 \Delta s_2,$$

  where l_j is the length of the ray in the jth block.

Clearly we have one equation for two unknowns, and hence there is *no unique solution*. In fact, for a given value of Δt_1, each time we fix Δs_1 we obtain a different Δs_2 (and vice versa). ■

We hope the reader is convinced, based on these disarmingly simple examples, that inverse problems present a large number of potential pathologies. We can now proceed to examine the therapeutic tools that are at our disposal for attempting to "heal the patient."

1.5 ▪ DA methods

Definition 1.7. *DA is the approximation of the true state of some physical system at a given time by combining time-distributed observations with a dynamic model in an optimal way.*

DA can be classically approached in two ways: as variational DA and as statistical[7] DA—see Figure 1.5. They will be briefly presented here and then in far more detail in Chapters 2 and 3, respectively. Newer approaches are also becoming available: nudging methods, reduced methods, ensemble methods, and hybrid methods that combine variational and statistical[8] approaches. These are the subject of the chapters on advanced methods—see Part II.

In both we seek an optimal solution—statistically we will, for example, seek a solution with minimum variance, whereas variationally we will seek a solution that minimizes a suitable cost (or error) function. In fact, in certain special cases the two approaches are identical and provide exactly the same solution. However, the statistical approach, though often more complex and time-consuming, can provide a richer information structure: an average solution and some characteristics of its variability (probability distribution). Clearly a temperature forecast of "15°C for tomorrow" is much less informative than a forecast of "an *average* of 15°C with a *standard deviation* of 1.5°C," or "the *probability* of a temperature below 10°C is 0.125 for tomorrow," or, as is now quite common (on our smartphones), "there is a 60% chance of rain at 09h00 in New York." Ideally (and this will be our recommendation), one should attempt to combine the two into a single, "hybrid" approach. We can then take advantage of the relative rapidity and robustness of the variational approach, and at the same time obtain an information-rich solution thanks to the statistical/probabilistic approach. This is easily said but (as we will see) is not trivial to implement and will be highly problem dependent and probably computationally expensive. However, we can and

[7] Alternatives are "filtering" or "probabilistic."
[8] *ibid.*

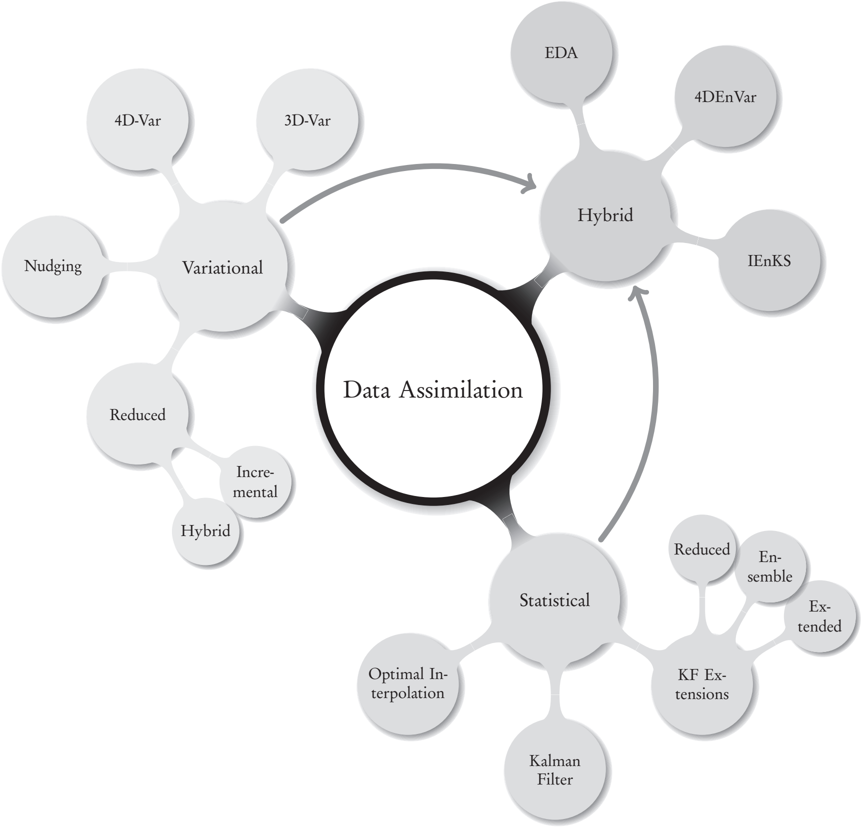


Figure 1.5. *DA methods: variational (Chapters* 2, 4, 5*); statistical (Chapters* 3, 5, 6*), hybrid (Chapter 7).*

will provide all the necessary tools (theoretical, algorithmic, numerical) and the indications for their implementation. It is worthwhile to point out that recently (since 2010) a number of the major weather-forecasting services across the world (Canada, France, United Kingdom, etc.) have started basing their operational forecasting systems on a new, hybrid approach, 4DEnVar (presented in Chapter 7), which combines 4D-Var (variational DA) with an ensemble, statistical approach.

To complete this introductory chapter, we will now briefly introduce and compare the two approaches of variational and statistical. Each one is subsequently treated, in far greater detail, in its own chapter—see Chapters 2 and 3, respectively.

1.5.1 ▪ Notation for DA and inverse problems

We begin by introducing the standard notation for DA problems as formalized by Ide et al. [1997]. We first consider a discrete model for the evolution of a physical (atmospheric, oceanic, mechanical, biological, etc.) system from time t_k to time t_{k+1}, described by a dynamic state equation

$$\mathbf{x}^{\mathrm{f}}(t_{k+1}) = \mathbf{M}_{k+1}\left[\mathbf{x}^{\mathrm{f}}(t_k)\right], \tag{1.1}$$

where $\mathbf{x}$ is the model's state vector of dimension n (see below for the definition of the superscripts) and $\mathbf{M}$ is the corresponding dynamics operator that can be time dependent. This operator usually results from a finite difference [Strikwerda, 2004] or finite element [Hughes, 1987] discretization of a (partial) differential equation (PDE). We associate an error covariance matrix $\mathbf{P}$ with the state $\mathbf{x}$ since the true state will differ from the simulated state (1.1) by random or systematic errors.

Observations, or measurements, at time t_k are defined by

$$\mathbf{y}^{\mathrm{o}} = \mathbf{H}_k\left[\mathbf{x}^t(t_k)\right] + \epsilon_k^{\mathrm{o}}, \tag{1.2}$$

where $\mathbf{H}$ is an *observation operator* that can be time dependent and ϵ^{o} is a *white noise process* with zero mean and associated covariance matrix $\mathbf{R}$ that describes instrument errors and representation errors due to the discretization. The observation vector, $\mathbf{y}_k^{\mathrm{o}} = \mathbf{y}^{\mathrm{o}}(t_k)$, has dimension p_k, which is usually much smaller than the state dimension, $p_k \ll n$.

Subscripts are used to denote the discrete time index, the corresponding spatial indices, or the vector with respect to which an error covariance matrix is defined; superscripts refer to the nature of the vectors/matrices in the DA process:

- "a" for *analysis*,
- "b" for *background* (or initial/first guess),
- "f" for *forecast*,
- "o" for *observation*, and
- "t" for the (unknown) *true* state.

Analysis is the process of approximating the true state of a physical system at a given time. Analysis is based on

- observational data,
- a model of the physical system, and
- background information on initial and boundary conditions.

An analysis that combines time-distributed observations and a dynamic model is called *data assimilation*.

Now let us introduce the continuous system. In fact, continuous time simplifies both the notation and the theoretical analysis of the problem. For a finite-dimensional system of ODEs, the equations (1.1)–(1.2) become

$$\dot{\mathbf{x}}^{\mathrm{f}} = \mathcal{M}(\mathbf{x}^{\mathrm{f}}, t),$$

and

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

where $\dot{()} = \mathrm{d}/\mathrm{d}t$ and $\mathcal{M}$ and $\mathcal{H}$ are nonlinear operators in continuous time for the model and the observation, respectively. This implies that $\mathbf{x}$, $\mathbf{y}$, and ϵ are also continuous-in-time functions. For PDEs, where there is in addition a dependence on space, attention must be paid to the function spaces, especially when performing variational analysis. Details will be provided in the next chapter. With a PDE model, the field (state) variable is commonly denoted by $\boldsymbol{u}(\mathbf{x}, t)$, where $\mathbf{x}$ represents the space variables (no longer the state variable as above!), and the model dynamics is now a nonlinear partial differential operator,

$$\mathcal{M} = \mathcal{M}\left[\partial_{\mathbf{x}}^{\alpha}, \boldsymbol{u}(\mathbf{x}, t), \mathbf{x}, t\right],$$

with $\partial_{\mathbf{x}}^{\alpha}$ denoting the partial derivatives with respect to the space variables of order up to $|\alpha| \leq m$, where m is usually equal to two and in general varies between one and four.

1.5.2 ▪ Statistical DA

Practical inverse problems and DA problems involve measured data. These data are inexact and are mixed with random noise. Only *statistical models* can provide rigorous, effective means for dealing with this measurement error. Let us begin with the following simple example.

1.5.2.1 ▪ A simple example

We want to estimate a *scalar* quantity, say the temperature or the ozone concentration at a fixed point in space. Suppose we have

- a model forecast, x^{b} (background, or a priori value), and
- a measured value, x^{o} (observation).

The simplest possible approach is to try a linear combination of the two,

$$x^{\mathrm{a}} = x^{\mathrm{b}} + w(x^{\mathrm{o}} - x^{\mathrm{b}}),$$

where x^{a} denotes the analysis that we seek and $0 \leq w \leq 1$ is a weight factor. We subtract the (always unknown) true state x^{t} from both sides,

$$x^{\mathrm{a}} - x^{\mathrm{t}} = x^{\mathrm{b}} - x^{\mathrm{t}} + w(x^{\mathrm{o}} - x^{\mathrm{t}} - x^{\mathrm{b}} + x^{\mathrm{t}}),$$

and, defining the three errors (analysis, background, observation) as

$$e^{\mathrm{a}} = x^{\mathrm{a}} - x^{\mathrm{t}}, \quad \mathrm{e}^{\mathrm{b}} = x^{\mathrm{b}} - x^{\mathrm{t}}, \quad e^{\mathrm{o}} = x^{\mathrm{o}} - x^{\mathrm{t}},$$

we obtain

$$e^{\mathrm{a}} = e^{\mathrm{b}} + w(e^{\mathrm{o}} - e^{\mathrm{b}}) = we^{\mathrm{o}} + (1-w)e^{\mathrm{b}}.$$

If we have many realizations, we can take an ensemble average,[9] denoted by $\langle\cdot\rangle$:

$$\langle e^{\mathrm{a}}\rangle = \left\langle e^{\mathrm{b}}\right\rangle + w\left(\langle e^{\mathrm{o}}\rangle - \left\langle e^{\mathrm{b}}\right\rangle\right).$$

Now if these errors are centered (have zero mean, or the estimates of the true state are *unbiased*), then

$$\langle e^{\mathrm{a}}\rangle = 0$$

also. So we are logically led to look at the *variance* and demand that it be as small as possible. The variance is defined, using the above notation, as

$$\sigma^2 = \left\langle (e - \langle e\rangle)^2\right\rangle.$$

[9] Please refer to a good textbook on probability and statistics for all the relevant definitions—e.g., DeGroot and Schervisch [2012].

So by taking variances of the error equation, and using the zero-mean property, we obtain

$$\sigma_{\mathrm{a}}^2 = \sigma_{\mathrm{b}}^2 + w^2 \left\langle \left(e^{\mathrm{o}} - e^{\mathrm{b}}\right)^2 \right\rangle + 2w \left\langle e^{\mathrm{b}} \left(e^{\mathrm{o}} - e^{\mathrm{b}}\right) \right\rangle.$$

This reduces to

$$\sigma_{\mathrm{a}}^2 = \sigma_{\mathrm{b}}^2 + w^2 \left(\sigma_{\mathrm{o}}^2 + \sigma_{\mathrm{b}}^2\right) - 2w\sigma_{\mathrm{b}}^2$$

if e^{o} and e^{b} are uncorrelated. Now, to compute a minimum, take the derivative of this last equation with respect to w and equate to zero,

$$0 = 2w\left(\sigma_{\mathrm{o}}^2 + \sigma_{\mathrm{b}}^2\right) - 2\sigma_{\mathrm{b}}^2,$$

where we have ignored all cross terms since the errors have been assumed to be independent. Finally, solving this last equation, we can write the optimal weight,

$$w_* = \frac{\sigma_{\mathrm{b}}^2}{\sigma_{\mathrm{o}}^2 + \sigma_{\mathrm{b}}^2} = \frac{1}{1 + \sigma_{\mathrm{o}}^2/\sigma_{\mathrm{b}}^2},$$

which, we notice, depends on the ratio of the observation and the background errors. Clearly $0 \le w_* \le 1$ and

- if the observation is perfect, $\sigma_{\mathrm{o}}^2 = 0$ and thus $w_* = 1$, the maximum weight;
- if the background is perfect, $\sigma_{\mathrm{b}}^2 = 0$ and $w_* = 0$, so the observation will not be taken into account.

We can now rewrite the analysis error variance as

$$\begin{aligned}
\sigma_{\mathrm{a}}^2 &= w_*^2\sigma_{\mathrm{o}}^2 + (1-w_*)^2\sigma_{\mathrm{b}}^2 \\
&= \frac{\sigma_{\mathrm{b}}^2\sigma_{\mathrm{o}}^2}{\sigma_{\mathrm{o}}^2 + \sigma_{\mathrm{b}}^2} \\
&= (1-w_*)\sigma_{\mathrm{b}}^2 \\
&= \frac{1}{\sigma_{\mathrm{o}}^{-2} + \sigma_{\mathrm{b}}^{-2}},
\end{aligned}$$

where we suppose that $\sigma_{\mathrm{b}}^2,\ \sigma_{\mathrm{o}}^2 > 0$. In other words,

$$\frac{1}{\sigma_{\mathrm{a}}^2} = \frac{1}{\sigma_{\mathrm{o}}^2} + \frac{1}{\sigma_{\mathrm{b}}^2}.$$

Finally, the analysis equation becomes

$$x^{\mathrm{a}} = x^{\mathrm{b}} + \frac{1}{1+\alpha}(x^{\mathrm{o}} - x^{\mathrm{b}}),$$

where $\alpha = \sigma_{\mathrm{o}}^2/\sigma_{\mathrm{b}}^2$. This is called the BLUE—best linear unbiased estimator—because it gives an unbiased, optimal weighting for a linear combination of two independent measurements.

We can isolate three special cases:

- If the observation is very accurate, $\sigma_{\mathrm{o}}^2 \ll \sigma_{\mathrm{b}}^2$, $\alpha \ll 1$, and thus $x^{\mathrm{a}} \approx x^{\mathrm{o}}$.

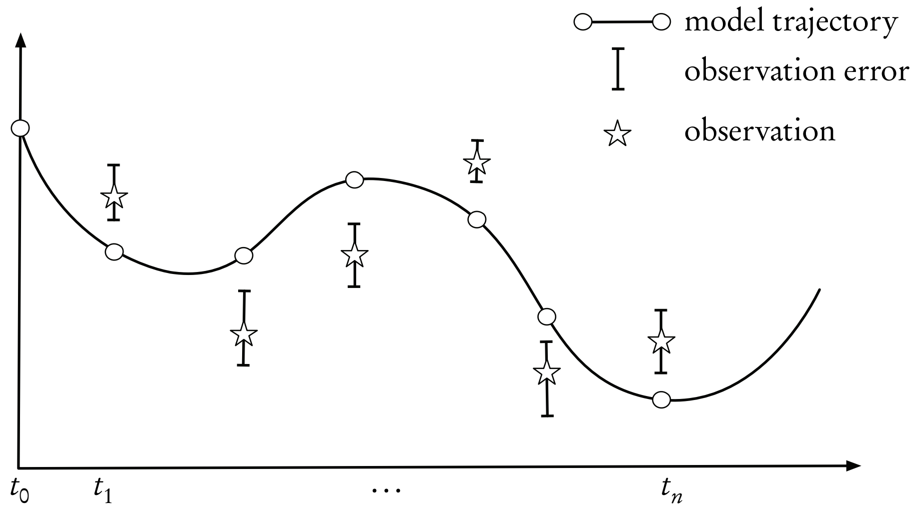


Figure 1.6. *Sequential assimilation. The x-axis denotes time; the y-axis is the assimilated variable.*

- If the background is accurate, $\alpha \gg 1$ and $x^{\mathrm{a}} \approx x^{\mathrm{b}}$.
- And, finally, if the observation and background variances are approximately equal, then $\alpha \approx 1$ and x^{a} is just the arithmetic average of x^{b} and x^{o}.

We can conclude that this simple, linear model does indeed capture the full range of possible solutions in a statistically rigorous manner, thus providing us with an "enriched" solution when compared with a nonprobabilistic, scalar response such as the arithmetic average of observation and background, which would correspond to only the last of the above three special cases.

1.5.2.2 ▪ The more general case: Introducing the Kalman filter

The above analysis of the temperature was based on a spatially dependent model. However, in general, the underlying process that we want to model will be time dependent. Within the significant toolbox of mathematical tools that can be used for statistical estimation from noisy sensor measurements, one of the most well-known and often-used tools is the Kalman filter (KF). The KF is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the *time-dependent* discrete-data linear filtering problem [Kalman, 1960].

We consider a dynamical system that evolves in time, and we seek to estimate a series of true states, $\mathbf{x}_k^{\mathrm{t}}$ (a sequence of random vectors), where discrete time is indexed by the letter k. These times are those when the observations or measurements are taken—see Figure 1.6. The assimilation starts with an unconstrained model trajectory from $t_0, t_1, \ldots, t_{k-1}, t_k, \ldots, t_n$ and aims to provide an optimal fit to the available observations/measurements given their uncertainties (error bars), depicted in the figure.

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

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

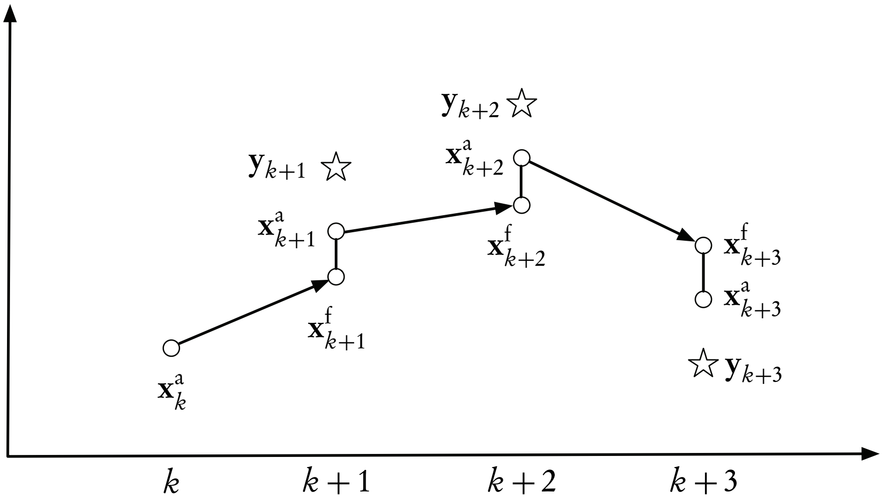

$\mathbf{y}_{k+2}$
$\mathbf{x}^{\mathrm{a}}_{k+2}$
$\mathbf{y}_{k+1}$
$\mathbf{x}^{\mathrm{a}}_{k+1}$
$\mathbf{x}^{\mathrm{f}}_{k+2}$
$\mathbf{x}^{\mathrm{f}}_{k+3}$
$\mathbf{x}^{\mathrm{a}}_{k+3}$
$\mathbf{x}^{\mathrm{f}}_{k+1}$
$\mathbf{y}_{k+3}$
$\mathbf{x}^{\mathrm{a}}_{k}$
k
$k+1$
$k+2$
$k+3$

Figure 1.7. *Sequential assimilation scheme for the KF. The x-axis denotes time; the y-axis is the assimilated variable. We assume scalar variables.*

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

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

The random vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process/modeling and measurement/observation errors, respectively. They are assumed[10] to be independent, white, and with Gaussian/normal probability distributions,

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k), \tag{1.3}$$

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k), \tag{1.4}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices (assumed known) and can in general be time dependent.

We can now set up a sequential DA scheme. The typical assimilation scheme is made up of two major steps: a *prediction/forecast* step and a *correction/analysis* step. At time t_k we have the result of a previous forecast, $\mathbf{x}^{\mathrm{f}}_k$ (the analogue of the background state $\mathbf{x}^{\mathrm{b}}_k$), and the result of an ensemble of observations in $\mathbf{y}_k$. Based on these two vectors, we perform an analysis that produces $\mathbf{x}^{\mathrm{a}}_k$. We then use the evolution model, which is usually (partial) differential equation–based, to obtain a prediction of the state at time t_{k+1}. The result of the forecast is denoted $\mathbf{x}^{\mathrm{f}}_{k+1}$ and becomes the background (or initial guess) for the next time step. This process is summarized in Figure 1.7.

We can now define forecast (a priori) and analysis (a posteriori) estimate errors in the same way as above for the scalar case, with their respective error covariance matrices, which generalize the variances used before, since we are now dealing with vector quantities. The goal of the KF is to compute an optimal a posteriori estimate, $\mathbf{x}^{\mathrm{a}}_k$, that is a linear combination of an a priori estimate, $\mathbf{x}^{\mathrm{f}}_k$, and a weighted difference between the actual measurement, $\mathbf{y}_k$, and the measurement prediction, $\mathbf{H}_k\left[\mathbf{x}^{\mathrm{f}}_k\right]$. This

[10] These assumptions are necessary in the KF framework. For real problems, they must often be relaxed.

is none other than the BLUE that we saw in the example above. The filter must be of the form

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k^{\mathrm{f}} \right), \tag{1.5}$$

where $\mathbf{K}$ is the Kalman gain. The difference $\left(\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k^{\mathrm{f}} \right)$ is called the *innovation* and reflects the discrepancy between the actual and the predicted measurements at time t_k. Note that for generality, the matrices are shown with a time dependence. Often this is not the case, and the subscripts k can then be dropped. The *Kalman gain* matrix, $\mathbf{K}$, is chosen to minimize the a posteriori error covariance equation. This is straightforward to compute: substitute (1.5) into the definition of the analysis error, then substitute in the error covariance equation, take the derivative of the trace of the result with respect to $\mathbf{K}$, set the result equal to zero, and, finally, solve for the optimal gain $\mathbf{K}$. The resulting optimal gain matrix is

$$\mathbf{K}_k = \mathbf{P}_k^{\mathrm{f}} \mathbf{H}^{\mathrm{T}} \left(\mathbf{H} \mathbf{P}_k^{\mathrm{f}} \mathbf{H}^{\mathrm{T}} + \mathbf{R} \right)^{-1},$$

where $\mathbf{P}_k^{\mathrm{f}}$ is the forecast error covariance matrix. Full details of this computation, as well as numerous examples, are provided in Chapter 3, where we will also generalize the approach to more realistic cases.

1.5.3 ▪ Variational DA

Unlike sequential/statistical assimilation (which emanates from estimation theory), variational assimilation is based on optimal control theory [Kwakernaak and Sivan, 1972; Friedland, 1986; Gelb, 1974; Tröltzsch, 2010], itself derived from the *calculus of variations*. The analyzed state is not defined as the one that maximizes a certain 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*.

1.5.3.1 ▪ Adjoint methods: An introduction

All descent-based optimization methods require the computation of the gradient, ∇J, of a cost function, J. If the dependence of J on the control variables is complex or indirect, this computation can be very difficult. Numerically, we can always manage by computing finite increments, but this would have to be done in all possible perturbation directions. We thus need to find a less expensive way to compute the gradient. This will be provided by the *calculus of variations* and the *adjoint approach*.

A basic example: Let us consider a classical inverse problem known as a parameter identification problem, based on the ODE (of convection-diffusion type)

$$\begin{cases} -b u''(x) + c u'(x) = f(x), & 0 < x < 1, \\ u(0) = 0, \ u(1) = 0, \end{cases} \tag{1.6}$$

where $'$ depicts the derivative with respect to x, f is a given function, and b and c are *unknown (constant) parameters* that we seek to identify using observations of $u(x)$ on the interval $[0,1]$. The mismatch (or least-squares error) cost function is then

$$J(b,c) = \frac{1}{2} \int_0^1 (u(x) - u^{\mathrm{o}}(x))^2 \, \mathrm{d}x,$$

where u^{o} is the observational data. The gradient of J can be calculated by introducing the *tangent linear model* (TLM). Perturbing the cost function by a small perturbation[11] in the direction α, with respect to the two parameters, b and c, gives

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

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. Now we divide by α and pass to the limit $\alpha \to 0$ to obtain the directional derivative (with respect to the parameters, in the direction of the perturbations),

$$\hat{J}[b,c](\delta b, \delta c) = \int_0^1 (u-u^{\mathrm{o}})\,\hat{u}\,\mathrm{d}x, \tag{1.7}$$

where we have defined

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

Then, passing to the limit in equation (1.6), we can define the TLM

$$\begin{cases} -b\hat{u}'' + c\hat{u}' = (\delta b)u'' - (\delta c)u', \\ \hat{u}(0) = 0, \ \hat{u}(1) = 0. \end{cases} \tag{1.8}$$

We would like to reformulate the directional derivative (1.7) to obtain a calculable expression for the gradient. For this we introduce the adjoint variable, p, satisfying *the adjoint model*

$$\begin{cases} -bp'' - cp' = (u-u^{\mathrm{o}}), \\ p(0) = 0, \ p(1) = 0. \end{cases} \tag{1.9}$$

Multiplying the TLM by this new variable, p, and integrating by parts enables us to finally write an explicit expression (see Chapter 2 for the complete derivation) for the gradient based on (1.7),

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

or, separating the two components,

$$\nabla_b J(b,c) = \int_0^1 pu''\,\mathrm{d}x,$$

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

Thus, for the additional cost of solving the adjoint model (1.9), 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 using a suitable descent algorithm. This important example is fully developed in Section 2.3.2, where all the steps are explicitly justified.

Note that this method generalizes to (linear and nonlinear) time-dependent PDEs and to inverse problems where we seek to identify the *initial conditions*. This latter problem is exactly the 4D-Var problem of DA. All of this will be amply described in Chapter 2.

[11] The exact properties of the perturbation will be fully explained in Chapter 2.

Algorithm 1.1 Iterative 3D-Var (in its simplest form).

```
$k = 0$, $x = x_0$
while $||\nabla J|| > \epsilon$ or $j \leq j_{\max}$
  compute $J$
  compute $\nabla J$
  gradient descent and update of $x_{j+1}$
  $j = j + 1$
end
```

1.5.3.2 ▪ 3D-Var

We have seen above that the BLUE requires the computation of an optimal gain matrix. We will show (in Chapters 2 and 3) that the optimal gain takes the form

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

to obtain an analyzed state,

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

that minimizes what is known as the 3D-Var cost function,

$$J(x) = \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right)^{\mathrm{T}} \mathbf{B}^{-1}\left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right) + \frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{y})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y}), \qquad (1.10)$$

where $\mathbf{R}$ and $\mathbf{B}$ (also denoted $\mathbf{P}^{\mathrm{f}}$) are the observation and background error covariance matrices, respectively. But the matrices involved in this calculation are often neither storable in memory nor manipulable because of their very large dimensions. The basic idea of variational methods is to overcome these difficulties by attempting to directly minimize the cost function, J. This minimization can be achieved, for inverse problems in general (and for DA in particular), by a combination of (1) an adjoint approach for the computation of the gradient of the cost function with (2) a descent algorithm in the direction of the gradient. For DA problems where there is no time dependence, the adjoint is not necessary and the approach is named 3D-Var, whereas for time-dependent problems we use the 4D-Var approach.

We recall that $\mathbf{R}$ and $\mathbf{B}$ are the observation and background error covariance matrices, respectively. When the observation operator $\mathbf{H}$ is linear, the gradient of J in (1.10) is given by

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

In the iterative 3D-Var Algorithm 1.1 we use as a stopping criterion the fact that ∇J is small or that the maximum number of iterations, $j_{\max}$, is reached.

1.5.3.3 ▪ A simple example of 3D-Var

We seek two temperatures, x_1 and x_2, in London and Paris. The climatologist gives us an initial guess (based on climate records) of $x^b = (10\ 5)^{\mathrm{T}}$, with background error covariance matrix

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

Algorithm 1.2 4D-Var in its basic form

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

We observe $\mathbf{y}^{\mathrm{o}} = 4$ in Paris, which implies that $\mathbf{H} = (0\ 1)$, with an observation error variance $\mathbf{R} = (0.25)$. We can now write the cost function (1.10) as

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

and its gradient can be easily seen to be

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

The minimum is obtained for $\nabla J(\mathbf{x}) = 0$, which yields

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

This is an optimal estimate of the two temperatures, given the background and observation errors.

1.5.3.4 ▪ 4D-Var

In 4D-Var,[12] the cost function is still expressed in terms of the initial state, $\mathbf{x}_0$, but it will include the model because the observation $\mathbf{y}_i^{\mathrm{o}}$ at time i is compared to $\mathbf{H}_i(\mathbf{x}_i)$, where $\mathbf{x}_i$ is the state at time i initialized by $\mathbf{x}_0$ and the adjoint is not simply the transpose of a matrix but also the "transpose" of the model/operator dynamics. To compute this will require the use of a more general adjoint theory, which is introduced just after the following example and fully explained in Chapter 2.

In step (1) of Algorithm 1.2, 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

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

[12] The 4 refers to the additional time dimension.

In step (2), we use

$$\nabla J(\mathbf{x}) = \mathbf{B}^{-1}\left(\mathbf{x}-\mathbf{x}^{\mathrm{b}}\right) - \\ \left[\mathbf{H}_0^{\mathrm{T}}\mathbf{R}_0^{-1}\mathbf{d}_0 + \mathbf{M}_1^{\mathrm{T}}\left[\mathbf{H}_1^{\mathrm{T}}\mathbf{R}_1^{-1}\mathbf{d}_1 + \mathbf{M}_2^{\mathrm{T}}\left[\mathbf{H}_2^{\mathrm{T}}\mathbf{R}_2^{-1}\mathbf{d}_2 + \cdots + \mathbf{M}_j^{\mathrm{T}}\mathbf{H}_j^{\mathrm{T}}\mathbf{R}_j^{-1}\mathbf{d}_j\right]\right]\right],$$

where we have assumed that **H** and **M** are *linear*.

1.6 ▪ Some practical aspects of DA and inverse problems

In this brief section we point out some important practical considerations. It should now be clear that there are four basic ingredients in any inverse or DA problem:

1. observation or measured data;
2. a forward or direct model of the real-world context;
3. a backward or adjoint model in the variational case and a probabilistic framework in the statistical case; and
4. an optimization cycle.

But where does one start? The traditional approach, often employed in mathematical and numerical modeling, is to begin with some simplified, or at least well-known, situation. Once the above four items have been successfully implemented and tested on this instance, we then proceed to take into account more and more reality in the form of real data, more realistic models, more robust optimization procedures, etc. In other words, we introduce uncertainty, but into a system where we at least control some of the aspects.

1.6.1 ▪ Twin experiments

Twin experiments, or synthetic runs, are a basic and indispensable tool for all inverse problems. To evaluate the performance of a DA system we invariably begin with the following methodology:

1. Fix all parameters and unknowns and define a reference trajectory, obtained from a run of the direct model—call this the "truth."
2. Derive a set of (synthetic) measurements, or background data, from this "true" run.
3. Optionally, perturb these observations to generate a more realistic observed state.
4. Run the DA or inverse problem algorithm, starting from an initial guess (different from the "true" initial state used above), using the synthetic observations.
5. Evaluate the performance, modify the model/algorithm/observations, and cycle back to step 1.

Twin experiments thus provide a well-structured methodological framework. Within this framework we can perform different "stress tests" of our system. We can modify the observation network, increase or decrease (even switch off) the uncertainty, test the robustness of the optimization method, and even modify the model. In fact, these experiments can be performed on the full physical model or on some simpler (or reduced-order) model.

1.6.2 ▪ Toy models and other simplifications

Toy models are, by definition, simplified models that we can play with, yes, but these are of course "serious games." In certain complex physical contexts, of which meteorology is a famous example, we have well-established toy models, often of increasing complexity. These can be substituted for the real model, whose computational complexity is often too large, and provide a cheaper test-bed.

Some well-known examples of toy models are

- Lorenz models—see Lorenz [1963]—which are used as an avatar for weather simulations;
- various harmonic oscillators that are used to simulate dynamic systems; and
- famous examples such as the Ising model in physics, the Lotka–Volterra model in life sciences, and the Schelling model in social sciences;

See Marzuoli [2008] for a more general discussion.

1.7 ▪ To go further: Additional comments and references

- Examples of inverse problems: 11 examples can be found in Keller [1966] and 16 in Kirsch [1996].
- As the reader may have observed, the formulation and solution of DA and inverse problems require a wide range of tools and competencies in functional analysis, probability and statistics, variational calculus, numerical optimization, numerical approximation of (partial) differential equations, and stochastic simulation. This monograph will not provide all of this, so the reader must resort to other sources for the necessary background "tools." A few bibliographic recommendations are
  - DeGroot and Schervisch [2012] for probability and statistics;
  - Courant and Hilbert [1989a] for variational calculus;
  - Nocedal and Wright [2006] for numerical optimization;
  - Kreyszig [1978] and Reed and Simon [1980] for functional analysis;
  - Strikwerda [2004] for finite difference methods;
  - Hughes [1987] and Zienkiewicz and Taylor [2000] for finite element methods;
  - Press et al. [2007] for stochastic simulation;
  - Quarteroni et al. [2007] for basic numerical analysis (integration, solution of ODEs, etc.); and
  - Golub and van Loan [2013] for numerical linear algebra.

Chapter 2

Optimal control and variational data assimilation

2.1 ▪ Introduction

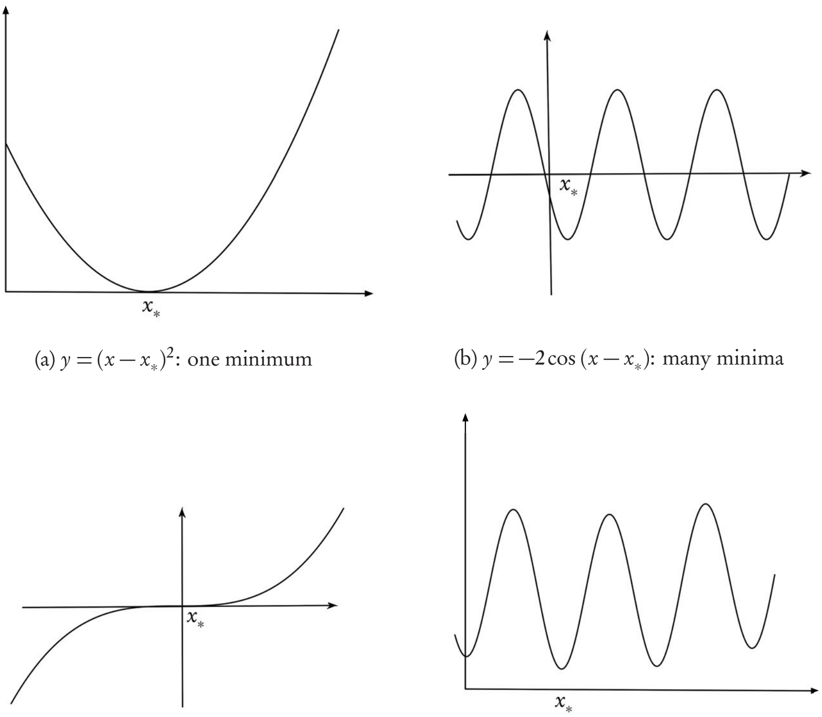
Unlike sequential assimilation (which emanates from statistical estimation theory and will be the subject of the next chapter), variational assimilation is based on optimal control theory.[13] The analyzed state is not defined as the one that maximizes a certain probability density function (PDF), but 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.

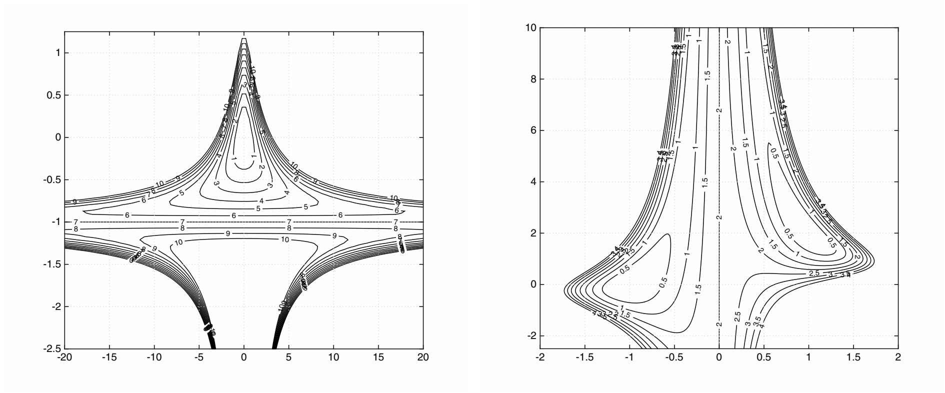
2.2 ▪ The calculus of variations

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

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

[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.

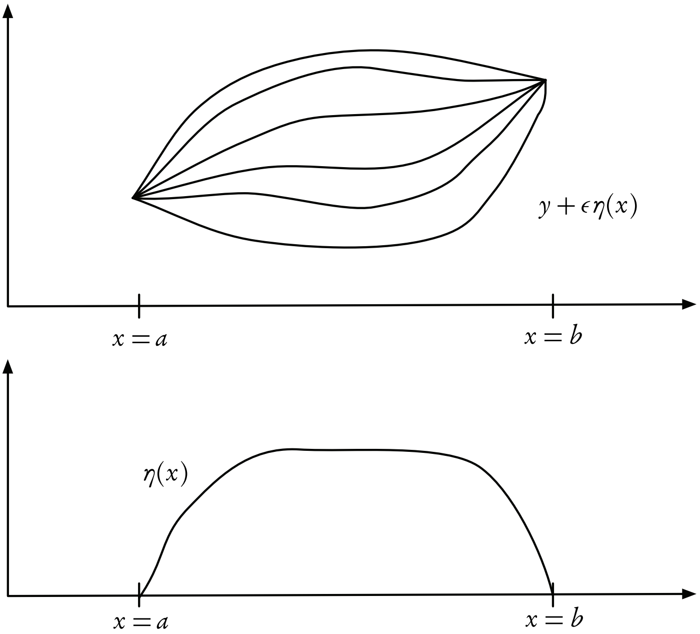


Figure 2.3. *Curve $\eta(x)$ and admissible functions $y+\epsilon\eta(x)$.*

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 $\widetilde{y} = y+\delta y$, with $\delta y = \epsilon\eta(x)$, where ϵ is an arbitrary (small) parameter—see Figure 2.3. This implies that all the functions $\widetilde{y}$ 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

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

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

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

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,

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

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

$$F_{u_xu_x}u_{xx}+2F_{u_xu_y}u_{xy}+F_{u_yu_y}u_{yy}+F_{u_xu}u_x+F_{u_yu}u_y+F_{u_xx}+F_{u_yy}-F_u=0.$$

We can now apply this result to the case where

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

Clearly,

$$F_{u_xu_x}=F_{u_yu_y}=1,$$

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,

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

[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, $\mathbf{u}^{\mathrm{obs}}$, 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, $\mathbf{L}$, or on the model parameters, $\mathbf{m}$.

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 $\mathbf{L}$ with respect to the model parameters, $\mathbf{m}$, 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),

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

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

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

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

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

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

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

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),

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

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:

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

which gives (term by term)

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

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

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

or, grouping terms,

$$\int_0^1 \left(-bp'' - cp'\right)\hat{u} = b\hat{u}'(1)p(1) - b\hat{u}'(0)p(0) + \int_0^1 \left((\delta b)u'' - (\delta c)u'\right)p. \tag{2.11}$$

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

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

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

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

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),

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

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

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

with the cost function

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

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

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

[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:

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

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:

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

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

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

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

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

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

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

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:

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

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

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

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,

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

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

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

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

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

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

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

which immediately yields

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

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 **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, $\mathbf{x}(t) \in \mathcal{X}$, 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,

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

where the initial condition is unknown (or inaccurately known). Suppose that we are in possession of observations $\mathbf{y}(t) \in \mathcal{O}$ and an observation operator $\mathcal{H}$ that describes the available observations. Then, to characterize the difference between the observations and the state, we define the objective (or cost) function

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

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 **R** and **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:

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

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 3*D-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,

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

where

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

Solving the equation, we find

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

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

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

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, $\mathbf{x}$,

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

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

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

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 4D-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),

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

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 $\nabla J$ 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

$$\mathbf{H} = (\begin{array}{ccccccc} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{array}).$$

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.

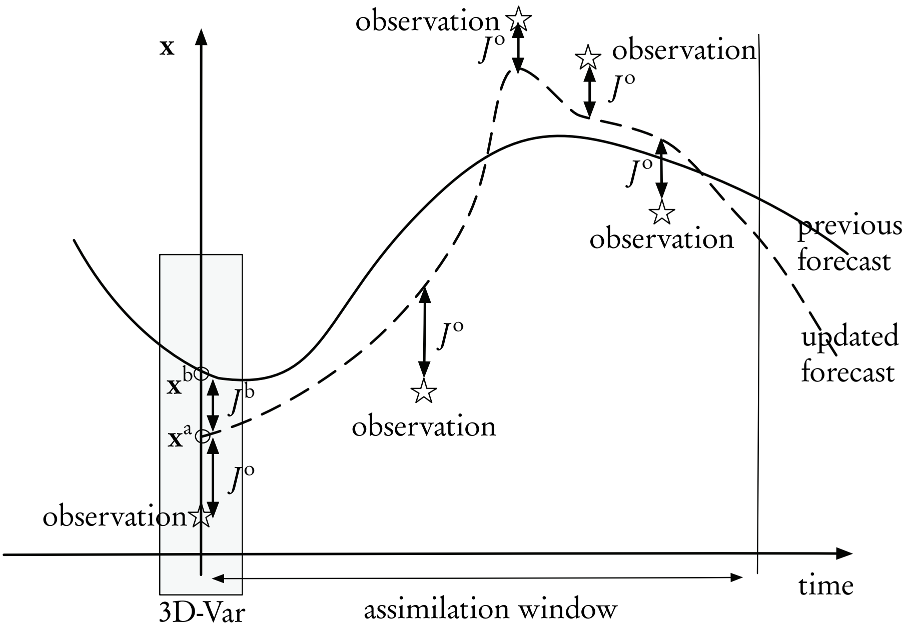


Figure 2.4. *3D- and 4D-Var.*

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 $\mathbf{y}_k^{\mathrm{o}}$ at time k is compared to $\mathbf{H}_k(\mathbf{x}_k)$, where $\mathbf{x}_k$ is the state at time k initialized by $\mathbf{x}$. The cost function is the sum of the background and the observation errors,

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

where the background term is the same as above:

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

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

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

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

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

This gives the final form of the observation term,

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

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 $\nabla J$ 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

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

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

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

and its Hessian is equal to

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

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 **B** and the observation error **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 **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))

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

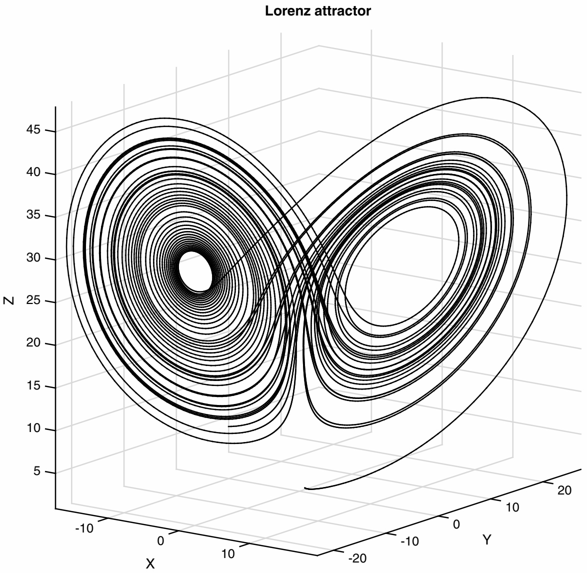


Figure 2.5. *Simulation of the chaotic Lorenz-*63 *system of three equations.*

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`.

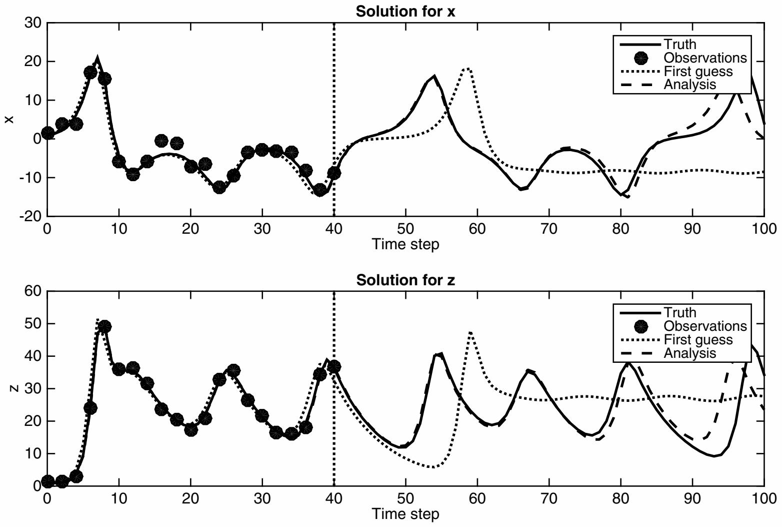


Figure 2.6. *Assimilation of the Lorenz-*63 *equations by standard* 4*D-Var, based on a twin experiment. The assimilation window is from step* 0 *to step* 40 *(*2 *seconds). The forecast window is from step* 41 *to* 100 *(*3 *seconds).*

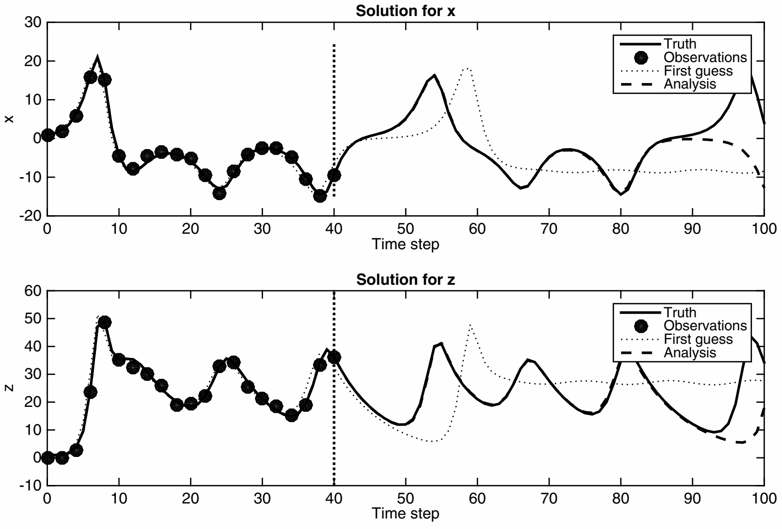


Figure 2.7. *Assimilation of the Lorenz-*63 *equations by incremental* 4*D-Var, based on a twin experiment. The assimilation window is from step* 0 *to step* 40 *(*2 *seconds). The forecast window is from step* 41 *to* 100 *(*3 *seconds).*

Chapter 3

Statistical estimation and sequential data assimilation

The road to wisdom?—Well, it's plain and simple to express:
Err
and err
and err again
but less
and less
and less.
—Piet Hein (1905–1996, Danish mathematician and inventor)

3.1 ▪ Introduction

In this chapter, we present the statistical approach to DA. This approach will be addressed from a Bayesian point of view. But before delving into the mathematical and algorithmic details, we will discuss some ideas about the history of weather forecasting and of the distinction between prediction and forecasting. For a broad, nontechnical treatment of prediction in a sociopolitical-economic context, the curious reader is referred to Silver [2012], where numerous empirical aspects of forecasting are also broached.

3.1.1 ▪ A long history of prediction

From Babylonian times, people have attempted to predict future events, for example in astronomy. Throughout the Renaissance and the Industrial Revolution there were vast debates on predictability.

In 1814, Pierre-Simon Laplace postulated that a perfect knowledge of the actual state of a system coupled with the equations that describe its evolution (natural laws) should provide perfect predictions! This touches on the far-reaching controversy between determinism and randomness/uncertainty … and if we go all the way down to the level of quantum mechanics, then due to Heisenberg's principle there cannot be a perfect prediction. However, weather (and many other physical phenomena) go no further than the molecular (not the atomic) level and as a result molecular chemistry and Newtonian physics are sufficient for weather forecasting. In fact, the (deterministic) PDEs that describe the large-scale circulation of air masses and oceans are remarkably precise and can reproduce an impressive range of meteorological conditions. This is equally true in a large number of other application domains, as described in Part III.

Weather forecasting is a success story: human and machine combining their efforts to understand and to anticipate a complex natural system. This is true for many other systems thanks to the broad applicability of DA and inverse problem methods and algorithms.

3.1.2 ▪ Stochastic versus deterministic

The simplest statistical approach to forecasting (rather like linear regression, but with a flat line) is to calculate the probability of an event (e.g., rain tomorrow) based on past knowledge and records—i.e., long-term averages. But these purely statistical predictions are of little value—they do not take into account the possibility and potential that we have of modeling the physics—this is where the progress (over the last 30 years) in numerical analysis and high-performance computing can come to the rescue. However, this is not a trivial pursuit, as we often notice when surprised by a rain shower, flood, stock market crash, or earthquake. So what goes wrong and impairs the accuracy/reliability of forecasts?

- The first thing that can go wrong is the resolution (spatial and temporal) of our numerical models ... but this is an easier problem: just add more computing power, energy, and money!
- Second, and more important, is *chaos* (see Section 2.5.1), which applies to dynamic, nonlinear systems and is closely associated with the well-posedness issues of Chapter 1—note that this has nothing to do with randomness, but rather is related to the lack of predictability. In fact, in weather modeling, for example, after approximately one week only, chaos theory swamps the dynamic memory of the atmosphere (as "predicted" by the physics), and we are better off relying on climatological forecasts that are based on historical averaged data.
- Finally, there is our imprecise knowledge of the initial (and boundary) conditions for our physical model and hence our simulations—this loops back to the previous point and feeds the chaotic nature of the system. Our measurements are both incomplete and (slightly) inaccurate due to the physical limitations of the instruments themselves.

All of the above needs to be accounted for, as well as possible, in our numerical models and computational analysis. This can best be done with a probabilistic[26] approach.

3.1.3 ▪ Prediction versus forecast

The terms *prediction* and *forecast* are used interchangeably in most disciplines but deserve a more rigorous definition/distinction. Following the philosophy of Silver [2012], a prediction will be considered as a deterministic statement, whereas a forecast will be a probabilistic one. Here are two examples:

- "A major earthquake will strike Tokyo on May 28th" is a prediction, whereas "there is a 60% chance of a major earthquake striking Northern California over the next 25 years" is a forecast.

[26] Equivalently, a statistical or stochastic approach can be used.

- Extrapolation is another example of prediction and is in fact a very basic method that can be useful in some specific contexts but is generally too simplistic and can lead to very bad predictions and decisions.

We notice the UQ in the forecast statement. One way to implement UQ is through Bayesian reasoning—let us explain this now.

3.1.4 ▪ DA is fundamentally Bayesian

Thomas Bayes[27] believed in a rational world of Newtonian mechanics but insisted that by gathering evidence we can get closer and closer to the truth. In other words, rationality is probabilistic. Laplace, as we saw above, claimed that with perfect knowledge of the present and of the laws governing its evolution, we can attain perfect knowledge of the future. In fact it was Laplace who formulated what is known as Bayes' theorem. He considered probability to be "a waypoint between ignorance and knowledge." This is not bad ... it corresponds exactly to our endeavor and what we are trying to accomplish throughout this book: use models and simulations to reproduce and then predict (or, more precisely, forecast or better understand) the actual state and future evolutions of a complex system. For Laplace it was clear: we need a more thorough understanding of probability to make scientific progress!

Bayes' theorem is a very simple algebraic formula based on conditional probability (the probability of one event, A, occurring, knowing or given that another event, B, has occurred—see Section 3.2 below for the mathematical definitions):

$$p_{A|B} = \frac{p_{B|A} p_A}{p_B}.$$

It basically provides us with a reevaluated probability (posterior, $p_{A|B}$) based on the prior knowledge, $p_{B|A} p_A$, of the system that is normalized by the total knowledge that we have, p_B. To better understand and appreciate this result, let us consider a couple of simple examples that illustrate the importance of Bayesian reasoning.

Example 3.1. The famous example of breast cancer diagnosis from mammograms shows the importance and strength of priors. Based on epidemiological studies, the probability that a woman between the ages of 40 and 50 will be afflicted by a cancer of the breast is low, of the order of $p_A = 0.014$ or 1.4%. The question we want to answer is: If a woman in this age range has a positive mammogram (event B), what is the probability that she indeed has a cancer (event A)? Further studies have shown that the false-positive rate of mammograms is $p = 0.1$ or 10% of the time and that the correct diagnosis (true positive) has a rate of $p_{B|A} = 0.75$. So a positive mammogram, taken by itself, would seem to be serious news. However, if we do a Bayesian analysis that factors in the prior information, we get a different picture. Let us do this now. The posterior probability can be computed from Bayes' formula,

$$p_{A|B} = \frac{p_{B|A} p_A}{p_B} = \frac{0.75 \times 0.014}{0.75 \times 0.014 + 0.1 \times (1 - 0.014)} = 0.096,$$

and we conclude that the probability is only 10% in this case, which is far less worrisome than the overall 75% true-positive rate. So the false positives have dominated the result thanks to the fact that we have taken into account the prior information of

[27] English clergyman and statistician (1701–1761).

low cancer incidence in this age range. For this reason, there is a tendency in the medical profession today to recommend that women (without antecedents, which would increase the value of p_A) start having regular mammograms starting from age 50 only because, starting from this age, the prior probability is higher. ■

Example 3.2. Another good example comes from global warming, now called climate change, and we will see why it is so important to quantify uncertainty in the interest of scientific advancement and trust. The study of global warming started around the year 2001. At this time, it was commonly accepted, and scientifically justified, that CO_2 emissions caused and would continue to cause a rise in global temperatures. Thus, we could attribute a high prior probability, $p_A = 0.95$, to the hypothesis of global warming (event A). However, over the subsequent decade from 2001 to 2011, we have observed (event B) that global temperatures have *not* risen as expected—in fact they appeared to have decreased very slightly.[28] So, according to Bayesian reasoning, we should reconsider our estimation of the probability of global warming—the question is, to what extent? If we had a good estimate of the uncertainty in short-term patterns of temperature variations, then the downward revision of the prediction would not be drastic. By analyzing the historical data again, we find that there is a 15% chance that there is no net warming over a decade even if the global warming hypothesis holds—this is due to the inherent variability in the climate. On the other hand, if temperature variations were purely random, and hence unpredictable, then the chance of having a decade in which there is actually a cooling would be 50%. So let us compute the revised estimate for global warming with Bayes' formula. We find

$$p_{A|B} = \frac{p_{B|A}p_A}{p_B} = \frac{0.15 \times 0.95}{0.15 \times 0.95 + 0.5 \times (1 - 0.95)} = 0.851,$$

so we should revise our probability, in light of the last decade's evidence, from 95% to 85%. This is a truly honest approximation that takes into account the observations and revises the uncertainty. Of course, when we receive a new batch of measurements, we can recompute and obtain an update. This is precisely what DA seeks to achieve. The major difference resides in our possession (in the DA context) of a sophisticated model for actually computing the conditional probability, $p_{B|A}$, the probability of the data, or observations, given the parameters. ■

3.1.5 ▪ First steps toward a formal framework

Now let us begin to formalize. It can be claimed that a major part of scientific discovery and research deals with questions of this nature: what can be said about the value of an unknown, or inaccurately known, variable θ that represents the parameters of the system, if we have some measured data $\mathscr{D}$ and a model $\mathscr{M}$ of the underlying mechanism that generated the data? But this is precisely the Bayesian context,[29] where we seek a quantification of the uncertainty in our knowledge of the parameters that, according

[28] A recent paper, published in *Science*, has rectified this by taking into account the evolution of instrumentation since the start of the study. Indeed, it now appears that there has been a steady increase! Apparently, the "hiatus" was the result of a double observational artefact [see T.R. Karl et al., *Science Express*, 4 June 2015].

[29] See Barber [2012], where Bayesian reasoning is extensively developed in the context of machine learning.

to Bayes' theorem takes the form

$$p(\theta \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \theta)p(\theta)}{p(\mathscr{D})} = \frac{p(\mathscr{D} \mid \theta)p(\theta)}{\int_\theta p(\mathscr{D} \mid \theta)p(\theta)}.$$

Here, the physical model is represented by the conditional probability (also known as the *likelihood*) $p(\mathscr{D} \mid \theta)$, and the prior knowledge of the system by the term $p(\theta)$. The denominator is considered as a normalizing factor and represents the total probability of $\mathscr{D}$. From these we can then calculate the resulting posterior probability, $p(\theta \mid \mathscr{D})$.

The most probable estimator, called the maximum a posteriori (MAP) estimator, is the value that maximizes the posterior probability

$$\theta_* = \arg\max_\theta p(\theta \mid \mathscr{D}).$$

Note that for a flat, or uninformative, prior $p(\theta)$, the MAP is just the maximum likelihood, which is the value of θ that maximizes the likelihood $p(\mathscr{D} \mid \theta)$ of the model that generated the data, since in this case neither $p(\theta)$ nor the denominator plays a role in the optimization.

3.1.6 ▪ Concluding remarks (as an opening ...)

There are links between the above and the theories of state space, optimal control, and optimal filtering. We will study KFs, whose original theory was developed in this state space context, below—see Friedland [1986] and Kalman [1960].

The following was the theme of a recent Royal Meteorological Society meeting (Imperial College, London, April 2013): "Should weather and climate prediction models be deterministic or stochastic?"—this is a very important question that is relevant for other physical systems.

In this chapter, we will argue that uncertainty is an inherent characteristic of (weather and most other) predictions and thus that no forecast can claim to be complete without an accompanying estimation of its uncertainty—what we call uncertainty quantification (UQ).

3.2 ▪ Statistical estimation theory

In statistical modeling, the concepts of sample space, probability, and random variable play key roles. Readers who are already familiar with these concepts can skip this section. Those who require more background on probability and statistics should definitely consult a comprehensive treatment, such as DeGroot and Schervisch [2012] or the excellent texts of Feller [1968], Jaynes [2003], McPherson [2001], and Ross [1997].

A sample space, $\mathscr{S}$, is the set of all possible outcomes of a random, unpredictable experiment. Each outcome is a point (or an element) in the sample space. Probability provides a means for quantifying how likely it is for an outcome to take place. Random variables assign numerical values to outcomes in the sample space. Once this has been done, we can systematically work with notions such as average value, or mean, and variability.

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

Definition 3.3. *A* probability space $(\mathscr{S},\mathscr{B},\mathscr{P})$ *consists of a set* $\mathscr{S}$ *called the* sample space, *a collection* $\mathscr{B}$ *of (Borel) subsets of* $\mathscr{S}$, *and a* probability function $\mathscr{P} : \mathscr{B} \to \mathbb{R}_+$ *for which*

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

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

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

The distribution function is nondecreasing and right continuous and satisfies

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

Definition 3.4. *A random variable* X *is called* discrete *if there exist countable sets* $\{x_i\}\subset\mathbb{R}$ *and* $\{p_i\}\subset\mathbb{R}_+$ *for which*

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

for each i, *and*

$$\sum_i p_i=1.$$

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

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

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

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

Definition 3.5. *A random variable* X *is called* continuous *if its distribution function,* F_X*, is absolutely continuous. In this case,*

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

and there exists a derivative of F_X,

$$p_X(x)=\frac{\mathrm{d}F_X}{\mathrm{d}x},$$

that is called the probability density function (PDF) *for* X.

Definition 3.6. *The* mean, *or* expected value, *of an r.v.* X *is given by the integral*

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

This is also known as the first moment of the random variable. If X is a continuous r.v., then

$$\mathrm{d}F_X(x) = p_X(x)\,\mathrm{d}x,$$

and, in the discrete case,

$$\mathrm{d}F_X(x) = p_X(x_i)\delta(x - x_i).$$

In the latter case,

$$\mathrm{E}(X) = \sum_i x_i\, p_X(x_i).$$

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

Definition 3.7. *The* variance *of an r.v. X is given by*

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

where

$$\mu = \mathrm{E}(X).$$

Definition 3.8. *The* mode *is the value of x for which the PDF $p_X(x)$ attains its maximal value.*

Definition 3.9. *Two r.v.'s, X and Y, are* jointly distributed *if they are both defined on the same probability space, $(\mathscr{S}, \mathscr{B}, \mathscr{P})$.*

Definition 3.10. *A* random vector, $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, *is a mapping from $\mathscr{S}$ into $\mathbb{R}^n$ for which all the components X_i are jointly distributed. The* joint distribution function *of $\mathbf{X}$ is given by*

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

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

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

Definition 3.11. *A random vector $\mathbf{X}$ is* continuous *with* joint PDF $p_{\mathbf{X}}$ *if*

$$F_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} p_{\mathbf{X}}(\mathbf{u})\,\mathrm{d}u_1 \ldots \mathrm{d}u_n.$$

Definition 3.12. *The mean, or* expected value, *of a random vector $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ is the n-vector* $\mathrm{E}(\mathbf{X})$ *with components*

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

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

$$[\mathrm{cov}(\mathbf{X})]_{ij} = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \sigma_{ij}^2, \quad 1 \leq i, j \leq n,$$

where

$$\mu_i = \mathrm{E}(X_i).$$

3.2.1 ▪ Gaussian distributions

A continuous random vector $\mathbf{X}$ has a Gaussian distribution if its joint PDF has the form

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

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

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

and the covariance matrix is

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

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

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

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

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

3.2.2 ▪ Estimators and their properties

We now present some fundamental concepts of statistical estimation. A far more complete treatment can be found in Garthwaite et al. [2002], for example. Let us begin by defining estimation and estimators.

We suppose that we are in possession of a random sample $(x_1,x_2,\ldots,x_n)$ (of measurements, say) of the corresponding r.v.'s $X_1,X_2,\ldots,X_n$, whose PDF is $p_{\mathbf{X}}(\mathbf{x};\theta)$. We want to use the observed values $x_1,x_2,\ldots,x_n$ to estimate the parameter θ, which is either unknown or imprecisely known. We then calculate (see methods below) an estimate $\hat{\theta}$ of θ as a function of $(x_1,x_2,\ldots,x_n)$. The corresponding function $\hat{\theta}(X_1,X_2,\ldots,X_n)$, which is an r.v. itself, is an *estimator* for θ. In a given situation, there can exist a number of possible estimators (see example below), and thus the questions of how to choose the best one and what we mean by "best," have to be answered.

The first criterion, considered as indispensable in most circumstances, is that of *unbiasedness*.

Definition 3.13. *The estimator $\hat{\theta}$ for θ is an* unbiased *estimator if the expected value*

$$\mathrm{E}(\hat{\theta})=\theta.$$

The bias of $\hat{\theta}$ is the quantity $\mathrm{E}(\hat{\theta})-\theta$.

The notion of unbiasedness implies that the distribution of $\hat{\theta}$ (recall that $\hat{\theta}$ is an r.v.) is centered exactly at the value θ and that thus there is no tendency to either under- or overestimate this parameter.

Example 3.14. To estimate the mean of a (scalar-valued) normal distribution $\mathcal{N}(\mu,\sigma)$ from a sample of n values, the most evident estimator is the *sample mean*,

$$\hat{X}=\frac{1}{n}\sum_{i=1}^{n}X_i,$$

which is unbiased (this is also the case for other distributions, not only for Gaussians). However, there are numerous other unbiased estimators, for example, the median, the mid-range, and even X_1. ■

We conclude that unbiasedness is usually not enough for choosing an estimator and that we need some other criteria to settle this issue. The classical ones (in addition to unbiasedness) are known as consistency, efficiency, and sufficiency. We will not go into the details here but will concentrate on some optimality conditions.

3.2.3 ▪ Maximum likelihood estimation

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

Definition 3.15. *A maximum likelihood estimator (MLE) for θ given $\mathbf{d}$ is a parameter vector $\hat{\theta}$ that maximizes the likelihood function*

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

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

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

3.2.4 ▪ Bayesian estimation

Now we can formalize the notions of Bayesian probability that were introduced in Sections 3.1.4 and 3.1.5. To this end, we must begin by discussing and defining conditional probability and conditional expectation.

Let $\mathbf{X}=(X_1,X_2,\ldots,X_n)$ and $\mathbf{Y}=(Y_1,Y_2,\ldots,Y_n)$ be jointly distributed discrete random vectors. Then $(\mathbf{X},\mathbf{Y})$ is also a discrete random vector.

Definition 3.16. *The joint PDF for $(\mathbf{X},\mathbf{Y})$ is given by*

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

The marginal PDF *of $\mathbf{X}$ is then defined as*

$$p_{\mathbf{X}}(\mathbf{x})=\sum_{\mathscr{P}\{\mathbf{Y}=\mathbf{y}\}>0} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y}),\quad \mathbf{x}\in\mathbb{R}^n. \tag{3.1}$$

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

$$p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\mid\mathbf{x})=\frac{p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y})}{p_{\mathbf{X}}(\mathbf{x})}, \tag{3.2}$$

where the denominator is nonzero.

So, the conditional probability $p(A|B)$ is the revised probability of an event A after learning that the event B has occurred.

Remark 3.17. *If* $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent random vectors, then the conditional density function of* $\mathbf{Y}$ *given* $\mathbf{X}=\mathbf{x}$ *does not depend on* $\mathbf{x}$ *and satisfies*

$$p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\,|\,\mathbf{x}) = \frac{p_{\mathbf{X}}(\mathbf{x})p_{\mathbf{Y}}(\mathbf{y})}{p_{\mathbf{X}}(\mathbf{x})} = p_{\mathbf{Y}}(\mathbf{y}). \tag{3.3}$$

Definition 3.18. *Let* $\phi : \mathbb{R}^n \to \mathbb{R}^k$ *be a measurable mapping. The* conditional expectation *of* $\phi(\mathbf{Y})$ *given* $\mathbf{X}=\mathbf{x}$ *is*

$$\mathrm{E}(\phi(\mathbf{Y})\,|\,\mathbf{X}=\mathbf{x}) = \sum_{\mathscr{P}\{\mathbf{Y}=\mathbf{y}\}>0} \phi(\mathbf{y})p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\,|\,\mathbf{x}), \quad \mathbf{x}\in\mathbb{R}^n. \tag{3.4}$$

Remark 3.19. *For continuous random vectors* $\mathbf{X}$ *and* $\mathbf{Y}$, *we can define the analogous concepts by replacing the summations in* (3.1)–(3.4) *with appropriate integrals:*

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{(\mathbf{X},\mathbf{Y})}(\mathbf{x},\mathbf{y})\,\mathrm{d}F_{\mathbf{Y}}(\mathbf{y}),$$
$$\mathrm{E}(\phi(\mathbf{Y})\,|\,\mathbf{X}=\mathbf{x}) = \int_{-\infty}^{\infty} \phi(\mathbf{y})p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\,|\,\mathbf{x})\,\mathrm{d}F_{\mathbf{Y}}(\mathbf{y}).$$

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

Theorem 3.20 (Bayes' law). *Let* $\mathbf{X}$ *and* $\mathbf{Y}$ *be jointly distributed random vectors. Then*

$$p_{(\mathbf{X}|\mathbf{Y})}(\mathbf{x}\,|\,\mathbf{y}) = \frac{p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\,|\,\mathbf{x})p_{\mathbf{X}}(\mathbf{x})}{p_{\mathbf{Y}}(\mathbf{y})}. \tag{3.5}$$

Proof. By the definition of conditional probability (3.2),

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

and the numerator is exactly equal to that of (3.5), once again by definition. □

Definition 3.21. *In the context of Bayes' law* (3.5), *suppose that* $\mathbf{X}$ *represents the variable of interest and that* $\mathbf{Y}$ *represents an observable (measured) quantity that depends on* $\mathbf{X}$. *Then,*

- $p_{\mathbf{X}}(\mathbf{x})$ *is called the* a priori *PDF, or the* prior*;*
- $p_{(\mathbf{X}|\mathbf{Y})}(\mathbf{x}\,|\,\mathbf{y})$ *is called the* a posteriori *PDF, or the* posterior*;*
- $p_{(\mathbf{Y}|\mathbf{X})}(\mathbf{y}\,|\,\mathbf{x})$, *considered as a function of* $\mathbf{x}$, *is the* likelihood *function;*
- *the denominator, called the* evidence, $p_{\mathbf{Y}}(\mathbf{y})$, *can be considered as a normalization factor; and*
- *the posterior distribution is thus proportional to the product of the likelihood and the prior distribution or, in applied terms,*

$$p(\text{parameter}\,|\,\text{data}) \propto p(\text{data}\,|\,\text{parameter})p(\text{parameter}).$$

Remark 3.22. *A few fundamental remarks are in order here. First, Bayes' law plays a central role in probabilistic reasoning since it provides us with a method for inverting probabilities, going from* $p(\mathbf{y} \mid \mathbf{x})$ *to* $p(\mathbf{x} \mid \mathbf{y})$. *Second, conditional probability matches perfectly our intuitive notion of uncertainty. Finally, the laws of probability combined with Bayes' law constitute a complete reasoning system for which traditional deductive reasoning is a special case [Jaynes, 2003].*

3.2.5 ▪ Linear least-squares estimation: BLUE, minimum variance linear estimation

In this section we define the two estimators that form the basis of statistical DA. We show that these are optimal, which explains their widespread use.

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_m)$ be two jointly distributed, real-valued random vectors with finite expected squared components:

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

This is an existence condition and is necessary to have a rigorous, functional space setting for what follows.

Definition 3.23. *The* cross-correlation *matrix for* $\mathbf{X}$ *and* $\mathbf{Z}$ *is the* $n \times m$ *matrix* $\Gamma_{\mathbf{XZ}} = \mathrm{E}(\mathbf{X}\mathbf{Z}^T)$ *with entries*

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

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

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

Remark 3.24. *Note that* $\Gamma_{\mathbf{ZX}} = \Gamma_{\mathbf{XZ}}^T$ *and that* $\Gamma_{\mathbf{XX}}$ *is symmetric and positive semidefinite, i.e.,* $\forall \mathbf{x}$, $\mathbf{x}^{\mathrm{T}}\Gamma_{\mathbf{Xx}}\mathbf{x} \geq 0$. *Also, if* $\mathrm{E}(\mathbf{X}) = 0$, *then the autocorrelation reduces to the covariance,* $\Gamma_{\mathbf{XX}} = \mathrm{cov}(\mathbf{X})$.

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

Proposition 3.25. *If a random vector* $\mathbf{X}$ *has finite expected squared components, then*

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

We are now ready to formally define the BLUE. We consider a linear model,

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

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

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

and $\mathbf{C}_{\mathbf{N}}$ is a known, nonsingular, $n \times n$ covariance matrix.

Definition 3.26. *The* best linear unbiased estimator (BLUE) *for* **x** *from the linear model* **z** *is the vector* $\hat{\mathbf{x}}_{\mathrm{BLUE}}$ *that minimizes the quadratic cost function*

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

subject to the constraints of linearity,

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

and unbiasedness,

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

In the case of a full-rank matrix **K**, the *Gauss–Markov theorem* [Sayed, 2003; Vogel, 2002] gives us an explicit form for the BLUE.

Theorem 3.27 (Gauss–Markov). *If* **K** *has full rank, then the BLUE is given by*

$$\hat{\mathbf{x}}_{\mathrm{BLUE}} = \hat{\mathbf{B}}\mathbf{z},$$

where

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

Remark 3.28. *If the noise covariance matrix* $\mathbf{C}_N = \sigma^2 I$ *(white, uncorrelated noise), and* **K** *has full rank, then*

$$\hat{\mathbf{x}}_{\mathrm{BLUE}} = \left(\mathbf{K}^T\mathbf{K}\right)^{-1} \mathbf{K}^T \mathbf{z} = \mathbf{K}^\dagger \mathbf{z},$$

where $\mathbf{K}^\dagger$ *is called the pseudoinverse of* **K**. *This corresponds, in the deterministic case, to the* least-squares problem

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

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

Definition 3.29. *Suppose that* **x** *and* **z** *are jointly distributed, random vectors with finite expected squares. The* minimum variance linear estimator (MVLE) *of* **x** *from* **z** *is given by*

$$\hat{\mathbf{x}}_{\mathrm{MVLE}} = \hat{\mathbf{B}}\mathbf{z},$$

where

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

Proposition 3.30. *If* $\Gamma_{\mathbf{ZZ}}$ *is nonsingular, then the MVLE of* **x** *from* **z** *is given by*

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

3.3 ▪ Examples of Bayesian estimation

In this section we provide some calculated examples of Bayesian estimation.

3.3.1 ▪ Scalar Gaussian distribution example

In this simple, but important, example we will derive in detail the parameters of the posterior distribution when the data and the prior are normally distributed. This will provide us with a richer understanding of DA.

We suppose that we are interested in forecasting the value of a *scalar* state variable, x, which could be a temperature, a wind velocity component, an ozone concentration, etc. We are in possession of a Gaussian prior distribution for x,

$$x \sim \mathcal{N}(\mu_X, \sigma_X^2),$$

with expectation μ and variance σ^2, which could come from a forecast model, for example. We are in possession of n independent, noisy observations,

$$\mathbf{y} = (y_1, y_2, \ldots, y_n),$$

each with conditional distribution

$$y_i \mid x \sim \mathcal{N}(x, \sigma^2),$$

that are conditioned on the true value of the parameter/process x. Thus, the conditional distribution of the data/observations is a product of Gaussian laws,

$$\begin{aligned} p(\mathbf{y} \mid x) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - x)^2\right\} \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - x)^2\right\}. \end{aligned}$$

But from Bayes' law (3.5),

$$p(x \mid \mathbf{y}) \propto p(\mathbf{y} \mid x) p(x),$$

so using the data and the prior distributions/models, we have

$$\begin{aligned} p(x \mid \mathbf{y}) &\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}(y_i - x)^2/\sigma^2 + (x - \mu_X)^2/\sigma_X^2\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[x^2\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_X^2}\right) - 2\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2} + \frac{\mu_X}{\sigma_X^2}\right)x\right]\right\}. \end{aligned}$$

Notice that this is the product of two Gaussians, which, by completing the square, can be show to be Gaussian itself. This produces the posterior distribution,

$$x \mid \mathbf{y} \sim \mathcal{N}\left(\mu_{x|\mathbf{y}}, \sigma_{x|\mathbf{y}}^2\right), \tag{3.6}$$

where

$$\mu_{x|\mathrm{y}} = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_X^2}\right)^{-1} \left(\sum_{i=1}^{n} \frac{y_i}{\sigma^2} + \frac{\mu_X}{\sigma_X^2}\right)$$

and

$$\sigma_{x|\mathrm{y}}^2 = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_X^2}\right)^{-1}.$$

Let us now study more closely these two parameters of the posterior law. We first remark that the inverse of the posterior variance, called the posterior *precision*, is equal to the sum of the prior precision, $1/\sigma_X^2$, and the data precision, n/σ^2. Second, the posterior mean, or conditional expectation, can also be written as a sum of two terms:

$$\begin{aligned} \mathrm{E}(x \mid \mathbf{y}) &= \frac{\sigma^2 \sigma_X^2}{\sigma^2 + n\sigma_X^2} \left(\frac{n}{\sigma^2}\bar{y} + \frac{\mu_X}{\sigma_X^2}\right) \\ &= w_y \bar{y} + w_{\mu_X} \mu_X, \end{aligned}$$

where the sample mean,

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

and the two weights,

$$w_y = \frac{n\sigma_X^2}{\sigma^2 + n\sigma_X^2}, \quad w_{\mu_X} = \frac{\sigma^2}{\sigma^2 + n\sigma_X^2},$$

add up to

$$w_y + w_{\mu_X} = 1.$$

We observe immediately that the posterior mean is the weighted sum/average of the data mean ($\bar{y}$) and the prior mean (μ_X). Now let us examine the weights themselves. If there is a large uncertainty in the prior, then $\sigma_X^2 \to \infty$ and hence $w_y \to 1$, $w_{\mu_X} \to 0$ and the likelihood dominates the prior, leading to what is known as the sampling distribution for the posterior:

$$p(x \mid \mathbf{y}) \to \mathcal{N}(\bar{y}, \sigma^2/n).$$

If we have a large number of observations, then $n \to \infty$ and the posterior now tends to the sample mean, whereas if we have few observations, then $n \to 0$ and the posterior

$$p(x \mid \mathbf{y}) \to \mathcal{N}(\mu_X, \sigma_X^2)$$

tends to the prior. In the case of equal uncertainties between data and prior, $\sigma^2 = \sigma_X^2$, and the prior mean has the weight of a single additional observation. Finally, if the uncertainties are small, either the prior is infinitely more precise than the data ($\sigma_X^2 \to 0$) or the data are perfectly precise ($\sigma^2 \to 0$).

We end this example by rewriting the posterior mean and variance in a special form. Let us start with the mean:

$$\begin{aligned} \mathrm{E}(x \mid \mathbf{y}) &= \mu_X + \frac{n\sigma_X^2}{\sigma^2 + n\sigma_X^2}(\bar{y} - \mu_X) \\ &= \mu_X + G(\bar{y} - \mu_X). \end{aligned} \tag{3.7}$$

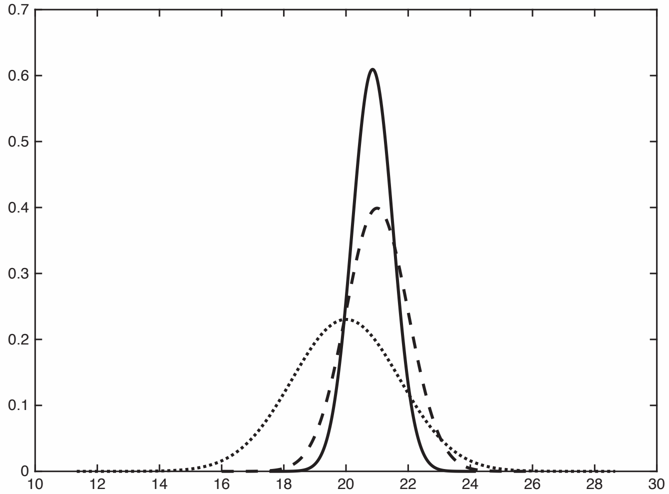


Figure 3.1. *Scalar Gaussian distribution example. Prior* $\mathcal{N}(20, 3)$ *(dotted), instrument* $\mathcal{N}(x, 1)$ *(dashed), and posterior* $\mathcal{N}(20.86, 0.43)$ *(solid) distributions.*

We conclude that the prior mean μ_X is adjusted toward the sample mean $\bar{y}$ by a gain (or amplification factor) of $G = 1/(1+\sigma^2/n\sigma_X^2)$, multiplied by the innovation $\bar{y}-\mu_X$, and we observe that the variance ratio, between data and prior, plays an essential role. In the same way, the posterior variance can be reformulated as

$$\sigma_{x|\mathbf{y}}^2 = (1-G)\sigma_X^2, \tag{3.8}$$

and the posterior variance is thus updated from the prior variance according to the same gain G. These last two equations, (3.7) and (3.8), are fundamental for a good understanding of DA, since they clearly express the interplay between prior and data and the effect that each has on the posterior.

Let us illustrate this with two initial numerical examples. Suppose we have a prior distribution $x \sim \mathcal{N}(\mu_X, \sigma_X^2)$ with mean 20 and variance 3. Suppose that our data model has the conditional law $y_i \mid x \sim \mathcal{N}(x, \sigma^2)$ with variance 1. Here the data are relatively precise compared to the prior. Say we have acquired two observations, $\mathbf{y} = (19, 23)'$. Now we can use (3.7) and (3.8) to compute the posterior distribution:

$$\mathrm{E}(x \mid \mathbf{y}) = 20 + \frac{6}{1+6}(21-20) = 20.86,$$
$$\sigma_{x|\mathbf{y}}^2 = \left(1 - \frac{6}{7}\right)3 = 0.43,$$

thus yielding the posterior distribution

$$y_i \mid x \sim \mathcal{N}(20.86, 0.43),$$

which represents the update of the prior according to the observations and takes into account all the uncertainties available—see Figure 3.1. In other words, we have obtained a complete forecast at a given point in time.

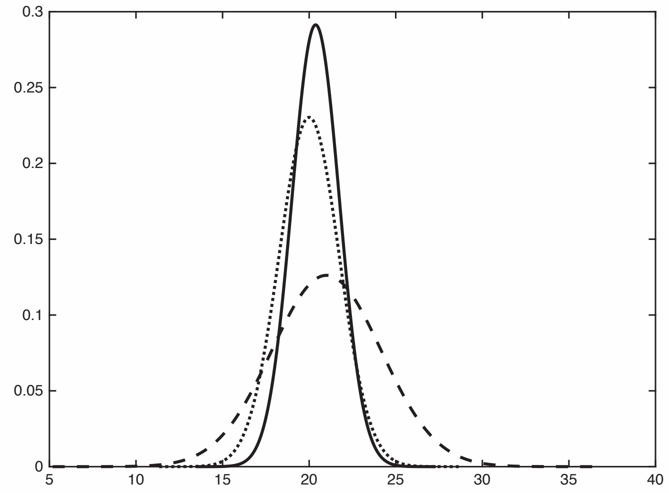


Figure 3.2. *Scalar Gaussian distribution example. Prior* $\mathcal{N}(20, 3)$ *(dotted), instrument* $\mathcal{N}(x, 10)$ *(dashed), and posterior* $\mathcal{N}(20.375, 1.875)$ *(solid) distributions.*

Now consider the same prior, $x \sim \mathcal{N}(20, 3)$, but with a relatively uncertain/imprecise observation model, $y_i \mid x \sim \mathcal{N}(x, 10)$, and the same two measurements, $\mathbf{y} = (19, 23)'$. Redoing the above calculations, we now find

$$\begin{aligned} \mathrm{E}(x \mid \mathbf{y}) &= 20 + \frac{6}{16}(21 - 20) = 20.375, \\ \sigma^2_{x|\mathbf{y}} &= \left(1 - \frac{6}{16}\right)3 = 1.875, \end{aligned}$$

thus yielding the new posterior distribution,

$$y_i \mid x \sim \mathcal{N}(20.375, 1.875),$$

which has virtually the same mean but a much larger variance—see Figure 3.2, where the scales on both axes have changed.

3.3.2 ▪ Estimating a temperature

Suppose that the outside temperature measurement gives 2°C and the instrument has an error distribution that is Gaussian with mean $\mu = 2$ and variance $\sigma^2 = 0.64$—see the dashed curve in Figure 3.3. This is the model/data distribution. We also suppose that we have a prior distribution that estimates the temperature, with mean $\mu = 0$ and variance $\sigma^2 = 1.21$. The prior comes from either other observations, a previous model forecast, or physical and climatological constraints—see the dotted curve in Figure 3.3. By combining these two, using Bayes' formula, we readily compute the posterior distribution of the temperature given the observations, which has mean $\mu = 1.31$ and variance $\sigma^2 = 0.42$. This is the update or the analysis—see the solid curve in Figure 3.3.

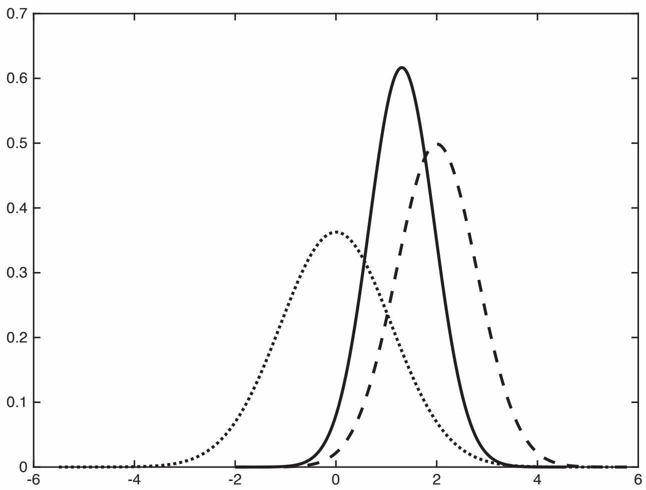


Figure 3.3. *A Gaussian product example for forecasting temperature: prior (dotted), instrument (dashed), and posterior (solid) distributions.*

The code for this calculation can be found in `gaussian_product.m` [DART toolbox, 2013].

3.3.3 ▪ Estimating the parameters of a pendulum

We present an example of a simple mechanical system and seek an estimation of its parameters from noisy measurements. Consider a model for the angular displacement, x_t, of an ideal pendulum (no friction, no drag),

$$x_t = \sin(\theta t) + \epsilon_t,$$

where ϵ_t is a Gaussian noise with zero mean and variance σ^2, the pendulum parameter is denoted by θ, and t is time. From these noisy measurements (suppose that the instrument is not very accurate) of x_t we want to estimate θ, which represents the physical properties of the pendulum—in fact $\theta = \sqrt{g/L}$, where g is the gravitational constant and L is the pendulum's length. Using this physical model, can we estimate (or infer) the unknown physical parameters of the pendulum?

If the measurements are independent, then the likelihood of a set of T observations $x_1, \ldots, x_T$ is given by the product

$$p(x_1, \ldots, x_T|\theta) = \prod_{t=1}^{T} p(x_t|\theta).$$

In addition, suppose that we have some prior estimation (before obtaining the measurements) of the probabilities of a set of possible values of θ. Then the posterior distribution of θ, given the measurements, can be calculated from Bayes' law, as seen above,

$$p(\theta|x_1, \ldots, x_T) \propto p(\theta) \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_t - \sin(\theta t))^2},$$

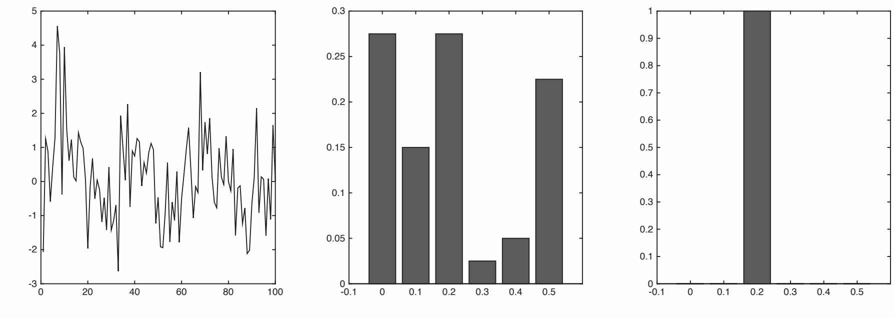


Figure 3.4. *Bayesian estimation of noisy pendulum parameter, $\theta = 0.2$. Observations of* 100 *noisy positions (left). Prior distribution of parameter values (center). Posterior distribution for θ (right).*

where we have omitted the denominator. We are given the following table of priors:

$[\theta_{\min}, \theta_{\max}]$	$p(\theta_{\min} < \theta < \theta_{\max})$
$[0, 0.05]$	0.275
$[0.05, 0.15]$	0.15
$[0.15, 0.25]$	0.275
$[0.25, 0.35]$	0.025
$[0.35, 0.45]$	0.05
$[0.45, 0.55]$	0.225

After performing numerical simulations, we observe (see Figure 3.4) that the posterior for θ develops a prominent peak for a large number ($T = 100$) of measurements, centered around the real value $\theta = 0.2$ (which was used to generate the time series, x_t).

3.3.4 ▪ Vector/multivariate Gaussian distribution example

As a final case that will lead us naturally to the following section, let us consider the vector/multivariate extension of the example in Section 3.3.1. We will now study a vector process, $\mathbf{x}$, with n components and a prior distribution

$$\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{B}),$$

where the mean vector μ and the covariance matrix $\mathbf{B}$ are assumed to be known (as usual from historical data, model forecasts, etc.). The observation now takes the form of a data vector, $\mathbf{y}$, of dimension p and has the conditional distribution/model:

$$\mathbf{y} \mid \mathbf{x} \sim \mathcal{N}(\mathbf{Hx}, \mathbf{R}),$$

where the $(p \times n)$ observation matrix $\mathbf{H}$ maps the process to the measurements and the error covariance matrix $\mathbf{R}$ is known. These are exactly the same matrices that we have already encountered in the variational approach—see Chapters 1 and 2. The difference is that now our modeling is placed in a richer, Bayesian framework.

As before, we would like to calculate the posterior conditional distribution of $\mathbf{x} \mid \mathbf{y}$, given by

$$p(\mathbf{x} \mid \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}).$$

Just as with the scalar/univariate case, the product of two Gaussians is Gaussian, and the posterior law is the multidimensional analogue of (3.6) and can be shown to take the form

$$\mathbf{x} \mid \mathbf{y} \sim \mathcal{N}\left(\mu_{x|\mathrm{y}}, \Sigma_{x|\mathrm{y}}\right),$$

where

$$\mu_{x|\mathrm{y}} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1}\right)^{-1}\left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}y + \mathbf{B}^{-1}\mu\right)$$

and

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

As above, we will now rewrite the posterior mean and variance in a special form. The posterior conditional mean becomes

$$\begin{aligned} \mathrm{E}(\mathbf{x} \mid \mathbf{y}) &= \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}y + \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1}\right)^{-1}\mathbf{B}^{-1}\mu \\ &= \mu + \mathbf{K}(y - \mathbf{H}\mu), \end{aligned} \tag{3.9}$$

where the gain matrix is now

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

In the same manner, the posterior conditional covariance matrix can be reformulated as

$$\begin{aligned} \Sigma(\mathbf{x} \mid \mathbf{y}) &= \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1}\right)^{-1} \\ &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B}, \end{aligned} \tag{3.10}$$

with the same gain matrix **K** as for the posterior mean. As before, these last two equations, (3.9) and (3.10), are fundamental for a good understanding of DA, since they clearly express the interplay between prior and data and the effect that each has on the posterior. They are, in fact, the veritable foundation of DA.

3.3.5 ▪ Connections with variational and sequential approaches

As was already indicated in the first two chapters of this book, the link between variational approaches and optimal BLUE is well established. The BLUE approach is also known as kriging in spatial geostatistics, or optimal interpolation (OI) in oceanography and atmospheric science. In the special, but quite widespread, case of a multivariate Gaussian model (for data and priors), the posterior mode (which is equivalent to the mean in this case) can equally be obtained by minimizing the quadratic objective function (2.31),

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

This is the fundamental link between variational and statistical approaches. Though strictly equivalent to the Bayes formulation, the variational approach has, until now, been privileged for operational, high-dimensional DA problems—though this is changing with the arrival of new hardware and software capabilities for treating "big data and extreme computing" challenges [Reed and Dongarra, 2015].

Since many physical systems are dynamic and evolve in time, we could improve our estimations considerably if, as new measurements became available, we could simply update the previous optimal estimate of the state process without having to redo all computations. The perfect framework for this sequential updating is the KF, which we will now present in detail.

3.4 ▪ Sequential DA and Kalman filters

We have seen that DA, from the statistical/Bayesian point of view, strives to have as complete a knowledge as possible of the a posteriori probability law, that is, the conditional law of the state given the observations. But it is virtually impossible to determine the complete distribution, so we seek instead an estimate of its statistical parameters, such as its mean and/or its variance. Numerous proven statistical methods can lead to best or optimal estimates [Anderson and Moore, 1979; Garthwaite et al., 2002; Hogg et al., 2013; Ross, 2014]; for example, the minimum variance (MV) estimator is the conditional mean of the state given the observations, and the maximum a posteriori (MAP) estimator produces the mode of the conditional distribution. As seen above, assuming Gaussian distributions for the measurements and the process, we can determine the complete a posteriori law, and, in this case, it is clear that the MV and MAP estimators coincide. In fact, the MV estimator produces the optimal interpolation (OI) or kriging equations, whereas the MAP estimator leads to 3D-Var. In conclusion, for the case of a linear observation operator together with Gaussian error statistics, 3D-Var and OI are strictly equivalent.

So far we have been looking at the spatial estimation problem, where all observations are distributed in space but at a single instant in time. For stationary stochastic processes, the mean and covariance are constant in time [Parzen, 1999; Ross, 1997], so such a DA scheme could be used at different times based on the invariant statistics. This is not so rare: in practice, for global NWP, the errors have been considered stationary over a one-month time scale.[30] However, for general environmental applications, the governing equations vary with time and we must take into account nonstationary processes.

Within the significant box of mathematical tools that can be used for statistical estimation from noisy sensor measurements over time, one of the most well known and often used tools is the Kalman filter (KF). The KF is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete data linear filtering problem [Kalman, 1960]. There exits a vast literature on the KF, and a very "friendly" introduction to the general idea of the KF can be found in Chapter 1 of Maybeck [1979]. As just stated above, it would be ideal and very efficient if, as new data or measurements became available, we could easily update the previous optimal estimates without having to recompute everything. The KF provides exactly this solution.

To this end, we will now consider a dynamical system that evolves in time, and we will seek to estimate a series of *true* states, $\mathbf{x}_k^{\mathrm{t}}$ (a sequence of random vectors), where discrete time is indexed by the letter k. These times are those when the observations or measurements are taken, as shown in Figure 3.5. The assimilation starts with an unconstrained model trajectory from $t_0, t_1, \ldots, t_{k-1}, t_k, \ldots, t_n$ and aims to provide an optimal fit to the available observations/measurements given their uncertainties (error bars). For example, in current synoptic scale weather forecasts, $t_k - t_{k-1} = 6$ hours; the time step is less for the convective scale.

3.4.1 ▪ Bayesian modeling

Let us recall the principles of Bayesian modeling from Section 3.2 on statistical estimation and rewrite them in the terminology of the DA problem. We have a vector, $\mathbf{x}$, of (unknown) unobserved quantities of interest (temperature, pressure, wind, etc.) and

[30] This assumption is no longer employed at MétéoFrance or ECWMF, for example.

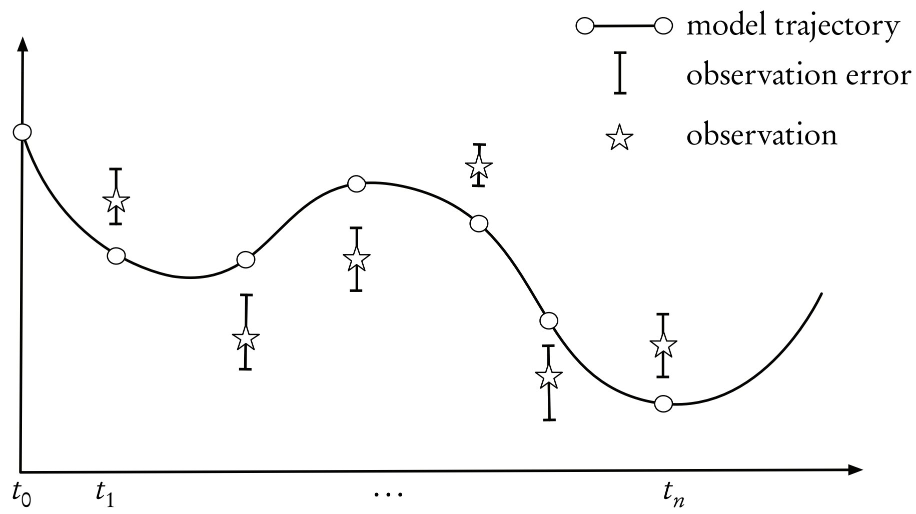


Figure 3.5. *Sequential assimilation: a computed model trajectory, observations, and their error bars.*

a vector, $\mathbf{y}$, of (known) observed data (at various locations, and at various times). The full joint probability model can always be factored into two components,

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

and thus

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

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

The KF can be rigorously derived from this Bayesian perspective following the presentation above in Section 3.3.4.

3.4.2 ▪ Stochastic model of the system

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

$$\mathbf{x}_{k+1} = \mathbf{M}_{k+1}\mathbf{x}_k + \mathbf{w}_k, \tag{3.11}$$

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

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k. \tag{3.12}$$

Note that $\mathbf{M}_{k+1}$ and $\mathbf{H}_k$ are considered linear here. The random vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process/modeling and measurement/observation errors, respectively. They are assumed to be independent, white-noise processes with Gaussian/normal probability distributions

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

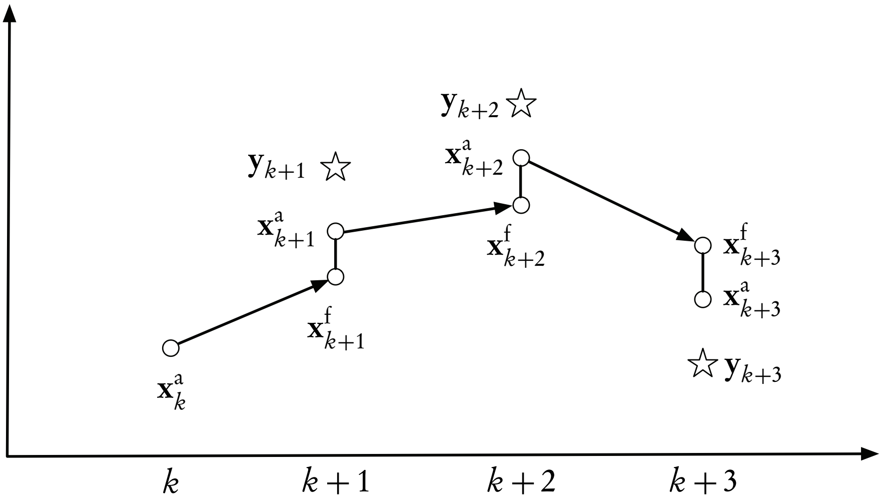

$\mathbf{y}_{k+2}$
$\mathbf{x}^{\mathrm{a}}_{k+2}$
$\mathbf{y}_{k+1}$
$\mathbf{x}^{\mathrm{a}}_{k+1}$
$\mathbf{x}^{\mathrm{f}}_{k+2}$
$\mathbf{x}^{\mathrm{f}}_{k+3}$
$\mathbf{x}^{\mathrm{a}}_{k+3}$
$\mathbf{x}^{\mathrm{f}}_{k+1}$
$\mathbf{y}_{k+3}$
$\mathbf{x}^{\mathrm{a}}_{k}$
k
$k+1$
$k+2$
$k+3$

Figure 3.6. *Sequential assimilation scheme for the KF. The x-axis denotes time; the y-axis denotes the values of the state and observation vectors.*

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices (assumed known) of the modeling and observation errors, respectively. All these assumptions about unbiased and uncorrelated errors (in time and between each other) are not limiting, since extensions of the standard KF can be developed should any of these not be valid—see below and Chapters 5, 6, and 7.

We note that for a broader mathematical view on the above system, we could formulate everything in terms of stochastic differential equations (SDEs). Then the theory of Itô can provide a detailed solution of the problem of optimal filtering as well as existence and uniqueness results—see Oksendal [2003], where one can find such a precise mathematical formulation.

3.4.3 ▪ Sequential assimilation scheme

The typical assimilation scheme is made up of two major steps: a prediction/forecast step and a correction/analysis step. At time t_k we have the result of a previous forecast, $\mathbf{x}^{\mathrm{f}}_k$ (the analogue of the background state $\mathbf{x}^{\mathrm{b}}_k$), and the result of an ensemble of observations in $\mathbf{y}_k$. Based on these two vectors, we perform an analysis that produces $\mathbf{x}^{\mathrm{a}}_k$. We then use the evolution model to obtain a prediction of the state at time t_{k+1}. The result of the forecast is denoted $\mathbf{x}^{\mathrm{f}}_{k+1}$ and becomes the background, or initial guess, for the next time step. This process is summarized in Figure 3.6. The KF problem can be summarized as follows: given a prior/background estimate, $\mathbf{x}^{\mathrm{f}}$, of the system state at time t_k, what is the best update/analysis, $\mathbf{x}^{\mathrm{a}}_k$, based on the currently available measurements, $\mathbf{y}_k$?

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

$$\mathbf{e}^{\mathrm{f}}_k = \mathbf{x}^{\mathrm{f}}_k - \mathbf{x}^{\mathrm{t}}_k,$$
$$\mathbf{e}^{\mathrm{a}}_k = \mathbf{x}^{\mathrm{a}}_k - \mathbf{x}^{\mathrm{t}}_k,$$

where $\mathbf{x}_k^{\mathrm{t}}$ is the (unknown) true state. Their respective error covariance matrices are

$$\begin{aligned}\mathbf{P}_k^{\mathrm{f}} &= \mathrm{cov}(\mathbf{e}_k^{\mathrm{f}}) = E\left[\mathbf{e}_k^{\mathrm{f}}(\mathbf{e}_k^{\mathrm{f}})^{\mathrm{T}}\right],\\ \mathbf{P}_k^{\mathrm{a}} &= \mathrm{cov}(\mathbf{e}_k^{\mathrm{a}}) = E\left[\mathbf{e}_k^{\mathrm{a}}(\mathbf{e}_k^{\mathrm{a}})^{\mathrm{T}}\right].\end{aligned} \tag{3.13}$$

The goal of the KF is to compute an optimal a posteriori estimate, $\mathbf{x}_k^{\mathrm{a}}$, that is a linear combination of an a priori estimate, $\mathbf{x}_k^{\mathrm{f}}$, and a weighted difference between the actual measurement, $\mathbf{y}_k$, and the measurement prediction, $\mathbf{H}_k\mathbf{x}_k^{\mathrm{f}}$. This is none other than the BLUE that we have seen above. The filter is thus of the linear, recursive form

$$\mathbf{x}_k^{\mathrm{a}} = \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^{\mathrm{f}}\right). \tag{3.14}$$

The difference $\mathbf{d}_k = \mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^{\mathrm{f}}$ is called the *innovation* and reflects the discrepancy between the actual and the predicted measurements at time t_k. Note that for generality, the matrices are shown with a time dependence. When this is not the case, the subscripts k can be dropped. The *Kalman gain* matrix, $\mathbf{K}$, is chosen to minimize the a posteriori error covariance equation (3.13).

To compute this *optimal gain* requires a careful derivation. Begin by substituting the observation equation (3.12) into the linear filter equation (3.14):

$$\begin{aligned}\mathbf{x}_k^{\mathrm{a}} &= \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k^{\mathrm{t}} + \mathbf{v}_k - \mathbf{H}_k\mathbf{x}_k^{\mathrm{f}}\right)\\ &= \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k\left(\mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{t}} - \mathbf{x}_k^{\mathrm{f}}\right) + \mathbf{v}_k\right).\end{aligned}$$

Now place this last expression into the definition of $\mathbf{e}_k^{\mathrm{a}}$:

$$\begin{aligned}\mathbf{e}_k^{\mathrm{a}} &= \mathbf{x}_k^{\mathrm{a}} - \mathbf{x}_k^{\mathrm{t}}\\ &= \mathbf{x}_k^{\mathrm{f}} + \mathbf{K}_k\left(\mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{t}} - \mathbf{x}_k^{\mathrm{f}}\right) + \mathbf{v}_k\right) - \mathbf{x}_k^{\mathrm{t}}\\ &= \mathbf{K}_k\left(-\mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right) + \mathbf{v}_k\right) + \left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right).\end{aligned}$$

Then substitute in the error covariance equation (3.13):

$$\begin{aligned}\mathbf{P}_k^{\mathrm{a}} &= \mathrm{E}\left[\mathbf{e}_k^{\mathrm{a}}(\mathbf{e}_k^{\mathrm{a}})^{\mathrm{T}}\right]\\ &= \mathrm{E}\left[\left(\mathbf{K}_k\left(\mathbf{v}_k - \mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right)\right) + \left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right)\right)\left(\mathbf{K}_k\left(\mathbf{v}_k - \mathbf{H}_k\left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right)\right) + \left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right)\right)^{\mathrm{T}}\right].\end{aligned}$$

Now perform the indicated expectations over the r.v.'s, noting that $\left(\mathbf{x}_k^{\mathrm{f}} - \mathbf{x}_k^{\mathrm{t}}\right) = \mathbf{e}_k^{\mathrm{f}}$ is the a priori estimation error, that this error is uncorrelated with the observation error $\mathbf{v}_k$, that by definition $\mathbf{P}_k^{\mathrm{f}} = \mathrm{E}\left[\mathbf{e}_k^{\mathrm{f}}(\mathbf{e}_k^{\mathrm{f}})^{T}\right]$ and that $\mathbf{R}_k = \mathrm{E}\left[\mathbf{v}_k\mathbf{v}_k^{\mathrm{T}}\right]$. We thus get

$$\begin{aligned}\mathbf{P}_k^{\mathrm{a}} &= \mathrm{E}\left[\left(\mathbf{K}_k\left(\mathbf{v}_k - \mathbf{H}_k\mathbf{e}_k^{\mathrm{f}}\right) + \mathbf{e}_k^{\mathrm{f}}\right)\left(\mathbf{K}_k\left(\mathbf{v}_k - \mathbf{H}_k\mathbf{e}_k^{\mathrm{f}}\right) + \mathbf{e}_k^{\mathrm{f}}\right)^{\mathrm{T}}\right]\\ &= \mathrm{E}\left[\left(\mathbf{e}_k^{\mathrm{f}}\right)\left(\mathbf{e}_k^{\mathrm{f}}\right)^{\mathrm{T}} - \left(\mathbf{K}_k\mathbf{H}_k\mathbf{e}_k^{\mathrm{f}}\right)\left(\mathbf{K}_k\mathbf{H}_k\mathbf{e}_k^{\mathrm{f}}\right)^{\mathrm{T}} + (\mathbf{K}_k\mathbf{v}_k)(\mathbf{K}_k\mathbf{v}_k)^{\mathrm{T}}\right]\\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{\mathrm{f}}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}.\end{aligned} \tag{3.15}$$

Note that this is a completely general formula for the updated covariance matrix and that it is valid for any gain $\mathbf{K}_k$, not necessarily optimal.

Now we still need to compute the optimal gain that minimizes the matrix entries along the principal diagonal of $\mathbf{P}_k^{\mathrm{a}}$, since these terms are the ones that represent the estimation error variances for the entries of the state vector itself. We will use the classical approach of variational calculus, by taking the derivative of the trace of the result with respect to $\mathbf{K}$ and then setting the resulting derivative expression equal to zero. But for this, we require two results from matrix differential calculus [Petersen and Pedersen, 2012]. These are

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{A}}\operatorname{Tr}(\mathbf{A}\mathbf{B}) = \mathbf{B}^{\mathrm{T}},$$

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{A}}\operatorname{Tr}\left(\mathbf{A}\mathbf{C}\mathbf{A}^{\mathrm{T}}\right) = 2\mathbf{A}\mathbf{C},$$

where Tr denotes the matrix trace operator and we assume that $\mathbf{AB}$ is square and that $\mathbf{C}$ is a symmetric matrix. The derivative of a scalar quantity with respect to a matrix is defined as the matrix of derivatives of the scalar with respect to each element of the matrix. Before differentiating, we expand (3.15) to obtain

$$\mathbf{P}_k^{\mathrm{a}} = \mathbf{P}_k^{\mathrm{f}} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}} - \mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{K}_k\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)\mathbf{K}_k^{\mathrm{T}}.$$

There are two linear terms and one quadratic term in $\mathbf{K}_k$. To minimize the trace of $\mathbf{P}_k^{\mathrm{a}}$, we can now apply the above matrix differentiation formulas (supposing that the individual squared errors are also minimized when their sum is minimized) to obtain

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{K}_k}\operatorname{Tr}\mathbf{P}_k^{\mathrm{a}} = -2\left(\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}\right)^{\mathrm{T}} + 2\mathbf{K}_k\left(\mathbf{H}_k\mathbf{K}_k\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right).$$

Setting this last result equal to zero, we can finally solve for the optimal gain. The resulting $\mathbf{K}$ that minimizes equation (3.13) is given by

$$\mathbf{K}_k = \mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)^{-1}, \tag{3.16}$$

where we remark that $\mathbf{H}\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k = \mathrm{E}\left[\mathbf{d}_k\mathbf{d}_k^{\mathrm{T}}\right]$ is the covariance of the innovation. Looking at this expression for $\mathbf{K}_k$, we see that when the measurement error covariance, $\mathbf{R}_k$, approaches zero, the gain, $\mathbf{K}_k$, weights the innovation more heavily, since

$$\lim_{\mathbf{R}\to 0}\mathbf{K}_k = \mathbf{H}_k^{-1}.$$

On the other hand, as the a priori error estimate covariance, $\mathbf{P}_k^{\mathrm{f}}$, approaches zero, the gain, $\mathbf{K}_k$, weights the innovation less heavily, and

$$\lim_{\mathbf{P}_k^{\mathrm{f}}\to 0}\mathbf{K}_k = 0.$$

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

The covariance matrix associated with the optimal gain can now be computed from (3.15). We already have

$$\begin{aligned}\mathbf{P}_k^{\mathrm{a}} &= (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{\mathrm{f}}(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}}+\mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}\\ &= \mathbf{P}_k^{\mathrm{f}}-\mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}-\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}+\mathbf{K}_k\left(\mathbf{H}_k\mathbf{K}_k\mathbf{H}_k^{\mathrm{T}}+\mathbf{R}_k\right)\mathbf{K}_k^{\mathrm{T}},\end{aligned}$$

and, substituting the optimal gain (3.16), we can derive three more alternative expressions:

$$\mathbf{P}_k^{\mathrm{a}} = \mathbf{P}_k^{\mathrm{f}}-\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}+\mathbf{R}_k\right)^{-1}\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}},$$

$$\mathbf{P}_k^{\mathrm{a}} = \mathbf{P}_k^{\mathrm{f}}-\mathbf{K}_k\left(\mathbf{H}_k\mathbf{P}_k^{\mathrm{f}}\mathbf{H}_k^{\mathrm{T}}+\mathbf{R}_k\right)\mathbf{K}_k^{\mathrm{T}},$$

and

$$\mathbf{P}_k^{\mathrm{a}} = (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{\mathrm{f}}. \tag{3.17}$$

Each of these four expressions for $\mathbf{P}_k^{\mathrm{a}}$ would give the same results with perfectly precise arithmetic, but in real-world applications some may perform better numerically. In what follows, we will use the simplest form (3.17), but this is by no means restrictive, and any one of the others could be substituted.

3.4.3.1 ▪ Predictor/forecast step

We start from a previous analyzed state, $\mathbf{x}_k^{\mathrm{a}}$, or from the initial state if $k=0$, characterized by the Gaussian PDF $p(\mathbf{x}_k^{\mathrm{a}} \mid \mathbf{y}_{1:k}^{\mathrm{o}})$ of mean $\mathbf{x}_k^{\mathrm{a}}$ and covariance matrix $\mathbf{P}_k^{\mathrm{a}}$. We use here the classical notation $\mathbf{y}_{i:j}=(\mathbf{y}_i,\mathbf{y}_{i+1},\dots,\mathbf{y}_j)$ for $i\le j$ that denotes conditioning on all the observations in the interval. An estimate of $\mathbf{x}_{k+1}^{\mathrm{t}}$ is given by the dynamical model, which defines the forecast as

$$\mathbf{x}_{k+1}^{\mathrm{f}} = \mathbf{M}_{k+1}\mathbf{x}_k^{\mathrm{a}}, \tag{3.18}$$

$$\mathbf{P}_{k+1}^{\mathrm{f}} = \mathbf{M}_{k+1}\mathbf{P}_k^{\mathrm{a}}\mathbf{M}_{k+1}^{T}+\mathbf{Q}_{k+1}, \tag{3.19}$$

where the expression for $\mathbf{P}_{k+1}^{\mathrm{f}}$ is obtained from the dynamics equation and the definition of the model noise covariance, $\mathbf{Q}$.

3.4.3.2 ▪ Corrector/analysis step

At time t_{k+1}, the PDF $p(\mathbf{x}_{k+1}^{\mathrm{f}} \mid \mathbf{y}_{1:k}^{\mathrm{o}})$ is known, thanks to the mean, $\mathbf{x}_{k+1}^{\mathrm{f}}$, and covariance matrix, $\mathbf{P}_{k+1}^{\mathrm{f}}$, just calculated, as well as the assumption of a Gaussian distribution. The analysis step then consists of correcting this PDF using the observation available at time t_{k+1} to compute $p(\mathbf{x}_{k+1}^{\mathrm{a}} \mid \mathbf{y}_{k+1:1}^{\mathrm{o}})$. This comes from the BLUE in the dynamical context and gives

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{\mathrm{f}}\mathbf{H}^{T}\left(\mathbf{H}\mathbf{P}_{k+1}^{\mathrm{f}}\mathbf{H}^{T}+\mathbf{R}_{k+1}\right)^{-1}, \tag{3.20}$$

$$\mathbf{x}_{k+1}^{\mathrm{a}} = \mathbf{x}_{k+1}^{\mathrm{f}}+\mathbf{K}_{k+1}\left(\mathbf{y}_{k+1}-\mathbf{H}\mathbf{x}_{k+1}^{\mathrm{f}}\right), \tag{3.21}$$

$$\mathbf{P}_{k+1}^{\mathrm{a}} = \left(\mathbf{I}-\mathbf{K}_{k+1}\mathbf{H}\right)\mathbf{P}_{k+1}^{\mathrm{f}}. \tag{3.22}$$

The predictor–corrector loop is illustrated in Figure 3.7 and can be immediately transposed into an operational algorithm.

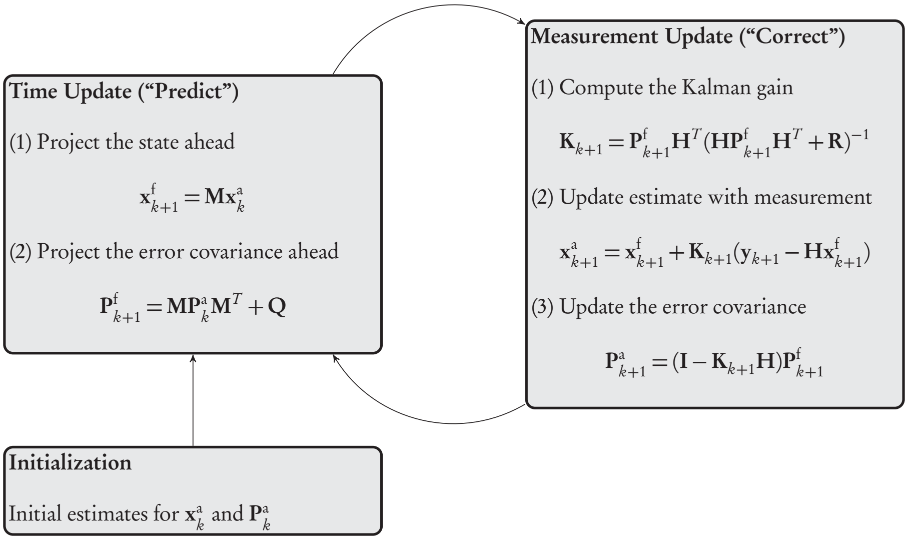

Time Update (“Predict”)
(1) Project the state ahead
$\mathbf{x}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathrm{a}}_{k}$
(2) Project the error covariance ahead
$\mathbf{P}^{\mathrm{f}}_{k+1} = \mathbf{M}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{T} + \mathbf{Q}$
Measurement Update (“Correct”)
(1) Compute the Kalman gain
$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{T}(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{T} + \mathbf{R})^{-1}$
(2) Update estimate with measurement
$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1})$
(3) Update the error covariance
$\mathbf{P}^{\mathrm{a}}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H})\mathbf{P}^{\mathrm{f}}_{k+1}$
Initialization
Initial estimates for $\mathbf{x}^{\mathrm{a}}_{k}$ and $\mathbf{P}^{\mathrm{a}}_{k}$

Figure 3.7. *Kalman filter loop.*

3.4.4 ▪ Note on the relation between Bayes and BLUE

If we know that the a priori and the observation data are both Gaussian, Bayes’ rule can be applied to compute the a posteriori PDF. The a posteriori PDF is then Gaussian, and its parameters are given by the BLUE equations. Hence, with Gaussian PDFs and a linear observation operator, there is no need to use Bayes’ rule. The BLUE equations can be used instead to compute the parameters of the resulting PDF. Since the BLUE provides the same result as Bayes’ rule, it is the best estimator of all.

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

3.5 ▪ Implementation of the Kalman Filter

We now describe some important implementation issues and discuss ways to overcome the difficulties that they give rise to.

3.5.1 ▪ Stability and convergence

Stability is a concern for any dynamic system. The KF will be uniformly asymptotically stable if the system model itself is controllable and observable. The reader is referred to Friedland [1986], Gelb [1974], and Maybeck [1979] for detailed explanations of these concepts.

If the model is linear and time invariant (i.e., system matrices do not vary with time), the autocovariances will converge toward steady-state values. Consequently, the KF gain will converge toward a steady-state KF gain value that can be precalculated by solving an algebraic Ricatti equation. It is quite common to use only the steady-state gain in applications. For a nonlinear system, the gain may vary with the operating point (if the system matrix of the linearized model varies with the operating point).

In practical applications, the gain may be recalculated as the operating point changes.

In practical situations, there are a vast number of different sources for nonconvergence. In Grewal and Andrews [2001], the reader can find a very well explained presentation of all these (and many more). In particular, as we will point out, there are various remedies for

- convergence, divergence, and failure to converge;
- testing for unpredictable behavior;
- effects due to incorrect modeling;
- reduced-order and suboptimal filtering (see Chapter 5);
- reduction of round-off errors and computational expenses;
- analysis and repair of covariance matrices (see next subsection).

3.5.2 ▪ Filter divergence and covariance matrices

If the a priori statistical information is not well specified, the filter might underestimate the variances of the state errors, $\mathbf{e}_k^{\mathrm{a}}$. Too much confidence is put in the state estimation and too little confidence is put in the information contained in the observations. The effect of the analysis is minimized, and the gain becomes too small. In the most extreme case, observations are simply rejected. This is known as *filter divergence*, where the filter seems to behave well, with low predicted analysis error variance, but where the analysis is in fact drifting away from the reality.

Very often filter divergence is easy to diagnose:

- state error variances are small,
- the time sequence of innovations is biased, and
- the Kalman gains tend to zero as time increases.

It is thus important to monitor the innovation sequence and check that it is “white,” i.e., unbiased and normally distributed. If this is not the case, then some of your assumptions are not valid.

There are a few rules to follow to avoid divergence:

- Do not underestimate model errors; rather, overestimate them.
- If possible, it is better to use an adaptive scheme to tune model errors by estimating them on the fly using the innovations.
- Give more weight to recent data, thus reducing the filter’s memory of older data and forcing the data into the KF.
- Place some empirical, relevant lower bound on the Kalman gains.

3.5.3 ▪ Problem size and optimal interpolation

The straightforward application of the KF implies the "propagation" of an $n \times n$ covariance matrix at each time step. This can result in a very large problem in terms of computations and storage. If the state has a spatial dimension of 10^7 (which is not uncommon in large-scale geophysical and other simulations), then the covariance matrices will be of order 10^{14}, which will exceed the resources of most available computer installations. To overcome this, we must resort to various suboptimal schemes (an example of which is detailed below) or switch to ensemble approaches (see Chapters 6 and 7).

If the computational cost of propagating $\mathbf{P}^{\mathrm{a}}_{k+1}$ is an issue, we can use a *frozen covariance matrix*,

$$\mathbf{P}^{\mathrm{a}}_{k} = \mathbf{P}^{\mathrm{b}}, \quad k = 1, \ldots, n.$$

This defines the OI class of methods. Under this simplifying hypothesis, the two-step assimilation cycle defined above becomes the following:

1. **Forecast:**

$$\begin{aligned}\mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{a}}_{k},\\ \mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{P}^{\mathrm{b}}.\end{aligned}$$

2. **Analysis:**

$$\begin{aligned}\mathbf{K}_{k+1} &= \mathbf{P}^{\mathrm{b}}\mathbf{H}^{T}\left(\mathbf{H}\mathbf{P}^{\mathrm{b}}\mathbf{H}^{T} + \mathbf{R}_{k+1}\right)^{-1},\\ \mathbf{x}^{\mathrm{a}}_{k+1} &= \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1} - \mathbf{H}\mathbf{x}^{\mathrm{f}}_{k+1}\right),\\ \mathbf{P}^{\mathrm{a}}_{k+1} &= \mathbf{P}^{\mathrm{b}}.\end{aligned}$$

There are at least two ways to compute the static covariance matrix $\mathbf{P}^{\mathrm{b}}$. The first is an analytical formulation,

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

where $\mathbf{D}$ is a diagonal matrix of variances and $\mathbf{C}$ is a correlation matrix that can be defined, for example, as

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

where a is a tuneable parameter, h is the grid size, and the exponential function provides a local spatial dependence effect that often corresponds well to the physics. The second approach uses an ensemble of N_{e} snapshots of the state vector taken from a model free run, from which we compute the first and second statistical moments as follows:

$$\mathbf{x}^{\mathrm{b}} = \frac{1}{N_{\mathrm{e}}}\sum_{l=1}^{N_{\mathrm{e}}}\mathbf{x}_l,$$

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

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

Other methods are detailed in the sections on reduced methods—see Chapter 5.

3.5.4 ▪ Evolution of the state error covariance matrix

In principle, equation (3.19) generates a symmetric matrix. In practice, this may not be the case, and numerical truncation errors may lead to an asymmetric covariance matrix and a subsequent collapse of the filter. A remedy is to add an extra step to enforce symmetry, such as

$$\mathbf{P}^{\mathrm{f}}_{k+1} = \frac{1}{2}\left(\mathbf{P}^{\mathrm{f}}_{k+1} + (\mathbf{P}^{\mathrm{f}}_{k+1})^{\mathrm{T}}\right),$$

or a square root decomposition—see Chapter 5.

3.6 ▪ Nonlinearities and extensions of the KF

In real-life problems, we are most often confronted with a nonlinear process and/or a nonlinear measurement operator. Our dynamic system now takes the more general form

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathcal{M}_{k+1}(\mathbf{x}_k) + \mathbf{w}_k,\\ \mathbf{y}_k &= \mathcal{H}_k(\mathbf{x}_k) + \mathbf{v}_k,\end{aligned}$$

where $\mathcal{M}_k$ now represents a nonlinear function of the state at time step k and $\mathcal{H}_k$ represents the nonlinear observation operator.

To deal with these nonlinearities, one approach is to linearize about the current mean and covariance, which is called the *extended Kalman filter* (EKF). This approach and its variants are presented in Chapter 6.

As previously mentioned, the KF is only optimal in the case of Gaussian statistics and linear operators, in which case the first two moments (the mean and the covariances) suffice to describe the PDF entering the estimation problem. Practitioners report that the linearized extension to nonlinear problems, the EKF, only works for moderate deviations from linearity and Gaussianity. The ensemble Kalman filter (EnKF) [Evensen, 2009] is a method that has been designed to deal with nonlinearities and non-Gaussian statistics, whereby the PDF is described by an ensemble of N_{e} time-dependent states $\mathbf{x}_{k,e}$. This method is presented in detail in Chapter 6. The appeal of this approach is its conceptual simplicity, the fact that it does not require any TLM or adjoint model (see Chapter 2), and the fact that it is extremely well suited to parallel programming paradigms, such as MPI [Gropp et al., 2014].

What happens if both the models are nonlinear and the PDFs are non-Gaussian? The KF and its extensions are no longer optimal and, more important, can easily fail the estimation process. Another approach must be used. A promising candidate is the *particle filter,* which is described below. The particle filter (see [Doucet and Johansen, 2011] and references therein) works sequentially in the spirit of the KF, but unlike the latter, it handles an ensemble of states (the particles) whose distribution approximates the PDF of the true state. Bayes' rule (3.5) and the marginalization formula (3.1) are explicitly used in the estimation process. The linear and Gaussian hypotheses can then be ruled out, in theory. In practice, though, the particle filter cannot yet be applied to very high dimensional systems (this is often referred to as "the curse of dimensionality").

Finally, there is a new class of hybrid methods, called *ensemble variational methods,* that attempt to combine variational and ensemble approaches—see Chapter 7 for a detailed presentation. The aim is to seek compromises to exploit the best aspects of (4D) variational and ensemble DA algorithms.

For further details of all these extensions, the reader should consult the advanced methods section (Part II) and the above references.

3.7 ▪ Particle filters for geophysical applications

Can we actually design a filtering numerical algorithm that converges to the Bayesian solution? Such a numerical approach would typically belong to the class of *sequential Monte Carlo* methods. That is to say, a PDF is represented by a discrete sample of the targeted PDF. Rather than trying to compute the exact solution of the Bayesian filtering equations, the transformations of such filtering (Bayes' rule for the analysis; model propagation for the forecast) are applied to the members of the sample. The statistical properties of the sample, such as the moments, are meant to be those of the targeted PDF. Obviously this sampling strategy can only be exact in the asymptotic limit, that is, in the limit where the number of members (or particles) goes to infinity.

This is the focus of a large body of applied mathematics that led to the design of many very successful Monte Carlo type methods [see, for instance, Doucet et al., 2001]. However, they have mostly been applied to very low dimensional systems (only a few dimensions). Their efficiency for high-dimensional models has been studied more recently, in particular thanks to a strong interest in these approaches in the geosciences. In the following, we give a brief biased overview of the subject as seen by the geosciences DA community.

3.7.1 ▪ Sequential Monte Carlo

The most popular and simple algorithm of Monte Carlo type that solves the Bayesian filtering equations is called the *bootstrap particle filter* [Gordon et al., 1993]. Its description follows.

3.7.1.1 ▪ Sampling

Let us consider a sample of particles $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$. The related PDF at time t_k is $p_k(\mathbf{x})$, where $p_k(\mathbf{x}) \simeq \sum_{i=1}^{m} \omega_i^k \delta(\mathbf{x}-\mathbf{x}_k^i)$, δ is the Dirac mass, and the sum is meant to be an approximation of the exact density that the sample emulates. A positive scalar, ω_k^i, weights the importance of particle i within the ensemble. At this stage, we assume that the weights, ω_i^k, are uniform and $\omega_i^k = 1/m$.

3.7.1.2 ▪ Forecast

At the forecast step, the particles are propagated by the model without approximation, $p_{k+1}(\mathbf{x}) \simeq \sum_{i=1}^{m} \omega_k^i \delta(\mathbf{x} - \mathbf{x}_{k+1}^i)$, with $\mathbf{x}_{k+1}^i = \mathcal{M}_{k+1}(\mathbf{x}_k)$. A stochastic noise can be optionally added to the dynamics of each particle (see below).

3.7.1.3 ▪ Analysis

The analysis step of the particle filter is extremely simple and elegant. The rigorous implementation of Bayes' rule ascribes to each particle a statistical weight that corresponds to the likelihood of the particle given the data. The weight of each particle is updated according to (see Figure 3.8)

$$\omega_{k+1}^{\mathrm{a},i} \propto \omega_{k+1}^{\mathrm{f},i}\, p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}^i)\,. \tag{3.23}$$

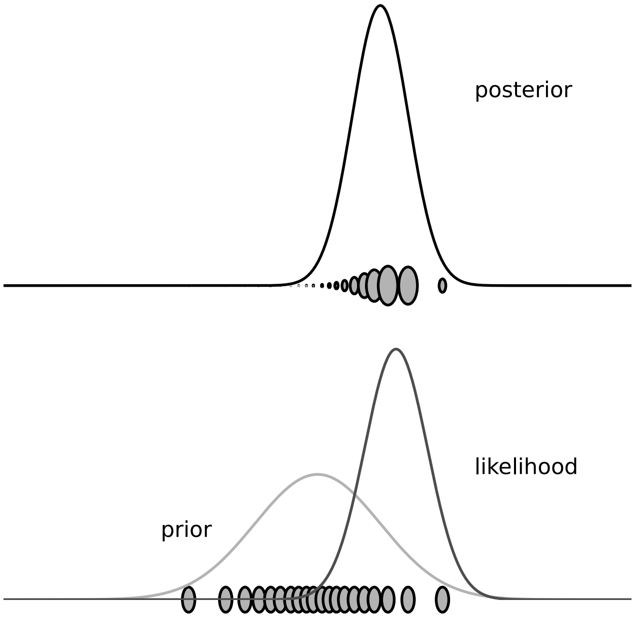


Figure 3.8. *Analysis of the particle filter. The initial ensemble of particles is sampled from a normal prior, with equal weights (bottom). Given an observation with Gaussian noise and the relative state likelihood (bottom), the particle filter analysis ascribes a weight to each particle, which is proportional to the likelihood of the particle given the observation (top). The major axis of the ellipses, representing the particles, is proportional to the particle weight.*

It is remarkable that the analysis is carried out with only a few multiplications. It does not involve inverting any system or matrix, as opposed, for instance, to the KF.

3.7.1.4 ▪ Resampling

Unfortunately, these normalized statistical weights have a potentially large amplitude of fluctuation. Even worse, as sequential filtering progresses, one particle (one trajectory of the model) will stand out from the others. Its weight will largely dominate the others ($\omega_i \lesssim 1$), while the other weights will vanish. Then the particle filter becomes very inefficient as an estimating tool since it has lost its variability. This phenomenon is called *degeneracy* of the particle filter [Kong et al., 1994]. An example of such degeneracy is given in Figure 3.9, where the statistical properties of the biggest weight are studied on an extensive meteorological toy model of 40 and 80 variables. In a degenerate case, the maximum weight will often reach 1 or close to 1, whereas in a balanced case, values very close to 1 will be less frequent.

One way to mitigate this phenomenon is to resample the particles by redrawing a sample with uniform weights from the degenerate distribution. After resampling, all particles have the same weight: $\omega_k^i = 1/m$.

The particle filter is very efficient for highly nonlinear models but with low dimensionality. Unfortunately, it is not suited for DA systems with models of high dimension, as soon as the dimension exceeds, say, about 10. Avoiding degeneracy requires a great number of particles. This number typically increases exponentially with

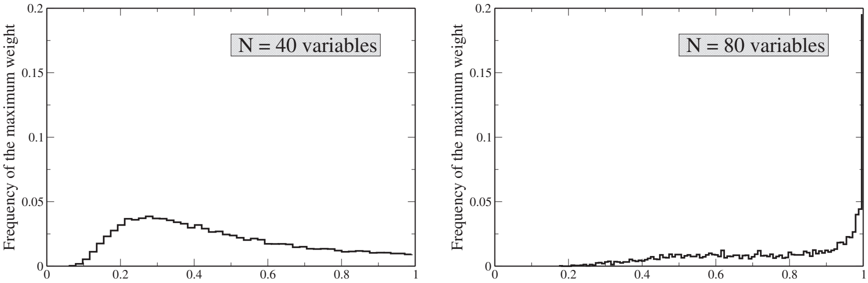


Figure 3.9. *On the left: statistical distribution of the maximal weight of the particle bootstrap filter in a balanced case. The physical system is a Lorenz-*95 *model with* 40 *variables [Lorenz and Emmanuel,* 1998*]. On the right: the same particle filter is applied to a Lorenz-*95 *low-order model, but with* 80 *variables. The maximal weight clearly degenerates with a peak close to* 1.

the system state space dimension. This is because the support of the prior PDF overlaps exponentially less with the support of the likelihood as the dimension of the state space of the systems increases. This is known as the *curse of dimensionality*.

For the forecast step, it could also be crucial to introduce stochastic perturbations of the states. Indeed, the ensemble will become impoverished with the many resamplings that it has to undergo. To enrich the sample, it is necessary to stochastically perturb the states of the system.

3.7.2 ▪ Application in the geosciences

The applicability of particle filters to high-dimensional models has been investigated in the geosciences [van Leeuwen, 2009; Bocquet et al., 2010]. The impact of the curse of dimensionality has been quantitatively studied in Snyder et al. [2008]. It has been shown, on a heuristic basis, that the number of particles m required to efficiently track the system must scale like the variance of the log-likelihood,

$$\ln(m) \propto \mathrm{Var}[\ln(p(\mathbf{y}|\mathbf{x}))], \tag{3.24}$$

which usually scales like the size of the system for typical geophysical problems. It is known [see, for instance, MacKay, 2003] that using an importance proposal to guide the particles toward regions of high probability will not change this trend, albeit with a smaller proportionality factor in (3.24). Snyder et al. [2015] confirmed this and gave bounds to the optimal proposal for particle filters that use an importance proposal leading to a minimal variance in the weights. They conclude again on the exponential dependence of the effective ensemble size with the problem dimension.

When smoothing is combined with a particle filter (which becomes a particle smoother) over a DA window, alternative and more efficient particle filters can be designed, such as the implicit particle filter [Morzfeld et al., 2012].

Particle filters can nevertheless be useful for high-dimensional models if the significant degree of nonlinearity is confined to a small subspace of the state space. For instance, in Lagrangian DA, the errors on the location of moving observation platforms have significantly non-Gaussian statistics. In this case, these degrees of freedom can be addressed with a particle filter, while the rest is controlled by an EnKF, which is practical for high-dimensional models [Slivinski et al., 2015].

If we drop the assumption that a particle filter should have the proper Bayesian asymptotic limit, it becomes possible to design nonlinear filters for DA with high-dimensional models such as the equal-weight particle filter (see [Ades and van Leeuwen, 2015] and references therein).

Finally, if the system cannot be split, then a solution to implement a particle filter in high dimension could come from localization, just as with the EnKF (Chapter 6). This was proven to be more difficult because locally updated particles cannot easily be glued together into global particles. However, an ensemble transform representation that has been built for the EnKF [Bishop et al., 2001] is better suited to ensure a smoother gluing of the local updates [Reich, 2013]. An astute merging of the particles has been shown to yield a local particle filter that could outperform the EnKF in specific regimes with a moderate number of particles [Poterjoy, 2016].

3.8 ▪ Examples

In this section we present a number of examples of special cases of the KF—both analytical and numerical. Though they may seem overly simple, the intention is that you, the user, gain the best possible feeling and intuition regarding the actual operation of the filter. This understanding is essential for more complex cases, such as those presented in the advanced methods and applications chapters.

Example 3.31. *Case without observations.* Here, the observation matrix $\mathbf{H}_k = 0$ and thus $\mathbf{K}_k = 0$ as well. Hence the KF equations (3.18)–(3.22) reduce to

$$\begin{aligned}\mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{a}}_{k},\\ \mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{T}_{k+1} + \mathbf{Q}_{k+1},\end{aligned}$$

and

$$\begin{aligned}\mathbf{K}_{k+1} &= 0,\\ \mathbf{x}^{\mathrm{a}}_{k+1} &= \mathbf{x}^{\mathrm{f}}_{k+1},\\ \mathbf{P}^{\mathrm{a}}_{k+1} &= \mathbf{P}^{\mathrm{f}}_{k+1}.\end{aligned}$$

Thus, we can completely eliminate the analysis stage of the algorithm to obtain

$$\begin{aligned}\mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{f}}_{k},\\ \mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{P}^{\mathrm{f}}_{k}\mathbf{M}^{T}_{k+1} + \mathbf{Q}_{k+1},\end{aligned}$$

initialized by

$$\begin{aligned}\mathbf{x}^{\mathrm{f}}_{0} &= \mathbf{x}_{0},\\ \mathbf{P}^{\mathrm{f}}_{0} &= \mathbf{P}_{0}.\end{aligned}$$

The model then runs without any input of data, and if the dynamics are neutral or unstable, the forecast error will grow without limit. For example, in a typical NWP assimilation cycle, where observations are obtained every 6 hours, the model runs for 6 hours without data. During this period, the forecast error grows and is damped only when the data arrives, thus giving rise to the characteristic “sawtooth” pattern of error variance evolution. ■

Example 3.32. *Perfect observations at all grid points.* In the case of perfect observations, the observation error covariance matrix $\mathbf{R}_k = 0$ and the observation operator $\mathbf{H}$ is the identity. Hence the KF equations (3.18)–(3.22) reduce to

$$\begin{aligned}
\mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{a}}_{k},\\
\mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{P}^{\mathrm{a}}_{k}\mathbf{M}^{T}_{k+1} + \mathbf{Q}_{k+1},
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{K}_{k+1} &= \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{T}\left(\mathbf{H}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{T}\right)^{-1} = \mathbf{I},\\
\mathbf{x}^{\mathrm{a}}_{k+1} &= \mathbf{x}^{\mathrm{f}}_{k+1} + \left(\mathbf{y}_{k+1} - \mathbf{x}^{\mathrm{f}}_{k+1}\right),\\
\mathbf{P}^{\mathrm{a}}_{k+1} &= \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}\right)\mathbf{P}^{\mathrm{f}}_{k+1} = 0.
\end{aligned}$$

This is obviously another case of ideal observations, and we can once again completely eliminate the analysis stage to obtain

$$\begin{aligned}
\mathbf{x}^{\mathrm{f}}_{k+1} &= \mathbf{M}_{k+1}\mathbf{x}^{\mathrm{f}}_{k},\\
\mathbf{P}^{\mathrm{f}}_{k+1} &= \mathbf{Q}_{k+1},
\end{aligned}$$

with initial conditions

$$\begin{aligned}
\mathbf{x}^{\mathrm{f}}_{0} &= \mathbf{y}_{0},\\
\mathbf{P}^{\mathrm{f}}_{0} &= 0.
\end{aligned}$$

Since $\mathbf{R}$ is in fact the sum of measurement and representation errors, $\mathbf{R} = 0$ implies that the only scales that are observed are those resolved by the model. The forecast is thus an integration of the observed state, and the forecast error reduces to the model error. ∎

Example 3.33. *Scalar case.* As in Section 2.4.5, let us consider the same scalar example, but this time apply the KF to it. We take the simplest linear forecast model,

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

with α a known positive constant. We assume the same discrete dynamics considered in (2.49) with a single observation at time step 3.

The stochastic system (3.11)–(3.12) is

$$\begin{aligned}
x^{\mathrm{t}}_{k+1} &= M(x^{\mathrm{t}}_{k}) + w_k,\\
y_{k+1} &= x^{\mathrm{t}}_{k} + v_k,
\end{aligned}$$

where $w_k \sim \mathcal{N}(0,\sigma_Q^2)$, $v_k \sim \mathcal{N}(0,\sigma_R^2)$, and $x^{\mathrm{t}}_0 - x^{\mathrm{b}}_0 \sim \mathcal{N}(0,\sigma_B^2)$. The KF steps are as follows:

Forecast:

$$\begin{aligned}
x^{\mathrm{f}}_{k+1} &= M(x^{\mathrm{a}}_{k}) = \gamma x_k,\\
P^{\mathrm{f}}_{k+1} &= \gamma^2 P^{\mathrm{a}}_{k} + \sigma_Q^2.
\end{aligned}$$

Analysis:

$$\begin{aligned}
K_{k+1} &= P^{\mathrm{f}}_{k+1} H \left(H^2 P^{\mathrm{f}}_{k+1} + \sigma_R^2 \right)^{-1}, \\
x^{\mathrm{a}}_{k+1} &= x^{\mathrm{f}}_{k+1} + K_{k+1}(x^{\mathrm{obs}}_{k+1} - H x^{\mathrm{f}}_{k+1}), \\
P^{\mathrm{a}}_{k+1} &= (1 - K_{k+1} H) P^{\mathrm{f}}_{k+1} = \left(\frac{1}{P^{\mathrm{f}}_{k+1}} + \frac{1}{\sigma_R^2} \right)^{-1}, \quad H = 1.
\end{aligned}$$

Initialization:

$$\begin{aligned}
x^{\mathrm{a}}_0 &= x^{\mathrm{b}}_0, \\
P^{\mathrm{a}}_0 &= \sigma_B^2.
\end{aligned}$$

We start with the initial state, at time step $k = 0$. The initial conditions are as above. The forecast is

$$\begin{aligned}
x^{\mathrm{f}}_1 &= M(x^{\mathrm{a}}_0) = \gamma x^{\mathrm{b}}_0, \\
P^{\mathrm{f}}_1 &= \gamma^2 \sigma_B^2 + \sigma_Q^2.
\end{aligned}$$

Since there is no observation available, $H = 0$, and the analysis gives

$$\begin{aligned}
K_1 &= 0, \\
x^{\mathrm{a}}_1 &= x^{\mathrm{f}}_1 = \gamma x^{\mathrm{b}}_0, \\
P^{\mathrm{a}}_1 &= P^{\mathrm{f}}_1 = \gamma^2 \sigma_B^2 + \sigma_Q^2.
\end{aligned}$$

At the next time step, $k = 1$, and the forecast gives

$$\begin{aligned}
x^{\mathrm{f}}_2 &= M(x^{\mathrm{a}}_1) = \gamma^2 x^{\mathrm{b}}_0, \\
P^{\mathrm{f}}_2 &= \gamma^2 P^{\mathrm{a}}_1 + \sigma_Q^2 = \gamma^4 \sigma_B^2 + (\gamma^2 + 1)\sigma_Q^2.
\end{aligned}$$

Once again there is no observation available, $H = 0$, and the analysis yields

$$\begin{aligned}
K_2 &= 0, \\
x^{\mathrm{a}}_2 &= x^{\mathrm{f}}_2 = \gamma^2 x^{\mathrm{b}}_0, \\
P^{\mathrm{a}}_2 &= P^{\mathrm{f}}_2 = \gamma^4 \sigma_B^2 + (\gamma^2 + 1)\sigma_Q^2.
\end{aligned}$$

Moving on to $k = 2$, we have the new forecast:

$$\begin{aligned}
x^{\mathrm{f}}_3 &= M(x^{\mathrm{a}}_2) = \gamma^3 x^{\mathrm{b}}_0, \\
P^{\mathrm{f}}_3 &= \gamma^2 P^{\mathrm{a}}_2 + \sigma_Q^2 = \gamma^6 \sigma_B^2 + (\gamma^4 + \gamma^2 + 1)\sigma_Q^2.
\end{aligned}$$

Now there is an observation, x^{o}_3, available, so $H = 1$, and the analysis is

$$\begin{aligned}
K_3 &= P^{\mathrm{f}}_3 \left(P^{\mathrm{f}}_3 + \sigma_R^2 \right)^{-1}, \\
x^{\mathrm{a}}_3 &= x^{\mathrm{f}}_3 + K_3(x^{\mathrm{o}}_3 - x^{\mathrm{f}}_3), \\
P^{\mathrm{a}}_3 &= (1 - K_3) P^{\mathrm{f}}_3.
\end{aligned}$$

Substituting and simplifying, we find

$$x_3^{\mathrm{a}} = \gamma^3 x_0^{\mathrm{b}} + \frac{\gamma^6 \sigma_B^2 + (\gamma^4 + \gamma^2 + 1)\sigma_Q^2}{\sigma_R^2 + \gamma^6 \sigma_B^2 + (\gamma^4 + \gamma^2 + 1)\sigma_Q^2} \left(x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}} \right). \tag{3.25}$$

Case 1: Assume we have a perfect model. Then $\sigma_Q^2 = 0$ and the KF state (3.25) becomes

$$x_3^{\mathrm{a}} = \gamma^3 x_0^{\mathrm{b}} + \frac{\gamma^6 \sigma_B^2}{\sigma_R^2 + \gamma^6 \sigma_B^2} \left(x_3^{\mathrm{o}} - \gamma^3 x_0^{\mathrm{b}} \right),$$

which is precisely the 4D-Var expression (2.51) obtained before.

Case 2: When the parameter α tends to zero, then γ tends to one, the model is stationary, and the KF state (3.25) becomes

$$x_3^{\mathrm{a}} = x_0^{\mathrm{b}} + \frac{\sigma_B^2 + 3\sigma_Q^2}{\sigma_R^2 + \sigma_B^2 + 3\sigma_Q^2} \left(x_3^{\mathrm{o}} - x_0^{\mathrm{b}} \right),$$

which, when $\sigma_Q^2 = 0$, reduces to the 3D-Var solution,

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

that was obtained before in (2.52).

Case 3: When α tends to infinity, then γ goes to zero, and we are in the case where there is no longer any memory with

$$x_3^{\mathrm{a}} = \frac{\sigma_Q^2}{\sigma_R^2 + \sigma_Q^2} x_3^{\mathrm{o}}.$$

Then, if the model is perfect, $\sigma_Q^2 = 0$ and $x_3^{\mathrm{a}} = 0$. If the observation is perfect, $\sigma_R^2 = 0$ and $x_3^{\mathrm{a}} = x_3^{\mathrm{o}}$.

This example shows the complete chain, from the KF solution through the 4D-Var and finally reaching the 3D-Var solution. We hope that this clarifies the relationship between the three and demonstrates why the KF provides the most general solution possible. ■

Example 3.34. *Brownian motion.* Here we compute a numerical application of the scalar case seen above in Example 3.33. We have the following state and measurement equations:

$$\begin{aligned} x_{k+1} &= x_k + w_k, \\ y_{k+1} &= x_k + v_k, \end{aligned}$$

where the dynamic transition matrix $M_k = 1$ and the observation operator $H = 1$. Let us suppose constant error variances of $Q_k = 1$ and $R_k = 0.25$ for the process and measurement errors, respectively. Here the KF equations (3.18)–(3.22) reduce to

$$\begin{aligned} x_{k+1}^{\mathrm{f}} &= x_k^{\mathrm{a}}, \\ P_{k+1}^{\mathrm{f}} &= P_k^{\mathrm{a}} + 1, \end{aligned}$$

and

$$\begin{aligned}
K_{k+1} &= P^{\mathrm{f}}_{k+1}\left(P^{\mathrm{f}}_{k+1}+0.25\right)^{-1},\\
x^{\mathrm{a}}_{k+1} &= x^{\mathrm{f}}_{k+1}+K_{k+1}\left(y_{k+1}-x^{\mathrm{f}}_{k+1}\right),\\
P^{\mathrm{a}}_{k+1} &= \left(I-K_{k+1}\right)P^{\mathrm{f}}_{k+1}.
\end{aligned}$$

By substituting for P^{f}_{k+1} from the forecast equation, we can rewrite the Kalman gain in terms of P^{a}_{k} as

$$K_{k+1} = \frac{P^{\mathrm{a}}_{k}+1}{P^{\mathrm{a}}_{k}+1.25},$$

and we obtain the update for the error variance:

$$P^{\mathrm{a}}_{k+1} = \frac{P^{\mathrm{a}}_{k}+1}{4P^{\mathrm{a}}_{k}+5}.$$

Plugging into the analysis equation, we now have the complete update:

$$\begin{aligned}
x^{\mathrm{a}}_{k+1} &= x^{\mathrm{a}}_{k}+\frac{P^{\mathrm{a}}_{k}+1}{P^{\mathrm{a}}_{k}+1.25}\left(y_{k+1}-x^{\mathrm{a}}_{k}\right),\\
P^{\mathrm{a}}_{k+1} &= \frac{P^{\mathrm{a}}_{k}+1}{4P^{\mathrm{a}}_{k}+5}.
\end{aligned}$$

Let us now, manually, perform a couple of iterations. Taking as initial conditions

$$x^{\mathrm{a}}_{0}=0, \quad P^{\mathrm{a}}_{0}=0,$$

we readily compute, for $k=0$,

$$\begin{aligned}
K_1 &= \frac{1}{1.25}=0.8,\\
x^{\mathrm{a}}_{1} &= 0+K_1(y_1-0)=0.8y_1,\\
P^{\mathrm{a}}_{1} &= \frac{1}{5}=0.2.
\end{aligned}$$

Then for $k=1$,

$$\begin{aligned}
K_2 &= \frac{0.2+1}{0.2+1.25}\approx 0.8276,\\
x^{\mathrm{a}}_{2} &= 0.8y_1+K_2(y_2-0.8y_1)\approx 0.138y_1+0.828y_2,\\
P^{\mathrm{a}}_{2} &= \frac{0.2+1}{0.8+5}=\frac{6}{29}\approx 0.207.
\end{aligned}$$

One more step for $k=2$ gives

$$\begin{aligned}
K_3 &= \frac{6/29+1}{6/29+1.25}\approx 0.8284,\\
x^{\mathrm{a}}_{3} &= 0.138y_1+0.828y_2+K_3(y_3-0.138y_1-0.828y_2)\approx 0.024y_1+0.142y_2+0.828y_3,\\
P^{\mathrm{a}}_{3} &= \frac{6/29+1}{24/29+5}\approx 0.207.
\end{aligned}$$

Let us see what happens in the limit, $k \to \infty$. We observe that $P_{k+1} \approx P_k$; thus

$$P^{\mathrm{a}}_{\infty} = \frac{P^{\mathrm{a}}_{\infty} + 1}{4P^{\mathrm{a}}_{\infty} + 5},$$

which is a quadratic equation for P^{a}_{∞}, whose solutions are

$$P^{\mathrm{a}}_{\infty} = \frac{1}{2}\left(-1 \pm \sqrt{2}\right).$$

The positive definite solution is

$$P^{\mathrm{a}}_{\infty} = \frac{1}{2}\left(-1 + \sqrt{2}\right) \approx 0.2071,$$

and hence

$$K_{\infty} = \frac{2 + 2\sqrt{2}}{3 + 2\sqrt{2}} \approx 0.8284.$$

We observe in this case that the KF tends toward a steady-state filter after only two steps. The reasons for this rapid convergence are that the dynamics are neutral and that the observation error covariance is relatively small when compared to the process error, $R \ll Q$, which means that the observations are relatively precise compared to the model error. In addition, the state (being scalar) is completely observed whenever an observation is available.

In conclusion, the combination of dense, precise observations with steady, linear dynamics will always lead to a stable filter. ■

Example 3.35. *Estimation of a random constant.* In this simple numerical example, let us attempt to estimate a scalar random constant, for example, a voltage. Let us assume that we have the ability to take measurements of the constant, but that the measurements are corrupted by a 0.1 volt root mean square (rms) white measurement noise (e.g., our analog-to-digital converter is not very accurate). In this example, our process is governed by the state equation

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

and the measurement equation

$$y_k = H x_k + v_k = x_k + v_k.$$

The state, being constant, does not change from step to step, so $M = I$. Our noisy measurement is of the state directly, so $H = 1$. We are in fact in the same Brownian motion context as the previous example.

The time update (forecast) equations are

$$\begin{aligned} x^{\mathrm{f}}_{k+1} &= x^{\mathrm{a}}_{k}, \\ P^{\mathrm{f}}_{k+1} &= P^{\mathrm{a}}_{k} + Q, \end{aligned}$$

and the measurement update (analysis) equations are

$$\begin{aligned} K_{k+1} &= P^{\mathrm{f}}_{k+1}(P^{\mathrm{f}}_{k+1} + R)^{-1}, \\ x^{\mathrm{a}}_{k+1} &= x^{\mathrm{f}}_{k+1} + K_{k+1}(y_{k+1} - x^{\mathrm{f}}_{k+1}), \\ P^{\mathrm{a}}_{k+1} &= (1 - K_{k+1})P^{\mathrm{f}}_{k+1}. \end{aligned}$$

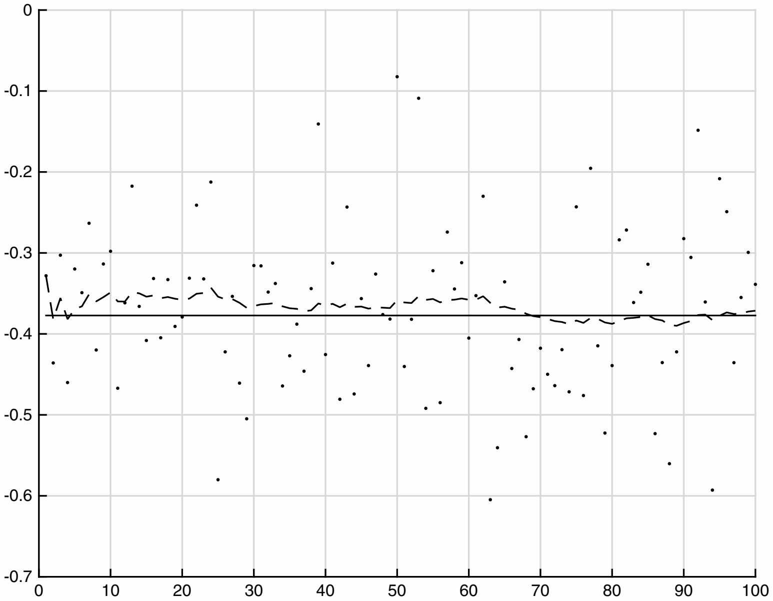


Figure 3.10. *Estimating a constant—simulation with $R = 0.01$. True value (solid), measurements (dots), KF estimation (dashed). The x-axis denotes time; the y-axis denotes the state variable.*

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

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

In the first simulation we fixed the measurement variance at $R = (0.1)^2 = 0.01$. Because this is the "true" measurement error variance, we would expect the "best" performance in terms of balancing responsiveness and estimate variance. This will become more evident in the second and third simulations. Figure 3.10 depicts the results of this first simulation. The true value of the random constant, $x = -0.37727$, is given by the solid line, the noisy measurements by the dots, and the filter estimate by the remaining dashed curve.

In Figures 3.11 and 3.12 we can see what happens when the measurement error variance, R, is increased or decreased by a factor of 100. In Figure 3.11, the filter was told

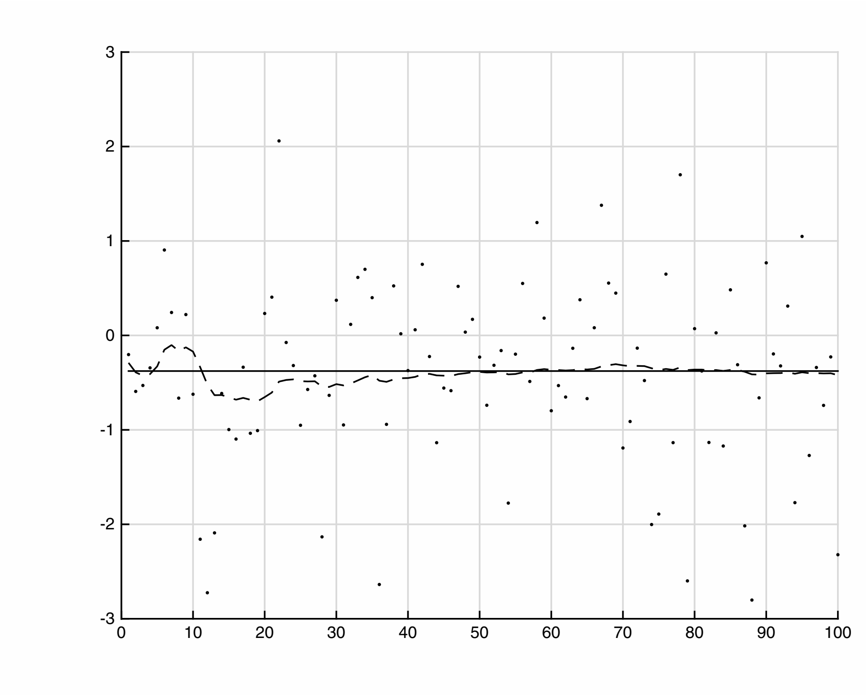


Figure 3.11. *Estimating a constant—simulation with $R = 1$. True value (solid), measurements (dots), KF estimation (dashed). The x-axis denotes time; the y-axis denotes the state variable.*

that the measurement variance was 100 times as great (i.e., $R = 1$), so it was "slower" to believe the measurements. In Figure 3.12, the filter was told that the measurement variance was 1/100th the size (i.e., $R = 0.0001$), so it was very "quick" to believe the noisy measurements.

While the estimation of a constant is relatively straightforward, this example clearly demonstrates the workings of the KF. In Figure 3.11 in particular the Kalman "filtering" is evident, as the estimate appears considerably smoother than the noisy measurements. We observe the speed of convergence of the variance in Figure 3.13.

Here is the MATLAB code used to perform the simulations.

```
% SCALAR EXAMPLE (estimate a constant):
%
% Define the system as a constant of  -0.37727 volts:
clear s
s.x =  -0.37727;
s.A = 1;
% Define a process noise:
s.Q = 0.00001; % variance
% Define the voltmeter to measure the voltage itself:
s.H = 1;
% Define a measurement error:
s.R   = 0.01^2; % variance, hence stdev^2
Rstd  = sqrt(s.R); % random measurement noise stdev
% Specify an initial state:
s.x =  -0.37727;
s.P = 1;
% Generate random voltages and perform the filter operation.
tru=[]; % true voltage
for t=1:100
```

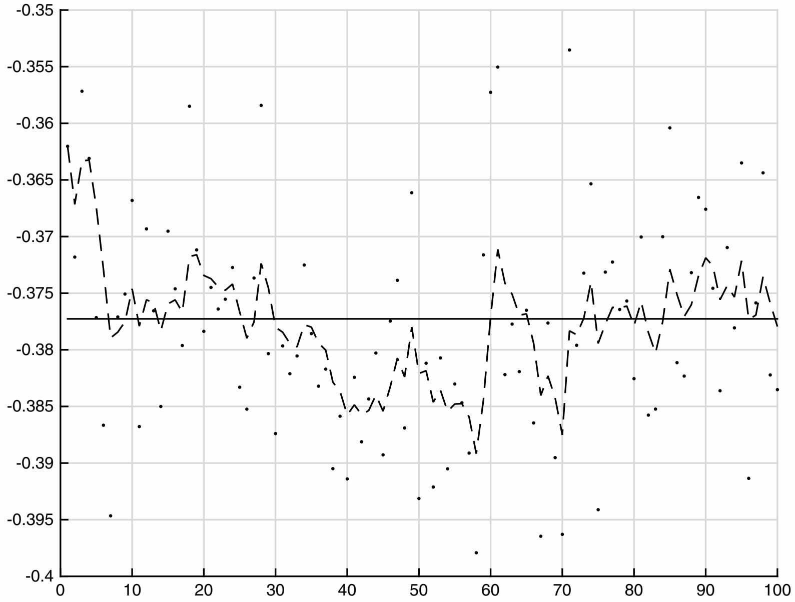


Figure 3.12. *Estimating a constant—simulation with $R = 0.0001$. True value (solid), measurements (dots), KF estimation (dashed). The x-axis denotes time; the y-axis denotes the state variable.*

```
    tru(end+1) =  -0.37727;
    s(end).y = tru(end) + Rstd*randn; % create a measurement
    s(end+1)=kalmanf(s(end)); % perform a Kalman filter iteration
end
% plot measurement data:
figure, hold on, grid on
hy=plot([s(1:end-1).y],'r.');
% plot a-posteriori state estimates:
hk=plot([s(2:end).x],'b--');
% plot true data
ht=plot(tru,'g-');
%legend([hy hk ht],'observations','Kalman output','true voltage',0)
%title('Estimating a constant')
hold off
% KALMANF - updates a system state vector estimate based upon an
%           observation, using a discrete Kalman filter.
%
% Version 1.1, August 13, 2015
%
% This function is based on the original of Michael C. Kleder
%
% INTRODUCTION
%
% Applying the filter to a basic linear system is actually very
% easy.
% This MATLAB file demonstrates that.
%
% An excellent paper on Kalman filtering at the introductory level,
% without detailing the mathematical underpinnings, is
% "An Introduction to the Kalman Filter"
```

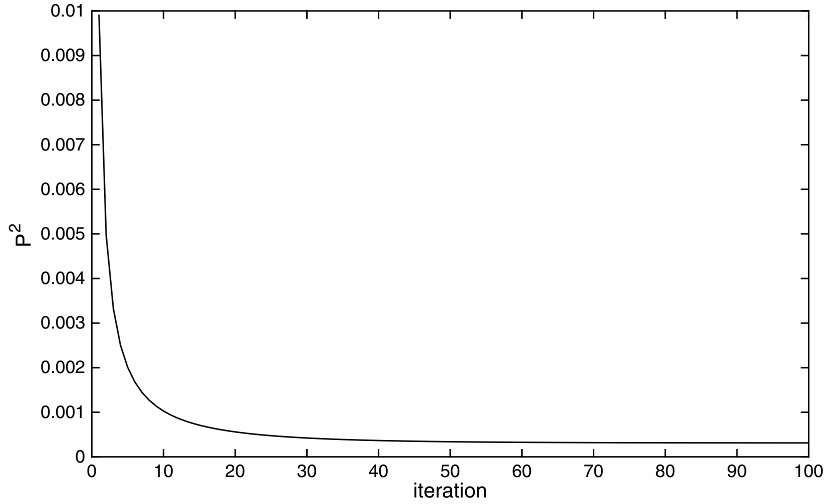


Figure 3.13. *Estimating a constant—convergence of the variance with* $R = 0.01$.

```
% Greg Welch and Gary Bishop, University of North Carolina
% http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html
%
% PURPOSE:
%
% The purpose of each iteration of a Kalman filter is to update
% the estimate of the state vector of a system (and the covariance
% of that vector) based upon the information in a new observation.
% The version of the Kalman filter in this function assumes that
% observations occur at fixed discrete time intervals. Also, this
% function assumes a linear system, meaning that the time evolution
% of the state vector can be calculated by means of a state
% transition matrix.
%
% USAGE:
%
% s = kalmanf(s)
%
% "s" is a "system" struct containing various fields used as input
% and output. The state estimate "x" and its covariance "P" are
% updated by the function. The other fields describe the mechanics
% of the system and are left unchanged. A calling routine may change
% these other fields as needed if state dynamics are time dependent;
% otherwise, they should be left alone after initial values are set.
% The exceptions are the observation vector "y"
%
% SYSTEM DYNAMICS:
%
% The system evolves according to the following difference
% equations, where quantities are further defined below:
%
% x = Ax + w       meaning the state vector x evolves during one
%                  time step by premultiplying by the "state
%                  transition matrix" A. There is also Gaussian
%                  process noise w.
% y = Hx + v       meaning the observation vector y is a linear
%                  function of the state vector, and this linear
%                  relationship is represented by premultiplication
%                  by "observation matrix" H. There is also
%                  Gaussian measurement noise v.
% where w ~ N(0,Q) meaning w is Gaussian noise with covariance Q
%       v ~ N(0,R) meaning v is Gaussian noise with covariance R
%
% VECTOR VARIABLES:
%
% s.x = state vector estimate. In the input struct, this is the
%       "a priori" state estimate (prior to the addition of the
%       information from the new observation). In the output struct,
%       this is the "a posteriori" state estimate (after the new
%       measurement information is included).
% s.y = observation vector
%
% MATRIX VARIABLES:
%
% s.A = state transition matrix (defaults to identity).
% s.P = covariance of the state vector estimate. In the input
%       struct, this is "a priori," and in the output it is
%       "a posteriori" (required unless autoinitializing as
%       described below).
% s.Q = process noise covariance (defaults to zero).
% s.R = measurement noise covariance (required).
% s.H = observation matrix (defaults to identity).
%
% NORMAL OPERATION:
%
% (1) define all state definition fields: A,H,Q,R
% (2) define initial state estimate: x,P
% (3) obtain observation vector: y
% (4) call the filter to obtain updated state estimate: x,P
% (5) return to step (3) and repeat
%
% INITIALIZATION:
%
% If an initial state estimate is unavailable, it can be obtained
% from the first observation as follows, provided that there are
% the same number of observable variables as state variables.
% This "auto-initialization" is done automatically if s.x is
% absent or NaN.
%
% x = inv(H)yz
% P = inv(H)*R*inv(H')
%
% This is mathematically equivalent to setting the initial state
% estimate covariance to infinity.
%

function s = kalmanf(s)

% set defaults for absent fields:
if ~isfield(s,'y'); error('Observation␣vector␣missing'); end
if ~isfield(s,'x'); s.x=nan*y; end
if ~isfield(s,'P'); s.P=nan; end
if ~isfield(s,'A'); s.A=eye(length(x)); end
if ~isfield(s,'Q'); s.Q=zeros(length(x)); end
if ~isfield(s,'R'); error('Observation␣covariance␣missing'); end
if ~isfield(s,'H'); s.H=eye(length(x)); end

if isnan(s.x)
   % initialize state estimate from first observation
```

```
      if diff(size(s.H))
         error('Observation␣matrix␣must␣be␣square␣and␣invertible␣' ...
               'for␣state␣auto-initialization')
      end
      s.x = inv(s.H)*s.y;
      s.P = inv(s.H)*s.R*inv(s.H');
   else

      % This is the code which implements the discrete Kalman filter:

      % Prediction for state vector and covariance:
      s.x = s.A*s.x;
      s.P = s.A * s.P * s.A' + s.Q;

      % Compute Kalman gain factor:
      K = (s.P)*(s.H')*inv(s.H*s.P*s.H'+s.R);

      % Correction based on observation:
      s.x = s.x + K*(s.y-s.H*s.x);
      s.P = s.P - K*s.H*s.P;

      % Note that the desired result, which is an improved estimate
      % of the system state vector x and its covariance P, was obtained
      % in only five lines of code, once the system was defined. (That's
      % how simple the discrete Kalman filter is to use.)
   end
   return
```

■

Example 3.36. *Estimation of a linear trajectory.* We now go one step further and introduce some simple linear dynamics into the problem. We consider a train moving at an unknown, constant velocity, and we would like to predict both its position and its velocity from noisy observations of the position only. Note that this example has many concrete applications in automatic pilots, robotics, and other inertial navigation instruments, where KFs are extensively employed. Let us write down the equations.

The equation of motion for the actual position, x, is

$$x(t) = x_0 + st,$$

where x_0 is the initial position and s is the constant speed of the train. For state space notation, we introduce the state vector,

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix},$$

where x is the position and $\dot{x}$ is the velocity. The state equation can then be written as

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

where we have assumed that the system is perturbed by a white Gaussian noise, $\mathbf{w}$, with known covariance. From the dynamics, we deduce the discrete form of the matrix,

$$\mathbf{M} = \begin{bmatrix} 1 & \mathrm{d}t \\ 0 & 1 \end{bmatrix},$$

where $\mathrm{d}t$ is the time step increment and corresponds to the instants when measurements are taken. The observation is scalar,

$$y_k = \mathbf{H}\mathbf{x}_k + v_k,$$

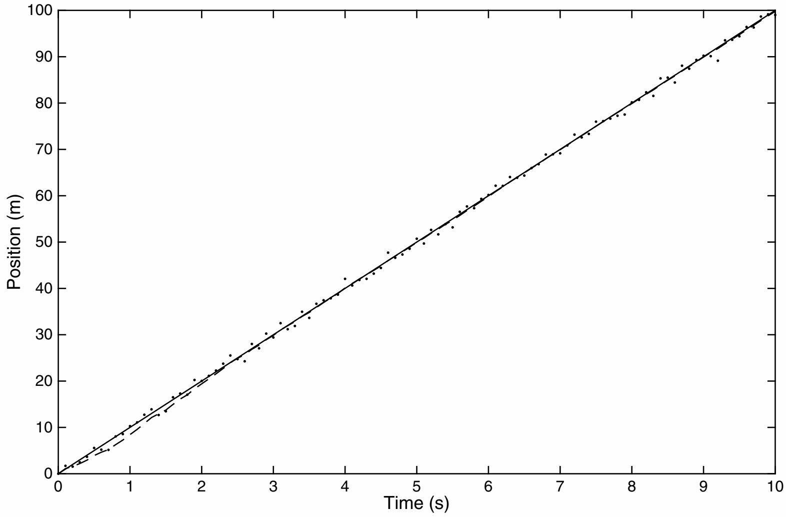


Figure 3.14. *Position estimation for constant-velocity dynamics. True value (solid); measurements (dots); KF estimation (dashed).*

with

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

We initialize our problem with $x_0 = 0$ and $\dot{x}_0 = 0.5\dot{x}^{\mathrm{t}}$, where the (unknown) true velocity $\dot{x}^{\mathrm{t}} = 10\ \mathrm{m\,s}^{-1}$. We assume that measurements are taken every $\mathrm{d}t = 0.1$ s from $t = 0$ until $t = 10$ and that they are subject to a noise with standard deviation of 1 ms^{-1}. We will suppose that the initial error covariance matrix $\mathbf{P}_0 = \mathbf{I}$ and that the process noise is small ($R = 0.0001$). We want to predict the train's position 2 seconds ahead, that is, at $t = 12$ s. We will use the KF, as we need an accurate and smooth estimate for the velocity to predict the train's position in the future.

The numerical computation gives the results and predictions shown in Figures 3.14, 3.15, 3.16, 3.17, and 3.18. The filter does a good job and "steers" a smoother path among the noisy measurements. When compared to an extrapolation, based on a running average for the velocity, the KF clearly outperforms other methods when it comes to the prediction of the position in the future. The convergence of the filter parameters to zero is a sign that the model is adequate and that the data and model are consistent and in good agreement.

The reader is strongly encouraged to modify the code of the previous example to reproduce these results and then to adjust the problem's parameters and observe the effects—as was done in the previous example. ∎

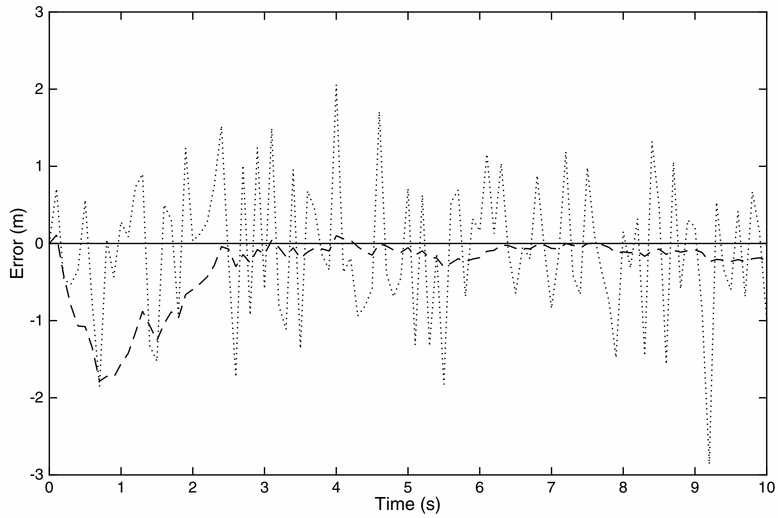


Figure 3.15. *Position estimation errors for constant-velocity dynamics. True value (solid); measurements (dotted); KF estimation (dashed).*

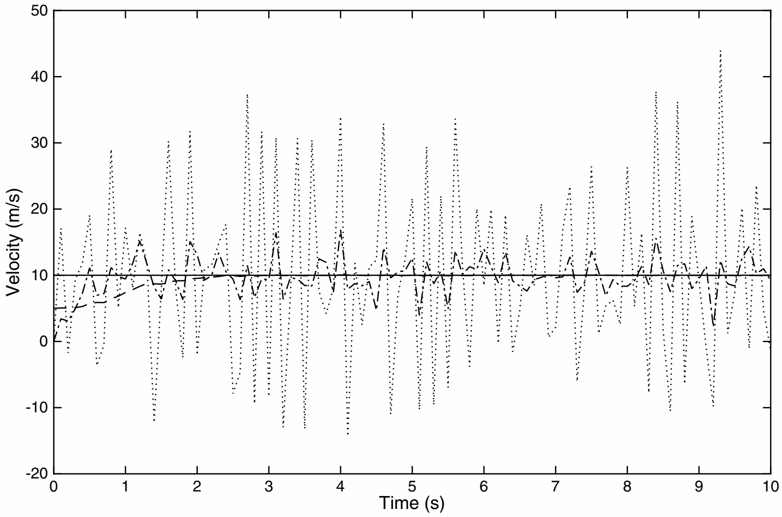


Figure 3.16. *Velocity estimation results for constant-velocity dynamics. True value (solid); from raw measurements (dotted); from running average of raw measurements (dotted-dashed); KF estimation (dash).*

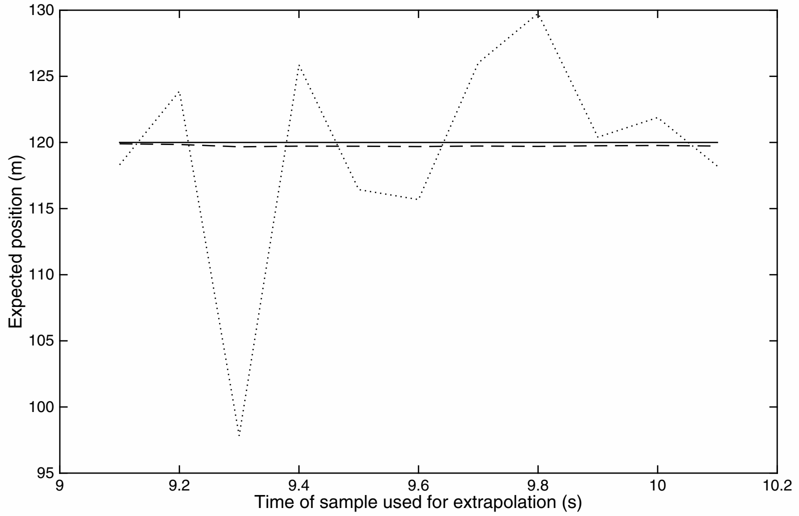


Figure 3.17. *Extrapolation of position 2 seconds ahead for constant-velocity dynamics. True value (solid); from running average of raw measurements (dotted); KF estimation (dashed).*

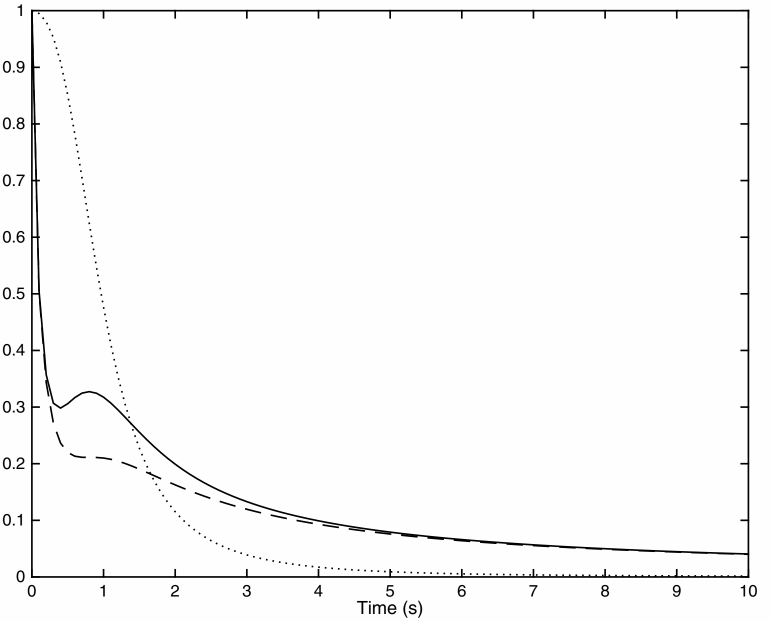


Figure 3.18. *Convergence of the KF parameters for constant-velocity dynamics. K (solid); P_{11} (dashed); P_{22} (dotted).*