



This chapter introduces the model-state- and parameter estimation problem from basic principles starting with Bayes' theorem. We define the general problem formulation and introduce the concept of Bayes' theorem solved recursively over a sequence of assimilation time windows. We also present different assimilation- and parameter-estimation problems, including model controls and errors, and show how they fit into a similar and general framework. Furthermore, this chapter introduces the notation and problem formulation considered in the following chapters.

2.1 Bayesian Formulation

This section introduces the concept of a data assimilation window, followed by the dynamical model and its uncertain quantities, the definitions of the model state, and the state vector. Then we discuss the measurements and the measurement equation before we formulate the general Bayesian data-assimilation problem.

2.1.1 Assimilation Windows

We typically solve the data-assimilation problem sequentially over a sequence of *assimilation time windows*. We adopt a rather general definition for an assimilation window to allow for various data-assimilation formulations and methods. For some methods, the assimilation windows are fixed-length intervals in time. E.g., in atmospheric data assimilation, it is common to use assimilation windows of six or twelve hours in length. In contrast, the assimilation window is the time interval between available measurements for Kalman-filter-type methods. We will later see that some assimilation methods update the model solution over the whole window, while others compute it at a particular time. Additionally, some assimilation methods treat each assimilation window independently, while others propagate information from one window to the next.

2.1.2 Model with Uncertain Inputs

We assume a forward model that describes a dynamic process with uncertainty over an assimilation window

$$\mathbf{x}_0 = \hat{\mathbf{x}}_0 + \mathbf{x}'_0, \quad (2.1)$$

$$\boldsymbol{\theta} = \hat{\boldsymbol{\theta}} + \boldsymbol{\theta}', \quad (2.2)$$

$$\mathbf{u} = \hat{\mathbf{u}} + \mathbf{u}', \quad (2.3)$$

$$\mathbf{q} = \mathbf{0} + \mathbf{q}', \quad (2.4)$$

$$\mathbf{x}_k = \mathbf{m}(\mathbf{x}_{k-1}, \boldsymbol{\theta}, \mathbf{u}_k, \mathbf{q}_k). \quad (2.5)$$

Here, \mathbf{x}_0 is a vector containing the model's *uncertain initial conditions* $\hat{\mathbf{x}}_0$ with uncertainty represented by \mathbf{x}'_0 . The vector $\boldsymbol{\theta}$ denotes a set of *uncertain model parameters* with prior values $\hat{\boldsymbol{\theta}}$ and uncertainty $\boldsymbol{\theta}'$. We assume parameters $\boldsymbol{\theta}$ to be constant in time. Furthermore, we define the *uncertain model errors* $\mathbf{q}^T = (\mathbf{q}_1^T, \dots, \mathbf{q}_K^T)$ with uncertainty \mathbf{q}' . The model errors account for missing physics in the model equations and numerical discretization errors. Note that we distinguish between the time-independent parameter uncertainty and time-dependent model errors by defining $\boldsymbol{\theta}$ and \mathbf{q} separately. The *uncertain model controls* $\mathbf{u}^T = (\mathbf{u}_1^T, \dots, \mathbf{u}_K^T)$ with uncertainty \mathbf{u}' represent various forms of time-dependent but uncertain model forcing. We have defined K as the number of time steps over the assimilation window. For simplicity, we have ignored any boundary conditions and their potential uncertainty, as this would add additional constraints to the model system.

2.1.3 Model State

We define $\mathbf{x}^T = (\mathbf{x}_0^T, \dots, \mathbf{x}_K^T)$ as a sequence of model-state vectors over an assimilation window. On some occasions, we will, for short, write $\mathbf{x} = \mathbf{m}(\mathbf{x}_0, \boldsymbol{\theta}, \mathbf{u}, \mathbf{q})$ where the model operator predicts the model state over the whole assimilation window. We differentiate between the model state \mathbf{x} and the data-assimilation problem's more general state vector, discussed next, although they will be the same in some cases.

2.1.4 State Vector

We define the data-assimilation problem's state vector \mathbf{z} , containing all the uncertain quantities we wish to estimate. The variables included in \mathbf{z} depend on the data-assimilation problem at hand and its formulation. There are, however, two main formulations to be aware of.

In the first formulation, \mathbf{z} includes the model prediction, or model state, \mathbf{x} , or a subset of \mathbf{x} (e.g., \mathbf{x}_K). In this case, we update the model state \mathbf{x} directly, and we denote it as the *model-state formulation*, where we can have $\mathbf{z}^T = (\mathbf{x}^T, \boldsymbol{\theta}^T, \mathbf{u}^T)$, excluding the model error \mathbf{q} .

In the second formulation, we treat the model error as an uncertain variable that we will estimate. We then write $\mathbf{z}^T = (\mathbf{x}_0^T, \boldsymbol{\theta}^T, \mathbf{u}^T, \mathbf{q}^T)$ and note that as soon as we give \mathbf{z} , we also determine the model state \mathbf{x} . We denote this formulation as the *forcing formulation* because the estimated model errors force the model.

It is important to note that we cannot simultaneously estimate both \mathbf{x} and \mathbf{q} as the model equations uniquely connect these variables. We will also see below that different assimilation methods will use either one or the other formulation.

2.1.5 Formulation Over Multiple Assimilation Windows

To understand which approximations we impose when solving the data-assimilation problem for a single assimilation window, we start by formulating the general or complete problem over multiple assimilation windows. The model state over L assimilation windows is the model-state trajectory $\mathcal{X}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_L^T)$. For the remainder of the book, we will use an index k denoting a particular time step t_k , while an index l refers to an assimilation window. We also define $\mathcal{U}^T = (\mathbf{u}_1^T, \dots, \mathbf{u}_L^T)$ and $\mathcal{Q}^T = (\mathbf{q}_1^T, \dots, \mathbf{q}_L^T)$ as the time sequences of controls and model errors.

We gather the model-state trajectory, \mathcal{X} , the parameters $\boldsymbol{\theta}$, the model controls \mathcal{U} , and the model errors \mathcal{Q} , into the *state vector* in either the state formulation $\mathcal{Z}^T = (\mathcal{X}^T, \boldsymbol{\theta}^T, \mathcal{U}^T)$ or the forcing formulation $\mathcal{Z}^T = (\mathcal{X}_0^T, \boldsymbol{\theta}^T, \mathcal{U}^T, \mathcal{Q}^T)$ where \mathcal{X}_0 is the initial condition for the first assimilation window. So \mathcal{Z} holds all uncertain quantities over all the assimilation windows.

2.1.6 Measurements with Errors

We also have a vector of measurements \mathcal{D} of the model predicted state \mathcal{X} represented by a measurement equation

$$\mathcal{D} = \mathcal{H}(\mathcal{X}) + \mathcal{E}. \quad (2.6)$$

Here \mathcal{H} is the so-called measurement operator, a potentially nonlinear function that maps the model state vector \mathcal{X} into measurement space. The matrix \mathcal{E} contains the measurement errors. Note that we will use the terms “measurement errors” and “observation errors” interchangeably.

We note that \mathcal{Z} includes \mathcal{X} in the state formulation, and we can equally write

$$\mathcal{D} = \mathcal{G}(\mathcal{Z}) + \mathcal{E}, \quad (2.7)$$

where \mathcal{G} relates the measurements to \mathcal{Z} .

In the forcing formulation we can still use Eq. (2.7) but rewrite it as

$$\mathcal{D} = \mathcal{G}(\mathcal{Z}) + \mathcal{E} = \mathcal{H}(\mathcal{M}(\mathcal{Z})) + \mathcal{E}, \quad (2.8)$$

where \mathcal{G} measures the model prediction from the input state vector, and $\mathcal{M}(\cdot)$ is the model operator that maps \mathcal{Z} to \mathcal{X} .

The term \mathcal{E} contains the measurement errors, including instrument errors and possibly a representation error that accounts for different representations of reality

by the measurements and the model. Representation errors are typically errors due to unresolved scales and processes, and Hodyss and Nichols (2015) and Van Leeuwen (2015) provide a further understanding of their origin and how to treat them in the data assimilation problem. An example of representation error occurs in satellite-altimetry data measuring the height of the sea surface. Apart from height differences induced by large-scale ocean currents and eddies, these measurements also contain height signals resulting from ocean tides. Large-scale ocean models often do not include these tides for computational reasons, and hence the observations have processes that the ocean models do not resolve. The representation errors may also result from using an erroneous measurement operator or errors introduced during the measurement preprocessing (Janjić et al., 2018).

2.1.7 Bayesian Inference

It is convenient to formulate the data-assimilation problem as a Bayesian inference problem. This formulation is entirely general and applies to both the state and forcing formulations discussed above. The starting point is an initial or prior probability density, $f(\mathcal{Z})$, of the quantity of interest, \mathcal{Z} . In the following, we use the notation $f(\cdot)$ to describe the argument's probability density function (pdf), meaning that, for instance, $f(\mathcal{Z})$ denotes the pdf of \mathcal{Z} , and $f(\mathbf{q})$ denotes the pdf of \mathbf{q} .

The measurements enter the data-assimilation problem via the likelihood $f(\mathcal{D}|\mathcal{Z})$. The likelihood is one of the less well-understood parts of the data-assimilation problem. To fully grasp its meaning, we should distinguish between the actual measurement process and how we treat measurements in the data-assimilation process. During the measurement process, the measurements are random variables. We measure them from the unknown true state of the system, to which nature adds a random draw from the measurement-error pdf. *Once we have collected the measurements, they are not random but fixed.* Of course, the measurements have errors, but that does not make them random because we know them exactly. This realization means that the likelihood is not the pdf of the measurements but rather a function of the state, and the measurements are fixed in that function. Detailed further discussions can be found in Van Leeuwen (2015, 2020).

To obtain the likelihood, we need to calculate $f(\mathcal{D}|\mathcal{Z})$ for each possible \mathcal{Z} , which is the (unnormalized) probability that this specific vector \mathcal{Z} would result in the fixed set of measurements. Note that to obtain the likelihood $f(\mathcal{D}|\mathcal{Z}) = f(\mathcal{E}) = f(\mathcal{D} - \mathcal{G}(\mathcal{Z}))$, we need to prescribe the probability-density function of the measurement errors \mathcal{E} .

Assuming knowledge of the probability density $f(\mathcal{Z})$ of all the uncertain variables in the state vector \mathcal{Z} and the likelihood $f(\mathcal{D}|\mathcal{Z})$, we can define a general form of the data-assimilation problem through Bayes' theorem. We can derive Bayes' theorem from the definition of conditional pdfs,

$$f(\mathcal{Z}, \mathcal{D}) = f(\mathcal{Z}|\mathcal{D})f(\mathcal{D}) = f(\mathcal{D}|\mathcal{Z})f(\mathcal{Z}), \quad (2.9)$$

where we solve for $f(\mathcal{Z}|\mathcal{D})$ to get

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathcal{D}|\mathcal{Z}) f(\mathcal{Z})}{f(\mathcal{D})}. \quad (2.10)$$

This equation becomes Bayes' theorem when used with the fixed measurement set \mathcal{D} . Lorenc (1988) and Tarantola (1987) introduced the Bayesian formulation for time-independent problems, and it was extended and generalized for time-dependent problems by Van Leeuwen and Evensen (1996). The pdf of the state vector \mathcal{Z} given the measurements \mathcal{D} , i.e., $f(\mathcal{Z}|\mathcal{D})$ is the solution to the data-assimilation problem. We stress that we know the observations in data assimilation, \mathcal{D} is not a random vector, and Bayes' theorem is a point-wise equation for each vector \mathcal{Z} .

The denominator on the right-hand side is the marginal pdf of the measurements, $f(\mathcal{D})$. It acts as a normalization constant that ensures that the posterior pdf integrates to one. Indeed, we can write

$$f(\mathcal{D}) = \int f(\mathcal{D}, \mathcal{Z}) d\mathcal{Z} = \int f(\mathcal{D}|\mathcal{Z}) f(\mathcal{Z}) d\mathcal{Z}, \quad (2.11)$$

making the normalization explicit. This normalization constant has also given rise to misunderstandings. Some have argued that $f(\mathcal{D}) = 1$ since we know the measurements, but this is incorrect. Instead, one should evaluate the pdf of the measurements at the value of the fixed measurements \mathcal{D} . This pdf centers on the unknown true state's measurement, so the above equation is the proper way to evaluate $f(\mathcal{D})$.

However, this normalization constant is seldom needed explicitly, and we will not use it in the rest of this book. Actually, $f(\mathcal{D})$ only shows up when evaluating different numerical models given a set of measurements via the "model evidence," which is the normalization constant assessed for each of these models, e.g., the ECMWF model versus the Met Office model. The model with the highest $f(\mathcal{D})$ is considered the best model. In contrast, a proper Bayesian would set up a prior over the different numerical models and interpret the normalization factor as the likelihood in this higher-order data-assimilation problem. Since the number of models tested is finite, this is a discrete data-assimilation problem where the state vector \mathcal{Z} contains a complete numerical model.

It is essential to realize that Bayes' theorem describes a forward problem and not an inverse problem. We start with the prior probability density function and multiply it with the likelihood to form the posterior probability density function, which is the solution to the problem. Of course, many practical data-assimilation methods solve an inverse problem to find an approximation to the posterior, but many others do not. As such, we can consider inverse problem theory as a subset of Bayesian inference; see, for instance, the introduction in Van Leeuwen et al. (2015).

2.2 Recursive Bayesian Formulation

The Bayesian formulation above updates the solution over the whole assimilation period, including multiple assimilation windows, in one go. This formulation is not always convenient, as, in many data-assimilation problems, the measurements

become available sequentially. Thus, we will next introduce two approximations to simplify the process of solving Eq. (2.10).

2.2.1 Markov Model

Approximation 1 (Model is 1st-order Markov process) *We assume the dynamical model is a 1st-order Markov process.* □

A first-order Markov process denotes that the future is independent of the past if the present is known. If the model is a 1st-order Markov process, it follows that we can compute the model solution in one assimilation window from the solution in the previous window. We can then express mathematically the condition

$$f(\mathbf{z}_l | \mathbf{z}_{l-1}, \mathbf{z}_{l-2}, \dots, \mathbf{z}_0) = f(\mathbf{z}_l | \mathbf{z}_{l-1}), \quad (2.12)$$

where we remind the reader that l is the index of an assimilation window.

From Approx. 1, we can use Eq. (2.12) and write $f(\mathcal{Z})$ as a recursion over the assimilation windows $l \in (1, \dots, L)$,

$$\begin{aligned} f(\mathcal{Z}) &= f(\mathbf{z}_0) f(\mathbf{z}_1 | \mathbf{z}_0) f(\mathbf{z}_2 | \mathbf{z}_1) \cdots f(\mathbf{z}_L | \mathbf{z}_{L-1}) \\ &= f(\mathbf{z}_0) \prod_{l=1}^L f(\mathbf{z}_l | \mathbf{z}_{l-1}). \end{aligned} \quad (2.13)$$

We notice that the assumption of the model being a Markov process affects the model-state evolution in time. However, we use the Markov property to formulate the model prior as a recursion over the assimilation windows.

2.2.2 Independent Measurements

Next, we introduce the following assumption on the measurements.

Approximation 2 (Independent measurements) *We assume that measurements are independent between different assimilation windows.*

Independent measurements mean that their errors are uncorrelated, so we now assume the measurement-error correlations are zero between measurements in different assimilation windows. If we use Approx. 2, we can write the likelihood for the measurement vector \mathcal{D} as a product of independent likelihoods, one for each assimilation window, as

$$f(\mathcal{D} | \mathcal{Z}) = \prod_{l=1}^L f(\mathbf{d}_l | \mathbf{z}_l). \quad (2.14)$$

The assumption of independence of measurements collected in different assimilation windows is the first grave approximation we make, as such correlations often exist, and we still neglect them. However, we retain the possibility of having correlated measurement errors for the measurements collected within each assimilation window.

Note that, by “independent measurements,” we do not mean that the measurements’ information is independent. This terminology typically conveys that the *measurement errors* are uncorrelated. E.g., we can measure a temperature at a location twice using different sensors, so the measurement errors are uncorrelated. But, since they measure the same quantity, we do not double the information. As Evensen and Eikrem (2018) discussed, there is redundancy in the measurement information. Likewise, spatial correlations in the measured quantity will also result in measurements with correlated information. The measurement errors can still be uncorrelated.

2.2.3 Recursive form of Bayes’

The general form of Bayes’ in Eq. (2.10) now becomes

$$f(\mathcal{Z}|\mathcal{D}) \propto \prod_{l=1}^L f(\mathbf{d}_l|\mathbf{z}_l) f(\mathbf{z}_l|\mathbf{z}_{l-1}) f(\mathbf{z}_0). \quad (2.15)$$

By rearranging the order of the multiplications, it is possible to write Eq. (2.15) as a recursion, following Evensen & Van Leeuwen, (2000),

$$f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0)}{f(\mathbf{d}_1)}, \quad (2.16)$$

$$f(\mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \quad (2.17)$$

⋮

$$f(\mathcal{Z}|\mathcal{D}) = \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}, \dots, \mathbf{z}_0|\mathbf{d}_{L-1}, \dots, \mathbf{d}_1)}{f(\mathbf{d}_L)}. \quad (2.18)$$

Thus, we have defined the data-assimilation problem as a recursion in time. We start in Eq. (2.16), with $f(\mathbf{z}_0)$ being the prior density for the initial conditions \mathbf{z}_0 . Then, $f(\mathbf{z}_1|\mathbf{z}_0)$ denotes the integration of the model from the initial condition to predict the pdf of \mathbf{z}_1 given \mathbf{z}_0 . The multiplication with the likelihood $f(\mathbf{d}_1|\mathbf{z}_1)$ conditions the model prediction on the measurements \mathbf{d}_1 . The posterior estimate is then the joint pdf for \mathbf{z}_0 and \mathbf{z}_1 conditioned on the measurements \mathbf{d}_1 , denoted by $f(\mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1)$.

The posterior estimate from Eq. (2.16) now becomes the prior in Eq. (2.17), where we again integrate the model, $f(\mathbf{z}_2|\mathbf{z}_1)$, and condition on the data, $f(\mathbf{d}_2|\mathbf{z}_2)$, to obtain the posterior $f(\mathbf{z}_2, \mathbf{z}_1, \mathbf{z}_0|\mathbf{d}_1, \mathbf{d}_2)$, which becomes the prior for the following recursion.

2.2.4 Marginal Bayes' for Filtering

When the aim is to make better predictions, we can simplify the recursion in Eqs. (2.16)–(2.17) by exploiting the Markovian property of the model and the sequential nature of the data-assimilation problem. Thus, by integrating out the model states of previous assimilation time windows, we can write the recursion in terms of the marginals as

$$f(\mathbf{z}_1|\mathbf{d}_1) = \frac{f(\mathbf{d}_1|\mathbf{z}_1) \int f(\mathbf{z}_1|\mathbf{z}_0) f(\mathbf{z}_0) d\mathbf{z}_0}{f(\mathbf{d}_1)} = \frac{f(\mathbf{d}_1|\mathbf{z}_1) f(\mathbf{z}_1)}{f(\mathbf{d}_1)}, \quad (2.19)$$

$$f(\mathbf{z}_2|\mathbf{d}_1, \mathbf{d}_2) = \frac{f(\mathbf{d}_2|\mathbf{z}_2) \int f(\mathbf{z}_2|\mathbf{z}_1) f(\mathbf{z}_1|\mathbf{d}_1) d\mathbf{z}_1}{f(\mathbf{d}_2)} = \frac{f(\mathbf{d}_2|\mathbf{z}_2) f(\mathbf{z}_2|\mathbf{d}_1)}{f(\mathbf{d}_2)}, \quad (2.20)$$

⋮

$$\begin{aligned} f(\mathbf{z}_L|\mathcal{D}) &= \frac{f(\mathbf{d}_L|\mathbf{z}_L) \int f(\mathbf{z}_L|\mathbf{z}_{L-1}) f(\mathbf{z}_{L-1}|\mathbf{d}_{L-1}, \dots, \mathbf{d}_1) d\mathbf{z}_{L-1}}{f(\mathbf{d}_L)} \\ &= \frac{f(\mathbf{d}_L|\mathbf{z}_L) f(\mathbf{z}_L|\mathbf{d}_{L-1})}{f(\mathbf{d}_L)}. \end{aligned} \quad (2.21)$$

Note that by solving for the marginals, we are not solving the complete original problem defined by Bayes' theorem in Eq. 2.10. We are applying the following approximation.

Approximation 3 (Filtering assumption) *We approximate the full smoother solution with a sequential data-assimilation solution. We only update the solution in the current assimilation window, and we do not project the measurement's information backward in time from one assimilation window to the previous ones.* □

We recursively accumulate more and more information from the measurements from time window to time window. Hence, \mathbf{z}_l contains the information from all the previous measurements, including those from the l 'th assimilation window, and is the ideal starting point for predicting \mathbf{z}_{l+1} and onwards.

We note that if the assimilation window is a single model time step, then Approx. 3 reduces to the standard filter approximation in, e.g., Kalman-filter methods, which update the solution at the current time before continuing the integration.

Another attractive property of these equations is their similarity. We are solving the same computational problem in each update step. We have a state vector for the current time window containing the information from all the previous measurements, and we combine it with the new measurements. Thus, if we define \mathbf{z}_l to represent the model prediction at time window l and \mathbf{d}_l the measurements for this time window, we can write the general update equation as

$$f(\mathbf{z}_l|\mathbf{d}_l) = \frac{f(\mathbf{d}_l|\mathbf{z}_l) f(\mathbf{z}_l)}{f(\mathbf{d}_l)}, \quad (2.22)$$

or more generally as

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{z}) f(\mathbf{z})}{f(\mathbf{d})}, \quad (2.23)$$

which is again just the Bayes formula in Eq. (2.10) but now for a subset of the state vector and the measurements.

2.3 Error Propagation

So far, we have considered formulations for updating the model solution over an assimilation window and assume that we have a prior pdf for the solution. We will now discuss how to obtain or estimate this prior distribution and propagate it to the next assimilation window.

2.3.1 Fokker–Planck Equation

Ideally, we know the full pdf at the end of the previous assimilation window or equivalently at the beginning of the current assimilation window. We can compute the evolution of the prior pdf from the start of one assimilation window to the next, $f(\mathbf{z}_l|\mathbf{z}_{l-1})$, from the Fokker–Planck equation. We start with a stochastic model with additive Gaussian model errors forming a Markov process, initialized with the model state \mathbf{x} from \mathbf{z}_l . We assume that the prior model parameters do not change over the assimilation window, so we omit them here for ease of notation. The stochastic model equation reads

$$d\mathbf{x} = \mathbf{m}(\mathbf{x}) dt + d\mathbf{q}, \quad (2.24)$$

giving the increment in the model state $d\mathbf{x}$ resulting from a time increment dt and the stochastic forcing $d\mathbf{q}$. From this model equation, one can derive the standard form of the Fokker–Planck equation (also named Kolmogorov’s equation),

$$\frac{\partial f(\mathbf{x})}{\partial t} + \sum_i \frac{\partial (m_i(\mathbf{x}) f(\mathbf{x}))}{\partial x_i} = \frac{1}{2} \mathbf{C}_{qq} \sum_{i,j} \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}, \quad (2.25)$$

which describes the time evolution of the model state vector’s probability density $f(\mathbf{x})$. For the case with non-additive model errors, it becomes more elaborate to derive an equation for the probability density’s time evolution, and we will not discuss that here. In Eq. (2.25), m_i is the component number i of the model operator \mathbf{m} , and \mathbf{C}_{qq} is the model-error covariance matrix. This equation is the fundamental equation for the evolution of error statistics. The equation describes the probability change in a local volume resulting from the model dynamics and diffusion terms. The model-dynamics term is a divergence of the probability flux into the local volume. In contrast, the diffusion term tends to smooth $f(\mathbf{x})$ due to the stochastic model forcing. Unfortunately, we cannot solve this equation in high dimensions. We refer to Jazwinski (1970) for the actual derivation and further discussion.

2.3.2 Covariance Evolution Equation

By taking moments of the Fokker–Planck equation, we can derive an evolution equation for statistical moments of the uncertainty in the model state vector \mathbf{x} . Alternatively, we can derive an evolution equation for the error covariance matrix by comparing the evolution of the true model state with that of our estimated model state as

$$\mathbf{x}_{k+1}^t = \mathbf{m}(\mathbf{x}_k^t) + \mathbf{q}_k \approx \mathbf{m}(\mathbf{x}_k) + \mathbf{M}_k(\mathbf{x}_k^t - \mathbf{x}_k) + \mathbf{q}_k, \quad (2.26)$$

$$\mathbf{x}_{k+1} = \mathbf{m}(\mathbf{x}_k), \quad (2.27)$$

where we used a first-order Taylor expansion in the first equation, assuming that $\mathbf{x}_k^t - \mathbf{x}_k$ is small. The superscript, t, denotes “true.” Note that the linearized model \mathbf{M}_k can include estimates of the model parameters. By subtracting Eq. (2.27) from Eq. (2.26), multiplying the resulting equation with its transpose, and taking the expectation, we obtain the error covariance equation used in Kalman filters,

$$\mathbf{C}_{xx,k+1} \approx \mathbf{M}_k \mathbf{C}_{xx,k} \mathbf{M}_k^T + \mathbf{C}_{qq,k}. \quad (2.28)$$

Here \mathbf{M}_k is the model’s tangent-linear operator evaluated at \mathbf{x}_k and \mathbf{C}_{qq} is the model error covariance matrix. The extended Kalman filter (EKF) also uses this linearized error-evolution equation. In addition to an immense computational load for real-size geoscience models, Evensen (1992) showed that the approximate linear equation can not saturate unstable modes of the model, and one can experience unbounded error-variance growth. See also the example in Chap. 12.

2.3.3 Ensemble Predictions

An alternative to the two previous approaches is to use a Monte-Carlo method for propagating the error statistics. For example, if we have an ensemble of samples from a pdf at time t_k , we can integrate these samples forward until t_{k+1} by running the dynamical model for each sample separately. If the model contains model errors, we can treat these by using a stochastic model description. Ensemble integration is the approach used in the ensemble methods. It involves creating a large and finite number of model realizations or samples that represent our prior understanding of the system and its uncertainties. Ensemble Kalman filters, particle filters, and particle-flow filters all use this approach. The only approximation of this approach is the limitation of the ensemble size to a finite number of model realizations or ensemble members. Thus, ensemble integration is an attractive method for propagating the uncertainty information over an assimilation window or from one assimilation window to the next. In Fig. 2.1, the ensemble integration is illustrated by the blue lines indicating the prior ensemble integration in an ensemble smoother. The figure also shows how the ensemble prediction (represented by the green lines) captures the uncertainty through the updated ensemble members’ spread.

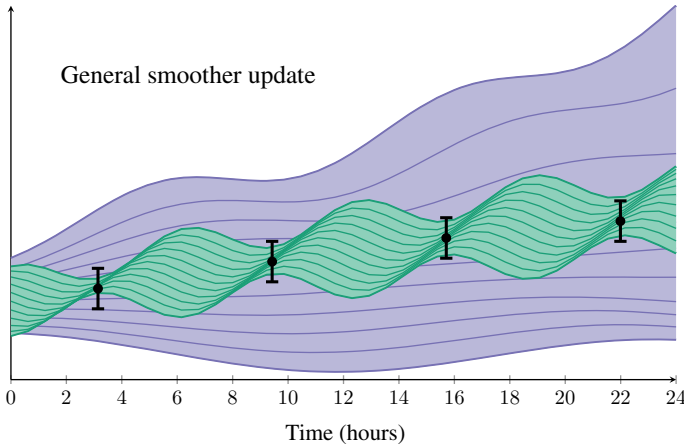


Fig. 2.1 The figure illustrates using a general ensemble smoother (Sect. 2.4.1). The black dots denote observations with an error of one standard deviation. We first run a full ensemble integration over the whole assimilation window, indicated by blue lines in the blue envelope. After that, we update the ensemble simultaneously in space and time, resulting in the green lines in the green envelope

2.4 Various Problem Formulations

In the current formulation, the state vector \mathbf{z} is rather general. We can define \mathbf{z} to contain the model state over an assimilation time window or at a single instant in time. It is also possible to include other uncertainties, such as model parameters, model controls, and model errors. We will see below that the specific solution procedure will be the same in any of these cases.

2.4.1 General Smoother Formulation

A general smoother formulation solves the original Bayes' formula (2.22) in an assimilation window. Let's define the state vector $\mathbf{z} = \mathbf{x} = \mathbf{m}(\mathbf{x}_0, \mathbf{q})$ as the model solution over the whole assimilation window with distributed measurements. We also allow the model to have errors \mathbf{q} but do not include uncertain parameters and controls in this example. In this case, we will update the model state \mathbf{x} , which connects to the model-predicted measurements \mathbf{y} through the functional

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{m}(\mathbf{x}_0, \mathbf{q})). \tag{2.29}$$

Here the vector function $\mathbf{g}(\mathbf{z})$ is just the application of the measurement operator acting on the model prediction to generate the predicted measurements. We then compare the predicted measurements to the actual measurements in the likelihood.

We illustrate the formulation in an ensemble setting. The general smoother starts with a prior ensemble integration over the assimilation window as indicated by the

blue lines in Fig. 2.1. We obtain the ensemble estimate of the posterior pdf by combining the prior ensemble in space and time with the likelihood. As such, it updates the prior ensemble and results in an updated ensemble that is closer to the measurements and has a reduced uncertainty, as illustrated by the green lines in Fig. 2.1.

For long windows and nonlinear dynamics the prior pdf becomes significantly non-Gaussian. In this case, methods that assume a Gaussian prior will struggle, as we illustrate when applying the ensemble smoother (ES) for the Lorenz (1963) model in Chap. 15.

2.4.2 Filter Formulation

Let's define $\mathbf{z} = \mathbf{x}_K$ as the model solution at the end of the assimilation window, where we assume we have measurements. We can then compute the so-called filter solution typically solved by particle filters and Kalman filter methods. In this case, we have the state \mathbf{x}_K connected to the model-predicted measurements \mathbf{y} through the measurement-functional $\mathbf{h}(\mathbf{x}_K)$,

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K). \quad (2.30)$$

The vector function $\mathbf{g}(\mathbf{z})$ now equals the measurement operator $\mathbf{h}(\mathbf{x}_K)$, which maps the model solution at the end of the assimilation window to the predicted measurements \mathbf{y} . We can then compute the updated solution at the end of the assimilation window from Bayes' formula in Eq. (2.23), as illustrated in Fig. 2.2. After that, we continue the integration through the next assimilation window. Thus, we have

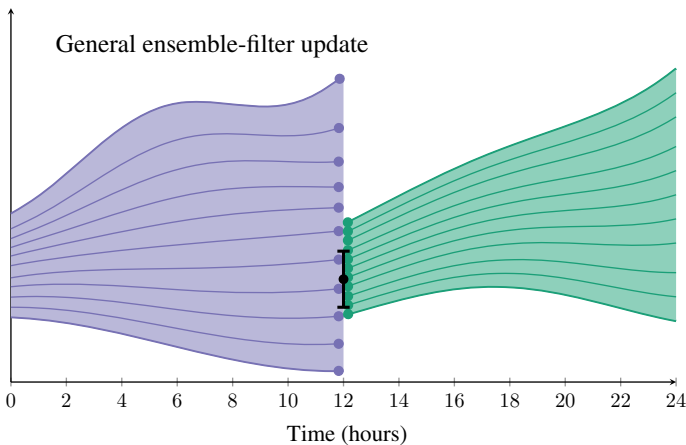


Fig. 2.2 The figure illustrates the general ensemble-filter update of an ensemble prediction (Sect. 2.4.2). The filter updates the ensemble at the measurement time before continuing the ensemble integration

integrated away the model solution from all previous time steps of the assimilation window to create the marginal pdf at the end of the assimilation window.

The filter approach can alleviate a potential practical disadvantage of the general smoother discussed in the previous section. The prior covers a large time window in the smoother-formulation and might be less accurate for measurements later in the window. Furthermore, if the model is highly nonlinear, strong non-Gaussian prior pdfs can develop. An advantage of the filter approach is that we can divide an assimilation window with observations at different time instances into several shorter windows, each ending at an observation time. This approach results in a sequential data-assimilation problem that facilitates, e.g., Kalman filters, ensemble Kalman filters, and particle filters. The more frequent model updating will keep the model closer to the observations. Furthermore, the prior at each measurement time remains relatively narrow and more Gaussian than in the smoother formulation, resulting in a sequence of more accurate updates. Several of the example chapters consider various filter applications.

2.4.3 Recursive Smoother Formulation

Evensen and Van Leeuwen (2000) introduced a recursive smoother formulation as an extension of the filtering problem. The state vector is now $\mathbf{z}^T = (\dots, \mathbf{x}_{j-1}^T, \mathbf{x}_j^T)$ containing the model state at all, or several previous and the current assimilation windows. And we have the measurements located at the end of the current assimilation window. Thus, we have a problem where

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{x}_K), \quad (2.31)$$

but we compute the update for the whole \mathbf{z} .

Instead of solving for the marginal in Eq. (2.20), we solve the recursion in Eq. (2.17) while processing the measurements sequentially as in the filter formulation. This approach recursively introduces the information from additional measurements in every new assimilation window and “projects” this information at previous times. The formulation inherits the advantages from the filter formulation discussed in Sect. 2.4.2. We will discuss this approach when applying the ensemble Kalman smoother in Chap. 15, and see also the Algorithm 7 in Chap. 6.

Note that the solution at the final assimilation time is identical to the filter solution. Therefore, the recursive smoother is not adding any value for prediction problems, but the formulation is an excellent alternative to the general smoother for hindcast problems (see Fig. 2.3). We can also use this formulation as a “lagged” smoother where we only update the state vector for a selected number of previous assimilation windows. We then exploit that the measurement’s information decorrelates with time in nonlinear models with model errors, so, in practice, we apply a form of localization as discussed in Chap. 10. Thus, the recursive smoother introduces the measurements sequentially as in the filter, while the general smoother in Sect. 2.4.1 computes one global update over the whole assimilation window in one go.

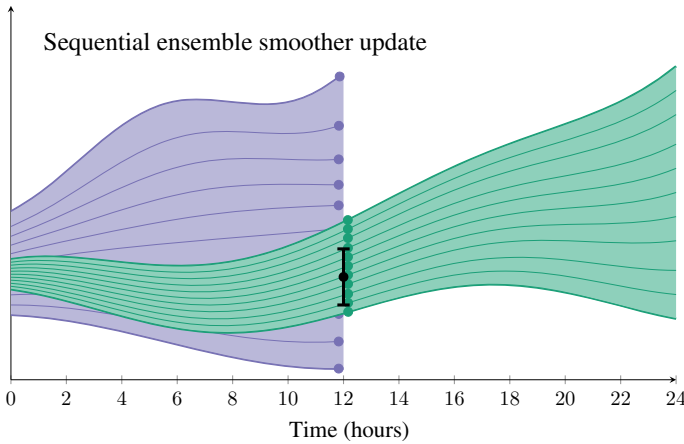


Fig. 2.3 The figure illustrates using a recursive ensemble smoother (Sect. 2.4.3). Like in the filter update, we update the ensemble at the measurement time before continuing the integration, but we also update the ensemble at all previous times

2.4.4 A Smoother Formulation for Perfect Models

An often used assimilation formulation assumes that the state vector $\mathbf{z} = \mathbf{x}_0$ only contains the model state at the beginning of the time window. Hence, we exclude uncertain parameters and controls, and we assume no model errors. We then estimate the initial model state for the assimilation window and obtain the solution over the assimilation window by integrating the model from the updated initial conditions. The equation for the predicted measurements \mathbf{y} is

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\mathbf{x}_0)). \quad (2.32)$$

Here we start with the solution at the beginning of the time window and integrate the model $\mathbf{x} = \mathbf{m}(\mathbf{x}_0)$ over the time window to make a prediction. Then we apply the measurement functional on the model prediction $\mathbf{h}(\mathbf{x})$ and compare this prediction and the actual observation in the likelihood, which also propagates this difference back to the beginning of the time window to perform an update. This formulation is the basis for deriving the so-called strong-constraint 4DVar schemes widely used for weather-prediction applications and the iterative ensemble smoothers used to history-match reservoir models in petroleum-engineering applications. Fig. 2.4 illustrates this alternative smoother formulation used by the ensemble version of 4DVar, En4DVar, and EnRML methods discussed below. This formulation also allows for measurements distributed over the assimilation window. In Sect. 2.5, we will see that this problem, and the ones in the following two sections, requires a modified form of Bayes' theorem.

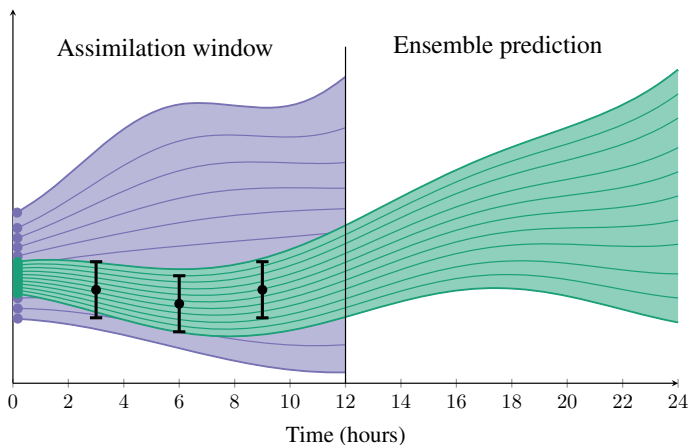


Fig. 2.4 The figure illustrates the recursive smoother formulation assuming a perfect model solved using an ensemble approach. One defines an assimilation time window and updates the initial conditions of the ensemble realizations at the beginning of the time window. Both the iterative ensemble smoothers and strong-constraint 4DVar use this approach. We based this graphic on an illustration from the ECMWF Forecast User Guide <https://confluence.ecmwf.int/display/FUG/Forecast+User+Guide>

2.4.5 Parameter Estimation

An analog problem to the smoother problem with a perfect model is the parameter-estimation problem for a model without uncertain controls and model errors. In this case, we define $\mathbf{z} = \boldsymbol{\theta}$ to contain the uncertain model parameters. This formulation leads to a situation where $\mathbf{z} = \boldsymbol{\theta}$, and we write, as usual,

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\boldsymbol{\theta})), \quad (2.33)$$

where $\mathbf{m}(\boldsymbol{\theta})$ shows that we need the forward model to evaluate how the parameters relate to the measurements. For the practical implementation and solution, this problem is analogous to the “smoother problem for perfect models” solved for each assimilation window in Sect. 2.4.4. This problem formulation is also the one we solve in petroleum applications (Evensen et al., 2019; Evensen, 2021) and Chap. 21. Thus, Fig. 2.4, used to illustrate state estimation, also represents the parameter-estimation problem. For the Bayesian formulation of the parameter estimation problem, we refer to Sect. 2.5.

2.4.6 Estimating Initial Conditions, Parameters, Controls, and Errors

In the final formulation, we present the problem of estimating the initial condition for the time window, together with model parameters, model controls, and model errors. Evensen (2019) explained how we could solve the recursive smoother problem for a time window when we include model errors, while Evensen (2021) considered the

case with additional control parameters. The controls can represent, for example, the imposed production rates or the aquifer strength in a reservoir model (Glegola et al., 2012; Peters et al., 2010) or the atmospheric forcing in ocean models (Vossepoel et al., 2004). Furthermore, Evensen (2021) illustrated how to include the model errors and controls in the state vector and simultaneously estimate the initial conditions \mathbf{x}_0 , the model errors \mathbf{q} , and the model controls \mathbf{u} over an assimilation window (see also the example in Chap. 21).

We define the state vector as $\mathbf{z}^T = (\mathbf{x}_0^T, \boldsymbol{\theta}^T, \mathbf{u}^T, \mathbf{q}^T)$, and given an estimate of \mathbf{z} , we compute the resulting model solution over the assimilation window, and hence the predicted measurements

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\mathbf{x}_0, \boldsymbol{\theta}, \mathbf{u}, \mathbf{q})). \quad (2.34)$$

Again the practical implementation and solution of this problem, although more complicated due to the extended state vector, is analogous to the problems solved in the two previous Sects. 2.4.4 and 2.4.5, and we again refer to the modified form of Bayes' theorem in the following section.

2.5 Including the Predicted Measurements in Bayes Theorem

In Sects. 2.4.4, 2.4.5, and 2.4.6 we solve a problem involving a relation between the state vector \mathbf{z} and the predicted measurements given by

$$\mathbf{y} = \mathbf{g}(\mathbf{z}) = \mathbf{h}(\mathbf{m}(\mathbf{z})). \quad (2.35)$$

In this case, it is not entirely clear how to evaluate the likelihood $f(\mathbf{d}|\mathbf{z})$ in Bayes' formula in Eq. (2.23). The reason is that we have measurements of the model state $\mathbf{x} = \mathbf{m}(\mathbf{z})$, which does not appear explicitly in Eq. (2.23). To solve this issue, we first write the likelihood to include the model state \mathbf{x} explicitly as follows

$$f(\mathbf{d}|\mathbf{z}) = \int f(\mathbf{d}, \mathbf{x}|\mathbf{z}) d\mathbf{x} = \int f(\mathbf{d}|\mathbf{x}, \mathbf{z}) f(\mathbf{x}|\mathbf{z}) d\mathbf{x}, \quad (2.36)$$

where, if \mathbf{z} contains the model initial condition \mathbf{x}_0 , we exclude it from \mathbf{x} . For a given model state, the model inputs \mathbf{z} do not provide extra information on the measurements. Thus, we can write

$$f(\mathbf{d}|\mathbf{x}, \mathbf{z}) = f(\mathbf{d}|\mathbf{x}), \quad (2.37)$$

leading to Bayes' theorem in the form

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{z})}{f(\mathbf{d})} \int f(\mathbf{d}|\mathbf{x}) f(\mathbf{x}|\mathbf{z}) d\mathbf{x}. \quad (2.38)$$

This problem is costly to compute as we have to integrate the model state over all possible model inputs as dictated by $f(\mathbf{x}|\mathbf{z})$. For example, in the pure parameter estimation problem of Sect. 2.4.5, we would need to evaluate the integral over all possible model initial conditions and all possible model trajectories in the assimilation window. And we would have to do that for each parameter vector $\boldsymbol{\theta}$. To avoid

this often intractable problem, one typically estimates both the model state \mathbf{x} and inputs in \mathbf{z} .

It is sometimes helpful to rewrite the likelihood of variables in observation space instead of in state space. Recall that the likelihood $f(\mathbf{d}|\mathbf{z})$ is a function of \mathbf{z} since we know the measurements \mathbf{d} in the assimilation process. We can transform the variable \mathbf{z} to observation space via the predicted measurements. Hence, we can augment the likelihood with the predicted measurements via

$$f(\mathbf{d}|\mathbf{z}) = \int f(\mathbf{d}, \mathbf{y}|\mathbf{z}) d\mathbf{y} = \int f(\mathbf{d}|\mathbf{y}, \mathbf{z}) f(\mathbf{y}|\mathbf{z}) d\mathbf{y}, \quad (2.39)$$

where the second equality follows from the definition of a conditional density.

We first note that once we know the predicted measurements, the state vector \mathbf{z} contains no new information on them, and we can write the likelihood

$$f(\mathbf{d}|\mathbf{y}, \mathbf{z}) = f(\mathbf{d}|\mathbf{y}). \quad (2.40)$$

Furthermore, using the relation between the predicted measurements and the state vector, $\mathbf{y} = \mathbf{g}(\mathbf{z})$, we can write

$$f(\mathbf{y}|\mathbf{z}) = \delta(\mathbf{y} - \mathbf{g}(\mathbf{z})), \quad (2.41)$$

where δ is the Dirac-delta function. We then have that a given state vector $\mathbf{z} = (\mathbf{x}_0, \boldsymbol{\theta}, \mathbf{u}, \mathbf{q})$ uniquely defines a set of predicted measurements \mathbf{y} . This equation is valid for predicted measurements over a whole assimilation window, also when we include model errors and controls as part of the state vector \mathbf{z} . Thus, it was a neat trick in Sect. 2.4.6 to include the model errors and controls in the state vector as a set of poorly known parameters (Evensen, 2019).

We now have the likelihood

$$f(\mathbf{d}|\mathbf{z}) = \int f(\mathbf{d}|\mathbf{y}) \delta(\mathbf{y} - \mathbf{g}(\mathbf{z})) d\mathbf{y} = f(\mathbf{d}|\mathbf{g}(\mathbf{z})), \quad (2.42)$$

allowing us to write Bayes theorem as

Bayes' theorem related to the predicted measurements

$$f(\mathbf{z}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{g}(\mathbf{z})) f(\mathbf{z})}{f(\mathbf{d})}. \quad (2.43)$$

To summarize, \mathbf{z} is the state vector we try to estimate in the data-assimilation problem. In the general-smoother in Sect. 2.4.1, $\mathbf{z} = \mathbf{x}$ is the model state trajectory over the assimilation window. In contrast, in the filter formulation in Sect. 2.4.2, the state vector represents the model state at the end of the assimilation window $\mathbf{z} = \mathbf{x}_K$. For the recursive smoother in Sect. 2.4.3, \mathbf{z} is the whole model state over the current and previous assimilation windows. Then in the smoother formulation for the perfect model in Sect. 2.4.4, where we estimate the initial state, the state vector is just the model initial conditions for the assimilation window $\mathbf{z} = \mathbf{x}_0$, which we consider as an input to the model. The situation is analogous in the parameter estimation problem in Sect. 2.4.5, where $\mathbf{z} = \boldsymbol{\theta}$, and in the final example in Sect. 2.4.6, where the state vector consists of all the uncertain model inputs $\mathbf{z}^T = (\mathbf{x}_0^T, \boldsymbol{\theta}^T, \mathbf{q}^T, \mathbf{u}^T)$.

The Bayes' formula in Eq. (2.43) applies to all these cases. For the state estimation examples in Sects. 2.4.1–2.4.3 we define $\mathbf{g}(\mathbf{z})$ as the measurement operator, while in the three last cases in Sects. 2.4.4–2.4.6, $\mathbf{g}(\mathbf{z})$ also includes the model integration. In the following, we will discuss popular data-assimilation methods that solve the Bayesian estimation problem in Eq. (2.43) under various approximations.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

