4

The time-dependent inverse problem: state estimation

4.1 Background

The discussion so far has treated models and data that most naturally represent a static world. Much data, however, describe systems that are changing in time to some degree. Many familiar differential equations represent phenomena that are intrinsically time-dependent; such as the wave equation,

$$\frac{1}{c^2}\frac{\partial^2 x\,(t)}{\partial t^2}-\frac{\partial^2 x(t)}{\partial r^2}=0. \tag{4.1}$$

One may well wonder if the methods described in Chapter 2 have any use with data thought to be described by (4.1). An approach to answering the question is to recognize that t is simply another coordinate, and can be regarded, e.g., as the counterpart of one of the space coordinates encountered in the previous discussion of two-dimensional partial differential equations. From this point of view, time-dependent systems are nothing but versions of the systems already developed. (The statement is even more obvious for the simpler equation,

$$\frac{\mathrm{d}^2 x\,(t)}{\mathrm{d}t^2}=q(t), \tag{4.2}$$

where the coordinate that is labeled t is arbitrary, and need not be time.)

On the other hand, time often has a somewhat different flavor to it than does a spatial coordinate because it has an associated direction. The most obvious example occurs when one has data up to and including some particular time t, and one asks for a *forecast* of some elements of the system at a future time $t' > t$. Even this role of time is not unique: one could imagine a completely equivalent spatial forecast problem, in which, e.g., one required extrapolation of the map of an ore body beyond some area in which measurements exist. In state estimation, time does not introduce truly novel problems. The main issue is really a computational one: problems in two or more spatial dimensions, when time-dependent, typically generate system

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

dimensions that are too large for conventionally available computer systems. To deal with the computational load, state estimation algorithms are sought that are computationally more efficient than what can be achieved with the methods used so far. Consider, as an example,

$$\frac{\partial C}{\partial t} = \kappa \nabla^2 C, \tag{4.3}$$

a generalization of the Laplace equation (a diffusion equation). Using a one-sided time difference, and the discrete form of the two-dimensional Laplacian in Eq. (1.13), one has

$$\begin{aligned}\frac{C_{ij}((n+1)\Delta t) - C_{ij}(n\Delta t)}{\Delta t} &= \kappa\{C_{i+1,j}(n\Delta t) - 2C_{i,j}(n\Delta t) + C_{i-1,j}(n\Delta t) \\ &\quad + C_{i,j+1}(n\Delta t) - 2C_{i,j}(n\Delta t) + C_{i,j-1}(n\Delta t)\}.\end{aligned} \tag{4.4}$$

If there are N^2 elements defining C_{ij} at each time $n\Delta t$, then the number of elements over the entire time span of T time steps would be TN^2, which grows rapidly as the number of time steps increases. Typically the relevant observation numbers also grow rapidly through time. On the other hand, the operation

$$\mathbf{x} = \mathrm{vec}(C_{ij}(n\Delta t)), \tag{4.5}$$

renders Eq. (4.4) in the familiar form

$$\mathbf{A}_1 \mathbf{x} = \mathbf{0}, \tag{4.6}$$

and with some boundary conditions, some initial conditions and/or observations, and a big enough computer, one could use without change any of the methods of Chapter 2. As $T,\ N$ grow, however, even the largest available computer becomes inadequate. Methods are sought that can take advantage of special structures built into time evolving equations to reduce the computational load. (Note, however, that $\mathbf{A}_1$ is very sparse.)

This chapter is in no sense exhaustive; many entire books are devoted to the material and its extensions. The intention is to lay out the fundamental ideas, which are algorithmic rearrangements of methods already described in Chapters 2 and 3, with the hope that they will permit the reader to penetrate the wider literature. Several very useful textbooks are available for readers who are not deterred by discussions in contexts differing from their own applications.[1] Most of the methods now being used in fields involving large-scale fluid dynamics, such as oceanography and meteorology, have been known for years under the general headings of control theory and control engineering. The experience in these latter areas is very helpful; the main issues in applications to fluid problems concern the size of the models

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

and data sets encountered: they are typically many orders of magnitude larger than anything contemplated by engineers. In meteorology, specialized techniques used for forecasting are commonly called "data assimilation."[2] The reader may find it helpful to keep in mind, through the details that follow, that almost all methods in actual use are, beneath the mathematical disguises, nothing but versions of least-squares fitting of models to data, but reorganized so as to increase the efficiency of solution, or to minimize storage requirements, or to accommodate continuing data streams.

Several notation systems are in wide use. The one chosen here is taken directly from the control theory literature; it is simple and adequate.[3]

4.2 Basic ideas and notation

4.2.1 Models

In the context of this chapter, "models" is used to mean statements about the connections between the system variables in some place at some time, and those in all other places and times. Maxwell's equations are a model of the behavior of time-dependent electromagnetic disturbances. These equations can be used to connect the magnetic and electric fields everywhere in space and time. Other physical systems are described by the Schrödinger, elastic, or fluid-dynamical equations. Static situations are special limiting cases, e.g., for an electrostatic field in a container with known boundary conditions.

A useful concept is that of the system "state." By that is meant the internal information at a single moment in time required to forecast the system one small time step into the future. The time evolution of a system described by the tracer diffusion equation (4.3), inside a closed container can be calculated with arbitrary accuracy at time $t + \Delta t$, if one knows $C(\mathbf{r}, t)$ and the boundary conditions $C_B(t)$, as $\Delta t \to 0$. $C\,(\mathbf{r}, t)$ is the state variable (the "internal" information), with the boundary conditions being regarded as separate externally provided variables (but the distinction is to some degree arbitrary, as we will see). In practice, because such quantities as initial and boundary conditions, container shape, etc., are obtained from measurements, and are thus always imperfectly known, the problems are conceptually identical to those already considered.

Consider any model, whether time dependent or steady, but rendered in discrete form. The "state vector" $\mathbf{x}\,(t)$ (where t is discrete) is defined as those elements of the model employed to describe fully the physical state of the system at any time and all places as required by the model in use. For the discrete Laplace/Poisson equation in Chapter 1, $\mathbf{x} = \mathrm{vec}(C_{ij})$ is the state vector. In a fluid model, the state vector might consist of three components of velocity, pressure, and temperature at

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

each of millions of grid points, and it would be a function of time, $\mathbf{x}(t)$, as well. (One might want to regard the complete description,

$$\mathbf{x}_B = [\mathbf{x}(1\Delta t)^{\mathrm{T}}, \mathbf{x}(2\Delta t)^{\mathrm{T}}, \ldots, \mathbf{x}(T\Delta t)^{\mathrm{T}}]^{\mathrm{T}}, \tag{4.7}$$

as the state vector, but by convention, it refers to the subvectors, $\mathbf{x}(t = n\Delta t)$, each of which, given the boundary conditions, is sufficient to compute any future one.)

Consider a partial differential equation,

$$\frac{\partial}{\partial t}\left(\nabla_h^2 p\right) + \beta \frac{\partial p}{\partial \eta} = 0, \tag{4.8}$$

subject to boundary conditions. ∇_h is the two-dimensional gradient operator. For the moment, t is a continuous variable. Suppose it is solved by an expansion,

$$p(\xi,\ \eta,\ t) = \sum_{j=1}^{N/2} [a_j(t)\cos(\mathbf{k}_j \cdot \mathbf{r}) + b_j(t)\sin(\mathbf{k}_j \cdot \mathbf{r})]. \tag{4.9}$$

$[\mathbf{k}_j = (k_\xi,\ k_\eta), \mathbf{r} = (\xi,\ \eta)]$, then $\mathbf{a}(t) = [a_1(t),\ b_1(t), \cdots a_j(t), b_j(t), \ldots]^{\mathrm{T}}$. The $\mathbf{k}_j$ are chosen to be periodic in the domain. The a_i, b_i are a partial-discretization, reducing the time-dependence to a finite set of coefficients. Substitute into Eq. (4.8),

$$\begin{aligned}\sum \{-|\mathbf{k}_j|^2(\dot{a}_j \cos(\mathbf{k}_j \cdot \mathbf{r}) + \dot{b}_j \sin(\mathbf{k}_j \cdot \mathbf{r}))\\ + \beta k_{1j}[-a_j \sin(\mathbf{k}_j \cdot \mathbf{r}) + b_j \cos(\mathbf{k}_j \cdot \mathbf{r})]\} = 0.\end{aligned}$$

The dot indicates a time-derivative, and k_{1j} is the η component of $\mathbf{k}_j$. Multiply this last equation through first by $\cos(\mathbf{k}_j \cdot \mathbf{r})$ and integrate over the domain:

$$-|\mathbf{k}_j|^2 \dot{a}_j + \beta k_{1j} b_j = 0.$$

Multiply by $\sin(\mathbf{k}_j \cdot \mathbf{r})$ and integrate again to give

$$|\mathbf{k}_j|^2 \dot{b}_j + \beta k_{1j} a_j = 0,$$

or

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} a_j \\ b_j \end{bmatrix} = \begin{Bmatrix} 0 & \beta k_{1j}/|\mathbf{k}_j|^2 \\ -\beta k_{1j}/|\mathbf{k}_j|^2 & 0 \end{Bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix}.$$

Each pair of a_j, b_j satisfies a system of ordinary differential equations in time, and each can be further discretized, so that

$$\begin{bmatrix} a_j\,(n\Delta t) \\ b_j\,(n\Delta t) \end{bmatrix} = \begin{Bmatrix} 1 & \Delta t \beta k_{1j}/|\mathbf{k}_j|^2 \\ -\Delta t \beta k_{1j}/|\mathbf{k}_j|^2 & 1 \end{Bmatrix} \begin{bmatrix} a_j\,((n-1)\,\Delta t) \\ b_j\,((n-1)\,\Delta t) \end{bmatrix}.$$

The state vector is then the collection

$$\mathbf{x}(n\Delta t) = [a_1(n\Delta t), b_1(n\Delta t), a_2(n\Delta t), b_2(n\Delta t), \ldots]^{\mathrm{T}},$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

at time $t = n\Delta t$. Any adequate discretization can provide the state vector; it is not unique, and careful choice can greatly simplify calculations.

In the most general terms, we can write any discrete model as a set of functional relations:

$$\begin{aligned}&\mathcal{L}[\mathbf{x}(0), \ldots, \mathbf{x}(t-\Delta t),\ \mathbf{x}(t),\ \mathbf{x}(t+\Delta t), \ldots, x(t_f), \ldots,\\ &\quad \mathbf{B}(t)\mathbf{q}(t), \mathbf{B}(t+\Delta t)\mathbf{q}(t+\Delta t), \ldots, t] = 0,\end{aligned} \tag{4.10}$$

where $\mathbf{B}(t)\mathbf{q}(t)$ represents a general, canonical, form for boundary and initial conditions/sources/sinks. A time-dependent model is a set of rules for computing the state vector at time $t = n\Delta t$, from knowledge of its values at time $t - \Delta t$ and the externally imposed forces and boundary conditions. We almost always choose the time units so that $\Delta t = 1$, and t becomes an integer (the context will normally make clear whether t is continuous or discrete). The static system equation,

$$\mathbf{Ax} = \mathbf{b}, \tag{4.11}$$

is a special case. In practice, the collection of relationships (4.10) always can be rewritten as a time-stepping rule – for example,

$$\mathbf{x}(t) = \mathbf{L}(\mathbf{x}(t-1),\ \mathbf{B}(t-1)\mathbf{q}(t-1),\ t-1), \quad \Delta t = 1, \tag{4.12}$$

or, if the model is linear,

$$\mathbf{x}(t) = \mathbf{A}(t-1)\,\mathbf{x}(t-1) + \mathbf{B}(t-1)\,\mathbf{q}(t-1). \tag{4.13}$$

If the model is time invariant, $\mathbf{A}(t) = \mathbf{A}$, and $\mathbf{B}\,(t) = \mathbf{B}$. $\mathbf{A}\,(t)$ is called the "state transition matrix." It is generally true that any linear discretized model can be put into this canonical form, although it may take some work. By the same historical conventions described in Chapter 1, solution of systems like (4.12), subject to appropriate initial and boundary conditions, constitutes the forward, or direct, problem. Note that, in general, $\mathbf{x}\,(0) = \mathbf{x}_0$ has subcomponents that formally precede $t = 0$.

Example *The straight-line model, discussed in Chapter 1 satisfies the rule*

$$\frac{\mathrm{d}^2\xi}{\mathrm{d}t^2} = 0, \tag{4.14}$$

which can be discretized as

$$\xi(t+\Delta t) - 2\xi(t) + \xi(t-\Delta t) = 0, \tag{4.15}$$

Define

$$x_1(t) = \xi(t),\ x_2(t) = \xi(t-\Delta t),$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

and $t \rightarrow n\Delta t$. One has

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t - \Delta t),$$

where

$$\mathbf{A} = \begin{Bmatrix} 2 & -1 \\ 1 & 0 \end{Bmatrix}, \tag{4.16}$$

which is of the standard form (4.13), *with* $\mathbf{B} = \mathbf{0}$. Let $\mathbf{x}(0) = [1, 0]^{\mathrm{T}}$. *Then* $\mathbf{A}\mathbf{x}(0) = \mathbf{x}(1) = [2, 1]^{\mathrm{T}}$, $\mathbf{x}(2) = [3, 2]^{\mathrm{T}}$, *etc., and the slope and intercept are both* 1. $\mathbf{x}(0) = [\xi(0), \xi(-1)]^{\mathrm{T}}$ *involves an element preceding* $t = 0$.

Example *The mass–spring oscillator satisfies the differential equation*

$$m\frac{\mathrm{d}^2\xi(t)}{\mathrm{d}t^2} + r\frac{\mathrm{d}\xi(t)}{\mathrm{d}t} + k\xi(t) = q(t),$$

where r is a damping constant. A one-sided time discretization produces

$$m(\xi(t + \Delta t) - 2\xi(t) + \xi(t - \Delta t)) + r\Delta t(\xi(t) - \xi(t - \Delta t)) + k(\Delta t)^2\xi(t) \\ = q(t)(\Delta t)^2,$$

or

$$\xi(t) = \left(2 - \frac{r\Delta t}{m} - \frac{k(\Delta t)^2}{m}\right)\xi(t - \Delta t) \\ + \left(\frac{r\Delta t}{m} - 1\right)\xi(t - 2\Delta t) + (\Delta t)^2\frac{q(t - \Delta t)}{m}, \tag{4.17}$$

which is

$$\begin{bmatrix} \xi(t) \\ \xi(t - \Delta t) \end{bmatrix} = \begin{Bmatrix} 2 - \frac{r}{m}\Delta t - \frac{k}{m}(\Delta t)^2 & \frac{r\Delta t}{m} - 1 \\ 1 & 0 \end{Bmatrix} \begin{bmatrix} \xi(t - \Delta t) \\ \xi(t - 2\Delta t) \end{bmatrix} \\ + \begin{bmatrix} (\Delta t)^2\,\frac{q(t - \Delta t)}{m} \\ 0 \end{bmatrix}, \tag{4.18}$$

and is the canonical form with $\mathbf{A}$ *independent of time, where*

$$\mathbf{x}(t) = [\xi(t) \quad \xi(t - \Delta t)]^{\mathrm{T}}, \qquad \mathbf{B}(t)\mathbf{q}(t) = [(\Delta t)^2\, q(t)/m \quad 0]^{\mathrm{T}}.$$

Example *A difference equation important in time-series analysis*[4] *is*

$$\xi(t) + a_1\xi(t - 1) + a_2\xi(t - 2) + \cdots + a_N\xi(t - N) = \eta(t), \tag{4.19}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005

Cambridge Books Online © Cambridge University Press, 2009

where $\eta(t)$ is a zero-mean, white-noise process (Eq. (4.19) is an example of an autoregressive process (AR)). To put this into the canonical form, write[5]

$$\begin{aligned} x_1(t) &= \xi(t-N),\\ x_2(t) &= \xi(t-N+1),\\ &\vdots\\ x_N(t) &= \xi(t-1),\\ x_N(t+1) &= -a_1\,x_N(t) - a_2\,x_{N-1}(t)\cdots - a_N\,x_1(t) + \eta(t). \end{aligned}$$

It follows that $x_1(t+1) = x_2(t)$, etc., or

$$\mathbf{x}(t) = \left\{\begin{matrix} 0 & 1 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & \cdots & 0 & 0\\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot\\ -a_N & -a_{N-1} & -a_{N-2} & \cdots & -a_2 & -a_1 \end{matrix}\right\}\mathbf{x}(t-1) + \begin{bmatrix}0\\0\\ \cdot\\1\end{bmatrix}\eta(t-1). \quad (4.20)$$

A *is known as a "companion" matrix. Here,* $\mathbf{B}(t) = [0 \quad 0 \quad \cdot \quad 1]^{\mathrm{T}}$, $\mathbf{q}(t) = \eta(t)$.

Given that most time-dependent models can be written as in (4.12) or (4.13), the forward-model solution involves marching forward from known initial conditions at $t = 0$, subject to specified boundary values. So, for example, the linear model (4.13), with given initial conditions $\mathbf{x}(0) = \mathbf{x}_0$, involves the sequence

$$\begin{aligned} \mathbf{x}(1) &= \mathbf{A}(0)\,\mathbf{x}_0 + \mathbf{B}(0)\,\mathbf{q}(0),\\ \mathbf{x}(2) &= \mathbf{A}(1)\,\mathbf{x}(1) + \mathbf{B}(1)\,\mathbf{q}(1),\\ &= \mathbf{A}(1)\,\mathbf{A}(0)\,\mathbf{x}_0 + \mathbf{A}(1)\,\mathbf{B}(0)\,\mathbf{q}(0) + \mathbf{B}(1)\,\mathbf{q}(1),\\ &\vdots\\ \mathbf{x}(t_f) &= \mathbf{A}(t_f-1)\,\mathbf{x}(t_f-1) + \mathbf{B}(t_f-1)\,\mathbf{q}(t_f-1)\\ &= \mathbf{A}(t_f-1)\,\mathbf{A}(t_f-2)\cdots\mathbf{A}(0)\,\mathbf{x}_0 + \cdots. \end{aligned}$$

Most of the basic ideas can be understood in the notationally simplest case of time-independent **A**, **B**, and that is usually the situation we will address with little loss of generality, so that $\mathbf{A}(t)\,\mathbf{A}(t-1) = \mathbf{A}^2$, etc. Figure 4.1 depicts the time history for the mass–spring oscillator, with the parameter choice $\Delta t = 1$, $k = 0.1$, $m = 1$, $r = 0$, so that

$$\mathbf{A} = \left\{\begin{matrix} 1.9 & -1\\ 1 & 0\end{matrix}\right\}, \qquad \mathbf{B}\mathbf{q}(t) = \begin{bmatrix}1\\0\end{bmatrix} u(t),$$

where $\langle u(t)^2\rangle = 1$, a random variable. The initial conditions were $\mathbf{x}(0) = [\xi(0) \;\; \xi(-1)]^{\mathrm{T}}$.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

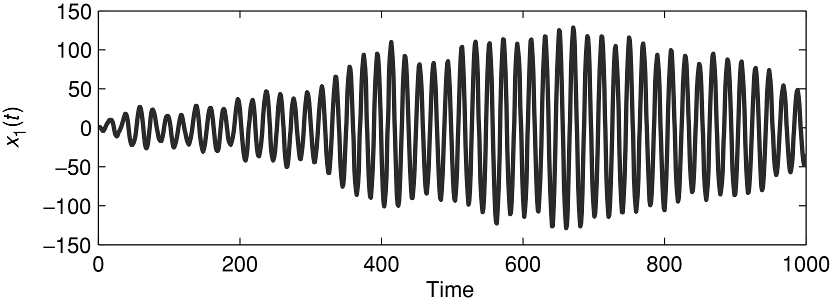

Figure 4.1 Time history of $x_1(t)$ for the linear oscillator with $\Delta t = 1, k = 0.1, m = 1, r = 0$ driven by a random sequence of zero mean and unit variance. Note the buildup in amplitude from the accumulating uncorrelated forcing increments.

It is important to recognize that this time-stepping procedure cannot be used if some of the elements of the initial conditions, $\mathbf{x}(0)$, are replaced, e.g., with elements of $\mathbf{x}(t_f)$, or more generally with elements of $\mathbf{x}(t)$ for arbitrary t. That is, the amount of information may be the same, and fully adequate, but not useful in straightforward time-stepping. Many of the algorithms developed here are directed at these less-conventional cases.

$\mathbf{A}$ is necessarily square. It is also often true that $\mathbf{A}^{-1}$ exists, and, if not, a generalized inverse can be used. If $\mathbf{A}^{-1}$ can be computed, one can contemplate the possibility (important later) of running a model backward in time, for example as

$$\mathbf{x}(t-1) = \mathbf{A}^{-1}\mathbf{x}(t) - \mathbf{A}^{-1}\mathbf{B}(t-1)\,\mathbf{q}(t-1).$$

Such a computation may be inaccurate if carried on for long times, but the same may well be true of the forward model.

Some attention must be paid to the structure of $\mathbf{B}(t)\,\mathbf{q}(t)$. The partitioning into these elements is not unique and can be done to suit one's convenience. The dimension of $\mathbf{B}$ is that of the size of the state vector by the dimension of $\mathbf{q}$, which typically would reflect the number of independent degrees of freedom in the forcing/boundary conditions. ("Forcing" is hereafter used to include boundary conditions, sources and sinks, and anything normally prescribed externally to the model.) Consider the model grid points displayed in Fig. 4.2. Suppose that the boundary grid points are numbered 1–5, 6, 10, 46–50, and all others are interior. If there are no interior forces, and all boundary values have a time history $q(t)$, then we could take

$$\mathbf{B} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \cdots 1 \quad 1]^{\mathrm{T}}, \tag{4.21}$$

where the ones occur at the boundary points, and the zeros at the interior ones.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005

Cambridge Books Online © Cambridge University Press, 2009

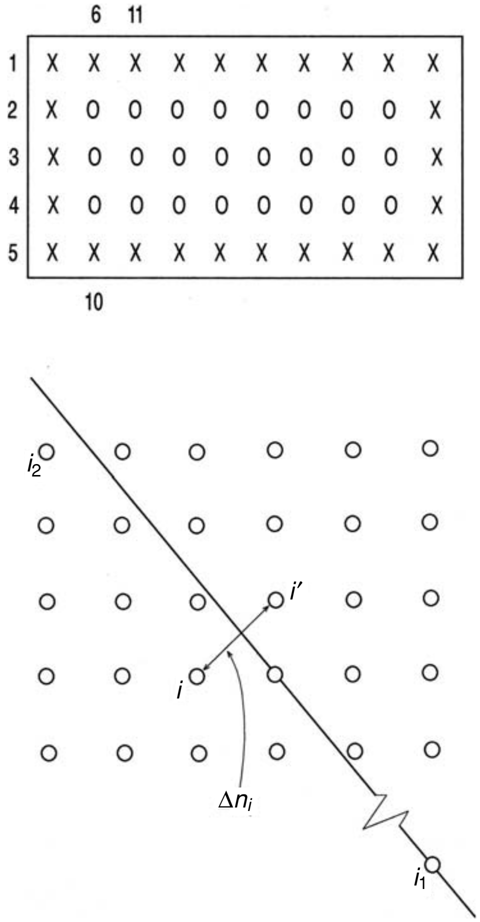

Figure 4.2 (a) Simple numerical grid for use of discrete form of model; × denote boundary grid points, and o are interior ones. Numbering is sequential down the columns, as shown. (b) Tomographic integral is assumed given between i_1, i_2, and the model values at the grid points would be used to calculate its predicted value.

Suppose, instead, that boundary grid point 2 has values $q_1(t)$, interior point 7 has a forcing history $q_2(t)$, and all others are unforced, then

$$\mathbf{Bq}(t) = \begin{Bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdot & 0 \end{Bmatrix}^{\mathrm{T}} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}. \tag{4.22}$$

A time-dependent **B** would correspond to time-evolving positions at which forces were prescribed – a somewhat unusual situation. It would be useful, for example, if one were driving a fluid model with a heat flux or stress in the presence of a prescribed moving ice cover. One could also impose initial conditions using a time-dependent $\mathbf{B}(t)$, which would vanish after $t = 0$.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

As with steady models, care is needed in understanding the propagation of errors in time and space. If we have some knowledge of the initial oceanic state, $\tilde{\mathbf{x}}(0)$, and are doing an experiment at a later time t, the prior information – the estimated initial conditions – carries information in addition to what is currently being measured. We seek to combine the two sets of information. How does information propagate forward in time? Formally, the rule (4.12) tells us exactly what to do. But because there are always errors in $\tilde{\mathbf{x}}(0)$, we need to be careful about assuming that a model computation of $\tilde{\mathbf{x}}(t)$ is useful. Depending upon the details of the model, the behavior of the errors through time can be distinguished: (1) The model has decaying components. If the amplitudes of these components are partially erroneous, then for large enough t, these elements will have diminished, perhaps to the point where they are negligible. (2) The model has neutral components. At time t, the erroneous elements have amplitudes the same as they were at $t = 0$. (3) The model has unstable components; at time t any erroneous parts may have grown to swamp everything else.

Realistic models, particularly fluid ones, can contain all three types of behavior simultaneously. It thus becomes necessary to determine which of the elements of the forecast $\tilde{\mathbf{x}}(t)$ can be used to help estimate the system state by combination with new data, and which elements should be suppressed as partially or completely erroneous. Simply assuming that all components are equally accurate can be a disastrous recipe.

Before proceeding, we reiterate the point that time need not be accorded a privileged position. Form the inclusive state vector, $\mathbf{x}_B$, defined in Eq. (4.7). Then, as in Eq. (4.6), models of the form (4.13) can be written in the "whole-domain" form,

$$\mathbf{A}_B \mathbf{x}_B = \mathbf{d}_B$$

$$\mathbf{A}_B = \left\{\begin{matrix} -\mathbf{A} & \mathbf{I} & \mathbf{0} & \cdot & \cdot & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{I} & \mathbf{0} & \cdot & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & -\mathbf{A} & \mathbf{I} \end{matrix}\right\}, \quad \mathbf{d}_B = \begin{bmatrix} \mathbf{Bq}(0) \\ \mathbf{Bq}(1) \\ \vdots \end{bmatrix}, \tag{4.23}$$

plus initial conditions, which is no different, except for its possibly enormous size, from that of a static system and can be handled by any of the methods of earlier chapters if the computational capacity is sufficient. If time-stepping is impossible because the initial condition is replaced by knowledge of $\mathbf{x}(t')$, $t' \neq 0$, the whole-domain form may be very attractive. Note the block-banded, sparse, nature of $\mathbf{A}_B$.

4.2.2 How to find the matrix $\mathbf{A}(t)$

Most modern large-scale time-evolving models, even if completely linear, are written in computer code, typically in languages such as Fortran90 or C/C++. The

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005
Cambridge Books Online © Cambridge University Press, 2009

state transition matrix is not normally explicitly constructed; instead, the individual elements of $x_i(t)$ are time-stepped to produce the $x_j(t+1)$, $\Delta t = 1$, usually using various vectorizations. $\mathbf{A}(t)$ is often neither required nor constructed, as all one cares about is the result of its operation on $\mathbf{x}(t)$, as generated from the model code. If one requires an explicit $\mathbf{A}(t)$ but has only the forward code, several methods can be used. For simplicity let $\mathbf{Bq}(t) = \mathbf{0}$ (the more general approach is obvious).

(1) Solve Eq. (4.13) N times, starting at time $t = 0$, subject to $\mathbf{x}^{(i)}(0) =$ column i of $\mathbf{I}$ – that is, the model is stepped forward for N different initial conditions corresponding to the N different problems of unit initial condition at a single grid or boundary point, with zero initial conditions everywhere else. Let each column of $\mathbf{G}(t, 0)$ correspond to the appropriate value of $\mathbf{x}(t)$ – that is,

$$\begin{aligned}
\mathbf{G}(0,0) &= \mathbf{I},\\
\mathbf{G}(1,0) &= \mathbf{A}(0)\mathbf{G}(0,0),\\
\mathbf{G}(2,0) &= \mathbf{A}(1)\mathbf{G}(1,0) = \mathbf{A}(1)\mathbf{A}(0),\\
&\vdots\\
\mathbf{G}(t,0) &= \mathbf{A}(t-1)\mathbf{A}(t-2)\cdots\mathbf{A}(0).
\end{aligned}$$

We refer to $\mathbf{G}(t, 0)$ as a *unit solution*; it is closely related to the Green function discussed in Chapter 2. The solution for arbitrary initial conditions is then

$$\mathbf{x}(t) = \mathbf{G}(t,0)\mathbf{x}(0), \tag{4.24}$$

and the modification for $\mathbf{Bq} \neq 0$ is straightforward. $\mathbf{A}(t)$ can be readily reconstructed from $\mathbf{G}(t, 0)$, most simply if $\mathbf{A}$ is time-independent and if one time-step is numerically accurate enough to represent $\mathbf{G}$. Otherwise, multiple time steps can be used until a sufficiently large change in $\mathbf{G}$ is produced.

Several other methods exist to obtain $\mathbf{A}$ from an existing computer model, but consider now only the case of a steady model, with no time-dependence in the governing matrices ($\mathbf{A}$, $\mathbf{B}$). We continue to simplify by setting $\mathbf{B} = \mathbf{0}$.

(2) Define N independent initial condition vectors $\mathbf{x}_0^{(i)}$, $i = 1, 2, \ldots, N$, and form a matrix,

$$\mathbf{X}_0 = \left\{\mathbf{x}_0^{(i)}\right\}.$$

Time-step the model once, equivalent to

$$\mathbf{X}_1 = \mathbf{A}\mathbf{X}_0,$$

and invert $\mathbf{X}_0$:

$$\mathbf{A} = \mathbf{X}_1\mathbf{X}_0^{-1}. \tag{4.25}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005

Cambridge Books Online © Cambridge University Press, 2009

The inverse will exist by the assumption of independence (a basis) in the initial condition vectors. One must again run the model N times in this approach. If $\mathbf{x}_0^{(i)} = \delta_{ij}$, $\mathbf{X}_1 = \mathbf{G}$.

Again, the changes from $\mathbf{X}_0$ to $\mathbf{X}_1$ may be too small for adequate numerical accuracy, and one might use multiple time-steps, computing, for time-independent $\mathbf{A}$,

$$\mathbf{X}_n = \mathbf{A}^n \mathbf{X}_0,$$

which would determine $\mathbf{A}^n$, and $\mathbf{A}$ itself can be found by one of the matrix root algorithms,[6] or by redefining unit time to $n\Delta t$. One might also define $\mathbf{A}^n$ as the average value of $\mathbf{A}\,(n)\,\mathbf{A}\,(n-1)\cdots\mathbf{A}\,(0)\,.$

(3) Suppose the statistics of the solutions are known, e.g.,

$$\mathbf{R}(0) = \langle \mathbf{x}(t)\mathbf{x}(t)^{\mathrm{T}} \rangle, \;\; \mathbf{R}(1) = \langle \mathbf{x}(t+1)\mathbf{x}(t)^{\mathrm{T}} \rangle,$$

perhaps because the model has been run many times from different initial conditions – making it possible to estimate these from stored output. Then we note that

$$\langle \mathbf{x}(t+1)\mathbf{x}(t)^{\mathrm{T}} \rangle = \mathbf{A}\langle \mathbf{x}(t)\mathbf{x}(t)^{\mathrm{T}} \rangle,$$

or

$$\mathbf{R}(1) = \mathbf{A}\mathbf{R}(0),$$

and

$$\mathbf{A} = \mathbf{R}(1)\mathbf{R}(0)^{-1}. \tag{4.26}$$

That is to say, knowledge of these covariances is equivalent to knowledge of the model itself (and vice versa).[7] Multiple time steps can again be used if necessary to infer that $\mathbf{A}^n = \mathbf{R}(n)\mathbf{R}(0)^{-1}$. By writing $\langle \mathbf{x}(t+1)\mathbf{x}(t)^{\mathrm{T}} \rangle = \mathbf{R}(1)$, etc., stationarity is implied. More generally, one may have $\langle \mathbf{x}(t+1)\mathbf{x}(t)^{\mathrm{T}} \rangle = \mathbf{R}(t,1)$. $\mathbf{R}(0)$ must be non-singular.

Note that determination of $\mathbf{B}$ can be done analogously – using a spanning set of $\mathbf{q}^{(i)}$ as initial conditions, setting $\mathbf{x}\,(0) = 0$.

(4) Automatic/algorithmic differentiation (AD) tools exist[8] that can take computer code (e.g., Fortran, C, Matlab®) for the forward model, and produce by analysis of the code, equivalent computer code (e.g., Fortran) for construction of $\mathbf{A}$. Some codes preferentially produce $\mathbf{A}^{\mathrm{T}}$, but transposition then can be employed. An example is provided in the Appendix to this chapter.

If the model is fully time-dependent, then $\mathbf{A}\,(t)$ has to be deduced at each time-step, as above. For some purposes, one might seek temporal averages, defining $\bar{\mathbf{A}}$ as

$$\bar{\mathbf{A}}^n = \mathbf{A}(n-1)\mathbf{A}(n-2)\cdots\mathbf{A}(1)\mathbf{A}(0).$$

Reintroduction of $\mathbf{B}$ is easily accommodated.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

4.2.3 Observations and data

Here, observations are introduced into the modeling discussion so that they stand on an equal footing with the set of model equations (4.12) or (4.13). Observations will be represented as a set of linear simultaneous equations at time $t = n\Delta t$,

$$\mathbf{E}(t)\mathbf{x}(t) + \mathbf{n}(t) = \mathbf{y}(t), \tag{4.27}$$

which is a straightforward generalization of the previous static systems where t did not appear explicitly; here, $\mathbf{E}$ is sometimes called the "design" or "observation" matrix. The notation used in Chapter 2 to discuss recursive estimation was chosen deliberately to be the same as used here.

The requirement that the observations be linear combinations of the state-vector elements can be relaxed if necessary, but most common observations are of that form. An obvious exception would be the situation in which the state vector included fluid velocity components, $u(t)$, $v(t)$, but an instrument measuring speed, $\sqrt{(u(t)^2 + v(t)^2)}$, would produce a non-linear relation between $y_i(t)$ and the state vector. Such systems are usually handled by some form of linearization.[9]

To be specific, the noise $\mathbf{n}(t)$ is supposed to have zero mean and known second-moment matrix

$$\langle \mathbf{n}(t) \rangle = 0, \qquad \langle \mathbf{n}(t)\mathbf{n}(t)^{\mathrm{T}} \rangle = \mathbf{R}(t). \tag{4.28}$$

But

$$\langle \mathbf{n}(t)\mathbf{n}(t')^{\mathrm{T}} \rangle = \mathbf{0}, \quad t \neq t'. \tag{4.29}$$

That is, the observational noise should not be correlated from one measurement time to another; there is a considerable literature on how to proceed when this crucial assumption fails (called the "colored-noise" problem[10]). Unless specifically stated otherwise, we will assume that Eq. (4.29) is valid.

The matrix $\mathbf{E}(t)$ can accommodate almost any form of linear measurement. If, at some time, there are no measurements, then $\mathbf{E}(t)$ vanishes, along with $\mathbf{R}(t)$. If a single element $x_i(t)$ is measured, then $\mathbf{E}(t)$ is a row vector that is zero everywhere except in column i, where it is 1. It is particularly important to recognize that many measurements are weighted averages of the state-vector elements. Some measurements – for example, tomographic ones[11] as described in Chapter 1 – are explicitly spatial averages (integrals) obtained by measuring some property along a ray traveling between two points (see Fig. 4.2). Any such data representing spatially filtered versions of the state vector can be written as

$$y(t) = \sum \alpha_j x_j(t), \tag{4.30}$$

where the α_j are the averaging weights.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005

Point observations often occur at positions not coincident with model grid positions (although many models, e.g., spectral ones, do not use grids). Then (4.27) is an interpolation rule, possibly either very simple or conceivably a full-objective mapping calculation, of the value of the state vector at the measurement point. Often the number of model grid points vastly exceeds the number of the data grid points; thus, it is convenient that the formulation (4.27) requires interpolation from the dense model grid to the sparse data positions (see Fig. 4.2). (In the unusual situation where the data density is greater than the model grid density, one can restructure the problem so the interpolation goes the other way.) More complex filtered measurements exist. In particular, one may have measurements of a state vector only in specific wavenumber bands; but such "band-passed" observations are automatically in the form (4.27).

As with the model (4.23), the observations of the combined state vector can be concatenated into a single observational set,

$$\mathbf{E}_B\mathbf{x}_B + \mathbf{n}_B = \mathbf{y}_B, \tag{4.31}$$

where

$$\mathbf{E}_B = \left\{\begin{matrix} \mathbf{I} & 0 & 0 & \cdot & 0 \\ 0 & \mathbf{E}(1) & 0 & \cdot & 0 \\ 0 & 0 & \mathbf{E}(2) & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \mathbf{E}(t_f) \end{matrix}\right\}, \quad \mathbf{n}_B = \begin{bmatrix} \mathbf{n}(0) \\ \mathbf{n}(1) \\ \vdots \\ \mathbf{n}(t_f) \end{bmatrix}, \quad \mathbf{y}_B = \begin{bmatrix} \tilde{\mathbf{x}}(0) \\ \mathbf{y}(1) \\ \vdots \\ \mathbf{y}(t_f) \end{bmatrix}.$$

Here the initial conditions have been combined with the observations. $\mathbf{E}_B$ is block-banded and sparse. If the size is no problem, the combined and concatenated model and observations could be dealt with using any of the methods of Chapter 2. The rest of this chapter can be thought of as an attempt to produce from the model/data combination the same type of estimates as were found useful in Chapter 2, but exploiting the special structure of matrices $\mathbf{A}_B$ and $\mathbf{E}_B$ so as to avoid having to store them all at once in the computer.

As one example of how the combined model and observation equations can be used together, consider the situation in which only the initial conditions $\mathbf{x}(0)$ are unknown. The unit solution formulation of p. 188 leads to a particularly simple reduced form. One has immediately,

$$\mathbf{y}(t) = \mathbf{E}(t)\mathbf{G}(t,0)\mathbf{x}(0) + \mathbf{n}(t), \quad t = 1, 2, \ldots, t_f, \tag{4.32}$$

which is readily solved in whole-domain form for $\mathbf{x}(0)$. If only a subset of the $\mathbf{x}(0)$ are thought to be non-zero, then the columns of $\mathbf{G}$ need to be computed only for those elements.[12]

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

4.3 Estimation

4.3.1 Model and data consistency

In many scientific fields, the central issue is to develop understanding from the combination of a skillful model with measurements. The model is intended to encompass all one's theoretical knowledge about how the system behaves, and the data are the complete observational knowledge of the same system. If done properly, and in the absence of contradictions between theory and experiment, inferences from the model/data combination should be no worse, and may well be very much better than those made from either alone. It is the latter possibility that motivates the development of state estimation procedures. "Best-estimates" made by combining models with observations are often used to forecast a system (e.g., to land an airplane), but this is by no means the major application.

Such model/data problems are ones of statistical inference with a host of specific subtypes. Some powerful techniques are available, but like any powerful tools (a chain saw, for example), they can be dangerous to the user! In general, one is confronted with a two-stage problem. Stage 1 involves developing a suitable model that is likely to be consistent with the data. "Consistent" means that, within the estimated data errors, the model is likely to be able to describe the features of interest. Obtaining the data errors is itself a serious modeling problem. Stage 2 produces the actual estimate with its error estimates.

One can go very badly wrong at stage 1, before any computation takes place. If elastic wave propagation is modeled using the equations of fluid dynamics, estimation methods will commonly produce some kind of "answer," but one that would be nonsensical. Model failure can, of course, be much more subtle, in which some omitted, supposed secondary element (e.g., a time-dependence) proves to be critical to a description of the data. Good technique alerts users to the presence of such failures, along with clues as to what should be changed in the model. But these issues do not, however, apply only to the model. The assertion that a particular data set carries a signal of a particular kind can prove to be false in a large number of ways. A temperature signal thought to represent the seasonal cycle might prove, on careful examination, to be dominated by higher or lower frequency structures, and thus its use with an excellent model of annual variation might prove disastrous. Whether this situation is to be regarded as a data or as a model issue is evidently somewhat arbitrary.

Thus stage 1 of any estimation problem has to involve understanding of whether the data and the model are physically and statistically consistent. If they are not, one should stop and reconsider. Often where they are believed to be generally consistent up to certain quantitative adjustments, one can combine the two stages. A model may have adjustable parameters (turbulent mixing coefficients, boundary

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

condition errors, etc.) that could bring the model and data into consistency, in which case the estimation procedure becomes, in part, an attempt to find those parameters in addition to the state. Alternatively, the data error covariance, $\mathbf{R}(t)$, may be regarded as incompletely known, and one might seek, as part of the state estimation procedure, to improve one's estimate of it. (Problems like this one fall under the subject of "adaptive filtering.")

Assuming for now that the model and data are likely to prove consistent, one can address what might be thought of as a form of interpolation: given a set of observations in space and time as described by Eq. (4.27), use the dynamics as described by the model (4.12) or (4.13) to estimate various state-vector elements at various times of interest. Yet another, less familiar, problem recognizes that some of the forcing terms $\mathbf{B}(t-1)\mathbf{q}(t-1)$ are partially or wholly unknown (e.g., the wind stress boundary conditions over the ocean are imperfectly known), and one might seek to estimate them from whatever ocean observations are available and from the known model dynamics.

The forcing terms – representing boundary conditions as well as interior sources/sinks and forces – almost always need to be divided into two parts: the known and the unknown. The latter will often be perturbations about the known values. Thus, rewrite (4.13) in the modified form

$$\mathbf{x}(t) = \mathbf{A}(t-1)\mathbf{x}(t-1) + \mathbf{B}(t-1)\mathbf{q}(t-1) + \boldsymbol{\Gamma}(t-1)\mathbf{u}(t-1), \quad \Delta t = 1, \tag{4.33}$$

where now $\mathbf{B}(t)\mathbf{q}(t)$ represent the known forcing terms and $\boldsymbol{\Gamma}(t)\mathbf{u}(t)$ the unknown ones, which we will generally refer to as the "controls," or "control terms." $\boldsymbol{\Gamma}(t)$ is known and plays the same role for $\mathbf{u}(t)$ as does $\mathbf{B}(t)$ for $\mathbf{q}(t)$. Usually $\mathbf{B}(t)$, $\boldsymbol{\Gamma}(t)$ will be treated as time independent, but this simplification is not necessary. Almost always, we can make some estimate of the size of the control terms, as, for example,

$$\langle \mathbf{u}(t) \rangle = \mathbf{0}, \qquad \langle \mathbf{u}(t)\mathbf{u}(t)^{\mathrm{T}} \rangle = \mathbf{Q}(t). \tag{4.34}$$

The controls have a second, somewhat different, role: they can also represent the model error. All models are inaccurate to a degree – approximations are always made to the equations describing any particular physical situation. One can expect that the person who constructed the model has some idea of the size and structure of the physics or chemistry, or biology, etc. that have been omitted or distorted in the model construction. In this context, $\mathbf{Q}(t)$ represents the covariance of the model error, and the control terms represent the missing physics. The assumption $\langle \mathbf{u}(t) \rangle = \mathbf{0}$ must be critically examined in this case, and, in the event of failure, some modification of the model must be made or the control variance artificially modified to include what is a model bias error. But the most serious problem is

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

that models are rarely produced with *any* quantitative description of their accuracy beyond one or two examples of comparison with known solutions. One is left to determine $\mathbf{Q}(t)$ by guesswork. Getting beyond such guesses is again a problem of adaptive estimation.

Collecting the standard equations of model and data:

$$\boxed{\begin{array}{ll}
\mathbf{x}(t)= \mathbf{A}(t-1)\mathbf{x}(t-1)+\mathbf{B}\mathbf{q}(t-1)+\boldsymbol{\Gamma}\mathbf{u}(t-1),\ t=1,2,\ldots,t_f, & (4.35) \\
\mathbf{E}(t)\mathbf{x}(t)+\mathbf{n}(t)=\mathbf{y}(t), \quad t=1,2,\ldots,t_f, \quad \Delta t=1, & (4.36) \\
\mathbf{n}(t)= \mathbf{0}, \quad \langle \mathbf{n}(t)\mathbf{n}(t)^{\mathrm{T}}\rangle=\mathbf{R}(t), \quad \langle \mathbf{n}(t)\mathbf{n}(t')^{T}\rangle=\mathbf{0},\ t\neq t', & (4.37) \\
\langle \mathbf{u}(t)\rangle= \mathbf{0}, \quad \langle \mathbf{u}(t)\mathbf{u}(t)^{T}\rangle= \mathbf{Q}(t), & (4.38) \\
\tilde{\mathbf{x}}(0)= \mathbf{x}_0, \quad \langle(\tilde{\mathbf{x}}(0)-\mathbf{x}(0))(\tilde{\mathbf{x}}(0)-\mathbf{x}(0))^{\mathrm{T}}\rangle=\mathbf{P}(0), & (4.39)
\end{array}}$$

where t_f defines the endpoint of the interval of interest. The last equation, (4.39), treats the initial conditions of the model as a special case – the uncertain initialization problem, where $\mathbf{x}(0)$ is the true initial condition and $\tilde{\mathbf{x}}(0)=\mathbf{x}_0$ is the value actually used but with uncertainty $\mathbf{P}(0)$. Alternatively, one could write

$$\mathbf{E}(0)\mathbf{x}(0)+\mathbf{n}(0)=\mathbf{x}_0,\ \mathbf{E}(0)= \mathbf{I},\ \langle \mathbf{n}(0)\mathbf{n}(0)^{\mathrm{T}}\rangle=\mathbf{P}(0), \tag{4.40}$$

and include the initial conditions as a special case of the observations – recognizing explicitly that initial conditions are often obtained that way. (Compare Eq. (4.31).)

This general form permits one to grapple with reality. In the spirit of ordinary least-squares and its intimate cousin, minimum-error variance estimation, consider the general problem of finding state vectors and controls, $\mathbf{u}(t)$, that minimize an objective function,

$$\begin{aligned}
J = {} & [\mathbf{x}(0)-\mathbf{x}_0]^{\mathrm{T}}\mathbf{P}(0)^{-1}[\mathbf{x}(0)-\mathbf{x}_0] \\
& + \sum_{t=1}^{t_f}[\mathbf{E}(t)\mathbf{x}(t)-\mathbf{y}(t)]^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{E}(t)\mathbf{x}(t)-\mathbf{y}(t)] \\
& + \sum_{t=0}^{t_f-1}\mathbf{u}(t)^{\mathrm{T}}\mathbf{Q}(t)^{-1}\mathbf{u}(t),
\end{aligned} \tag{4.41}$$

subject to the model, Eqs. (4.35), (4.38) and (4.39). As written here, this choice of an objective function is somewhat arbitrary but perhaps reasonable as the direct analogue to those used in Chapter 2. It seeks a state vector $\mathbf{x}(t),\ t=0,1,\ldots,t_f$, and a set of controls, $\mathbf{u}(t),\ t=0,1,\ldots,t_f-1$, that satisfy the model and that agree with the observations to an extent determined by the weight matrices $\mathbf{R}(t)$ and $\mathbf{Q}(t)$, respectively. From the previous discussions of least-squares and minimum-error variance estimation, the minimum-square requirement, Eq. (4.41), will produce a solution identical to that derived from minimum variance estimation by the specific

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

choice of the weight matrices as the corresponding prior uncertainties, $\mathbf{R}(t)$, $\mathbf{Q}(t)$, $\mathbf{P}(0)$. In a Gaussian system, it also proves to be the maximum likelihood estimate. The introduction of the controls, $\mathbf{u}(t)$, into the objective function represents an acknowledgment that arbitrarily large controls (forces) would not usually be an acceptable solution; they should be consistent with $\mathbf{Q}(t)$.

Note on notation As in Chapter 2, any values of $\mathbf{x}(t)$, $\mathbf{u}(t)$ minimizing J will be written $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{u}}(t)$ and these symbols sometimes will be substituted into Eq. (4.41) if it helps their clarity.

Much of the rest of this chapter is directed at solving the problem of finding the minimum of J subject to the solution satisfying the model. Notice that J involves the state vector, the controls, and the observations over the entire time period under consideration, $t = 0, 1, \ldots, t_f$. This type of objective function is the one usually of most interest to scientists attempting to understand their system – in which data are stored and employed over a finite time. In some other applications, most notably forecasting, which is taken up immediately below, one has only the past measurements available; this situation proves to be a special case of the more general one.

Although we will not keep repeating the warning each time an objective function such as Eq. (4.41) is encountered, the reader is reminded of a general message from Chapter 2: *The assumption that the model and observations are consistent and that the minimum of the objective function produces a meaningful and useful estimate must always be tested after the fact.* That is, at the minimum of J, $\tilde{\mathbf{u}}(t)$ must prove consistent with $\mathbf{Q}(t)$, and $\tilde{\mathbf{x}}(t)$ must produce residuals consistent with $\mathbf{R}(t)$. Failure of these and other posterior tests should lead to rejection of the model. As always, one can thus reject a model (which includes $\mathbf{Q}(t)$, $\mathbf{R}(t)$) on the basis of a failed consistency with observations. But a model is never "correct" or "valid," merely "consistent." (See Note 8, Chapter 1.)

4.3.2 The Kalman filter

We begin with a special case. Suppose that by some means, at time $t = 0$, $\Delta t = 1$, we have an unbiassed estimate, $\tilde{\mathbf{x}}(0)$, of the state vector with uncertainty $\mathbf{P}(0)$. At time $t = 1$, observations from Eq. (4.36) are available. How would the information available best be used to estimate $\mathbf{x}(1)$?

The model permits a forecast of what $\mathbf{x}(1)$ should be, were $\tilde{\mathbf{x}}(0)$ known perfectly,

$$\tilde{\mathbf{x}}(1, -) = \mathbf{A}\tilde{\mathbf{x}}(0) + \mathbf{B}\mathbf{q}(0), \tag{4.42}$$

where the unknown control terms have been replaced by the best estimate we can make of them – their mean, which is zero, and $\mathbf{A}$ has been assumed to be time independent. A minus sign has been introduced into the argument of $\tilde{\mathbf{x}}(1, -)$ to

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

show that *no data at* $t = 1$ *have yet been used to make the estimate* at $t = 1$, in a notation we will generally use. How good is this forecast?

Suppose the erroneous components of $\tilde{\mathbf{x}}(0)$ are

$$\gamma(0) = \tilde{\mathbf{x}}(0) - \mathbf{x}(0), \tag{4.43}$$

then the erroneous components of the forecast are

$$\begin{aligned}\gamma(1) \equiv \tilde{\mathbf{x}}(1,-) - \mathbf{x}(1) &= \mathbf{A}\tilde{\mathbf{x}}(0) + \mathbf{B}\mathbf{q}(0) - (\mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{q}(0) + \boldsymbol{\Gamma}\mathbf{u}(0)) \\ &= \mathbf{A}\gamma(0) - \boldsymbol{\Gamma}\mathbf{u}(0),\end{aligned} \tag{4.44}$$

that is, composed of two distinct elements: the propagated erroneous portion of $\tilde{\mathbf{x}}(0)$, and the unknown control term. Their second moments are

$$\begin{aligned}\langle\gamma(1)\,\gamma(1)^{\mathrm{T}}\rangle &= \langle(\mathbf{A}\gamma(0) - \boldsymbol{\Gamma}\mathbf{u}(0))(\mathbf{A}\gamma(0) - \boldsymbol{\Gamma}u(0))^{\mathrm{T}}\rangle \\ &= \mathbf{A}\langle\gamma(0)\mathbf{P}(0)\,\gamma(0)^{\mathrm{T}}\rangle\mathbf{A}^{\mathrm{T}} + \boldsymbol{\Gamma}\langle\mathbf{u}(0)\,\mathbf{u}(0)^{\mathrm{T}}\rangle\boldsymbol{\Gamma}^{\mathrm{T}} \\ &= \mathbf{A}\mathbf{P}(0)\mathbf{A}^{\mathrm{T}} + \boldsymbol{\Gamma}\mathbf{Q}(0)\boldsymbol{\Gamma}^{\mathrm{T}} \\ &\equiv \mathbf{P}(1,-),\end{aligned} \tag{4.45}$$

by the definitions of $\mathbf{P}(0)$, $\mathbf{Q}(0)$ and the assumption that the unknown controls are not correlated with the error in the state estimate at $t = 0$. We now have an estimate of $\mathbf{x}(1)$ with uncertainty $\mathbf{P}(1,-)$ and a set of observations,

$$\mathbf{E}(1)\,\mathbf{x}(1) + \mathbf{n}(1) = \mathbf{y}(1). \tag{4.46}$$

To combine the two sets of information, use the recursive least-squares solution from Eqs. (2.434)–(2.436). By assumption, the uncertainty in $\mathbf{y}(1)$ is uncorrelated with that in $\tilde{\mathbf{x}}(1,-)$. Making the appropriate substitutions into those equations,

$$\begin{aligned}\tilde{\mathbf{x}}(1) &= \tilde{\mathbf{x}}(1,-) + \mathbf{K}(1)\,[\mathbf{y}(1) - \mathbf{E}(1)\,\tilde{\mathbf{x}}(1,-)], \\ \mathbf{K}(1) &= \mathbf{P}(1,-)\,\mathbf{E}(1)^{\mathrm{T}}[\mathbf{E}(1)\,\mathbf{P}(1,-)\,\mathbf{E}(1)^{\mathrm{T}} + \mathbf{R}(1)]^{-1},\end{aligned} \tag{4.47}$$

with new uncertainty

$$\mathbf{P}(1) = \mathbf{P}(1,-) - \mathbf{K}(1)\,\mathbf{E}(1)\,\mathbf{P}(1,-). \tag{4.48}$$

(Compare to the discussion on p. 139.) Equation (4.47) is best interpreted as being the average of the model estimate with the estimate obtained from the data alone, but disguised by rearrangement.

Thus there are four steps:

1. Make a forecast using the model (4.35) with the unknown control terms $\boldsymbol{\Gamma}\mathbf{u}$ set to zero.
2. Calculate the uncertainty of this forecast, Eq. (4.45), which is made up of the sum of the errors owing to initial conditions and to missing controls.
3. Do a weighted average (4.47) of the forecast with the observations, the weighting being chosen to reflect the relative uncertainties.
4. Compute the uncertainty of the final weighted average, Eq. (4.48).

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

Such a computation is called a "Kalman filter";[13] it is conventionally given a more formal derivation. **K** is called the "Kalman gain." At the stage where the forecast (4.42) has already been made, the problem was reduced to finding the minimum of the objective function,

$$\begin{aligned} J = {} & [\tilde{\mathbf{x}}(1,-) - \tilde{\mathbf{x}}(1)]^{\mathrm{T}}\mathbf{P}(1,-)^{-1}[\tilde{\mathbf{x}}(1,-) - \tilde{\mathbf{x}}(1)] \\ & + [\mathbf{y}(1) - \mathbf{E}(1)\,\tilde{\mathbf{x}}(1)]^{\mathrm{T}}\mathbf{R}(1)^{-1}[\mathbf{y}(1) - \mathbf{E}(1)\,\tilde{\mathbf{x}}(1)], \end{aligned} \tag{4.49}$$

which is a variation of the objective function used to define the recursive least-squares algorithm (Eq. 2.425). In this final stage, the explicit model has disappeared, being present only implicitly through the uncertainty $\mathbf{P}(1,-)$. After the averaging step, all of the information about the observations has been used too, and is included in $\tilde{\mathbf{x}}(t)$, $\mathbf{P}(t)$ and the data can be discarded. For clarity, tildes have been placed over all appearances of $\mathbf{x}(t)$.

A complete recursion can now be defined through Eqs. (4.42)–(4.48), replacing all the $t = 0$ variables with $t = 1$ variables, the $t = 1$ variables becoming $t = 2$ variables, etc. In terms of arbitrary t, the recursion is:

$$\tilde{\mathbf{x}}(t,-) = \mathbf{A}(t-1)\tilde{\mathbf{x}}(t-1) + \mathbf{B}(t-1)\mathbf{q}(t-1), \tag{4.50}$$

$$\mathbf{P}(t,-) = \mathbf{A}(t-1)\mathbf{P}(t-1)\mathbf{A}(t-1)^{\mathrm{T}} + \boldsymbol{\Gamma}\mathbf{Q}(t-1)\boldsymbol{\Gamma}^{\mathrm{T}}, \tag{4.51}$$

$$\tilde{\mathbf{x}}(t) = \tilde{\mathbf{x}}(t,-) + \mathbf{K}(t)[\mathbf{y}(t) - \mathbf{E}(t)\tilde{\mathbf{x}}(t,-)], \tag{4.52}$$

$$\mathbf{K}(t) = \mathbf{P}(t,-)\mathbf{E}(t)^{\mathrm{T}}[\mathbf{E}(t)\mathbf{P}(t,-)\mathbf{E}(t)^{\mathrm{T}} + \mathbf{R}(t)]^{-1}, \tag{4.53}$$

$$\mathbf{P}(t) = \mathbf{P}(t,-) - \mathbf{K}(t)\mathbf{E}(t)\mathbf{P}(t,-), \quad t = 1, 2, \ldots, t_f. \tag{4.54}$$

These equations are those for the complete Kalman filter. Note that some authors prefer to write equations for $\tilde{\mathbf{x}}(t+1,-)$ in terms of $\tilde{\mathbf{x}}(t)$, etc. Equation (2.36) permits the rewriting of Eq. (4.54) as

$$\mathbf{P}(t) = [\mathbf{P}(t,-)^{-1} + \mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}\mathbf{E}(t)]^{-1}, \tag{4.55}$$

and an alternate form for the gain is[14]

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}, \tag{4.56}$$

These rewritten forms are often important for computational efficiency and accuracy. Note that in the special case where the observations are employed one-at-a-time, $\mathbf{E}(t)$ is a simple row vector, $\mathbf{E}(t)\,\mathbf{P}(t,-)\,\mathbf{E}(t)^{\mathrm{T}} + \mathbf{R}(t)$ is a scalar, and no matrix inversion is required in Eqs. (4.50)–(4.54). The computation would then be dominated by matrix multiplications. Such a strategy demands that the noise be uncorrelated from one observation to another, or removed by "pre-whitening," which, however, itself often involves a matrix inversion. Various re-arrangements are worth examining in large problems.[15]

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

Notice that the model is being satisfied exactly; in the terminology introduced in Chapter 2, it is a hard constraint. But as was true with the static models, the hard constraint description is misleading, as the presence of the terms in $\mathbf{u}(t)$ means that model errors are permitted. Notice too, that $\mathbf{u}(t)$ has not been estimated.

Example *Consider again the mass–spring oscillator described earlier, with time history in Fig. 4.1. It was supposed that the initial conditions were erroneously provided as* $\tilde{\mathbf{x}}(0) = [10, 10]^{\mathrm{T}}$, $\mathbf{P}(0) = \mathrm{diag}([100, 100])$, *but that the forcing was completely unknown. Observations of* $x_1(t)$ *were provided at every time step with a noise variance* $R = 50$. *The Kalman filter was computed by (4.50)–(4.54) and used to estimate the position at each time step. The result for part of the time history is in Fig. 4.3(a), showing the true value and the estimated value of component* $x_1(t)$. *The time history of the uncertainty of* $x_1(t)$, $\sqrt{P_{11}(t)}$, *is also depicted and rapidly reaches an asymptote. Overall, the filter manages to track the position of the oscillator everywhere within two standard deviations.*

If observations are not available at some time step, t, the best estimate reduces to that from the model forecast alone, $\mathbf{K}(t) = \mathbf{0}$, $\mathbf{P}(t) = \mathbf{P}(t, -)$ and one simply proceeds. Typically in such situations, the error variances will grow from the accumulation of the unknown $\mathbf{u}(t)$, at least, until such times as an observation does become available. If $\mathbf{u}(t)$ is purely random, the system will undergo a form of random walk.[16]

Example *Consider again the problem of fitting a straight line to data, as discussed in Chapter 2, but now in the context of a Kalman filter, using the canonical form derived from (4.50)–(4.54). "Data" were generated from the state transition matrix of Eq. (4.16) and an unforced model, as depicted in Fig. 4.4. The observation equation is*

$$y(t) = x_1(t) + n(t),$$

that is, $\mathbf{E}(t) = \{1 \quad 0\}$, $R(t) = 50$, but observations were available only every 25th time step. *There were no unknown control disturbances – that is* $Q(t) = 0$, *but the initial state estimate was set erroneously as* $\tilde{\mathbf{x}}(0) = [30, 10]^{\mathrm{T}}$, *with an uncertainty* $\mathbf{P}(0) = \mathrm{diag}([900, 900)]$. *The result of the computation for the fit is shown in Fig. 4.5 for 100 time steps. Note that with the Kalman filter the estimate diverges rapidly from the true value (although well within the estimated error) and is brought discontinuously toward the true value when the first observations become available.*

If the state vector is redefined to consist of the two model parameters a, b, then $\mathbf{x} = [a \quad b]^{\mathrm{T}}$ and $\mathbf{A} = \mathbf{I}$. *Now the observation matrix is* $\mathbf{E} = [1 \quad t]$ – *that is, time-dependent. The state vector has changed from a time-varying one to a constant. The incorrect estimates* $\tilde{\mathbf{x}}(0) = [10 \quad 10]^{\mathrm{T}}$ *were used, with*

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

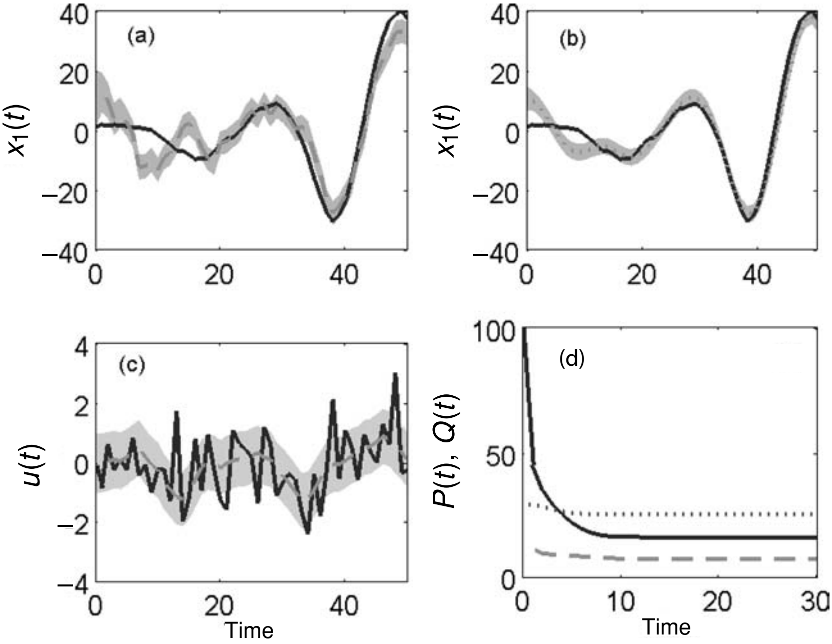

Figure 4.3 (a) Forward run (solid line) of a forced mass–spring oscillator with $r = 0$, $k = 0.1$ and initial condition $\mathbf{x}(0) = [1, 0]^T$. The dashed line is a Kalman filter estimate, $\tilde{x}_1(t)$ started with $\tilde{\mathbf{x}}(0) = [10, 10]$, $\mathbf{P}(0) = \text{diag}([100, 100])$. "Observations" were provided for $x_1(t)$ at every time step, but corrupted with white noise of variance $R = 50$. The shaded band is the one-standard deviation error bar for $\tilde{x}_1(t)$ computed from $\sqrt{P_{11}(t)}$ in the Kalman filter. Rapid convergence toward the true value occurs despite the high noise level. (b) The dotted line now shows $\tilde{x}_1(t, +)$ from the RTS smoothing algorithm. The solid line is again the "truth." Although only the first 50 points are shown, the Kalman filter was run out to $t = 300$, where the smoother was started. The band is the one standard deviation of the smoothed estimate from $\sqrt{P_{11}(t, +)}$ and is smaller than $\sqrt{P_{11}(t)}$. The smoothed estimate is closer to the true value almost everywhere. As with the filter, the smoothed estimate is consistent with the true values within two standard deviations. (c) Estimated $\tilde{u}(t)$ (dashed) and its standard error from the smoother. The solid line is the "true" value (which is itself white noise). That $\tilde{u}(t)$ lacks the detailed structure of the true $u(t)$ is a consequence of the inability of the mass–spring oscillator to respond instantaneously to a white noise forcing. Rather, it responds to an integrated value. (d) The solid line is $P_{11}(t)$, dashed is $P_{11}(t, +)$, and dotted curve is 30Q $(t, +)$ with the scale factor used to make it visible. (Squares of values shown as bands in the other panels.) Note the rapid tendency towards a steady state. Values are largest at $t = 0$ as data are only available in the future, not the past. Q is multiplied by a large factor to make it visible.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635489.005
Cambridge Books Online © Cambridge University Press, 2009

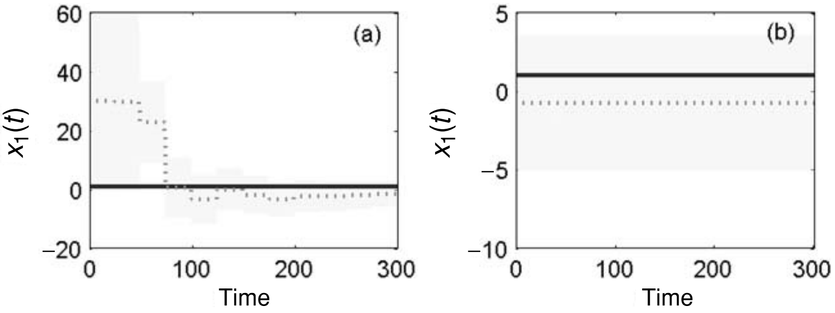

Figure 4.4 (a) $x_1(t)$ (solid) and $\tilde{x}_1(t) = \tilde{a}$ (dotted) from the straight-line model and the Kalman filter estimate when the state vector was defined to be the intercept and slope, and $\mathbf{E}(t) = [1, t]$. (b) Smoothed estimate, $\tilde{x}_1(t, +)$, (dotted) and its uncertainty corresponding to (a).

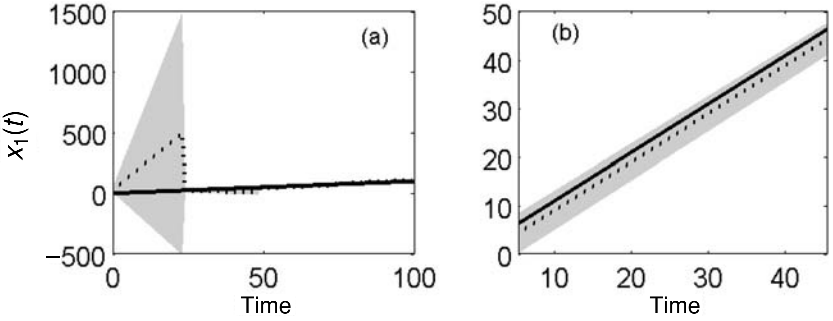

Figure 4.5 A straight line computed from the statespace model with **A** from Eq. (4.16) with no forcing. (a) The solid line shows the true values. Noisy observations were provided every 25th point, and the initial condition was set erroneously to $\tilde{\mathbf{x}}(0) = [30, 10]^T$ with $\mathbf{P}(0) = \text{diag}([900, 900])$. The estimation error grows rapidly away from the incorrect initial conditions until the first observations are obtained. Estimate is shown as the dashed line. The gray band is the one standard deviation error bar. (b) Result of applying the RTS smoother to the data and model in (a).

$\mathbf{P}(0) = \text{diag}([10, 10])$ *(the correct values are* $a = 1$*,* $b = 2$*) and with the time histories of the estimates depicted in Fig. 4.3. At the end of 100 time steps, we have* $\tilde{a} = 1.85 \pm 2.0$*,* $\tilde{b} = 2.0 \pm 0.03$*, both of which are consistent with the correct values. For reasons the reader might wish to think about, the uncertainty of the intercept is much greater than for the slope.*

Example *For the mass–spring oscillator in Fig. 4.3, it was supposed that the same noisy observations were available, but only at every 25th time step. In general, the presence of the model error, or control uncertainty, accumulates over the 25 time steps as the model is run forward without observations. The expected error of such a system is shown for 100 time steps in Fig. 4.6(d). Notice (1) the growing*

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

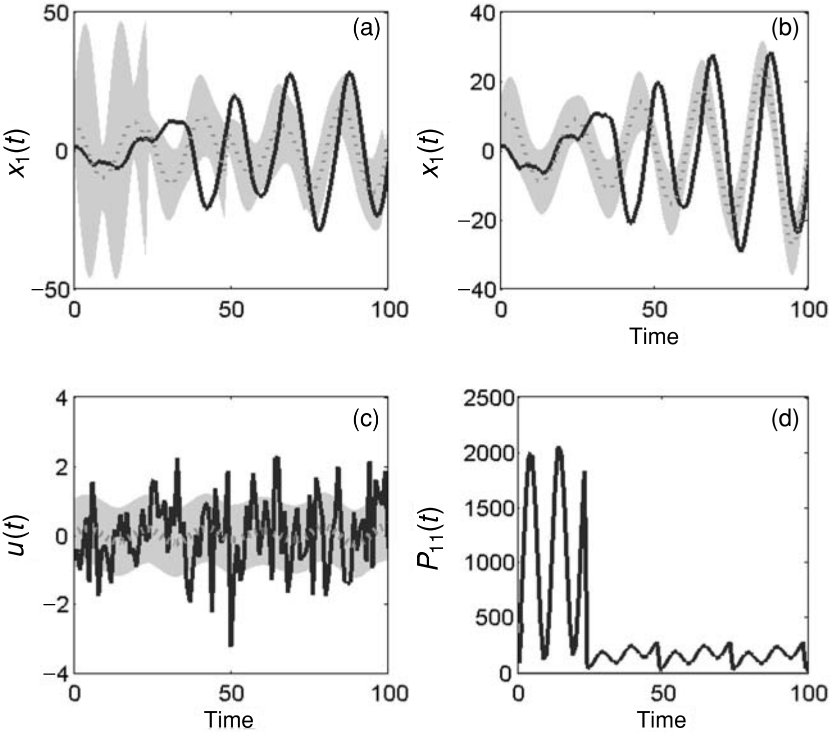

Figure 4.6 For the same model as in Fig. 4.3, except that noisy observations were available only every 25th point ($R = 50$). (a) Shows the correct trajectory of $x_1(t)$ for 100 time steps, the dotted line shows the filter estimate and the shaded band is the standard error of the estimate. (b) Displays the correct value of $x_1(t)$ compared to the (dotted) RTS smoother value with its standard error. (c) Is the estimated control (dotted) with its standard error, and the true value applied to mass–spring oscillator. (d) Shows the behavior of $P_{11}(t)$ for the Kalman filter with very large values (oscillating with twice the frequency of the oscillator) and which become markedly reduced as soon as the first observations become available at the 25th point.

envelope as uncertainty accumulates faster than the observations can reduce it; (2) the periodic nature of the error within the growing envelope; and (3) that the envelope appears to be asymptoting to a fixed upper bound for large t. The true and estimated time histories for a portion of the time history are shown in Fig. 4.6(d). As expected, with fewer available observations, the misfit of the estimated and true values is larger than with data at every time step. At every 25th point, the error norm drops as observations become available, but with the estimated value of $x_1(t)$ undergoing a jump when the observation is available.

If the observation is that of the velocity $x_1(t) - x_2(t) = \xi(t) - \xi(t-1)$, then $E = \{1 \quad -1\}$. A portion of the time history of the Kalman filtered estimate with a velocity observation available only at every 25th point may be seen in Fig. 4.7.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

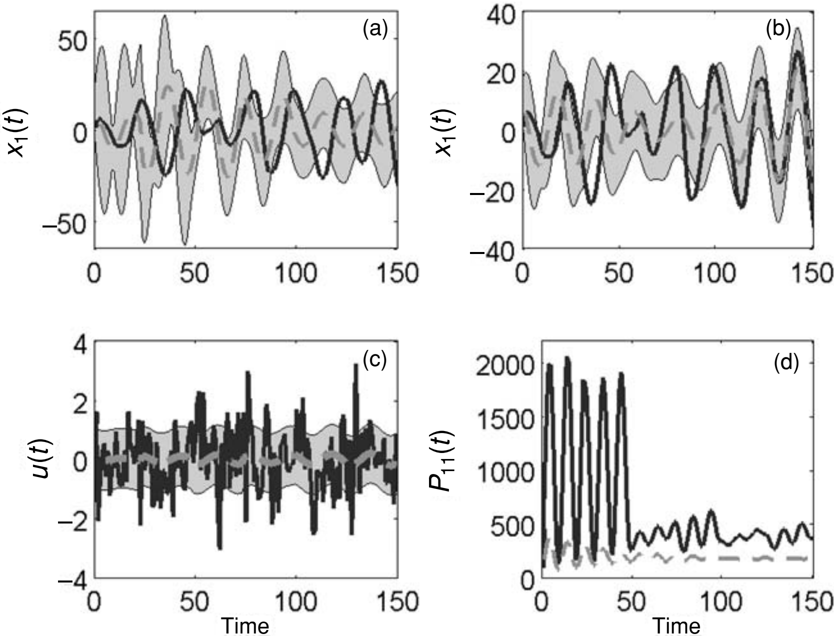

Figure 4.7 (a) $x_1(t)$ and Kalman filter estimate (dashed) when the noisy observations ($R = 50$) are of the velocity ($\mathbf{E} = [1, -1]$) every 25th point for the mass–spring oscillator. (b) RTS smoother estimate (dashed) and its uncertainty corresponding to (a). (c) The estimated control (dotted) and the correct value (solid). As seen previously, the high frequency variability in $u(t)$ is not detected by the moving mass, but only an integrated version. (d) $P_{11}(t)$ (solid) corresponding to the standard error in (a), and $P_{11}(t, +)$ (dashed) corresponding to that in (b).

Velocity observations are evidently useful for estimating position, owing to the connection between velocity and position provided by the model and is a simple example of how observations of almost anything can be used to improve a state estimate.

A number of more general reformulations of the equations into algebraically equivalent forms are particularly important. In one form, one works not with the covariances, $\mathbf{P}(t, -), \ldots,$ but with their inverses, the so-called information matrices, $\mathbf{P}(t, -)^{-1}$, etc. (See Eq. (4.55).) This "information filter" form may be more efficient if, for example, the information matrices are banded and sparse while the covariance matrices are not. Or, if the initial conditions are infinitely uncertain, $\mathbf{P}(0)^{-1}$ can be set to the zero matrix. In another formulation, one uses the square roots (Cholesky decomposition) of the covariance matrices rather than the matrices themselves. This "square root filter" is important, as there is a tendency for the computation of the updated values of $\mathbf{P}$ in Eq. (4.54) to become non-positive-definite owing to round-off errors, and the square root formulation guarantees a positive definite result.[17]

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

The Kalman filter does *not* produce the minimum of the objective function Eq. (4.41) because the data from times later than t are not being used to make estimates of the earlier values of the state vector or of $\mathbf{u}(t)$. At each step, the filter is instead minimizing an objective function of the form in Eq. (4.50). To obtain the needed minimum, we have to consider what is called the "smoothing problem," to which we will turn in a moment. Note too, that the time history of $\mathbf{x}(t)$ does not satisfy a known equation at the time observations are introduced. When no observation is available, the time evolution obeys the model equation with zero control term; the averaging step of the filter, however, leads to a change between t and $t-1$ that compensates for the accumulated error. The evolution equation is no longer satisfied in this interval.

The Kalman filter is, nonetheless, extremely important in practice for many problems. In particular, if one must literally make a forecast (e.g., such filters are used to help land airplanes or, in a primitive way, to forecast the weather), then the future data are simply unavailable, and the state estimate made at time t, using data up to and including time t, is the best one can do.[18]

For estimation, the Kalman filter is only a first step – owing to its failure to use data from the formal future. It also raises questions about computational feasibility. As with all recursive estimators, the uncertainties $\mathbf{P}(t,-)$, $\mathbf{P}(t)$ must be available so as to form the weighted averages. If the state vector contains N elements, then the model (4.50) requires multiplying an N-dimensional vector by an $N \times N$ matrix at each time step. The covariance update (4.51) requires updating each of N columns of $\mathbf{P}(t)$ in the same way, and then doing it again (i.e., in practice, one forms $\mathbf{A}(t)\mathbf{P}(t)$, transposes it, and forms $\mathbf{A}(t)(\mathbf{A}(t)\mathbf{P}(t))^{\mathrm{T}}$, equivalent to running the model $2N$ times at each time step). In many applications, particularly in geophysical fluids, this covariance update step dominates the calculation, renders it impractical, and leads to some of the approximate methods taken up presently.

The Kalman filter was derived heuristically as a simple generalization of the ideas used in Chapter 2. Unsurprisingly, the static inverse results are readily recovered from the filter in various limits. As one example, consider the nearly noise-free case in which both process and observation noise are very small, i.e. $\|\mathbf{Q}\|, \|\mathbf{R}\| \to 0$. Then if $\mathbf{P}(t,-)$ is nearly diagonal, $\mathbf{P}(t,-) \sim \delta^2 \mathbf{I}$, and

$$\mathbf{K}(t) \longrightarrow \mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1},$$

assuming existence of the inverse, and

$$\begin{aligned}\tilde{\mathbf{x}}(t) &\sim \mathbf{A}\tilde{\mathbf{x}}(t-1) + \mathbf{B}\mathbf{q}(t-1)\\ &\qquad + \mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1}\{\mathbf{y}(t) - \mathbf{E}[\mathbf{A}\tilde{\mathbf{x}}(t-1) + \mathbf{B}\mathbf{q}(t-1)]\}\\ &= \mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1}\mathbf{y}(t)\\ &\qquad + [\mathbf{I} - \mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1}\mathbf{E}]\,[\mathbf{A}\tilde{\mathbf{x}}(t-1) + \mathbf{B}\mathbf{q}(t-1)].\end{aligned} \tag{4.57}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

$\mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1}\mathbf{y}(t)$ is just the expression in Eq. (2.95) for the direct estimate of $\mathbf{x}(t)$ from a set of underdetermined full-rank, noise-free observations. It is the static estimate we would use at time t if no dynamics were available. The columns of $\mathbf{I}-\mathbf{E}^{\mathrm{T}}(\mathbf{E}\mathbf{E}^{\mathrm{T}})^{-1}\mathbf{E}$ are the nullspace of $\mathbf{E}$ (recall the definition of $\mathbf{H}$ in Eq. (2.97)) and (4.57) thus employs only those elements of the forecast lying in the nullspace of the observations – a sensible result given that the observations here produce perfect estimates of components of $\mathbf{x}(t+1)$ in the range of $\mathbf{E}$. Thus, in this particular limit, the Kalman filter computes from the noise-free observations those elements of $\mathbf{x}(t+1)$ that it can, and for those which it cannot, it forecasts them from the dynamics. The reader ought to examine other limiting cases – retaining process and/or observational noise – including the behavior of the error covariance propagation.

Example *It is interesting to apply some of these expressions to the simple problem of finding the mean of a set of observations, considered before on p. 133. The model is of an unchanging scalar mean,*

$$x(t)=x(t-1),$$

observed in the presence of noise,

$$y(t)=x(t)+n(t),$$

where $\langle n\,(t)\rangle=0,\ \langle n(t)^2\rangle=R$, *so* $E=1$, $A=1$, $Q=0$, $t=0,1,\ldots,m-1$. *In contrast to the situation on p. 133, the machinery used here requires that the noise be uncorrelated:* $\langle n(t)n(t')\rangle=0$, $t\neq t'$, *although as already mentioned, methods exist to overcome this restriction. Suppose that the initial estimate of the mean is* 0 *– that is,* $\tilde{x}(0)=0$, *with uncertainty* $P(0)$. *Equation (4.51) is* $P(t,-)=P(t-1)$, *and the Kalman filter uncertainty, in the form (4.55), is*

$$\frac{1}{P(t)}=\frac{1}{P(t-1)}+\frac{1}{R},$$

a difference equation, with known initial condition, whose solution by inspection is

$$\frac{1}{P(t)}=\frac{t}{R}+\frac{1}{P(0)}.$$

Using (4.52) with $E=1$, *and successively stepping forward, produces*[19]

$$\tilde{x}(m-1)=\frac{R}{R+mP(0)}\left\{\frac{P(0)}{R}\sum_{j=0}^{m-1}y(j)\right\}, \tag{4.58}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005
Cambridge Books Online © Cambridge University Press, 2009

whose limit as $t \to \infty$ is

$$\tilde{x}(m-1) \longrightarrow \frac{1}{m}\sum_{j=0}^{m-1} y(j),$$

the simple average, with uncertainty $P(t) \to 0$, as $t \to \infty$. If there is no useful estimate available of $P(0)$, rewrite Eq. (4.58) as

$$\tilde{x}(m-1) = \frac{R}{R/P(0)+m}\left\{\frac{1}{R}\sum_{j=0}^{m-1} y(j)\right\}, \tag{4.59}$$

and take the agnostic limit, $1/P(0) \to 0$, or

$$\tilde{x}(m-1) = \frac{1}{m}\left\{\sum_{j=0}^{m-1} y(j)\right\}, \tag{4.60}$$

which is wholly conventional. (Compare these results to those on p. 133. The problem and result here are necessarily identical to those, except that now $x(t)$ is identified explicitly as a state vector rather than as a constant. Kalman filters with static models reduce to ordinary least-squares solutions.)

4.3.3 The smoothing problem

The Kalman filter permits one to make an optimal forecast from a linear model, subject to the accuracy of the various assumptions being made. Between observation times, the state estimate evolves smoothly according to the model dynamics. But when observations become available, the averaging can draw the combined state estimate abruptly towards the observations, and in the interval between the last unobserved state and the new one, model evolution is not followed. To obtain a state trajectory that is both consistent with model evolution and the data at all times, the state estimate jumps at the observation times need to be removed, and the problem solved as originally stated. Minimization of J in Eq. (4.41) subject to the model is still the goal. Begin the discussion again with a one-step process,[20] for the problem Eqs. (4.35)–(4.39), but where there are only two times involved, $t = 0,\ 1$. There is an initial estimate $\tilde{\mathbf{x}}(0),\ \tilde{\mathbf{u}}(0) \equiv 0$ with uncertainties $\mathbf{P}(0)$, $\mathbf{Q}(0)$ for the initial state and control vectors respectively, a set of measurements at time-step 1, and the model. The objective function is

$$\begin{aligned} J = {} & [\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)]^{\mathrm{T}}\, \mathbf{P}(0)^{-1}\, [\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)] \\ & + [\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)]^{\mathrm{T}}\, \mathbf{Q}(0)^{-1}\, [\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)] \\ & + [\mathbf{y}(1) - \mathbf{E}(1)\,\tilde{\mathbf{x}}(1)]^{\mathrm{T}}\, \mathbf{R}(1)^{-1}\, [\mathbf{y}(1) - \mathbf{E}(1)\,\tilde{\mathbf{x}}(1)]\,, \end{aligned} \tag{4.61}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005

Cambridge Books Online © Cambridge University Press, 2009

subject to the model

$$\tilde{\mathbf{x}}(1) = \mathbf{A}(0)\,\tilde{\mathbf{x}}(0,+) + \mathbf{B}(0)\mathbf{q}(0) + \boldsymbol{\Gamma}\tilde{\mathbf{u}}(0,+), \tag{4.62}$$

with the weight matrices again chosen as the inverses of the prior covariances. Tildes have now been placed on all estimated quantities. A minimizing solution to this objective function would produce a new estimate of $\mathbf{x}(0)$, denoted $\tilde{\mathbf{x}}(0,+)$, with error covariance $\mathbf{P}(0,+)$; the $+$ denotes use of *future* observations, $\mathbf{y}(1)$, in the estimate. On the other hand, we would still denote the estimate at $t = 1$ as $\tilde{\mathbf{x}}(1)$, coinciding with the Kalman filter estimate, because only data prior to and at the same time would have been used. The estimate $\tilde{\mathbf{x}}(1)$ must be given by Eq. (4.47), but it remains to improve $\tilde{\mathbf{u}}(0)$, $\tilde{\mathbf{x}}(0)$, while simultaneously eliminating the problem of the estimated state vector jump at the filter averaging (observation) time.

The basic issue can be understood by observing that the initial estimates $\tilde{\mathbf{u}}(0) = \mathbf{0}$, $\tilde{\mathbf{x}}(0)$ lead to a model forecast that disagrees with the final best estimate $\tilde{\mathbf{x}}(1)$. If either of $\tilde{\mathbf{u}}(0)$, or $\tilde{\mathbf{x}}(0)$ were known perfectly, the forecast discrepancy could be ascribed to the other one, permitting ready computation of the new required value. In practice, both are somewhat uncertain, and the modification must be partitioned between them; one would not be surprised to find that the partitioning proves to be proportional to their initial uncertainties.

To find the stationary point (we will not trouble to prove it a minimum rather than a maximum), set the differential of J with respect to $\tilde{\mathbf{x}}(0,+)$, $\tilde{\mathbf{x}}(1)$, $\tilde{\mathbf{u}}(0,+)$ to zero (recall Eq. 2.91),

$$\begin{aligned}\frac{\mathrm{d}J}{2} = {}& \mathrm{d}\tilde{\mathbf{x}}(0,+)^{\mathrm{T}}\mathbf{P}(0)^{-1}[\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)]\\ &+ \mathrm{d}\tilde{\mathbf{u}}(0,+)^{\mathrm{T}}\mathbf{Q}(0)^{-1}[\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)]\\ &- \mathrm{d}\tilde{\mathbf{x}}(1)^{\mathrm{T}}\mathbf{E}(1)^{\mathrm{T}}\mathbf{R}(1)^{-1}[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)] = 0.\end{aligned} \tag{4.63}$$

The coefficients of the differentials cannot be set to zero separately because they are connected via the model (4.62), which provides the relationship

$$\mathrm{d}\tilde{\mathbf{x}}(1) = \mathbf{A}(0)\,\mathrm{d}\tilde{\mathbf{x}}(0,+) + \boldsymbol{\Gamma}(0)\,\mathrm{d}\tilde{\mathbf{u}}(0,+). \tag{4.64}$$

Eliminating $\mathrm{d}\tilde{\mathbf{x}}(1)$,

$$\begin{aligned}\frac{\mathrm{d}J}{2} = {}& \mathrm{d}\tilde{\mathbf{x}}(0,+)^{\mathrm{T}}\{\mathbf{P}(0)^{-1}\,[\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)]\\ &\qquad - \mathbf{A}(0)^{\mathrm{T}}\mathbf{E}(1)^{\mathrm{T}}\mathbf{R}(1)^{-1}\,[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)]\}\\ &+ \mathrm{d}\tilde{\mathbf{u}}(0,+)^{\mathrm{T}}\{\mathbf{Q}(0)^{-1}\,[\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)]\\ &\qquad + \boldsymbol{\Gamma}^{\mathrm{T}}(0)\mathbf{E}(1)^{\mathrm{T}}\mathbf{R}(1)^{-1}\,[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)]\}.\end{aligned} \tag{4.65}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

dJ vanishes only if the coefficients of d$\tilde{\mathbf{x}}(0,+)$, d$\tilde{\mathbf{u}}(0,+)$ separately vanish, yielding

$$\tilde{\mathbf{x}}(0,+) = \tilde{\mathbf{x}}(0) + \mathbf{P}(0)\mathbf{A}(0)^{\mathrm{T}}\mathbf{E}(1)^{\mathrm{T}}\mathbf{R}(1)^{-1}\left[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)\right], \tag{4.66}$$

$$\tilde{\mathbf{u}}(0,+) = \tilde{\mathbf{u}}(0) + \mathbf{Q}(0)\mathbf{\Gamma}(0)^{\mathrm{T}}\mathbf{E}(1)^{\mathrm{T}}\mathbf{R}(1)^{-1}\left[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)\right], \tag{4.67}$$

and

$$\begin{aligned}\tilde{\mathbf{x}}(1) = \tilde{\mathbf{x}}(1,-) + \mathbf{P}(1,-)\mathbf{E}(1)^{\mathrm{T}}[\mathbf{E}(1)\mathbf{P}(1,-)\mathbf{E}(1)^{\mathrm{T}} + \mathbf{R}(1)]^{-1}\\ \times [\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1,-)],\end{aligned} \tag{4.68}$$

using the previous definitions of $\tilde{\mathbf{x}}(1,-)$, $\mathbf{P}(1,-)$. As anticipated, Eq. (4.68) is recognizable as the Kalman filter estimate. At this point we are essentially done: an estimate has been produced not only of $\mathbf{x}(1)$, but an improvement has been made in the prior estimate of $\mathbf{x}(0)$ using the future measurements, and the control term has been estimated. Notice that the corrections to $\tilde{\mathbf{u}}(0)$, $\tilde{\mathbf{x}}(0)$ are proportional to $\mathbf{Q}(0)$, $\mathbf{P}(0)$, respectively, as anticipated. The uncertainties of these latter quantities are still needed.

First rewrite the estimates (4.66) and (4.67) as

$$\begin{aligned}\tilde{\mathbf{x}}(0,+) &= \tilde{\mathbf{x}}(0) + \mathbf{L}(1)\left[\tilde{\mathbf{x}}(1) - \tilde{\mathbf{x}}(1,-)\right], \quad \mathbf{L}(1) = \mathbf{P}(0)\mathbf{A}(0)^{\mathrm{T}}\mathbf{P}(1,-)^{-1},\\ \tilde{\mathbf{u}}(0,+) &= \tilde{\mathbf{u}}(0) + \mathbf{M}(1)\left[\tilde{\mathbf{x}}(1) - \tilde{\mathbf{x}}(1,-)\right], \quad \mathbf{M}(1) = \mathbf{Q}(0)\mathbf{\Gamma}(0)^{\mathrm{T}}\mathbf{P}(1,-)^{-1},\end{aligned} \tag{4.69}$$

which can be done by extended, but uninteresting, algebraic manipulation. The importance of these latter two expressions is that both $\tilde{\mathbf{x}}(0,+)$, $\tilde{\mathbf{u}}(0,+)$ are expressed in terms of their prior estimates in a weighted average with the difference between the prediction of the state at $t = 1$, $\tilde{\mathbf{x}}(1,-)$ and what was actually estimated there following the data use, $\tilde{\mathbf{x}}(1)$. (But the data do not appear explicitly in (4.69).) It is also possible to show that

$$\begin{aligned}\mathbf{P}(0,+) &= \mathbf{P}(0) + \mathbf{L}(1)[\mathbf{P}(1) - \mathbf{P}(1,-)]\mathbf{L}(1)^{\mathrm{T}},\\ \mathbf{Q}(0,+) &= \mathbf{Q}(0) + \mathbf{M}(1)[\mathbf{P}(1) - \mathbf{P}(1,-)]\mathbf{M}(1)^{\mathrm{T}}.\end{aligned} \tag{4.70}$$

Based upon this one-step derivation, a complete recursion for any time interval can be inferred. Suppose that the Kalman filter has been run all the way to a terminal time, t_f. The result is $\tilde{\mathbf{x}}(t_f)$ and its variance $\mathbf{P}(t_f)$. With no future data available, $\tilde{\mathbf{x}}(t_f)$ cannot be further improved. At time $t_f - 1$, we have an estimate $\tilde{\mathbf{x}}(t_f - 1)$ with uncertainty $\mathbf{P}(t_f - 1)$, which could be improved by knowledge of the future observations at t_f. But this situation is precisely the one addressed by the objective function (4.61), replacing $t = 1 \to t_f$, and $t = 0 \to t_f - 1$. Now having improved the estimate at $t_f - 1$ and calling it $\tilde{\mathbf{x}}(t_f - 1, +)$ with uncertainty $\mathbf{P}(t_f - 1, +)$, this new estimate is used to improve the prior estimate $\tilde{\mathbf{x}}(t_f - 2)$, and step all the way

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575099.005
Cambridge Books Online © Cambridge University Press, 2009

back to $t = 0$. The complete recursion is

$$\boxed{\begin{aligned}
\tilde{\mathbf{x}}(t,+) &= \tilde{\mathbf{x}}(t) + \mathbf{L}(t+1)\,[\tilde{\mathbf{x}}(t+1,+) - \tilde{\mathbf{x}}(t+1,-)], \\
\mathbf{L}(t+1) &= \mathbf{P}(t)\mathbf{A}(t)^{\mathrm{T}}\,\mathbf{P}(t+1,-)^{-1}, & (4.71) \\
\tilde{\mathbf{u}}(t,+) &= \tilde{\mathbf{u}}(t) + \mathbf{M}(t+1)\,[\tilde{\mathbf{x}}(t+1,+) - \tilde{\mathbf{x}}(t+1,-)], \\
\mathbf{M}(t+1) &= \mathbf{Q}(t)\boldsymbol{\Gamma}(t)^{\mathrm{T}}\mathbf{P}(t+1,-)^{-1}, & (4.72) \\
\mathbf{P}(t,+) &= \mathbf{P}(t) + \mathbf{L}(t+1)[\mathbf{P}(t+1,+) - \mathbf{P}(t+1,-)]\mathbf{L}(t+1)^{\mathrm{T}}, & (4.73) \\
\mathbf{Q}(t,+) &= \mathbf{Q}(t) + \mathbf{M}(t+1)[\mathbf{P}(t+1,+) - \mathbf{P}(t+1,-)]\mathbf{M}(t+1)^{\mathrm{T}}, & (4.74)
\end{aligned}}$$

with $\tilde{\mathbf{x}}(t_f,+) \equiv \tilde{\mathbf{x}}(t_f),\ \mathbf{P}(t_f,+) \equiv \mathbf{P}(t_f)$, for $t = 0, 1, \ldots, t_f - 1$.

This recipe, which uses the Kalman filter on a first forward sweep to the end of the available data, and which then successively improves the prior estimates by sweeping backwards, is called the "RTS algorithm" or smoother.[21] The particular form has the advantage that the data are not involved in the backward sweep, because all of the available information has been used in the filter calculation. It does have the potential disadvantage of requiring the storage at each time step of $\mathbf{P}(t)$. ($\mathbf{P}(t,-)$ is readily recomputed, without $\mathbf{y}(t)$, from (4.51) and need not be stored.) By direct analogy with the one-step objective function, the recursion (4.71)–(4.74) is seen to be the solution to the minimization of the objective function (4.61) subject to the model. Most important, assuming consistency of all assumptions, the resulting state vector trajectory $\tilde{\mathbf{x}}\,(t,+)$ now satisfies the model and no longer displays the jump discontinuities at observation times of the Kalman filter estimate.

As with the Kalman filter, it is possible to examine limiting cases of the RTS smoother. Suppose again that $\mathbf{Q}$ vanishes, and $\mathbf{A}^{-1}$ exists. Then

$$\mathbf{L}(t+1) \longrightarrow \mathbf{P}(t)\mathbf{A}^{\mathrm{T}}\,(\mathbf{A}\mathbf{P}(t)\mathbf{A}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1}, \tag{4.75}$$

and Eq. (4.71) becomes

$$\tilde{\mathbf{x}}(t,+) \longrightarrow \mathbf{A}^{-1}\,[\tilde{\mathbf{x}}(t+1,+) - \mathbf{B}\mathbf{q}(t)]. \tag{4.76}$$

A sensible backward estimate obtained by simply solving

$$\tilde{\mathbf{x}}(t+1) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}\mathbf{q}(t), \tag{4.77}$$

for $\tilde{\mathbf{x}}(t)$. Other limits are also illuminating but are left to the reader.

Example *The smoother result for the straight-line model (4.15) is shown in Figs. 4.4 and 4.5 for both forms of state vector. The time-evolving estimate is now a nearly perfect straight line, whose uncertainty has a terminal value equal to that for the Kalman filter estimate, as it must, and reaches a minimum near the middle of the estimation period, before growing again toward $t = 0$, where the initial uncertainty was very large. In the case where the state vector consisted of*

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

the constant intercept and slope of the line, both smoothed estimates are seen, in contrast to the filter estimate, to conform very well to the known true behavior. It should be apparent that the best-fitting straight-line solution of Chapter 2 is also the solution to the smoothing problem, but with the data and model handled all at once, in a whole-domain method, rather than sequentially.

Figures 4.3, 4.6 and 4.7 show the state estimate and its variance for the mass–spring oscillator made from a smoothing computation run backward from $t = 300$. *On average, the smoothed estimate is closer to the correct value than is the filtered estimate, as expected. The standard error is also smaller for the smoothed estimate. The figures display the variance,* $Q_{11}(t, +)$, *of the estimate one can make of the scalar control variable* $u(t)$. $\tilde{u}\,(t)$ *does not show the high frequency variability present in practice, because the mass–spring oscillator integrates the one time-step variability in such a way that only an integrated value affects the state vector. But the estimated value is nonetheless always within two standard errors of the correct value.*

Example *Consider a problem stimulated by the need to extract information from transient tracers,* C, *in a fluid, which are assumed to satisfy the equation*

$$\frac{\partial C}{\partial t} + \mathbf{v} \cdot \nabla C - \kappa \nabla^2 C = -\lambda C + q(\mathbf{r}, t), \tag{4.78}$$

where q *represents sources/sinks and* λ *is a decay rate if the tracer is radioactive. To have a simple model that will capture the structure of this problem, the fluid is divided into a set of boxes as depicted in Fig. 4.8. The flow field is represented by exchanges between boxes given by the* $J_{ij} \geq 0$. *That is, the* J_{ij} *are a simplified representation of the effects of advection and mixing on a dye* C. *(A relationship can be obtained between such simple parameterizations and more formal and elaborate finite-difference schemes. Here, it will only be remarked that* J_{ij} *are chosen to be mass conserving so that the sum over all* J_{ij} *entering and leaving a box vanishes.) The discrete analogue of (4.78) is taken to be*

$$\begin{aligned} C_i(t+1) = {} & C_i(t) - \lambda \Delta t C_i(t) \\ & - \frac{\Delta t}{V} \sum_{j \in N(i)} C_i(t) J_{ij} + \frac{\Delta t}{V} \sum_{j \in N(i)} C_j(t) J_{ji}, \end{aligned} \tag{4.79}$$

where the notation $j \in N(i)$ *denotes an index sum over the neighboring boxes to box* i, V *is a volume for the boxes, and* Δt *is the time step. This model can easily be put into canonical form,*

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1) + \mathbf{B}\mathbf{q}(t-1) + \boldsymbol{\Gamma}\mathbf{u}(t-1), \qquad \mathbf{Q} = \mathbf{0}, \tag{4.80}$$

with the state vector composed of box concentrations $C_i(t)$, $C_i(t-1)$.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

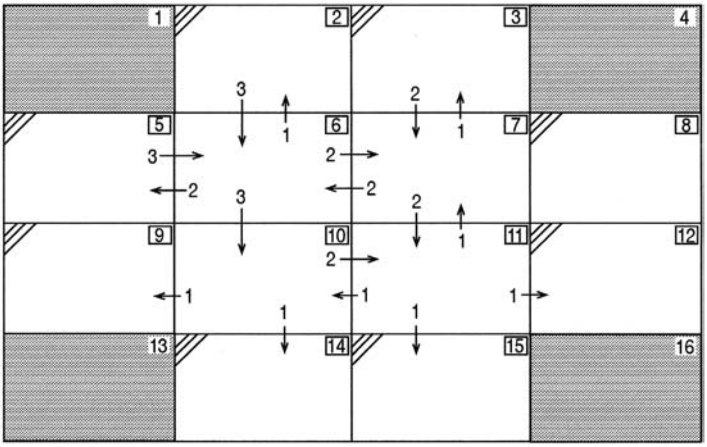

Figure 4.8 Tracer box model where J_{ij} represent fluxes between boxes and are chosen to be mass conserving. Boxes with shaded corners are boundary boxes with externally prescribed concentrations. Numbers in the upper-right corners are used to identify the boxes. Stippled boxes are unconnected and completely passive here. (Source: Wunch, 1988)

A forward computation was run with initial concentrations everywhere of 0*, using the boundary conditions depicted in Fig. 4.9, resulting in interior box values as shown. Based upon these correct values, noisy "observations" of the interior boxes only were constructed at times* $t = 10,\ 20,\ 30$*. The noise variance was 0.01.*

An initial estimate of interior tracer concentrations at $t = 0$ *was taken (correctly) to be zero, but this estimate was given a large variance (*$\operatorname{diag}(\mathbf{P}(0)) = 4$*). The a-priori boundary box concentrations were set erroneously to* $C = 2$ *for all t and held at that value. A Kalman filter computation was run as shown in Fig. 4.10. Initially, the interior box concentration estimates rise erroneously (owing to the dye leaking in from the high non-zero concentrations in the boundary boxes). At* $t = 10$*, the first set of observations becomes available, and the combined estimate is driven much closer to the true values. By the time the last set of observations is used, the estimated and correct concentrations are quite close, although the time history of the interior is somewhat in error. The RTS algorithm was then applied to generate the smoothed histories shown in Fig. 4.10 and to estimate the boundary concentrations (the controls). As expected, the smoothed estimates are closer to the true time history than are the filtered ones. Unless further information is provided, no other estimation procedure could do better, given that the model is the correct one.*

Other smoothing algorithms exist. Consider one other approach. Suppose the Kalman filter has been run forward to some time t_c, producing an estimate $\tilde{\mathbf{x}}(t_c)$ with uncertainty $\mathbf{P}(t_c)$. Now suppose, perhaps on the basis of some further observations,

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

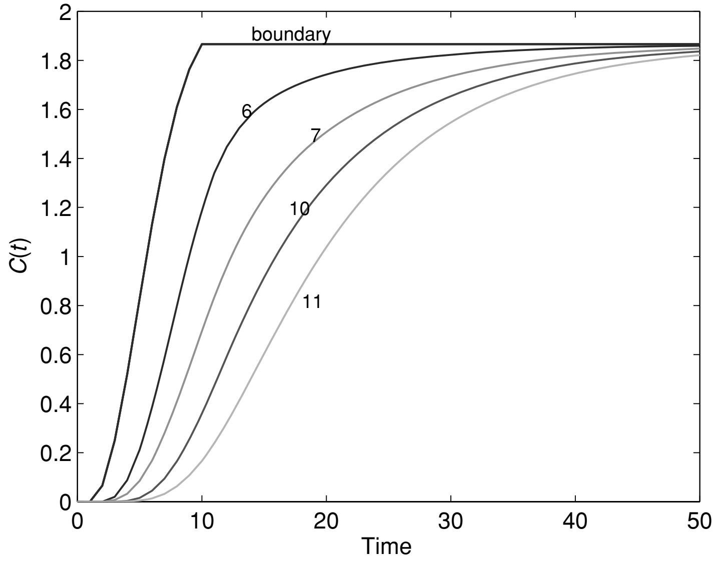

Figure 4.9 Time histories of the forward computation in which boundary concentrations shown were prescribed, and values computed from the forward model for boxes 6, 7, 10, 11. These represent the "truth." Here $\Delta t = 0.05$.

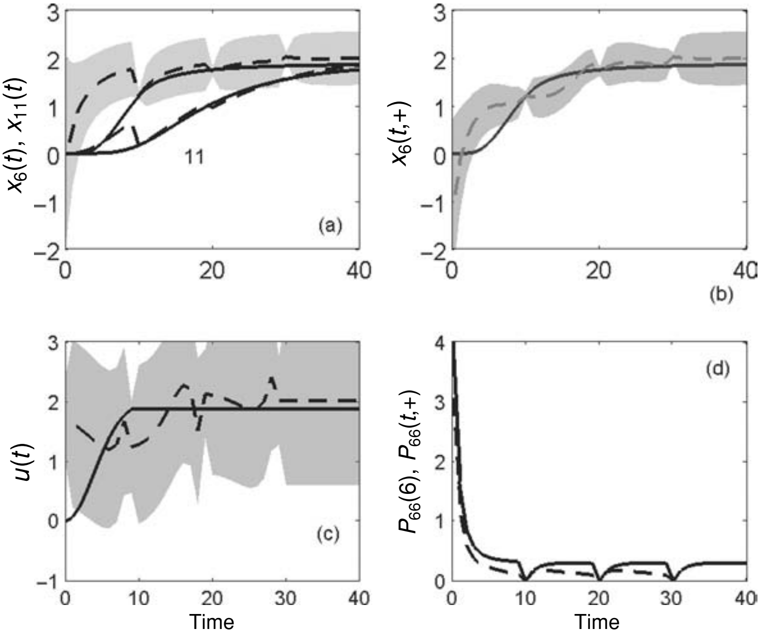

Figure 4.10 (a) Kalman filter estimate (dashed line) with one standard error band, compared to the "true" value in box 6, $x_t(t)$. Filter and true value are shown (without the error band) also in box 11. Observations of all interior values were available at $t = 10, 20, 30$, but the calculations were carried out beyond the last observation. (b) Smoothed estimate $\tilde{x}_6(t, +)$ and one standard error band. (c) Estimated control $\tilde{u}(t, +)$ (dashed) with one standard error, and the correct value (solid). (d) $P_{66}(t)$ (solid) and $P_{66}(t, +)$ (dashed). The smoothed variance is everywhere less than the variance of the filter estimate for all $t < 30$ (the time of the last observation).

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

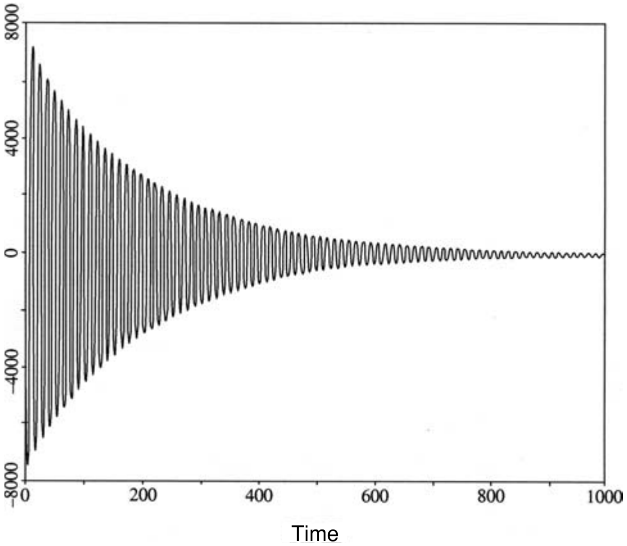

Figure 4.11 Mass–spring oscillator model with friction ($r = 0.01$) run backwards in time from conditions specified at $t = 1000$. The system is unstable, and small uncertainties in the starting conditions would amplify. But the Kalman filter run backwards remains stable because its error estimate also grows – systematically downweighting the model forecast relative to any data that become available at earlier times. A model with unstable elements in the forward direction would behave analogously when integrated in time with growing estimated model forecast error. (Some might prefer to reverse the time scale in such plots.)

that at a *later* time $t_f > t_c$, an independent estimate $\tilde{\mathbf{x}}(t_f)$ has been made, with uncertainty $\mathbf{P}(t_f)$. The independence is crucial – we suppose this latter estimate is made without using any observations at time t_c or earlier so that any errors in $\tilde{\mathbf{x}}(t_c)$ and $\tilde{\mathbf{x}}(t_f)$ are uncorrelated.

Run the model *backwards* in time from t_f to $t_f - 1$:

$$\tilde{\mathbf{x}}_b(t_f - 1) = \mathbf{A}^{-1}\tilde{\mathbf{x}}(t_f) - \mathbf{A}^{-1}\mathbf{B}\mathbf{q}(t_f - 1), \tag{4.81}$$

where the subscript b denotes a backwards-in-time estimate (see Fig. 4.11). The reader may object that running a model backwards in time will often be an unstable operation; this objection needs to be addressed, but ignore it for the moment. The uncertainty of $\tilde{\mathbf{x}}(t_f - 1)$ is

$$\mathbf{P}_b(t_f - 1) = \mathbf{A}^{-1}\mathbf{P}(t_f)\mathbf{A}^{-\mathrm{T}} + \mathbf{A}^{-1}\boldsymbol{\Gamma}\mathbf{Q}(t_f - 1)\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{-\mathrm{T}}, \tag{4.82}$$

as in the forward-model computation. This backwards computation can be continued to time t_c, at which point we will have an estimate, $\tilde{\mathbf{x}}_b(t_c)$, with uncertainty $\mathbf{P}_b(t_c)$.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

The two independent estimates of $\mathbf{x}(t_c)$ can be combined to make an improved estimate using the relations Chapter 2, Eq. (2.444),

$$\tilde{\mathbf{x}}(t_c, +) = \tilde{\mathbf{x}}(t_c) + \mathbf{P}(t_c)(\mathbf{P}(t_c) + \mathbf{P}_b(t_c))^{-1}(\tilde{\mathbf{x}}_b(t_c) - \tilde{\mathbf{x}}(t_c)), \tag{4.83}$$

and (Eq. 2.446)

$$\begin{aligned}\mathbf{P}(t_c) &= \langle[\tilde{\mathbf{x}}(t_c, +) - \mathbf{x}(t_c)]\,[\tilde{\mathbf{x}}(t_c, +) - \mathbf{x}(t_c)]^{\mathrm{T}}\rangle \\ &= [\mathbf{P}(t_c)^{-1} + \mathbf{P}_b(t_c)^{-1}]^{-1}.\end{aligned} \tag{4.84}$$

This estimate is the same as would be obtained from the RTS algorithm run back to time t_c – because the same objective function, model, and data have been employed. The computation has been organized differently in the two cases. The backwards-running computation can be used at all points of the interval, as long as the data used in the forward and backwards computations are kept disjoint so that the two estimates are uncorrelated.

Running a model backwards in time may indeed be unstable if it contains any dissipative terms. A forward model may be unstable too, if there are unstable elements, either real ones or numerical artifacts. But the expressions in (4.83) and (4.84) are stable, because the computation of $\mathbf{P}_b(t)$ and its use in the updating expression (4.83) automatically downweights unstable elements whose errors will be very large, and which will not carry useful information from the later state concerning the earlier one. The same situation would occur if the forward model had unstable elements – these instabilities would amplify slight errors in the statement of their initial conditions, rendering the initial conditions difficult to estimate from observations at later times. Examination of the covariance propagation equation and the filter gain matrix shows that these elements are suppressed by the Kalman filter, with correspondingly large uncertainties. The filter/smoother formalism properly accounts for unstable, and hence difficult-to-calculate parameters, by estimating their uncertainty as very large, thus handling very general ill-conditioning. In practice, one needs to be careful, for numerical reasons, of the pitfalls in computing and using matrices that may have norms growing exponentially in time. But the conceptual problem is solved. As with the Kalman filter, it is possible to rewrite the RTS smoother expressions (4.71)–(4.74) in various ways for computational efficiency, storage reduction, and improved accuracy.[22]

The dominant computational load in the smoother is again the calculation of the updated covariance matrices, whose size is square of the state-vector dimension, at every time step, leading to efforts to construct simplified algorithms that retain most of the virtues of the filter/smoother combination but with reduced load. For example, it may have already occurred to the reader that in some of the examples displayed, the state vector uncertainties, $\mathbf{P}$, in both the filter and the smoother

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

appear to rapidly approach a steady state. This asymptotic behavior in turn means that the gain matrices, $\mathbf{K}$, $\mathbf{L}$, $\mathbf{M}$ will also achieve a steady state, implying that one no longer needs to undertake the updating steps – fixed gains can be used. Such steady-state operators are known as "Wiener filters" and "smoothers" and they represent a potentially very large computational saving. One needs to understand the circumstances under which such steady states can be expected to appear, and we will examine the problem in Section 4.5.

4.3.4 Other smoothers

The RTS algorithm is an example of what is usually called a "fixed-interval" smoother because it assumed that the results are required for a particular interval $t = 0, 1, \ldots, t_f$. Other forms for other purposes are described in the literature, including "fixed-lag" smoothers in which one is interested in an estimate at a fixed time $t_f - t_1$ as t_f advances, usually in real-time. A "fixed-point" smoother addresses the problem of finding a best estimate $\tilde{\mathbf{x}}(t_1)$ with t_1 fixed and t_f continually increasing. When $t_1 = 0$, as data accumulates, the problem of estimating the initial conditions is a special case of the fixed-point smoother problem.

4.4 Control and estimation problems

4.4.1 Lagrange multipliers and adjoints

The results of the last section are recursive schemes for computing first a filtered, and then a smoothed estimate. As with recursive least-squares, the combination of two pieces of information to make an improved estimate demands knowledge of the uncertainty of the information. For static problems, the recursive methods of Chapter 2 may be required, either because all the data were not available initially or because one could not handle them all at once. But, in general, the computational load of the combined least-squares problem introduced in Chapter 2, Eq. (2.424), is less than the recursive one, if one chooses not to compute any of the covariance matrices.

Because the covariance computation will usually dominate, and potentially overwhelm, the filter/smoother algorithms, it is at least superficially very attractive to find algorithms that do not require the covariances – that is, which employ the entire time domain of observations simultaneously – a "whole-domain" or "batch" method. The algorithms that emerge are best known in the context of "control theory." Essentially, there is a more specific focus upon determining the $\mathbf{u}(t)$: the control variables making a system behave as desired. Conventional control engineering has been directed at finding the electrical or physical impulses, e.g., to make

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575099.005

a robotic machine tool assemble an automobile, to land an airplane at a specified airfield, or to shift the output of a chemical plant. Because the motion of an airplane is described by a set of dynamical equations, the solution to the problem can equally well be thought of as making a *model* behave as required instead of the actual physical system. Thus if one observes a fluid flow, one whose behavior differs from what one's model said it should, we can seek those controls (e.g., boundary or initial conditions or internal parameters) that will force the model to be consistent with the observed behavior. It may help the reader who further explores these methods to recognize that we are still doing *estimation*, combining observations and models, but sometimes using algorithms best known under the control rubric.

To see the possibilities, consider again the two-point objective function (4.61) where **P**, etc., are just weight matrices, not necessarily having a statistical significance. We wish to find the minimum of the objective function subject to (4.62). Now append the model equations as done in Chapter 2 (as in Eq. (2.149)), with a vector of Lagrange multipliers, $\boldsymbol{\mu}(1)$, for a new objective function,

$$\begin{aligned} J &= (\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0))^{\mathrm{T}}\, \mathbf{P}(0)^{-1}(\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)) \\ &\quad + (\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0))^{\mathrm{T}}\mathbf{Q}(0)^{-1}(\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)) \\ &\quad + (\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1))^{\mathrm{T}}\mathbf{R}(1)^{-1}(\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)) \\ &\quad - 2\boldsymbol{\mu}(1)^{\mathrm{T}}[\tilde{\mathbf{x}}(1) - \mathbf{A}\tilde{\mathbf{x}}(0,+) - \mathbf{B}\mathbf{q}(0) - \boldsymbol{\Gamma}\tilde{\mathbf{u}}(0,+)]. \end{aligned} \tag{4.85}$$

As with the filter and smoother, the model is being imposed as a hard constraint, *but with the control term permitting the model to be imperfect*. Tildes have once again been placed over all variables to be estimated. The presence of the Lagrange multiplier now permits treating the differentials as independent; taking the derivatives of J with respect to $\tilde{\mathbf{x}}(0,+)$, $\tilde{\mathbf{x}}(1)$, $\tilde{\mathbf{u}}(0,+)$, $\boldsymbol{\mu}(1)$ and setting them to zero,

$$\mathbf{P}(0)^{-1}\,[\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)] + \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(1) = \mathbf{0}, \tag{4.86}$$

$$\mathbf{E}\,(1)^{\mathrm{T}}\,\mathbf{R}(1)^{-1}\,[\mathbf{y}(1) - \mathbf{E}\,(1)\,\tilde{\mathbf{x}}(1)] + \boldsymbol{\mu}(1) = \mathbf{0}, \tag{4.87}$$

$$\mathbf{Q}(0)^{-1}\,[\tilde{\mathbf{u}}(0,+) - \tilde{\mathbf{u}}(0)] + \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(1) = \mathbf{0}, \tag{4.88}$$

$$\tilde{\mathbf{x}}(1) - \mathbf{A}\tilde{\mathbf{x}}(0,+) - \mathbf{B}\mathbf{q}(0) - \boldsymbol{\Gamma}\tilde{\mathbf{u}}(0,+) = \mathbf{0}. \tag{4.89}$$

Equation (4.86) is the "adjoint model" for $\boldsymbol{\mu}(1)$ involving $\mathbf{A}^{\mathrm{T}}$.

Because the objective function in (4.85) is identical to that used with the smoother for this problem, and because the identical dynamical model has been imposed, Eqs. (4.86)–(4.89) must produce the same solution as that given by the smoother. A demonstration that Eqs. (4.86)–(4.89) can be manipulated into the form of (4.69) and (4.71) is an exercise in matrix identities.[23] As with smoothing algorithms, finding the solution of (4.86)–(4.89) can be done in a number of different ways, trading computation against storage, coding ease, convenience, etc.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005

Let us show explicitly the identity of smoother and Lagrange multiplier methods for a restricted case – that for which the initial conditions are known exactly, so that $\tilde{\mathbf{x}}(0)$ is not modified by the later observations. For the one-term smoother, the result is obtained by dropping (4.86), as $\tilde{\mathbf{x}}(0)$ is no longer an adjustable parameter. Without further loss of generality, put $\tilde{\mathbf{u}}(0) = \mathbf{0}$, and set $\mathbf{R}(1) = \mathbf{R}$, reducing the system to

$$\tilde{\mathbf{x}}(1) = \mathbf{A}\tilde{\mathbf{x}}(0) + \mathbf{B}\mathbf{q}(0) + \boldsymbol{\Gamma}\tilde{\mathbf{u}}(0, +), \tag{4.90}$$

$$\begin{aligned} \tilde{\mathbf{u}}(0, +) &= -\mathbf{Q}(0)\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(1) \\ &= \mathbf{Q}(0)\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y}(1) - \mathbf{E}(1)\tilde{\mathbf{x}}(1)]. \end{aligned} \tag{4.91}$$

Eliminating $\tilde{\mathbf{u}}(0, +)$ from (4.90) produces

$$\tilde{\mathbf{x}}(1) = \mathbf{A}\tilde{\mathbf{x}}(0) + \mathbf{B}\mathbf{q}(0) + \boldsymbol{\Gamma}\mathbf{Q}(0)\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y}(1) - \mathbf{E}\tilde{\mathbf{x}}(1)]. \tag{4.92}$$

With no initial error in $\mathbf{x}(0)$, $\mathbf{P}(1, -) = \boldsymbol{\Gamma}\mathbf{Q}(0)\boldsymbol{\Gamma}^{\mathrm{T}}$, and with

$$\tilde{\mathbf{x}}(1, -) \equiv \mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{q}(0), \tag{4.93}$$

(4.92) can be written as

$$[\mathbf{I} + \mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{E}]\tilde{\mathbf{x}}(1) = \tilde{\mathbf{x}}(1, -) + \mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}(1), \tag{4.94}$$

or (factoring $\mathbf{P}(1, -)$)

$$\begin{aligned} \tilde{\mathbf{x}}(1) = {}& [\mathbf{P}(1, -)^{-1} + \mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{E}]^{-1}\mathbf{P}(1, -)^{-1}\tilde{\mathbf{x}}(1, -) \\ &+ [\mathbf{P}(1, -)^{-1} + \mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{E}]^{-1}\mathbf{E}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}(1). \end{aligned}$$

Applying the matrix inversion lemma in the form (2.35), to the first term on the right, and in the form (2.36) to the second term on the right,

$$\begin{aligned} \tilde{\mathbf{x}}(1) = {}& \{\mathbf{P}(1, -) - \mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}[\mathbf{E}\mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}} + \mathbf{R}]^{-1}\mathbf{E}\mathbf{P}(1, -)\} \\ &\times \mathbf{P}(1, -)^{-1}\tilde{\mathbf{x}}(1, -) + \mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}[\mathbf{R} + \mathbf{E}\mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}]^{-1}\mathbf{y}(1), \end{aligned} \tag{4.95}$$

or

$$\tilde{\mathbf{x}}(1) = \tilde{\mathbf{x}}(1, -) + \mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}}[\mathbf{E}\mathbf{P}(1, -)\mathbf{E}^{\mathrm{T}} + \mathbf{R}]^{-1}[\mathbf{y}(1) - \mathbf{E}\tilde{\mathbf{x}}(1, -)]. \tag{4.96}$$

This last result is the ordinary Kalman filter estimate, as it must be, but it results here from the Lagrange multiplier formalism.

Now consider this approach for the entire interval $t = 0, 1, \ldots, t_f$. Start with the objective function (4.41) and append the model consistency demand using Lagrange

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

multipliers:

$$
\begin{aligned}
J = {} & [\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)]^{\mathrm{T}}\, \mathbf{P}(0)^{-1}\, [\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)] \\
& + \sum_{t=1}^{t_f} [\mathbf{y}(t) - \mathbf{E}(t)\tilde{\mathbf{x}}(t,+)]^{\mathrm{T}} \mathbf{R}(t)^{-1} [\mathbf{y}(t) - \mathbf{E}(t)\tilde{\mathbf{x}}(t,+)] \\
& + \sum_{t=0}^{t_f-1} \tilde{\mathbf{u}}(t,+)^{\mathrm{T}} \mathbf{Q}(t)^{-1} \tilde{\mathbf{u}}(t,+) \\
& - 2\sum_{t=1}^{t_f} \boldsymbol{\mu}(t)^{\mathrm{T}} [\tilde{\mathbf{x}}(t,+) - \mathbf{A}\tilde{\mathbf{x}}(t-1,+) - \mathbf{B}\mathbf{q}(t-1,+) - \boldsymbol{\Gamma}\tilde{\mathbf{u}}(t-1,+)].
\end{aligned}
\tag{4.97}
$$

Note the differing summation limits.

Note on notation Equation (4.97) has been written with $\tilde{\mathbf{x}}(t,+)$, $\tilde{\mathbf{u}}(t,+)$ to make it clear that the estimates will be based upon all data, past and future. But unlike the filter/smoother algorithm, there will only be a single estimated value, instead of the multiple estimates previously computed: $\tilde{\mathbf{x}}(t,-)$ (from the model forecast), $\tilde{\mathbf{x}}(t)$ (from the Kalman filter), and $\tilde{\mathbf{x}}(t,+)$ from the smoother, and similarly for $\mathbf{u}(t)$. Of necessity, $\tilde{\mathbf{x}}(t_f,+) = \tilde{\mathbf{x}}(t)$ from the Kalman filter. $\mathbf{x}_0$ is any initial condition estimate with uncertainty $\mathbf{P}(0)$ obtained from any source.

Setting all the derivatives to zero gives the normal equations:

$$\frac{1}{2}\frac{\partial J}{\partial \tilde{\mathbf{u}}(t,+)} = \mathbf{Q}(t)^{-1}\tilde{\mathbf{u}}(t,+) + \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t+1) = 0,\ t = 0, 1, \ldots, t_f - 1, \tag{4.98}$$

$$\frac{1}{2}\frac{\partial J}{\partial \boldsymbol{\mu}(t)} = \tilde{\mathbf{x}}(t,+) - \mathbf{A}\tilde{\mathbf{x}}(t-1,+) - \mathbf{B}\mathbf{q}(t-1) - \boldsymbol{\Gamma}\tilde{\mathbf{u}}(t-1,+) = 0, \quad t = 0, 1, \ldots, t_f \tag{4.99}$$

$$\frac{1}{2}\frac{\partial J}{\partial \tilde{\mathbf{x}}(0,+)} = \mathbf{P}(0)^{-1}\big(\tilde{\mathbf{x}}(0,+) - \tilde{\mathbf{x}}(0)\big) + \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(1) = 0, \tag{4.100}$$

$$\frac{1}{2}\frac{\partial J}{\partial \tilde{\mathbf{x}}(t,+)} = -\mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{y}(t) - \mathbf{E}(t)\tilde{\mathbf{x}}(t,+)] - \boldsymbol{\mu}(t) + \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t+1) = 0, \quad t = 1, 2, \ldots, t_f - 1, \tag{4.101}$$

$$\frac{1}{2}\frac{\partial J}{\partial \tilde{\mathbf{x}}(t_f)} = -\mathbf{E}(t_f)^{\mathrm{T}}\mathbf{R}(t_f)^{-1}[\mathbf{y}(t_f) - \mathbf{E}(t_f)\tilde{\mathbf{x}}(t_f)] - \boldsymbol{\mu}(t_f) = 0, \tag{4.102}$$

where the derivatives for $\tilde{\mathbf{x}}(t,+)$, at $t = 0$, $t = t_f$, have been computed separately for clarity. The so-called adjoint model is now given by (4.101). An equation count shows that the number of equations is exactly equal to the number of unknowns

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

$[\tilde{\mathbf{x}}(t,+),\ \tilde{\mathbf{u}}(t,+),\ \boldsymbol{\mu}(t)]$. With a large enough computer, we could contemplate solving them all at once. But for real fluid models with large time spans and large state vectors, even the biggest supercomputers are swamped, and one needs to find other methods.

The adjoint model in Eq. (4.101) is

$$\boldsymbol{\mu}(t) = \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t+1) + \mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{E}(t)\tilde{\mathbf{x}}(t,+) - \mathbf{y}(t)],$$

in which the model/data misfit appears as a "source term" (compare Eq. (2.355), noting that the $\mathbf{A}$ matrices are defined differently). It is sometimes said that time runs backwards in this equation, with $\boldsymbol{\mu}(t)$ being computed most naturally from $\boldsymbol{\mu}(t+1)$ and the source term, with Eq. (4.102) providing an initial condition. But in fact, time has no particular direction here, as the equations govern a time interval, $t = 1, 2, \ldots, t_f$. Indeed if $\mathbf{A}^{-1}$ exists, there is no problem in rewriting Eq. (4.101) so that $\boldsymbol{\mu}(t+1)$ is given in terms of $\mathbf{A}^{-\mathrm{T}}\boldsymbol{\mu}(t)$.

The Lagrange multipliers – that is, the adjoint solution – have the same interpretation that they did for the steady models described in Chapter 2 – that is, as a measure of the objective function sensitivity to the data,

$$\frac{\partial J'}{\partial \mathbf{Bq}(t)} = 2\boldsymbol{\mu}(t+1). \tag{4.103}$$

The physics of the adjoint model, as in Chapter 2, are again represented by the matrix $\mathbf{A}^{\mathrm{T}}$. For a forward model that is both linear and self-adjoint ($\mathbf{A}^{\mathrm{T}} = \mathbf{A}$), the adjoint solution would have the same physical behavior as the state vector. If the model is not self-adjoint (the usual situation), the evolution of the $\boldsymbol{\mu}(t)$ may have a radically different interpretation than $\mathbf{x}(t)$. Insight into that physics is the road to understanding of information flow in the system. For example, if one employed a large numerical model to compute the flux of heat in a fluid, and wished to understand the extent to which the result was sensitive to the boundary conditions, or to a prescribed flux somewhere, the adjoint solution carries that information. In the future, one expects to see display and discussion of the results of the adjoint model on a nearly equal footing with that of the forward model.

4.4.2 Terminal constraint problem: open-loop control

Consider the adjoint approach in the context of the simple tracer box model already described and depicted in Fig. 4.8. At $t = 0$, the tracer concentrations in the boxes are known to vanish – that is, $\mathbf{x}(0) = \mathbf{x}_0 = \mathbf{0}$ (the initial conditions are supposedly known exactly). At $t = t_f$, a survey is made of the region, and the concentrations $\mathbf{y}(t_f) = \mathbf{E}(t_f)\mathbf{x}(t_f) + \mathbf{n}(t_f)$, $\mathbf{E}(t_f) \equiv \mathbf{I}$, $\langle\mathbf{n}(t)\rangle = \mathbf{0}$, $\langle\mathbf{n}(t_f)\mathbf{n}(t_f)^{\mathrm{T}}\rangle = \mathbf{R}$ are known. No other observations are available. The question posed is: If the boundary

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

conditions are all unknown a priori – that is, $\mathbf{Bq} \equiv \mathbf{0}$, and all boundary conditions are control variables – what boundary conditions would produce the observed values at t_f within the estimated error bars? Write $\mathbf{x}_d = \mathbf{y}(t_f)$ – denoting the "desired" final state.

The problem is an example of a "terminal constraint control problem" – it seeks controls (forces, etc.) able to drive the system from an observed initial state, here zero concentration, to within a given tolerance of a required terminal state, $\mathbf{x}_d$.[24] (The control literature refers to the "Pontryagin Principle.") But in the present context, we interpret the result as an *estimate* of the actual boundary condition with uncertainty $\mathbf{R}(t_f)$. For this special case, take the objective function,

$$\begin{aligned} J = [\tilde{\mathbf{x}}(t_f) - \mathbf{x}_d]^{\mathrm{T}}\mathbf{R}(t_f)^{-1}[\tilde{\mathbf{x}}(t_f) - \mathbf{x}_d] + \sum_{t=0}^{t_f-1} \tilde{\mathbf{u}}^{\mathrm{T}}(t)\mathbf{Q}(t)^{-1}\tilde{\mathbf{u}}(t) \\ - 2\sum_{1}^{t_f} \boldsymbol{\mu}(t)^{\mathrm{T}}[\tilde{\mathbf{x}}(t) - \mathbf{A}\tilde{\mathbf{x}}(t-1) - \mathbf{Bq}(t-1) - \boldsymbol{\Gamma}\tilde{\mathbf{u}}(t-1)]. \end{aligned} \tag{4.104}$$

From here on, the notation $\tilde{\mathbf{x}}(t,+)$, $\tilde{\mathbf{u}}(t,+)$ in objective functions is suppressed, using $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{u}}(t)$ with the understanding that any solution is an estimate, from whatever data are available, past, present, or future. The governing normal equations are

$$\boldsymbol{\mu}(t-1) = \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t), \quad t = 1, 2, \ldots, t_f, \tag{4.105}$$

$$\boldsymbol{\mu}(t_f) = \mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f) - \mathbf{x}_d), \tag{4.106}$$

$$\mathbf{Q}(t)^{-1}\tilde{\mathbf{u}}(t) = -\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t+1), \tag{4.107}$$

plus the model. Eliminating

$$\tilde{\mathbf{u}}(t) = -\mathbf{Q}(t)\,\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t+1), \tag{4.108}$$

and substituting into the model, the system to be solved is

$$\tilde{\mathbf{x}}(t) = \mathbf{A}\tilde{\mathbf{x}}(t-1) - \boldsymbol{\Gamma}\mathbf{Q}(t)\,\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t), \qquad \tilde{\mathbf{x}}(0) = \mathbf{x}_0 \equiv \mathbf{0}, \tag{4.109}$$

$$\boldsymbol{\mu}(t-1) = \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t), \quad t = 1, 2, \ldots, t_f - 1, \tag{4.110}$$

$$\boldsymbol{\mu}(t_f) = \mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f) - \mathbf{x}_d). \tag{4.111}$$

As written, this coupled problem has natural initial conditions for the state vector, $\mathbf{x}(t)$, at $t = 0$, and for $\boldsymbol{\mu}(t)$ at $t = t_f$, but with the latter in terms of the still unknown $\mathbf{x}(t_f)$ – recognizing that the estimated terminal state and the desired one will almost always differ, that is, $\tilde{\mathbf{x}}(t_f) \neq \mathbf{x}_d$.

By exploiting its special structure, this problem can be solved in straightforward fashion without having to deal with the giant set of simultaneous equations.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

Using (4.111), step backwards in time from t_f via (4.110) to produce

$$\begin{aligned}\boldsymbol{\mu}(t_f) &= \mathbf{A}^{\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d),\\ &\vdots\\ \boldsymbol{\mu}(1) &= \mathbf{A}^{(t_f)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d),\end{aligned} \tag{4.112}$$

so that $\boldsymbol{\mu}(t)$ is given in terms of the known $\mathbf{x}_d$ and the still unknown $\tilde{\mathbf{x}}(t_f)$. Substituting into (4.109) generates

$$\begin{aligned}\tilde{\mathbf{x}}(1) &= \mathbf{A}\tilde{\mathbf{x}}(0) - \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d),\\ \tilde{\mathbf{x}}(2) &= \mathbf{A}\tilde{\mathbf{x}}(1) - \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-2)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d),\\ &= \mathbf{A}^2\tilde{\mathbf{x}}(0) - \mathbf{A}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d)\\ &\quad - \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-2)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d),\\ &\vdots\\ \tilde{\mathbf{x}}(t_f) &= \mathbf{A}^{t_f}\tilde{\mathbf{x}}(0) - \mathbf{A}^{(t_f-1)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d)\\ &\quad - \mathbf{A}^{(t_f-2)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-2)\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d)\\ &\quad - \cdots - \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{R}^{-1}(\tilde{\mathbf{x}}(t_f)-\mathbf{x}_d).\end{aligned} \tag{4.113}$$

The last equation permits us to bring the terms in $\tilde{\mathbf{x}}(t_f)$ over to the left-hand side and solve for $\tilde{\mathbf{x}}(t_f)$ in terms of $\mathbf{x}_d$ and $\tilde{\mathbf{x}}(0)$:

$$\begin{aligned}&\left\{\mathbf{I} + \mathbf{A}^{(t_f-1)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}}\mathbf{R}^{-1}\right.\\ &\quad \left. + \mathbf{A}^{(t_f-2)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-2)\mathrm{T}}\mathbf{R}^{-1} + \cdots + \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{R}^{-1}\right\}\tilde{\mathbf{x}}(t_f)\\ &= \mathbf{A}^{t_f}\tilde{\mathbf{x}}(0) + \left\{\mathbf{A}^{(t_f-1)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}}\mathbf{R}^{-1}\right.\\ &\qquad \left. + \mathbf{A}^{(t_f-2)}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-2)\mathrm{T}}\mathbf{R}^{-1} + \cdots + \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{R}^{-1}\right\}\mathbf{x}_d.\end{aligned} \tag{4.114}$$

With $\tilde{\mathbf{x}}(t_f)$ now known, $\boldsymbol{\mu}(t)$ can be computed for all t from (4.110) and (4.111). Then the control $\tilde{\mathbf{u}}(t)$ is also known from (4.107) and the state vector can be found from (4.109). The resulting solution for $\tilde{\mathbf{u}}(t)$ is in terms of the externally prescribed $\tilde{\mathbf{x}}(0)$, $\mathbf{x}_d$ and is usually known as "open-loop" control.

The canonical form for a terminal constraint problem usually used in the control literature differs slightly; it is specified in terms of a given, non-zero, initial condition $\mathbf{x}(0)$, and the controls are determined so as to come close to a desired zero terminal state. By linearity, the solution to this so-called deadbeat control (driving the system to rest) problem can be used to solve the problem for an arbitrary desired terminal state.

Example *Consider the tracer forward problem in Fig. 4.8 where only boundary box 2 now has a non-zero concentration, fixed at $C = 1$, starting at $t = 1$. A concentration is readily imposed by zeroing the corresponding row of* $\mathbf{A}$, *so that* $\mathbf{Bq}\,(t)$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

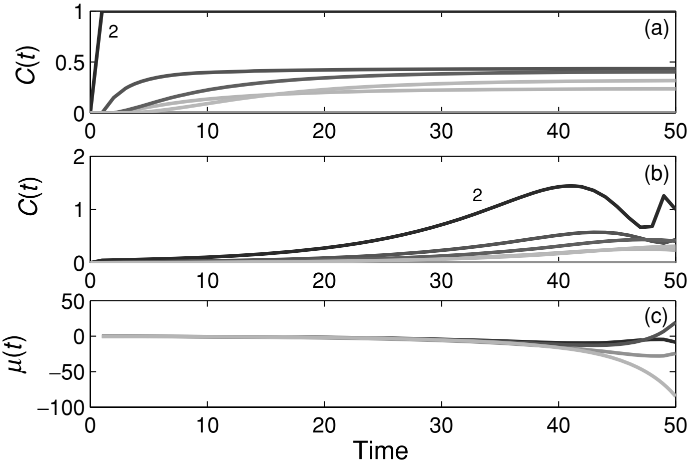

Figure 4.12 Box model example of terminal control. Here the "forward" calculation fixes the concentration in boundary box number 2 as $C = 1$, and all other boundary box concentrations are fixed at zero. (a) The box 2 and interior box concentrations for 50 time-steps with initial condition of zero concentration everywhere. (b) The estimated concentration from the terminal control calculation, in which $\mathbf{R} = 10^{-4}\mathbf{I}$, $\mathbf{Q} = 1$, where the only control value was the box 2 concentration. Thus a slight misfit is permitted to the terminal values $\mathbf{C}(50\Delta t)$, $\Delta t = 0.05$. (c) The Lagrange multipliers (adjoint solution) corresponding to the interior boxes. Having the largest values near the termination point is characteristic, and shows the sensitivity to the near terminal times of the constraints.

or $\boldsymbol{\Gamma}\mathbf{u}(t)$ set the concentration. (An alternative is to put the imposed concentration into the initial conditions and use the corresponding row of **A** *to force the concentration to be exactly that in the previous time step.) The initial conditions were taken as zero and the forward solution is in Fig. 4.12. Then the same figure shows the solution to the terminal time control problem for the concentration in box 2 giving rise to the terminal values. A misfit was permitted between the desired (observed) and calculated terminal times – with an RMS value of* 2×10^{-4}. *Clearly the "true" solution is underdetermined by the provision of initial and terminal time tracer concentrations alone. Also shown in the figure are the Lagrange multipliers (adjoint solution) corresponding to the model equations for each box.*[25]

In the above formulation, the boundary boxes were contained in the **A** matrix, but the corresponding rows were all zero, permitting the **B** matrix (here a vector) to control the boundary box concentrations. A variation on this problem is obtained by setting column element j_0 corresponding to boundary box j_0, in **A** to unity. **B** would then control the time rate of change of the boundary box concentrations. Suppose then that **B** is a column vector, vanishing in all elements except for unity

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

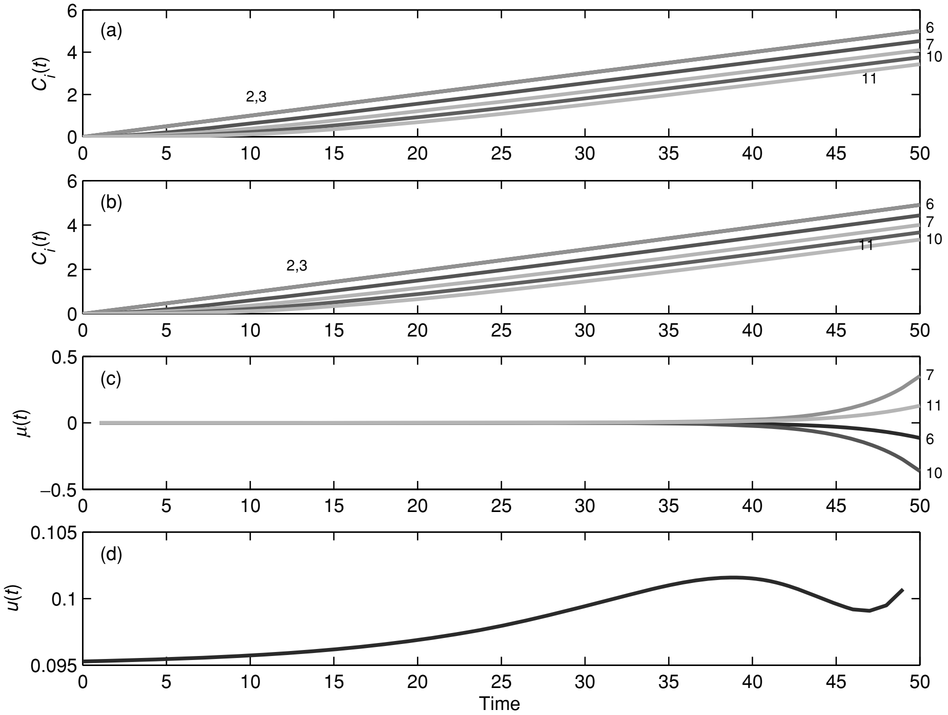

Figure 4.13 The same box model as in Fig. 4.12, except that now $\mathbf{\Gamma u}(t)$ controls the rate of change of concentration rather than concentration itself, and all boundary boxes have a constant rate of change of 0.1. (a) “True” solution. (b) The solution deduced from the terminal state control, with a near-perfect requirement on the terminal values and $\mathbf{Q}$. (c) The Lagrange multipliers for the interior box constraints of the model in Fig. 4.8. (d) Estimated control $\tilde{\mathbf{u}}(t)$. Note the highly compressed amplitude scale.

in all active boundary boxes (the corner boxes are passive here). Then Fig. 4.13 shows the concentration and the result of the terminal control problem in this case.

The smoothing problem has been solved without having to compute the uncertainties, and is the major advantage of the Lagrange multiplier methods over the sequential estimators. Lagrange multiplier methods solve for the entire time domain at once; consequently, there is no weighted averaging of intermediate solutions and no need for the uncertainties. On the other hand, the utility of solutions without uncertainty estimates must be questioned.

In the context of Chapter 1, problems of arbitrary posedness are being solved. The various methods using objective functions, prior statistics, etc., whether in time-evolving or static situations, permit stable, useful estimates to be made under almost any circumstances, using almost any sort of available information. But the reader will by now appreciate that the use of such methods can produce structures in the solution, pleasing or otherwise, that may be present because they are required

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005

by (1) the observations, (2) the model, (3) the prior statistics, (4) some norm or smoothness demand on elements of the solution, or (5) all of the preceding in concert. A solution produced in ignorance of these differing sources of structure can hardly be thought very useful, and it is the uncertainty matrices that are usually the key to understanding. Consequently, we will later briefly examine the problem of obtaining the missing covariances. In the meantime, one should note that the covariances of the filter/smoother will also describe the uncertainty of the Lagrange multiplier method solution, because they are the same solution to the same set of equations deriving from the same objective function.

There is one situation where a solution without uncertainty estimates is plainly useful – it is where one simply inquires, "Is there a solution at all?" – that is, when one wants to know if the observations actually contradict the model. In that situation, mere existence of an acceptable solution may be of greatest importance, suggesting, for example, that a model of adequate complexity is already available and that the data errors are understood.

4.4.3 Representers and boundary Green functions

The particular structure of Eqs. (4.105)–(4.107) permits several different methods of solution, of which the version just given is an example. To generalize this problem, assume observations at a set of arbitrary times (not just the terminal time)

$$\mathbf{y}(t) = \mathbf{E}(t)\,\mathbf{x}(t) + \mathbf{n}(t),$$

and seek a solution in "representers."

Take the objective function to be

$$\begin{aligned} J = {} & \sum_{t=1}^{t_f} [\mathbf{y}(t) - \mathbf{E}(t)\,\mathbf{x}(t)]^{\mathrm{T}}\,\mathbf{R}(t)^{-1}\,[\mathbf{y}(t) - \mathbf{E}(t)\,\mathbf{x}(t)] \\ & + \sum_{t=0}^{t_f-1} \mathbf{u}(t)^{\mathrm{T}}\,\mathbf{Q}(t)^{-1}\,\mathbf{u}(t) - 2\sum_{t=1}^{t_f} \boldsymbol{\mu}(t)^{\mathrm{T}} \\ & \times\, [\mathbf{x}(t) - \mathbf{A}\mathbf{x}(t-1) - \mathbf{B}\mathbf{q}(t-1) - \boldsymbol{\Gamma}(t-1)\mathbf{u}(t-1)], \end{aligned} \tag{4.115}$$

so that the terminal state estimate is subsumed into the first term with $\mathbf{E}(t_f) = \mathbf{I}$, $\mathbf{R}(t_f) = \mathbf{P}(t_f)$. (The tildes are now being omitted.) Let $\mathbf{x}_a(t)$ be the (known) solution to the pure, unconstrained, forward problem,

$$\mathbf{x}_a(t) = \mathbf{A}\mathbf{x}_a(t-1) + \mathbf{B}\mathbf{q}(t-1),\;\; \mathbf{x}_a(0) = \mathbf{x}_0. \tag{4.116}$$

Redefine $\mathbf{x}(t)$ to be the difference $\mathbf{x}(t) \to \mathbf{x}(t) - \mathbf{x}_a(t)$, that is, the deviation from what can be regarded as the a-priori solution. The purpose of this redefinition is to remove any inhomogeneous initial or boundary conditions from the

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

problem – exploiting the system linearity. The normal equations are then

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{u}(t)} = \mathbf{Q}(t)^{-1}\mathbf{u}(t) + \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t+1) = \mathbf{0}, \quad t = 0, 1, \ldots, t_f - 1,$$

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{x}(t)} = \mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)] + \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t+1) - \boldsymbol{\mu}(t) = \mathbf{0},$$

$$t = 1, 2, \ldots, t_f,$$

$$\frac{1}{2}\frac{\partial J}{\partial \boldsymbol{\mu}(t)} = \mathbf{x}(t) - \mathbf{A}\mathbf{x}(t-1) - \boldsymbol{\Gamma}(t-1)\mathbf{u}(t-1) = \mathbf{0},\ \mathbf{x}(0) = \mathbf{0},$$

$$t = 1, 2, \ldots, t_f.$$

Eliminating the $\mathbf{u}(t)$ in favor of $\boldsymbol{\mu}(t)$, we have, as before,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1) - \boldsymbol{\Gamma}\mathbf{Q}(t-1)\boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t),\ \mathbf{x}(0) = \mathbf{0}, \tag{4.117}$$

$$\boldsymbol{\mu}(t) = \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t+1) + \mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)]. \tag{4.118}$$

The system is linear, so we can examine the solution forced by the inhomogeneous term in (4.118) at one time, $t = t_m$. This inhomogeneous term, $\mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}[\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)]$, in Eq. (4.118) is, however, unknown until $\mathbf{x}(t)$ has been determined. So to proceed, *first solve the different problem*,

$$\mathbf{M}(t, t_m) = \mathbf{A}^{\mathrm{T}}\mathbf{M}(t+1, t_m) + \mathbf{I}\delta_{t,t_m}, \quad t \le t_m, \tag{4.119}$$

$$\mathbf{M}(t, t_m) = 0, \quad t > t_m, \tag{4.120}$$

where the second argument, t_m, denotes the time of one set of observations (notice that $\mathbf{M}$ is a matrix). Time-step Eq. (4.119) backwards from $t = t_m$. There is then a corresponding solution to (4.117) with these values of $\boldsymbol{\mu}(t)$,

$$\mathbf{G}(t+1, t_m) = \mathbf{A}\mathbf{G}(t, t_m) - \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{M}(t+1, t_m), \tag{4.121}$$

which is stepped-forward in time starting with $\mathbf{G}(0, t_m) = \mathbf{0}$, until $t + 1 = t_m$. Both $\mathbf{G}$, $\mathbf{M}$ are computable independent of the actual data values. Now put

$$\mathbf{m}(t, t_m) = \mathbf{M}(t, t_m)\{\mathbf{E}(t_m)^{\mathrm{T}}\mathbf{R}(t_m)^{-1}[\mathbf{E}(t_m)\mathbf{x}(t_m) - \mathbf{y}(t_m)]\}, \tag{4.122}$$

which is a vector that, by linearity, $\boldsymbol{\mu}(t) = \mathbf{m}(t, t_m)$ is the solution to (4.118) once $\mathbf{x}(t_m)$ is known. Let

$$\boldsymbol{\xi}(t, t_m) = \mathbf{G}(t, t_m)\{\mathbf{E}(t_m)^{\mathrm{T}}\mathbf{R}(t_m)^{-1}[\mathbf{E}(t_m)\boldsymbol{\xi}(t_m, t_m) - \mathbf{y}(t_m)]\}, \tag{4.123}$$

which is another vector, such that $\tilde{\mathbf{x}}(t) = \boldsymbol{\xi}(t, t_m)$ would be the solution sought. Setting $t = t_m$ in Eq. (4.123) and solving,

$$\boldsymbol{\xi}(t_m, t_m) = -[\mathbf{I} - \mathbf{G}(t_m, t_m)\mathbf{E}(t_m)^{\mathrm{T}}\mathbf{R}(t_m)^{-1}\mathbf{E}(t_m)]^{-1} \times [\mathbf{G}(t_m, t_m)\mathbf{E}(t_m)^{\mathrm{T}}\mathbf{R}(t_m)^{-1}]\mathbf{y}(t_m). \tag{4.124}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

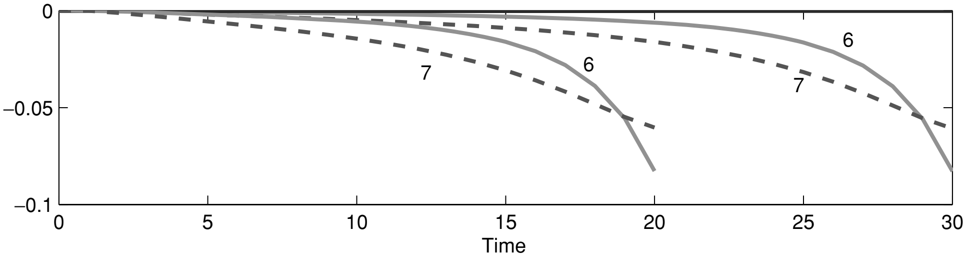

Figure 4.14 Representer (Green function) G for interior box 7, with the columns corresponding to boxes 6, 7 displayed through time. The Green function used numerically is the sum of these two, and displays a near-discontinuity (typical of Green functions) at the data points that are available at $t = 20, 30$.

With $\boldsymbol{\xi}(t_m, t_m)$ known, Eq. (4.123) produces a fully determined $\tilde{\mathbf{x}}(t) = \boldsymbol{\xi}(t, t_m)$ in representer form. This solution is evidently just a variant of Eqs. (4.113) and (4.114). Now suppose that there are multiple observations times, t_m. The problem is a linear one so that solutions can be superimposed,

$$\tilde{\mathbf{x}}(t) = \sum_{t_m} \boldsymbol{\xi}(t, t_m),$$

and after adding $\mathbf{x}_a(t)$ to the result, the entire problem is solved.

The solutions $\mathbf{M}(t, t_m)$ are the Green function for the adjoint model equation; the $\mathbf{G}(t, t_m)$ are "representers,"[26] and they exist independently of the data. *If the data distribution is spatially sparse, one need only compute the subsets of the columns or rows of* $\mathbf{M}, \mathbf{G}$ *that correspond to measured elements of* $\mathbf{x}(t)$. That is, in Eq. (4.119) any zero columns in $\mathbf{E}$, representing elements of the state vector not involved in the measurements, multiply the corresponding columns of $\mathbf{M}, \mathbf{G}$, and hence one need never compute those columns.

Example *Consider again the* 4×4 *box model of Fig. 4.8, in the same configuration as used above, with all the boundary boxes having a fixed tracer concentration of* $C = 1$, *and zero initial condition. Now, it is assumed that observations are available in all interior boxes (6, 7, 10, 11) at time* $t = 20, 30$. *The representer G is shown in Fig. 4.14.*

The representer emerged naturally from the Lagrange multiplier formulation. Let us re-derive the solution without the use of Lagrange multipliers to demonstrate how the adjoint model appears in unconstrained l_2 norm problems (soft constraints). Introduce the model into the same objective function as above, except we do it by substitution for the control terms; let $\boldsymbol{\Gamma} = \mathbf{I}$, making it possible to solve for $\mathbf{u}(t) = -[\mathbf{x}(t+1) - \mathbf{x}(t)]$ explicitly and producing the simplest results. The

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

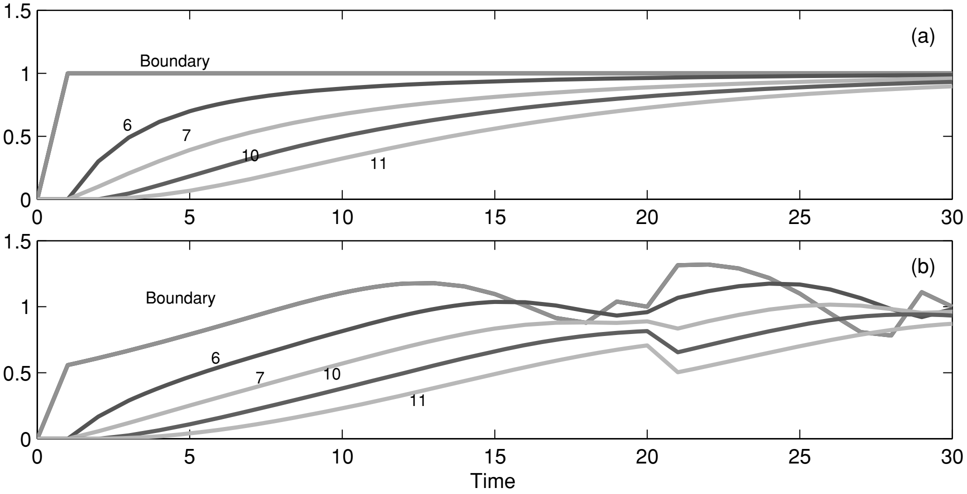

Figure 4.15 (a) The forward "truth" in the box model and (b) the estimated values from the representer displayed in Fig. 4.14. Data were treated as nearly perfect at the two observation times.

objective function then is

$$J = \sum_{t=0}^{t_f} [\mathbf{y}(t) - \mathbf{E}(t)\mathbf{x}(t)]^{\mathrm{T}} \mathbf{R}(t)^{-1} [\mathbf{y}(t) - \mathbf{E}(t)\mathbf{x}(t)] + \sum_{t=0}^{t_f-1} [\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)]^{\mathrm{T}} \mathbf{Q}(t)^{-1} [\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)]. \tag{4.125}$$

We again assume that $\mathbf{x}(t)$ is the anomaly relative to the known $\mathbf{x}_a(t)$.

The normal equations include:

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{x}(t)} = \mathbf{E}(t)^{\mathrm{T}} \mathbf{R}(t)^{-1} [\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)] - \mathbf{A}^{\mathrm{T}} \mathbf{Q}(t)^{-1} [\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)] + \mathbf{Q}(t)^{-1} [\mathbf{x}(t) - \mathbf{A}\mathbf{x}(t-1)] = 0. \tag{4.126}$$

Define

$$\boldsymbol{\nu}(t+1) = -\mathbf{Q}(t-1)^{-1} [\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t)], \tag{4.127}$$

so that the system (4.126) can be written as

$$\boldsymbol{\nu}(t) = \mathbf{A}^{\mathrm{T}} \boldsymbol{\nu}(t+1) + \mathbf{E}(t)^{\mathrm{T}} \mathbf{R}(t)^{-1} [\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)], \tag{4.128}$$

which, along with (4.127), is precisely the same system of equations (4.117) and (4.118) that emerged from the Lagrange multiplier approach, if we let $\boldsymbol{\mu} \to \boldsymbol{\nu}$, $\boldsymbol{\Gamma} = \mathbf{I}$. Representers are again defined as the unit disturbance solution

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

to the system. As a by-product, we see once again, that l_2-norm least-squares and the adjoint method are simply different algorithmic approaches to the same problem.[27]

4.4.4 The control Riccati equation

Consider yet another solution of the problem. (If the reader is wondering why such a fuss is being made about these equations, the answer, among others, is that it will turn out to be an important route to reducing the computational load required for the Kalman filter and various smoothing algorithms.) We look at the same special case of the objective function (4.104) and the equations that follow from it ((4.105)–(4.107) plus the model). Let $\mathbf{x}_d = \mathbf{0}$, the deadbeat requirement defined above. For this case, the adjoint equation is

$$\boldsymbol{\mu}(t) = \mathbf{A}^{\mathrm{T}}\boldsymbol{\mu}(t+1) + \mathbf{R}(t)^{-1}\mathbf{x}(t), \quad t = 1, 2, \ldots, t_f, \tag{4.129}$$

stipulating that $\mathbf{R}(t)^{-1} = \mathbf{0}$, $t \neq t_f$, if the only requirement is at the terminal time. For simplicity, let $\mathbf{Q}(t) = \mathbf{Q}$.

Take a trial solution, an “ansatz,” of the form

$$\boldsymbol{\mu}(t) = \mathbf{S}(t)\mathbf{x}(t), \tag{4.130}$$

where $\mathbf{S}(t)$ is unknown. Then Eq. (4.107) becomes

$$\mathbf{Q}^{-1}\mathbf{u}(t-1) + \boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t)\mathbf{x}(t) = \mathbf{0}, \tag{4.131}$$

or, using the model,

$$\mathbf{Q}^{-1}\mathbf{u}(t) + \boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)(\mathbf{A}\mathbf{x}(t) + \boldsymbol{\Gamma}\mathbf{u}(t)) = \mathbf{0}. \tag{4.132}$$

So that

$$\begin{aligned}\mathbf{u}(t) &= -\{\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\boldsymbol{\Gamma} + \mathbf{Q}^{-1}\}^{-1}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A}\mathbf{x}(t)\\ &= -\mathbf{L}(t+1)^{-1}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A}\mathbf{x}(t)\\ \mathbf{L}(t+1) &= \boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\boldsymbol{\Gamma} + \mathbf{Q}^{-1}.\end{aligned} \tag{4.133}$$

Substituting (4.133), and (4.130) for $\boldsymbol{\mu}(t)$, into the adjoint model (4.129),

$$\{\mathbf{A}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{S}(t+1)\boldsymbol{\Gamma}\mathbf{L}(t+1)^{-1}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A} - \mathbf{S}(t) + \mathbf{R}(t)^{-1}\}\,\mathbf{x}(t) = \mathbf{0}. \tag{4.134}$$

Unless $\mathbf{x}(t)$ is to vanish identically,

$$\mathbf{S}(t) = \mathbf{A}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{S}(t+1)\boldsymbol{\Gamma}\mathbf{L}(t+1)^{-1}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\mathbf{A} + \mathbf{R}(t)^{-1}. \tag{4.135}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005
Cambridge Books Online © Cambridge University Press, 2009

Equation (4.135) is a non-linear difference equation known as the matrix "Riccati equation," and produces a backwards recursion for $\mathbf{S}(t)$. Start the recursion with

$$\mathbf{S}(t_f)\mathbf{x}(t_f) = \mathbf{R}(t_f)^{-1}\mathbf{x}(t_f) \qquad \text{or} \qquad \mathbf{S}(t_f) = \mathbf{R}(t_f)^{-1}, \tag{4.136}$$

(recalling $\mathbf{x}_d = 0$) and step backwards to $t = 0$. The problem has now been solved by what is called the "sweep method."[28] Notice that with $\mathbf{S}(t)$ known, the control is in the form

$$\mathbf{\Gamma u}(t) = \mathbf{K}_c(t)\mathbf{x}(t). \tag{4.137}$$

This is known as "feedback control" because the values to be applied are determined by the value of the state vector at that time. It contrasts with the open-loop control form derived above, but necessarily produces an identical answer.

With feedback control, the computation of the model update step would now be

$$\mathbf{x}(t) = (\mathbf{A} - \mathbf{K}_c)\mathbf{x}(t-1) + \mathbf{Bq}(t-1). \tag{4.138}$$

The structure of the matrix,

$$\mathbf{A}' = \mathbf{A} - \mathbf{K}_c, \tag{4.139}$$

is the center of a discussion of the stability of the scheme, which we will not pursue here.

4.4.5 The initialization problem

Another special case of wide interest is determination of the initial conditions, $\tilde{\mathbf{x}}(0)$, from later observations. For notational simplicity and without loss of generality, assume that the known controls vanish so that the model is

$$\mathbf{x}(t) = \mathbf{Ax}(t-1) + \mathbf{\Gamma u}(t-1), \tag{4.140}$$

that there is an existing estimate of the initial conditions, $\tilde{\mathbf{x}}(0)$, with estimated uncertainty $\mathbf{P}(0)$, and that there is a single terminal observation of the complete state,

$$\mathbf{y}(t_f) = \mathbf{Ex}(t_f) + \mathbf{n}(t_f), \quad \mathbf{E} = \mathbf{I}, \tag{4.141}$$

where the observational noise covariance is again $\mathbf{R}(t_f)$. This problem can now be solved in five different ways:

1. The terminal observations can be written explicitly in terms of the initial conditions as

$$\begin{aligned}\mathbf{y}(t_f) = \mathbf{A}^{t_f}\tilde{\mathbf{x}}(0) + \mathbf{A}^{t_f-1}\mathbf{\Gamma}\tilde{\mathbf{u}}(0) + \mathbf{A}^{t_f-2}\mathbf{\Gamma}\tilde{\mathbf{u}}(1) + \cdots \\ + \mathbf{\Gamma}\tilde{\mathbf{u}}(t_f-1) + \mathbf{n}(t_f),\end{aligned} \tag{4.142}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

which, in canonical observation equation form, is

$$\mathbf{y}(t_f) = \mathbf{E}_p\tilde{\mathbf{x}}(0) + \mathbf{n}_p(t_f), \quad \mathbf{E}_p = \mathbf{A}^{t_f},$$
$$\mathbf{n}_p = \mathbf{A}^{t_f-1}\boldsymbol{\Gamma}\tilde{\mathbf{u}}(0) + \cdots + \boldsymbol{\Gamma}\tilde{\mathbf{u}}(t_f - 1) + \mathbf{n}(t_f),$$

and where the covariance of this combined error is

$$\mathbf{R}_p \equiv \left\langle \mathbf{n}_p\mathbf{n}_p^{\mathrm{T}} \right\rangle = \mathbf{A}^{t_f-1}\boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{A}^{(t_f-1)\mathrm{T}} + \cdots + \boldsymbol{\Gamma}\mathbf{Q}\boldsymbol{\Gamma}^{\mathrm{T}} + \mathbf{R}(t_f). \tag{4.143}$$

Then the least-squares recursive solution leads to

$$\tilde{\mathbf{x}}(0,+) = \tilde{\mathbf{x}}(0) + \mathbf{P}(0)\mathbf{E}_p^{\mathrm{T}}\left[\mathbf{E}_p\mathbf{P}(0)\mathbf{E}_p^{\mathrm{T}} + \mathbf{R}_p\right]^{-1}[\mathbf{y}(t_f) - \mathbf{E}_p\tilde{\mathbf{x}}(0)], \tag{4.144}$$

and the uncertainty estimate follows immediately.

2. A second method (which the reader should confirm produces the same answer) is to run the Kalman filter forward to t_f and then run the smoother backwards to $t = 0$. There is more computation here, but a by-product is an estimate of the intermediate values of the state vectors, of the controls, and their uncertainty.
3. Write the model in backwards form,

$$\mathbf{x}(t) = \mathbf{A}^{-1}\mathbf{x}(t+1) - \mathbf{A}^{-1}\boldsymbol{\Gamma}\mathbf{u}, \tag{4.145}$$

and use the Kalman filter on this model, with time running backwards. The observation equation (4.141) provides the initial estimate of $\mathbf{x}(t_f)$, and its error covariance becomes the initial estimate covariance $\mathbf{P}(t_f)$. At $t = 0$, the original estimate of $\tilde{\mathbf{x}}(0)$ is treated as an observation, with uncertainty $\mathbf{P}(0)$ taking the place of the usual $\mathbf{R}$. The reader should again confirm that the answer is the same as in (1).

4. The problem has already been solved using the Lagrange multiplier formalism.
5. The Green function representation (4.32) is immediately solvable for $\tilde{\mathbf{x}}(0,+)$.

4.5 Duality and simplification: the steady-state filter and adjoint

For linear models, the Lagrange multiplier method and the filter/smoother algorithms produce identical solutions. In both cases, the computation of the uncertainty remains an issue – in the former case because it is not part of the solution, and in the latter because it can overwhelm the computation. However, if the uncertainty is computed for the sequential estimator solutions, it must also represent the uncertainty derived from the Lagrange multiplier principle. In the interests of gaining insight into both methods, and of ultimately finding uncertainty estimates, consider again the covariance propagation equations for the Kalman filter:

$$\mathbf{P}(t,-) = \mathbf{A}(t-1)\mathbf{P}(t-1)\mathbf{A}(t-1)^{\mathrm{T}} + \boldsymbol{\Gamma}(t-1)\mathbf{Q}(t-1)\boldsymbol{\Gamma}(t-1)^{\mathrm{T}}, \tag{4.146}$$

$$\mathbf{P}(t) = \mathbf{P}(t,-) - \mathbf{P}(t,-)\mathbf{E}(t)^{\mathrm{T}}[\mathbf{E}(t)\mathbf{P}(t,-)\mathbf{E}(t)^{\mathrm{T}} + \mathbf{R}(t)]^{-1}\mathbf{E}(t)\mathbf{P}(t,-), \tag{4.147}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Table 4.1. *Correspondences between the variables of the control formulation and that of the Kalman filter, which lead to the Riccati equation. Note that time runs backward for control cases and forward for the filter.*

Adjoint/control	Kalman filter
$\mathbf{A}$	$\mathbf{A}^{\mathrm{T}}$
$\mathbf{S}(t,-)$	$\mathbf{P}(t+1)$
$\mathbf{S}(t+1)$	$\mathbf{P}(t+1,-)$
$\mathbf{R}^{-1}$	$\Gamma\mathbf{Q}\Gamma^{\mathrm{T}}$
Γ	$\mathbf{E}^{\mathrm{T}}$
$\mathbf{Q}^{-1}$	$\mathbf{R}$

where $\mathbf{K}(t)$ has been written out. Make the substitutions shown in Table 4.1; the equations for evolution of the uncertainty of the Kalman filter are identical to those for the control matrix $\mathbf{S}(t)$, given in Eq. (4.135); hence, the Kalman filter covariance *also satisfies a matrix Riccati equation*. To see that, in Eq. (4.135)[29] put

$$\mathbf{S}(t,-) \equiv \mathbf{S}(t+1) - \mathbf{S}(t+1)\boldsymbol{\Gamma}[\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1)\boldsymbol{\Gamma} + \mathbf{Q}^{-1}]^{-1}\boldsymbol{\Gamma}^{\mathrm{T}}\mathbf{S}(t+1), \tag{4.148}$$

and then

$$\mathbf{S}(t) = \mathbf{A}^{\mathrm{T}}\mathbf{S}(t,-)\mathbf{A} + \mathbf{R}(t)^{-1}, \tag{4.149}$$

which correspond to Eqs. (4.146) and (4.147). Time runs backwards in the control formulation and forwards in the estimation problem, but this difference is not fundamental. The significance of this result is that simplifications and insights obtained from one problem can be employed on the other (some software literally makes the substitutions of Table 4.1 to compute the Kalman filter solution from the algorithm for solving the control Riccati equation).

This feature – that both problems produce a matrix Riccati equation – is referred to as the "duality" of estimation and control. It does *not* mean that they are the same problem; in particular, recall that the control problem is equivalent not to filtering, but to smoothing.

Covariances usually dominate the Kalman filter (and smoother) calculations and sometimes lead to the conclusion that the procedures are impractical. But as with all linear least-squares, like estimation problems, the state vector uncertainty does not depend upon the actual data values, only upon the prior error covariances. Thus, the filter and smoother uncertainties (and the filter and smoother gains) can be

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005
Cambridge Books Online © Cambridge University Press, 2009

computed in advance of the actual application to data, and stored. The computation can be done, e.g., by stepping through the recursion in Eqs. (4.146) and (4.147), starting from $t = 0$.

Furthermore, it was pointed out that, in Kalman filter problems, the covariances and Kalman gain can approach a steady state, in which $\mathbf{P}(t)$, $\mathbf{P}(t, -)$, $\mathbf{K}(t)$ become time independent. Physically, the growth in error from the propagation equation (4.146) is then just balanced by the reduction in uncertainty from the incoming data stream (4.147). This simple description supposes the data come in at every time-step; often the data appear only intermittently, but periodically, and the steady-state solution is periodic – errors displaying a characteristic saw-tooth structure between observation times.

If these steady-state values can be found, then the necessity to update the covariances and gain matrix disappears, and the computational load is much reduced, potentially by many orders of magnitude (see also Chapter 5). The equivalent steady state for the control problem is best interpreted in terms of the feedback gain control matrix, $\mathbf{K}_c$, which can also become time independent, meaning that the value of the control to be applied depends only upon the state observed at time t and need not be recomputed at each time step.

The great importance of steady-state estimation and control has led to a large number of methods for obtaining the solution of the various steady-state Riccati equations requiring one of ($\mathbf{S}(t) = \mathbf{S}(t-1)$, $\mathbf{S}(t, -) = \mathbf{S}(t-1, -)$, $\mathbf{P}(t) = \mathbf{P}(t-1)$, or $\mathbf{P}(t, -) = \mathbf{P}(t-1, -)$).[30] The steady-state equation is often known as the "algebraic Riccati equation."[31]

A steady-state solution to the Riccati equation corresponds not only to a determination of the steady-state filter and smoother covariances but also to the steady-state solution of the Lagrange multiplier normal equations – a so-called steady-state control. Generalizations to the steady-state problem exist; an important one is the possibility of a periodic steady state.[32]

Before seeking a steady-state solution, one must determine whether one exists. That no such solution will exist in general is readily seen by considering a physical system in which certain components (elements of the flow) are not readily observed. If these components are initialized with partially erroneous values, then, if they are unstable, they will grow without bound, and there will be no limiting asymptotic value for the uncertainty, which will also have to grow without bound. Alternatively, suppose that there are elements of the state vector whose values cannot be modified by the available control variables. Then no observations of the state vector produce information about the control variables; if the control vector uncertainty is described by $\mathbf{Q}$, then this uncertainty will accumulate from one time-step to another, growing without bound with the number of time steps.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

4.6 Controllability and observability

In addition to determining whether there exists a steady-state solution either to the control or estimation Riccati equations, there are many reasons for examining in some detail the existence of many of the matrix operations that have been employed routinely. Matrix inverses occur throughout the developments above, and the issue of whether they exist has been ignored. Ultimately, however, one must face up to questions of whether the computations are actually possible. The questions are intimately connected to some very useful structural descriptions of models and data that we will now examine briefly.

Controllability

Can controls can be found to drive a system from a given initial state $\mathbf{x}(0)$ to an arbitrary $\mathbf{x}(t_f)$? If the answer is "yes," the system is said to be *controllable*. To find an answer, consider for simplicity,[33] a model with $\mathbf{B} = \mathbf{0}$ and with the control, u, a scalar. Then the model time-steps can be written as

$$\begin{aligned}
\mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) + \boldsymbol{\Gamma}u(0),\\
\mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) + \boldsymbol{\Gamma}u(1),\\
&= \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\boldsymbol{\Gamma}u(0) + \boldsymbol{\Gamma}u(1),\\
&\vdots\\
\mathbf{x}(t_f) &= \mathbf{A}^{t_f}\mathbf{x}(0) + \sum_{j=0}^{t_f-1}\mathbf{A}^{t_f-1-j}\boldsymbol{\Gamma}u(j),\\
&= \mathbf{A}^{t_f}\mathbf{x}(0) + [\boldsymbol{\Gamma}\ \ \mathbf{A}\boldsymbol{\Gamma}\cdots\mathbf{A}^{t_f-1}\boldsymbol{\Gamma}]\begin{bmatrix} u(t_f-1)\\ \vdots\\ u(0)\end{bmatrix}.
\end{aligned}$$

To determine $u(t)$, we must be able to solve the system

$$[\boldsymbol{\Gamma}\ \ \mathbf{A}\boldsymbol{\Gamma}\cdots\mathbf{A}^{t_f-1}\boldsymbol{\Gamma}]\begin{bmatrix} u(t_f-1)\\ \vdots\\ u(0)\end{bmatrix} = \mathbf{x}(t_f) - \mathbf{A}^{t_f}\mathbf{x}(0), \tag{4.150}$$

or

$$\mathbf{C}\mathbf{u} = \mathbf{x}(t_f) - \mathbf{A}^{t_f}\mathbf{x}(0),$$

for $u(t)$. The state vector dimension is N; therefore, the dimension of $\mathbf{C}$ is N by the number of columns, t_f (a special case – with scalar $u(t)$, $\boldsymbol{\Gamma}$ is $N \times 1$). Therefore, Eq. (4.150) has no (ordinary) solution if t_f is less than N. If $t_f = N$ and $\mathbf{C}$ is non-singular – that is, of rank N – there is a unique solution, and the system is

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

controllable. If the dimensions of $\mathbf{C}$ are non-square, one could have a discussion, familiar from Chapter 2, of solutions for $u(t)$ with nullspaces present. If $t_f < N$, there is a nullspace of the desired output, and the system would not be controllable. If $t_f > N$, then there will still be a nullspace of the desired output, unless the rank is N, when $t_f = N$, and the system is controllable. The test can therefore be restricted to this last case.

This concept of controllability can be described in a number of interesting and useful ways[34] and generalized to vector controls and time-dependent models. To the extent that a model is found to be uncontrollable, it shows that some elements of the state vector are not connected to the controls, and one might ask why this is so and whether the model cannot then be usefully simplified.

Observability

The concept of "observability" is connected to the question of whether given N perfect observations, it is possible to infer all of the initial conditions. Suppose that the same model is used, and that we have (for simplicity only) a scalar observation sequence,

$$y(t) = \mathbf{E}(t)\mathbf{x}(t) + n(t), \quad t = 0, 1, \ldots, t_f. \tag{4.151}$$

Can we find $\mathbf{x}(0)$? The sequence of observations can be written, with $u(t) \equiv 0$, as

$$\begin{aligned} y(1) &= \mathbf{E}(1)\mathbf{x}(1) = \mathbf{E}(1)\mathbf{A}\mathbf{x}(0), \\ &\vdots \\ y(t_f) &= \mathbf{E}(t_f)\mathbf{A}^{t_f}\mathbf{x}(0), \end{aligned}$$

which is

$$\mathbf{O}\mathbf{x}(0) = [\, y(1) \ldots y(t_f)\,]^{\mathrm{T}}$$

$$\mathbf{O} = \left\{ \begin{matrix} \mathbf{E}(1)\mathbf{A} \\ \vdots \\ \mathbf{E}(t_f)\mathbf{A}^{t_f} \end{matrix} \right\}. \tag{4.152}$$

If the "observability matrix" is square – that is, $t_f = N$ and $\mathbf{O}$ is full rank – there is a unique solution for $\mathbf{x}(0)$, and the system is said to be observable. Should it fail to be observable, it suggests that at least some of the initial conditions are not determinable by an observation sequence and are irrelevant. Determining why that should be would surely shed light on the model one was using. As with controllability, the test (4.152) can be rewritten in a number of ways, and the concept can be extended to more complicated systems. The concepts of "stabilizability," "reachability," "reconstructability," and "detectability" are closely related.[35] A close connection

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

also exists between observability and controllability and the existence of a steady-state solution for the algebraic Riccati equations.

In practice, one must distinguish between mathematical observability and controllability and practical limitations imposed by the realities of observational systems. It is characteristic of fluids that changes occurring in some region at a particular time are ultimately communicated to all locations, no matter how remote, at later times, although the delay may be considerable, and the magnitudes of the signal may be much reduced by dissipation and geometrical spreading. Nonetheless, one anticipates that there is almost no possible element of a fluid flow, no matter how distant from a particular observation, that is not in principle observable. This subject is further discussed in Chapter 5.

4.7 Non-linear models

Fluid flows are non-linear by nature, and one must address the data/model combination problem where the model is non-linear. (There are also, as noted above, instances in which the data are non-linear combinations of the state vector elements.) Nonetheless, the focus here on linear models is hardly wasted effort. As with more conventional systems, there are not many general methods for solving non-linear estimation or control problems; rather, as with forward modeling, each situation has to be analyzed as a special case. Much insight is derived from a thorough understanding of the linear case, and indeed it is difficult to imagine tackling any non-linear problem without a thorough grasp of the linear one. Not unexpectedly, the most accessible approaches to non-linear estimation/control are based upon linearizations.

A complicating factor in the use of non-linear models is that the objective functions need no longer have a unique minimum. There can be many nearby, or distant, minima, and the one chosen by the usual algorithms may depend upon exactly where one starts in the parameter space and how the search for the minimum is conducted. Indeed, the structure of the cost function may come to resemble a chaotic function, filled with hills, plateaus, and valleys into which one may stumble, never to get out again.[36] The combinatorial methods described in Chapter 3 are a partial solution.

4.7.1 The linearized and extended Kalman filter

If one employs a non-linear model,

$$\mathbf{x}(t) = \mathbf{L}(\mathbf{x}(t-1),\ \mathbf{Bq}(t-1),\ \mathbf{\Gamma}(t-1)\mathbf{u}(t-1)), \tag{4.153}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

then reference to the Kalman filter recursion shows that the forecast step can be taken as before,

$$\tilde{\mathbf{x}}(t,-) = \mathbf{L}(\tilde{\mathbf{x}}(t-1),\ \mathbf{B}\mathbf{q}(t-1),\ 0), \tag{4.154}$$

but it is far from clear how to propagate the uncertainty from $\mathbf{P}(t-1)$ to $\mathbf{P}(t,-)$, the previous derivation being based upon the assumption that the error propagates linearly, independently of the true value of $\mathbf{x}(t)$ (or equivalently, that if the initial error is Gaussian, then so is the propagated error). With a non-linear system one cannot simply add the propagated initial condition error to that arising from the unknown controls. A number of approaches exist to finding approximate solutions to this problem, but they can no longer be regarded as strictly optimal, representing different linearizations.

Suppose that we write

$$\mathbf{x}(t) = \mathbf{x}_o(t) + \Delta\mathbf{x}(t), \qquad \mathbf{q} = \mathbf{q}_o(t) + \Delta\mathbf{q}(t), \tag{4.155}$$

Then

$$\begin{aligned} \mathbf{x}_o(t) &= \mathbf{L}_o(\mathbf{x}_o(t-1),\ \mathbf{B}\mathbf{q}_o(t-1),\ t-1), \\ \mathbf{L}_o &= \mathbf{L}(\mathbf{x}_o(t-1),\ \mathbf{B}\mathbf{q}_o(t-1),\ 0,\ t-1) \end{aligned} \tag{4.156}$$

defines a nominal solution, or trajectory, $\mathbf{x}_o(t)$.

Non-linear models, in particular, can have trajectories that bifurcate in a number of different ways, so that, subject to slight differences in state, the trajectory can take widely differing pathways as time increases. This sensitivity can be a very serious problem for a Kalman filter forecast, because a linearization may take the incorrect branch, leading to divergences well beyond any formal error estimate. Note, however, that the problem is much less serious in a smoothing problem, as one then has observations available indicating the branch actually taken.

Assuming a nominal solution is available, we have an equation for the solution perturbation:

$$\begin{aligned} \Delta\mathbf{x}(t) = {}& \mathbf{L}_x(\mathbf{x}_o(t-1),\ \mathbf{B}\mathbf{q}_o(t-1), 0, t)^{\mathrm{T}}\Delta\mathbf{x}(t-1) \\ & + \mathbf{L}_q^{\mathrm{T}}\Delta\mathbf{q}(t-1) + \mathbf{L}_u^{\mathrm{T}}\mathbf{u}(t-1), \end{aligned} \tag{4.157}$$

where

$$\mathbf{L}_x(\mathbf{x}_o(t),\ \mathbf{B}\mathbf{q}_o(t), 0, t) = \frac{\partial\mathbf{L}}{\partial\mathbf{x}(t)}, \quad \mathbf{L}_q(\mathbf{x}_o(t),\ \mathbf{B}\mathbf{q}_o(t), 0, t) = \frac{\partial\mathbf{L}}{\partial\mathbf{q}(t)},$$

$$\mathbf{L}_u(\mathbf{x}_o(t),\ \mathbf{B}\mathbf{q}_o(t), 0, t) = \frac{\partial\mathbf{L}}{\partial\mathbf{u}(t)},$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

which is linear – called the "tangent linear model," and of the form already used for the Kalman filter, but with redefinitions of the governing matrices. The model is assumed to be differentiable in this manner; discrete models are differentiable, numerically, barring a division by zero somewhere. They are by definition discontinuous, and discrete differentiation automatically accommodates such discontinuities. Numerical models often have switches, typically given by "if xx, then yy" statements. Even these models are differentiable in the sense we need, except at the isolated point where the "if" statement is evaluated; typically the code representing the derivatives will also have a switch at this point. The full solution would be the sum of the nominal solution, $\mathbf{x}_o(t)$, and the perturbation $\Delta\mathbf{x}(t)$. This form of estimate is sometimes known as the "linearized Kalman filter," or the "neighboring optimal estimator." Its usage depends upon the existence of a nominal solution, differentiability of the model, and the presumption that the controls $\Delta\mathbf{q}$, $\mathbf{u}$ do not drive the system too far from the nominal trajectory.

The so-called "extended Kalman filter" is nearly identical, except that the linearization is taken instead about the most recent estimate $\tilde{\mathbf{x}}(t)$; that is, the partial derivatives in (4.156) are evaluated using not $\mathbf{x}_o(t-1)$, but $\tilde{\mathbf{x}}(t-1)$. This latter form is more prone to instabilities, but if the system drifts very far from the nominal trajectory, it could well be more accurate than the linearized filter. Linearized smoothing algorithms can be developed in analogous ways, and as already noted, the inability to track strong model non-linearities is much less serious with a smoother than with a filter. The references go into these questions in detail. Problems owing to forks in the trajectory[37] can always be overcome by having enough observations to keep the estimates close to the true state. The usual posterior checks of model and data residuals are also a very powerful precaution against a model failing to track the true state adequately.

Example *Suppose, for the mass–spring oscillator on p. 183, there is a non-linear perturbation in the difference equation (4.17) – a non-linear term proportional to $\varepsilon\xi(t)^3$, where ε is very small. Then the governing difference equation becomes, with arbitrary Δt,*

$$\begin{bmatrix} \xi(t) \\ \xi(t-\Delta t) \end{bmatrix} = \begin{Bmatrix} 2 - \frac{r}{m}\Delta t - \frac{k}{m}(\Delta t)^2 & \frac{r\Delta t}{m} - 1 \\ 1 & 0 \end{Bmatrix} \begin{bmatrix} \xi(t-\Delta t) \\ \xi(t-2\Delta t) \end{bmatrix}$$
$$+ \varepsilon \begin{bmatrix} \xi\,(t-\Delta t)^3 \\ 0 \end{bmatrix} + \begin{bmatrix} (\Delta t)^2\,\frac{q(t-\Delta t)}{m} \\ 0 \end{bmatrix},$$

or,

$$\mathbf{x}\,(t) = \mathbf{A}\mathbf{x}\,(t-\Delta t) + \varepsilon\mathbf{L}_1\,(\mathbf{x}\,(t-\Delta t)) + \mathbf{B}\mathbf{q}\,(t)\,, \quad \mathbf{x}\,(t) = \mathbf{x}(0),$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

a discrete analogue of the so-called hard spring equation. Define a nominal trajectory, $x_0(t)$, satisfying the linearized version of this last equation (there is no general necessity for the nominal trajectory to be linear):

$$\mathbf{x}_0(t) = \mathbf{A}\mathbf{x}_0(t-\Delta t) + \mathbf{B}\mathbf{q}(t), \quad \mathbf{x}_0(0) = \mathbf{x}(0),$$

and let $\mathbf{x}(t) = \mathbf{x}_0(t) + \varepsilon\Delta\mathbf{x}(t)$, so that, to $O(\varepsilon)$,

$$\Delta\mathbf{x}(t) = \mathbf{A}\Delta\mathbf{x}(t-\Delta t) + \varepsilon\mathbf{L}_1(\mathbf{x}_0(t-\Delta t)), \quad \Delta\mathbf{x}(0) = \mathbf{0}, \tag{4.158}$$

which is a special case of (4.157). $\varepsilon\mathbf{L}_1(\mathbf{x}_0(t-\Delta t))$ takes the role of $\mathbf{B}\mathbf{q}(t)$ in the linear problem. Whether $\varepsilon\Delta\mathbf{x}(t)$ remains sufficiently small with time can be determined empirically (the continuous version of this problem is the Duffing equation, about which a great deal is known – the approximation we have made can lead to unbounded perturbations).[38] *In general, one would expect to have to relinearize at finite time. Figure 4.16 shows the full non-linear trajectory of $x_1(t)$, the linear trajectory, $x_{01}(t)$, and the sum of the solution to Eq. (4.158) and that of $x_{01}(t)$. The linear and non-linear solutions ultimately diverge, and in an estimation problem one would almost certainly re-linearize about the estimated trajectory.*

It is possible to define physical systems for which sufficiently accurate or useful derivatives of the system cannot be defined[39] so that neither Lagrange multiplier nor linearized sequential methods can be used. Whether such systems occur in practice, or whether they are somewhat like the mathematical pathologies used by mathematicians to demonstrate the limits of conventional mathematical tools (e.g., the failure to exist of the derivative of $\sin(1/t),\ t \to 0$, or of the existence of space-filling curves) is not so clear. It is clear that all linearization approaches do have limits of utility, but they are and will likely remain, the first choice of practitioners necessarily aware that no universal solution methods exist.

4.7.2 Parameter estimation and adaptive estimation

Often models contain parameters whose values are poorly known. In fluid flow problems, these often concern parameterized turbulent mixing, with empirical parameters which the user is willing to adjust to provide the best fit of the model to the observations. Sometimes, this approach is the only way to determine the parameters.

Suppose that the model is linear in $\mathbf{x}(t)$ and that it contains a vector of parameters, $\mathbf{p}$, whose nominal values, $\mathbf{p}_0$, we wish to improve, while also estimating the state

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511635649.005

Cambridge Books Online © Cambridge University Press, 2009

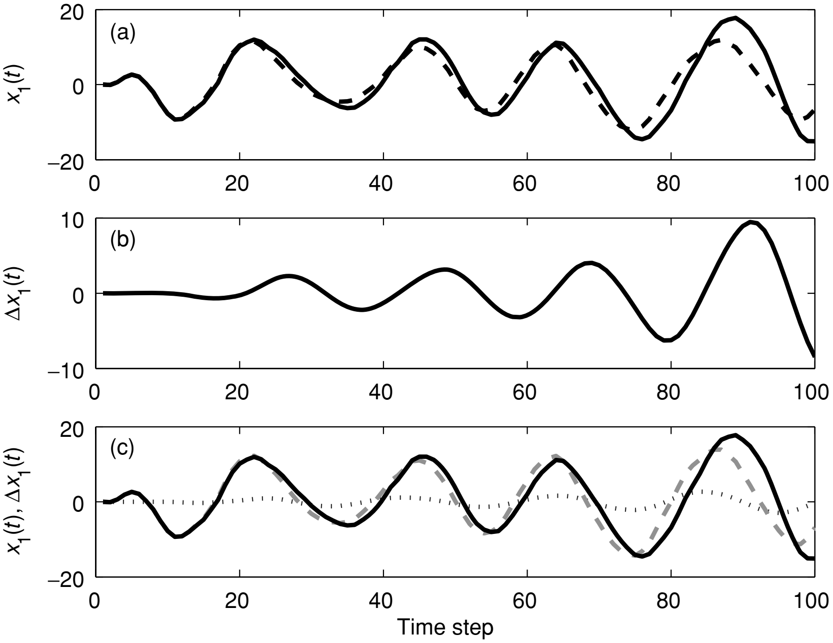

Figure 4.16 (a) The solid curve depicts the time-stepped solution, $x_1(t)$, to the non-linear finite difference oscillator. The dashed line shows the same solution in a linear approximation (here $\varepsilon = 8 \times 10^{-5}$, $\Delta t = 1$, $r = 0.02$, $k = 0.1$). (b) Shows the difference between the linearized and non-linear solutions. (c) The solid curve is the full non-linear solution (as in (a)), the dotted curve is the anomaly solution (to Eq. (4.158)), and the dashed curve is the sum of the linear and anomaly solutions. One would probably re-linearize about the actual solution at a finite time if observational information were available.

vector. Write the model as

$$\mathbf{x}(t) = \mathbf{A}(\mathbf{p}(t-1))\mathbf{x}(t-1) + \mathbf{B}\mathbf{q}(t-1) + \boldsymbol{\Gamma}\mathbf{u}(t-1), \tag{4.159}$$

where the time dependence in $\mathbf{p}(t)$ arises from the changing estimate of its value rather than a true physical time dependence. A general approach to solving this problem is to augment the state vector. That is,

$$\mathbf{x}_A(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{p}(t) \end{bmatrix}. \tag{4.160}$$

Then write a model for this augmented state as

$$\mathbf{x}_A(t) = \mathbf{L}_A\left[\mathbf{x}_A(t-1),\ \mathbf{q}(t-1),\ \mathbf{u}(t-1)\right], \tag{4.161}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

where

$$\mathbf{L}_A = \left\{\begin{matrix} \mathbf{A}(\mathbf{p}(t-1)) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{matrix}\right\} \mathbf{x}_A\,(t-1) + \mathbf{B}\mathbf{q}(t-1) + \boldsymbol{\Gamma}\mathbf{u}(t-1). \tag{4.162}$$

The observation equation is augmented simply as

$$\mathbf{y}_A(t) = \mathbf{E}_A(t)\mathbf{x}_A(t) + \mathbf{n}_A(t),$$
$$\mathbf{E}_A(t) = \{\mathbf{E}(t) \quad \mathbf{0}\}, \quad \mathbf{n}_A(t) = \mathbf{n}(t),$$

assuming that there are no direct measurements of the parameters. The evolution equation for the parameters can be made more complex than indicated here. A solution can be found by using the linearized Kalman filter, for example, linearizing about the nominal parameter values. Parameter estimation is a very large subject.[40]

A major point of concern in estimation procedures based upon Gauss–Markov type methods lies in specification of the various covariance matrices, especially those describing the model error – here lumped into $\mathbf{Q}(t)$. The reader will probably have concluded that there is, however, nothing precluding deduction of the covariance matrices from the model and observations, given that adequate numbers of observations are available. The possibility is briefly discussed on p. 273.

4.7.3 Non-linear adjoint equations: searching for solutions

Consider now a non-linear model in the context of the Lagrange multipliers approach. Let the model be non-linear in either the state vector or model parameters, or both, so that a typical objective function is

$$\begin{aligned} J = {} & [\mathbf{x}(0) - \mathbf{x}_0]^{\mathrm{T}}\, \mathbf{P}(0)^{-1}\, [\mathbf{x}(0) - \tilde{\mathbf{x}}_0] \\ & + \sum_{t=1}^{t_f} [\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)]^{\mathrm{T}}\, \mathbf{R}(t)^{-1}\, [\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)] \\ & + \sum_{t=0}^{t_f - 1} \mathbf{u}(t)^{\mathrm{T}} \mathbf{Q}(t)^{-1} \mathbf{u}(t) \\ & - 2 \sum_{t=1}^{t_f} \boldsymbol{\mu}(t)^{\mathrm{T}}\, [\mathbf{x}(t) - \mathbf{L}\,[\mathbf{x}(t-1),\ \mathbf{B}\mathbf{q}(t-1),\ \boldsymbol{\Gamma}\mathbf{u}(t-1)]]\,. \end{aligned} \tag{4.163}$$

Here $\mathbf{x}_o$ is the a-priori estimate of the initial conditions with uncertainty $\mathbf{P}(0)$, and the tildes have been omitted from the remaining variables. The observations continue to be treated as linear in the state vector, but even this assumption can be

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

relaxed. The normal equations are:

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{u}(t)} = \mathbf{Q}(t)^{-1}\mathbf{u}(t) + \left(\frac{\partial \mathbf{L}(\mathbf{x}(t),\ \mathbf{B}\mathbf{q}(t),\ \boldsymbol{\Gamma}\mathbf{u}(t))}{\partial \mathbf{u}(t)}\right)^{\mathrm{T}} \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{\mu}(t+1) = \mathbf{0}, \tag{4.164}$$

$$t = 0, 1, \ldots, t_f - 1,$$

$$\frac{1}{2}\frac{\partial J}{\partial \boldsymbol{\mu}(t)} = \mathbf{x}(t) - \mathbf{L}\left[\mathbf{x}(t-1),\ \mathbf{B}\mathbf{q}(t-1),\ \boldsymbol{\Gamma}\mathbf{u}(t-1)\right] = \mathbf{0}, \quad t = 1, 2, \ldots, t_f, \tag{4.165}$$

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{x}(0)} = \mathbf{P}(0)^{-1}\left[\mathbf{x}(0) - \mathbf{x}_0\right] + \left(\frac{\partial \mathbf{L}(\mathbf{x}(0),\ \mathbf{B}\mathbf{q}(0),\ \boldsymbol{\Gamma}\mathbf{u}(0))}{\partial \mathbf{x}(0)}\right)^{\mathrm{T}} \boldsymbol{\mu}(1) = \mathbf{0}, \tag{4.166}$$

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{x}(t)} = \mathbf{E}(t)^{\mathrm{T}}\mathbf{R}(t)^{-1}\left[\mathbf{E}(t)\mathbf{x}(t) - \mathbf{y}(t)\right] - \boldsymbol{\mu}(t) \tag{4.167}$$

$$+ \left(\frac{\partial \mathbf{L}(\mathbf{x}(t),\ \mathbf{B}\mathbf{q}(t),\ \boldsymbol{\Gamma}\mathbf{u}(t))}{\partial \mathbf{x}(t)}\right)^{\mathrm{T}} \boldsymbol{\mu}(t+1) = \mathbf{0}, \quad t = 1, 2, \ldots, t_f - 1,$$

$$\frac{1}{2}\frac{\partial J}{\partial \mathbf{x}(t_f)} = \mathbf{E}(t_f)^{\mathrm{T}}\mathbf{R}(t_f)^{-1}\left[\mathbf{E}(t_f)\mathbf{x}(t_f) - \mathbf{y}(t_f)\right] - \boldsymbol{\mu}(t_f) = \mathbf{0}. \tag{4.168}$$

These are non-linear because of the non-linear model (4.165) – although the adjoint model (4.167) remains linear in $\boldsymbol{\mu}(t)$ – and the linear methods used thus far will not work directly. The operators that appear in the above equations,

$$\left(\frac{\partial \mathbf{L}(\mathbf{x}(t), \mathbf{B}\mathbf{q}(t), \boldsymbol{\Gamma}\mathbf{u}(t), t)}{\partial \mathbf{u}(t)}\right), \quad \left(\frac{\partial \mathbf{L}(\mathbf{x}(t), \mathbf{B}\mathbf{q}(t), \boldsymbol{\Gamma}\mathbf{u}(t), t)}{\partial \mathbf{x}(t)}\right), \tag{4.169}$$

are, as in Eq. (4.156), the derivatives of the model with respect to the control and state vectors. Assuming that they exist, they represent a linearization of the model about the state and control vectors and again are the tangent linear model. Their transposes are, in this context, the adjoint model. There is some ambiguity about the terminology: the form of (4.169) or the transposes are definable independent of the form of J. Otherwise, Eq. (4.167) and its boundary condition (4.168) depend upon the actual observations and the details of J; one might call this pair the "adjoint evolution" equation to distinguish it from the adjoint model.

If the non-linearity is not too large, perturbation methods may work. This notion leads to what is usually called "neighboring optimal control."[41] Where the non-linearity is large, the approach to solution is an iterative one. Consider what one is trying to do. At the optimum, if we can find it, J will reach a stationary value in which the terms multiplying the $\boldsymbol{\mu}(t)$ will vanish. Essentially, one uses *search*

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

methods that are able to find a solution (there may well be multiple such solutions, each corresponding to a local minimum of J).

There are many known ways to seek approximate solutions to a set of simultaneous equations, linear or non-linear, using various search procedures. Most such methods are based upon what are usually called "Newton" or "quasi-Newton" methods, or variations on steepest descent. The most popular approach to tackling the set (4.164)–(4.168) has been a form of conjugate gradient algorithm.[42] The iteration cycles are commonly carried out by making a first estimate of the initial conditions and the boundary conditions – for example, setting $\mathbf{u} = \mathbf{0}$. One integrates (4.165) forwards in time to produce a first guess for $\mathbf{x}(t)$. A first guess set of Lagrange multipliers is obtained by integrating (4.167) backwards in time. Normally, (4.164) is not then satisfied, but because the values obtained provide information on the gradient of the objective function with respect to the controls, one knows the sign of the changes to make in the controls to reduce J. Perturbing the original guess for $\mathbf{u}(t)$ in this manner, one does another forward integration of the model and backward integration of the adjoint. Because the Lagrange multipliers provide the partial derivatives of J with respect to the solution (Eq. (4.166) permits calculation of the direction in which to shift the current estimate of $\mathbf{x}(0)$ to decrease J), one can employ a conjugate gradient or steepest descent method to modify $\tilde{\mathbf{x}}(0)$ and carry out another iteration.

In this type of approximate solution, the adjoint solution, $\tilde{\boldsymbol{\mu}}(t)$, is really playing two distinct roles. On the one hand, it is a mathematical device to impose the model constraints; on the other, it is being used as a numerical convenience for determining the direction and step size to best reduce the objective function – as it contains information on the change in J with parameter perturbations. The problem of possibly falling into the wrong minimum of the objective function remains.

In practice, $\mathbf{L}(\mathbf{x}(t-1), \mathbf{Bq}(t-1), \boldsymbol{\Gamma}\mathbf{u}(t-1), t-1)$ is represented as many lines of computer code. Generating the derivatives in Eq. (4.169) can be a major undertaking. Fortunately, and remarkably, the automatic differentiation (AD) tools mentioned above can convert the code for $\mathbf{L}[\mathbf{x}(t), \mathbf{Bq}(t), \boldsymbol{\Gamma}\mathbf{u}(t), t]$ into the appropriate code for the derivatives. While still requiring a degree of manual intervention, this AD software renders Lagrange multiplier methods a practical approach for model codes running to many thousands of lines.[43] The basic ideas are sketched in the next subsection and in the Appendix to this chapter.

4.7.4 Automatic differentiation, linearization, and sensitivity

The linearized and extended filters and smoothers (e.g., Eq. (4.156)) and the normal equations (4.164)–(4.168) involve derivatives such as $\partial\mathbf{L}/\partial\mathbf{x}(t)$. One might wonder how these are to be obtained. Several procedures exist, including the one used above

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511575549.005

for the weakly non-linear spring, but to motivate what is perhaps the most elegant method, begin with a simple example of a two-dimensional non-linear model.

Example *Let*

$$\mathbf{x}(t) = \mathbf{L}(\mathbf{x}(t-1)) = \begin{bmatrix} a\mathbf{x}^{\mathrm{T}}(t-1)\mathbf{x}(t-1) + c \\ b\mathbf{x}^{\mathrm{T}}(t-1)\mathbf{x}(t-1) + d \end{bmatrix}, \tag{4.170}$$

where a, b, c, d are fixed constants. Time-stepping from $t = 0$,

$$\mathbf{x}(1) = \begin{bmatrix} a\mathbf{x}^{\mathrm{T}}(0)\mathbf{x}(0) + c \\ b\mathbf{x}^{\mathrm{T}}(0)\mathbf{x}(0) + d \end{bmatrix}, \tag{4.171}$$

$$\mathbf{x}(2) = \begin{bmatrix} a\mathbf{x}^{\mathrm{T}}(1)\mathbf{x}(1) + c \\ b\mathbf{x}^{\mathrm{T}}(1)\mathbf{x}(1) + d \end{bmatrix},$$

$$\cdots$$

Consider the dependence, $\partial\mathbf{x}(t)/\partial\mathbf{x}(0)$,

$$\frac{\partial\mathbf{x}(t)}{\partial\mathbf{x}(0)} = \left\{ \begin{matrix} \dfrac{\partial x_1(t)}{\partial x_1(0)} & \dfrac{\partial x_2(t)}{\partial x_1(0)} \\ \dfrac{\partial x_1(t)}{\partial x_2(0)} & \dfrac{\partial x_2(t)}{\partial x_2(0)} \end{matrix} \right\}. \tag{4.172}$$

For $t = 2$, by the definitions and rules of Chapter 2, we have

$$\frac{\partial\mathbf{x}(2)}{\partial\mathbf{x}(0)} = \left\{ \begin{matrix} \dfrac{\partial x_1(2)}{\partial x_1(0)} & \dfrac{\partial x_2(2)}{\partial x_1(0)} \\ \dfrac{\partial x_1(2)}{\partial x_2(0)} & \dfrac{\partial x_2(2)}{\partial x_2(0)} \end{matrix} \right\} = \frac{\partial\mathbf{L}(\mathbf{L}(\mathbf{x}(0)))}{\partial\mathbf{x}(0)} = \mathbf{L}'(\mathbf{x}(0))\mathbf{L}'(\mathbf{L}(\mathbf{x}(0))). \tag{4.173}$$

Note that

$$\frac{\partial\mathbf{x}(2)}{\partial\mathbf{x}(0)} = \frac{\partial\mathbf{L}(\mathbf{x}(1))}{\partial\mathbf{x}(0)} = \frac{\partial\mathbf{x}(1)}{\partial\mathbf{x}(0)}\frac{\partial\mathbf{L}(\mathbf{x}(1))}{\partial\mathbf{x}(1)} = \frac{\partial\mathbf{L}(\mathbf{x}(0))}{\partial\mathbf{x}(0)}\frac{\partial\mathbf{L}(\mathbf{x}(1))}{\partial\mathbf{x}(1)}, \tag{4.174}$$

where we have used the "chain rule" for differentiation. Substituting into (4.174),

$$\begin{aligned} \frac{\partial\mathbf{x}(2)}{\partial\mathbf{x}(0)} &= \left\{ \begin{matrix} 2ax_1(0) & 2bx_1(0) \\ 2ax_2(0) & 2bx_2(0) \end{matrix} \right\} \left\{ \begin{matrix} 2ax_1(1) & 2bx_1(1) \\ 2ax_2(1) & 2bx_2(1) \end{matrix} \right\} \\ &= \left\{ \begin{matrix} 2ax_1(0) & 2bx_1(0) \\ 2ax_2(0) & 2bx_2(0) \end{matrix} \right\} \left\{ \begin{matrix} 2a^2 & 2ab \\ 2ab & 2b^2 \end{matrix} \right\} \mathbf{x}^{\mathrm{T}}(0)\mathbf{x}(0). \end{aligned}$$

By direct calculation from Eq. (4.171), we have

$$\frac{\partial\mathbf{x}(2)}{\partial\mathbf{x}(0)} = 4(a^2 + b^2)\mathbf{x}(0)^{\mathrm{T}}\mathbf{x}(0) \left\{ \begin{matrix} ax_1(0) & bx_1(0) \\ ax_2(0) & bx_2(0) \end{matrix} \right\}. \tag{4.175}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

Substituting (4.170) into Eq. (4.173), then

$$\mathbf{L}'(\mathbf{x}(0)) = \begin{Bmatrix} 2ax_1(0) & 2bx_1(0) \\ 2ax_2(0) & 2bx_2(0) \end{Bmatrix},$$

and

$$\mathbf{L}'\left(\mathbf{L}\left(\mathbf{x}\left(0\right)\right)\right) = \mathbf{L}'\left(\mathbf{x}\left(1\right)\right) = \begin{Bmatrix} 2ax_1\left(1\right) & 2bx_1(1) \\ 2ax_2\left(1\right) & 2bx_2\left(1\right) \end{Bmatrix}$$
$$= \begin{Bmatrix} 2a^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \\ 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2b^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \end{Bmatrix}.$$

Multiplying, as in (4.173),

$$\begin{Bmatrix} 2ax_1\left(0\right) & 2bx_1(0) \\ 2ax_2\left(0\right) & 2bx_2\left(0\right) \end{Bmatrix} \begin{Bmatrix} 2a^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \\ 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2b^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \end{Bmatrix}$$
$$= 4\left(a^2+b^2\right)\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right)\begin{Bmatrix} ax_1\left(0\right) & bx_1\left(0\right) \\ ax_2\left(0\right) & bx_2\left(0\right) \end{Bmatrix},$$

which is consistent with (4.175). Hence, as required,

$$d\mathbf{x}\left(2\right) = d\mathbf{x}\left(0\right)^{\mathrm{T}} \begin{Bmatrix} 2ax_1\left(0\right) & 2bx_1(0) \\ 2ax_2\left(0\right) & 2bx_2\left(0\right) \end{Bmatrix} \begin{Bmatrix} 2a^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \\ 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2b^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \end{Bmatrix}.$$

As a computational point note that this last equation involves a matrix–matrix multiplication on the right. But if written as a transpose,

$$\mathrm{d}\mathbf{x}\left(2\right)^{\mathrm{T}} = \begin{Bmatrix} 2a^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \\ 2ab\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) & 2b^2\mathbf{x}\left(0\right)^{\mathrm{T}}\mathbf{x}\left(0\right) \end{Bmatrix}^{\mathrm{T}} \begin{Bmatrix} 2ax_1\left(0\right) & 2bx_1(0) \\ 2ax_2\left(0\right) & 2bx_2\left(0\right) \end{Bmatrix}^{\mathrm{T}} \mathrm{d}\mathbf{x}\left(0\right),$$

$\mathrm{d}\mathbf{x}\left(2\right)^{\mathrm{T}}$ *can be found from matrix–vector multiplications alone, which for large matrices is a vastly reduced computation. This reduction in computational load lies behind the use of so-called reverse mode methods described below.*

Going much beyond these simple statements takes us too far into the technical details.[44]

The chain rule can be extended, such that

$$\frac{\partial\mathbf{x}(t)}{\partial\mathbf{x}(0)} = \frac{\partial\mathbf{L}(\mathbf{x}(t-1))}{\partial\mathbf{x}(0)} = \frac{\partial\mathbf{x}(t-1)}{\partial\mathbf{x}(0)}\frac{\partial\mathbf{L}(\mathbf{x}(t-1))}{\partial\mathbf{x}(t-1)} = \frac{\partial\mathbf{L}(\mathbf{x}(t-2))}{\partial\mathbf{x}(0)}\frac{\partial\mathbf{L}(\mathbf{x}(t-1))}{\partial\mathbf{x}(t-1)} \tag{4.176}$$
$$= \frac{\partial(\mathbf{x}(t-2))}{\partial\mathbf{x}(0)}\frac{\partial\mathbf{L}(\mathbf{x}(t-2))}{\partial\mathbf{x}(t-2)}\frac{\partial\mathbf{L}(\mathbf{x}(t-1))}{\partial\mathbf{x}(t-1)} = \cdots$$
$$= \frac{\partial\mathbf{L}(\mathbf{x}(0))}{\partial\mathbf{x}(0)}\cdots\frac{\partial\mathbf{L}(\mathbf{x}(t-2))}{\partial\mathbf{x}(t-2)}\frac{\partial\mathbf{L}(\mathbf{x}(t-1))}{\partial\mathbf{x}(t-1)}.$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

Although this result is formally correct, such a calculation could be quite cumbersome to code and carry out for a more complicated model (examples of such codes do exist). An alternative, of course, is to systematically and separately perturb each of the elements of $\mathbf{x}(0)$, and integrate the model forwards from $t = 0$ to t_f, thus numerically evaluating $\partial\mathbf{x}(t)/\partial x_i(0)$, $i = 1, 2, \ldots, N$. The model would thus have to be run N times, and there might be issues of numerical accuracy. (The approach is similar to the determination of numerical Green functions considered above.)

Practical difficulties such as these have given rise to the idea of "automatic (or algorithmic) differentiation" in which one accepts from the beginning that a computer code will be used to define $\mathbf{L}(\mathbf{x}(t), t, \mathbf{q}(t))$ (reintroducing the more general definition of $\mathbf{L}$).[45] One then seeks automatic generation of a second code, capable of evaluating the elements in Eq. (4.176), that is, terms of the form $\partial\mathbf{L}(\mathbf{x}(n))/\partial\mathbf{x}(n)$, for any n. Automatic differentiation (AD) tools take the computer codes (typically in Fortran, C, or Matlab) and generate new codes for the *exact* partial derivatives of the code. Various packages go under names like ADIFOR, TAF, ADiMAT, etc. (see www.autodiff.org). The possibility of using such procedures has already been alluded to, where it was noted that for a linear model, the first derivative would be the state transition matrix $\mathbf{A}$, which may not otherwise be explicitly available. That is,

$$\mathbf{A}(t) = \frac{\partial\mathbf{L}(\mathbf{x}(t), t, \mathbf{q}(t))}{\partial\mathbf{x}(t)}.$$

Actual implementation of AD involves one deeply in the structures of computer coding languages, and is not within the scope of this book. Note that the existing implementations are not restricted to such simple models as we used in the particular example, but deal with the more general $\mathbf{L}(\mathbf{x}(t), t, \mathbf{q}(t))$.

In many cases, one cares primarily about some scalar quantity, $H(\tilde{\mathbf{x}}(t_f))$, e.g., the heat flux or pressure field in a flow, as given by the state vector at the end time, $\mathbf{x}(t_f)$, of a model computation. Suppose[46] one seeks the sensitivity of that quantity to perturbations in the initial conditions (any other control variable could be considered), $\mathbf{x}(0)$. Let $\mathbf{L}$ continue to be the operator defining the time-stepping of the model. Define $\Psi_t = \mathbf{L}(\mathbf{x}(t), t, \mathbf{q}(t))$. Then

$$H = H(\Psi_{t_f}[\Psi_{t_f-1}[\cdots\Psi_1[\mathbf{x}(0)]]]),$$

that is, the function of the final state of interest is a nested set of operators working on the control vector $\mathbf{x}(0)$. Then the derivative of H with respect to $\mathbf{x}(0)$ is again obtained from the chain rule,

$$\frac{\partial H}{\partial\mathbf{x}(0)} = H'(\Psi'_{t_f}[\Psi'_{t_f-1}[\cdots\Psi'_1[\mathbf{x}(0)]]]), \tag{4.177}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

where the prime denotes the derivative with respect to the argument of the operator $\mathbf{L}$ evaluated at that time,

$$\frac{\partial \Psi_t(\mathbf{p})}{\partial \mathbf{p}}.$$

Notice that these derivatives, are the Jacobians (matrices) of dimension $N \times N$ at each time-step, and are the same derivatives that appear in the operators in (4.169). The nested operator (4.177) can be written as a matrix product,

$$\frac{\partial H}{\partial \mathbf{x}(0)} = \nabla h^{\mathrm{T}} \frac{\partial \Psi_{t_f}(\mathbf{p})}{\partial \mathbf{p}} \frac{\partial \Psi_{t_f}(\mathbf{p})}{\partial \mathbf{p}} \cdots \frac{\partial \Psi_1(\mathbf{p})}{\partial \mathbf{p}}. \tag{4.178}$$

∇h is the vector of derivatives of function H (the gradient) and so (4.178) is a column vector of dimension $N \times 1$. $\mathbf{p}$ represents the state vector at the prior time-step for each Ψ_t. The adjoint compilers described above compute $\partial H / \partial \mathbf{x}(0)$ in what is called the "forward mode," producing an operator that runs from right to left, multiplying $t_f - N \times N$ matrices starting with $\partial \Psi_1(\mathbf{p}) / \partial \mathbf{p}$.

If, however, Eq. (4.178) is transposed, then

$$\left(\frac{\partial H}{\partial \mathbf{x}(0)}\right)^{\mathrm{T}} = \left(\frac{\partial \Psi_1(\mathbf{p})}{\partial \mathbf{p}}\right)^{\mathrm{T}} \left(\frac{\partial \Psi_2(\mathbf{p})}{\partial \mathbf{p}}\right)^{\mathrm{T}} \cdots \left(\frac{\partial \Psi_{t_f}(\mathbf{p})}{\partial \mathbf{p}}\right)^{\mathrm{T}} \nabla h, \tag{4.179}$$

where the first multiplication on the right involves multiplying the column vector ∇h by an $N \times N$ matrix, thus producing another $N \times 1$ vector. More generally, the set of products in (4.179), again taken from right to left, involves only multiplying a vector by a matrix, rather than a matrix by a matrix as in (4.178), with a potentially very large computational saving. Such evaluation is the *reverse* or *adjoint mode* calculation alluded to above (the transposes generate the required adjoint operators, although a formal transpose is not actually formed) and have become available in some automatic differentiation tools only comparatively recently. In comparing the computation in the forward and reverse modes, one must be aware that there is a storage penalty in (4.179) not incurred in (4.178).[47] In practice, the various operators $\partial \Psi_i(\mathbf{p}) / \partial \mathbf{p}$ are not obtained explicitly, but are evaluated.

Historically, the forward mode was developed first, and remains the most common implementation of AD. It permits one to systematically linearize models, and, by repeated application of the AD tool, to develop formal Taylor series for non-linear models. With the rise in fluid state estimation problems of very large dimension, there has recently been a much greater emphasis on the reverse mode.

Many fluid models rely on "if ... then ..." and similar branching statements, such as assignment of a variable to the maximum value of a list. For example, if some region is statically unstable owing to cooling at the surface, a test for instability may lead the model to homogenize the fluid column; otherwise, the

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

stratification is unaffected. Objections to AD are sometimes raised, apparently based on the intuitive belief that such a model cannot be differentiated. In practice, once a branch is chosen, the state vector is well-defined, as is its derivative, the AD code itself then having corresponding branches or, e.g., assignments to maxima or minima of a list. A brief example of this issue is given in the Appendix to this chapter. Our employment so far of the adjoint model and the adjoint evolution equation has been in the context of minimizing an objective function – and, to some degree, the adjoint has been nothing but a numerical convenience for algorithms that find minima. As we have seen repeatedly, however, Lagrange multipliers have a straightforward interpretation as the sensitivity of an objective function, J, to perturbations in problem parameters. This use of the multipliers can be developed independently of the state estimation problem.

Example *Consider the linear time invariant model*

$$\mathbf{x}(n) = \mathbf{A}\mathbf{x}(n-1),$$

such that

$$\mathbf{x}(n) = \mathbf{A}^n\mathbf{x}(0).$$

Suppose we seek the dependence of $H = \mathbf{x}(n)^{\mathrm{T}}\mathbf{x}(n)/2 = (\mathbf{x}(0)^{\mathrm{T}}\mathbf{A}^{nT}\mathbf{A}^n\mathbf{x}(0))/2$ *on the problem parameters. The sensitivity to the initial conditions is straightforward:*

$$\frac{\partial H}{\partial \mathbf{x}(0)} = \mathbf{A}^{n\mathrm{T}}\mathbf{A}^n\mathbf{x}(0).$$

Suppose instead that $\mathbf{A}$ *depends upon an internal parameter,* k, *perhaps the spring constant in the example of the discrete mass–spring oscillator, for which* H *would be an energy. Then*

$$\frac{\partial H}{\partial k} = \frac{1}{2}\frac{\partial(\mathbf{x}(n)^{\mathrm{T}}\mathbf{x}(n))}{\partial k} = \frac{1}{2}\left(\frac{\partial(\mathbf{x}(n)^{\mathrm{T}}\mathbf{x}(n))}{\partial \mathbf{x}(n)}\right)^{\mathrm{T}}\frac{\partial \mathbf{x}(n)}{\partial k} = \mathbf{x}(n)^{\mathrm{T}}\frac{\partial \mathbf{x}(n)}{\partial k}.$$

We have, from Eq. (2.32),

$$\frac{\mathrm{d}\mathbf{x}(n)}{\mathrm{d}k} = \frac{\mathrm{d}\mathbf{A}^n}{\mathrm{d}k}\mathbf{x}(0) = \left[\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\mathbf{A}^{n-1} + \mathbf{A}\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\mathbf{A}^{n-2} + \cdots + \mathbf{A}^{n-1}\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\right]\mathbf{x}(0),$$

and so,

$$\frac{\partial H}{\partial k} = \mathbf{x}(0)^{\mathrm{T}}\mathbf{A}^{n\mathrm{T}}\left[\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\mathbf{A}^{n-1} + \mathbf{A}\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\mathbf{A}^{n-2} + \cdots + \mathbf{A}^{n-1}\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}k}\right]\mathbf{x}(0).$$

and with evaluation of $\mathrm{d}\mathbf{A}/\mathrm{d}k$ *being straightforward, we are finished.*

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

The Appendix to this chapter describes briefly how computer programs can be generated to carry out these operations.

4.7.5 Approximate methods

All of the inverse problems discussed, whether time-independent or not, were reduced ultimately to finding the minimum of an objective function, either in unconstrained form (e.g., (2.352) or (4.61)) or constrained by exact relationships (e.g., models) (2.354) or (4.97). Once the model has been formulated, the objective function agreed on, and the data obtained in appropriate form (often the most difficult step), the formal solution is reduced to finding the constrained or unconstrained minimum. "Optimization theory" is a very large, very sophisticated subject directed at finding such minima, and the methods we have described here – sequential estimation and Lagrange multiplier methods – are only two of a number of possibilities.

As we have seen, some of the methods stop at the point of finding a minimum and do not readily produce an estimate of the uncertainty of the solution. One can distinguish inverse methods from optimization methods by the requirement of the former for the requisite uncertainty estimates. Nonetheless, as noted before in some problems, mere knowledge that there is at least one solution may be of intense interest, irrespective of whether it is unique or whether its stability to perturbations in the data or model is well understood.

The reader interested in optimization methods generally is referred to the literature on that subject.[48] Geophysical fluid problems often fall into the category of extremely large, non-linear optimization, one that tends to preclude the general use of many methods that are attractive for problems of more modest size.

The continued exploration of ways to reduce the computational load without significantly degrading either the proximity to the true minimum or the information content (the uncertainties of the results) is a very high priority. Several approaches are known. The use of steady-state filters and smoothers has already been discussed. Textbooks discuss a variety of possibilities for simplifying various elements of the solutions. In addition to the steady-state assumption, methods include: (1) "state reduction" – attempting to remove from the model (and thus from the uncertainty calculation) elements of the state vector that are either of no interest or comparatively unchanging;[49] (2) "reduced-order observers,"[50] in which some components of the model are so well observed that they do not need to be calculated; and (3) proving or assuming that the uncertainty matrices (or the corresponding information matrices) are block diagonal or banded, permitting use of a variety of sparse algorithms. This list is not exhaustive.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

4.8 Forward models

The focus we have had on the solution of inverse problems has perhaps given the impression that there is some fundamental distinction between forward and inverse modeling. The point was made at the beginning of this book that inverse *methods* are important in solving forward as well as inverse *problems*. Almost all the inverse problems discussed here involved the use of an objective function, and such objective functions do not normally appear in forward modeling. The presence or absence of objective functions thus might be considered a fundamental difference between the problem types.

But numerical models do not produce universal, uniformly accurate solutions to the fluid equations. Any modeler makes a series of decisions about which aspects of the flow are most important for accurate depiction – the energy or vorticity flux, the large-scale velocities, the non-linear cascades, etc. – and which cannot normally be achieved simultaneously with equal fidelity. It is rare that these goals are written explicitly, but they could be, and the modeler could choose the grid and differencing scheme, etc., to minimize a specific objective function. The use of such explicit objective functions would prove beneficial because it would quantify the purpose of the model.

One can also consider the solution of ill-posed forward problems. In view of the discussion throughout this book, the remedy is straightforward: one must introduce an explicit objective function of the now-familiar type, involving state vectors, observations, control, etc., and this approach is precisely that recommended. If a Lagrange multiplier method is adopted, then Eqs. (2.334) and (2.335) show that an over- or underspecified forward model produces a complementary under- or overspecified adjoint model, and it is difficult to sustain a claim that modeling in the forward direction is fundamentally distinct from that in the inverse sense.

Example *Consider the ordinary differential equation*

$$\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} - k^2 x(t) = 0. \tag{4.180}$$

Formulated as an initial value problem, it is properly posed with Cauchy conditions $x(0) = x_0$, $x'(0) = x'_0$. *The solution is*

$$x(t) = A\exp(kt) + B\exp(-kt), \tag{4.181}$$

with A, B determined by the initial conditions. If we add another condition – for example, at the end of the interval of interest, $x(t_f) = x_{t_f}$ *– the problem is ill-posed because it is now overspecified. To analyze and solve such a problem using the methods of this book, discretize it as*

$$x(t+1) - (2+k^2)x(t) + x(t-1) = 0, \tag{4.182}$$

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

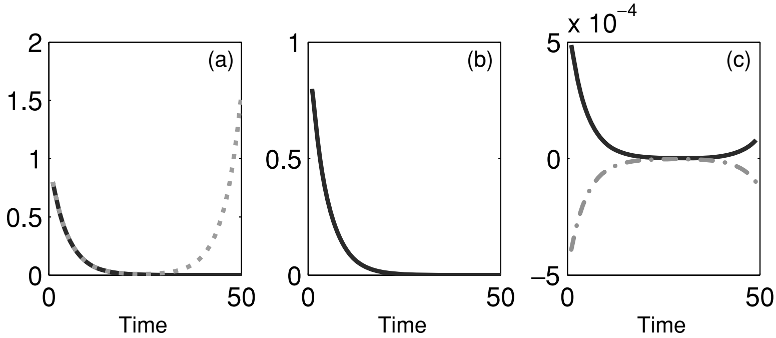

Figure 4.17 (a) Stable solution. $x_1(t)$ (dashed curve) to Eq. (4.182) obtained by setting $k^2 = 0.05$, $\mathbf{x}(0) = [0.800, 1.00]^{\mathrm{T}}$. The dotted line is an unstable solution obtained by modifying the initial condition to $\mathbf{x}(0) = [0.80001, 1.00]^{\mathrm{T}}$. The growing solution takes awhile to emerge, but eventually swamps the stable branch. (b) Solution obtained by overspecification, in which $\tilde{\mathbf{x}}(0) = [0.80001, 1.00]^{\mathrm{T}}$, $\mathbf{P}(0) = .01\mathbf{I}_2$, $\tilde{\mathbf{x}}(50) = [1.4 \times 10^{-5}, 1]$, $\mathbf{P}(50) = \mathrm{diag}([10^{-4}, 10^4])$. (c) Lagrange multiplier values used to impose the initial and final conditions on the model. Solid curve is $\mu_1(t)$, and chain is $\mu_2(t)$.

taking $\Delta t = 1$, with corresponding redefinition of k^2. A canonical form is

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1), \quad \mathbf{x}(t) = [x(t), \quad x(t-1)]^{\mathrm{T}}, \quad \mathbf{A} = \begin{Bmatrix} 2+k^2 & -1 \\ 1 & 0 \end{Bmatrix}.$$

The reduced form of equations (4.164)–(4.168) is easily solved (the only "observations" are at the final time) by a backwards sweep of the adjoint model (4.101) to obtain $\boldsymbol{\mu}(1)$, which through (4.114) produces $\tilde{\mathbf{x}}(1)$ in terms of $\mathbf{x}(t_f) - \mathbf{x}_d(t_f)$. A forward sweep of the model, to t_f, produces the numerical value of $\tilde{\mathbf{x}}(t_f)$; the backwards sweep of the adjoint model gives the corresponding numerical value of $\tilde{\mathbf{x}}(1)$, and a final forward sweep of the model completes the solution. The subproblem forward and backwards sweeps are always well-posed. This recipe was run for

$$k^2 = 0.05, \quad \Delta t = 1, \quad \tilde{\mathbf{x}}(1) = [0.805, \; 1.0]^{\mathrm{T}}, \quad \mathbf{P}(1) = 10^{-2}\mathbf{I},$$
$$\tilde{x}(t_f) = 1.427 \times 10^{-5}, \quad \mathbf{P}(t_f) = \mathrm{diag}\{\, 10^{-4} \quad 10^4 \,\}, \quad t_f = 50,$$

with results as shown in Fig. 4.17. (The large subelement uncertainty in $\mathbf{P}(50)$, corresponding to scalar element, $x(49)$, is present because we sought to specify only scalar element $x(50)$, in $\mathbf{x}(50)$.) The solution produces a new estimated value $\tilde{\mathbf{x}}(0) = [0.800, 1.00]^{\mathrm{T}}$, which is exactly the value used in Fig. 4.17 to generate the stable forward computation. Notice that the original ill-posedness in both overspecification and instability of the initial value problem have been dealt with. The Lagrange multipliers (adjoint solution) are also shown in the figure, and imply that

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

the system sensitivity is greatest at the initial and final times. For a full GCM, the technical details are much more intricate, but the principle is the same.

This example can be thought of as the solution to a forward problem, albeit ill-posed, or as the solution to a more or less conventional inverse one. The distinction between forward and inverse problems has nearly vanished. Any forward model that is driven by observed conditions is ill-posed in the sense that there can again be no unique solution, only a most probable one, smoothest one, etc. As with an inverse solution, forward calculations no more produce unique solutions in these circumstances than do inverse ones. All problems involving observed parameters, initial or boundary conditions are necessarily ill-posed.

4.9 A summary

Once rendered discrete for placement on a digital computer, time-dependent inverse problems all can be reduced to the minimum variance/least-squares problems already considered in Chapters 2 and 3, depending upon how the weight matrices are chosen. With a large enough and fast enough computer, they could even be solved by the same methods used for the static problems. Given, however, the common need to reduce the computing and storage burdens, a number of algorithms are available for finding solutions by considering the problem either in pieces, as in the sequential methods of filter/smoother, or by iterations as in the Lagrange multiplier methods. Solutions can be either accurate or approximate depending upon one's needs and resources. In the end, however, the main message is that one is still seeking the solution to a minimum variance/least-squares problem, and the differences among the techniques are algorithmic ones, with trade-offs of convenience and cost.

Appendix. Automatic differentiation and adjoints

The utility of automatic differentiation (AD) of computer model codes was alluded to on pp. 189 and 241, both as a way to determine the state transition matrix $\mathbf{A}$, when it was only implicit in a code, and as a route to linearizing non-linear models. The construction of software capable of taking (say) a Fortran90 code and automatically generating a second Fortran90 code for the requisite derivatives of the model is a remarkable, if not altogether complete, achievement of computer science. Any serious discussion is beyond the author's expertise, and well outside the scope of this book. But because only AD methods have made the Lagrange multiplier (adjoint) method of state estimation a practical approach for realistic

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

fluid problems, we briefly sketch the possibilities with a few simple examples. The references given in note 43 should be consulted for a proper discussion.

Consider first the problem of finding the state transition matrix. A simple time-stepping code written in Matlab for a 2-vector is

```
function y=lin(x);
y(1)=0.9*x(1)+0.2*x(2);
y(2)=0.2*x(1)+0.8*x(2);
```

Here **x** would be the state vector at time $t - 1$, and **y** would be its value one time-step in the future. A matrix/vector notation is deliberately avoided so that **A** is not explicitly specified. When the AD tool ADiMat[51] is used, it writes a new Matlab code:

```
function [g_y, y]= g_lin(g_x, x) %lin.m;
%x is assumed to be a 2-vector
g_lin_0= 0.9* g_x(1);
lin_0= 0.9* x(1);
g_lin_1= 0.2* g_x(2);
lin_1= 0.2* x(2);
g_y(1)= g_lin_0+ g_lin_1;
y(1)= lin_0+ lin_1;
clear lin_0 lin_1 g_lin_0 g_lin_1 ;
g_lin_2= 0.2* g_x(1);
lin_2= 0.2* x(1);
g_lin_3= 0.8* g_x(2);
lin_3= 0.8* x(2);
g_y(2)= g_lin_2+ g_lin_3;
y(2)= lin_2+ lin_3;
clear lin_2 lin_3 g_lin_2 g_lin_3 ;
```

The notation has been cleaned up somewhat to make it more readable. Consider for example, the new variable, `g_lin_0` $=$ `0.9` $*$ `g_x(1)`. The numerical value 0.9 is the partial derivative of $y\,(1)$ with respect to $x\,(1)$. The variable `g_x(1)` would be the partial derivative of $x\,(1)$ with respect to some other independent variable, permitting the chain rule to operate if desired. Otherwise, one can set it to unity on input. Similarly the notation `g_lin_i` denotes the corresponding derivative of $y(1)$ with respect to $x(i)$. By simple further coding, one can construct the **A** matrix of the values of the partial derivatives. Here, ADiMat has produced the tangent linear model, which is also the exact forward model. More interesting examples can be constructed.

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

The Matlab code corresponding to a simple switch is

```
function y= switch1(a); if a > 0, y= a; else, y=
  a^2+2*a; end
```

that is, $y = a$ if independent variable a (externally prescribed) is positive, or else $y = a^2 + 2a$. Running this code through AdiMat produces (again after some cleaning up of the notation):

```
function [g_y, y]= g_switch1(g_a, a);
 if a> 0, g_y= g_a;
 y= a;
else, g_tmp_2=2* a^(2- 1)* g_a;
   tmp_0= a^2;
   g_tmp_1= 2* g_a;
   tmp_1= 2* a;
   g_y= g_tmp_0+ g_tmp_1;
   y= tmp_0+ tmp_1;
   end
```

The derivative, `g_y`, is 1 for positive a, otherwise it is `a+2.` `g_a` can be interpreted as the derivative of a with respect to another, arbitrary, independent variable, again permitting use of the chain rule. The derivative is continuous for all values of a except $a = 0$.

Consider now a physical problem of a reservoir as shown in Fig. 4.18.[52] The model is chosen specifically to have discontinuous behavior: there is inflow, storage, and outflow. If the storage capacity is exceeded, there can be overspill, determined by the `max` statement below. A forward code, now written in Fortran, is:

Thresholds: a hydrological reservoir model (I)

```
do t = 1, msteps
```

- get sources and sinks at time t:

inflow, evaporation, release (*read fields*)

- calculate water release based on storage:

`release(t) = 0.8*storage(t-1)`$^{**0.7}$

- calculate projected stored water:

storage = *storage* + *inflow-release-evaporation*

```
nominal = storage(t-1) +
   h*( infl(t)-release(t)-evap(t) )
```

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

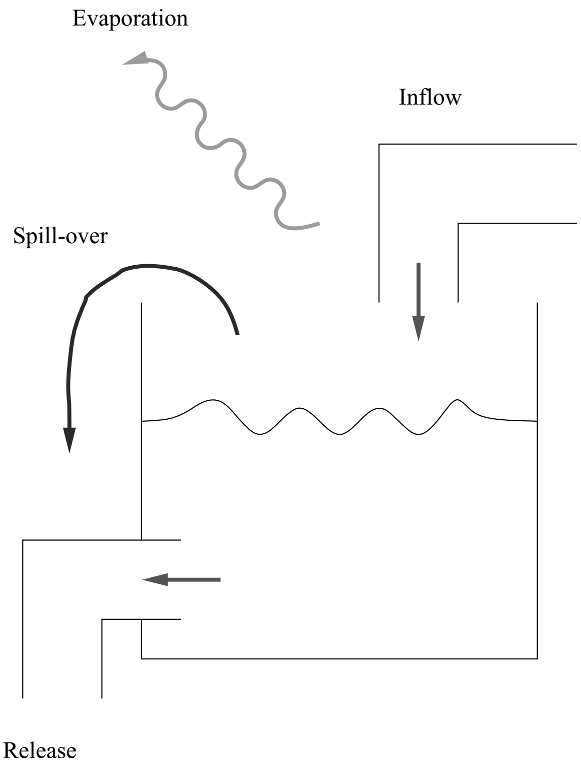

Figure 4.18 Reservoir model used to demonstrate automatic/algorithmic differentiation. The possibility of spill-over is an example of a switch in a computer model. (From P. Heimbach, personal communication, 2004.)

- If threshold capacity is exceeded, spill-over:

```
spill(t) = MAX(nominal-capac , 0.)
```

- re-adjust projected stored water after spill-over:

```
storage(t) = nominal - spill(t)
```

- determine outflow:

```
out(t) = release(t) + spill(t)/h

end do
```

Note the presence of the `max` statement.

When run through the AD tool TAF (a product of FastOpt®), one obtains for the tangent linear model,

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009

Thresholds: a hydrological reservoir model (II)

The tangent linear model

```
do t=1, msteps
g_release(t) =0.56*g_storage(t-1)*storage(t-1)**
  (-0.3)
release(t)=0.8*storage(t-1)**0.7
g_nominal=-g_release(t)*h+g_storage(t-1)
nominal=storage(t-1)+h*(infl(t)-release(t)-evap(t))
g_spill(t)=g_nominal*(0.5+sign(0.5,nominal-
  capac-0.))
spill(t)=max(nominal-capac,0.)
g_storage(t)=g_nominal-g_spill(t)
storage(t)=nominal-spill(t)
g_out(t)=g_release(t)+g_spill(t)/h
out(t)=release(t)+spill(t)/h
end do
```

- `g_release(t)` not defined for `storage(t-1) = 0`
- `g_spill(t)` not defined for `nominal = capac`.

Note how the maximum statement has given rise to the new variable `g_spill(t)`, its corresponding tangent linear variable.

Note that the AD tool can cope with such apparently non-differentiable operators as the maximum of a vector. In practice, it internally replaces the function `max` with a loop of tests for relative sizes of successive elements. Not all AD tools can cope with all language syntaxes, not all all structures are differentiable, and one must be alert to failures owing to incomplete handling of various structures. Nonetheless, the existing tools are a considerable achievement.

TAF and some other AD tools are capable of producing the reverse mode. A major issue in optimization calculations is the ability to restart the computation from intermediate results, in an operation called "checkpointing."[53]

Notes

1 Liebelt (1967), Gelb (1974), Bryson and Ho (1975), Anderson and Moore (1979), and Brown and Hwang (1997) are especially helpful.
2 Daley (1991).
3 Meteorologists have tended to go their own idiosyncratic way – see Ide *et al.* (1997) – with some loss in transparency to other fields.
4 Box *et al.* (1994).
5 Luenberger (1979).
6 Stammer and Wunsch (1996), Menemenlis and Wunsch (1997).
7 von Storch *et al.* (1988).
8 Giering and Kaminski (1998), Marotzke *et al.* (1999).
9 See Bryson and Ho (1975, p. 351).
10 For example, Stengel (1986).
11 Munk *et al.* (1995).

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005

Cambridge Books Online © Cambridge University Press, 2009

12 A method exploited by Stammer and Wunsch (1996).
13 Kalman (1960). Kalman's derivation was for the discrete case. The continuous case, which was derived later, is known as the "Kalman–Bucy" filter and is a much more complicated object.
14 Stengel (1986, Eq. 4.3–22).
15 For example, Goodwin and Sin (1984, p. 59).
16 Feller (1957).
17 Anderson and Moore (1979) discuss these and other variants of the Kalman filter equations.
18 Some history of the idea of the filter, its origins in the work of Wiener and Kolmogoroff, and a number of applications, can be found in Sorenson (1985).
19 Bryson and Ho (1975, p. 363) or Brown and Hwang (1997, p. 218).
20 Adapted from Bryson and Ho (1975, Chapter 13), whose notation is unfortunately somewhat difficult.
21 For Rauch *et al.* (1965).
22 Gelb (1974), Bryson and Ho (1975), Anderson and Moore (1979), Goodwin and Sin (1984), Sorenson (1985).
23 Some guidance is provided by Bryson and Ho (1975, pp. 390–5) or Liebelt (1967). In particular, Bryson and Ho (1975) introduce the Lagrange multipliers (their equations 13.2.7–13.2.8) simply as an intermediate numerical device for solving the smoother equations.
24 Luenberger (1979).
25 Wunsch (1988) shows a variety of calculations as a function of variations in the terminal constraint accuracies. An example of the use of this type of model is discussed in Chapter 6.
26 Bennett (2002) has a comprehensive discussion, albeit in the continuous time context.
27 The use of the adjoint to solve l_2-norm problems is discussed by Bryson and Ho (1975, Section 13.3), who relax the restriction of full controllability, $\mathbf{\Gamma} = \mathbf{I}$. Because of the connection to regulator/control problems, a variety of methods for solution is explored there.
28 Bryson and Ho (1975).
29 Franklin *et al.* (1998).
30 Anderson and Moore (1979), Stengel (1986), Bittanti *et al.* (1991), Fukumori *et al.* (1992), Fu *et al.* (1993), Franklin *et al.* (1998).
31 See Reid (1972) for a discussion of the history of the Riccati equation in general; it is intimately related to Bessel's equation and has been studied in scalar form since the eighteenth century. Bittanti *et al.* (1991) discuss many different aspects of the matrix form.
32 Discussed by Bittanti *et al.* (1991).
33 Franklin *et al.* (1998).
34 For example, Stengel (1986), Franklin *et al.* (1998).
35 Goodwin and Sin (1984) or Stengel (1986).
36 Miller *et al.* (1994) discuss some of the practical difficulties.
37 For example, Miller *et al.* (1994).
38 For example, Nayfeh (1973).
39 For example, Lea *et al.* (2000), Köhl and Willebrand (2002).
40 For example, Anderson and Moore (1979), Goodwin and Sin (1984), Haykin (2002).
41 Stengel (1986, Chapter 5).
42 Gilbert and Lemaréchal (1989).
43 Giering and Kaminski (1998), Marotzke *et al.* (1999), Griewank (2000), Corliss *et al.* (2002), Heimback *et al.* 2005.
44 See Marotzke *et al.* (1999) or Giering (2000).
45 Rall (1981) and Griewank (2000).
46 We follow here, primarily, Marotzke *et al.* (1999).
47 Restrepo *et al.* (1995) discuss some of the considerations.
48 Luenberger (1984), Scales (1985), Gill *et al.* (1986), Tarantola (1987).
49 Gelb (1974).
50 O'Reilly (1983), Luenberger (1984).
51 Available through www.sc.rwth-aachen.de/vehresschild/adimat.
52 Example due to P. Heimbach. For more information see http://hdl.handle.net/1721.1/30598.
53 Restrepo *et al.* (1995).

Downloaded from https://www.cambridge.org/core. University of Victoria Libraries, on 29 Jul 2020 at 00:49:01, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9780511535949.005
Cambridge Books Online © Cambridge University Press, 2009