4.2 Basic Ideas and Notation

4.2.1 Models

In the context of this chapter, by "models" is meant 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 Schrodinger, 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. So for example, the time evolution of a system described by the tracer diffusion equation (6.1), 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 $\Delta t \rightarrow 0$.(external information). $C(\mathbf{r},t)$ is the state variable (the "internal" information), with the boundary conditions being regarded as separate externally provided variables (but the distinction is, as we will see, to some degree an arbitrary one). In practice, such quantities as initial and boundary conditions, container shape, etc. are typically obtained from measurements, are thus always imperfectly known, and 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" $\mathbf{x}(t)$ (t 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, $\mathbf{x} = \mathbf{vec}(C_{ij})$ is the state vector. In a fluid model, the state vector might consist of three components of velocity, pressure and

temperature at 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,

{xbig}
$$\mathbf{x}_B = \left[\mathbf{x}\left(1\Delta t\right)^T, \mathbf{x}\left(2\Delta t\right)^T, .., \mathbf{x}\left(T\Delta t\right)^T\right]^T, \tag{4.7}$$

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

Consider a partial differential equation,

{61002}
$$\frac{\partial}{\partial t}(\nabla_h^2 p) + \beta\frac{\partial p}{\partial \eta} = 0\,, \tag{4.8}$$

subject to boundary conditions. For the moment, t is a continuous variable. Suppose it is solved by an expansion,

{61003}
$$p(\xi,\, \eta,\, t) = \sum_{j=1}^{N/2} a_j(t)\cos(\mathbf{k}_j\cdot\mathbf{r}) + b_j(t)\sin(\mathbf{k}_j\cdot\mathbf{r}). \tag{4.9}$$

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

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

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 then by $\sin(\mathbf{k}_j\cdot\mathbf{r})$ and integrate over the domain:

$$\begin{aligned} -\left|\mathbf{k}_j\right|^2\dot{a}_j + \beta k_{1j} b_j &= 0 \\ \left|\mathbf{k}_j\right|^2\dot{b}_j + \beta k_{1j} a_j &= 0 \end{aligned}$$

or

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

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

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

The state vector is then the collection,

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

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,

$$\mathcal{L}\left[\mathbf{x}(0), \ldots, \mathbf{x}(t-\Delta t),\, \mathbf{x}(t),\, \mathbf{x}(t+\Delta t), \ldots \mathbf{x}(t_f) \ldots,\, \mathbf{B}(t)\mathbf{q}(t)\,, \mathbf{B}(t)\mathbf{q}(t+\Delta t), \ldots, t\right] = 0 \quad (4.10) \quad \{61004\}$$

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,

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (4.11) \quad \{61005\}$$

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

$$\mathbf{x}(t) = \mathbf{L}\big(\mathbf{x}(t-1),\, \mathbf{B}(t-1)\mathbf{q}(t-1),\, t-1\big), \quad \Delta t = 1, \quad (4.12) \quad \{61006\}$$

or, if the model is linear,

$$\mathbf{x}(t) = \mathbf{A}(t-1)\,\mathbf{x}(t-1) + \mathbf{B}(t-1)\,\mathbf{q}(t-1)\,. \quad (4.13) \quad \{61007\}$$

If the model is time invariant, $\mathbf{A}(t) = \mathbf{A}$, and $\mathbf{B}\,(t) = \mathbf{B}$. 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.

Example.

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

$$\frac{d^2\xi}{dt^2} = 0\,, \quad (4.14) \quad \{61009\}$$

which can be discretized as

$$\xi(t+\Delta t) - 2\xi(t) + \xi(t-\Delta t) = 0\,, \quad (4.15) \quad \{61010\}$$

Define

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

and $t \to n\Delta t$. *One has*

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

where

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

{sl1}

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

emassspring}

Example.

The elementary mass-spring oscillator satisfies the differential equation

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

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

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

or

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

which is

$$\begin{bmatrix} \xi(t) \\ \xi(t - \Delta t) \end{bmatrix} = \begin{Bmatrix} 2 - \frac{r}{m}\Delta t - \frac{k}{m}(\Delta t)^2 & \frac{r\Delta t}{m} - 1 \\ 1 & 0 \end{Bmatrix} \begin{bmatrix} \xi(t - \Delta t) \\ \xi(t - 2\Delta t) \end{bmatrix} + \begin{bmatrix} (\Delta t)^2\,\frac{q(t - \Delta t)}{m} \\ 0 \end{bmatrix}, \tag{4.17}$$

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

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

Example

A difference equation important in time-series analysis[104] *is,*

{61018}

$$\xi(t) + a_1\xi(t - 1) + a_2\xi(t - 2) + \cdots + a_N\xi(t - N) = \eta(t) \tag{4.18}$$

where $\eta(t)$ is a zero-mean, white-noise process (Equation (4.18) *is an example of an autoregressive process (AR)). To put this into the canonical form, write*[105]*,*

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

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

$$\mathbf{x}(t) = \begin{Bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ -a_N & -a_{N-1} & -a_{N-2} & \cdots & -a_2 & -a_1 \end{Bmatrix} \mathbf{x}(t-1) + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 1 \end{bmatrix} \eta(t-1)\,. \qquad (4.19) \quad \texttt{\{61020\}}$$

A *is known as a "companion" matrix. Equation* (4.19) *connects this Chapter with the field of time-series analysis. Here,* $\mathbf{B}(t) = [0 \quad 0 \quad \cdot \quad 1]^T$, $\mathbf{q}(t) = \eta(t)$.

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,

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

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 p depicts the time history for the harmonic oscillator,

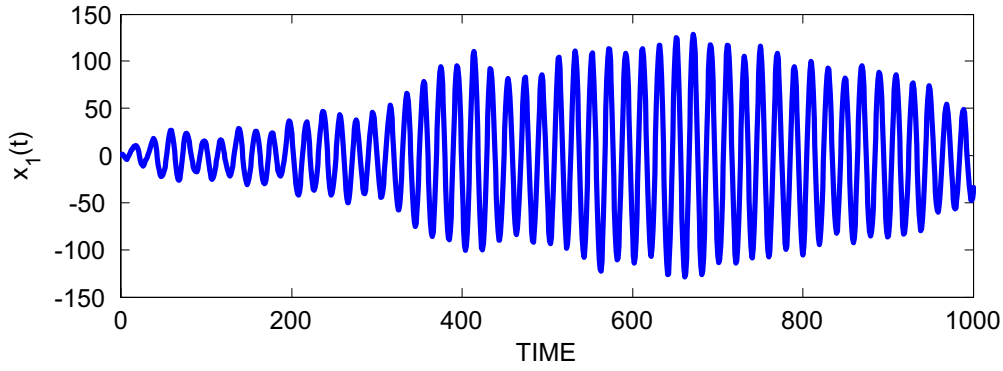

Figure 4.1: Time history of $x_1(t)$ for the linear oscillator with $\Delta t = 1, k = 0.1, m = 1, r = 0$ driven by a random sequence of zero mean and unit variance. Note buildup in amplitude from the accumulating uncorrelated forcing increments. {linoscts.eps}

with the parameter choice $\Delta t = 1$, $k = 0.1$, $m = 1$, $r = 0$, so that,

$$\mathbf{A} = \begin{Bmatrix} 1.9 & -1 \\ 1 & 0 \end{Bmatrix}, \qquad \mathbf{Bq}(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t),$$

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

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 a generalized inverse can be used if necessary. 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

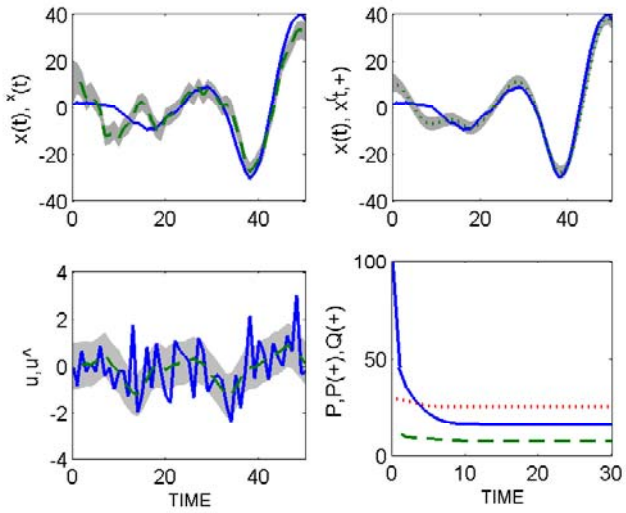

Figure 4.2: (a) Forward run (solid line) of a forced mass-spring oscillator with $r = 0$, $k = 0.1$ with initial condition $\mathbf{x}(0) = [1, 0]^T$. Dotted line is a Kalman filter estimate, started with $\tilde{\mathbf{x}}(0) = [10, 10]$, $\mathbf{P}(0) = \text{diag}([100, 100])$. "Observations" were provided of $x_1(t)$ at every time step, but corrupted with white noise of variance $R = 50$. 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) Dotted line now shows $\tilde{x}_1(t, +)$ from the RTS smoothing algorithm. 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. 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. 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, smoothing out the underlying rapid variations. (d) 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.

{kfrts1.eps}

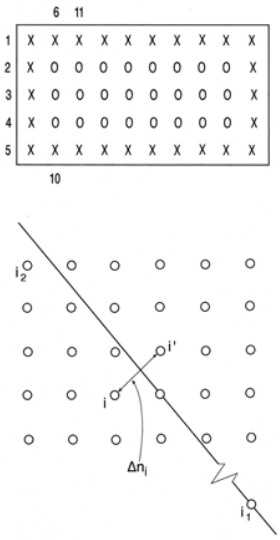

Figure 4.3: (a) Simple numerical grid for use of discrete form of model; × denote boundary grid points, and o are interior ones. Numbering is sequential down the columns, as shown. (b) Tomographic integral is assumed given between i_1, i_2, and the model values at the grid points would be used to calculate its predicted value.

{ocip6_2.tif}

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 Figure 4.3a. 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,

{61022}
$$\mathbf{B} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \cdots 1 \quad 1]^T, \tag{4.20}$$

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

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

{61023}
$$\mathbf{Bq}(t) = \begin{Bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \cdot & 0 \end{Bmatrix}^T \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}. \tag{4.21}$$

A time-dependent $\mathbf{B}$ 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 $t = 0$.

As with steady models, we need to be careful about 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 we are currently measuring. 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, one can qualitatively distinguish the behavior of the errors through time. (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 computed by the model.

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 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 models of the form (4.13) can be written in the "whole-domain" form,

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

$$\mathbf{A}_B = \begin{Bmatrix} -\mathbf{A} & \mathbf{I} & \mathbf{0} & \cdot & \cdot & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{I} & \mathbf{0} & \cdot & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & -\mathbf{A} & \mathbf{I} \end{Bmatrix}, \quad \mathbf{d}_B = \begin{bmatrix} \mathbf{B}\mathbf{q}(0) \\ \mathbf{B}\mathbf{q}(1) \\ \vdots \end{bmatrix}, \tag{4.22}$$

{big1}

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 $\mathbf{x}(t'), t' \neq 0$, the whole-domain form may be very attractive. Note the block-banded nature of $\mathbf{A}_B$ and the 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 state transition matrix is not normally explicitly constructed; instead,the individual elements of $x_i(t)$ are time-stepped to produce $x_i(t+1)$, 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,

{62021b}

$$\mathbf{x}(t) = \mathbf{G}(t,0)\mathbf{x}(0)\,, \tag{4.23}$$

{pageunitsol}

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 $\mathbf{x}_0^{(i)}$, $1 \leq i \leq N$, 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 $\mathbf{X}_0$:

{A11}
$$\mathbf{A} = \mathbf{X}_1 \mathbf{X}_0^{-1}. \tag{4.24}$$

The inverse will exist by the assumption of independence (spanning set) in the initial condition vectors. One must again run the model $N-$times in this approach. A special, useful, case would be when $\mathbf{x}_o^{(i)} = \delta_{ij}$ (unit perturbation initial conditions), in which case the solution $\mathbf{X}_1 = \mathbf{G}$.

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,

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

which would determine $\mathbf{A}^n$, and $\mathbf{A}$ itself can be found by one of the matrix root algorithms.[106]

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

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

perhaps because the model has been run many times from different initial conditions—making it possible to estimate these from stored output. Then noting,

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

or

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

and,

$$\mathbf{A} = \mathbf{R}(1)\, \mathbf{R}(0)^{-1}. \tag{4.25}$$ {A22}

That is to say, knowledge of these covariances is equivalent to knowledge of the model itself (and vice-versa).[107] Multiple time steps can again be used if necessary to infer $\mathbf{A}^n = \mathbf{R}(n)\, \mathbf{R}(0)^{-1}$. By writing $\left\langle \mathbf{x}(t+1)\, \mathbf{x}(t)^T \right\rangle = \mathbf{R}(1)$, etc. stationarity is implied. More generally, one may have $\left\langle \mathbf{x}(t+1)\, \mathbf{x}(t)^T \right\rangle = \mathbf{R}(t,1)$.

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 differentiation (AD) tools exist[108] which 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 $\mathbf{A}^T$, but transposition then can be employed. An example is provided in the Chapter Appendix. {pagead1}

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, so defining an $\bar{\mathbf{A}}$ as,

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

Reintroduction of $\mathbf{B}$ is easily accomodated. Generalization of Eq. (4.25) leads to

$$\bar{\mathbf{A}}^n = \mathbf{R}(n)\,\mathbf{R}(0)^{-1}\,. \tag{4.26}$$ {A33}

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 $t = n\Delta t$,

{61025}
$$\mathbf{E}(t)\,\mathbf{x}(t) + \mathbf{n}(t) = \mathbf{y}(t)\,, \tag{4.27}$$

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 nonlinear relation between $y_i(t)$ and the state vector. Such systems are usually handled by some form of linearization.[109]

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

{61026}
$$\langle \mathbf{n}(t) \rangle = 0, \qquad \langle \mathbf{n}(t)\,\mathbf{n}(t)^T \rangle = \mathbf{R}(t)\,. \tag{4.28}$$

But

{61027}
$$\langle \mathbf{n}(t)\,\mathbf{n}(t')^T \rangle = \mathbf{0}, \quad t \neq t'\,. \tag{4.29}$$

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[110]). Unless specifically stated otherwise, we will assume that (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[111] as described in Chapter 1—are explicitly spatial averages (integrals) obtained by measuring some property

along a ray travelling between two points; see Fig. 4.3b. Any such data representing spatially filtered versions of the state vector can be written,

{61028}
$$y(t) = \sum \alpha_j x_j(t)\,. \tag{4.30}$$

where the α_j are the averaging weights.

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) demands interpolation from the dense model grid to the sparse data positions; see Fig. 4.3b. (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, Eq. (4.22), the observations of the combined state vector can be concatenated into a single observational set,

$$\mathbf{E}_B\mathbf{x}_B + \mathbf{n}_B = \mathbf{y}_B, \tag{4.31}$$ {61030}

where

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

Here the initial conditions have been combined with the observations. $\mathbf{E}_B$ is block-banded and often very sparse. If the size is no problem, the 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. 200 leads to a particularly simple reduced form. One has immediately,

$$\mathbf{y}(t) = \mathbf{E}(t)\mathbf{G}(t,0)\mathbf{x}(0) + \mathbf{n}(t)\,, \quad 1 \le t \le t_f\,, \tag{4.32}$$ {green1}

which are readily solved in whole-domain form for $\mathbf{x}(0)$. If only a subset of the $\mathbf{x}(0)$ are thought to be nonzero, the columns of $\mathbf{G}$ need to be computed only for those elements.[112]